The Definitive Guide To OpenWebStart

Table of Contents

Introduction

Java Web Start (JWS) was deprecated in Java 9, and starting with Java 11, Oracle removed JWS from their JDK distributions. This means that clients that have the latest version of Java installed can no longer use JWS-based applications. And since public support of Java 8 has ended in Q2/2019, companies no longer get any updates and security fixes for Java Web Start.

This is why some enthusiasts at Karakun decided to create OpenWebStart, an open source reimplementation of the Java Web Start technology. This guide describes how you can use OpenWebStart as a replacement for JWS and continue using your JNLP-based applications with

little or no change at all.

We appreciate your feedback. If you feel that there's a lack of documentation in a certain area or if you find inaccuracies in the documentation, please don't hesitate to contact us at openwebstart@karakun.com or the support forum.

What is OpenWebStart?

OpenWebStart is an open source reimplementation of the Java Web Start technology, released under the GPL with Classpath Exception. It provides the most commonly used features of Java Web Start and the JNLP standard, so that your customers can continue using applications based on Java Web Start and JNLP without any change. OpenWebStart is based on Iced-Tea-Web and follows the JNLP-specification defined in JSR-56.

The main focus of OpenWebStart is the execution of JNLP-based applications. Additionally, the tool contains various modules that simplify your Web Start workflows and let you configure OpenWebStart to suit your needs:

App Manager
: manages the versions of any JNLP-based application that is executed by OpenWebStart.

JVM Manager
: manages Java versions and Java updates on the client.

Control Panel
: provides a graphical user interface to configure OpenWebStart.

Updater
: downloads and installs new versions of OpenWebStart.

Installation

OpenWebStart can be installed on Windows, MacOS and Linux operating systems and there are two different ways to install OpenWebStart:

- Using the **interactive installation** with auto-update functionality
- Using the **unattended installation** for automated roll-outs

If you use Web Start for several small customers or on your own, we recommend using the interactive installer. Our native installer will set up everything on your Windows, Mac, or Linux system so that OpenWebStart is immediately ready to use. OpenWebStart checks for updates automatically, and the Updater component keeps the tool current without the need for any user interaction.

If you or your customers are companies with IT departments of their own, we recommend an unattended installation to roll out OpenWebStart on multiple client machines. Instead of walking through the graphical installer of OWS on every machine your IT department can pre-define the responses for the installation options in a response varfile.

Interactive Installation

Windows

1. Open the ZIP-file.
2. Run the installer.
3. Choose a language and click **OK** to open the OpenWebStart Setup wizard.
4. Click **Next** to start the OpenWebStart installation.
5. Browse to the directory where to install OpenWebStart, and click **Next**.
   Windows default: `C:\Program Files\OpenWebStart`
6. Enable the checkbox to associate the .JNLP suffix with OpenWebStart, and click **Next**.
7. Please wait for OpenWebStart to be installed on your computer.
8. Click **Finish** on the completion screen to close the wizard.

macOS

1. Open the OpenWebStart disk image (DMG file) to mount it.
2. Run the `Open Web Start Installer.app`.
3. Choose a language and click **OK** to open the OpenWebStart Setup wizard.
4. Click **Next** to start the OpenWebStart installation.
5. Browse to the directory where to install OpenWebStart, and click **Next**.
   Default: `/Applications/Open Web Start`
6. Enable the checkbox to associate the .JNLP suffix with OpenWebStart, and click **Next**.
7. Please wait for OpenWebStart to be installed on your computer.
8. Click **Finish** on the completion screen to close the wizard.

Linux

1. Go to the directory where the installer (DEB file) is stored and run the file from the terminal
   `sudo dpkg -i OpenWebStart_linux_1_1_8.deb`
2. Enter your root password.
3. Choose a language and click OK to open the OpenWebStart Setup wizard.
4. Click Next to start the OpenWebStart installation.
5. Browse to the directory where to install OpenWebStart, and click Next.
   Default: `/opt/openwebstart`
6. Enable the checkbox to associate the .JNLP suffix with OpenWebStart, and click Next.
7. Please wait for OpenWebStart to be installed on your computer.
8. Click Finish on the completion screen to close the wizard.

If you need help installing OpenWebStart, also have a look at the public installation and

configuration discussions at the Support Forum.

Unattended Installation

If you or your customers are companies with IT departments of their own, we recommend an unattended installation to roll out OpenWebStart on multiple client machines. In this scenario, the auto-update functionality is inactive; your IT department is free to plan and handle rollouts of new versions based on your internal workflows.

When installing OpenWebStart, several properties can be predefined in a so-called `response.varfile` file.

