

Building with Maven



Maven 3.0.0 or later is required to build Struts 2

First, let's review some Maven basics. Maven uses the notion of a build *life-cycle* to which plugins can attach. Plugins are similar to Ant tasks. When a Maven build is invoked, we specify a point in the life-cycle up to which the build *should* proceed. The *compile* phase comes before *test*, and *test* comes before *package*, and *package* comes before *install*. Once we have Maven setup, we can invoke the Struts build, and specify which phase the build should use.

Installing

The *install* phase builds up the project ("package"), and installs any JARs it needs into your local repository (e.g. `~/m2/repository`). Once installed, the JARs can be used by any other Maven project you build. The *install* phase is the **default** phase if none is specified.

To run a basic install, change to the root of the source distribution, and enter

Go!

```
> mvn
```

That's it! Maven will download any dependencies the build needs; run all unit tests; package up the JARs; then install the new JARs locally. For your convenience, copies of the JARs can be found in the *target* directories of each module. For example, after the build, the main JAR can found at `core/target/struts2-core-2.0-SNAPSHOT.jar`.



It's supposed to be automatic, but you might still have to press the button

Sometimes, licensing restrictions prevent Maven from downloading all the JARs that a build might need. For example, JavaMail and Activation, can only be downloaded from Sun. When this happens, Maven will display a helpful message that explains how to install these JARs manually. After downloading the required JAR, follow the instructions to install it to the your local repository. Once installed, the JAR is available to all your Maven builds, not just Struts.

Initial Build Successful

```
[INFO] Reactor Summary:
[INFO]
[INFO] Struts 2 ..... SUCCESS [6.016s]
[INFO] XWork: Core ..... SUCCESS [1:08.086s]
[INFO] Struts 2 Core ..... SUCCESS [2:42.604s]
[INFO] Struts Plugins ..... SUCCESS [0.944s]
[INFO] Struts 2 Spring Plugin ..... SUCCESS [13.523s]
[INFO] Struts 2 Convention Plugin ..... SUCCESS [16.473s]
[INFO] Struts 2 JUnit Plugin ..... SUCCESS [13.325s]
[INFO] Webapps ..... SUCCESS [1.027s]
[INFO] Blank Webapp ..... SUCCESS [9.572s]
[INFO] JBoss Blank Webapp ..... SUCCESS [7.814s]
[INFO] Mail Reader Webapp ..... SUCCESS [7.028s]
[INFO] Struts 2 Portlet Plugin ..... SUCCESS [20.995s]
[INFO] Struts 2 DWR Plugin ..... SUCCESS [4.432s]
[INFO] Struts 2 Tiles Plugin ..... SUCCESS [4.451s]
[INFO] Struts 2 Portlet Tiles Plugin ..... SUCCESS [3.741s]
[INFO] Portlet Webapp ..... SUCCESS [28.480s]
[INFO] Struts 2 Struts 1 Plugin ..... SUCCESS [8.353s]
[INFO] Struts 2 Dojo Plugin ..... SUCCESS [30.181s]
[INFO] Struts 2 JSF Plugin ..... SUCCESS [6.000s]
[INFO] Struts 2 Configuration Browser Plugin ..... SUCCESS [7.531s]
[INFO] Struts 2 Sitemesh Plugin ..... SUCCESS [6.159s]
[INFO] Struts 2 JSON Plugin ..... SUCCESS [16.944s]
[INFO] Showcase Webapp ..... SUCCESS [20.303s]
[INFO] Struts 2 REST Plugin ..... SUCCESS [10.015s]
[INFO] Struts 2 Rest Showcase Example ..... SUCCESS [6.928s]
[INFO] Struts 2 Codebehind Plugin ..... SUCCESS [10.394s]
[INFO] Struts 2 Java Templates Plugin ..... SUCCESS [9.633s]
[INFO] Struts 2 Jasper Reports Plugin ..... SUCCESS [5.920s]
[INFO] Struts 2 JFreeChart Plugin ..... SUCCESS [18.296s]
[INFO] Struts 2 Pell Multipart Plugin ..... SUCCESS [3.891s]
[INFO] Struts 2 Plexus Plugin ..... SUCCESS [4.715s]
[INFO] Struts 2 Sitegraph Plugin ..... SUCCESS [8.009s]
[INFO] Struts 2 TestNG Plugin ..... SUCCESS [6.012s]
[INFO] Struts 2 Oval Plugin ..... SUCCESS [9.629s]
[INFO] Struts 2 OSGi Plugin ..... SUCCESS [8.118s]
[INFO] Struts 2 Embedded JSP Plugin ..... SUCCESS [30.524s]
[INFO] Struts 2 GXP Plugin ..... SUCCESS [4.878s]
[INFO] Struts 2 CDI Plugin ..... SUCCESS [7.620s]
[INFO] Struts 2 Tiles 3 Plugin ..... SUCCESS [5.132s]
[INFO] Struts OSGi Bundles ..... SUCCESS [0.254s]
[INFO] Struts 2 OSGi Admin Bundle ..... SUCCESS [5.204s]
[INFO] Struts 2 OSGi Demo Bundle ..... SUCCESS [4.582s]
[INFO] Struts 2 Maven Archetypes ..... SUCCESS [0.198s]
[INFO] Struts 2 Archetypes - Angular JS ..... SUCCESS [10.509s]
[INFO] Struts 2 Archetypes - Blank ..... SUCCESS [1.440s]
[INFO] Struts 2 Archetypes - Blank Convention ..... SUCCESS [1.324s]
[INFO] Struts 2 Archetypes - Database Portlet ..... SUCCESS [0.990s]
[INFO] Struts 2 Archetypes - Plugin ..... SUCCESS [0.923s]
[INFO] Struts 2 Archetypes - Portlet ..... SUCCESS [0.888s]
[INFO] Struts 2 Archetypes - Starter ..... SUCCESS [1.992s]
[INFO] Struts 2 Assembly ..... SUCCESS [1:37.556s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12:23.728s
[INFO] Finished at: Wed Oct 30 17:26:27 CET 2013
[INFO] Final Memory: 57M/945M
[INFO] -----
```

