

How Do I Install Java Report Engine?

This article gives detailed information about installing the Java Report Engine.

Requirements

- [Java 1.7](#) or later (beginning with version 12.5)
- If running Java 1.5 or earlier (with an older version of the Java Report Engine that can support it), and you have images in your templates, it is recommended to install the [Java Advanced Imaging \(JAI\) I/O library](#). This can substantially increase performance.
- A Java Report Engine license key ([request a trial license key](#) if you don't have one)

Install the Files

Go to [Windward Studios Downloads](#) and download the latest version of Java Report Engine. For Windows, click on the link [Java Engine Installer – \(EXE – Windows only\)](#). For other platforms, click on the link [Java Engine – \(ZIP file – any O/S\)](#).

Once the download is complete, double-click on the installer, or unzip the zip file. For Windows, follow the steps of the Installation Wizard. The Wizard installs all files, and adds WindwardReports.jar and third-party libraries to the CLASSPATH.

For other platforms, unzip the file SetupWindwardReports.zip in your system. Then add WindwardReports.jar and all files in the jars subdirectory (where you installed the Engine) to your CLASSPATH. For more details see [Running the Java Engine in Linux](#).

You may enter your license key during the installation or afterward.

Set CLASSPATH

The Installation Wizard added the Engine jar files to your CLASSPATH when it was executed, but if some of those jar files were already in your CLASSPATH, then you have duplicates. Even if the duplicates are identical files, Java still views them as different classes with the same name, and they can cause obscure problems later. We strongly recommend removing any duplicate entries from your CLASSPATH.

If for some reason you must manually add the Java Report Engine jar files to the CLASSPATH, we suggest adding to your environment variables this entry:

```
WR_HOME=C:\Program Files (x86)\Windward Studios\Windward Java Engine\
```

This allows you to move the jar files later and only change one environment variable. Once this is set, add the following to your CLASSPATH (all jar files, including WindwardReports.jar, in the jars\ directory):

```
%WR_HOME%jars\*;%WR_HOME%jars\olingo-odata2-dist-lib-2.0.5-lib\*;%WR_HOME%jars\twelve-monkeys-image-io\*
```

Properties File

! Java Report Engine must be able to find the file WindwardReports.properties to run.

Default Properties File Location

Java Report Engine looks for the properties file WindwardReports.properties in the default location. (Note that WindwardReportsServer.jar must use the properties file.)

The properties file can be either a file in a directory or a file in the WindwardReports.jar file. Java Report Engine looks for the file in the following locations in the following order:

1. The thread's class loader
2. The class loader that loaded Java Report Engine
3. The system class loader
4. A file on the hard drive

The default location can be overridden with the following two mechanisms:

- Add to the Java command line:

```
-DWindwardReports.properties.filename=MyWindwardReports.properties
```

- To set the value programmatically, add the following to your code (before calling Java Report Engine):

```
System.setProperty("WindwardReports.properties.filename", "MyWindwardReports.properties");
```

Example #1: -DWindwardReports.properties.filename=wr.prop searches in the following order:

1. Looks for the file wr.prop in the root of the jar file that Java Report Engine was loaded from (typically WindwardReports.jar).
2. Looks in the JVM's default directory for the file wr.prop.

Example #2: -DWindwardReports.properties.filename=net/windward/wr.prop searches in the following order:

1. Looks for the file `wr.prop` in the `net/windward` directory of the jar file that Java Report Engine was loaded from (typically `WindwardReports.jar`).
2. Looks in the subdirectory (of the JVM's default directory) for the file `net/windward/wr.prop`.

Other Properties File Locations

Depending on where you place the call to Java Reports Engine, your Java Reports Engine programmer may need to implement code indicating where the `WindwardReports.properties` file is located. For example, if Java Report Engine is called from within an application server such as Tomcat, WebSphere or WebLogic, then you must tell Java Report Engine where the `WindwardReports.properties` file resides.

Confer with your programmer and decide on the placement that works best for your system, and then implement the necessary code, as described below.

We recommend you place the properties file in one of the application server's directories *and not hard code it as `C:\WindwardReports.properties`*.

If you are running Java Report Engine on a single J2EE server, then it's best practice to place the properties file in the `WEB-INF` directory, and add this code to your program:

```
String propFile = actionServlet.getServletContext().getRealPath("/WEB-INF/
WindwardReports.properties");
System.setProperty("WindwardReports.properties.filename", propFile);
ProcessReport.init();
```

If you are placing Java Report Engine in your J2EE library directory, then it's best practice to place `WindwardReports.properties` in the same directory. Add this code (your J2EE server may have a slightly different directory structure):

```
String propFile = actionServlet.getServletContext().getRealPath../../../../lib/
WindwardReports.properties");
System.setProperty("WindwardReports.properties.filename", propFile);
ProcessReport.init();
```

Note that in the code samples above, the property is set before `ProcessReport.init()` is called.

