

Ozone Quick Start Guide

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. Related Documents	1
2. Components	2
2.1. Toolbar	2
2.2. User menu	2
2.3. Dashboards	3
2.3.1. Dashboards menu	4
2.4. Widgets	5
2.4.1. Widget toolbar	5
2.5. Intents	6
2.5.1. How to Use Intents	6
2.6. Themes	7
2.7. The Marketplace	7
3. The Ozone Bundle	9
3.1. Prerequisites	9
3.1.1. Node.js and NPM	9
3.1.2. Java Runtime Environment (JRE)	9
3.2. Instructions	9
3.2.1. Overview	9
3.2.2. Unzipping the bundle	9
3.2.3. Starting OWF	10
3.2.3.1. Windows start script	10
3.2.3.2. Linux start script	11
3.2.4. Accessing OWF	11
3.3. Authenticating to OWF	11
3.3.1. Using the login form	11
3.3.2. Certificate authentication	12
3.3.2.1. Installing a user certificate using Internet Explorer (IE)	12
3.3.2.2. Installing a user certificate using Firefox	13
4. Development Build Process	15
4.1. Additional requirements	15
4.2. Starting the frontend	15
4.3. Starting the backend	15
4.3.1. Accessing OWF	16
5. Allowing Remote Access to OWF	17
5.1. Identifying a Server Name	17
5.2. Generate a Server Certificate	17
5.2.1. Install the Server Certificate	17
5.3. Modify the Externalized Configuration Files	18
Glossary	19

1. Introduction

1.1. Objectives

The purpose of this guide is to explain how to use the Ozone Widget Framework (OWF). This is including, but not limited to, the use of application components, full applications and their configuration settings. This guide provides an introduction to the Ozone Widget Framework (OWF). OWF consists of an environment and a set of tools used for discovering, organizing and displaying Web applications in a single browser window. The guide explains how to use OWF, set up an OWF environment on a user's local machine and navigate OWF security.

1.2. Document Scope

This guide is not an exhaustive reference. It is intended for users, developers and administrators seeking a quick introduction on how to deploy, start and use OWF. For information about specific areas, see the relevant documentation included with the OWF bundle.

In this document, the terms Store and Marketplace are used interchangeably. Both applications share features described in this guide and both may be configured to OWF.

The OWF Bundle ships with Tomcat v8.5.23 which requires JDK 1.7 or higher. If running OWF with a web server other than Tomcat, please see that Web server's documentation for requirements.

1.3. Related Documents

Table 1. Related Documents

Document	Purpose
Quick Start Guide	Walkthrough of basic OWF functions such as using widgets; unpacking the OWF bundle; setting up a local instance of OWF; installing security certificates; truststore and keystore configuration.
User's Guide	Understanding the OWF user interface; adding, deleting, modifying widgets and using intents; accessing and using the Store; using dashboards; creating, deleting, adding, switching, modifying dashboard pages; defining accessibility features such as high-contrast themes.
Administrator's Guide	Understanding administrative tools: adding, deleting, and editing users, groups, widgets, and dashboards; creating default content for users, groups and group dashboards.
Configuration Guide	Overview of basic architecture and security; OWF installation instructions; instructions for modifying default settings; database set up and logging guidance; framework and theme customization instructions; OWF upgrade instructions; directions for adding and deleting help content.

2. Components

This is an overview explaining how to start using the Ozone widget Framework. Find detailed information in the OWF User’s Guide.

2.1. Toolbar

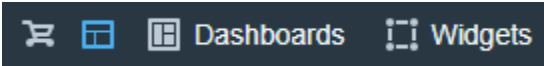


Figure 1. Toolbar (left)

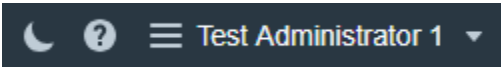


Figure 2. Toolbar (right)

Table 2. Toolbar buttons

	Marketplace view Select and switch the view to a Marketplace, to discover and add dashboards and widgets. (Note: If OWF is not connected to a Marketplace, this icon will not appear.)
	Dashboard view Switch the view the active dashboard.
	Dashboards menu Used to find, start and manage dashboards.
	Widgets menu Used to list available widgets and them to the dashboard.
	Theme toggle Toggle between Light to Dark themes.
	Help Online repository of Ozone guides and tutorials.

