HITACHI
Inspire the Next

How to Install and Use the Pentaho Support Utility

HITACHI
Inspire the Next

Change log (if you want to use it):

Date	Version	Author	Changes

Contents

This page intentionally left blank.

Overview

The Pentaho Support Utility gathers environment and Pentaho configuration information into a zip file bundle. This bundle can then be attached to a support ticket, which will provide support with most of the required background information to be able to identify and resolve your issue.

This improves the ticket resolution time by reducing the number of back and forth requests asking for additional information/configuration files while also reducing your effort in collecting and providing this information.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version(s)
Pentaho	7.x, 8.x

The Components Reference in Pentaho Documentation has a complete list of supported software and hardware.

The Pentaho Support Utility consists of three distinct tools:

- **Pentaho Server Plugin**: A plugin and UI for the Pentaho Server to gather information from the currently running server.
- **Pentaho Data Integration Plugin**: A plugin for Pentaho Data Integration (PDI) to gather information from the currently running Spoon environment.
- **Command Line Utility**: A command line tool for gathering information from the Pentaho Server or Pentaho Data Integration.

The Pentaho Server and Pentaho Data Integration plugins are the preferred mechanisms for running the Pentaho Support Utility. The command line utility should be used only as a last resort but is provided for situations where Pentaho fails to start, or policies prohibit the installation of the plugins.

If any of the reports fails to run, the rest of the report will still run. Even if some of the collectors fail, the report will still be valuable to the support team. Review the logs for the utility or look inside the output zip file for `.failed` files to identify if any collectors failed to run.

Some files and information collected by the utility may contain passwords. The utility makes a best attempt to remove all passwords during collection; however, Pentaho cannot guarantee that every password will be removed. This is especially true for the `kettle.properties` file.

Table 1: Pentaho Support Utility Data Collection

Item	Description	Pentaho Server Plugin?	PDI Plugin?	Command Line Utility?
Java environment information	Includes the number of processors, used and maximum assigned memory, system memory usage, number of garbage collections and duration, system variables including loaded `kettle.properties`, and environment settings	Yes	Yes	Partial
MD5 Checksum	Checksum on the files within the Pentaho installation directory, identifying which files have been changed from the base installation	Yes	Yes	Yes
Configuration files	`.xml` and `.properties` files from the Pentaho installation directory and subdirectories	Yes	Yes	Yes
Karaf files	Karaf-specific `.cfg` files from the Pentaho installation directory and subdirectories	Yes	Yes	Yes
Plugins	List of plugins installed in the environment	Yes	Yes	Yes
Scripts	All `.sh` and `.bat` scripts from the Pentaho installation directory and subdirectories	Yes	Yes	Yes
Repository configuration	Repository connection information from the Pentaho Server installation directory	Yes	No	Server
File system	List of files, user and group ownership, and permissions for the files in the Pentaho installation directory	Yes	Yes	Yes
Logs	Log files from Pentaho	Yes	Yes	Yes
Licenses	Collects the license information using the license installer script. If the utility cannot find the license installer script, it will collect the `.installedLicenses.xml` file.	Yes	Yes	Yes
Processes	List of processes currently active on the system for the user who is running the utility	Yes	Yes	Yes
Data sources	List of JDBC, Mondrian schemas, metadata schemas, and Wizard data sources from the Pentaho Server. Also collects JDBC connection information for each JDBC connection. Does not collect the actual Mondrian or metadata schemas, just the list of them.	Yes	No	Server

Item	Description	Pentaho Server Plugin?	PDI Plugin?	Command Line Utility?
Repository tree	List of files and their associated permissions from the Pentaho Repository. Does not collect the files themselves, just the list.	Yes	No	Server
Schedules	List of scheduled tasks from the Pentaho Server	Yes	No	Server
`.kettle`	Contents of `.kettle` folder including `kettle.properties` and `shared.xml`	Yes	Yes	Yes
`.pentaho`	Contents of the .pentaho folder including the Pentaho metastore and simple-jndi configuration	Yes	Yes	Yes
Session properties	Current session properties from the user's session	Yes	No	No
Report engine	Loaded report engine services	Yes	No	No
Ulimit	Soft ulimits for the user who started Pentaho	On Linux	On Linux	On Linux
Thread dump	Dumps current Java threads for analysis	Yes	Yes	No

Install the Pentaho Support Utility

To install the Pentaho Support Utility, you will need to install and set up the components of the utility you want to use.

