



## Troubleshooting the Pentaho BI Suite: Common Problems and Solutions



This document is copyright © 2011 Pentaho Corporation. No part may be reprinted without written permission from Pentaho Corporation. All trademarks are the property of their respective owners.

## Help and Support Resources

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

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

For information about how to purchase support or enable an additional named support contact, please contact your sales representative, or send an email to [sales@pentaho.com](mailto: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](mailto:communityconnection@pentaho.com)

Sales Inquiries: [sales@pentaho.com](mailto:sales@pentaho.com)

Documentation Suggestions: [documentation@pentaho.com](mailto:documentation@pentaho.com)

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

# Contents

Introduction.....	5
General.....	6
File Names and Paths.....	6
JDBC Driver Problems.....	6
Adding a JDBC Driver.....	6
Version Check.....	7
I don't know what the default login is for the DI Server, Enterprise Console, and/or Carte7	
Examining Log Files.....	7
Upgrade.....	9
Tomcat Logs Report Memory Leaks.....	9
context.xml Changes Do Not Take Effect After Re-deploying a WAR.....	9
javax.jcr.RepositoryException: no search manager configured for this workspace.....	10
Upgrading a Data Integration Server.....	10
User Console Themes Render Improperly After Upgrade.....	11
BI Server and Pentaho Enterprise Console.....	12
Address Already In Use: JVM_Bind When Starting Enterprise Console.....	12
Library Conflicts.....	12
Report Parameters That Include Accented Characters Fail to Validate.....	12
Modifying server.xml To Work With Accented Characters.....	12
vfs-provider.xml Duplicates.....	13
Varying Context and Data Source Configuration Methods.....	13
Licenses Not Found After Installation.....	13
Evaluation Licenses are Not Granted.....	13
Cannot Create Hibernate Tables in MySQL.....	14
Case-Insensitivity for Usernames, Passwords, and Filenames.....	14
Unable to Use the Database Init Scripts for PostgreSQL.....	14
HTTP500or"UnabletoconnecttoBIServer"ErrorsWhenTryingtoAccessEnterpriseConsole	
Configuring the Proxy Trusting Filter.....	14
JBoss Fails to Start When the Pentaho HSQLDB Sample Database Is Running.....	15
JBoss Fails to Start After Manually Unpacking pentaho.war.....	15
Web-App Path doesn't validate in Enterprise Console.....	15
Report Designer.....	16
Enabling Multi-Value Parameters for Metadata-based Queries Created in Previous Version of Report Designer	
Hive Database Disappears From Database Connection Dialog Box.....	16
Report Elements With Dynamic Heights Overlap Other Elements.....	16
Reports Using Hive Metadata Data Sources Stop Working.....	16
Analysis.....	18
Old Analysis Schemas Still Show Up in Pentaho User Console.....	18
Removing Mondrian Data Sources.....	18
Multi-Byte Characters Don't Appear In PDFs Exported From Analyzer.....	19
Setting a Default Font for PDF Exports.....	19

Data Integration.....	20
Action Sequences That Call PDI Content Won't Run.....	20
Adding PDI Enterprise Repository Content Support to the BI Server.....	20
Jobs scheduled on the DI Server cannot execute a transformation on a remote Carte server.....	20
Executing Scheduled Jobs on a Remote Carte Server.....	21
Metadata.....	23
Managing Multiple Outer-Joins.....	23
Using the Delay Outer Join Conditions Property .....	25

# Introduction

---

This document contains no unique content; it is a compilation of all of the troubleshooting content contained in Pentaho Enterprise Edition documentation. Each guide has its own troubleshooting section, where applicable, that contains problems and solutions that Pentaho customers and partners have encountered in the past or are anticipated to encounter in the future. The *Troubleshooting Guide* is designed to address customers, partners, consultants, and internal Pentaho employees who are already familiar with the installation, configuration, and operation of the BI Suite, and only need to consult the documentation to solve a problem.

# General

---

This section contains problems and solutions that apply to the BI Suite in general.

## 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.

## 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 BI 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 BI Suite Administrator's Guide* that explains driver locations for all parts of the BI Suite.



**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.

## Adding a JDBC Driver

Before you can connect to a data source in any Pentaho server or client tool, you must first install the appropriate database driver. Your database administrator, CIO, or IT manager should be able to provide you with the proper driver JAR. If not, you can download a JDBC driver JAR file from your database vendor or driver developer's Web site. Once you have the JAR, follow the instructions below to copy it to the driver directories for all of the BI Suite components that need to connect to this data source.



