pentaho

Version 5.3

Administer BA Server

Copyright Page

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.

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

Introduction

Prerequisites

Before you do any of these tasks, you must have installed the Pentaho software and configured the BA Server.

Expertise

The topics covered in this section are written for IT administrators who know where data is stored, how to connect to it, details about the computing environment, and how to use the command line interface for Windows or Linux.

Tools

We provide a web application, the User Console, which you use to perform many administration tasks.

Login Credentials

All of the tasks in this section, as well as those that use the User Console, **Administration** page, require that you log on to the User Console with the administrator user name and password.

Manage Schedules

As an administrator, you may create, run, delete, stop, edit, or resume schedules through the **Schedules** page in the User Console. After a report is scheduled by any user, an admin may edit, pause, or delete that schedule, as well as change the frequency of the report, change parameters, or adjust email settings. To distribute reports using email, make sure you have configured the email server. You also have the ability to prevent users from scheduling specific files or reports.

You might need to control when schedules can run, by setting up periods of time to forbid schedules. This allows you to perform administrative functions, such as system maintenance or managing server traffic during peak usage times.

Prevent Scheduling of a Single Report or File

All new reports are enabled for scheduling by default, but as an administrator, you can control which files can be scheduled to run.

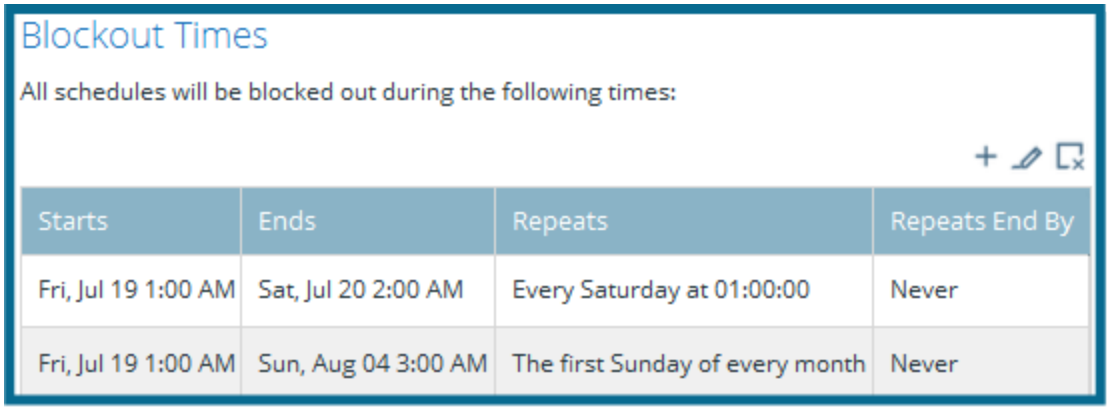
1. From the User Console **Home** menu, click **Browse Files**. The **Browse Files** page appears.
2. Browse to the location of a report from the **Browse** pane on the left of the **Browse Files** page. The **Files** pane in the center populates with a list of reports.
3. Click to select the report for which you want to deny scheduling, then click **Properties** in the **Folder Actions** pane. The **Properties** window for that report appears.
4. Click the **General** tab in the **Properties** window, then disable the box next to **Allow Scheduling**.
5. Click **OK**.

Users are no longer able to see the **Schedule** option next to this report and the report is not able to be scheduled.

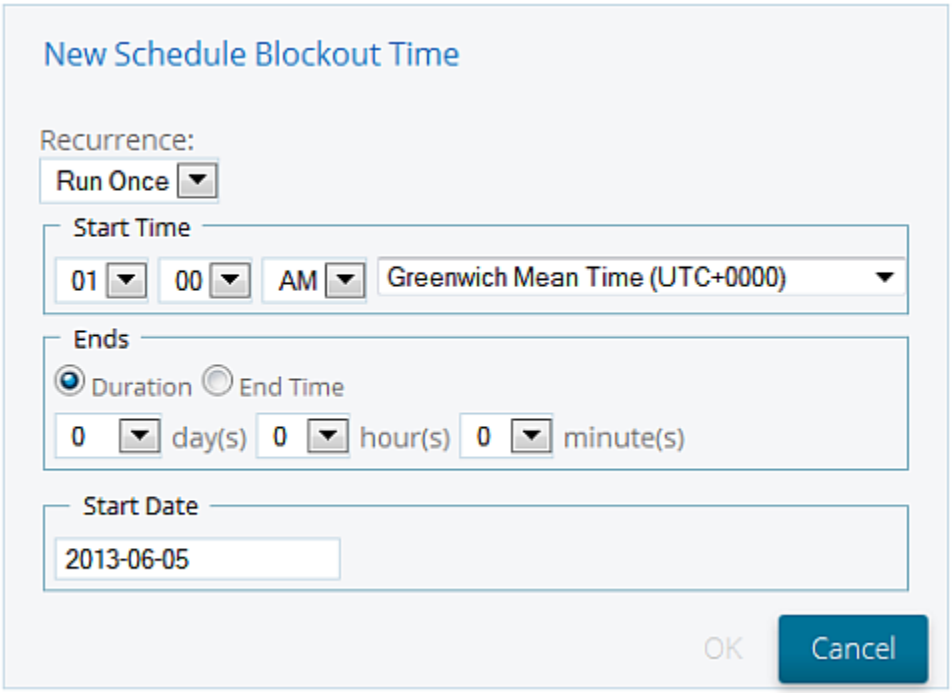
Prevent Scheduling by Setting Blockout Times

Specific times can be designated on the server to block the running of schedules. This allows you to perform administrative functions such as system maintenance or managing server traffic during peak usage times.

1. From the User Console **Home** menu, click **Schedules**. The **Schedules** page appears.



2. Click the **Create Blockout Time** button or the green **+** in the **Blockout Times** widget on the left side of the **Schedules** screen. The **Blockout Time** dialog box appears.



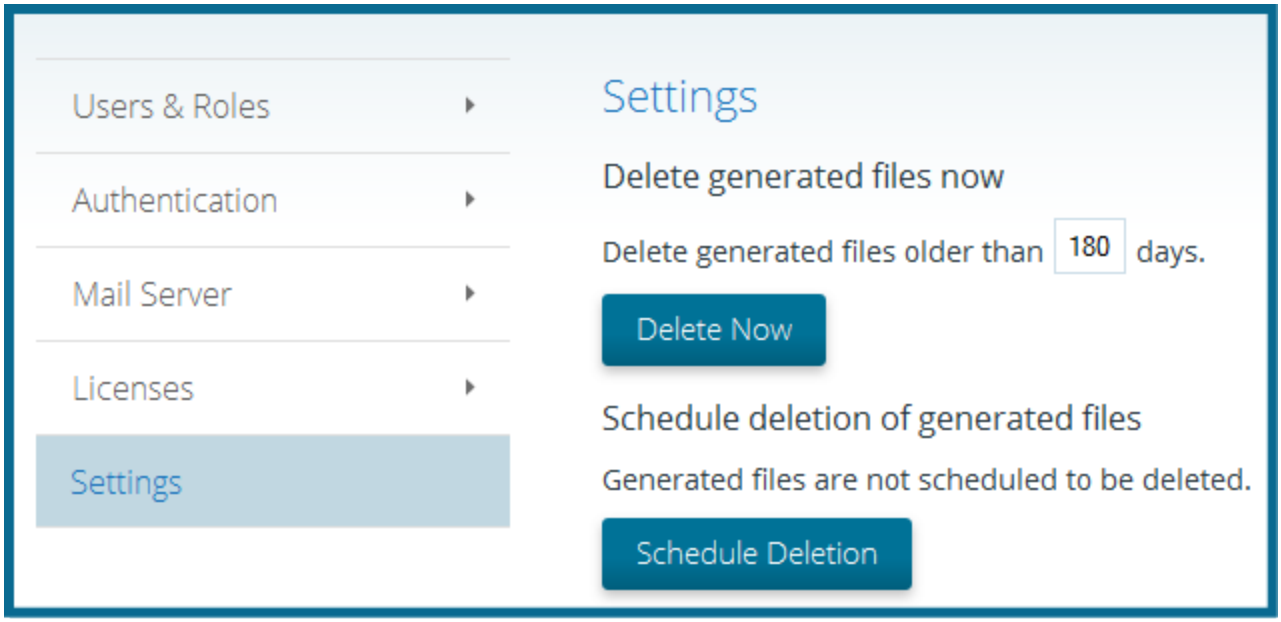
3. Choose a **Recurrence**, **Start Time**, **Duration** or **End Time**, and a **Start Date** for the blocked out time using the available fields.

