



# Troubleshoot BA Server Issues

## Copyright Page

---

This document supports Pentaho Business Analytics Suite 5.2 GA and Pentaho Data Integration 5.2 GA, documentation revision October 7, 2014, copyright © 2014 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

To view the most up-to-date help content, visit <https://help.pentaho.com>.

If you do not find answers to your questions here, 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, visit <http://www.pentaho.com/training>.

### Liability Limits and Warranty Disclaimer

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

The trademarks, logos, and service marks ("Marks") displayed on this website are the property of Pentaho Corporation or third party owners of such Marks. You are not permitted to use, copy, or imitate the Mark, in whole or in part, without the prior written consent of Pentaho Corporation or such third party. Trademarks of Pentaho Corporation include, but are not limited, to "Pentaho", its products, services and the Pentaho logo.

Trademarked names may appear throughout this website. Rather than list the names and entities that own the trademarks or inserting a trademark symbol with each mention of the trademarked name, Pentaho Corporation 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.

### Third-Party Open Source Software

For a listing of open source software used by each Pentaho component, navigate to the folder that contains the Pentaho component. Within that folder, locate a folder named licenses. The licenses folder contains HTML files that list the names of open source software, their licenses, and required attributions.

### Contact Us

Global Headquarters Pentaho Corporation Citadel International, Suite 460

5950 Hazeltine National Drive Orlando, FL 32822

Phone: +1 407 812-OPEN (6736)

Fax: +1 407 517-4575

<http://www.pentaho.com>

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

## Introduction

---

This document is a compilation of all of the troubleshooting content contained in the Pentaho documentation. Each section has its own troubleshooting topics, where applicable, which contain problems and solutions that Pentaho customers and partners have encountered in the past or are anticipated to encounter in the future. Troubleshoot BA Server Issues is designed to address customers, partners, consultants, and internal Pentaho employees who are already familiar with the installation, configuration, and operation of Business Analytics, and only need to consult the documentation to solve a problem.

## General

---

This section contains problems and solutions that apply to Business Analytics in general.

- [JDBC Driver Issues](#)
- [Examining Log Files](#)

## JDBC Driver Issues

---

Before you begin troubleshooting suspected JDBC driver issues, make sure that the [correct JDBC driver](#) JARs are installed to the correct locations, then double-check to make sure that there are no conflicting driver versions. Confirm with your database or driver vendor if you suspect you have having JDBC driver compatibility issues.

The BA and DI Servers need the appropriate driver to connect to the database that stores your data. You can download drivers from your database vendor's website. A JDBC Drivers reference article, with links to the corresponding websites, can be found [here](#).

Do these steps to install your driver.

1. Stop the BA Server.
2. Copy your driver into this location: `<pentaho-install-directory>/server/biserver-ee/tomcat/lib`.
3. Start the BA Server.

Now you should be able to use the [Manage Data Sources](#) window to create Pentaho data sources.

- [Data Conversion Issues With MySQL Driver 5.0.8](#)
- [Fixing JTDS varchar\(MAX\) Limitations in MSSQL 2005](#)

## Data Conversion Issues With MySQL Driver 5.0.8

---

The MySQL JDBC driver version 5.0.8 may cause data conversion errors in the BA Server. For example, SQL statements that convert a numeric field to a string are returned as a string with version 5.0.7, but return as a byte array in 5.0.8. To solve this problem, you must replace the **mysql-connector-java-5.0.8.jar** with the **mysql-connector-java-5.0.7.jar** in your client tool or application server's **lib** directory.

## Fixing JTDS varchar(MAX) Limitations in MSSQL 2005

---

Creating a report that uses a column of type varchar(MAX) may result in a `net.sourceforge.jtds.jdbc.ClobImpl@83cf00` error when using the JTDS SQL Server driver. This is caused by inherent limitations in older versions of the JTDS driver. Additionally, the SQL Server JDBC driver version 1.2.2828 also has issues accessing a varchar(MAX) column.

The solution is to upgrade the MSSQL 2005 JDBC driver to version 1.0.809.102 or later. To do this, download and install the <http://msdn.microsoft.com/en-us/sqlserver/aa937724> file from microsoft.com, then restart your MSSQL server.



