



## Upgrading to Pentaho Data Integration 4.0



This document is copyright © 2010 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.....	4
What's new in 4.0?.....	5
Upgrade Best Practices.....	6
Upgrade Checklist.....	7
Creating Backups.....	9
Backing Up Content Files.....	9
Backing Up a Database Repository.....	9
Backing Up the .kettle Directory.....	9
Installing PDI 4.0.....	10
Obtaining the Archive Packages.....	10
Server Installation Procedure.....	10
Server Archive Package Deployment.....	10
Post-Install Configuration.....	11
Workstation Installation Procedures.....	13
Workstation Archive Package Deployment.....	13
Upgrading or Importing Old PDI Content.....	15
The Advantages of an Enterprise Repository.....	15
When NOT to Upgrade.....	15
Moving Old Content Files to a New Location.....	15
Upgrading a Database Repository to the New DB Schema.....	16
Switching to an Enterprise Repository.....	16
Creating an Enterprise Repository.....	16
Converting a Database Repository to an Enterprise Repository.....	17
Importing Old Content Files Into an Enterprise Repository.....	17
Note on Schedules.....	18
Testing and Cleanup.....	19
FAQ: New Features in 4.0.....	20

# Introduction

---

This guide shows current Pentaho Data Integration Enterprise Edition customers how to upgrade to PDI 4.0. Previous PDI releases were relatively easy to upgrade, but there are major structural improvements in PDI 4.0 that require a more detailed upgrade path.



**Note:** The oldest version of PDI (or Kettle) that you can upgrade from is version **2.5**.



**Note: This guide is only for upgrading to PDI 4.0 Enterprise Edition.** You can upgrade from previous releases of either Enterprise Edition or Community Edition, up to and including Kettle 4.0 Community Edition. However, you must have an Enterprise Edition license key in order to complete the upgrade process from a Community Edition release.

## What's new in 4.0?

---

Pentaho Data Integration 4.0 represents the single largest update in the product's history. You can read the complete list of new features and functions at: <http://wiki.pentaho.com/display/EAI/What%27s+new+in+PDI+4.0>. Here are the highlights from this release:

- **New pluggable "enterprise repository"** for storing, managing, and sharing content among multiple BI developers.
  - **Impact:** In order to take full advantage of the new enterprise repository features, you must import your old content files or convert your old database-based repository. If you have a large volume of PDI content, this may take some time.
- **User interface changes:** mouse-over context help; improved menu organization; more informative and useful welcome screen; easier hop creation for one-button mice; improved error handling configuration; new perspectives for Agile BI visualisations, modelling, and scheduling.
  - **Impact:** Users may need training to understand how the PDI interface has changed.
- **Job changes:** drill-down into running job entries; visual indicators of running and completed job entries with success and failure mini-icons; mouse-over completion mini-icons show details of execution results; log capturing of completed job entries.
  - **Impact:** Users may need training to understand how PDI job handling has changed.
- **Transformation changes:** drill-down into running transformation entries and mappings; row input/output sniff testing -- see what rows are passing; remote input/output sniff testing on a Carte server.
  - **Impact:** Users may need training to understand how PDI transformation handling has changed.
- **A new logging architecture:** reduced memory consumption; incremental log updates; global log buffer size limit for long-running jobs/transformations; interval logging; automatic cleanup of old log records; log record timeouts; log record lineage; log record color coding in Spoon (blue and red for error lines); step logging; job entry logging; execution lineage logging; renaming of individual columns; global configuration options for all log tables.
  - **Impact:** Users and administrators may need training to learn how to take full advantage of PDI's new logging capabilities to measure performance and troubleshoot errors.
- **A new plugin architecture** that enables developers to more easily extend PDI's functionality.
  - **Impact:** No negative impact to existing users or administrators. Developers interested in extending PDI should examine the Javadoc and developer notes on the PDI wiki.
- **New steps:** SAP Input; Data Grid; OLAP Input; Salesforce Delete, Insert, Update, Upsert; Add fields changing sequence; User Defined Java Class; Send information using Syslog; Java Filter; Memory Group By; Farrage streaming bulk loader; Teradata Fastload Bulk loader; experimental steps added: Get table names, Email messages input.
  - **Impact:** Users may need training to learn how to take full advantage of PDI's new step types.
- **New jobs:** Send information using Syslog; Check DB connections.
  - **Impact:** Users may need training to learn how to take full advantage of PDI's new job types.