These include the Pentaho Server Plugin, the Pentaho Data Integration Plugin, and the Command Line Utility. If you are not sure which parts you need, refer to the Data Collection table.

To install the Pentaho Server Plugin:

1. Download the `pentaho-support-utility-server-plugin.zip` file.
2. Unzip this file to `<pentaho-server>/pentaho-solutions/system`.
3. If needed, configure the password removal regular expression.
4. Restart the Pentaho Server.

To install the PDI Plugin:

1. Download the `pentaho-support-utility-pdi-plugin.zip` file.
2. Unzip this file to `<data-integration>/plugins`.
3. If needed, configure the password removal regular expression.
4. Restart the PDI client.

To install the Command Line Utility:

1. Download the `pentaho-support-utility-command-line.zip` file.
2. Unzip this file to a location of your choice.
3. If needed, configure the password removal regular expression.

Run the Pentaho Support Utility

Putting the Pentaho Support Utility to use involves a few steps depending on what way you plan to use the utility. You can find more information on the following topics in these sections:

Using the Pentaho Server Plugin

The Pentaho Support Utility on the Pentaho Server is only available to administrators. Non-administrator users may not access or use the plugin.

Note that if your browser disables popups, you will need to enable popups before running the report. If the report does not download, it is because your browser does not allow popups.

To use the Pentaho Server plugin:

1. Log into the Pentaho User Console (PUC) with administrator credentials.
2. Go to **Tools → Support Utility** to open the plugin:

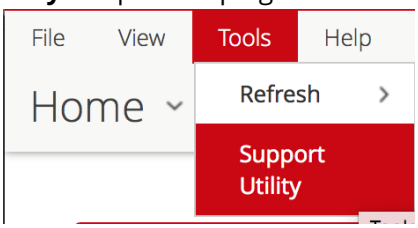

Figure 1: Opening the Support Utility

3. Enter a **Report name**. This will be the name of the downloaded file output from the utility.
4. Provide an **Environment name** (such as Dev, QA, UAT, or Prod).
5. Select **Run all reports** or the individual reports you want to collect.

