Upgrading From PDI 4.0 to 4.1.0

Help and Support Resources

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

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

For information about how to purchase support or enable an additional named support contact, please contact your sales representative, or send an email to sales@pentaho.com.

For information about instructor-led training on the topics covered in this guide, visit http://www.pentaho.com/training.

Limits of Liability and Disclaimer of Warranty

The author(s) of this document have used their best efforts in preparing the content and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these programs or the documentation contained in this book.

The author(s) and Pentaho shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims.

Trademarks

Pentaho (TM) and the Pentaho logo are registered trademarks of Pentaho Corporation. All other trademarks are the property of their respective owners. Trademarked names may appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, Pentaho states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

Company Information

Pentaho Corporation
Citadel International, Suite 340
5950 Hazeltine National Drive
Orlando, FL 32822
Phone: +1 407 812-OPEN (6736)
Fax: +1 407 517-4575
http://www.pentaho.com

E-mail: communityconnection@pentaho.com

Sales Inquiries: sales@pentaho.com

Documentation Suggestions: documentation@pentaho.com

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

Contents

Introduction

This guide shows current Pentaho Data Integration Enterprise Edition customers how to upgrade from PDI 4.0.1 to 4.1.0. If you are upgrading from a version of PDI older than 4.0.1, refer instead to the *Upgrading to Pentaho Data Integration 4.0* guide, and be sure to use 4.1.0 packages instead.

Note: This guide is only for Enterprise Edition upgrades. You cannot upgrade a PDI 4.0.1 Community Edition installation to a PDI 4.1.0 Enterprise Edition instance using this process.

What's new in 4.1.0?

Pentaho Data Integration 4.1.0 offers a variety of expanded cloud computing features:

Note: Most of this functionality requires a Pentaho BI Suite For Hadoop license entitlement.

- **Hadoop integration:** PDI can now be deployed on a Hadoop node, and used for running Hadoop jobs.
  - **Impact:** If you intend to deploy PDI on your Hadoop cluster and create Hadoop jobs, consult the relevant PDI installation, administration, and user instructions are available in the Pentaho Knowledge Base.
- **Hadoop MapReduce support :** PDI now has the ability to use jobs to coordinate Hadoop job execution, and transformations as MapReduce jobs in Hadoop. Amazon Elastic MapReduce customers can use PDI to create EMR-based jobs.
  - **Impact:** Developers who design Hadoop and PDI jobs should consult the latest PDI documentation to learn how to use these new features.
- **Hive JDBC support:**
  - **Impact:** If you intend to deploy PDI on your Hadoop cluster and create Hadoop jobs, you must expand your Pentaho support entitlement and install a new Pentaho BI Suite For Hadoop license key. Relevant installation, administration, and user instructions are available in all of the standard PDI documentation in the Pentaho Knowledge Base.
- **Improved virtual filesystem dialogues:** new Amazon S3 and HDFS virtual filesystem dialogues make it easier to execute file operations on S3 accounts and Hadoop nodes.
  - **Impact:** If you intend to deploy PDI on your Hadoop cluster and create Hadoop jobs, you must expand your Pentaho support entitlement and install a new Pentaho BI Suite For Hadoop license key. Relevant installation, administration, and user instructions are available in all of the standard PDI documentation in the Pentaho Knowledge Base.

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 4.1.0.

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

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 PDI 4.1.0 client and server packages from the Pentaho Knowledge Base or Enterprise Edition FTP site.	
Step 2	Stop your DI Server and Pentaho Enterprise Console server. **Ensure that the DI Server process is stopped;** if it is not, file copy and delete commands could silently fail later on.	
Step 3	Back up your content files, database repository, or enterprise repository; your ~/.kettle directory; and your `/pentaho/server/data-integration-server/pentaho-solutions/` directory. **Modify this and all future paths to match your PDI instance.**	
Step 4	On your PDI server, rename the `/pentaho/server/data-integration-server/` directory to **data-integration-server-old**.	
Step 5	Create a new, empty `/pentaho/server/data-integration-server/` directory.	
Step 6	Rename the `/pentaho/server/enterprise-console/` directory to **enterprise-console-old**.	
Step 7	Create a new, empty `/pentaho/server/enterprise-console/` directory.	
Step 8	Unpack the new **pdi-ee-server-4.1.0-GA** archive to `/pentaho/server/data-integration-server/`.	
Step 9	Copy all of the **applicationContext** files from `/pentaho/server/data-integration-server-old/pentaho-solutions/system/` (the old solutions directory) to the new one.	
Step 10	Copy the **pentaho-spring-beans.xml** file from `/pentaho/server/data-integration-server-old/pentaho-solutions/system/` (the old solutions directory) to the new one.	
Step 11	Transfer the admin role information and merge any custom changes from your old **pentaho.xml** and **repository.spring.xml** files to the new ones.	
Step 12	Copy the entire **quartz** directory from `/data-integration-server-old/pentaho-solutions/` to the new one.	
Step 13	Copy the entire **repository** directory from `/data-integration-server-old/pentaho-solutions/system/jackrabbit/` to the new one.	
Step 14	Copy the entire **jre** directory from `/data-integration-server-old/` to the new one.	
Step 15	Merge any custom changes from your old DI Server configuration files to the new ones.	
Step 16	Copy the following files from the old Pentaho Enterprise Console to the new one: console.xml, console.properties, login.properties, login.conf, log4j.xml.	
Step 17	Start the DI Server and Pentaho Enterprise Console and test their functionality and availability.	
Step 18	On your PDI workstations, delete the `/pentaho/design-tools/data-integration/` directory after you have ensured that there are no KJB or KTR content files stored there.	
Step 19	Unpack the **pdi-ee-client-4.1.0-GA** archive to the `/pentaho/design-tools/` directory.	
Step 20	Start the Data Integration client tools that you normally use, and ensure that they work properly, can access existing content, create and share new content, run existing schedules, create new schedules, and have a connection to the DI Server (if you are using it).	
Step 21	Perform any other necessary testing, then delete any installation artifacts and the **data-integration-server-old** directory, and inform users that the upgrade is complete.	