# Upgrade Best Practices

---

All production software upgrades, including Pentaho Data Integration, should be performed during off-peak hours and with enough time to restore from a backup before off-peak ends if something should go wrong. Ideally you would perform the upgrade on a test machine that mirrors the production environment, take notes along the way, and perform the same procedure on the production server when you know how long the entire process will take and are sure that there will be no unexpected problems.

This guide contains instructions for performing the safest possible upgrade. There may be quicker ways, but the software's architects recommend the path outlined in this guide for the safest and most predictable transition to PDI 4.0.

**Always back up your production data, and test the backup before proceeding with an upgrade.**

PDI 4.0 abstracts the Data Integration Server (DI Server) from the Data Integration client tools (Spoon, Pan, and Kitchen). To take full advantage of the performance gains that this abstraction offers, you should dedicate a separate machine to host the DI Server and Pentaho Enterprise Console. This may require a new hardware purchase; alternatively, you could install the DI Server and Enterprise Console to a high-powered workstation or other development machine that has spare resources or is seldom used.

# Upgrade Checklist

The Upgrade Checklist is a concise list of instructions intended to show a high-level overview of the upgrade process. It also serves as a method of verifying that each task is performed in the correct order. You may find it useful to print the checklist out and physically mark each step in the Done column as you complete it. **The checklist is not the complete instruction set**; consult the verbose instructions throughout this guide for more details on each step.

Step	Procedure	Done
Step 1	Download the upgrade materials from the Enterprise Edition FTP site: <b>biserver-manual-ee-3.5.2-GA.zip</b> , <b>pec-ee-3.5.2-GA.zip</b> , any Pentaho User Console plugins you have licenses for, and the BI Suite installer for your platform, then unpack them to an accessible and convenient location.	
Step 2	Stop your application server and the Pentaho Enterprise Console server.	
Step 3	Back up your <code>/pentaho/server/enterprise-console/</code> directory, Pentaho Web application ( <code>pentaho.war</code> or its uncompressed directory in your application server's deploy directory), and the <code>/pentaho/server/biserver-ee/pentaho-solutions/</code> directory.	
Step 4	If you are using an Oracle solution repository, copy the <b>quartz-oracle-1.5.2.jar</b> file to the appropriate directory to be included in the WAR or EAR build.	
Step 5	If you have made any changes to <b>web.xml</b> , merge your customizations with the <code>/biserver-manual-ee/build-resources/pentaho-webapp/WEB-INF/web.xml</code> file, then copy that merged file to the <code>/biserver-manual-ee/build-resources/WEB-INF/</code> directory.	
Step 6	Copy your old <b>context.xml</b> file (for both JBoss and Tomcat deployments), and your <b>pentahoHibernate-ds.xml</b> and <b>quartz-ds.xml</b> files (for JBoss deployments only) to the <code>/biserver-manual-ee/build-resources/custom-pentaho-webapp/META-INF/</code> directory.	
Step 7	Build new Pentaho WAR or EAR packages with the BI Server 3.5.2 manual deployment materials.	
Step 8	Copy the new EAR or WARs to the <code>/server/webapps/</code> directory in Tomcat, or the <code>/server/default/deploy/</code> directory in JBoss.	
Step 9	Copy all of your solutions directories from your old <code>/pentaho/server/biserver-ee/pentaho-solutions/</code> directory to the new one shipped with the manual deployment package. Manually update your Spring Security files by examining your old Acegi Security configuration files and migrating settings by hand. When you're done, move or delete your old <code>pentaho-solutions</code> directory and copy the new one to the same location.	
Step 10	Upgrade the Pentaho Enterprise Console by unpacking the Pentaho Enterprise Console to a temporary location, then copying the following files from the old Pentaho Enterprise Console to the new one: <code>console.xml</code> , <code>console.properties</code> , <code>login.properties</code> , <code>login.conf</code> , <code>log4j.xml</code> . When you are finished, delete the old <code>enterprise-console</code> directory and copy the new one to the same location.	