4. Click **OK**. If **Run Once** is chosen as the **Recurrence**, a dash shows under the Start Times and End Time fields in the blockout list, until the blockout time passes.

The **Blockout Time** is created and no schedules will run on the BA Server during that time. Users are able to view a list of blocked out times when they are creating schedules.

Schedule Deletion of Generated Files

As an administrator, you may want to delete older generated files or content in order to free up space in the BA Repository.



1. Click on **Administration** in the drop-down menu on the upper right, then click on **Settings**. The **Settings** interface appears.
2. To manually delete generated files now, confirm the age of the files to be deleted, then press **Delete Now**. The generated files older than the specified age are deleted.
3. To schedule regular deletion of generated files, press **Schedule Deletion**. The **Schedule Deletion** window appears.
4. Enter an age for the generated files in the field specifying number of days.
5. Choose a time interval for the schedule from the **Recurrence** list.
6. Choose a **Start Time** and a **Recurrence Pattern**, as well as **Start** and **End Dates** for the scheduled deletion. Click **OK**.

The **Deletion Schedule** is created and the generated files older than the specified age are deleted during the scheduled times.

These sections explain how to navigate the **Schedules** page, how to use the page to create, edit, and delete schedules, and describe reference data on the Quartz Cron engine for scheduling.

- Schedules
- About Scheduling Reports
- Quartz Cron Attributes

Upload and Download from the BA Repository

Uploading and downloading from the BA Repository is done with the User Console or the command line interface. The ability to upload and download assumes that you have already created a data source, that data content exists to be pushed, and defines permissions for the repository.

For uploading, any starting location can be selected. Permission settings are inherited through the folder structure if the destination location has existing permission settings. It is advisable to keep existing security settings as defaults for the upload. Uploading more than one file or folder requires the files to be zipped before beginning the upload.

For downloading, you are able to select the destination location for the downloaded file or folder. The download process always creates a .zip file that includes a manifest file along with the downloaded content. The manifest file contains the collection of permissions settings for the downloaded files and folders and is found in the root directory of the .zip file.

Supported File Types	Hidden File Types
These are the supported file types and artifacts for uploading and downloading from the BA Repository.	These are the file types that are hidden by default in the BA Repository.
• Reporting (.prpt, .prpti, .xml) • Analyzer (.xanalyzer) • Dashboards (.xdash) • Solution Files (.xaction, .locale)	• Web (.html, .htm) • Reporting (.xml) • Solution Files (.properties) • Graphics (.png, .jpg, .gif, .svg)

Upload Folders and Files

The User Console can be used to upload files and folders to the BA Repository. A quick word about manifests: when a user downloads content out of the repository, a manifest file is created within the zip file which contains the permission and ownership information about the file.

NOTE:

For **Retain permission on upload file**, the file permission contained in the uploaded zip (`exportManifest.xml`) will be the permission applied the repository. If the file doesn't have an entry in the `exportManifest.xml` for the permission, then it will use the default permission, which is `inherit`. This is equivalent to the command line switch, `--permission=true`.

NOTE:

For **Set Owner based on uploaded file**, the owner found in the uploaded zip (`exportManifest.xml`) will be the owner of the file in the repository. If the file doesn't have an entry in the `exportManifest.xml` for the `Owner`, then it will set the `Owner` to the user who is uploading the zip. This equivalent to the command line switch, `--retainOwnership=true`.

These steps walk you through uploading to the repository with the User Console.

1. From the User Console **Home**, click **Browse Files**. The **Browse Files** page appears.
2. From the **Browse** pane on the left, click to choose the destination folder for the upload.
3. With the destination folder highlighted, click **Upload** in the **Folder Actions** pane on the right. The **Upload** dialog box appears.
4. Browse to the file or folder to be uploaded by clicking **Browse**.
5. Click **OK** to begin upload using the default settings.
6. Choose preferences for the upload by clicking to expand the **Advanced Options** menu.

g. A. Choose **Replace the Existing File** or **Do Not Upload** from the first menu.

h. B. Choose **File Permissions** from the second menu. The choices are **Do Not Change Permissions** or **Retain Permissions on the Uploaded File**.

i. C. If you selected **Retain Permissions on the Uploaded File**, choose **File Ownership** by selecting **Do Not Change Owner** or **Set Owner Based on Uploaded File** from the third menu.

j. D. Choose **None**, **Short**, or **Verbose** from the Logging menu.

7. Click **OK**.

The upload runs and the files or folders are uploaded to the repository. If the upload fails, an error log window opens with specific information.

Upload from the Command Line

1. Open the command line interface by clicking **Start** and typing **cmd**. Press **Enter**.
2. From the command line interface, go to the location where you have a local copy of Biserver-EE installed, such as `C:/dev/pentaho/biserver-ee`.
3. Enter a space, then type the arguments for upload into the command line interface. A completed upload argument would look something like this:

```
import-export.bat --import --url=http://localhost:8080/pentaho --
username=dvader --password=password --source=file-system --type=files --
charset=UTF-8 --path=/public--file-path=C:/Users/dvader/Downloads/pentaho-
solutions.zip --overwrite=true --permission=true -retainOwnership=true
```

1. Press **Enter** after the arguments are typed.

The upload process runs and the results are displayed in the command interface. If an argument is required for successful upload and has not been provided, the missing requirement is displayed in the command interface.

Command	Description	Values	Required
-i,--import	Upload Command	n/a	Yes
-a,--url <arg>	URL of the repository (for example, http://localhost:8080/pentaho).	URL	Yes
-c,--charset <arg>	Charset to use for the repository. Characters from external systems are converted to this charset	UTF-8 (default)	No
-f,--path <arg>	Repository path to which the uploaded files are added (for example, /public)	File path	Yes
-h,--help	Prints this message	n/a	No
-p,--password <arg>	Repository password	Alphanumeric	Yes
-u,--username <arg>	Repository username	Alphanumeric	Yes
-x,--source <arg>	External system type	legacy-db or file-system (default)	Yes
-o , --overwrite <arg>	Overwrites file(s) on upload. Default value is True.	Boolean	No
-m, --permission <arg>	Applies ACL using manifest file. Default value is True.	Boolean	No

Command	Description	Values	Required
-r, --retainOwnership	Replaces the file ownership upon upload with the ownership of the original download. Default value is True.	Boolean	No
-l, --logfile <arg>	Path to local file system with name of file to write	File path	No
-t, --type <arg>	The type of content being uploaded - files (default), metadata.	File type	No
-a_ds,--analysis-datasource <arg>	Analysis datasource type.	Alphanumeric	No
-a_param,--analysis-parameters <arg>	Analysis datasource parameters. Alphanumeric No	Alphanumeric	No
-a_xmla,--xmla-enabled <arg>	Analysis XMLA enabled flag.	Boolean	No
-cat,--catalog <arg>	Catalog description.	Alphanumeric	No
-ds,--datasource-type <arg>	Datasource type.	Alphanumeric	No
-m_id,--metadata-domain-id <arg>	Metadata domain ID.	Alphanumeric	No
-params,--params <arg>	Parameters to pass to REST service call.	Alphanumeric	No
-res,--resource-type <arg>	Import/Export resource type.	Alphanumeric	No
-rest,--rest	Use the REST (default) version (not local to the BA Server).	Alphanumeric	No
-s,--legacy <arg>	If True - use the legacy Import/Export on local BA server.	Boolean	No
-v,--service <arg>	This is the REST Service call, for example: ACL, children, properties.	URL	No

Download Folders and Files

Downloading folders and files can be done through the User Console or through the command line interface. The download process always creates a .zip file that includes a manifest file along with the downloaded content. The manifest file is a collection of the permissions settings for the downloaded files and folders and is found in the root directory of the .zip file.