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 and Data Integration client tools. Here are the packages you need for each platform and distribution:

- **DI Server for Windows:** `pdi-ee-server-4.1.0-GA.zip`
- **DI Server for Linux/Solaris/OS X:** `pdi-ee-server-4.1.0-GA.tar.gz`
- **Data Integration client tool Windows package:** `pdi-ee-client-4.1.0-GA.zip`
- **Data Integration client tool Linux/Solaris/OS X package:** `pdi-ee-client-4.1.0-GA.tar.gz`
- **Data Integration for Hadoop:** `phd-ee-4.1.0-GA.zip`

If you download the **pdi-ee-server** package, you must also download the Pentaho Enterprise Console package:

- **Pentaho Enterprise Console for Linux/Solaris/OS X:** `pec-3.8.0-GA.tar.gz`
- **Pentaho Enterprise Console for Windows:** `pec-3.8.0-GA.zip`

Creating Backups

You should back up your content files or repository and your Kettle settings in case something goes wrong with the upgrade. Refer to the sections below that apply to your situation.

Backing Up Content Files

If you do not use a database or enterprise 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 complete repository to XML** functionality in Spoon's **Repository Explorer** dialogue, which is accessible from the **File** menu. 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.

How to Back Up the Enterprise Repository

Follow the instructions below to create a backup of your PDI enterprise repository.

Note: If you've made any changes to the Pentaho Enterprise Console or DI Server Web application configuration, such as changing the port number or base URL, you will have to modify this procedure to include the entire `/pentaho/server/` directory.

1. Stop the DI Server.

   ```
   /pentaho/server/data-integration-server/stop-pentaho.sh
   ```

2. Create a backup archive or package of the `/pentaho/server/data-integration-server/pentaho-solutions/` directory.

   ```
   tar -cf pdi_backup.tar /pentaho/server/data-integration-server/pentaho-solutions/
   ```

3. Copy the backup archive to removable media or an online backup server.
4. Start the DI Server.

   ```
   /pentaho/server/data-integration-server/start-pentaho.sh
   ```

Your DI Server's stored content, settings, schedules, and user/role information is now backed up.

To restore from this backup, simply unpack it to the same location, overwriting all files that already exist there.

Backing Up the .kettle Directory

The **.kettle** directory stores all of your client tool 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 the PDI client tools are installed to.

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

Upgrading a Data Integration Server

Ensure that the DI Server and Pentaho Enterprise Console are stopped before continuing. You must have a PDI 4.0.1 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 4.1.0.

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-4.1.0-GA** package to the parent of the directory you just renamed.

   ```
   tar zxvf ~/downloads/pdi-ee-server-4.1.0-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. Open the **repository.xml** file in your new `/pentaho-solutions/system/jackrabbit/` directory. Find the **WorkspaceSecurity** element and copy it to your text buffer. Now edit the **workspace.xml** files in the following directories and paste the new WorkspaceSecurity element over the old one:

   - `/data-integration-server/pentaho-solutions/system/jackrabbit/repository/workspaces/default/`
   - `/data-integration-server/pentaho-solutions/system/jackrabbit/repository/workspaces/security/`

   Note: The security class names have changed in the new DI Server. This step replaces the old class names in your old workspace.xml files with the new ones in the new repository.xml. This will not affect any security or permissions settings that you've implemented.

   ```
   <WorkspaceSecurity>
   ```