Step	Procedure	Done
Step 11	Run the <b>migration.sql</b> file for your database type to update the hibernate database.	
Step 12	On Linux or Solaris servers, create a new system account (with a home directory) named <b>pentaho</b> . This is the user account that you should control both the Pentaho Enterprise Console and the BI Server with from now on.	
Step 13	Establish a <b>PENTAHO_JAVA_HOME</b> system variable and point it to your JRE 1.5.0 directory.	
Step 14	Install any Pentaho User Console plugins for which you have support entitlements.	
Step 15	Perform the documented steps for upgrading single-sign on if you previously used this as your authentication method.	
Step 16	If you have any custom login dialogues, update them according to the documented instructions.	
Step 17	On client machines that run Report Designer, upgrade them to Report Designer 3.5.2, then open all reports that contain charts and verify that the angle of the label-rotation properties is acceptable.	
Step 18	Modify any system startup or init scripts that call the old Pentaho Enterprise Console start and stop script file names. The new scripts are: start-pec.sh (or .bat) and stop-pec.sh (or .bat).	
Step 19	Start the JBoss or Tomcat server and the Pentaho Enterprise Console server.	
Step 20	Reinstall your Enterprise Edition license keys in the Pentaho Enterprise Console, or with the command line license tool.	
Step 21	<b>Check that all of your content is accessible.</b> Log into the Pentaho User Console and verify that your product licenses work. Run at least two existing reports for each data source to ensure that there are no bugs or problems with them. Verify at least two existing analysis views for each data source. Also verify that any previously created dashboards are functioning properly. Lastly, check all of your schedules to ensure that they are still properly configured and operational.	
Step 22	Back up your new configuration in Pentaho Enterprise Console.	
Step 23	Use the BI Suite installer or individual archive packages to upgrade the Pentaho client tools on user workstations.	
Step 24	Delete the /biserver-manual-ee/ directory.	

# Creating Backups

---

Backing up your production content isn't just a good idea -- it's required step in upgrading to an enterprise repository. Additionally, it's a good idea to back up your Kettle settings in case something goes wrong with the client tool upgrade.

## Backing Up Content Files

---

If you do not use a database repository for storing PDI content, then you are saving individual KJB and KTR files on a local or network drive. Hopefully you have created a sensible directory structure and naming convention for them. If not, this may be a good time to organize them properly.

Once you have all of your content in one directory, simply create a Zip or tar archive of it and copy the archive to a safe location outside of your local machine, such as a network drive or removable media.

**This should be part of your normal production backup routine outside of this upgrade process.**

## Backing Up a Database Repository

---

Backing up your PDI database repository is as simple as using the **export** functionality in Spoon's then copy the resulting file to a safe location outside of the machine you are upgrading.

**This should be part of your normal production backup routine outside of this upgrade process.**

## Backing Up the .kettle Directory

---

The **.kettle** directory stores all of your Spoon configuration settings and preferences. It is located in `~/.kettle` on Linux, Solaris, and OS X; and `C:\Documents and Settings\username\.kettle` on Windows, where *username* refers to the user account that Spoon is installed to.

Create a Zip or tar archive of this directory and copy the archive to a safe location before upgrading.

# Installing PDI 4.0

---

The instructions in this section explain how to install PDI 4.0 using archive packages.

 **Note:** Before you commence with installation, you should rename your old data-integration directory to **data-integration-old** so that it is not accidentally overwritten by the installation process.

## Obtaining the Archive Packages

---

Consult the Welcome Kit email that was sent to you after completing the sales process. This email contains user credentials for the Enterprise Edition FTP site, where you can download individual archive packages for the DI Server, Data Integration client tools, and Pentaho Enterprise Console. Here are the direct links:

- **DI Server for Windows:** `ftp://subscription.pentaho.com/enterprisesw/Pentaho_Data_Integration/4.0.0-GA/server/pdi-ee-data-integration-server-4.0.0-GA.zip`
- **DI Server for Linux/Solaris/OS X:** `ftp://subscription.pentaho.com/enterprisesw/Pentaho_Data_Integration/4.0.0-GA/server/pdi-ee-data-integration-server-4.0.0-GA.tar.gz`
- **Enterprise Console for Windows:** `ftp://subscription.pentaho.com/enterprisesw/Pentaho_Data_Integration/4.0.0-GA/server/pec-3.6.0-GA.zip`
- **Enterprise Console for Linux/Solaris/OS X:** `ftp://subscription.pentaho.com/enterprisesw/Pentaho_Data_Integration/4.0.0-GA/server/pec-3.6.0-GA.tar.gz`
- **Data Integration client tool Windows package:** `ftp://subscription.pentaho.com/enterprisesw/Pentaho_Data_Integration/4.0.0-GA/client/pdi-ee-client-4.0.0-GA.zip`
- **Data Integration client tool Linux/Solaris/OS X package:** `ftp://subscription.pentaho.com/enterprisesw/Pentaho_Data_Integration/4.0.0-GA/client/pdi-ee-client-4.0.0-GA.tar.gz`

## Server Installation Procedure

---

To install and configure the Data Integration Server and Pentaho Enterprise Console from archive packages, follow the below procedures in the order they are presented. This process should take less than 20 minutes.