2.2. User menu

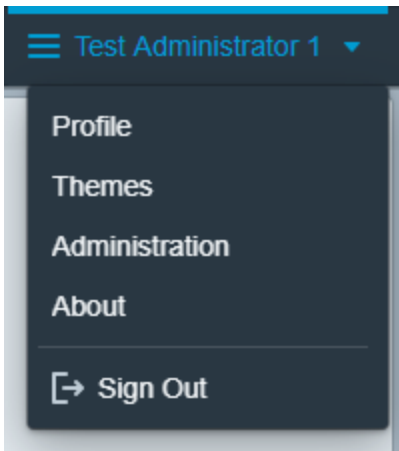


Figure 3. User drop-down menu

Table 3. User menu options

Profile	View user information for the currently logged in user.
Administration	Manage stacks, widgets, users, groups and system configuration settings. <i>(Note: This option will only appear to users with Administrator privileges.)</i>
About	Information box containing a description of the current version of the application.
Sign Out	Log out of the application (if enabled).

2.3. Dashboards

In simple terms, a dashboard is a screen where a user can dictate which widgets to load, which layouts to use, and the arrangement of the widgets within the specified layouts.

Each time a saved dashboard loads, the layout maintain the same look and feel as the last time the dashboard was accessed by the user.

Users can receive dashboards by the following methods:


- Create their own
- Add from the Marketplace
- Assign to individual user by an administrator
- Assign to a group by an administrator

Group-assigned dashboards provide identical layouts and widgets for each member of a group. Each group member can customize their instance of a pre-configured dashboard. dashboards that have not been created by the user can be restored to their default states.

2.3.1. Dashboards menu

The dashboard menu lists all of the user's dashboards. The dashboards included here are Ozone dashboards, either created in OWF or obtained from the Marketplace.

From the dashboard menu, users can start a dashboard, create new ones, or search for dashboards in the Marketplace. Dashboards can be shared, restored, edited or deleted through the manager buttons in the dashboard menu.

To open the dashboard menu, click the dashboard button  in the OWF toolbar.

