OWF Build Instructions

DOD GOSS

Table of Contents

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.

2   Requirements

This document is targeted to widget developers and assumes a basic familiarity with the target development environment, be it Microsoft Windows or Linux. Before any of the build tasks can run, the developer must install all 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.

2.1   Download and Install Applications

OWF Development requires the use of the Java Development Kit (JDK), Groovy (optional), Node.js, NPM, Gradle (optional), and Grails (optional). Since we provide the gradlew, the Groovy, Grails, and Gradle installs are optional. If you do not use the gradlew but use the gradle commands then you will need these installed. The following table lists the versions of each tool required by the last few releases of OWF and the Store.

The term Store and Marketplace can be used interchangeably.

Table 1: OWF Development Tools and Version Numbers

Application	JDK	ANT	Ant Contrib	GRAILS	RUBY	COMPASS*	SASS*	GROOVY
OWF 4	1.6	1.7	-	1.3.7	1.8.7	0.11.3	3.1.3	-
OWF 5	1.6	1.8	-	1.3.7	1.8.7	0.11.3	3.1.3	-
OWF 6	1.6	1.8	1.0b3	1.3.7	1.9.2	0.11.3	3.1.3	1.8.8
OWF 7	1.6	1.8.3	1.0b3	1.3.7	1.9.2	0.11.3	3.1.3	1.8.8
OWF 7.16.0	1.7	1.8.4	1.0b3	2.3.7	1.9.2	0.11.7	3.1.3	• 2.1.9 (included with Grails) • 1.8.8 for OWF
OWF 7.17.0	1.8	1.8.4	-	2.4.0	1.9.2	0.11.7	3.1.3	• 2.3.0 (included with Grails) • 1.8.8 for OWF

Table 2: OWF Server 7.17.2+ Development Tools and Version Numbers

Application	JDK	Grails	Groovy	Gradle	*Node.JS*	NPM
OWF 7.17.2.0	1.8	3.3.1	2.4.12	4.2.1	*8.6.0*	5.3.0

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.

Table 3: Tool Provider Websites

Application	Location
JDK	http://www.oracle.com/technetwork/java/javase/downloads/index.html
Grails	http://www.grails.org/
Groovy	http://groovy-lang.org/

Application	Location
Gradle	https://gradle.org/releases/
Node.js	https://nodejs.org/en/
NPM	+https://www.npmjs.com/+

2.2 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:

1. Enter **java –version**

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)

1. Enter **grails –version**

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

1. Enter **groovy –version**

C:\> groovy –version
Groovy Version: 2.4.12

1. Enter **node –v**

C:\> node –v
v8.6.0

1. Enter **npm –v**

C:\> npm –v
5.3.0

3   Install Build Dependencies

Note: Dependencies must be installed in the order listed below.

Order	Module	Latest Branch	Command	Dependencies
1	**ozone-classic-bom**	develop	gradle install	n/a
2	**owf-appconfig**	develop	gradle install	ozone-classic-bom
2	**owf-auditing**	develop	gradle install	ozone-classic-bom
2	**owf-messaging**	develop	gradle install	ozone-classic-bom
2	**owf-security**	develop	gradle install	ozone-classic-bom
2	**owf-custom-tomcat**	develop	gradle install	ozone-classic-bom
3	**owf-framework**	develop	gradle bundle	ozone-classic-bom, owf-appconfig, owf-auditing, owf-messaging, owf-security, owf-custom-tomcat

All build steps are run using Gradle.

To use the Gradle wrapper instead of a locally installed version, use the gradlew script found in the root project directory in place of the gradle command.

owf-framework\>: gradlew :bundle

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

owf-framework\>: gradle clean

4 Run OWF Framework Server (Development)

The **:bootRun** task will build the project, and start the OWF Framework server in development mode.
owf-framework\>: gradle :bootRun

5 Run OWF Framework Server (Production)

The **:bootRun** task will build the project, and start the OWF Framework server in production mode.
The **-Dowf.db.init=true** option is only required for the first run, if the database is persisting changes between runs.
owf-framework\>: gradle :bootRun -Dgrails.env=production -Dowf.db.init=true

6 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.*

owf-framework\\>: gradle :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 OWF v.7.17.2.0, you must follow the instructions in section 1.7**

Batch files and shell scripts to start the server are included in the root of the bundle: start (.sh|.bat) for production mode, and start-dev (.sh|.bat) for development mode.

Important: for the first time running the bundle, the database must be initialized using:

start.bat /init

Or

start.sh init

7 Install Patch

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

1. Download the OWF v7.17.2 WAR Bundle or Create WAR and Tomcat 8.5 Bundle *(See Section 1.6 - Above)*

2. Download & Extract OWF v7.17.1 WAR Bundle

3. Copy the v17.17.1 WAR (**OWF-bundle-7.17.1/apache-tomcat/webapps/owf.jar**) into the **new** bundle root directory (e.g. **/OWF-bundle-7.17.2/owf.jar**)

4. From the **new** bundle root directory run patch to do an in-place update of the new owf.jar

owf-bundle/>: java -jar owf-patch.jar owf.war tomcat/webapps/owf.war

1. Verify patch has been successfully applied.

1. **Check OWF-bundle-7.17.2/tomcat/webapps/ folder and confirm the patched WAR owf.war file and a copy of the original, unpatched WAR owf.war.orig exists**

7.1 Additional Build Commands

7.1.1 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.
owf-framework/>: gradle :buildJavascript
This task supports incremental compilation: if the input files have not changed, the task does not have to be re-run.

7.1.2 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.
owf-framework/>: gradle :buildThemes
This task supports incremental compilation: if the input files have not changed, the task does not have to be re-run.