### Server Archive Package Deployment

Follow the below instructions to install the Data Integration Server and Pentaho Enterprise Console on a dedicated server. If you intend to deploy the client tools and servers on one machine, you can combine this with [Workstation Installation Procedures](#) on page 13, or use the PDI graphical installer.

 **Note:** The example commands in this and other sections are specific to Linux. You will have to adjust or ignore them on other operating systems.

1. Create a **/pentaho/server/** directory in an appropriate place in your hierarchy.

This directory should be accessible to the system users who will be controlling services. Typically only root or the users in the wheel or administrator group will need to do this.

 **Note:** If you are using the graphical installer, it will create this directory structure for you, so you can skip this step.

```
mkdir -p /usr/local/pentaho/server/
```

2. Unpack the **pdi-ee-data-integration-server-4.0.0-GA** archive to `/pentaho/server/`.

```
tar zxvf pdi-ee-data-integration-server-4.0.0-GA.tar.gz -C /usr/local/pentaho/server/
```

3. Unpack the **pec-3.6.0-GA** archive to `/pentaho/server/`.

```
tar zxvf pec-3.6.0-GA.tar.gz -C /usr/local/pentaho/server/
```

4. Switch to the `/pentaho/server/data-integration-server/` directory and run the **start-pentaho** script to start the DI Server.

```
cd /usr/local/pentaho/server/data-integration-server/ && ./start-pentaho.sh
```

5. Switch to the `/pentaho/server/enterprise-console-server/` directory and run the **start-pec** script to start the Pentaho Enterprise Console.

```
cd /usr/local/pentaho/server/data-integration-server/ && ./start-pentaho.sh
```

The DI Server and Enterprise Console are now installed, and should be operational. The DI Server will not be accessible from workstations until a license key is installed.

## Post-Install Configuration

After you've installed PDI software to your server and workstations, you must perform some extra tasks to register license keys, connect your workstations to the server, and configure the server to start at boot time. Follow the sections below that apply to your situation.

### Installing or Updating an Enterprise Edition Key

You must install Pentaho Enterprise Edition keys associated with products for which you have purchased support entitlements. The keys you install determine the layout and capabilities of the Pentaho Enterprise Console, and the functionality of the BI Platform. Follow the instructions below to install an Enterprise Edition key through the Pentaho Enterprise Console for the first time, or to update an expired or expiring key. If you would prefer to use a command line tool instead, see [Appendix: Working From the Command Line Interface](#) on page 11.

 **Note:** If your Pentaho Enterprise Console server is running on a different machine than your BI Platform server, you must use the command line tool to install and update license files; you cannot use the Pentaho Enterprise Console for this task.

1. If you have not done so already, log into the Pentaho Enterprise Console by opening a Web browser and navigating to `http://diserver-hostname:8088`, changing **diserver-hostname** to the hostname or IP address of your server.
2. Click the **+** (plus) button in the upper right corner of the Subscriptions section. An **Install License** dialog box appears.
3. Click **Browse**, then navigate to the location you saved your LIC files to, then click **Open**. LIC files for each of your supported Pentaho products were emailed to you along with your Pentaho Welcome Kit. If you did not receive this email, or if you have lost these files, contact your Pentaho support representative. If you do not yet have a support representative, contact the Pentaho salesperson you were working with.

 **Note:** Do not open your LIC files with a text editor; they are binary files, and will become corrupt if they are saved as ASCII.

4. Click **OK**.

The Setup page changes according to the LIC file you installed.

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

If your license files fail to install properly on Linux, you must establish a Java system preferences directory for the Pentaho Web application. See the [Installing an Enterprise Edition Key on Linux \(CLI\)](#) on page 12 for instructions.

### Appendix: Working From the Command Line Interface

Though the Pentaho Enterprise Console is the quickest, easiest, and most comprehensive way to manage PDI and/or the BI Server, some Pentaho customers may be in environments where it is difficult

or impossible to deploy or use the console. This appendix lists alternative instructions for command line interface (CLI) configuration.