## Examining Log Files

---

If the BA Server fails to start or work properly, open the **pentaho.log** file in the `biserver-ee/bin` directory. The contents of this file provide messages that can help you track down the problem.

For JBoss, the **pentaho.log** resides in the JBoss folder structure under the `biserver-ee/logs` directory.

## Upgrade

---

This section contains problems and solutions that apply to the upgrades for both the BA Server and Pentaho Data integration.

- [Tomcat Logs Report Memory Leaks](#)
- [context.xml Changes Do Not Take Effect After Deploying a WAR](#)
- [javax.jcr.RepositoryException: no search manager configured for this workspace](#)
- [User Console Themes Render Improperly After Upgrade](#)

## 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 BA 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 Deploying a WAR

---

With a manual installation, if you deploy a WAR with a custom **context.xml**, the context.xml file may not be overwritten.

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 deploying the `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 5.2.0 to 5.2. 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.

## User Console Themes Render Improperly After Upgrade

---

If you are seeing strange rendering problems in the Pentaho User Console shortly after performing a BA 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.

## Troubleshoot the BA Server

---

This section contains problems and solutions that pertain to the BA Server and Pentaho User Console.

- [Library Conflicts](#)
- [Report Parameters That Include Accented Characters Fail to Validate](#)
- [vfs-provider.xml Duplicates](#)
- [Varying Context and Data Source Configuration Methods](#)
- [Cannot Create Jackrabbit Tables in MySQL](#)
- [Unable to Use the Database Init Scripts for PostgreSQL](#)
- [JBoss Fails to Start When the Pentaho HSQLDB Sample Database Is Running](#)
- [JBoss Fails to Start After Manually Unpacking pentaho.war](#)
- [BA Server Does Not Start When Installed on Virtual Machine](#)
- [LDAP Incorrectly Authenticates User IDs That Do Not Match Letter Case](#)
- [LDAP Roles Are Not "Admin" and "Authenticated"](#)
- [With LDAP Authentication, the PDI Repository Explorer is Empty](#)
- [Troubleshooting](#)

## Library Conflicts

---

The BA Server relies on many third-party libraries that provide everything from database connectivity to specific Java classes that add necessary features to the BA 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 BA 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 [Modifying server.xml To Work With Accented Characters](#).

- [Modifying server.xml To Work With Accented Characters](#)

## 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
```

## 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 section instruct you to edit the **context.xml** file to override Tomcat default settings. 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 section 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.

## Cannot Create Jackrabbit 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.

## 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 Jackrabbit 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 (such as `pentaho_user`) will be able to connect. The default on Debian-based systems is for local connections you 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 a local `pentaho_user` account. It's easier to just change the authentication method to something less restrictive, if your IT manager allows that approach.

## 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 BA 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/sampledatab -dbname.0 sampledatab -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.



## BA Server Does Not Start When Installed on Virtual Machine

---

If the BA Server does not start when it is installed on a Virtual Machine, and if the BA Server was deployed on the JBoss Web Application Server, increase the amount of time JBoss allows for application deployment.

Increase the deployment time to 240 seconds or longer. For information on how to do this, see [Increase the Amount of Time JBoss Allows for BA Server Deployment](#).

## LDAP Incorrectly Authenticates User IDs That Do Not Match Letter Case

---

Some LDAP implementations are case-insensitive, most notably Microsoft Active Directory. When using one of these LDAP distributions as a BA Server authentication back end, you might run into an issue where a valid user name with invalid letter cases will improperly validate. For instance, if **Bill** is the valid user ID, and someone types in **bill** at the User Console login screen, that name will authenticate, but it might have improper access to parts of the BA Server.

The fix for this is documented: [LDAP Authenticates User IDs That Do Not Match Case](#).

- [LDAP Authenticates User IDs That Do Not Match Case](#)

## LDAP Authenticates User IDs That Do Not Match Case

---

Some LDAP implementations are case-insensitive for user names, most notably Microsoft Active Directory. You might run into an issue where a user name typed into the login screen does not exactly match the letter case of that user's ID in the directory, but the server will authenticate it anyway and may give the user improper access to parts of the BA Server. For example, if **Bill** is the valid user ID, and someone types in **bill** at the User Console login screen, the incorrectly typed one will authenticate, but it may have improper access to parts of the BA Server.

Follow these instructions to force case-sensitivity and fix this potential security risk.

1. Stop the BA Server.
2. Edit the `/pentaho/server/biserver-ee/pentaho-solutions/system/applicationContext-spring-security-ldap.xml` file.
3. Find the **daoAuthenticationProvider** bean, and below the last `</constructor-arg>` therein, and add the **<property>** definition shown in the example:

