OZONE Widget Framework

Build Instructions

December 21, 2012

Publication/Revision History

Release	Date
Revised – OWF 7	December 21, 2012
Initial Document – OWF 6	September 28, 2012

Contents

Tables

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 (`owf-server.war` + `cas-server.war` + `owf-samples` + Tomcat) `ZIP` file as well as the release version.

- **OWF Security JAR** – A new version of the `owf-security JAR` file.

- **Build-Base** – A common set of files used by all the other OZONE project builds. Its role is described below in section *2: Build-Base*.

1.2   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 Ant 1.7.1, Ant Contrib 1.0b3 and Grails 1.3.7. 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.

1.2.1   Download and Install Applications

OWF Development requires the use of the Java Development Kit (JDK), Apache Ant, Apache Ant Contrib, Ruby, RubyGems, Compass and Sass, as well as Groovy. The following table lists the versions of each tool required by the last few OWF and Marketplace releases.

Table 1: OWF Development Tools and Version Numbers

Application	JDK	ANT	Ant Contrib	GRAILS	RUBY	RubyGem*	COMPASS*	SASS*	GROOVY
OWF 4	1.6	1.7	-	1.3.7	1.8.7	1.8.16	0.11.3	3.1.3	-
OWF 5	1.6	1.8	-	1.3.7	1.8.7	1.8.16	0.11.3	3.1.3	-
OWF 6	1.6	1.8	1.0b3	1.3.7	1.9.2	1.8.16	0.11.3	3.1.3	1.8.8
OWF 7	1.6	1.8.3	1.0b3	1.3.7	1.9.2	1.8.16	0.11.3	3.1.3	1.8.8

*RubyGem is provided by the Ruby Installation. The Compass and Sass configuration is described in section 1.2.3: Configuring Ruby Gems (Compass and Sass).

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 2: Tool Provider Websites

Application	LOCATION
JDK	http://www.oracle.com/technetwork/java/javase/downloads/index.html
ANT	http://projects.apache.org/projects/ant.html
Grails	http://www.grails.org/
Groovy	http://groovy.codehaus.org/
Ruby	http://www.ruby-lang.org/en/

1.2.2   Setting Environment Variables

The build environment uses Apache Ant for most tasks. For Ant to run properly, it should discover the locations of the other supporting tools. This is accomplished by setting appropriate environment variables describing the install directory of each tool. On Windows 7, the following steps will set the **ANT_HOME** variable, assuming an install location of `C:\Apache_Ant\apache-ant-1.8.3`.

1) Go to the Start menu and select Control Panel.

2) Click the System icon and select the Advanced System Settings link in the resulting window.

3) Click on the Advanced tab in the resulting pop-up and then select the Environment Variables button. For the following steps, if Administrator rights are available, use the System Variables section. If not, use the User variables section.

4) Create/Edit a variable named `ANT_HOME`. Set its value to `C:\Apache_Ant\apache-ant-1.8.3`. Save the new variable. On your system, the value should be the location where Ant was actually installed.

5) Create/Edit the Path variable. If the Path variable already contains the value `%ANT_HOME%/bin`, click Cancel. If the Path doesn't contain the variable, append "`;%ANT_HOME%/bin`" (don't include the quotes) to the end of the Path variable. Click OK until the system properties dialog is closed.

6) Download and install the Ant Contrib ZIP file in a Windows environment:

    a. Download **ant-contrib-1.0b3-bin.zip** from http://sourceforge.net/projects/ant-contrib/files/ant-contrib/1.0b3/.

    b. Unzip the contents of the file on the hard drive of the local machine. The result should be a folder named **ant-contrib** on the hard drive.

    c. Copy **ant-contrib-1.0b3.jar** to the lib directory of the local Ant installation.

This method should be repeated to set a **JAVA_HOME**, **ANT_HOME**, **GRAILS_HOME**, **GROOVY_HOME** and **RUBY_HOME** environment variable for each of their installation folders, respectively.