#### *Installing an Enterprise Edition Key on Windows (CLI)*

To install a Pentaho Enterprise Edition Key from the command line interface, follow the below instructions.

 **Note:** Do not open your LIC files with a text editor; they are binary files, and will become corrupt if they are saved as ASCII.

1. Navigate to the `\pentaho\server\enterprise-console\` directory.
2. Run the **install\_license.bat** script with the **install** switch and the location and name of your license file.

```
install_license.bat install biserver-ee.lic
```

Upon completing this task, you should see a message that says, "The license has been successfully processed. Thank you."

#### *Installing an Enterprise Edition Key on Linux (CLI)*

To install a Pentaho Enterprise Edition Key from the command line interface, follow the below instructions.

 **Note:** Do not open your LIC files with a text editor; they are binary files, and will become corrupt if they are saved as ASCII.

1. Navigate to the `/pentaho/server/enterprise-console/` directory.
2. Run the **install\_license.sh** script with the sole parameter of the location and name of your license file.

```
install_license.sh install /home/pgibbons/downloads/biserver-ee.lic
```

Upon completing this task, you should see a message that says, "The license has been successfully processed. Thank you."

### **Starting the DI Server At Boot Time On Linux**

This procedure assumes that you will be running your DI Server and Pentaho Enterprise Console server under the **pentaho** local user account. If you are using a different account to start these services, substitute it in the script below.

You can start and stop the BI Server at any time by running the **start-pentaho.sh** and **stop-pentaho.sh** scripts, or you can start and stop the entire stack -- MySQL, Tomcat, and the Pentaho Enterprise Console -- with the provided `ctlscrip.sh` script in the **/pentaho/** directory. To start the Tomcat server automatically at boot time, and stop automatically during shutdown, follow the below procedure.

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

```
#!/bin/sh -e
### BEGIN INIT INFO
# Provides: pdi
# Required-Start: networking
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Description: Pentaho DI Server
### END INIT INFO

case "$1" in
"start")
su - pentaho -c "/home/pentaho/pentaho/server/data-integration-server/start-pentaho.sh"
su - pentaho -c "cd /home/pentaho/pentaho/server/enterprise-console && ./start-pec.sh"
;;
```

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

3. Save the file and close the text editor.
4. Make the init script executable.

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

5. Add the pentaho init script to the standard runlevels so that it will run when the system starts, and stop when the system is shut down or rebooted, by using the update-rc.d command.  
This command may not exist on your computer if it is not Debian-based. If that is the case, consult your distribution documentation or contact your distribution's support department to determine how to add init scripts to the default runlevels.

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

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

### Starting the BI Server At Boot Time On Windows

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

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

1. Open a Command Prompt window.
2. Use the SC.exe command to create a new Windows service for the BI Server: **sc create PentahoDIserver binpath= C:\pentaho\server\data-integration-server\start-pentaho.bat type= own start= auto.**
3. Now create one for the Pentaho Enterprise console: **sc create PentahoEnterpriseConsole binpath= C:\pentaho\server\enterprise-console\start-pec.bat type= own start= auto.**

The Pentaho DI Server will now start at boot time.

## Workstation Installation Procedures

---

To install and configure the Data Integration client tools from the archive package, follow the below procedures in the order they are presented. This process should take less than 10 minutes.

### Workstation Archive Package Deployment

Follow the below instructions to install the Data Integration client tools on your workstations. If you intend to deploy the client tools and servers on one machine, you can combine this with [Server Installation Procedure](#) on page 10, or use the PDI graphical installer.

 **Note:** The example commands in this and other sections are specific to Linux. You will have to adjust or ignore them on other operating systems.

1. Create a **/pentaho/design-tools/** directory in an appropriate place in your hierarchy.  
This directory should be accessible to all of the user accounts on this system that will run PDI.

 **Note:** If you are using the graphical installer, it will create this directory structure for you, so you can skip this step.

```
mkdir -p /usr/local/pentaho/design-tools/
```

2. Unpack the **pdi-ee-client-4.0.0-GA** archive to `/pentaho/design-tools/`.

```
tar zxvf pdi-ee-client-4.0.0-GA.tar.gz -C /usr/local/pentaho/design-tools/
```

3. Navigate to the `/pentaho/design-tools/license-installer/` directory.

4. Run the **install\_license.sh** script with the sole parameter of the location and name of your license file.

```
./install_license.sh install /home/rwilco/downloads/Pentaho\ PDI\ Enterprise  
\ Edition.lic
```

The Data Integration client tools are now installed.