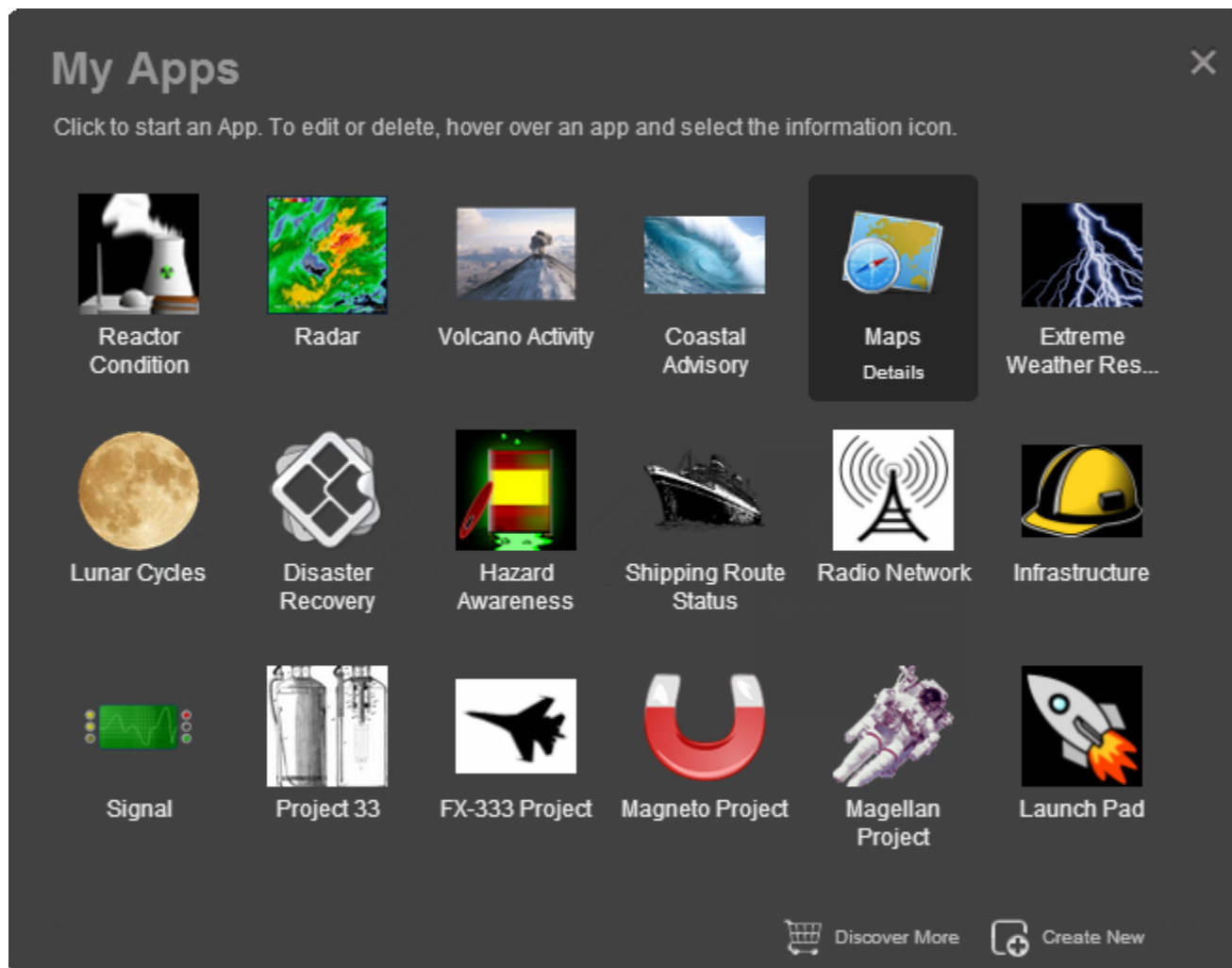


Figure 4. Dashboards menu

Opening a dashboard

Dashboards can contain one or more layouts called Pages. If a dashboard has only one page, then clicking its icon in the dashboard Menu will start it.


To open a dashboard:

1. Click the dashboards Menu button in the toolbar to open the dashboards Menu.
2. Click a dashboard.
 - a. If it has one page, it will automatically open.
 - b. If it has more than one page, then clicking the dashboard's icon will open a carousel displaying all the dashboards's pages.

2.4. Widgets

An widget is a lightweight, single-purpose application that offers a summary or limited view of a larger application or dashboard. In OWF, a widget is a global description for a piece of Web content that can be configured by the user and displayed within an dashboard.

2.4.1. Widget toolbar

Users can access their widgets from the widget toolbar by clicking the widgets button  in the toolbar. Once open, the Widget Toolbar appears at the left of the screen.

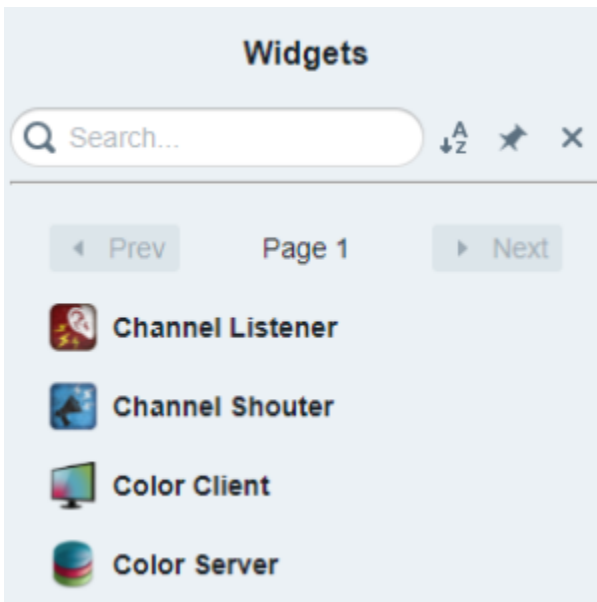




Figure 5. Widget toolbar

Adding a widget to a dashboard

To start a widget in a dashboard:

1. Open the widget toolbar by clicking the widgets button  in the toolbar.
2. Start a widget using one of the following methods:
 - Click the widget to immediately add it to the active dashboard.

- Drag the widget from the widget Toolbar into the active dashboard.
 - i. The widget toolbar disappears revealing the active dashboard.
 - ii. Drop the widget in the desired location of the dashboard.
 - iii. The widget toolbar reappears after the widget has been added.
- 3. Repeat this action to open another widget.
- 4. When finished, close the widget toolbar by clicking the  in the upper-right corner.