Download a Folder

1. From the User Console **Home**, click **Browse Files**. The **Browse Files** page appears.
2. From the **Browse** pane on the left, browse to the location of the folder to be downloaded.
3. With the folder highlighted, click **Download** in the **Folder Actions** pane on the right.
4. Choose **Save File** in the window that appears, and click **OK**.

The folder is saved as a .zip file with the manifest located in the top level of the file.

Download a File

1. From the User Console **Home**, click **Browse Files**. The **Browse Files** page appears.
2. Browse to the location of the file by clicking through the folders in the **Browse** pane on the left. The **Files** pane in the center populates with a list of reports.
3. Click to select the file in the **Files** pane and choose **Download** in the **Folder Actions** pane on the right.
4. Choose **Save File** in the window that appears, and click **OK**.

The file is saved as a .zip file with the manifest located in the top level of the file.

Download from the Command Line

1. Open the command line interface by clicking **Start** and typing **cmd**. Press **Enter**.
2. From the command line interface, go to the location where you have a local copy of Biserver-EE installed, such as `C:/dev/pentaho/`biserver-ee.
3. Enter a space, then type the arguments for download into the command line interface. A completed download argument would look something like this:

```
import-export.bat --import --url=http://localhost:8080/pentaho --
username=dvader --password=password --source=file-system --type=files --
charset=UTF-8 --path=/public--file-path=C:/Users/dvader/Downloads/pentaho-
solutions.zip --overwrite=true --permission=true –retainOwnership=true
```

1. Press **Enter** after the arguments are typed.

The download process runs and the results are displayed in the command interface. The file is saved as a .zip file with the download manifest located in the top level of the file. If an argument is required for successful download and has not been provided, the missing requirement is displayed in the command interface.

Command	Description	Values	Required
-e, --export	Download command	n/a	Yes
-a,--url <arg>	URL of repository (for example, http://localhost:8080/pentaho)	URL	Yes
-c,--charset <arg>	Charset to use for the repository. Characters from external systems are converted to this charset.	UTF-8 (default)	No
-h,--help	Prints this message.	n/a	No
-f,--path <arg>	Repository path to which the uploaded files are added (for example, /public)	File path	Yes
-p,--password <arg>	Repository password	Alphanumeric	Yes
-fp,--filepath <arg>	Location that the zip file is downloaded to	File path	Yes
-u,--username <arg>	Repository username	Alphanumeric	Yes
-l, --logfile <arg>	Path to local file system with name of file to write	File path	No
-w, --withManifest <arg>	If true, includes Manifest.xml inside zip. If false, download excludes this file.	Boolean	No
-a_ds,--analysis-datasource <arg>	Analysis datasource type.	Alphanumeric	No
-a_param,--analysis-parameters <arg>	Analysis datasource parameters.	Alphanumeric	No
-a_xmla,--xmla-enabled <arg>	Analysis XMLA enabled flag.	Boolean	No
-cat,--catalog <arg>	Catalog description.	Alphanumeric	No
-ds,--datasource-type <arg>	Datasource type.	Alphanumeric	No
-m_id,--metadata-domain-id <arg>	Metadata domain ID.	Alphanumeric	No

Command	Description	Values	Required
-params,--params <arg>	Parameters to pass to REST service call.	Alphanumeric	No
-res,--resource-type <arg>	Import/Export resource type.	Alphanumeric	No
-rest,--rest	Use the REST (default) version (not local to the BA Server).	Alphanumeric	No
-s,--legacy <arg>	If True - use the legacy Import/Export on local BA server.	Boolean	No
-v,--service <arg>	This is the REST Service call, for example: ACL, children, properties.	URL	No

Localize Folders and Reports

You can localize the names and descriptions for reports and folders that appear in the User Console. Localization is helpful if your co-workers work in different countries and speak different languages, but use the console to access the same reports and folders. For example, localization allows German speakers to view report names in German and Americans to view the same report names in English.

NOTE:

Localization language packs have been created through a community initiative that uses an installer built by Webdetails, a Pentaho company. The language packs apply to both EE and CE, and are maintained by community members. The packs are not officially supported by Pentaho.

Localization information is stored in the BA Repository, along with other report and folder information. Typically, to localize names and descriptions of reports and folders you do three things.

- Download the report or folder from the BA Repository. Localization information is stored in one or more text files that have the **.locale** extension.
- Add or edit the localization files downloaded with the reports or folders.
- Re-upload the report or folder, along with the localization files, into the BA Repository.

If you want to create or add a Pentaho localization file, but are unfamiliar with its structure, we recommend that you read Localization File Structure first.

Localization File Structure

When you download a report or folder from the BA Repository, localization information is stored in a text file that has a `.locale` extension. You add or edit localization files, but must upload them to the BA Repository for the changes to appear in the User Console.

Each report and folder that appears in the console should have a default localization file associated with it. The default localization file should indicate the name that appears when you view the name of the report or folder in the console. Optionally, the localization file can contain the report description. The report description appears when you hover the mouse pointer over the report or folder name in the console.

If multiple localization files are present, the User Console displays the localization information contained in them if you set the language indicated in the localization file as the default in the console or in your web browser. Otherwise, the information in the default localization file is displayed instead.

Localization information appears in two places in a localization file: in the file name and in variables inside the file.

Localization File Names

Localization file names consist of several parts that are joined by underscores. This is an example of a localization file name for a folder: `index_es_PA.locale`

In folder localization, file names the word `index` appears first, as in the previous example. If the localization file is for a report, the name of the report appears instead of `index`, like this: `Inventory Report_es_PA.prpt.locale`

The second and third parts of the file name indicate the two-letter language code and the two-letter country code. Language codes adhere to ISO 639-1; country codes can be found in ISO-3166. In the previous example, `es` is the language code for Spanish and `PA` is the country code for Panama. A list of often-used language and country codes appears here.
Note: Although rarely used, the dialect code can appear after the country code.

File extensions, which appear after the period, indicate the report type and end with `.locale`. In the previous example `prpt` indicates the report type.

Language, country, and dialect codes are optional parts of file names. The report type extension should not be included in folder localization file names. Only the report name (or the word `index` for folders) is required as is the .locale extension. But, if a country code is present, the language code must also be present, and if a dialect code is present, both the language and country codes must be present.

If a localization file name contains no language, country, or dialect code, the console assumes that the file is the default for a report or folder. A default localization file name for a folder looks like this: `index.locale`.

Localization Variables

Localization files have at least two variables that contain localization information.

- **file.title** holds the name of the report or folder that appears in the User Console.
- **file.description** holds the text for the Tool Tip that appears when you hover the mouse pointer over the report or folder name in the User Console.

Here is an example of a localization file for a report.

```
#Locale = es
#Wed Apr 17 13:55:53 EDT 2013
file.title=Inventario Region 23
file.description=Lista del inventario de la región 23
```

Note: In localization `.properties` files, **file.name** and **file.url-name** are sometimes used instead of **file.title** and **url-description** is sometimes used instead of the file description.

Before variable values set in the localization file can be displayed, you must adjust either the User Console or your web browser's language so that it matches the localization file's language. You must also upload the localization file into the BA Repository along with the other report files.

Using the previous example, the User Console displays the value of **file.title** (**Inventario Region 23**) when you open it. If you hover the mouse pointer over the report name in the console, a Tool Tip appears that displays the value of **file.description** (**Lista del inventario de la region 23**).

Note: Unicode can be used to display non-Latin languages, such as Chinese or Japanese.

Popularly Used Country and Location Codes

Language and country codes that are used to construct localization file names appear in standards ISO 639-1 and ISO 3166. Here is a list of popularly used language and country codes.