# Upgrading or Importing Old PDI Content

---

This section contains instructions for four different content upgrade scenarios:

1. Converting an old database repository to a new enterprise repository
2. Importing old content files into a new enterprise repository
3. Moving your old content files to your new PDI directory
4. Upgrading from the old database repository to the new one

These upgrade paths are mutually exclusive, so you only need to follow one.

## The Advantages of an Enterprise Repository

---

The new enterprise repository in PDI 4.0 offers content management functionality beyond the traditional database repository:

- Secure, selective content sharing
- Content scheduling directly from Spoon, or from Enterprise Console
- Version history

These are key features for PDI production environments. Even if you don't think you'll be using some of them, it's a good idea to upgrade to an enterprise repository because that will be the focus of PDI Enterprise Edition development in the future.

## When NOT to Upgrade

While it is almost certainly advantageous for all existing Pentaho customers to upgrade to PDI 4.0, there are a few imaginable scenarios in which you may not want to upgrade from a file-based or database repository to the new enterprise repository. If you are managing your PDI database repository with a source control system, then you should probably stay with that instead of switching to an enterprise repository, which cannot be managed in that way. If you only have a few transformations or jobs that you are emailing to other PDI users and do not have the time, equipment, or expertise to manage a DI Server, it may not be advantageous to you to switch to an enterprise repository. Lastly, if you are programmatically generating jobs or transformations, you will not be able to integrate that process into the enterprise repository.

## Moving Old Content Files to a New Location

---



**Note:** If you haven't saved any KTR or KJB files in the top-level data-integration directory, there is no need to follow this process.

The default location to save jobs and transformations to is the top-level PDI directory. If you have saved your content files to this directory or a subdirectory therein, you must move them before you can remove the old copy of PDI.

While you can simply copy all of the content files into your new data-integration directory, it would be better to create a directory outside of the standard PDI folder to store content files. This prevents you from having to deal with the issue of moving content files from old PDI directories during future upgrades.

Once all of your KTRs and KJBs have been moved out of the old PDI directory, you should be able to access them normally in their new location through the file dialogues in the new version of Spoon.



**Note:** PDI 4.0 has a new option for file-based content repositories. This is only useful in specific development scenarios that involve importing and exporting content files stored in an enterprise repository, and other rare use cases. Under all other circumstances, you do not need to create a file-based repository. Simply continue to save and open files without any repository as you have in the past.

## Upgrading a Database Repository to the New DB Schema

---

Follow the below procedure to import your backed-up database repository and convert it to the new schema in PDI 4.0.

 **Note:** There is an automatic upgrade function in Spoon that will convert an older database repository to the new repository schema. Under most circumstances, it works perfectly. However, late in the testing cycle, some odd upgrade problems were discovered. They have all been fixed, but there may still be a chance of a silent failure or bug in the procedure. Therefore, if you have a smaller repository that won't be difficult to test by hand after the upgrade, feel free to try the upgrade function. Otherwise, follow the below procedure to import your backed-up PDI repository.

1. Using your preferred database management tool or method, create a new database or table space for your new PDI 4.0 repository.
2. Start Spoon.
3. When prompted for a repository connection, click the round green + icon in the upper right corner of the window.  
The **repository type** dialogue will appear.
4. Select **Kettle database repository**, then click **OK**.  
The **Repository information** dialogue will appear.
5. Select your old PDI database connection from the **Database Connection** drop-down box, then click **Edit**.  
The **Database Connection** dialogue will appear.
6. Change the **Database Name** from the old name to the one you just created for your new PDI 4.0 repository, then click **OK**.
7. Type in a short name and a descriptive name for this new repository connection in the **ID** and **Name** fields, respectively, then click **Create or Upgrade**.
8. Click **Yes** when asked if you're sure you'd like to continue.
9. When asked if you'd like to try a dry run to evaluate the SQL, choose whichever option appeals to you.
10. If the operation was successful, a message will pop up to inform you; click **OK** to continue, then **OK** again at the Repository information screen.
11. Enter in your repository username and password in the appropriate fields, then click **OK**.  
The default credentials are **admin** and **admin**.
12. Go to the **Tools** menu, select the **Repository** submenu, then click **Import Repository....**
13. Navigate to your backed-up PDI repository, select it, then click **OK**.
14. Double-click the top-level directory to import the entire repository.  
A **Repository Import** dialogue will appear, and show you the import progress.
15. When your content is done importing, click **Close**.