```
    <AccessControlProvider
 class="org.pentaho.platform.repository2.unified.jcr.jackrabbit.security.PentahoAccessContr
        <!-- must match /Repository/Security/LoginModule/param -->
        <param name="adminRole" value="pentahoRepoAdministrators" />
        <param name="helperClass"
 value="org.pentaho.platform.repository2.unified.jcr.jackrabbit.security.DefaultPentahoJack
>
        <!-- users with administerSecurity get Permission.ALL if effective ACL is at
 tenant root or below -->
        <param name="wildcardDynamicMask0" value="/pentaho/
{0}=org.pentaho.security.administerSecurity:4095" />
        <!-- users with read get Permission.READ and Permission.READ_ACL if effective
 ACL is at tenant root -->
        <param name="dynamicMask0" value="/pentaho/
{0}=org.pentaho.repository.read:33" />
        <!-- users with read get Permission.READ and Permission.READ_ACL if effective
 ACL is at tenant etc/pdi -->
        <param name="dynamicMask1" value="/pentaho/{0}/etc/
pdi=org.pentaho.repository.read:33" />
        <!-- users with create get Permission.READ and Permission.READ_AC and
 Permission.MODIFY_AC and Permission.LOCK_MNGMT and Permission.VERSION_MNGMT and
 Permission.NODE_TYPE_MNGMT and Permission.REMOVE_NODE and Permission.REMOVE_PROPERTY
 and Permission.ADD_NODE and Permission.SET_PROPERTY if effective ACL is at tenant
 etc/pdi -->
        <param name="dynamicMask2" value="/pentaho/{0}/etc/
pdi=org.pentaho.repository.create:1023" />
    </AccessControlProvider>
</WorkspaceSecurity>
```

9. 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.

10. 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.

11. 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 4.1.0. Continue on to the next subsection to upgrade the Pentaho Enterprise Console.

Upgrading the Pentaho Enterprise Console

The upgraded DI Server will not work properly without upgrading the Pentaho Enterprise Console. To upgrade, follow the below process.

1. Rename the `/pentaho/server/enterprise-console/` directory to **enterprise-console-old**.

   ```
   mv enterprise-console enterprise-console-old
   ```

2. Unpack the `pec-ee-3.8.0-GA` zip or tar.gz file to `/pentaho/server/`.

   ```
   tar zxvf ~/downloads/pec-ee-3.8.0-GA.tar.gz -C /home/pentaho/pentaho/server/
   ```

3. Copy the following files from your old `/pentaho/server/enterprise-console-old/resource/config/` directory into the new one:

   - console.xml
   - console.properties
   - login.properties
   - login.conf

- log4j.xml

```
cp /home/pentaho/pentaho/server/enterprise-console-old/resource/config/console.* /
home/pentaho/pentaho/server/enterprise-console/resource/config/ && cp /pentaho/server/
enterprise-console-old/resource/config/log* /home/pentaho/pentaho/server/enterprise-
console/resource/config/
```

The Pentaho Enterprise Console has been upgraded to version 3.8.

Upgrading a Data Integration Workstation

Ensure that the Data Integration client tools (Spoon, Pan, Kitchen) are not running on the machine before proceeding.

Note: If you use an enterprise repository to store your jobs and transformations, you should perform the DI Server upgrade process before this in order to properly test DI Server connectivity from your upgraded workstations.

Follow the directions below to upgrade your PDI workstations to version 4.1.0. Adjust the installation paths to match your scenario.

1. Remove your existing `/data-integration/` directory.

   ```
   rm -rf /home/rwilco/pentaho/design-tools/data-integration/
   ```
2. Unpack the **pdi-ee-client-4.1.0-GA** package to the same location that you just deleted.

   ```
   cd /home/rwilco/pentaho/design-tools/ && tar zxvf ~/downloads/pdi-ee-client-4.1.0-GA.tar.gz
   ```
3. Start Data Integration and ensure that you can still connect to the DI Server (if you have an enterprise repository connection) and access all of your old content.

This workstation is now upgraded to PDI 4.1.0-GA. Repeat this procedure for other workstations that you have support entitlements for.

Testing and Cleanup

You should now have a complete PDI 4.1.0 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.
- If you are using an enterprise repository, ensure that each PDI workstation can connect to it.
- Create a new job, transformation, and/or Agile BI analysis schema and save it as you normally would.
- Schedule a job or transformation and ensure that the schedule executes properly.
- Ensure that existing schedules are still valid.
- If you are using an enterprise repository, share a job or transformation between PDI users and verify that both can access it.
- Physically restart the server and ensure that the DI Server and Enterprise Console are automatically started as services, if you have them configured as such.

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 **data-integration-server-old** directory.