Missing Properties File Exception

If Java Report Engine can't find the properties file, it throws an exception that lists the full path and filename of the properties file it looked for. The exception is:

```
Exception in thread "main" net.windward.util.LicenseException: The properties file was
not found java.io.FileNotFoundException: WindwardReports.properties(The system cannot
find the file specified) C:\test\WindwardReports.properties at net.windward.xmlreport.
```

```
ProcessReport._init(ProcessReport.java:322) at net.windward.xmlreport.ProcessReport.  
init(ProcessReport.java:170) at net.windward.xmlreport.RunReport.main(RunReport.java:47)
```

In this case it was looking for C:\test\WindwardReports.properties.

License Key

A license key is required to run Java Report Engine. This key enforces Java Report Engine licensing restrictions such as:

- How many reports a day your system may generate.
- If your system is a development server (only one report at a time).

The license is an uuencoded string and is applied by editing the following line in your WindwardReports.properties file (this is not a valid key):

```
license=vt0OucWy76Z9g9hH9xb+3cz2rSAJQE/HnM84UYnU30hJJGaHnh9/VmeMFxyf5lLL\  
2t7+d4yqZTxZ60Xn+tZ4WSjEX1Rqn1Gt2BQt122a7zS6xUEL99bO3czy58D94u3J\  
nh8vVySk16KfC3rlth7+scz2rSANSclH9ZTq2en+zW7NKfzl
```

This can be entered as a single line, or it can be entered as shown here with a "\" (backslash) at the end of each line, except the last line.

If your key is not a legitimate key, Java Report Engine prints the key it loaded. Check the problem license key to ensure it wasn't truncated and there are no spaces in it.

Log4j

Java Report Engine uses log4j for its logging. Log4j must be configured both for the cases where the embedding application using Java Report Engine also uses log4j, and for the case where Java Engine is the only component using log4j. (Note that WindwardReportsServer.jar must use the log4j configuration file.)

Java Report Engine can't just turn off logging because that would cause problems for embedding application components using log4j. There is no way for Java Report Engine to know if log4j is being used elsewhere.

Your three options are:

1. If you use log4j and you initialize it before calling Java Report Engine, then delete the `logging.filename=...` entry in WindwardReports.properties. In this case Java Report Engine assumes log4j is already initialized.
2. If you use log4j but do not initialize it before first calling Java Report Engine, then set the `logging.filename=...` entry in Windward Reports.properties to point to your log4j.properties file. This entry is first tried as a URL and if that fails, then as a filename. (Note that a filename is not a legitimate URL.)

For more information on log4j configuration, please refer to the [log4j documentation](#). The filename you set for log4j is passed to log4j using the call (`logFile` is the string from the property `logging.filename`):

```
PropertyConfigurator.configure( new URL( logFile ) );
```

And if that fails, then:

```
PropertyConfigurator.configure( logFile );
```

3. To turn logging off (this is not recommended), set the entry to:

```
logging.filename=off
```

If you do not use log4j elsewhere, then set `logging.filename=log4j.properties` (fully specify the file location) and create a log4j.properties file.

For standard logging, we suggest these properties:

```
log4j.rootLogger=INFO, file
log4j.appender.file=org.apache.log4j.RollingFileAppender
# Write log messages to this file
log4j.appender.file.File=C:/temp/JavaEngine.log
# when file reaches this size, will start a new file
log4j.appender.file.MaxFileSize=10MB
#when starting a new file, previous file will be JavaEngine.log.1, up to JavaEngine.log.
10
log4j.appender.file.MaxBackupIndex=10
log4j.appender.file.layout=org.apache.log4j.PatternLayout
# This is the information logged
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%x -
%m%n
```


For debug logging (when requested by Support), use these properties:

```
log4j.rootLogger=DEBUG, file
log4j.appender.file=org.apache.log4j.FileAppender
# Write log messages to this file
log4j.appender.file.File=C:/temp/JavaEngine.log
log4j.appender.file.layout=org.apache.log4j.PatternLayout
# This is the information logged
log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%x -
%m%n
```

Log4j Settings to Log All Engine Components

Java Report Engine makes use of several 3rd-party libraries. To turn on logging for all components you must set log4j to enable logging for all of the following classes:

```
log4j.logger.net.windward=DEBUG
log4j.logger.com.lowagie=DEBUG
log4j.logger.org.apache.commons.codec=DEBUG
log4j.logger.org.apache.poi=DEBUG
log4j.logger.org.apache.xerces.impl.xpath=DEBUG
log4j.logger.org.bouncycastle=DEBUG
log4j.logger.org.dom4j=DEBUG
log4j.logger.org.jaxen=DEBUG
log4j.logger.org.relaxng=DEBUG
# versions 6 - 11 only:
log4j.logger.org.jfree=DEBUG
```

 If logging is slowing you down, please see this article from Apache to [remove or disable unneeded loggers, appenders or layouts](#).

Ready to Run

Java Report Engine is now ready to run. You do not need to reboot your system, however, you may need to reopen your console windows to refresh them with any environment changes.

If you have updated Java Report Engine (or any of its the third-party libraries), then you need to restart any JVMs that use them so they see the new libraries. In the case of a J2EE server, this usually means restarting it.

Test the Installation

Follow the instructions on [Getting Started with Java Report Engine](#) to test your installation.