2.5. Intents

Intents are the instructions for carrying out an widget's intentions. One widget requests an action (think of actions as verbs like view, share, edit, etc.), then another widget receives that request and performs the action. Intents build on OWF's publish / subscribe feature by allowing users to choose the widget(s) that will use data. This binding capability enables two widgets to share data in a way that improves their function.

For example, the NYSE widget charts data about the stock exchange. Some users may want to view that data as a Web page. This is possible if the NYSE widget has an Intent that tells it to send data to widgets that display data in a Web format.



Widgets may have multiple intents associated with them. Users cannot create widget intents. Administrators and developers (logged in as administrators) add widget intents through the OWF administrator interface. Developers also add the intents through widget descriptor URLs.

2.5.1. How to Use Intents

When a widget sends an intent request, a pop-up window appears displaying all of the open widgets that can receive the requested intent action and data for an intended purpose (graphing, displaying, etc.).

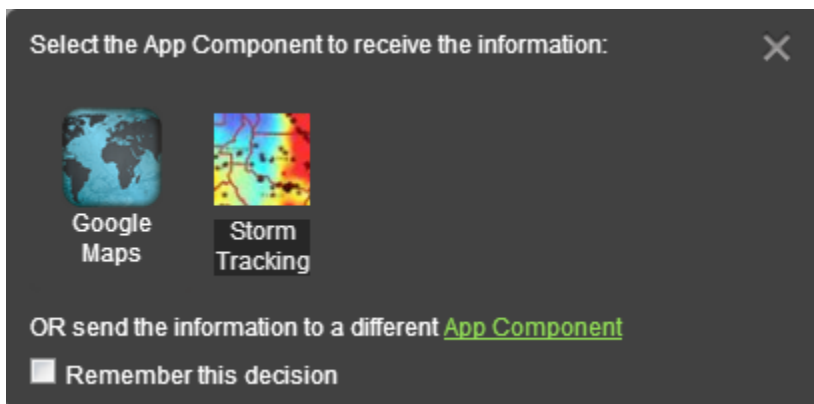


Figure 6. Widget selection dialog for intents

Select a widget to accept the requested Intent:

- Click one of the widgets displayed on the window OR
- Click the widget link to send the information to a widget that is not open on the screen:

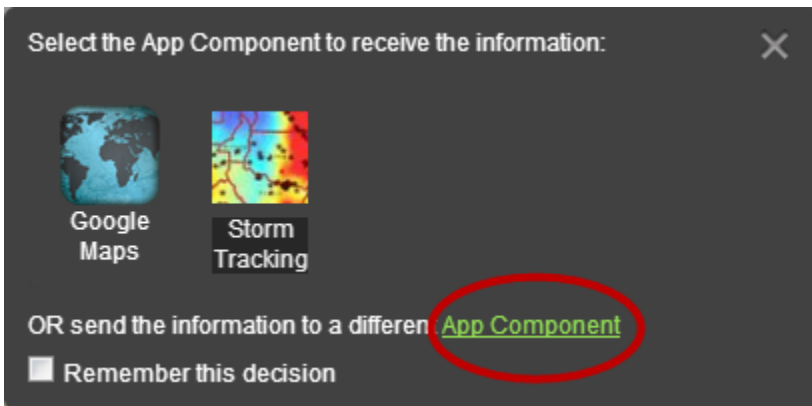


Figure 7. Select an alternate widget for the intent



Checking the "Remember" box will allow the selected widget to automatically open the requesting widget's data. This function will continue until the user breaks the connection by closing either the sending or receiving widget.

After a user selects a receiving widget, the intent data is automatically sent to and processed by the receiving widget. To place the widget on the dashboard, click or drag it from the menu into the dashboard. Once the receiving widget is placed and open in the dashboard, it will receive the sending widget intent request.

2.6. Themes

OWF currently provides a light and a dark theme.

To toggle the selected theme:

1. Click the Theme button  in the toolbar.

2.7. The Marketplace

The Ozone Marketplace, similar to a commercial app store, operates as a thin-client registry of applications and services. The Marketplace provides search and discovery functionality that enables OWF users to find, add and share useful tools including (but not limited to) widgets, dashboards, and web applications.