Table 1. Popularly Used Country and Location Codes

Language	Language Code	Country	Country Code
Chinese	zh	China	CN
Dutch	nl	Netherlands	NL
French	fr	France	FR
German	de	Germany	DE
Italian	it	Italy	IT
Japanese	ja	Japan	JP
Korean	ko	Republic of Korea	KR
Portuguese	pt	Brazil	BR
Portuguese	pt	Portugal	PT
Spanish	es	Argentina	AR
Spanish	es	Spain	ES

Set Default Localization for Reports and Folders

Complete these steps if you want to set up default localization information.

1. Download the report or folder to which you want to add localization information. Instructions appear in Download Folders and Files.
2. Unzip the downloaded report or folder, then determine whether a default localization file already exists. Localization files have a .local extension and are named either `index.locale` (for folders) or `<report name>.<report type>.locale` for reports. If a default localization file exists you do not need to continue with these steps.
3. To add new localization information, you must create a default localization file. Localization file names follow very specific naming conventions. To determine the name of the new localization file, do this.
    a. For folders, the default localization file name convention is `index.locale`.
    b. For reports, the default localization file name convention is `<report name>.<report type>.locale`. An example of a valid file name is `Inventory Report.prpt.locale`.
4. Use a text editor to create a blank localization file that has the file name you constructed in the previous step, then save the file in the directory of the folder or report.

5. Type the following in the blank file.

```
#Localization File
file.title=
file.description=
```

6. Type the name of the report or folder that you want to appear in the User Console after the **file.title** variable. Type the name of the report or folder exactly as you want it to appear. Example: `file.title=Inventario`
7. Type the description for the report or folder after the file.description variable. The value of the **file.description** appears when you hover the mouse pointer over the report or folder name in the console. Type the description exactly as you want it to appear. Example: `file.description=Inventario para El Rey`
8. Save and close the file.
9. Upload the localization file along with other report or folder files into the BA Repository. Instructions for how to do this appear in Upload Folders and Files.

Localize for Additional Languages

Complete these steps if you want to display report and folder names and descriptions in a language other than the default. If you want to change the default language displayed only, we recommend that you edit the default locale file.

1. Download the report or folder to which you want to add localization information. Instructions appear in Download Folders and Files.
2. Unzip the downloaded report or folder.
3. To add new localization information, you must create a new localization file. Localization file names follow very specific naming conventions. You can optionally specify country and language codes, as well as the report type. To determine the name of the new localization file, do this.
    a. For folders, the localization file name convention is `index_<language code>_<country code>.locale`. An example of a valid file name is `index_es_PA.locale`. Note that the language and country codes are optional.
    b. For reports, the localization file name convention is `<report name>_<language code>_<country code>.<report type>.locale`. An example of a valid file name is `Inventory Report_es_PA.prpt.locale`. Note that the language and country codes are optional.
4. Use a text editor to create a blank localization file that has the file name you constructed in the previous step, then save the file in the directory of the folder or report.
5. Type the following in the blank file.

```
#Localization File
file.title=
file.description=
```

6. Type the name of the report or folder that you want to appear in the User Console after the **file.title** variable. Type the name of the report or folder exactly as you want it to appear. Example: `file.title=Inventario`

7. Type the description for the report or folder after the file.description variable. The value of the **file.description** appears when you hover the mouse pointer over the report or folder name in the console. Type the description exactly as you want it to appear. Example:
   `file.description=Inventario para El Rey`
8. Save and close the file.
9. Upload the localization file along with other report or folder files into the BA Repository. Instructions for how to do this appear in Upload Folders and Files.

Edit Existing Localization Information

Complete these steps if you want to edit localization information for report and folder names and descriptions that appear in the User Console.

1. Download the report or folder for which you want to edit localization information. Instructions for how to do this appear in Download Folders and Files .
2. Unzip the downloaded report or folder.
3. Use a text editor to open the localization file you want to modify. Localization files have the `.local` extension. Localization files for folders begin with the word `index`. Localization files for reports begin with the report name.
4. To change the report or folder name that appears in the console, edit the **file.title** text. Example:
   `file.title=Inventario`
5. To change the report or folder description Tool Tip that appears when you hover the mouse pointer over the report or folder name in the console, edit the value of the **file.description** variable. Example:
   `file.description=Inventario para El Rey`
6. Save and close the file.
7. Re-upload the localization file, along with the other report or folder files, into the BA Repository. Instructions for how to do this appear in Upload Folders and Files.

Manage Licenses with the CLI

Although the **Administration** page of the User Console is the quickest, easiest, and most comprehensive way to manage the BA Server, you may be in environments where it is difficult or impossible to deploy or use the console. See the alternative instructions for command line interface (CLI) license registration for step-by-step instructions.

To ensure that the BA Server and DI Server use the same location to store and retrieve Pentaho licenses, you must create a **PENTAHO_INSTALLED_LICENSE_PATH** system variable for your **pentaho** user account. If you do not set this variable, the User Console and the command line license installation script will store license data in a place that the DI Server and BA Server will not look.

It does not matter what you set this location to; however, it needs to be available to the user account(s) that run the User Console and the BA and DI Server.

Set the License Path Variable on Linux

1. Edit your **/etc/environment** file with a text editor. If you're using Solaris, you will have to set this environment variable through whatever means are available to you.
2. Add this line in a convenient place (changing the path as explained above, if necessary): **export PENTAHO_INSTALLED_LICENSE_PATH=/home/pentaho/.installedLicenses.xml**.
3. You must log out and log back into the operating system for the change to take effect.
4. Verify that the variable is properly set.

```
env | grep PENTAHO_INSTALLED_LICENSE_PATH
```

The PENTAHO_INSTALLED_LICENSE_PATH variable is now set.

Set the License Path Variable on Windows

1. In **Windows 2008 Server**, right-click on **Computer**, then select **Properties** from context menu, then click **Advanced System Settings**. The **System Properties** window will come up.
2. In the System Properties window, click the **Advanced** tab, then click **Environment Variables**.
3. In the System Variable section, click New.
4. A popup dialog will ask for a variable name and value. Type **PENTAHO_INSTALLED_LICENSE_PATH** into the name field, and the directory you intend to install licenses to plus **.installedLicenses.xml** in the value field, then click **OK**.

```
C:/pentaho/.installedLicenses.xml
```

5. In the parent window, click **Apply Changes**.
6. You must restart your computer for the change to take effect.

7. Verify that the variable is properly set.

```
echo %PENTAHO_INSTALLED_LICENSE_PATH%
```

The PENTAHO_INSTALLED_LICENSE_PATH variable is now set.
If you run Tomcat automatically as a Windows service, you must also configure it to run on the Windows user account where the **.installedLicenses.xml** file is located.

Install or Update License Keys from the Command Line

1. Download the .lic file you want to install.
2. Navigate to the `/license-installer/` directory.
3. Copy your .lic files to the `/license-installer/` directory.
4. Run the license installation script.
   a. **Linux:** Run `install_license.sh` with the install switch and the location and name of your .lic file as a parameter. You can specify multiple .lic files separated by spaces. Be sure to use backslashes to escape any spaces in the path or file name.

   ```
   ./install_license.sh install Pentaho\ BI\ Platform\
   Enterprise\ Edition.lic
   ```

   b. **Windows:** Run `install_license.bat` with the install switch and the location and name of your license file as a parameter.

   ```
   install_license.bat install "C:/Users/dvader/Downloads/
   Pentaho BA Platform Enterprise Edition.lic"
   ```

List or Remove License Keys from the Command Line

To list or remove Pentaho keys in the command line interface, follow the below instructions.

1. Navigate to the `/pentaho/server/license-installer/` directory.
2. Run the **install_license.bat** (on Windows) or **./install_license.sh** (on Linux) script with the **display** switch.