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

### Backing up old drivers

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

### Installing JDBC drivers

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



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

- **BI Server:** `/pentaho/server/biserver-ee/tomcat/lib/`

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



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

## Restarting

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

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

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

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

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



**Note:** Use the **sqljdbc\_auth.dll** file, in the x86 folder, if you are running a 32-bit Java Virtual Machine (JVM) even if the operating system is version x64. Use the **sqljdbc\_auth.dll** file in the x64 folder, if you are running a 64-bit JVM on a x64 processor. Use the **sqljdbc\_auth.dll** file in the IA64 folder, you are running a 64-bit JVM on an Itanium processor.

## Version Check

---

The instructions in this guide are specific to the Pentaho BI Server Enterprise Edition version 4.0.0-GA. The installation process can change significantly between BI 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 BI Server version, including the open source BI Server Community Edition version 4.0-stable.

## I don't know what the default login is for the DI Server, Enterprise Console, and/or Carte

---

For the DI Server administrator, it's username **admin** and password **secret**.

For Enterprise Console administrator, it's username **admin** and password **password**.

For Carte, it's username **cluster** and password **cluster**.

Be sure to change these to new values in your production environment.



**Note:** DI Server users are not the same as BI Server users.

## Examining Log Files

---

If the BI 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.



# Upgrade

---

This section contains problems and solutions that apply to the Upgrade Guide for both the BI Server and Pentaho Data integration.

## Tomcat Logs Report Memory Leaks

---

When shutting down Tomcat, you may see some SEVERE-level warnings in the log file similar to these:

```
Dec 17, 2010 10:18:19 AM org.apache.catalina.loader.WebappClassLoader
clearReferencesJdbc
SEVERE: The web application [/pentaho] registered the JDBC driver
[mondrian.olap4j.MondrianOlap4jDriver] but failed to unregister it when the
web application was stopped. To prevent a memory leak, the JDBC Driver has
been forcibly unregistered.
Dec 17, 2010 10:18:19 AM org.apache.catalina.loader.WebappClassLoader
clearReferencesThreads
SEVERE: The web application [/pentaho] appears to have started a thread named
[HSQldb Timer @49cf9f] but has failed to stop it. This is very likely to
create a memory leak.
Dec 17, 2010 10:18:19 AM org.apache.catalina.loader.WebappClassLoader
clearReferencesThreads
SEVERE: The web application [/pentaho] appears to have started a thread named
[MySQL Statement Cancellation Timer] but has failed to stop it. This is very
likely to create a memory leak.
Dec 17, 2010 10:18:19 AM org.apache.catalina.loader.WebappClassLoader
clearThreadLocalMap
SEVERE: The web application [/pentaho] created a ThreadLocal
with key of type [java.lang.InheritableThreadLocal] (value
[java.lang.InheritableThreadLocal@a1320e]) and a value of type
[org.pentaho.platform.engine.security.session.TrustedSystemStartupSession]
(value
[org.pentaho.platform.engine.security.session.TrustedSystemStartupSession@111089b])
but failed to remove it when the web application was stopped. This is very
likely to create a memory leak.
```

These warnings are nothing to be concerned about when shutting down the Tomcat server, since they report problems with processes that are immanently being killed. However, they can have significance if you are only restarting or redeploying the Pentaho BI Server or DI Server Web applications. To avoid any memory leak problems in redeployment, you should restart Tomcat instead of redeploying or restarting the Web application with a live server.

## 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.

## javax.jcr.RepositoryException: no search manager configured for this workspace

If you see this error in your PDI server log, then there was an error in the upgrade process from PDI 4.1.3 to 4.2.0. Specifically, the **SearchIndex** XML nodes were not properly modified. To fix this, refer to the **Upgrading a Data Integration Server** piece and closely follow the instructions for modifying repository configuration files.


### Upgrading a Data Integration Server

**Ensure that the DI Server and Pentaho Enterprise Console are stopped before continuing.** You must have a PDI 4.1 DI Server installed in order to follow this procedure; if you do not use the DI Server, this upgrade task is unnecessary.

 **Note:** For a smoother post-upgrade test experience, you should perform this procedure before upgrading your PDI workstations.