If OWF has been configured for is connected to one or more Marketplaces, the Marketplace button allows the user to select and launch the Marketplace within the OWF user interface.

When a single Marketplace is connected, clicking the Marketplace button immediately launch the default Marketplace.

If multiple Marketplaces are connected, clicking the Marketplace button opens the Marketplace Switcher, allowing the user to select the desired Marketplace to launch.

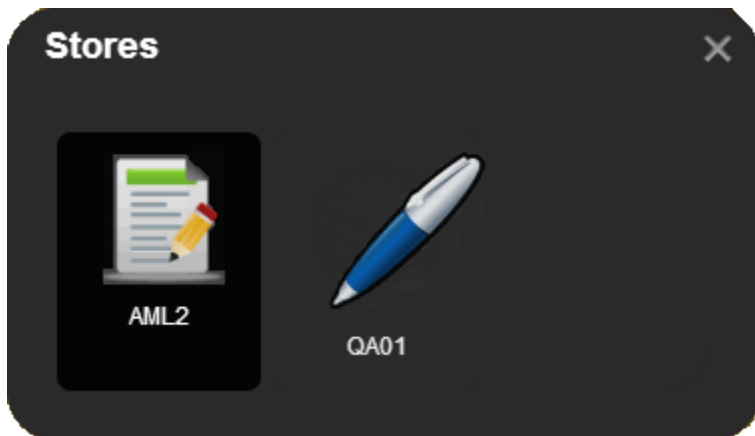


Figure 8. Marketplace selection dialog

3. The Ozone Bundle

3.1. Prerequisites

OWF is normally distributed as a zipped bundle. This bundle contains the two main components needed to deploy, run and develop widgets for the framework. They are as follows:

3.1.1. Node.js and NPM

3.1.2. Java Runtime Environment (JRE)

Running the OWF bundle requires that the Java Runtime Environment (JRE) be installed on the host machine.

OWF has been developed and fully tested using Java 8. Versions prior to or later than Java 8 are not currently supported.

The full instructions for the installation of the JRE is beyond the scope of this document. However, installation media and instructions may be obtained from the Oracle Java download page at the following location:

- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

3.2. Instructions

3.2.1. Overview



The following is a summary. Please refer to the sections below for extended details.

1. Unzip the zipped OWF bundle.
2. From a command-line, start the bundle
`owf-framework-8.X.X.X/tomcat/start.sh --dev`
or
`owf-framework-8.X.X.X/tomcat/start.bat /dev`
3. In a supported browser, navigate to: <https://localhost:8443/owf/>
4. Authenticate access to OWF by entering username `testAdmin1` and password `password`

3.2.2. Unzipping the bundle

A user can start the OWF server by unpacking the OWF bundle into the directory from where it will be run, and activating the start script. This requires the use of a ZIP utility. Detailed steps for each operation are shown below.

Example (Linux)

```
mkdir /opt/owf-framework
cp owf-framework-8.X.X.X.zip /opt/owf
cd /opt/owf
unzip owf-framework-8.X.X.X.zip
```

The following example shows how an administrator might unpack OWF from the bundle on Windows operating systems:

1. Right-click on the OWF bundle.
2. Using a ZIP utility, select "Extract to [folder name]."
If the ZIP utility does not support this feature, manually create a sub-directory and extract the ZIP contents into that directory.

3.2.3. Starting OWF

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.

3.2.3.1. Windows start script

Example:

```
C:\owf-framework-8.X.X.X\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
```

3.2.3.2. Linux start script

Example:

```
/owf-framework-8.X.X.X/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
```

3.2.4. Accessing OWF

After the backend server has finished the initialization process, and the frontend is being served, OWF can be accessed by opening a web browser and navigating to the following URL:

- <https://localhost:8443/owf/>

3.3. Authenticating to OWF

In development mode, no login is required and a default profile will be used.

In production mode, users must be authenticated by the system before they can access OWF services.

In the default installation, the user may authenticate by using the default login form using a username and password.

3.3.1. Using the login form

To login as the default administrator, use the following credentials at the login form:

- username: `testAdmin1`
- password: `password`

To login as a default unprivileged user, use the following credentials at the login form:

- username: `testUser1`
- password: `password`

3.3.2. Certificate authentication

If the system is configured for certificate-based (X509/PKI) authentication, sample user certificates are provided and must be installed in the user's browser. These certificates can be found under the tomcat/certs/ directory where the OWF bundle was deployed. See screenshots on the pages that follow for general instructions on importing certificates.