```
<property name="userDetailsContextMapper">
    <ref local="ldapContextMapper" />
</property>
```

4. After the `</bean>` tag for **daoAuthenticationProvider**, add the following bean definition, changing the **ldapUsernameAttribute** from **samAccountName** to the value that matches your environment:

```
<bean id="ldapContextMapper" class="org.pentaho.platform.engine.security.
UseridAttributeLdapContextMapper">
    <property name="ldapUsernameAttribute" value="samAccountName" />
</bean>
```

5. Start the BA Server.

The BA Server will now force case sensitivity in LDAP user names.

## LDAP Roles Are Not "Admin" and "Authenticated"

---

You must not use **Admin** and **Authenticated** roles in your LDAP. Instead you must configure your system to use **pentahoAdmins** and **pentahoUsers** or other easily identifiable role names. Edit `/pentaho-solutions/system/applicationContext-spring-security.xml`. At the bottom of this file, you will find a number of lines that look like: `A/docs/. *Z=Anonymous,Authenticated`.

These are entries for URL Security. They are regular expressions to match a path on the browser's URL that require the user to be a member of the defined role to gain access. In the example above, both **Anonymous** and **Authenticated** get access. In the example above, use **pentahoUsers** in the place of **Authenticated**. by entering `A/docs/. *Z=Anonymous,pentahoUsers`. For all entries that show **Authenticated**, replace it with **pentahoUsers** or your chosen name. Replace **Admin** with **pentahoAdmins** or your chosen name. For the change from **Authenticated** to **pentahoUsers** replace all occurrences. For **Admin** to **pentahoAdmins** you need to be a little more careful because there are some entries that look like this: `A/admin.*Z=pentahoAdmins`.

Edit the `/pentaho-solutions/system/repository.spring.xml` file and change:

```
<bean id="singleTenantAuthenticatedAuthorityName" class="java.lang.String">
  <constructor-arg value="Authenticated" />
</bean>
```

to:

```
<bean id="singleTenantAuthenticatedAuthorityName" class="java.lang.String">
  <constructor-arg value="pentahoUsers" />
</bean>
```

and:

```
<bean id="singleTenantAdminAuthorityName" class="java.lang.String">
  <constructor-arg value="Admin" />
</bean>
```

to:

```
<bean id="singleTenantAdminAuthorityName" class="java.lang.String">  
  <constructor-arg value="pentahoAdmins" />  
</bean>
```

## With LDAP Authentication, the PDI Repository Explorer is Empty

---

If you log into a solution repository from Spoon before you switch the authentication to LDAP, then the repository IDs and security structures will be broken. You won't see an error message, but the solution repository explorer will be empty and you won't be able to create new folders or save PDI content to it. To fix the problem, you will have to delete the security settings established with the previously used authentication method, which will force the DI Server to regenerate them for LDAP.

**CAUTION:**

Following this procedure will destroy any previously defined DI repository users, roles, and access controls. You should back up the files that you delete in these instructions.

1. Stop the DI Server
2. Delete the security and default directories from the following directory: `/pentaho-solutions/system/jackrabbit/repository/workspaces/`
3. Start the DI Server

You should now have a proper LDAP-based DI Repository that can store content and create new directories.

## Troubleshooting

---

This section contains known problems and solutions relating to the procedures covered in previous sections.

- [Windows Domains Won't Authenticate When Using the JTDS Driver](#)
- [Geo Maps partially rendered](#)

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

---

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

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



## Troubleshooting Geo Maps

---

### Geo Maps Partially Rendering

If your Geo Map visualizations in Analyzer are not correctly displaying, it's possible that the BA Server is not giving them enough time to fully render.

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

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

### Extend the Cache Value for Geo Maps

If you run a Geo Map report and you do not see all of the points on the map that you expected, one thing you could try is extending the cache value for Geo Maps.

1. Locate the `...biserver-ee/tomcat/webapps/pentaho/WEB-INF/classes` directory and open `ehcache.xml` with a text editor.
2. Look for the `cache` element and change the value for `maxElementsInMemory` to a larger number.