1.2.3 Configuring Ruby Gems (Compass and Sass)

Configuring the required Ruby gems requires completion of the previous sections. Assuming a correct Ruby installation and Ruby being available on the Path variable, the following steps will install Compass and Sass:

1) Open a new Command Prompt window and enter the following:

```
gem install compass –version 0.11.3
```

2) There should be a response similar to the following:

```
C:\Users\myusername>gem install compass -version 0.11.3
Fetching: sass-3.1.15.gem (100%)
Fetching: chunky_png-1.2.5.gem (100%)
Fetching: fssm-0.2.9.gem (100%)
Fetching: compass-0.11.3.gem (100%)
Successfully installed sass-3.1.15
Successfully installed chunky_png-1.2.5
Successfully installed fssm-0.2.9
Successfully installed compass-0.11.3
4 gems installed
Installing ri documentation for sass-3.1.15...
Installing ri documentation for chunky_png-1.2.5...
Installing ri documentation for fssm-0.2.9...
Installing ri documentation for compass-0.11.3...
Installing RDoc documentation for sass-3.1.15...
Installing RDoc documentation for chunky_png-1.2.5...
Installing RDoc documentation for fssm-0.2.9...
Installing RDoc documentation for compass-0.11.3...
C:\Users\ myusername>
```

3) Then run the following command:

```
gem install sass–version 3.1.3
```

4) There should be a response similar to:

```
C:\Users\myusername>gem install sass --version 3.1.3
Fetching: sass-3.1.3.gem (100%)
Successfully installed sass-3.1.3
1 gem installed
Installing ri documentation for sass-3.1.3...
Installing RDoc documentation for sass-3.1.3...
C:\Users\myusername>
```

5) Ensure that the correct Ruby, Sass and Compass versions are installed.
6) From a Command Prompt window, run the following command:

```
sass –v
```

There should be a response that Sass 3.1.3 (Brainy Betty) is being used.

If a different version is running or a Sass error is displayed (e.g., "no such file to load"), execute the following steps to correct the Sass install:

1) In the Command Prompt window enter:

```
gem uninstall sass
```

Then choose which version to uninstall.

```
C:\Users\myusername>gem uninstall sass
Select gem to uninstall:
 1. sass-3.1.3
 2. sass-3.1.15
 3. All versions
> 2
Remove executables:
Scss in addition to the gem? [Yn]  y
Removing scss
Successfully uninstalled sass-3.1.15
```

2) Run the sass -v again; see the example below:

```
C:\Users\myusername>sass -v
Sass 3.1.3 (Brainy Betty)
C:\Users\myusername>
```

1.2.4   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:\Users\myusername>java –version
java version "1.6.0_32-ea"
Java(TM) SE Runtime Environment (build 1.6.0_32-ea-b03)
Java HotSpot(TM) 64-Bit Server VM (build 20.7-b02, mixed mode)
```

2) Enter ant –version

```
C:\Users\myusername>ant –version
Apache Ant(TM) version 1.8.3 compiled on February 26 2012
```

3) Enter grails –version

```
C:\Users\myusername>grails -version
Welcome to Grails 1.3.7 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: C:\Grails\grails-1.3.7
```

4) Enter groovy –version

```
C:\Users\myusername>groovy –version
Groovy Version: 1.8.8 JVM: 1.6.0_32
```

5) Enter ruby –v

```
C:\Users\myusername>ruby –v
ruby 1.9.2p290 (2011-07-09) [i386-mingw32]
```

6) Enter gem –v

```
C:\Users\myusername>gem –v
1.8.16
```

7) Enter sass –v

```
C:\Users\myusername>sass –v
Sass 3.1.3 (Brainy Betty)
```

8) Enter compass –v

```
C:\Users\myusername>compass -v
Compass 0.11.3 (Antares)
Copyright (c) 2008-2012 Chris Eppstein
Released under the MIT License. Compass is charityware.
Please make a tax deductable donation for a worthy cause: http://umdf.org/compass
```

2 Build-Base

All of the Ozone project build files use Ivy (**http://ant.apache.org/ivy**, version 2.1.0) for their transitive dependency management as well as a set of Subversion (SVN) Ant tasks for executing release tasks. Rather than have each project contain duplicate build information pertaining to the installation and configuration of Ivy and the SVN Ant tasks, the build-base directory in the Ozone project trunk contains the build tasks and files that are common across all of the Ozone projects. Build-base doesn't need to be checked out from SVN despite the other projects dependence on it. The Ozone Ivy SVN repository contains a zipped and versioned instance of build-base and each Ozone project treats build-base as a dependency. The first time an Ozone project is checked out from the GOSS repo, the Ant task 'init-build' needs to be run from within the project directory. This task does the following:

1) Retrieves the correct version of the **build-base.zip** file from the Ivy repo and unzips it into the project directory; This creates a sub-directory named **build-base-<version>**.

2) Runs the '**init-ivy**' task which is located in the **build-base.xml** file in the newly installed build-base directory. This task installs the correct version of Ivy as well as the necessary JAR files for the SVN Ant tasks.

Once '**init-build**' runs, the developer can run the project-specific build tasks as described below:

2.1 OWF and Server Builds

If changes are being made to the server code, the developer does not normally need to run build tasks to test them. Instead, use the standard grails scripts to run or test the server code. Use the Ant build when there is the need to build a server bundle, i.e. package the **WAR** files, along with a Tomcat instance into a **ZIP** file that can then be distributed to an end user. To build a server bundle, open a Command Prompt window and '**cd**' into the server's base directory. Verify that the '**init-build**' task ran, as described in the previous section:

```
C:\owf-server>ant init-build
```

To build the bundle, type the following on the command line and press Enter:

```
C:\owf-server>ant bundle
```

The bundle task will do a clean and build of the server code (retrieving any dependencies), run the server tests (both unit and integration) and then build the zipped bundle. The results of the build are written to the staging directory within the server directory. The staging directory of a successful build should contain the zipped bundle,

as well as the unzipped contents of the bundle. The latter is provided as a convenience to test the bundle by running the **start.bat** (or **start.sh** on Linux systems) in the staging/**apache-tomcat-x.x.x** directory. This will start the Tomcat server which is initially configured to load the Ozone server and the CAS server **WAR** files.

2.1.1 Additional Build Command Line Options

When running the build, use any (or all) of the following command line parameters:

- **-logfile build.log** – This will redirect the output of the build to the specified log file. This can be useful when there are problems with the build because the output is often too verbose to be completely captured by the Command Prompt window.

- **-Dno-test=true** – This will prevent the grails unit and integration tests from running. Often, they've already run. Using this option will speed up the build.

- **-Dno-jsdoc=true** – This will prevent the generation of the JavaScript documentation. Generating the documentation is time consuming and slows the build progress.

- **-Dgroovy_all** – This property can be used to manually set the location of **groovy-all.jar**. This property needs to be set correctly when building to the **GROOVY_HOME** environment variable. The purpose of this property is to allow building on systems where Groovy was installed in such a way that the **GROOVY_HOME** variable is not set, such as Linux systems that installed Groovy using a centralized package manager. See below for an example:

```
ant bundle -Dgroovy_all=/usr/share/java/groovy-all.jar
```

2.1.2 Building Offline

When building the Marketplace or OWF projects on a network that is not externally connected, try the following steps:

1) Download the Ozone Ivy repo from:
   https://owfgoss.org/svn/repos/ozone/ivy-repo
   This is a very large (~2GB) download.

2) Modify **application.properties** to include the following property:
   offline_repo=c:/path/to/ivy-repo/no-namespace
   This provides the Grails pointer to the offline repo. The value must contain the '**no-namespace**' portion of the path.

3) Add a flag to the Ant command line: `-Ddisconnected=true`.
   This switches the build process to use ivy properties from `build-base/ivysettings_offline.xml`.

4) If it is not already there, add a version of the `ivysvnresolver*.jar` to the Ant classpath. The version found in the following location can be copied and placed under the `ant/lib` directory: `ivy-repo/no-namespace/org.apache/ant-custom/zips/ant-custom-1.7.1.zip/lib`.
   This provides a SvnResolver implementation for Ivy.

5) Modify file `build-base/ivysettings_offline.xml`: if the token `${env.OFFLINE_REPO}` exists, replace it with `${offline_repo}`
   The `ivy:settings` task has visibility to the declared Ant properties, but not the environmental properties that the parent project has within its scope.

6) [Optional] Run any grails command explicitly by adding the command line flag: `-DOFFLINE_REPO= c:/path/to/ivy-repo/no-namespace`.

2.1.3 Troubleshooting

- Cannot connect to the OWF GOSS SVN download repository?
  - Investigate the use of an HTTP Proxy.
  - Verify the use of a valid GOSS username and password, issued as a Contributor account. The password can be reset through the "Account Management" link on the http://owfgoss.org homepage.

- Grails plugins used by the applications have their own dependencies. In some cases these are not required for the base application; e.g. In Alpha/Beta releases these may be experimental or partially built-out capabilities. If a plugin is causing issues, it may be removed by:
  - Modifying the [root]`/application.properties` file and commenting out the line beginning with '`plugins.[pluginName]`'.
  - Moving the [root]`/plugins/[pluginName]` directory out of the project.

  Note: Other references to the plugin may exist in places and this procedure may be insufficient to avoid all possible errors.

- Testing Grails compilation outside of the Ant script can be useful for diagnosing Grails-level configuration issues without executing all other aspects of the build process. To do this, run the following from a Command Prompt Window: `grails compile -DOFFLINE_REPO= c:/path/to/ivy-repo/no-namespace`.

2.2 Ozone Security Jar

If a new release of the security **JAR** file is needed, take the following steps:

1) Go to the **JAR**'s project directory (…**\commons\ozone-security)** and edit the **build.properties** file, incrementing the **JAR**'s version number.

2) From the project's command line, run the following two commands:

   `C:\commons\ozone-security>ant init-build -f owf-build.xml`

   `C:\commons\ozone-security>ant pre-release -f owf-build.xml`

   Because the security project gets distributed with a project **ZIP** file, the normal '**build.xml**' is only used to build the **JAR** file. Use the '**owf-build.xml**' file to build new release versions of the **JAR** files. The 'pre-release' target does a full build of the project and then publishes the project's artifact(s) to the local pre-release repo (**.ivy2/local**).

3) To test the pre-release version of the **JAR**, go to the directory of the OWF or Marketplace server project and edit the server **application.properties** file to update the version number of the dependency ('**owf.security.rev**' and '**mp.security.rev**' in the OWF and Marketplace projects, respectively).

4) Build the server bundle (Ant bundle). Go to the **staging/apache-tomcat-X.X.X/webapps/<server>.war**. Navigate the server **WAR** file and go to the **WEB-INF/libs** directory and verify that the new version of the security **JAR** is there. It is best to test the server app as well.

5) If there are problems with the new security **JAR**, make necessary modifications and repeat steps 2 through 4.

6) Once satisfied with the new **JAR**, go to the command line and run the release target.

   `C:\commons\ozone-security>ant release -f owf-build.xml`

*Note: A Subversion 1.6 client must be used. Currently the **JARS** that Ant's Subversion tasks use do not work with code that has been checked out with a 1.7 Subversion client. If a 1.6 client is not used, a vague exception will be thrown inside the Ant SVN tasks.*

2.3 CAS Server War

To make a new release of the CAS server **WAR** file, do the following:

1) Go the CAS server project directory and edit the **build.properties** file, incrementing the version number (**cas.server.rev**).

2) From the project's command line, run the following two commands:

```
C:\ozone\cas-server>ant init-build

C:\ozone\cas-server>ant pre-release
```

The 'pre-release' target does a full build of the CAS server project and then publishes the project's artifact(s) to the local pre-release repo (**.ivy2/local**).

3) To test the pre-release version of the **WAR**, go to the directory of a server project that uses this dependency (in this case **owf-server**) and edit the server **application.properties** file to update the version number of the CAS server (**cas.server.rev**).

4) Build the server bundle (**ant bundle**). Go to the **staging/apache-tomcat-X.X.X/webapps** directory. It should contain the CAS server **WAR** file. The file name should include the new CAS server version. Test the changes by running the server application.

5) If there are problems with the new **WAR**, make edits and repeat steps 2 through 4.

6) Once satisfied with the new **JAR**, go to the command line and run the release target:

```
C:\ozone\cas-server>ant release
```

2.4   Tomcat-OWF-Custom

To make a new release of the **tomcat-owf-custom.zip** file:

1) Go the Tomcat server project directory and edit the **build.properties** file, incrementing the version number (**tomcat.custom.version**).

2) From the project's command line, run the following two commands:

```
C:\ozone\tomcat-owf-custom>ant init-build

C:\ozone\tomcat-owf-custom>ant pre-release
```

The 'pre-release' target does a full build of the **tomcat-owf-custom ZIP** file and then publishes it to the *local* pre-release repo (**.ivy2/local**).

3) To test the pre-release version of the **ZIP** file, go to the directory of the server project that uses this dependency (in this case **owf-server**) and edit the project's **application.properties** file to update the version number of the CAS server (**tomcat.custom.rev**).

4) Build the server bundle (**ant bundle**). Go to the **staging/apache-tomcat-X.X.X** directory and verify that the new Tomcat changes are present.

5) If there are problems with the new Tomcat, make edits and repeat steps 2 through 4.

6) Once satisfied with the new Tomcat, go to the command line and run the release target:

`C:\ozone\tomcat-owf-custom>ant release`

Appendix A   Contact Information

A.1   Discussion Group

The Ozone Developers Discussion Group is hosted through Google Groups at http://groups.google.com/group/ozone-developers. This forum is for the distribution of release announcements, Q&A related to OWF and for additional inquiries about widgets and features being developed across the user base. To access the group, request an invitation at http://groups.google.com/group/ozone-developers or contact the Community Support Team at goss-support@owfgoss.org.

A.2   Additional POCs

For information about the Ozone Widget Framework or access to its resources, please email goss-support@owfgoss.org. Additional resources can be found at http://owfgoss.org.