Some of the supported properties are lockable. If a property is lockable, you can define an additional property of type `PROPERTY_NAME.locked=true` to prevent users from editing the property in the user interface. For example, to define a value for the `ows.jvm.manager.server.default` property that cannot be changed in the user interface, specify the following two properties:

```
ows.jvm.manager.server.default=https://my.custom.server
ows.jvm.manager.server.default.locked=true
```

The following table provides an overview of all properties that can be specified in the response.varfile:

property	lockable	description
ows.jvm.manager.cache.dir	yes	Allows to specify the directory where the JVM cache is located. The follow example shows two examples for Windows: ows.jvm.manager.cache.dir=c:\\temp\\JVMCacheDir or ows.jvm.manager.cache.dir=c\:/temp/JVMCacheDir
ows.jvm.manager.server.default	yes	This property must contain a valid URL that defines the server that is used to download new JVMs.
ows.jvm.manager.server.allowFromJnlp	yes	Defines if a custom URL can be used to download a JVM. Such URL can be part of a JNLP file.
ows.jvm.manager.vendor	yes	Defines a specifc JVM vendor. By doing so, only JVMs from that vendor will be downloaded. You can use ‘*’ to allow any vendor.

property	lockable	description
ows.jvm.manager.vendor.allowFromJnlp	yes	Defines if a vendor attribute in a java/j2se tag of the JNLP file should be respected. Default is false i.e. the vendor from the settings is taken.
ows.jvm.manager.updateStrategy	yes	When starting a JNLP application, OpenWebStart can check if an updated JVM is available to run the application. This property defines how OpenWebstart behaves in the JVM check. Possible values are DO_NOTHING_ON_LOCAL_MATCH, ASK_FOR_UPDATE_ON_LOCAL_MATCH and AUTOMATICALLY_DOWNLOAD
ows.jvm.manager.versionRange	yes	Allows to limit the possible JVM versions. Must be valid version-string according to JSR-56 Appendix A.
deployment.proxy.http.host	yes	The HTTP proxy hostname.
deployment.proxy.https.host	yes	The HTTPS proxy hostname.
deployment.proxy.http.port	yes	The HTTP proxy port.
deployment.proxy.https.port	yes	The HTTPS proxy port.
deployment.proxy.bypass.local	yes	All local hosts should be bypassed. Default is false.
deployment.proxy.bypass.list	yes	A comma separated list of host names that should bypass the proxy.
deployment.proxy.type	yes	The proxy type that should be used. Possible values are 0 (no proxy), 1 (manual proxy, default), 2 (PAC based proxy), 3 (Firefox), 4 (system proxy)
deployment.proxy.auto.config.url	yes	The URL for the proxy auto-config (PAC) file that will be used.
deployment.proxy.same	yes	If true use the same web server and port for https and ftp as is configured for http. (This is only valid if deployment.proxy.type = 1 (manual proxy). Default is false.
deployment.cache.max.size	yes	The cache maximum size. Default is -1
deployment.https.noenforce	yes	If set to true http urls are not converted to https. Default is false.

property	lockable	description
deployment.assumeFileSystemInCodebase	yes	Defines if files from the local filesystem are always handled as if they would be part of the codebase.
deployment.security.whitelist	no	A comma separated list of urls that are defined as whitelist. The whitelist is checked whenever OpenWebStart will download a resource (like a JAR file).
ows.jvm.manager.maxDaysUnusedInJvmCache	yes	Max number of days an unused JVM stays in the JVM cache. The default is 30.
deployment.log	no	If set to true debug logging is enabled. Default is false
deployment.log.file	no	If set to true log is outputted to file. Default is false
ows.update.activated	yes	Defines if OpenWebStart should automatically search for updates.
ows.checkUpdate	yes	This property has no effect and is only used to lock functionality in the user interface. If this property is locked, a user cannot manually search for OpenWebStart updates.
ows.update.strategy.settings	yes	Defines how often OpenWebStart should search for updates when opening the settings windows. Allowed values are ON_EVERY_START, DAILY, WEEKLY, MONTHLY, and NEVER.
ows.update.strategy.launch	yes	Defines how often OpenWebStart should search for updates when starting an application. Allowed values are ON_EVERY_START, DAILY, WEEKLY, MONTHLY, and NEVER.

To create a `response.varfile` file, run the installation of OpenWebStart at least once manually. By doing so a `response.varfile` file is created in OpenWebStart installation folder in your system. In the installation folder, you find a `.install4j` folder that contains the basic `response.varfile` file. The content of such a file looks like this:

```
sys.adminRights$Boolean=false
sys.fileAssociation.extensions$StringArray="jnlp","jnlpx"
sys.fileAssociation.launchers$StringArray="313","313"
sys.installationDir=/Applications/OpenWebStart
sys.languageId=de
```

You can easily edit this file and add additional properties based on the table in this article. Do not change the initial content of the file, and add new properties always to the end of the file. After editing, a `response.varfile` file might look like this:

```
sys.adminRights$Boolean=false
sys.fileAssociation.extensions$StringArray="jnlp","jnlpx"
sys.fileAssociation.launchers$StringArray="313","313"
sys.installationDir=/Applications/OpenWebStart
sys.languageId=de
ows.jvm.manager.server.default=https://my.custom.server
ows.jvm.manager.server.default.locked=true
```

If you now use such a file to install OpenWebStart, all the properties will be automatically imported and used at the first start of OpenWebStart.

Updates

OpenWebStart can be configured to automatically check for new releases and perform automatic updates.

To do so go to the "Updates" Panel in the OWS Settings. It is possible to define an update strategy on every `start`, `daily`, `weekly`, `monthly`, or `never`.

