

# **OWF Build Instructions**

DOD GOSS

Exported on Oct 16, 2018

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Objectives .....	3
1.2	Document Scope .....	3
1.3	Requirements .....	3
1.3.1	Download and Install Development Dependencies .....	3
1.3.2	Verify Tool Installations .....	4
<b>2</b>	<b>Build Instructions.....</b>	<b>6</b>
2.1	Copy OWF v7.17.1 WAR File.....	6
2.2	Install Project Dependencies.....	6
2.3	Run OWF Framework Server (Development) .....	7
2.4	Run OWF Framework Server (Production) .....	7
2.5	Create WAR and Tomcat 8.5 Bundle .....	7
2.6	Clean the Build Directory (optional).....	7
2.7	Additional Build Commands .....	8
2.7.1	Build JavaScript Resources .....	8
2.7.2	Build Themes Resources .....	8
2.8	Install Patch .....	8
2.9	Running the Bundle .....	9
2.9.1	Windows start script.....	9
2.9.2	Linux start script.....	9

# 1 Introduction

## 1.1 Objectives

The purpose of this document is to describe how to build the following projects:

- **OWF Server** – A local copy of the bundled ZIP file as well as the release version.
- **OZONE Security JAR** – A new version of the ozone-security JAR file.

## 1.2 Document Scope

This document is intended for widget developers and those who wish to build the application from source.

It assumes familiarity with the target development environment, such as Microsoft Windows or Linux, as well as basic command line operations.

## 1.3 Requirements

Before any of the build tasks can run, the application dependencies. The following sections list the tools, the version requirements for recent OWF releases and nominal configuration elements.

Instructions provided below assume a Microsoft Windows development environment for illustrative purposes only.

### 1.3.1 Download and Install Development Dependencies

The build and development process for OWF requires that the following development dependencies be installed:

- Java Development Kit (JDK)
- Node.js
- NPM

The following dependencies are provided as part of the build process, but may be installed locally at the developer's preference:

- Gradle <sup>1</sup>
- Groovy
- Grails

<sup>1</sup> A build wrapper script, `gradlew`, is provided in the source distribution. If you do not wish to use the wrapper, a local Gradle installation is required, and the `gradle` command may be used in place of the `gradlew` command in the examples.

**IMPORTANT: Only the dependency versions listed in the tables below are tested and supported. Deviations from the listed versions may cause the build to be unsuccessful.**

#### 1.3.1.1 OWF Development Tools and Version Numbers (for versions 7.17.2.0 and later)

Application	JDK	Grails	Groovy	Gradle	Node.JS	NPM
<b>OWF 7.17.2.0</b>	1.8	3.3.1	2.4.12	4.2.1	8.6.0	5.3.0

## 1.3.1.2 OWF Development Tools and Version Numbers (for versions prior to 7.17.2.0)

Application	JDK	ANT	Ant Contrib	GRAILS	RUBY	COMPASS*	SASS*	GROOVY
<b>OWF 4</b>	1.6	1.7	-	1.3.7	1.8.7	0.11.3	3.1.3	-
<b>OWF 5</b>	1.6	1.8	-	1.3.7	1.8.7	0.11.3	3.1.3	-
<b>OWF 6</b>	1.6	1.8	1.0b3	1.3.7	1.9.2	0.11.3	3.1.3	1.8.8
<b>OWF 7</b>	1.6	1.8.3	1.0b3	1.3.7	1.9.2	0.11.3	3.1.3	1.8.8
<b>OWF 7.16.0</b>	1.7	1.8.4	1.0b3	2.3.7	1.9.2	0.11.7	3.1.3	<ul style="list-style-type: none"> <li>2.1.9 (included with Grails)</li> <li>1.8.8 for OWF</li> </ul>
<b>OWF 7.17.0</b>	1.8	1.8.4	-	2.4.0	1.9.2	0.11.7	3.1.3	<ul style="list-style-type: none"> <li>2.3.0 (included with Grails)</li> <li>1.8.8 for OWF</li> </ul>

Obtain installation media and instructions for the various operating systems from the primary websites for each tool or trusted download source. The default locations are provided below. Also, install the tools in the order listed below. Once all tools have been installed, the following sections will describe how to configure the environment.

