The Defintive Guide To OpenWebStart

Table of Contents

Introduction

<this documentation is work in progress>

What is OpenWebStart?

<this documentation is work in progress>

Installation

<this documentation is work in progress>

Basic Installation

Unattended Installation

OWS also offers an unattended installation mode which is useful when you have to install OWS on many computers. Instead of walking through the graphical installer of OWS on every machine you can provide the options as a response file.

<add more details as on the homepage>

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
```