Configuration

There is an extra application to configure OpenWebStart. The executable is located in the installation directory and is named `itw-settings`.

All settings are stored on the file system. For Windows the file is located at `${USER_HOME}\.config\icedtea-web\deployment.properties`. For Mac and Linux the file is located at `${USER_HOME}/.config/icedtea-web/deployment.properties`. This file can be edited with a regular text editor. For some expert configurations this may be necessary but for most cases the graphical UI will be sufficient.

Besides the per user configuration there exists also the possibility to define a system wide configuration. This allows setting up a common configuration for multiple users on a single computer. Or helps in managing a corporate infrastructure where many computers need to be configured identically.

For more details see the sections below.

JVM Manager

<TODO: describe OWS settings options>

JVM Download Server

OpenWebStart can fetch JVMs and JVM updates from a download server that is specified in the JVM Manager Configuration of the OWS Settings application. The default points to `https://download-openwebstart.com/jvms.json`.

Setup a Custom Download Server

If you want to set up your own JVM download server you must provide a json file which lists all available JVMs.

This json file must contain the following data:

```
{
    "cacheTimeInMillis":<miliseconds>,
    "runtimes":[
        {
            "version":<JVM version>,
            "vendor":<vendor name>,
            "os":<OS identifier>,
            "href":<absolute url to the archive containing the JVM>
        },

        ... more runtime definitions
}
```

cacheTimeInMillis

The time which needs to elapse before a client is allowed to contact the server again. Usually the server is accessed once per application startup.

os

Possible values are: MAC64, MAC32, LINUX64, LINUX32, WIN64, WIN32

Cache Management

<TODO: describe OWS settings options>

Proxy Settings

<TODO: describe OWS settings options>

Certificates

<TODO: describe OWS settings options>

Security Settings

<TODO: describe OWS settings options>

Server Whitelists

The "Server Whitelists" panel in OWS settings displays the server whitelist. To define a server whitelist you have to edit the `deployment.properties` file in your config directory with a text editor by adding a new line similar to the following:

```
deployment.security.whitelist=10.10.10.10, google.com, some.server.net
```

The different servers are listed as a comma separated string. Localhost is implicitly always in the white list. If you delete the line again then no whitelisting is applied and all servers are reachable.

Note that whitelisting only applies while downloading resources (jars and jnlps) and not while an application is running. Thus an application can open a connection to a server which is not in the white list.

It is also possible to specify the content of the whitelist in the response file of an unattended OWS installation.

Logging

<TODO: describe OWS settings options>

Remote Debugging

<TODO: describe OWS settings options>

System Configuration

When loading the configuration during the start of OpenWebStart the following steps are executed:

1. Load the default values which are hardcoded in the source code.
2. Search for a system configuration.
3. Load the system configuration if one was found.
4. Load the user configuration.

Whenever a configuration is loaded the values which are already defined are updated. There is however the possibility to lock a property. If a property is locked then subsequent configurations

may not modify the value. This allows enforcing certain values on a system level. Any changes the user makes in his personal configuration file will not have any effect on the locked property.

Defining the System Configuration

The system configuration needs to be defined in the following way.

Windows: create the file `%windir%\Sun\Java\deployment\deployment.config` and add the following properties:

MacOs and Linux: create the file `/etc/.java/deployment/deployment.config` and add the following properties:

deployment.system.config

The URL to the system configuration. The name of the file can be freely chosen. Special characters need escaping. See the following examples:

- `deployment.system.config=file\:/C\:/Window/Sun/Java/global.properties`
- `deployment.system.config=file\:/etc/.java/deployment/base.properties`
- `deployment.system.config=https\:192.168.1.1./javaws/system.properties`

deployment.system.config.mandatory

If set to `true` then OpenWebStart will fail if it is unable to load the system settings This property is optional. The default value is `false`.

The final file should look something like this:

```
deployment.system.config=https\:192.168.1.1./javaws/system.properties
deployment.system.config.mandatory=true
```

Content of the System Configuration

The simplest way to create a system configuration is to start the `itw-settings`. After saving the configuration the modified properties are written to the user configuration file. For Windows the file is located at `${USER_HOME}\.config\icedtea-web\deployment.properties`. For Mac and Linux the file is located at `${USER_HOME}/.config/icedtea-web/deployment.properties`.

The customized user configuration can be used as a starting point for the system configuration. Simply copy the file and remove the properties which should not be defined on the system level.

OpenWebStart does not save properties which have the default value. Therefore the generated user configuration may not contain all the values you wish to enforce on the system level.

Please contact openwebstart@karakun.com if you need to know the key and valid values for a specific configuration.

Locking a property

One of the use cases is to enforce some configurations to all users in your corporate environment. This can be achieved by locking configuration on a system level. To lock a property you need to define a second entry with a `.locked` postfix.

Here an example:

```
ows.jvm.manager.server.default=https\://192.168.1.1/jvms.json
ows.jvm.manager.server.default.locked=true
```

TIP the value of `ows.jvm.manager.server.default.locked` is ignored. The presence of the key is sufficient for locking the property.