# Frequently Asked Questions on OpenWebStart

## Table of Contents



| NOTE | The authors assume no responsibility or liability for any errors or omissions in the content of this documentation. The information contained in this documentation is provided on an "as is" basis with no guarantees of completeness, accuracy, usefulness, timeliness or of the results obtained from the use of this information. |
|---|---|

<this documentation is work in progress>

# General

### What is OpenWebStart?

<this documentation is work in progress>

# Subscription and Premium Support

### Is there a subscription for OpenWebStart?

Basic subscription provided by Karakun is the cost-effective way to support OpenWebStart. With

this option you may report bugs using a dedicated communication channel and your bug reports will have a higher priority. If you wish, we will add your company logo to our sponsor list for free. Plus, you'll get a 10% discount for development of individual features.

### Is there a premium support for OpenWebStart?

With premium support provided by Karakun you may report bugs or issues using the premium communication channel. Your bug reports will have the highest priority. Karakun will guarantee reaction times depending on the severity of bugs (during business hours). Plus, you'll get a 25% discount for the development of individual features. Upon your wish, your company logo will be listed as a premium sponsor on our website.

### What are the costs for subscription and premium support?

Please contact openwebstart@karakun.com to figure out which support model fits your needs best.

# Installation

### Do I have to install a Java on my system to run OWS?

OpenWebStart comes with a JVM Manager and a bundled Java runtime. There is no need to have any JVM installed on your system.

The bundled JVM is used solely to run OWS. The JVMs manged by the JVM Manager are used to run the JNLP applications.

### Which Java versions can be used to run my JNLP applications?

Your JNLP applications can be run with either Java 8 and 11. We have not run any tests with Java 12 or 13 yet but from the experience collected while adding support for Java 11 we do not expect any big obstacles.

OpenWebStart JVM Manager will download and manage JVMs as they are requested by an application. Currently, we provide Azul and Adopt JDKs on the default download server. In the future we will add more vendors.

If your preferred JVM is not provided by our default download server, you can either add locally installed JVMs or host your own download server for JVMs to run your JNLP applications.

### Is there an MSI installer for OpenWebStart?

Currently, we do not provide an MSI installer. More details on the reasons can be found at https://github.com/karakun/OpenWebStart/issues/98.

# Debugging and Error Reporting

### Does OWS write log files and where do I find them?

The log files of OWS are located in a hidden directory below your user home directory:

```
<user-home>/.config/icedtea-web/log
```

where `<user-home>` depends on your operating system. By default this should be

```
<root>\Users\<your-username>    (Windows 10)
/Users/<your-username>          (MacOS)
/home/<your-username>/          (Linux)
```

Note that logging has to be activated using the OWS Settings application by checking "Activate debug logging" and "Log to file" on the "Logging" tab.

# Functionality

### Does OpenWebStart support Applets?

Applets are not supported and there are no plans to support them in the future. We also do not consider this as a deviation from the JNLP-standard as this is an optional feature according to the JSR-56 specs.

### How can I define a server whitelist for OWS?

This field is currently not editable in the UI. Edit the deployment properties file ${userHome}/.config/icedtea-web/deployment.properties file with a text editor by adding a new line:

```
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 when installing OWS (unattended installation), See https://openwebstart.com/installation/ and https://openwebstart.com/configuration/ for further details.

### How to run OpenJFX based JavaFX applications with OpenWebStart?

#### With JDK 8

To be able to run a JavaFX application with OWS using Java 8 requires an installation of Java 8 JVM that includes JavaFX. Some of the vendors that have JavaFX as part of their Java 8 JVMs are Oracle, Azul, BellSoft and Amazon. OpenJDK 8 from Adopt does not include JavaFX.

Using the JVM Server feature of OWS JVM Manager it can be ensured that a suitable JDK 8 with JavaFX will be installed on the machine for OWS to start a JavaFX app. The required JVM from a preferred vendor can be specified in the Jnlp file:

```
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0+" codebase="https://myhost.com">
<information>
    <title>JavaFX 8 App</title>
    <vendor>Karakun AG</vendor>
    <offline-allowed/>
</information>
<security>
    <all-permissions/>
</security>
<resources>
    <java version="1.8*" vendor="zulu" href="http://myjvmserver.com/jvms/jvms.json"/>
    <jar href="generated-jars/javafx-test.jar"/>
</resources>
<application-desc main-class="com.karakun.ows.javafx_test.HelloWorld"/>
</jnlp>
```

In the above Jnlp file, the <java> tag specifies the name of the JVM vendor and URL of the JVM server which hosts a JSON that points to the appropriate JVM with JavaFX. For example:

```
{
  "cacheTimeInMillis":5000,
  "runtimes":
  [
    {
     "version":"1.8.0_252",
     "vendor":"Zulu Community Edition",
     "os":"WIN64",
     "href":"https://cdn.azul.com/zulu/bin/zulu8.46.0.19-ca-fx-jdk8.0.252-win_x64.zip"
    }
  ]
}
```

When the above Jnlp file is started with OWS, OWS will install the specified JVM with JavaFX for running the JavaFX application.

**With JDK 11+**

Create a JavaFX project which should have OpenJFX libraries on its path for compilation. Required OpenJFX version can be obtained from: OpneJFX Download.

In order to deploy a JavaFX application using OWS:

1. package the JavaFX application in a jar
2. gather platform (OS) specific jars from the OpenJFX libraries.
3. all jars must be signed and must have required security attributes in their manifests.
4. deploy all jars in a server
5. create a Jnlp file. For example The following file is meant to run on Windows. However one can also specify OS specific jars under OS specific <resources>

```
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0+" codebase="https://myhost.com">
<information>
    <title>JavaFX 11 App</title>
    <vendor>Karakun AG</vendor>
    <offline-allowed/>
</information>
<security>
    <all-permissions/>
</security>
<resources>
    <java version="11+"/>
    <jar href="jars/jfxapp.jar"/>
    <jar href="jars/javafx-controls-11.0.2-win.jar"/>
    <jar href="jars/javafx-graphics-11.0.2-win.jar"/>
    <jar href="jars/javafx-base-11.0.2-win.jar"/>
    <jar href="jars/javafx-fxml-11.0.2-win.jar"/>
</resources>
<application-desc main-class="com.karakun.ows.javafx_test.HelloWorld11Launcher"/>
</jnlp>
```

**Note:** When running with Java 11+ the JavaFX Application is required to be launched via a launcher class:

```
// Launcher for JavaFX application which is specified in the Jnlp file
public class HelloWorld11Launcher {
    public static void main(String[] args) {
        HelloWorld11.main(args);
    }
}

// JavaFX Application
public class HelloWorld11 extends Application {
    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        ... // JavaFX code
    }
    ...
}
```