```
<cache
  name="pentaho-geo-{CUSTOM_ROLE_TYPE}"
  maxElementsInMemory="1000"
  eternal="false"
  overflowToDisk="true"
  timeToIdleSeconds="0"
  timeToLiveSeconds="0"
  diskPersistent="false"
  diskExpiryThreadIntervalSeconds="120"
/>
```

1. Save and close the `ehcache.xml` file.
2. Restart the BA Server.

## Report Designer and Reporting Engine

---

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

- [Report Elements With Dynamic Heights Overlap Other Elements](#)
- [Columns Unexpectedly Merge When Exporting to Excel](#)

## 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.

## Columns Unexpectedly Merge When Exporting to Excel

---

If you export content from Report Designer to Excel, and end up with unexpectedly merged columns in the output, there is probably a horizontal alignment problem with your column header or footer labels. If a label spans two columns -- even by a tiny amount -- then the Reporting engine will force the two columns to merge in the output.

Check your horizontal elements for column overlap. If you need more information on this topic, refer to the section [Aligning Elements](#).

## Analysis

---

This section contains problems and solutions that pertain to Pentaho Analysis.

- [Multi-Byte Characters Don't Appear In PDFs Exported From Analyzer](#)

## 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 [Set a Default Font for PDF Exports](#).

## Set a Default Font for PDF Exports

---

Before following this procedure, stop the BA Server and the User 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 BA 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.

- [Troubleshooting a Corrupted DI Server Repository](#)
- [Action Sequences That Call PDI Content Won't Run](#)
- [Jobs Scheduled on DI Server Cannot Execute Transformation on Remote Carte Server](#)
- [Kitchen can't read KJBs from a Zip export](#)
- [Using ODBC](#)
- [Troubleshooting MonetDB for Instaview](#)



## Troubleshooting a Corrupted DI Server Repository

---

If the DI Repository becomes corrupt, it will be unresponsive, content may be missing or inaccessible, and an error message similar to this will appear in the `/data-integration-server/tomcat/logs/catalina.out` log file:

```
ERROR [ConnectionRecoveryManager] could not execute statement, reason: File
corrupted while reading record: "page[48970] data leaf table:8 entries:1
parent:49157 keys:[118547] offsets:[737]". Possible solution: use the recovery
tool [90030-131], state/code: 90030/90030
```

If this happens, shut down the DI Server and restore your solution repository from a recent backup.

If you do not have a viable backup, you may be able to minimize data loss by identifying the exact file that is corrupt. To do this, enable debug logging by adding the following XML snippet above the `<root>` element in the `/WEB-INF/classes/log4j.xml` inside your deployed `pentaho.war`:

```
<category name="org.pentaho.platform">
  <priority value="DEBUG"/>
</category>
```

Restart the DI Server and retry the action that caused the original error. If it occurs again, shut down the DI Server and open the `catalina.out` log file in Tomcat. Look for the last line that appears before the error; it usually contains the name of the file that has been damaged. When you are finished investigating the data corruption, remove the extra logging capabilities so that your DI Server log files don't become large and unmanageable.