The password for the user certificate is **password**.

3.3.2.1. Installing a user certificate using Internet Explorer (IE)

1. Navigate to Tools → Internet Options → Content → Certificates → Personal.
2. Click the Import button and navigate to the /tomcat/certs directory where the OWF bundle was deployed.
3. Select the testUser1 certificate and click OK.
4. Click Next and enter **password** as the password when prompted.
5. Select a folder to house the certificate.
6. Click Finish.
7. A dialog box should display, stating that the import was successful.



#1 — In certain versions of Internet Explorer, certificate/connection failures are shown, despite a successful import of the testUser1 certificate. If this is the case, follow the directions above and select the X.509 from the drop-down and import the certificate titled ca.crt.

#2 — The default dialog filter for a particular browser may be set for CER or CRT files. If this is the case, the drop-down for file type must be set for .p12. Once .p12, is selected, the certificate will show up as being available for importing.

#3 — Some Intranets may require additional customization.

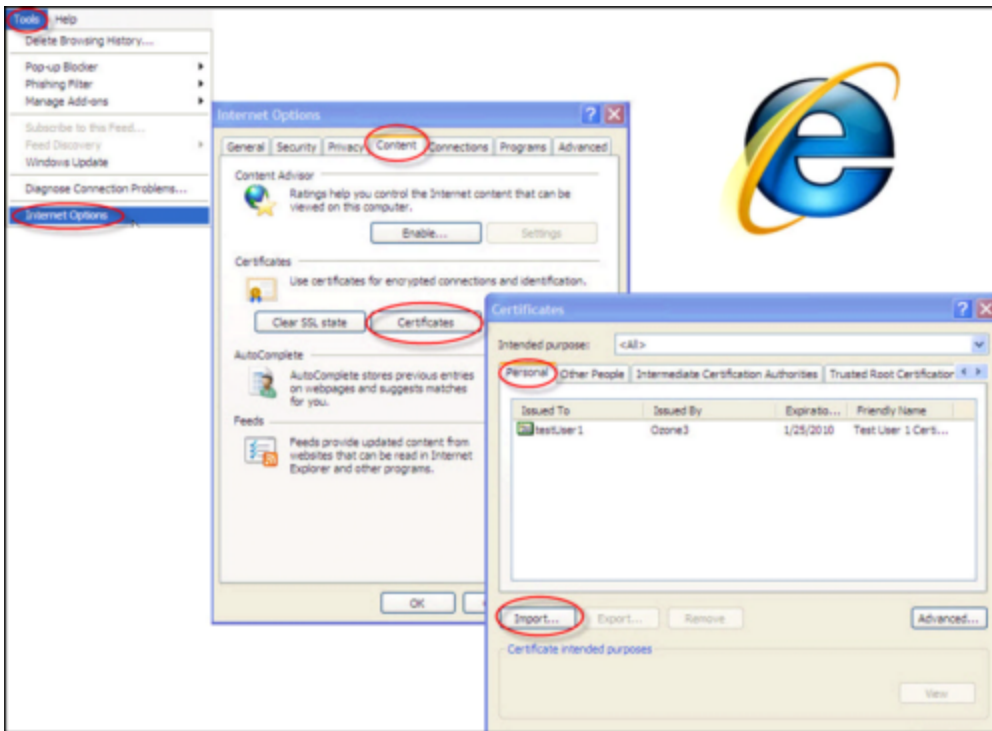


Figure 9. Internet Explorer user certificate dialog

3.3.2.2. Installing a user certificate using Firefox

1. Navigate to Tools → Options → Advanced → Encryption → View Certificate → Your Certificates.
(Newer versions: → Options → Advanced → Certificates → View Certificates → Your Certificates.)
2. Click the import button and navigate to the /tomcat/certs directory where the OWF bundle was deployed.
3. Select the testUser1 certificate, click OK.
4. Enter **password** as the password when prompted.
5. Click Finish.
6. A dialog box should display, stating that the import was successful.