Follow the instructions below to upgrade your Data Integration Server to version 4.2.

1. Rename the `/data-integration-server/` directory to **data-integration-server-old**.

 **Note:** If you are coming from a BI Server upgrade, you already have a **server\_old** directory. If this is the case, use `/server_old/data-integration-server/` in place of `/data-integration-server-old/`.

```
mv /home/pentaho/pentaho/server/data-integration-server/ /home/pentaho/pentaho/server/data-integration-server-old/
```

2. Unpack the **pdi-ee-server-4.2.0-GA** package to the parent of the directory you just renamed.

```
tar zxvf ~/downloads/pdi-ee-server-4.2.0-GA.tar.gz -C /home/pentaho/pentaho/server/
```

3. Copy all of the **applicationContext** files from the `/data-integration-server-old/pentaho-solutions/system/` directory to the new one, overwriting the equivalent files that are already there.

```
cp applicationContext-* ~/pentaho/server/data-integration-server/pentaho-solutions/system/
```

4. Copy the **pentaho-spring-beans.xml** file from the `/data-integration-server-old/pentaho-solutions/system/` directory to the new one, overwriting the equivalent file that is already there.

```
cp pentaho-spring-beans.xml ~/pentaho/server/data-integration-server/pentaho-solutions/system/
```

5. Transfer the information about the **admin role** from the following two old files to the new ones: **/pentaho-solutions/system/pentaho.xml** and **/pentaho-solutions/system/repository.spring.xml**

```
<acl-voter>
<!-- What role must someone be in to be an ADMIN of Pentaho -->
  <admin-role>Admin</admin-role>
</acl-voter>
```

```
<!-- The name of the authority which is granted to all admin users in
single-tenancy mode. -->
<bean id="singleTenantAdminAuthorityName" class="java.lang.String">
  <constructor-arg value="Admin" />
</bean>
```

6. Copy the entire old **quartz** directory from `/data-integration-server-old/pentaho-solutions/` to the new one.

```
cp -r ./quartz ~/pentaho/server/data-integration-server/pentaho-solutions/
```

7. Copy the entire old **repository** directory from `/data-integration-server-old/pentaho-solutions/system/jackrabbit/` to the new one.

```
cp -r ./jackrabbit/repository/ ~/pentaho/server/data-integration-server/pentaho-solutions/system/jackrabbit/
```

8. Copy the old `/pentaho-solutions/system/jackrabbit/repository.xml` file to the new jackrabbit directory.

```
cp ./jackrabbit/repository.xml ~/pentaho/server/data-integration-server/
pentaho-solutions/system/jackrabbit/
```

9. Open the **repository.xml** file with a text editor.
10. In `repository.xml`, uncomment all **SearchIndex** nodes.
11. In `repository.xml`, delete all **param** elements in all **SearchIndex** nodes except for the **path** param.

An example of what an original, unmodified node might look like:

```
<SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
  <param name="path" value="{wsp.home}/index"/>
  <param name="textFilterClasses" value="..." />
  <param name="extractorPoolSize" value="2" />
  <param name="supportHighlighting" value="true" />
</SearchIndex>
```

And this is what it would look like after you've performed this modification:

```
<SearchIndex class="org.apache.jackrabbit.core.query.lucene.SearchIndex">
  <param name="path" value="{wsp.home}/index"/>
</SearchIndex>
```

12. Save your changes to `repository.xml`, then perform the same modifications to the following two files in your new solutions directory:

1. `/jackrabbit/repository/workspaces/default/workspace.xml`
2. `/jackrabbit/repository/workspaces/security/workspace.xml`

13. Copy the **scripts** directory from `/data-integration-server-old/` directory to the new `data-integration-server` directory.



**Note:** The `scripts` directory will only exist if you installed PDI from a graphical installation utility. If you installed via archive packages, it won't be there. If you do not see a `scripts` directory, then skip this step.

14. Copy the entire **jre** directory from `/data-integration-server-old/` to the new one.

```
cp -r jre ~/pentaho/server/data-integration-server/
```

This step is optional. If you already have a supported JRE or JDK installed on your system, you can skip copying this directory and simply ensure that you have a `JAVA_HOME` or `PENTAHO_JAVA_HOME` system variable that points to your Java instance.

15. If you have not already done so, merge any custom changes you have made to DI Server configuration files from the old ones to the new ones.