```
reading file with id 'xyz' and path '/public/a.txt'
```

You can also try to recover your PDI data from the damaged database by using a recovery program, as explained in [Using the H2 Database Recovery Tool](#).

Note: If the database has become corrupt, the damaged rows will not be exported with the recovery tool, and whatever information they contained will be lost.

- [Using the H2 Database Recovery Tool](#)

## Using the H2 Database Recovery Tool

---

The DI Server includes a third-party H2 database recovery tool that enables you to extract raw data from your DI Repository. This is primarily useful in situations where the repository has become corrupt, and you don't have any relevant backups to restore from.

Note: If the database has become corrupt, the corrupted rows will not be exported. Any information contained in corrupted rows is unrecoverable through this method.

The recovery tool is a JAR that must be run via the Java command. The output is a SQL dump that you can then attempt to import after you've re-initialized your DI Server database.

You can read more about the recovery tool on the H2 Web site: [http://www.h2database.com/html/advanced.html#using\\_recover\\_tool](http://www.h2database.com/html/advanced.html#using_recover_tool).

Follow the directions below to use the recovery tool.

1. Open a terminal on (or establish an SSH connection to) your DI Server.
2. Navigate to the `/pentaho-solutions/system/jackrabbit/repository/version/` directory.

```
cd data-integration-server/pentaho-solutions/system/jackrabbit/repository/
version/
```

3. Run the **h2-1.2.131.jar** H2 database JAR with the recovery tool option.

```
java -cp h2-1.2.131.jar org.h2.tools.Recover
```

4. Create a directory to move your old database files to.

```
mkdir old
```

5. Move the old database files to the directory you just created.

```
mv db.h2.db db.trace.db old
```

6. Re-initialize the repository with the **RunScript** option, using the salvaged SQL dump as the source.

```
java -cp h2-1.2.131.jar org.h2.tools.RunScript -url jdbc:h2:./db -user sa -
script db.h2.sql
```

7. The backup directory you created earlier (**old** in the above example) can be removed after you're sure that you don't need the corrupted database files anymore. However, it's a good idea to keep this around just in case you need it later.

You've successfully extracted all of the usable data from your corrupt solution repository, removed the damaged database files, and re-initialized the repository with the salvaged data. Start the DI Server and check for further errors. If repository errors persist, contact Pentaho support and request developer assistance.

## Action Sequences That Call PDI Content Won't Run

---

If you've established a solution 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.

## Jobs Scheduled on DI Server Cannot Execute Transformation on 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](#).

## Execute 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. Copy the **repositories.xml** file from the `.kettle` directory on your workstation to the same location on your Carte slave. Without this file, the Carte slave will be unable to connect to the DI Repository to retrieve PDI content.
3. Open the `/pentaho/server/data-integration-server/tomcat/webapps/pentaho-di/WEB-INF/web.xml` file with a text editor.
4. 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>TrustedIpAddr</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>
```

5. 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 -->

```

6. Save and close the file, then edit the **carte.sh** or **Carte.bat** startup script on the machine that runs your Carte server.
7. 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+$@}"

```

8. Save and close the file.

## 9. Start your Carte and DI Server

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

## Kitchen can't read KJBs from a Zip export

---

Note: This also applies to Pan and KTR files in Zip archives.

If you are trying to read a KJB file from a Zip export but are getting errors, you may have a syntax error in your Kitchen command. Zip files must be prefaced by a ! (exclamation mark) character. On Linux and other Unix-like operating systems, you must escape the exclamation mark with a backslash: \!

### Windows:

```
Kitchen.bat /file:"zip:file:///C:/Pentaho/PDI_Examples/Sandbox/linked_executable_
job_and_transform.zip!Hourly_Stats_Job_Unix.kjb"
```

### Linux:

```
./kitchen.sh -file:"zip:file:///home/user/pentaho/pdi-ee/my_package/linked_
executable_job_and_transform.zip\!Hourly_Stats_Job_Unix.kjb"
```



## Using ODBC

---

Although ODBC can be used to connect to a JDBC compliant database, Pentaho does not recommend using it and it is not supported. For details, this article explains "Why you should avoid ODBC."

<http://wiki.pentaho.com/pages/viewpage.action?pageId=14850644>.

## Troubleshooting MonetDB for Instaview

---

Instaview loads ETL models, dimensional schema, and data in a MonetDB-based database. When a new Instaview project is created or an existing one is opened, the MonetDB server is started. Most issues that can occur relate to the way that Pentaho controls MonetDB.

### Connection Errors

Connection errors sometimes occur if there are MonetDB port conflicts. This sometimes occurs if you have installed Instaview in the past using one method, then reinstalled it using a different method. Check the share.xml file and modify the MonetDB port number in the Agile BI section so that it is not 50000, but another number such as 50006. Exit from Instaview and Spoon and try to start them both again.