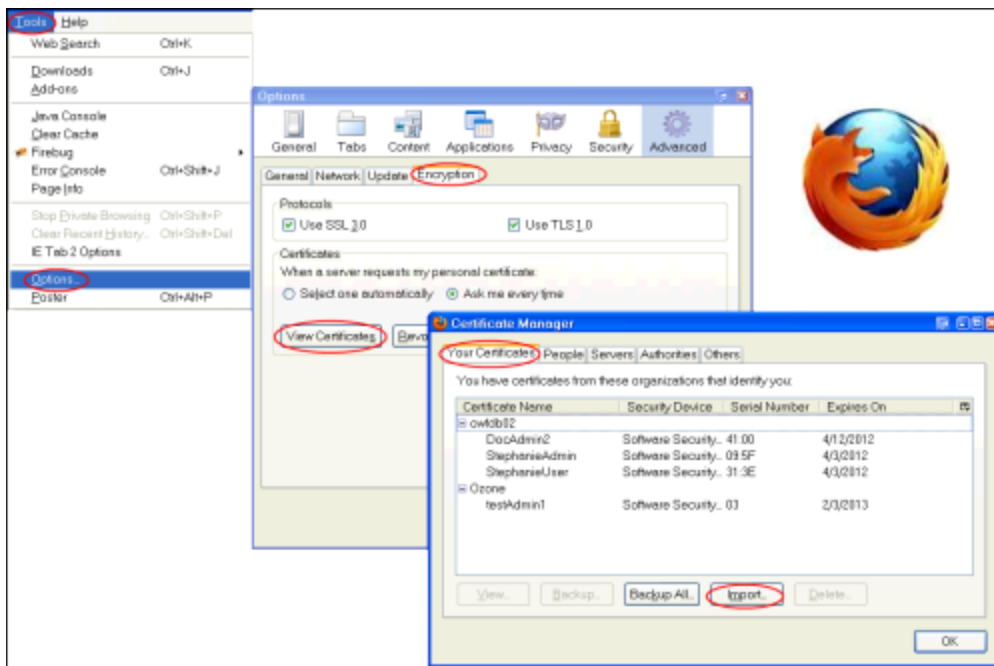


Figure 10. Firefox user certificate dialog

4. Development Build Process

4.1. Additional requirements

- Gradle (or use the included Gradle wrapper script, gradlew)
- Java JDK
- Cloned ozone GitHub repository
- Docker (optional)

4.2. Starting the frontend

Start using Node.js and npm

```
/ozone/ozone-framework-client$ npm install  
/ozone/ozone-framework-client$ gradle install  
/ozone/ozone-framework-client$ npm run start
```

OR

Start using Docker

```
./redeploy-client.sh
```

4.3. Starting the backend

Start using Gradle

```
// Navigate to ozone-framework-server directory from within the cloned ozone  
repository  
ozone$ cd ozone-framework-server  
  
// Install all required dependencies with gradle or gradlew  
ozone/ozone-framework-server$ cd ozone-classic-bom/ && gradle install && cd ..  
ozone/ozone-framework-server$ cd owf-appconfig/ && gradle install && cd ..  
ozone/ozone-framework-server$ cd owf-auditing/ && gradle install && cd ..  
ozone/ozone-framework-server$ cd owf-messaging/ && gradle install && cd ..  
ozone/ozone-framework-server$ cd owf-security/ && gradle install && cd ..  
ozone/ozone-framework-server$ cd owf-custom-tomcat/ && gradle install && cd ..  
ozone/ozone-framework-server$ cd owf-framework/ && gradle install  
  
// Start the backend server  
/ozone/ozone-framework-server/owf-framework$ gradle :buildRun
```

OR

Start using Docker

```
ozone$ ./redeploy-server.sh
```

4.3.1. Accessing OWF

After the backend server has finished the initialization process, and the frontend is being served, OWF can be accessed by opening a web browser and navigating to the following URL:

- <https://localhost:3000/>

5. Allowing Remote Access to OWF

To run OWF remotely, and NOT from a localhost environment, execute the following steps:

1. Identify a server host name.
2. Generate a server certificate.
3. Install the server certificate.
4. Modify configuration files.

5.1. Identifying a Server Name

The server host name can be chosen arbitrarily and entered into the users' HOST files, or it can be obtained from DNS. This Quick Start Guide will refer to the selected server host name as `servername` and to OWF as `https://servername:port/owf/`.