***Optional**: provide a maximum number of log files to collect and/or the first date to include in the logs.*

6. Click **Generate report** at the bottom and wait while the report generates.

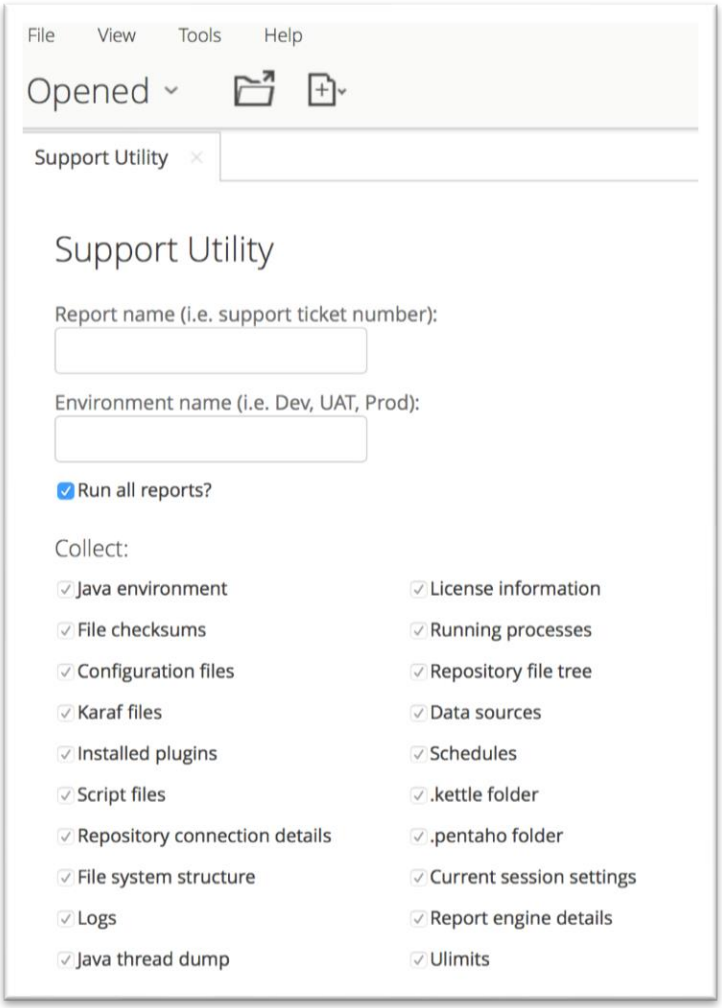

Figure 2: Configure Support Utility

Using the PDI Plugin

The Pentaho Data Integration plugin is available from the PDI client for any user, so a user can get information that Pentaho Support would need to troubleshoot.

To use the Pentaho Data Integration Plugin:

1. Open the PDI client.
2. Go to **Help → Support Utility** to open the plugin:

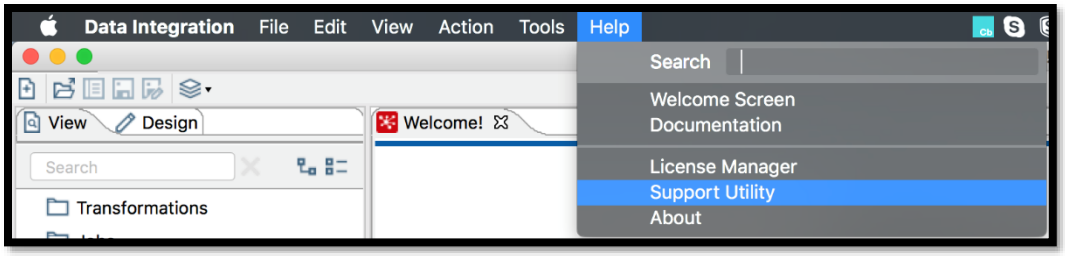

Figure 3: Open Support Utility

3. Provide a location to write the output. It will create a zip file, so the location must be a filename and not a directory:

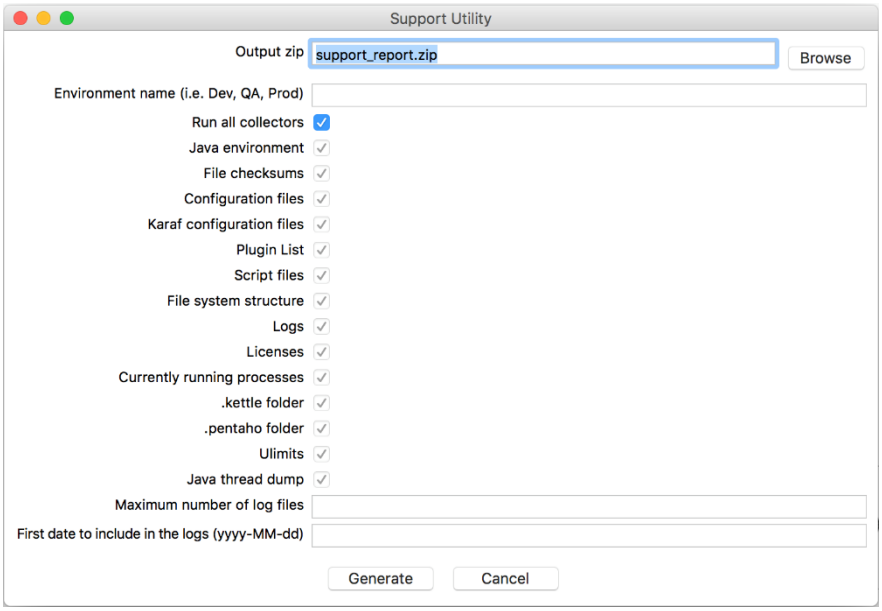

Figure 4: Output Zip

4. Provide an **environment name** (such as Dev, QA, UAT, or Steve's Laptop).
5. Choose to run all collectors or the individual reports you want to collect.