Your DI Server is now upgraded to version 4.2. Continue on to the next subsection to upgrade the Pentaho Enterprise Console.

## User Console Themes Render Improperly After Upgrade

If you are seeing strange rendering problems in the Pentaho User Console shortly after performing a BI Server upgrade, the problem may be related to old JavaScript files being held in the browser cache. To fix this problem, clear your Web browser's cache, then reload the Pentaho User Console.

# BI Server and Pentaho Enterprise Console

---

This section contains problems and solutions that pertain to the BI Server and Pentaho Enterprise Console.

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

---

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

## Library Conflicts

---

The BI Server relies on many third-party libraries that provide everything from database connectivity to specific Java classes that add necessary features to the BI Server. If you have incompatible versions of any of these third-party libraries in your application server's global lib directory, they can cause a variety of problems related to starting and running the BI Server. You will have to discover and individually canonicalize these files according to your needs.

Some known-problematic JARs are:

- commons-collections-3.2.jar (from Pentaho)
- commons-collections.jar (from JBoss in `/jboss/server/default/lib/`)
- jettison-1.01.jar (from Pentaho)
- jettison.jar (from JBoss in `/jboss/default/deploy/jbossws.sar`)

## Report Parameters That Include Accented Characters Fail to Validate

---

If you run a report that has parameters that include accented characters and you see an error message that says "This parameter value is of an invalid value," then you must make the Tomcat server modification explained in this task: [Modifying server.xml To Work With Accented Characters](#) on page 12.

### Modifying server.xml To Work With Accented Characters

This procedure is only necessary if you plan to use character sets that include accents, such as Spanish and French.

Follow the instructions below to implement accented character set support. Change the paths to match your configuration.

1. Stop the Tomcat service.

```
sudo /etc/init.d/tomcat stop
```

2. Open the `/tomcat/server/conf/server.xml` file with a text editor.

3. Locate each **Connector** node (typically there are four in a default Tomcat configuration) and add a **URIEncoding="UTF-8"** parameter to it, as shown below:

```
<Connector URIEncoding="UTF-8" port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

4. Save and close the file, then restart the Tomcat service.

```
sudo /etc/init.d/tomcat start
```

Tomcat is now properly configured to handle accented characters.

## vfs-provider.xml Duplicates

---

The above-referenced configuration file may be present in a number of JARs in other applications that you've deployed to your Java application server. Having multiple instances of this file will cause classpath exceptions. You must merge the multiple files into one canonical edition in order to solve the problem.

## Varying Context and Data Source Configuration Methods

---

The instructions in this guide direct you to edit a **context.xml** override for Tomcat and build it into the WAR. Your application server may not support this method, or may not accept these files by default. This will be especially true if you are using unsupported application servers or versions, or if you are on Linux and your distribution provider alters the default package configuration for Tomcat.

There are other methods of changing context settings, especially through other kinds of configuration files such as **server.xml** or application-specific XML files that your application server creates for each deployed WAR. If the context and data source instructions in this guide do not work for you, consult your application server's documentation to learn the preferred method of configuration, and adapt the Pentaho-supplied process to accommodate it.

## Licenses Not Found After Installation

---

If you've successfully installed the BI 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 BI 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 BI 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
```

## Evaluation Licenses are Not Granted

---

If you see a message in the graphical installation utility's error log that says, **Evaluation licenses were not granted for the following products**, this means that you've installed this version of the BI Suite (or some piece of it) in the past, and one or more evaluation licenses have already been generated and installed. An evaluation license expires 30 days after it is generated; once expired, it cannot be reactivated. The only way to use the BI Suite's full functionality is to install a new, active license that overwrites the old one.

In some cases, you may have delayed or extended your Pentaho evaluation beyond the automatic license's term. If you need more time, contact your Pentaho sales representative or pre-sales support engineer.

## 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.

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

---

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

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

## 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 to 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.

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

---

If you have trouble controlling the BI Server through the Pentaho Enterprise Console, it is likely because you've changed the BI Server's IP address or hostname (the **fully-qualified-server-url** node in **web.xml**) from the default 127.0.0.1 to something else without also applying this change to the Proxy Trusting Filter in the **web.xml** file. See [Configuring the Proxy Trusting Filter](#) on page 14 for instructions on how to fix this.