   ```
   Windows: install_license.sh display
   ```

   ```
   Linux: ./install_license.sh display
   ```

   If you have installed any Enterprise Edition keys, a list of them will appear, along with the products they cover and the duration of the license.
3. To remove a license, run the same script with the **uninstall** switch.

```
Windows: install_license.sh uninstall
```

```
Linux: ./install_license.sh uninstall
```

A list of installed licenses will appear, followed by a prompt for the license ID you would like to remove. If you press **Enter** at this prompt, it will exit without taking any action.

4. Type in the license ID number that you want to remove, then press **Enter**.

   After removing a key, if you had more than one installed, the list will regenerate and the prompt will reappear. You can choose to remove another license key, or you can press **Enter** to exit the script.

   If you would prefer not to be prompted for confirmation, or if you intend to call this program as part of a script, use the **-q** switch to suppress prompting.

Maintain the BA Server Manually

Overview

The BA Server has options that must be set manually, outside of the Administration page of the User Console. You can do any or all of these tasks depending on your needs.

Disable Home Perspective Widgets

The default console **Home** perspective contains the **Getting Started** widget, which has easy instructions and tutorials for evaluators. The directions below show you how to hide not only the **Getting Started** widget, but also other **Home** perspective widgets, as needed.

1. Shut down the BA Server, if it is currently running.
2. If you have not deployed yet, navigate to the `/pentaho-platform/user-console/source/org/pentaho/mantle/home/properties/config.properties` file.
3. If you have manually deployed and want to hide widgets at a later time, navigate to `biserver-ee/tomcat/webapps/pentaho/mantle/home/properties/config.properties` file.
4. Find the line that starts with `disabled-widgets=` and type in the ID of the widget `getting-started`. You can also disable the **Recents** and **Favorites** widgets using the same method, as shown here. Save and close the file.

```
disabled-widgets=getting-started,recents,favorites
```

5. Locate the `bi-server/tomcat/webapps/pentaho/mantle/home` and open the `index.jsp` file with any text editor.
6. Find this line and comment it out, then save and close the file.

```
<script language='JavaScript' type='text/javascript' src='http://admin.
brightcove.com/js/BrightcoveExperiences.js'></script>
```

7. Start the BA Server and log in to the User Console.

You now have a Home page that is scrubbed of the specified widget(s).

Turn Autocomplete Off for Web App Login Screen

The User Console's login settings have autocomplete turned off by default; if you need to, here is how to manually disable the autocomplete.

1. Stop the BA Server.

2. Navigate to the `/biserver-ee/tomcat/webapps/pentaho/jsp` directory and open the `PUCLogin.jsp` file with any text editor.
3. Find the following two sections of text and change the `autocomplete` entry to `off`, as shown.

```
<input id="j_username" name="j_username" type="text" placeholder=""
autocomplete="off">
```

```
<input id="j_password" name="j_password" type="password" placeholder=""
autocomplete="off">
```

4. Save and close the `PUCLogin.jsp` file.
5. Navigate to the `/data-integration-server/tomcat/webapps/pentaho-di/jsp` directory and open that `PUCLogin.jsp` file with any text editor.
6. Find the following two sections of text and change the `autocomplete` entry to `off`, as shown.

```
<input id="j_username" name="j_username" type="text" placeholder=""
autocomplete="off">
```

```
<input id="j_password" name="j_password" type="password" placeholder=""
autocomplete="off">
```

7. Save and close the `PUCLogin.jsp` file.
8. Restart the BA Server.

Autocomplete for user names and passwords is now disabled for the User Console login screen.

Remove Sample Data from the BA Server

By default, Pentaho provides a sample data source and a solution directory filled with example content. These samples are provided for evaluation and testing. Once you are ready to move from an evaluation or testing scenario to development or production, you can remove the sample content. Follow the instructions below to completely remove the Pentaho sample data and solutions.

1. Stop the BA Server.
2. Delete the **samples** zip file from the `/pentaho/server/biserver-ee/pentaho-solutions/system/default-content` directory. If you performed a manual WAR build and deployment, then the file path is `/pentaho/server/biserver-ee/pentaho-solutions/system`.
3. Edit the `/pentaho/WEB-INF/web.xml` file inside of the deployed pentaho.war. As laid down by the Pentaho graphical installer and archive packages, this path should be `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/web.xml`. If you performed a manual WAR build and deployment, then you must adjust the path to fit your configuration.
4. Remove the **hsqldb-databases** section from the `/pentaho/WEB-INF/web.xml` file:

```
<!-- [BEGIN HSQLDB DATABASES] -->
    <context-param>
        <param-name>hsqldb-databases</param-name>
```

```
        <param-value>sampledata@../../data/hsqldb/sampledata</param-value>
    </context-param>
<!-- [END HSQLDB DATABASES] -->
```

5. Remove the **hsqldb-starter** section from the `/pentaho/WEB-INF/web.xml` file:

```
<!-- [BEGIN HSQLDB STARTER] -->
<listener>
<listener-class>org.pentaho.platform.web.http.context.
HsqldbStartupListener</listener-class>
</listener>
<!-- [END HSQLDB STARTER] -->
```

6. Remove the **SystemStatusFilter**:

NOTE:

This is not part of the Pentaho samples; it provides error status messages that are only useful for development and testing purposes, and should be removed from a production system.

```
<filter>
    <filter-name>SystemStatusFilter</filter-name>
    <filter-class>com.pentaho.ui.servlet.SystemStatusFilter</filter-class>
    <init-param>
        <param-name>initFailurePage</param-name>
        <param-value>InitFailure</param-value>
        <description>This page is displayed if the PentahoSystem fails to
properly initialize.</description>
    </init-param>
</filter>
```

7. Remove the filter mapping for the **SystemStatusFilter**:

```
<filter-mapping>
    <filter-name>SystemStatusFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

8. Save and close the **web.xml** file.

9. Delete the `/pentaho/server/biserver-ee/data/` directory. This directory does not exist if you installed Pentaho with the installation wizard. It contains a sample database, control scripts for that database and the environment settings it needs to run, and SQL scripts to initialize a new repository.

10. Restart the BA Server.

11. Log on to the User Console with the administrator user name and password and go to the **Browse Files** page.

    l. A. In the **Folders** pane, expand the **Public** folder and click to highlight the folder containing the **Steel Wheels** sample data. Click **Move to Trash** in the **Folder Actions** pane and confirm the deletion.

m. B. Highlight the folder containing the **Pentaho Operations Mart** sample data. Click **Move to Trash** in the **Folder Actions** pane and confirm the deletion.

Your BA Server instance is now cleaned of samples and development/testing pieces, and is streamlined for production.

Enable JPivot in the User Console

We recommend that you use Pentaho Analyzer for analytic reporting. However, you can enable JPivot on the BA Server if you need to by following these steps.

CAUTION:

JPivot was deprecated in Pentaho 5.0 and is no longer supported by Pentaho Customer Support.

1. Shut down the BA Server if it is currently running.
2. Navigate to `/pentaho/server/biserver-ee/pentaho-solutions/system/pentaho-jpivot-plugin-legacy/plugin.xml` file and delete file.
3. Rename the file called `/pentaho/server/biserver-ee/pentaho-solutions/system/pentaho-jpivot-plugin-legacy/plugin.xml.enabled` to `/pentaho/server/biserver-ee/pentaho-solutions/system/pentaho-jpivot-plugin-legacy/plugin.xml` and save it. You can either turn off the deprecation message for JPivot, or restart the BA Server at this point.

Turn Off Deprecation Message for JPivot - Optional

1. Shut down the BA Server if it is currently running.
2. Navigate to the `/pentaho-solutions/system/pentaho-jpivot-plugin-legacy/settings.xml` and open the file.
3. Change the Boolean value as shown and save the file.
From:

```
<show-deprecation-warning>true</show-deprecation-warning>
```

To:

```
<show-deprecation-warning>false</show-deprecation-warning>
```

4. Restart the BA Server.

The JPivot plug-in is enabled in the User Console and the deprecation message is turned off.

Change the Location of the Server Log File

For Windows:

If you used the install wizard to install Pentaho Business Analytics, the `pentaho.log` is written to the `C:/WINDOWS/system32` directory. To change the location of the pentaho.log file, you must edit the `log4j.xml` available at `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/classes/`.

Modify the location as shown in the sample below, using the appropriate path to your installation of Pentaho Business Analytics.