### General Errors

The Monetdb server is started when a new Instaview project is created or an existing one is opened. The server is shutdown when the project is closed. Pentaho provides scripts to start and stop the server. These scripts are located in the data-integration/plugins/spoon/agile-bi/platform/pentaho-solutions/system/instaview/scripts directory.

For testing or debugging, these scripts can be run with no arguments passed in. However, defaults are used for parameters that are not passed in. Each operating system has different defaults, the configuration definitions in the previous examples drive the values passed in to these scripts when executed from within PDI/Instaview. When running the scripts, take care to pass in the correct values, otherwise the database may not start up properly. Open the script in a text editor before calling it to determine what parameters you might need. Here are some example paramaters. None of the parameters are required, but are key if you stray from the default configurations.

### For Windows

```
startMonetDb.bat DB_NAME MONETDB_INSTALL_DIR
```

Parameter	Default Value
DB_NAME	pentaho-instaview
MONETDB_INSTALL_PATH	MONETDB_INSTALL_DIR environment variable

### For Mac or Linux

```
./startMonetDb.sh DB_NAME DB_FARM_LOCATION MONETDB_CMD MONETDBD_CMD MONETDB_PORT
PASSPHRASE
```

Parameter	Mac Default Value	Linux Default Value	Comment
DB_NAME	pentaho-instaview		
DB_FARM_LOCATION	\$HOME/.kettle/instaview/dbfarm	\$HOME/.kettle/instaview/dbfarm	
MONETDB_CMD	/usr/local/monetdb/bin/monetdb	/usr/bin/monetdb	
MONETDBD_CMD	/usr/local/monetdb/bin/monetdbd	/usr/bin/monetdbd	
MONETDB_PORT	50000	50000	To avoid port conflicts, this port number is sometimes changed to 50006.
PASSPHRASE	monetdb	monetdb	Required for remote control of Monetdb

Also, check the AgileBI shared connection in PDI that Instaview creates and uses to talk to Monetdb. You can open this connection in PDI and test or explore it. This connection is used by the templates. If you open the `.ktr` file used by Instaview you can run it that way as long as Monetdb is running.

## 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.

- [Publish](#)
- [Manage Multiple Outer-Joins](#)

## Publish

---

If publishing to the BA Server fails, here are a few things you may want to verify.

- Ensure that the BA Server is running.
- Verify that your security access is defined as the Pentaho administrator role. The default is **admin**.
- Check that you have not used trailing or leading slashes in the Web Publish URL or Domain Name

## Manage 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:

File:/48\_outer\_joins.png

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.

File:/49\_execution\_of\_joins.png

Below is the query that is generated:

File:/50\_query\_outer\_joins.png

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.

File:/51\_relationship\_properties\_dialog\_box.png

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.

- [Use the Delay Outer Join Conditions Property](#)

## Use 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.

## Pentaho Metadata-Based Reports Generate Inconsistent Results

---

### Pentaho Metadata-Based Reports Produce Slow Responses and Generate Inconsistent Results

If a report contains inconsistent or incorrect results, and the SQL command contains several joins, make sure the join order has been specified for that report. Beginning in version 5.0, inner joins are resolved before outer joins. If you have several tables that require outer joins, the order in which the tables are joined is critical because different join orders produce different results. To make the join order consistent, set the **Join Order Key** field in the **Relationship Properties** window. For more information, see [Managing Outer Joins](#).

### Pentaho Metadata-Based Reports Generated in Version 4.8 Produce Different Results in Version 5.x

In version 5.0 of the software, inner joins are resolved before outer joins by default. If you are migrating a report from version 4.8 to 5.x, you can specify which method to use by setting the `legacy_join_order` parameter in the report's `.xmi` file. Set `legacy_join_order` to `true` to use the 4.8 method. Set `legacy_join_order` to `false` to use the 5.0 method.

In addition to setting the `legacy_join_order` parameter, set the **Join Order Key** field in the **Relationship Properties** window. For more information, see [Managing Outer Joins](#).