### Configuring the Proxy Trusting Filter

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

1. Stop the Pentaho Enterprise Console.
2. Stop the BI Server.

3. Open `biserver-ee/tomcat/webapps/pentaho/WEB-INF/web.xml` and search for **TrustedIpAddrs**.



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

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

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

## JBoss Fails to Start When the Pentaho HSQLDB Sample Database Is Running

---



**Note:** This problem can also manifest as the Pentaho sample database refusing to start when the BI Server is deployed to JBoss.

The Pentaho-supplied HSQLDB sample database operates on the default HSQLDB port of 9001. JBoss has its own HSQLDB instance running on the same port; therefore, the port collision will prevent the JBoss version from starting, and cause the startup process to halt. You can change the Pentaho sample database port by editing the **start\_hypersonic** script and adding the **-port 9002** switch to the last line:

```
"$ _PENTAHO_JAVA" -cp $THE_CLASSPATH org.hsqldb.Server -port 9002 -database.0
$DIR_REL/hsqldb/sampledatabasename.0 sampledatabasename -database.1 $DIR_REL/hsqldb/
hibernate -dbname.1 hibernate -database.2 $DIR_REL/hsqldb/quartz -dbname.2
quartz
```

## JBoss Fails to Start After Manually Unpacking pentaho.war

---

If you unpack the `pentaho.war` file by hand, you must name the resultant directory **pentaho.war** as well. If you unpack it to any other directory name, including "pentaho" without the `.war` extension, JBoss will fail to deploy the WAR without any meaningful warnings.

## Web-App Path doesn't validate in Enterprise Console

---

If you are deploying to JBoss and don't unpack the `pentaho.war` into a directory, the Pentaho Enterprise Console will not be able to find the correct Web-App Path value. To fix the problem, unpack the `pentaho.war` to a directory of the same name.



# Report Designer

---

This section contains problems and solutions that pertain to Pentaho Report Designer.

## Enabling Multi-Valued report Parameters for Metadata-based Queries Created with Previous Versions of Report Designer

---

In versions 3.7 and prior, there was no support for multi-value parameters in a Metadata query. If you have a report created in an earlier version, which contains a Metadata query and an "exactly matches" condition, the report will continue to work as is; however, if you try to change the parameter from a drop-down to a multi-selection type, such as a checkbox containing *more than one value*, the report will fail.

To resolve the problem, simply open the query for editing (Query Editor) and click **OK**. This adjusts MQL query to use the EQUALS function instead of the = operator. No additional changes are necessary.

## Hive Database Disappears From Database Connection Dialog Box

---

If your Hive database type disappears from the list in the database connection dialog box in Report Designer or Metadata Editor, this probably means that your Pentaho BI Suite For Hadoop license has expired, or the license file has been moved, deleted, or become corrupt. You must install a valid license key in order to reintroduce this functionality in Report Designer.

If you do not update the licence key, all of your existing Hive-based reports will still work; however, you will not be able to change the Hive connection details or create a new Hive connection.

## Report Elements With Dynamic Heights Overlap Other Elements

---

If you have overlapping elements in your report whenever you use the **dynamic-height** style property, or if you'd like to create proper table rows so that elements of the second row get pushed down by the expanding content of the first row, then follow the directions below to create a two-row details band.

1. Select your **Details** band in the **Structure** pane, then go to the **Style** pane and change the value of **layout** to **block**.
2. Right-click the **Details** band, then select **band** from the **Add Element** context menu.
3. Move or add the elements for the first row into the band you just created.
4. Add another band, then move or copy all elements for the second row into the second band.

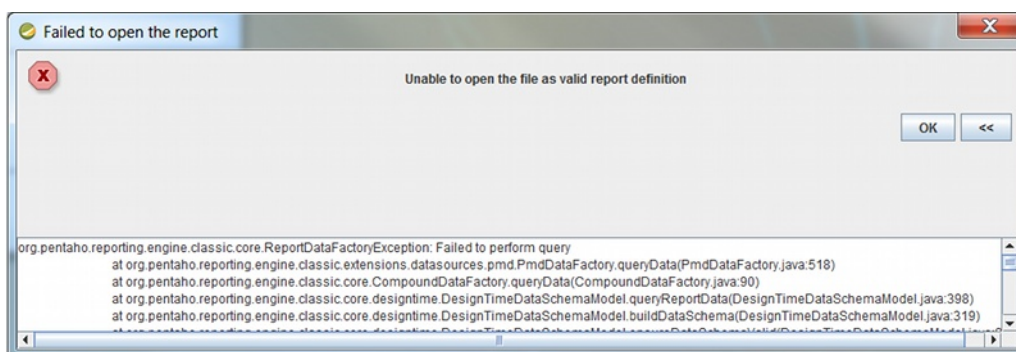
When your first row elements expand, your second row elements will be pushed down. Repeat this process as necessary for multiple rows.