***Optional**: provide the maximum number of log files to collect, and/or the first date of logs to include in the report in `yyyy-MM-dd` format.*

6. Click **Generate**.
7. When the success dialog appears, the report has finished generating:

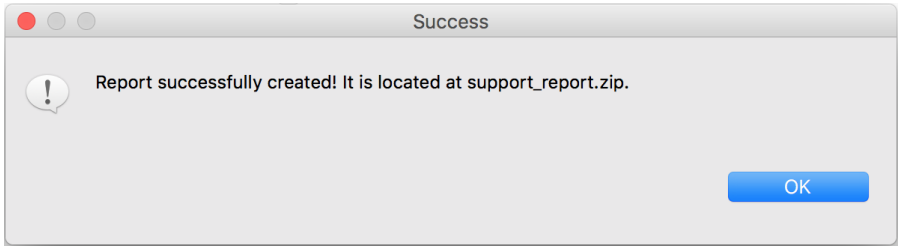

Figure 5: Report Creation Success

Using the Command Line Utility

Use the Command Line Utility when the PDI client or the Pentaho Server will not start, or when you are unable to use those versions of the Support Utility.

Run this utility as the same user who starts Pentaho. This will allow the utility to collect the most accurate information, including file permissions and environment variables.

To use the command line utility, enter this in the command line:

```
./support-utility.[sh/bat] [pdi/server] {options}
```

For Windows, run `support-utility.bat`, and for Linux/MacOS, run `support-utility.sh`. Whether you put `pdi` or `server` after the filename indicates which support utility you want to run: PDI or Pentaho Server.

When run from the command line, the Java environment collection is not able to collect memory or garbage collection information.

Options for the Command Line Utility for PDI

There are many options you can use in the command to run the utility for PDI:

Table 2: Command Line Utility PDI Options

Short Option	Long Option	Required	Description
`-i`	`--interactive`	No	Runs the support utility in interactive mode. It will prompt for all of the required options. No other options are required when running in interactive mode.
`-o <FILENAME>`	`--output <FILENAME>`	Yes	The output filename to store the report
`-pdi <PDI_HOME>`	`--pdi <PDI_HOME>`	Yes	The root directory where Pentaho Data Integration is installed.
`-env <ENV_NAME>`	`--environment <ENV_NAME>`	No	A user-friendly name to describe the environment the utility is being run in. For example: **Dev**, **UAT**, or **Prod**.
`-all`	`--all`	No	Collect all of the possible information that the utility can collect
`-c`	`--checksum`	No	Collect checksums on all files inside the PDI installation directory
`-f`	`--config-files`	No	Collect the `.xml` and `.properties` configuration files from the Pentaho installation directory and subdirectories that have been modified from the base installation

Short Option	Long Option	Required	Description
-fs	--file-system	No	Collect the details about every file in the PDI installation directory including file names, paths, user and group ownership, and permissions
-je	--java-env	No	Collect the Java environment information including Java version, memory settings, and environment variables
-k	--karaf	No	Collect the Karaf .cfg files from PDI
-kettle	--kettle	No	Collect the contents of the .kettle folder
-l	--logs	No	Collect the PDI log files
-lic	--licenses	No	Collect the installed license information
-p	--plugins	No	Collect the list of installed plugins
-pentaho	--pentaho	No	Collect the contents of the .pentaho folder
-proc	--processes	No	Collect the list of processes running on the system by the user who ran the utility
-sc	--scripts	No	Collect the script files used to start Pentaho
-u	--ulimit	No	If running on Linux, collect the soft ulimits for the user who ran the utility
-cf <FILE>	--checksum-file <FILE>	No	By default, the utility will automatically retrieve the list of expected checksums. If this option is provided, the utility will use the provided properties file as the comparison for checksums
-maxlogs <INT>	--max-logs <INT>	No	The maximum number of log files to collect
-minlogdate <yyyy-MM-dd>	--min-log-date <yyyy-MM-dd>	No	The first date to include in the logs collected. Provided in yyyy-MM-dd format
-debug		No	Run the utility with debug logging.