```
<param name="File" value="C:/Program Files/pentaho/server/biserver-ee/logs/
pentaho.log"/>
<param name="Append" value="true"/>
```

For Linux:

If you are using Linux, the `log4j.xml` file is found at `/home/pentaho/pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/classes/`.

Modify the location as shown in the sample below, using the appropriate path to your installation of Pentaho Business Analytics.

```
<param name="File" value="home/pentaho/server/biserver-ee/logs/pentaho.log"/>
<param name="Append" value="true"/>
```

Change the Quartz Misfire Threshold

With Quartz, sometimes scheduled jobs, transformations, or reports might try to run several times when they are manually stopped and restarted, instead of running only once. This is typically caused by the `misfireThreshold` property in Quartz being set at too high of a number.

These steps show how to reset the `misfireThreshold` to a lower numerical value.

1. Stop the BA Server.
2. Locate the `biserver-ee/pentaho-solutions/system/quartz` directory.
3. Open the `quartz.properties` file with any text editor.
4. Find the property shown below and change the default to a smaller number, such as 5000. The default value represents the number of milliseconds.

   ```
   org.quartz.jobStore.misfireThreshold = 60000
   ```

5. Save and close the `quartz.properties` file.
6. Restart the BA Server.

Set System Max Row Limit for Interactive Reports

You can prevent too many resources from hitting your database server at once by setting a system-wide maximum row-limit for Pentaho Interactive Reports. Your users can still define their own row-limits in PIR, but they will never be able to go over the maximum number of rows that you have specified.

1. Shut down the BA Server.
2. Locate the `biserver-ee/pentaho-solutions/system/pentaho-interactive-reporting` directory.
3. Open the `settings.xml` file with any text editor.
4. Find the `<query-limit>` tag and change the default number of 1000 within the tags to the maximum number of rows desired.

```
<!-- The maximum number of rows that will be rendered in a report on PIR edit
and
view mode. A zero value means no limit. -->
<query-limit>1000</query-limit>
```

5. Save and close the `settings.xml` file.
6. Start the BA Server.

If you are migrating content from a previous version, you will need to add the `<query-limit>` tag to your `settings.xml` for PIR.

Increase the CSV File Upload Limit

You may find that you need to increase the size of the upload limit for your CSV files. These steps guide you through this process.

1. Go to `/biserver-ee/pentaho-solutions/system` and open the `pentaho.xml` file.
2. Edit the XML as needed (sizes are measured in bytes):

```
<file-upload-defaults>
      <relative-path>/system/metadata/csvfiles/</relative-path>

      <!-- max-file-limit is the maximum file size, in bytes, to allow to be
uploaded to the server -->
      <max-file-limit>10000000</max-file-limit>

      <!-- max-folder-limit is the maximum combined size of all files in the
upload folder, in bytes. -->
      <max-folder-limit>500000000</max-folder-limit>

</file-upload-defaults>
```

3. Save your changes to the file.
4. In the **User Console**, go to **Tools** > **Refresh System Settings** to ensure that the change is implemented.
5. Restart the User Console.

Change the Staging Database for CSV Files

Hibernate is the default staging database for CSV files. Follow these instructions if you want to change the staging database.

1. Go to `/pentaho-solutions/system/data-access` and open the `settings.xml` file with any text editor.
2. Edit the `settings.xml` file as needed. The default value is shown in the sample below.

```
<!-- settings for Agile Data Access -->
<data-access-staging-jndi>hibernate</data-access-staging-jndi>
```

This value can be a JNDI name or the name of a Pentaho Database Connection .

3. Save and close the file.
4. Restart the User Console.

Change the Port Numbers for the BA Server

Follow the instructions below to change the port through which the BA Server runs:

1. Stop the BA Server.
2. Navigate to the `/pentaho/server/biserver-ee/tomcat/conf/` directory.
3. Open the `server.xml` file with any text editor, and search for the value for `Define a non-SSL HTTP/1.1 Connector`. Change the port number in the connector port element below from `8080` to your preferred port number.

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
    <Connector port="8080" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" redirectPort="8443" acceptCount="100"
               connectionTimeout="20000" disableUploadTimeout="true" />
```

4. Save and close the `server.xml` file.
5. Navigate to the `/tomcat/webapps/pentaho/WEB-INF/web.xml` directory and open the `web.xml` file with any text editor.
6. Change the `fully-qualified-server-url` entry to match the new port number you specified in `server.xml`.

```
<context-param>
    <param-name>fully-qualified-server-url</param-name>
    <param-value>http://localhost:8080/pentaho/</param-value>
</context-param>
```

7. Save and close the `web.xml` file.
8. Restart the BA Server.

Change the Web Application Name or Port

The Pentaho BA Server and web application default port number is **8080**. The default web application name is **pentaho**, which is the name of the `.war` file archive, the name of the directory that your application server creates, and also part of the URL structure for all content in the User Console.

If you need to change the User Console application name to something else, or if your Web application server is running on a port other than 8080, follow these instructions for either JBoss or Tomcat.

Change the Web Application Name on Tomcat

These instructions only work on Tomcat servers that are configured to accept `context.xml` overrides built into deployed `.war` files. Some Tomcat deployments may not have this feature turned on. You can change the Tomcat configuration on your own, or consult your Tomcat documentation to learn about other methods of changing a web application context. Use the XML snippet in these instructions in whichever configuration file you end up creating or modifying.

Follow these instructions to change the web application context for a Pentaho `.war` file that you deployed to a Tomcat server.

1. After deployment is successful, delete the original `pentaho.xml` file.
2. Navigate to the `META-INF` directory and open the `context.xml` file in a text editor.
3. In the context.xml file, change the `pentaho` references in the `Context path` tag to your preferred context name:

```
<context path="/pentaho" docbase="webapps/pentaho/">
```

4. Save and close the file.
5. While still in the `WEB-INF` directory, open the `web.xml` file.
6. Edit the `pentaho.xml` file found there, and change the `fully-qualified-server-url` entry to match the context name you specified previously, and your Tomcat port number.

```
<context-param>
    <param-name>fully-qualified-server-url</param-name>
    <param-value>http://localhost:5150/example/</param-value>
</context-param>
```

Change the Web Application Name on JBoss

Follow these instructions to change the Web application context for a `Pentaho.war` file that you deployed to a JBoss server.

1. After deployment is successful, delete the original `pentaho.xml` file.
2. Open the jboss-web.xml file with any text editor.
3. Add the following text inside of the `jboss-web` section, replacing **example** with your new context name:

```
<context-root>example</context-root>
```

4. Save and close that file, then open the `web.xml` file and modify the `fully-qualified-server-url` entry to match your new context name and port number.

```
<context-param>
    <param-name>fully-qualified-server-url</param-name>
    <param-value>http://localhost:5150/example/</param-value>
</context-param>
```

5. Save and close the file.

Cluster the Application Server

A Pentaho node is made up of a Tomcat Web App server, and one BA or DI server. Multiple nodes that are joined make up a cluster. You can create a cluster using any version of Pentaho BA Suite 5.x.

These sections explain the requirements for clustering servers, how to initialize and configure the repository, configure a jackrabbit journal and quartz, and how to test the cluster.

Prerequisites for Clustering

Before you begin the process of clustering your servers, there are a few tasks that you need to do and some specific requirements that must be met in order to successfully implement a Pentaho deployment on a Tomcat or JBoss cluster.