## Reports Using Hive Metadata Data Sources Stop Working

---

If, when designing, editing, or running a report that contains a Hive metadata data source, you see an error like the one below, then your Pentaho BI Suite For Hadoop license has expired, or the license file has been moved, deleted, or become corrupt:





You must install a valid license key in order to reintroduce this functionality.

# Analysis

---

This section contains problems and solutions that pertain to Pentaho Analysis, including JPivot, Schema Workbench, and Pentaho Analyzer.

## Old Analysis Schemas Still Show Up in Pentaho User Console

---

If you have unwanted analysis schemas in the schema list in the Pentaho User Console -- this is the list that appears when you create a new analysis view or Analyzer report -- then you must edit the `datasources.xml` file and remove the unwanted entries as explained in [Removing Mondrian Data Sources](#) on page 18.

### Removing Mondrian Data Sources

Every time you publish a schema from Schema Workbench or Pentaho Data Integration, a new XMLA data source entry is created in `/pentaho/server/biserver-ee/pentaho-solutions/system/olap/datasources.xml`. If you modify a schema and republish it, the data source entry will be updated accordingly. You do not have to manually add anything to this file, and under most circumstances you won't have to modify or remove anything from it, either. However, there is no automatic method to expire a registered data source, so each time you publish a new schema, a new entry is permanently added to `datasources.xml`.

The Pentaho User Console uses `datasources.xml` to populate a schema drop-down selection list that appears when users create new analysis views and Analyzer reports. As you phase out old analysis schemas, you will have to manually remove their `<DataSource>` entries in `datasources.xml`.

Below is the example `datasources.xml` that ships with a default Pentaho configuration:

#### **datasources.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSources>
  <DataSource>
    <DataSourceName>Provider=Mondrian;DataSource=Pentaho</DataSourceName>
    <DataSourceDescription>Pentaho BI Platform Datasources</
DataSourceDescription>
    <URL>http://localhost:8080/pentaho/Xmla?userid=joe&password=password</
URL>
    <DataSourceInfo>Provider=mondrian</DataSourceInfo>
    <ProviderName>PentahoXMLA</ProviderName>
    <ProviderType>MDP</ProviderType>
    <AuthenticationMode>Unauthenticated</AuthenticationMode>
    <Catalogs>
      <Catalog name="SteelWheels">
        <DataSourceInfo>Provider=mondrian;DataSource=SampleData</
DataSourceInfo>
        <Definition>solution:steel-wheels/analysis/
steelwheels.mondrian.xml</Definition>
      </Catalog>
      <Catalog name="SampleData">
        <DataSourceInfo>Provider=mondrian;DataSource=SampleData</
DataSourceInfo>
        <Definition>solution:steel-wheels/analysis/
SampleData.mondrian.xml</Definition>
      </Catalog>
    </Catalogs>
  </DataSource>
</DataSources>
```

## Multi-Byte Characters Don't Appear In PDFs Exported From Analyzer

---

If you are using a multi-byte character set, such as would be used for languages like Japanese or Chinese, and find that you have missing or corrupted output when exporting Analyzer reports to PDF, you will have to specify a default TrueType font for PDF rendering that supports multi-byte characters. The default PDF font in Analyzer is Helvetica, which does not support multi-byte character sets. To change this setting, see [Setting a Default Font for PDF Exports](#) on page 19.

### Setting a Default Font for PDF Exports

Before following this procedure, stop the BI Server and Pentaho Enterprise Console.

When displaying data in Analyzer, your reports will use the default browser fonts. However, the PDF export function may not have the same fonts available to it when creating a PDF from your Analyzer report, resulting in output that doesn't look the same way in PDF format as it does in the browser. The default font for PDFs is Helvetica, but you can specify any TrueType font or collection to replace it. Follow the instructions below to specify a different font for PDF exports.



**Note:** If you have localized your schema in a language that uses a multi-byte character set (most Asian languages fit into this category), this process is required to make PDF output appear without errors.

1. Edit the **analyzer.properties** file in the `/pentaho/server/biserver-ee/pentaho-solutions/system/analyzer/` directory.
2. Uncomment the **renderer.pdf.font.path** line.  