## 1.3.1.3 Tool Provider Websites

Application	Location
<b>JDK</b>	<a href="http://www.oracle.com/technetwork/java/javase/downloads/index.html">http://www.oracle.com/technetwork/java/javase/downloads/index.html</a>
<b>Grails</b>	<a href="http://www.grails.org/">http://www.grails.org/</a>
<b>Groovy</b>	<a href="http://groovy-lang.org/">http://groovy-lang.org/</a>
<b>Gradle</b>	<a href="https://gradle.org/releases/">https://gradle.org/releases/</a>
<b>Node.js</b>	<a href="https://nodejs.org/en/">https://nodejs.org/en/</a>
<b>NPM</b> <sup>1</sup>	<a href="https://www.npmjs.com/">https://www.npmjs.com/</a>

<sup>1</sup> NPM is usually provided as part of the Node.js distribution and may not need to be installed separately.

## 1.3.2 Verify Tool Installations

To verify the installation and version of the tools, use the version command for each tool in a Command Prompt Window.

Example commands and output for an OWF 7 environment follow:

**Java**

```
C:\> java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)
```

**Grails**

```
C:\> grails -version
Welcome to Grails 3.3.3 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: C:\Grails\grails-3.3.3
```

### **Groovy**

```
C:\> groovy -version
Groovy Version: 2.4.12
```

### **Node**

```
C:\> node -v
v8.6.0
```

### **NPM**

```
C:\> npm -v
5.3.0
```

## 2 Build Instructions

### 2.1 Copy OWF v7.17.1 WAR File

**IMPORTANT:** This step must be completed prior to building the OWF Framework.

Building the OWF v7.17.2.0 patch release requires that the OWF v7.17.1 WAR file be placed in the `lib` directory in the source folders.

1. Download and extract the v7.17.1 bundle.  
<https://github.com/ozoneplatform/owf-framework/releases/download/v7.17.1/OWF-bundle-7.17.1.zip>
2. Copy the `owf.war` file from the v7.17.1 bundle Tomcat webapps folder to the `lib` folder in the OWF Framework source folders (example: `.\owf-framework\lib\owf.war`)

```
C:\> copy C:\owf-bundle-7.17.1\apache-tomcat\webapps\owf.war
C:\src\owf-framework\lib
```

### 2.2 Install Project Dependencies

**IMPORTANT:** Dependencies must be installed in the order listed below.

Order	Module	Command	Dependencies
1	<b>ozone-classic-bom</b>	gradlew install	n/a
2	<b>owf-appconfig</b>	gradlew install	ozone-classic-bom
2	<b>owf-auditing</b>	gradlew install	ozone-classic-bom
2	<b>owf-messaging</b>	gradlew install	ozone-classic-bom
2	<b>owf-security</b>	gradlew install	ozone-classic-bom
2	<b>owf-custom-tomcat</b>	gradlew install	ozone-classic-bom
3	<b>owf-framework</b>	gradlew bundle	ozone-classic-bom, owf-appconfig, owf-auditing, owf-messaging, owf-security, owf-custom-tomcat

*Example:*

```
C:\src\ozone-classic-bom> gradlew install

C:\src\owf-appconfig> gradlew install

C:\src\owf-auditing> gradlew install

C:\src\owf-messaging> gradlew install

C:\src\owf-security> gradlew install
```

```
C:\src\owf-custom-tomcat> gradlew install
```

## 2.3 Run OWF Framework Server (Development)

The `:bootRun` task will build the project and start the OWF Framework server in development mode.

```
C:\src\owf-framework> gradlew :bootRun
```

## 2.4 Run OWF Framework Server (Production)

The `:bootRun` task will build the project and start the OWF Framework server in production mode.

```
C:\src\owf-framework> gradlew :bootRun -Dgrails.env=production
```

## 2.5 Create WAR and Tomcat 8.5 Bundle

The `:bundle` task will build the projects, create the OWF WAR file, and create the Tomcat 8.5 ZIP distribution bundle.

```
C:\src\owf-framework> gradlew :bundle
```

After it is built, the ZIP file containing the pre-configured Tomcat 8.5 container and the OWF WAR can be found at `.\owf-framework\build\ozone-framework-V.V.V.zip`.