Your old PDI repository has been successfully upgraded to the new schema and imported into PDI 4.0.

## Switching to an Enterprise Repository

---

Follow the instructions in the subsections below to switch from a collection of content files or an old database repository to a new enterprise repository.

### Creating or Connecting to an Enterprise Repository

Follow the below instructions to create a new enterprise repository from a PDI workstation, or to connect to an existing repository.

1. Start Spoon by running the `/pentaho/design-tools/data-integration/Spoon` script. The **Repository Connection** dialogue will appear.
2. Click the round green **+** icon in the upper right corner of the window. The **Repository Type** dialogue will appear.
3. Select **Enterprise Repository** in the list, then click **OK**. The **Repository Configuration** dialogue will appear.
4. Ensure that the **URL** field corresponds to your DI Server address and port number. Type in a system-identifiable value (a unique internal name for this repository instance) in the **ID** field, and a friendly name or description in the **Name** field.  
If you only intend to have one repository for all users, you can un-check the **Show this dialogue at startup** option before clicking OK. This will prevent the dialogue from appearing every time you start Spoon. If you need to make repository connection changes later, you can still get to this screen through the **Tools** menu.
5. Use the default credentials of **admin** and **secret** for this repository, and click **OK** to complete repository configuration.  
This account is part of the default PDI configuration. Refer to the *PDI Administrator's Guide* to learn more about setting up users and roles in PDI.

You are now connected to an enterprise repository, and are enabled to begin creating users and roles for your organization.

## Converting a Database Repository to an Enterprise Repository

Follow the below procedure to import your old database repository into your new enterprise repository.



**Note:** There is an automatic upgrade function in Spoon that will convert an older database repository to a new enterprise repository. Under most circumstances, it works perfectly. However, late in the testing cycle, some odd upgrade problems were discovered. They have all been fixed, but there may still be a chance of a silent failure or bug in the procedure. Therefore, if you have a smaller repository that won't be difficult to test by hand after the upgrade, feel free to try the upgrade function. Otherwise, follow the below procedure to import your backed-up PDI repository.

1. Start Spoon and connect to your new enterprise repository.
2. Go to the **Tools** menu, select the **Repository** submenu, then click **Import Repository....**
3. Navigate to your backed-up PDI repository, select it, then click **OK**.
4. Double-click the top-level directory to import the entire repository.  
A **Repository Import** dialogue will appear, and show you the import progress.
5. When your content is done importing, click **Close**.

Your old PDI repository has been successfully imported into a PDI 4.0 enterprise repository.

## Importing Old Content Files Into an Enterprise Repository

If you have not been using a database repository to store your PDI content and do not wish to make any changes to this situation, you can skip this step; PDI jobs and transformations do not need to be upgraded. The process below explains how to upgrade from a directory of files to an enterprise repository.



**Note:** Currently there is no quick and easy way to accomplish this process. If you have a lot of KTRs and KJBs, this manual upgrade process could take some time, especially if you have multiple references to other PDI files in your jobs and transformations.

1. Start Spoon and connect to your new enterprise repository.
2. Go to the **File** menu and click **Import from an XML file**.
3. Navigate to your content file directory, select the file you want to import, then click **OK**.
4. Go to the **File** menu and click **Save**, and save the file to the repository as you normally would.

5. Repeat this process for all of your content files.



**Note:** If you have any references to other job or transformation files in your saved jobs, you must update each of those references to point to the new location in the enterprise repository. This requires that you import the referenced content before editing the job.

Your old content files are now imported into your new enterprise repository.

## Note on Schedules

---

In previous versions of PDI, the preferred method of scheduling content was through the BI Platform via an action sequence. This method will still work, assuming you have a BI Platform instance accessible and running, and your action sequences still point to the correct files in the correct locations.

In PDI 4.0, the preferred scheduling mechanism is the DI Server. Content can easily be scheduled from within Spoon, or from the Pentaho Enterprise Console. All new content should be scheduled in this way so that it can be properly managed through the built-in scheduling interface.