```
renderer.pdf.font.path=C:/WINDOWS/Fonts/MSGOTHIC.TTC,1
```
3. Replace the value of this line with the TrueType font or collection that you want to use as the default.  
If you are specifying a collection, you must put a `,1` after the font name, as shown in the above example. This does not apply to individual fonts (TTF files).  

```
renderer.pdf.font.path=/usr/share/fonts/truetype/freefont/FreeSans.ttf
```
4. Save and close the file, and start the BI Server.

Your PDF exports from Analyzer should have the font you specified.

# Data Integration

---

This section contains problems and solutions that pertain to Pentaho Data Integration.

## 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 BI Server, the action sequence will not execute. This is because the BI 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 BI 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 BI Server, you must install some BI 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 4.2.0 client tool archive package from the Pentaho Knowledge Base or Enterprise Edition FTP Site.  
The package name (available in both tar.gz and zip formats) is: **pdi-ee-client-4.2.0-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 BI 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 BI Server is now configured to

## Jobs scheduled on the DI Server cannot execute a transformation on a remote Carte server

---


You may see an error like this one when trying to schedule a job to run on a remote Carte server:

```
ERROR 11-05 09:33:06,031 - !
UserRoleListDelegate.ERROR_0001_UNABLE_TO_INITIALIZE_USER_ROLE_LIST_WEBSVC!
com.sun.xml.ws.client.ClientTransportException: The server
sent HTTP status code 401: Unauthorized
```

To fix this, follow the instructions in [Executing Scheduled Jobs on a Remote Carte Server](#) on page 21

## Executing Scheduled Jobs on a Remote Carte Server

Follow the instructions below if you need to schedule a job to run on a remote Carte server. Without making these configuration changes, you will be unable to remotely execute scheduled jobs.

 **Note:** This process is also required for using the DI Server as a load balancer in a dynamic Carte cluster.

1. Stop the DI Server and remote Carte server.
2. Open the `/pentaho/server/data-integration-server/tomcat/webapps/pentaho-di/WEB-INF/web.xml` file with a text editor.
3. Find the **Proxy Trusting Filter** filter section, and add your Carte server's IP address to the **param-value** element.

```
<filter>
  <filter-name>Proxy Trusting Filter</filter-name>
  <filter-
class>org.pentaho.platform.web.http.filters.ProxyTrustingFilter</filter-
class>
  <init-param>
    <param-name>TrustedIpAddrs</param-name>
    <param-value>127.0.0.1,192.168.0.1</param-value>
    <description>Comma separated list of IP addresses of a trusted
hosts.</description>
  </init-param>
  <init-param>
    <param-name>NewSessionPerRequest</param-name>
    <param-value>true</param-value>
    <description>true to never re-use an existing IPentahoSession
in the HTTP session; needs to be true to work around code put in for
BISERVER-2639</description>
  </init-param>
</filter>
```

4. Uncomment the proxy trusting filter-mappings between the `<!-- begin trust -->` and `<!-- end trust -->` markers.

```
<!-- begin trust -->
<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/authorizationPolicy</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/roleBindingDao</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/userRoleListService</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/unifiedRepository</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/userRoleService</url-pattern>
</filter-mapping>
```

```
<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/Scheduler</url-pattern>
</filter-mapping>

<filter-mapping>
  <filter-name>Proxy Trusting Filter</filter-name>
  <url-pattern>/webservices/repositorySync</url-pattern>
</filter-mapping>
<!-- end trust -->
```

5. Save and close the file, then edit the **carte.sh** or **Carte.bat** startup script on the machine that runs your Carte server.
6. Add **-Dpentaho.repository.client.attemptTrust=true** to the **java** line at the bottom of the file.