Common Command Line Examples for PDI

Here are a few examples of common command line scripts for PDI:

- Run all reports

```
./support-utility.sh pdi -o pentaho_support.zip -pdi /opt/pentaho/data-integration -env PROD -all
```

- Run in interactive mode

```
./support-utility.sh pdi -i
```

- Run log report

```
./support-utility.sh pdi -o pentaho_support.zip -pdi /opt/pentaho/data-integration -env PROD -l -maxlogs 10 -minlogdate 2018-11-01
```

Options for the Command Line Utility for Pentaho Server

There are also many options you can use in the command to run the utility for the Pentaho Server.

Note that in environments configured for Single Sign On (SSO), the `-all`, `--repo-tree`, `--data-sources`, and `-schedules` options will not work from the command line.

Table 3: Command Line Utility Pentaho Server Options

Short Option	Long Option	Required	Description
`-i`	`--interactive`	No	Runs the support utility in interactive mode. It will prompt for all of the required options. No other options are required when running in interactive mode.
`-o <FILENAME>`	`--output <FILENAME>`	Yes	The output filename to store the report
`-s <SERVER_HOME>`	`--pentaho-server <SERVER_HOME>`	Yes	The root directory where the Pentaho Server is installed.
`-env <ENV_NAME>`	`--environment <ENV_NAME>`	No	A user-friendly name to describe the environment the utility is being run in. For example: **Dev**, **UAT**, or **Prod**.
`-all`	`--all`	No	Collect all the possible information that the utility can collect
`-c`	`--checksum`	No	Collect checksums on all files inside the server installation directory
`-ds`	`--data-sources`	No	Collect the list of data sources configured on the Pentaho Server
`-f`	`--config-files`	No	Collect the .xml and .properties configuration files from the PDI installation directory that have been modified from the base installation
`-fs`	`--file-system`	No	Collect the details about every file in the server installation directory including file names, paths, user and group ownership, and permissions
`-je`	`--java-env`	No	Collect the Java environment information including Java version, memory settings, and environment variables

Short Option	Long Option	Required	Description
-k	--karaf	No	Collect the Karaf .cfg files from server
-kettle	--kettle	No	Collect the contents of the .kettle folder
-l	--logs	No	Collect the server log files
-lic	--licenses	No	Collect the installed license information
-p	--plugins	No	Collect the list of installed plugins
-pentaho	--pentaho	No	Collect the contents of the .pentaho folder
-proc	--processes	No	Collect the list of processes running on the system by the user who ran the utility
-r	--repository	No	Collect the Pentaho repository configuration files
-rt	--repo-tree	No	Collect the list of files stored in the Pentaho repository
-sc	--scripts	No	Collect the script files used to start Pentaho
-sch	--schedules	No	Collect the list of scheduled tasks from the Pentaho server
-u	--ulimit	No	If running on Linux, collect the soft ulimits for the user who ran the utility
-cf <FILE>	--checksum-file <FILE>	No	By default, the utility will automatically retrieve the list of expected checksums. If this option is provided, the utility will use the provided properties file as the comparison for checksums.
-maxlogs <INT>	--max-logs <INT>	No	The maximum number of log files to collect, starting with the most recent and continuing to older files until this limit is reached.
-minlogdate <yyyy-MM-dd>	--min-log-date <yyyy-MM-dd>	No	The first date to include in the logs collected. Provided in yyyy-MM-dd format.
-serverurl <URL>	--server-url <URL>	If -rt, -sch, -ds, or -all	The full URL of the Pentaho Server. For example: http://localhost:8080/pentaho.
-us <USERNAME>	--username <USERNAME>	If -rt, -sch, -ds, or -all	The username to authenticate to the Pentaho server.
-w <PASSWORD>	--password <PASSWORD>	If -rt, -sch, -ds, or -all	The password to authenticate to the Pentaho server.
-W	--prompt-password	No	Rather than providing the password on the command line, prompt for the user's password
-debug		No	Run the utility with debug logging.

