

# Ozone Build Instructions

DOD GOSS

Version 8.0.0.0, 2019-07-01

# Table of Contents

- 1. Introduction** ..... **1**
  - 1.1. Objectives..... 1
  - 1.2. Document Scope..... 1
  - 1.3. Requirements ..... 1
- 2. Download and Install Development Dependencies** ..... **2**
- 3. Verify Tool Installations**..... **4**
- 4. Build Instructions** ..... **5**
  - 4.1. Build & Install Project Modules ..... 5
  - 4.2. Run OWF Framework Server (Development) ..... 6
  - 4.3. Run OWF Framework Server (Production) ..... 6
  - 4.4. Build WAR and Tomcat Bundle ..... 6
  - 4.5. Clean the Build Directory (optional) ..... 6
  - 4.6. Running the Bundle ..... 7
    - 4.6.1. Windows start script ..... 7
    - 4.6.2. Linux start script..... 7

# 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.

## 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 development dependencies must be installed. 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. 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.



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.

Table 1. Development Tools and Version Numbers

Dependency	Version	Download Location
Oracle Java JDK	1.8	<a href="http://www.oracle.com/">http://www.oracle.com/</a>
Node.js	10.16.0	<a href="https://nodejs.org/">https://nodejs.org/</a>
NPM <sup>1</sup>	6.9.0	<a href="https://www.npmjs.com/">https://www.npmjs.com/</a>
Gradle	4.2.1	<a href="https://gradle.org/">https://gradle.org/</a>
Grails	3.3.2	<a href="http://www.grails.org/">http://www.grails.org/</a>
Groovy	2.4.12	<a href="http://groovy-lang.org/">http://groovy-lang.org/</a>

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

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. Once all tools have been installed, the following sections will describe how to configure the environment.

### 3. 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 3.3.2 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: C:\Grails\grails-3.3.2
```

#### *Groovy*

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

#### *Node*

```
C:\> node -v
v10.16.0
```

#### *NPM*

```
C:\> npm -v
6.9.0
```

## 4. Build Instructions

### 4.1. Build & Install Project Modules



Modules must be installed in the order listed below.

Table 2. Project Modules

Order	Module	Location	Command
1	ozone-classic-bom	/ozone/ozone-framework-server/ozone-classic-bom/	gradlew install
2	owf-appconfig	/ozone/ozone-framework-server/owf-appconfig/	gradlew install
3	owf-auditing	/ozone/ozone-framework-server/owf-auditing/	gradlew install
4	owf-messaging	/ozone/ozone-framework-server/owf-messaging/	gradlew install
5	owf-security	/ozone/ozone-framework-server/owf-security/	gradlew install
6	owf-custom-tomcat	/ozone/ozone-framework-server/owf-custom-tomcat/	gradlew install
7	ozone-client	/ozone/ozone-framework-client/	gradlew install
8	owf-framework	/ozone/ozone-framework-server/owf-framework/	gradlew bundle

Example

```
C:\ozone\ozone-framework-server\ozone-classic-bom> gradlew install

C:\ozone\ozone-framework-server\owf-appconfig> gradlew install

C:\ozone\ozone-framework-server\owf-auditing> gradlew install

C:\ozone\ozone-framework-server\owf-messaging> gradlew install

C:\ozone\ozone-framework-server\owf-security> gradlew install

C:\ozone\ozone-framework-server\owf-custom-tomcat> gradlew install

C:\ozone\ozone-framework-client> gradlew install
```

## 4.2. Run OWF Framework Server (Development)

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

*Example*

```
C:\ozone\ozone-framework-server\owf-framework> gradlew :bootRun
```

## 4.3. Run OWF Framework Server (Production)

The `:bootRun` task and the `-Dgrails.env=production` argument will build the project and start the OWF Framework server in production mode.

*Example*

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

## 4.4. Build WAR and Tomcat Bundle

The `:bundle` task will build the projects, create the OWF WAR file, and create the Tomcat v8.5.23 ZIP distribution bundle.

*Example*

```
C:\ozone\ozone-framework-server\owf-framework> gradlew :bundle
```

After it is built, the ZIP file containing the pre-configured Tomcat v8.5.23 container and the OWF WAR can be found at `.\owf-framework\build\ozone-framework-8.0.0.0.zip`.

## 4.5. Clean the Build Directory (optional)

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

*Example*

```
C:\ozone\ozone-framework-server\owf-framework> gradlew :clean
```



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



## 4.6. 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.

### 4.6.1. 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
```

### 4.6.2. 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
```