5.2. Generate a Server Certificate

The certificates that ship with OWF are configured with a server host name of `localhost`. If the host name is changed, new certificates are required. The server certificate must reflect the selected host name.

Navigate to the `/etc/tools` folder and execute `create-certificates.bat` or `.sh`, depending on the operating system in use.



Once this is done, the default user p12 certificates (`testUser1` and `testAdmin1`) will no longer be compatible. To correct this, create new user certificates using the `create-certificates` script.

Follow the prompts on screen and create the necessary certificates for the installation.

5.2.1. Install the Server Certificate

The OWF start script, located at `ozone/tomcat/bin/setenv.bat` (or `setenv.sh`) must be edited to point to the new keystore (defined while answering the prompts discussed in the previous section) file found in the `setenv` script.

Update the key store file name to reflect the new certificate.

Example: setenv.sh

```
set CATALINA_OPTS=-
Djavax.net.ssl.trustStore="%CATALINA_HOME%\certs\servername.jks"
-Djavax.net.ssl.keyStore="%CATALINA_HOME%\certs\servername.jks"
-Djavax.net.ssl.keyStorePassword=changeit
-Djavax.net.ssl.trustStorePassword=changeit server
-Xmx1024m -Xms512m -XX:PermSize=128m -XX:MaxPermSize=256m %JAVA_OPTS%
```

The Tomcat configuration file, located at `ozone/tomcat/conf/server.xml`, must also be edited to point to the new keystore file. This section can be found below the "Define a SSL..." section of the XML file:

Example: server.xml

```
<Connector port="8443"
  protocol="HTTP/1.1"
  scheme="https"
  secure="true"
  SSLEnabled="true"
  maxThreads="150"
  keystoreFile="certs/servername.jks"
  keystorePass="changeit"
  truststoreFile="certs/servername.jks"
  truststorePass="changeit"
  clientAuth="want"
  sslProtocol="TLS" />
```

5.3. Modify the Externalized Configuration Files

In order to access OWF from remote computers, externalized configuration files must point to the correct location.

Find the `owf.server.backendUrl` and `frontendUrl` properties in the `application.yml` file, and update the URLs with the new server host name.

Example: application.yml

```
owf:
  server:
    backendUrl: "https://servername:8443/owf"
    frontendUrl: "https://servername:8443/owf"
```

Glossary

Accordion (layout)

Display widgets in equal, horizontal panes that do not scroll (each individual widget may scroll using its own scroll bar).

Affiliated Store

A store that another organization uses for their system. When a local store is connected to an affiliated store, users in the local store can search for and add listings from the affiliated store (assuming the user has proper authentication for the affiliated store).

App

Deprecated term for a Stack.

App Component

Deprecated term for a widget.

Dashboard

An organized collection of widgets with a customizable layout.

Filters

A feature used to reduce the number of search results by type or category.

Fit (layout)

Allows a user to place a single widget on the screen.

Help

Repository of instructional guides and video tutorials.

Intent

Instructions for carrying out a widget's intentions.

Listing

Any software dashboard or widget that a user enters into the Store is called a "Listing." Listings can be a various types of Web content.

Marketplace

A searchable catalog of shared listings of widgets and dashboards (also referred to as the Store).

OWF

Abbreviation for Ozone Widget Framework.

Pages

Deprecated term for a dashboard.

Portal (layout)

A column-oriented layout that organizes widgets of varying heights. Each new widget loads above the first one on the screen. The user drags a dividing bar to specify widget's height. The widgets and the Ozone window scroll.

Required Listings

An association between Listings. *Example: if Listing A needs Listing B to function, Listing B is a Required Listing.*

Stack

A collection of Dashboards (pages). Allows administrators and users to group Dashboards into folder-like collections that allow for easy transition from one to another.

Store

Commonly used term for the Ozone Marketplace.

Tabbed (layout)

Display one widget per screen, with tabs the top of the screen to switch from one widget to another.

Toolbar

The navigation bar at the top of the application. It links to a user's stacks, widgets, the Store, online Help and options from the drop-down User Menu.

User

A person signed into the Ozone application, usually referring to a person without administrative privileges.

Widget

A light-weight, single-purpose Web application that offers a summary or limited view of a larger Web application and may be configured by the user and displayed within a Dashboard.

Widget Menu

The Widgets Menu displays all available widgets. Use this feature to start or add widgets to a dashboard.