As you update old content -- and especially if you've imported content files into an enterprise repository -- you must deactivate each action sequence and transfer its work to the scheduling interface in Spoon.

## Testing and Cleanup

---

You should now have a complete PDI 4.0 environment, from the DI Server to individual client workstations. Before you go back into production, you should perform the following tests:

- Open old jobs and transformations and ensure that they execute properly.
- Ensure that each PDI workstation has an enterprise repository connection.
- Create a new job, transformation, and/or Agile BI analysis schema.
- Schedule a job or transformation and ensure that the schedule executes properly.
- From one PDI user workstation, share a job or transformation with another user and verify that this other user can access it.
- Physically restart the server and ensure that the DI Server and Enterprise Console are automatically started as services.

Once you're certain that your PDI environment is ready for production, you can remove any installation artifacts, such as ZIP or tar.gz archives and installers. You can also remove your old PDI database repository if there is one, and the old data-integration directory.



**Note:** You should now continue on to the *PDI Administrator's Guide* for further guidance on system administration, configuration, and maintenance.

## FAQ: New Features in 4.0

---

Pentaho customers have frequently asked the following questions about Pentaho Data Integration 4.0:

### **Do I need Carte at all, or does the Data Integration Server replace it?**

The DI Server is the Enterprise Edition solution for executing and scheduling remote PDI jobs and transformations. Carte will still be available as the open source alternative. Future Pentaho development efforts will be geared toward DI Server.

### **Does the enterprise repository always use the local H2 database, or can the database be remote and of a different type, like SQL Server, etc.? My old Kettle database repository was able to reside on any DB type and on a remote machine.**

The enterprise repository persistence today is delegated to the JackRabbit CMS. You can read about the options for persistence here: <http://wiki.apache.org/jackrabbit/PersistenceManagerFAQ#ObjectPersistenceManager>.

Any changes you may make to the enterprise repository implementation or configuration are not supported by Pentaho. This is a "black box" offering that does not allow for extra configuration support.

### **Is the new enterprise repository intended to replace the older Kettle database repository for Enterprise Edition customers? Should I use it to replace my current database or file-based repository?**

The enterprise repository will not automatically replace the database repository; if you want to convert to it, you must do this yourself. The *Upgrading to Pentaho Data Integration 4.0 Guide* explains how to do this. The enterprise repository provides several key features beyond the old database repository: versioning, collaborative development, and enhanced security. If you do not need any of these things, then there is no harm in staying with (or creating a new) file-based or database repository.

### **Is there a command line import tool/utility that takes a KTR or KJB file and transfers to a repository? I know how to do this in Spoon, but our consumers need an automated (non-UI) way to import jobs into the enterprise repository.**

This is a current feature request that may be added in a future PDI release: <http://jira.pentaho.com/browse/PDI-26>.

### **When jobs are started in Spoon via the Schedule function in the Action menu, are they stored on the DI Server? Will they be visible through Enterprise Console?**

Yes, they go by default to your DI Server as defined by the enterprise repository connection. Executions are visible through Enterprise Console as long as the DI Server server that executes the schedule is the same DI Server server that is registered in Enterprise Console.

### **Is Enterprise Console connected to Carte, or to the DI Server? It asks to register for Carte, but with PDI 4.0 I would prefer to connect it to the DI Server instead so that my jobs are monitored in Enterprise Console.**

When you register a remote execution server in Enterprise Console, it can be a Carte server or a Data Integration Server. The registration dialog says "Carte," but it will accept a DI Server server registration as well. Only one remote execution server can be registered at a time through Enterprise Console.

### **Can I auto-register the DI Server server in Pentaho Enterprise Console?**

Yes. You can add a SQL statement like the one shown below to the `/pentaho/server/enterprise-console/resource/config/hsqldb/pentaho-mgmt-svcs.script` script.

```
INSERT INTO PAC_PROPERTIES VALUES('PDI','pdi.carte.username','joe')
```

```
INSERT INTO PAC_PROPERTIES VALUES('PDI','pdi.carte.password','password')
INSERT INTO PAC_PROPERTIES VALUES('PDI','pdi.carte.service','http://
localhost:9080/pentaho-di/kettle/')
```

### **Where is the new scheduling Web services API?**

You can access the code for this API in the **bi-platform-scheduler2** open source project: <http://source.pentaho.org/viewvc/svnroot/bi-platform-v2/trunk/bi-platform-scheduler2/>.