Requirement	Description
Make sure that all of your application nodes are set up with identical configurations and BA deployments.	Your application nodes all need the same configurations and BA deployments installed already in order for clustering to work.
Establish a load balancer.	This will make sure that computing resources are spread evenly among the nodes.
Each node and the load balancer must be time-synchronized via NTP.	All machines that make up the cluster have to have the same system time. If they do not, execution times of objects will be effected.
You must run only one node per machine (or NIC).	It is possible to run multiple application servers on each node with a modified configuration, but this scenario does not offer any benefit for load balancing (performance) or hardware failover (redundancy), and therefore is not covered in this guide. Refer to your application server's clustering documentation for more information.
You must use either Tomcat 6.0 or JBoss 7.x.	You may be able to use this guide as a basic blueprint for configuring other application servers or versions of Tomcat and JBoss for a clustered environment, but Pentaho support will not be able to assist you if you run into any problems with the BA or DI Servers.
You must have permission to install software and modify service configurations.	If you do not have permissions, you must have access to someone at your company who does have the correct permission levels - typically root access.

Requirement	Description
Only the Pentaho BA Server will be deployed to the cluster.	It is possible to modify the configuration to deploy other WARs or EARs. However, for ease of testing and support, Pentaho only supports deployment of the pentaho and pentaho-style WARs to the cluster.
You must use a single repository location.	Most people use a database-based solution repository. Keep in mind that you are not clustering the database server in this procedure, only the application server. If you are using a file-based repository, you will have to create one official location for the solution repository, preferably on a network share so that the location can be the same for all nodes.
You must have sticky sessions enabled.	This will tie your session to a single node.

Initialize and Configure Repository

After you have determined that your systems meet all of the requirements listed in the checklist, you need to first initialize and then configure the repository for clustering. Finally, there are a few steps to take in order to verify your clustering setup, before you move on to setting up the jackrabbit journal.

1. Initialize your database using the steps in the appropriate article for your system. Initialize Repository has sections for PostgreSQL, MySQL, and Oracle databases.
2. After you have initialized your database, you will need to configure the data connections to the BA Repository. Specify Connections walks you through the steps for JDBC and JNDI connections for PostgreSQL, MySQL, and Oracle.
3. The next step is to configure your repository using the appropriate tasks in the Configure Repository article.
4. After you have initialized and configured your repository, you should clean up these files by following these steps.
    - Locate the `...biserver-ee/tomcat` directory and remove all files and folders from the `temp` folder.
    - Locate the `...biserver-ee/tomcat` directory and remove all files and folders from the `work` folder.
    - Locate the `...biserver-ee/pentaho-solutions/system/jackrabbit/repository` directory and remove all files and folders from the final `repository` folder.
    -

Locate the `...biserver-ee/pentaho-solutions/system/jackrabbit/repository` directory and remove all files and folders from the `workspaces` folder.

You now have a configured repository and are ready to move to the next step for clustering.

Configure Jackrabbit Journal

These directions explain how to set up the Jackrabbit journal for your cluster. Make sure that each node has a unique ID.

1. Locate the `repository.xml` file in the `.../bi-server/pentaho-solutions/system/jackrabbit` directory and open it with any text editor.
2. Scroll to the bottom of the file and replace the section that begins with `<!-- Run with a cluster journal -->` with the correct code for your type of database repository.
3. Save and close the file.

For PostgreSQL only:

```
<!--
Run with a cluster journal
-->
<Cluster id="Unique_ID ">
    <Journal class="org.apache.jackrabbit.core.journal.DatabaseJournal">
      <param name="revision" value="${rep.home}/revision.log"/>
      <param name="url" value="jdbc:postgresql://HOSTNAME:PORT/jackrabbit"/>
      <param name="driver" value="org.postgresql.Driver"/>
      <param name="user" value="jcr_user"/>
      <param name="password" value="password"/>
      <param name="databaseType" value="postgresql"/>
      <param name="janitorEnabled" value="true"/>
      <param name="janitorSleep" value="86400"/>
      <param name="janitorFirstRunHourOfDay" value="3"/>
    </Journal>
</Cluster>
```

For MySQL only:

```
<!--
Run with a cluster journal
-->
<Cluster id="Unique_ID">
    <Journal class="org.apache.jackrabbit.core.journal.DatabaseJournal">
      <param name="revision" value="${rep.home}/revision.log"/>
      <param name="url" value="jdbc:mysql://HOSTNAME:PORT/jackrabbit"/>
```

```
      <param name="driver" value="com.mysql.jdbc.Driver"/>
      <param name="user" value="jcr_user"/>
      <param name="password" value="password"/>
      <param name="schema" value="mysql"/>
      <param name="databaseType" value="mysql"/>
      <param name="janitorEnabled" value="true"/>
      <param name="janitorSleep" value="86400"/>
      <param name="janitorFirstRunHourOfDay" value="3"/>
    </Journal>
</Cluster>
```

For Oracle only:

```
<!--
Run with a cluster journal
-->

<Cluster id="Unique_ID">
    <Journal class="org.apache.jackrabbit.core.journal.OracleDatabaseJournal">
        <param name="revision" value="${rep.home}/revision.log" />
        <param name="url" value="jdbc:oracle:thin://localhost:1521/di_
jackrabbit"/>
        <param name="driver" value="oracle.jdbc.OracleDriver"/>
        <param name="user" value="jcr_user"/>
        <param name="password" value="password"/>
        <param name="schema" value="oracle"/>
     </Journal>
</Cluster>"
```

Jackrabbit journaling is now set up for your BA cluster. The Jackrabbit Wiki has additional information about journaling. Next, you need to cluster the quartz tables to avoid duplicate scheduling on each node.

Configure Quartz

There are a few edits that you will need to make in the quartz.properties file to configure Quartz to work with your cluster.

`org.quartz.scheduler.instanceId = AUTO`

1. Locate the `quartz.properties` file in the `.../bi-server/pentaho-solutions/system/quartz` directory and open it with any text editor.
2. Find the `org.quartz.scheduler.instanceId = INSTANCE_ID` line and change `INSTANCE_ID` to `AUTO`.

3.

```
org.quartz.scheduler.instanceId = AUTO
```

4. Find the `#_replace_jobstore_properties` section and change the default value of `org.quartz.jobStore.isClustered` to `true` as shown.

5.

```
#_replace_jobstore_properties

org.quartz.jobStore.misfireThreshold = 60000
org.quartz.jobStore.driverDelegateClass = org.quartz.impl.jdbcjobstore.
PostgreSQLDelegate
org.quartz.jobStore.useProperties = false
org.quartz.jobStore.dataSource = myDS
org.quartz.jobStore.tablePrefix = QRTZ5_
org.quartz.jobStore.isClustered = true
```

6.

Add this line just after the `org.quartz.jobStore.isClustered = true` line.

```
org.quartz.jobStore.clusterCheckinInterval = 20000
```

Quartz is now configured for your cluster. The Quartz Configuration Reference has additional information about clustering with Quartz.

Start and Test the Cluster

Follow the below instructions to start the cluster and verify that it is working properly.

1. Start the solution database.
2. Start the application server on each node.
3. Make sure that the load balancer is able to ping each node.
4. Repeat for each node that you have set up.
5. Test the cluster by accessing the BA Server through the load balancer's IP address, hostname, or domain name. Begin whatever test procedure you have designed for this scenario.

Monitor System Performance Using Pentaho Operations Mart

The Pentaho Operations Mart makes it easy for you to monitor system performance for information, such as the amount of memory being used during report creation or how long it takes to run certain reports. We automatically set up a way for you to do this using the Operations Mart and Analyzer, Interactive Reports, Report Designer, and Dashboards. The Operation Mart aggregates data from the BA and DI Server log files into pre-built audit reports that provide information you might need. If these reports do not meet your needs, you can use Pentaho Data Integration Operations Mart to change them.

Choose a Pre-Built Operations Mart Report

We provide pre-built Interactive Reports and a data mart called the Pentaho Operations Mart. The Operations Mart contains all the data from system log files. This table shows the reports that we have pre-built for you. Choose the report that fits your needs and follow the instructions in View and Edit Operations Mart Reports.