```
java $OPT -Dpentaho.repository.client.attemptTrust=true
org.pentaho.di.www.Carte "${1+$@}"
```

7. Save and close the file.
8. Start your Carte and DI Server

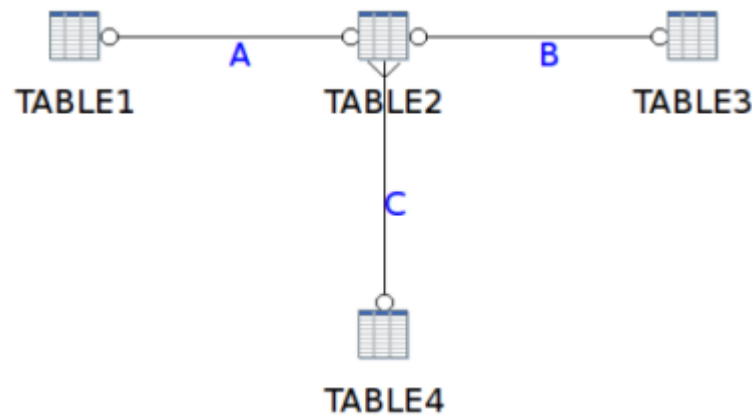
You can now schedule a job to run on a remote Carte instance.

# Metadata

This section contains problems and solutions that pertain to Pentaho Metadata, including Metadata Editor and the Agile BI and Pentaho User Console components that auto-generate metadata models.

## Managing Multiple Outer-Joins

When you have three or more tables that require outer joins, the *order* in which the tables are joined is critical. Consider the example below:



In the sample preview below, the entries, 1, 2, 3, and 4, in Table4 are taken and outer-joined with the records in the two other tables. The three other tables contain fewer records. The relationships are defined but now the order of execution critical. Relationship A is executed first, followed by B, then C.

Examine preview data				
Rows of step: The first 5000 rows from the generated query				
#	BC_TABLE1_T1_PK	BC_TABLE2_T2_PK	BC_TABLE3_T3_PK	BC_TABLE4_T4_PK
1	1	1	1	1
2	2	2		2
3				3
4				4

Below is the query that is generated:

```

SELECT DISTINCT
    TABLE1.PRIMARYKEY AS COL0
    ,TABLE2.PRIMARYKEY AS COL1
    ,TABLE3.PRIMARYKEY AS COL2
    ,TABLE4.PRIMARYKEY AS COL3

FROM TABLE4 LEFT OUTER JOIN
(
    TABLE3 FULL OUTER JOIN
    (
        TABLE1 FULL OUTER JOIN TABLE2
        ON ( TABLE1.PRIMARYKEY = TABLE2.FOREIGNKEY )
    )
    ON ( TABLE2.PRIMARYKEY = TABLE3.FOREIGNKEY )
)
ON ( TABLE2.FOREIGNKEY = TABLE4.PRIMARYKEY )

ORDER BY
    TABLE4.PRIMARYKEY

```

The nested join syntax that is generated forces the order of execution:

- Join Table1 and Table2 (shown in red)
- Join Table3 and A = B (shown in blue)
- Join Table4 with B = Result

Other orders of execution are just as valid depending on the business context to which they are applied. Another order of execution will generate a different result. To allow business model designers to ensure that user selections are executed in a specific way, a **Join Order Key** is added to the **Relationship Properties** dialog box.

The join order key is relevant only in instances in which outer joins are deployed in business models. To make the importance of the execution order apparent, this information is displayed in the graphical view of the model, as shown below.



**Note:** It is not mandatory to use uppercase letters, (A, B, C, as shown in the first image), to set the order in which tables are executed. Any alphanumeric characters (0-9, A-Z) can be used. The system will calculate the ASCII values of each character; the values are then used to determine the order of execution. In the example, A, B, C, AA, AB, Pentaho Metadata Editor will execute the table relationships in the following order: A, AA, AB, B, C.



## Using the Delay Outer Join Conditions Property

To force conditions that would ordinarily be processed in the JOIN condition to be processed in the WHERE clause, follow the directions below to create a **delay\_outer\_join\_conditions** custom property.

1. Right-click on a business model and select **Edit**.
2. Add a property by clicking the green **+** icon.
3. Select **Add a Custom Property** and set its ID to **delay\_outer\_join\_conditions** and select **boolean** for the **Type**, then click **OK**.
4. Select the newly-created **delay\_outer\_join\_conditions** property, then click the checkbox for **delay\_outer\_join\_conditions** under the **Custom** heading on the right side of the window, then click **OK**.

Instead of the conditions being rolled into the JOIN clause, they will be allowed to roll down into the WHERE clause.