Common Pentaho Server Examples

Common command line examples for PDI include:

- Run all reports

```
./support-utility.sh server -o pentaho_support.zip -s /opt/pentaho/pentaho-server -env PROD -all -serverurl http://localhost:8080/pentaho -us admin -w password
```

- Run in interactive mode

```
./support-utility.sh server -i
```

- Run log report

```
./support-utility.sh server -o pentaho_support.zip -s /opt/pentaho/pentaho-server -env PROD -l -maxlogs 10 -minlogdate 2018-11-01
```

Checksums: Finding the Expected Checksum File

The checksum process attempts to find the expected checksums to compare against, which identifies those files that have been changed from the standard Pentaho installation. The utility will look in several locations to find the checksum file. The order of precedence to find the checksum file is:

1. Checksum file explicitly provided as a report option. Only available from the command line.
2. Looks for a Java or kettle property: `SUPPORT_EXPECTED_CHECKSUM_FILE`.
3. Looks for a file named `[PDI/SERVER].<pentaho-version>.checksum.properties`. For example:

```
PDI.8.0.0.0.28.checksum.properties
```

or

```
SERVER.8.0.0.0.28.checksum.properties
```

   - **Command line**: Looks in the command line utility's base directory.
   - **Pentaho Server plugin**: Looks in `pentaho-solutions/system/pentaho-support-utility-server-plugin`.
   - **PDI plugin**: Looks in `plugins/pentaho-support-utility- pdi-plugin`.
4. Tries to download the expected checksum file from the Pentaho Checksum Master location. The download will not work if the Pentaho Server or machine is behind a web proxy.

Retrieving the checksum file from the web can be disabled by setting a Java property or `kettle.property SUPPORT_ENABLE_CHECKSUM_DOWNLOAD=N`. The `kettle.property` will not work when running from the command line.

5. Defaults to no expected checksums available. Will still collect the checksums but will not compare against expected checksums.

Password Replacement in Collected Files

The Pentaho support utility searches every file collected to identify and remove passwords using regular expressions. The default regular expression to identify and remove passwords is:

```
(?i).*<[^>]*pass.*?>([^$<][^{<][^<]+)<.*|.*?pass[^=]*="?([^$"][^{"][^"]+)"?
|.*name\\s*=\\s*"password"\\s+value\\s*=\\s*"([^"]+)".*
```

This default regular expression handles the following cases:

- The string `pass` followed by an equal sign, some text, and the end of the line. It removes everything after the equal sign.

  For example: password_for_my_server=password.

- An attribute within an XML tag that includes the string `pass` in the attribute name.

  For example: <properties.password="password">.

- Attributes within an XML tag in the `name="*pass*" value="password"` pattern.

  For example: <name="password" value="mypassword">.

This regular expression is configurable and can be extended for your own use case. To configure the password removal regular expression:

1. Create a `support.properties` file in the command line utility or plugin's root directory.
2. Add a property into this file called `SUPPORT_PASSWORD_REGEX` with the regular expression to use to remove passwords.
    a. Make sure the regular expression is properly escaped for a Java properties file. For example, `\` must be escaped `\\`.
    b. The capture groups within the regular expression are what will be removed from the files.
    c. The regular expression must match the entire line of text from the file.
    d. This regular expression replaces the default regular expression.

Related Information

Here are some links to information that you may find helpful while using this best practices document:

- Download the Pentaho Support Utility
- Pentaho Checksum Master location
- Log into the Pentaho User Console (PUC)
- Components Reference