Table 1. Pre-Built Audit Mart Reports

Information Shown	Report Name
Amount of time it takes a report to run	Content Duration
File names of all content failures within a defined length of time	Content Failures
Compare login metrics for days within a specified month	Content Request Day of Month
Compare login metrics for a days in a week within a specified year	Day of Week Request and Login Metrics
List of content sorted by type, used within a defined length of time	Content Type Usage
List of content usage within a defined length of time	Content Usage
Compare login metrics by hour for a specified day	Hours in Day Request and Login Metrics
Length of time for logins and the number of logins per user for specified length of time	Session Duration Counts

View and Edit Operations Mart Reports

1. Identify the report that you want to open from Choose a Pre-Built Operations Mart Report.
2. Double-click **Browse Files**.
3. Click on **public** > **Pentaho Operations Mart** > **BA Audit Reports**.

4. Select the appropriate file and click **Open**. Your monitoring and logging data appears in the pre-built report.
5. If you need to edit this report, you can do so from within the opening tool.

See Build Web-Based Data Analysis, Reports, and Dashboards for more information about editing a report in Analyzer or Interactive Reporting.

See Design Print-Quality Reports for more information about editing a report using Report Designer.

Create Operations Mart Reports

If the reports we provide are not quite right for you, you can create your own reports.

1. From within the User Console, select **Create New** and then choose the type or report you want to create.
2. Select the data source that you need.

Information Shown	Data Source
Detail information related to the execution of the.xactions. that run the reports on the BA Server	pentaho_operations_mart: BA Operations Mart - Component
Information related to the execution of content , such as which tool or which user ran the content on the BA Server	pentaho_operations_mart: BA Operations Mart - Content
Information for the BA Server related to a user, such as number of sessions, how long, and what time	pentaho_operations_mart: BA Operations Mart - User Session
Information about individual job entry executions on the DI Server	pentaho_operations_mart: PDI Operations Mart - Job Entry
Detailed performance information for the DI Server	pentaho_operations_mart: PDI Operations Mart - Performance
Detailed information about individual step executions on the DI Server	pentaho_operations_mart: PDI Operations Mart - Step
Information related to transformations and jobs run on the DI Server	pentaho_operations_mart: PDI_Operations_Mart

3. Create a new report as described in the appropriate user section.

See Build Web-Based Data Analysis, Reports, and Dashboards for more information about building a report in Analyzer or Interactive Reporting.

See Design Print-Quality Reports to build a report using Report Designer.

Update the Operations Mart

You may need to update the report date and time, or the data that is populated into the report itself.

1. From within the User Console, select **Browse Files** > **public** > **Pentaho Operations Mart** > **Update Audit Mart**.

2. Double-click either **Update Operations Mart Date & Time** or **Update BA Operations Mart Data**.
3. View a report to confirm that the updates applied.

Performance Monitoring and Logging

To help you maintain the health of your Pentaho system, we provide features that help you diagnose processing errors and monitor the BA Server performance.

Disable Server and Session-Related Timeouts to Debug

Follow the instructions below to disable server and session timeouts associated with the User Console.
Important: These instructions are applicable when you are in a *test* environment. Once you go live, it is recommended that you set your timeouts to five or ten minutes so that sensitive BA Server-related data can be protected. The time must be expressed in minutes.

1. Open the file, **server.xml**, located under ...`\biserver-ee\tomcat\conf`.
2. Find the **connectionTimeout="20000"** parameter and change its value to zero ("**0**"). If this value is set to a negative number it will never timeout.
3. Open the file **web.xml**, located under `... \bi-server_ee\tomcat\webapps\penaho\WEB-INF\web.xml`.
4. Find the **session-timeout** parameter and change its value to negative one ("**-1**").
5. Save the file and refresh the user console.

Log Rotation

This procedure assumes that you do not have or do not want to use an operating system-level log rotation service. If you are using such a service on your Pentaho server, connect to the BA Server and Data Integration Server and use that instead of implementing this solution.

The Business Analysis and Data Integration servers use the Apache log4j Java logging framework to store server feedback. The default settings in the log4j.xml configuration file may be too verbose and grow too large for some production environments. Follow these instructions to modify the settings so that Pentaho server log files are rotated and compressed.

1. Stop all relevant Pentaho servers.
2. Download a .zip archive of the Apache Extras Companion for log4j package: Apache Logging Services.
3. Unpack the **apache-log4j-extras** JAR file from the zip archive, and copy it to the following locations:
   - **Business Analytics Server:** `/tomcat/webapps/pentaho/WEB-INF/lib/`
   - **Data Integration Server:** `/tomcat/webapps/pentaho-di/WEB-INF/lib/`
6. Edit the **log4j.xml** settings file for each server that you are configuring. The files are in the following locations:
   - **BA Server:** `/tomcat/webapps/pentaho/WEB-INF/classes/`
   - **DI Server:** `/tomcat/webapps/pentaho-di/WEB-INF/classes/`
9. Remove all **PENTAHOCONSOLE** appenders from the configuration.

10. Modify the **PENTAHOFILE** appenders to match the log rotation conditions that you prefer. You may need to consult the log4j documentation to learn more about configuration options. Two examples that many Pentaho customers find useful are listed:

    Daily (date-based) log rotation with compression:

```
<appender name="PENTAHOFILE" class="org.apache.log4j.rolling.
RollingFileAppender">
    <!-- The active file to log to; this example is for BA/DI Server.-->
    <param name="File" value="../logs/pentaho.log" />
    <param name="Append" value="false" />
    <rollingPolicy class="org.apache.log4j.rolling.TimeBasedRollingPolicy">
        <!-- See javadoc for TimeBasedRollingPolicy -->
        <param name="FileNamePattern" value="../logs/pentaho.%d.log.gz" />
    </rollingPolicy>
    <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
    </layout>
</appender>
```

    Size-based log rotation with compression:

```
<appender name="PENTAHOFILE" class="org.apache.log4j.rolling.
RollingFileAppender">
    <!-- The active file to log to; this example is for BA/DI Server.-->
    <param name="File" value="../logs/pentaho.log" />
    <param name="Append" value="false" />
    <rollingPolicy class="org.apache.log4j.rolling.
FixedWindowRollingPolicy">
        <param name="FileNamePattern" value="../logs/pentaho.%i.log.gz" />
        <param name="maxIndex" value="10" />
        <param name="minIndex" value="1" />
    </rollingPolicy>
    <triggeringPolicy class="org.apache.log4j.rolling.
SizeBasedTriggeringPolicy">
        <!-- size in bytes -->
        <param name="MaxFileSize" value="10000000" />
    </triggeringPolicy>
    <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern" value="%d %-5p [%c] %m%n" />
    </layout>
</appender>
```

11. Save and close the file, then start all affected servers to test the configuration.

You have an independent log rotation system in place for all modified Pentaho servers.

Define Result Row Limit and Timeout

When a query in the User Console returns an unusually large number of rows, this may impact server performance. To limit the number of rows returned by a query and to set up a timeout, you must create two custom properties, **max_rows** and **timeout**, in the Metadata Editor.

The values you define for the row number limit (max-rows) and timeout properties are passed to the JDBC driver.

To define max rows and timeout...

1. In the Metadata Editor, expand the **Business Model** node and select **Orders**.
2. Right-click Orders and choose **Edit**. The **Business Model Properties** page displays a list of properties that were previously defined.
3. In the **Business Model Properties** page, click the **Add** icon. The **Add New Property page** dialog box appears.
4. Enable **Add a custom property**.
5. In the **ID** text box, type **max_rows**. **IMPORTANT**: The ID is case-sensitive and must be typed exactly as shown.
6. Click the down-arrow in the **Type** field and choose, **Numeric**. The **Business Model Properties** page appears. The **max_rows** property is listed under **Custom** in the navigation tree.
7. In the right pane, under **Custom**, enter a value for your **max_rows** property. For example, if you enter "3000" as your value, the number of rows allowed to display in a query result is constrained to 3,000.
8. Repeat steps 3 through 6 to for the **timeout** custom property.
9. In the right pane, under **Custom**, enter a value for your **timeout** property. The timeout property requires a numeric value defined in number of seconds. For example, if you enter, 3600, the limit for query results is one minute.
10. Click **OK** in the Business Model Properties page to save your newly created properties.