**IMPORTANT:** To run the generated OWF v.7.17.2.0 application bundle, you must follow the instructions in the Install Patch section.

## 2.6 Clean the Build Directory (optional)

The `:clean` task will clear the build cache for the projects.

**Notice:** If the build fails (especially after pulling new changes), run a full clean of the project, and then retry to build.

```
C:\src\owf-framework> gradlew clean
```

## 2.7 Additional Build Commands

### 2.7.1 Build JavaScript Resources

*Note: This task is automatically run during the build and does not need to be run manually.*

The `:buildJavascript` task will build/compile the client JavaScript files using Node.js and Gulp.

```
C:\src\owf-framework> gradlew :buildJavascript
```

This task supports incremental compilation. If the input files have not changed, the task will be automatically skipped.

### 2.7.2 Build Themes Resources

*Note: This task is automatically run during the build and does not need to be run manually.*

The `:buildThemes` task will build/compile the theme files using JRuby, SASS, and Compass.

```
C:\src\owf-framework> gradlew :buildThemes
```

This task supports incremental compilation. If the input files have not changed, the task will be automatically skipped.

## 2.8 Install Patch

**IMPORTANT:**The patch **MUST** be applied using a pre-existing OWF v7.17.1.0 .WAR file.

1. Download and extract the v7.17.1 bundle.  
<https://github.com/ozoneplatform/owf-framework/releases/download/v7.17.1/OWF-bundle-7.17.1.zip>
2. Copy the v7.17.1 .WAR file into the new bundle root directory:

```
C:\> copy C:\owf-bundle-7.17.1\apache-tomcat\webapps\owf.war
C:\owf-bundle
```

3. From the root directory of the v7.17.2.0 bundle, run the patch to do an in-place update of the new `owf.war` file.

```
C:\owf-bundle> java -jar owf-patch.jar owf.war
tomcat\webapps\owf.war
```

4. Verify that the patch has been successfully applied.  
The Tomcat `webapps` folder (example: `C:\owf-bundle\tomcat\webapps`) should contain the following files:
  - a. `owf.war` – the patched WAR file
  - b. `owf.war.orig` – a copy of the original unpatched WAR file



## 2.9 Running the Bundle

Scripts to start the server are included in the bundle in the `tomcat` directory.

The server may be started in **development mode** to utilize the embedded, in-memory H2 database. The initial data is generated by the application.

To run the server in **production mode**, the database must be pre-populated with the initial data using the supplied database scripts. Please refer to the **OWF Configuration Guide** for instructions on configuring the database and running the initial data scripts.

### 2.9.1 Windows start script

*Example:*

```
C:\owf-bundle\tomcat> start.bat /dev
```

*Usage:*

```
start.bat [/dev] [/db database]
/dev      Start in DEVELOPMENT mode
/db       Use the selected database configuration
database  h2      - Embedded H2 file-based database (default)
          pg      - PostgreSQL
          mysql   - MySQL
          oracle  - Oracle RDBMS
          mssql   - Microsoft SQL Server
```

### 2.9.2 Linux start script

*Example:*

```
/opt/owf-bundle/tomcat$ ./start.sh --dev
```

*Usage:*

```
./start.sh [--dev] [--db database]
--dev      Start in DEVELOPMENT mode
--db       Use the selected database configuration
database  h2      - Embedded H2 file-based database (default)
          pg      - PostgreSQL
          mysql   - MySQL
          oracle  - Oracle RDBMS
          mssql   - Microsoft SQL Server
```