Other phases

There are other phases that can be useful when working with Maven. The *package* phase will just JAR (or WAR) the modules. The *test* phase will only execute unit-tests. The *compile* phase will only build the source-code (but not the test sources). And the *clean* phase will remove all artifacts, typically the entire *target* directory.

Building IDE project files

Maven has a great feature that will allow you to build up your IDEA or Eclipse project files based on the project build structure. Using the Maven project files helps keep developers in-sync and efficient. The IDEA project files are pre-configured to define in the Run/Debug menu many common tasks, such as "execute all tests", "launch the showcase sample app", and so fort.

IDEA by JetBrains

The Maven IDEA plugin creates IDEA project and modules files, based on the Maven build configuraton (pom.xml).

First, be sure you have the latest IDEA plugin for Maven.

Get the latest plugin!

```
mvn -DconnectionUrl=scm:svn:http://svn.apache.org/repos/asf/maven/plugins/trunk/maven-idea-plugin \
    -Dgoals=install \
    scm:bootstrap
```

Most often, we use Maven to build projects, but it can also run plugins directly. To build an IDEA project file for Struts 2, including all the modules, run

Generate the project files

```
> mvn idea:idea -Papps,extras,thirdparty,xwork
```

Maven will generate *struts-parent.ipr*, *struts-parent.iws*, and an *iml* file for each module listed. Open up *struts-parent.ipr* in IDEA, and you should be good to go.

If you ever need to rebuild your projects, running the `idea:idea` command again will update your files without overriding any information that doesn't conflict. If you do want to overwrite the project files, specify the `overwrite` parameter as `true`.

Starting over

```
> mvn idea:idea -Doverwrite=true -Papps,extras,thirdparty,plugins,xwork
```

If you only need to rebuild some of the modules, adjust the list of "profiles".



Clearing the cache

If you find the `xwork` module causing you problems, such as displaying as "XWork" when it should be named "xwork", the problem is likely to be within IDEA. Try clearing out your IDEA system cache and then run it again.

Eclipse

For Eclipse, try

Generating Eclipse

```
> mvn -Pthirdparty,plugins,xwork eclipse:eclipse
```



Feedback Wanted

Many Struts 2 developers use IDEA, and the Eclipse project files are not as well-tested or featureful as the IDEA versions. But as far as we know, they work!

IMPORTANT: Running the Sample Applications from Eclipse or non-IDEA IDEs

Currently the sample applications can be deployed using the [Maven 2 Jetty plugin](#). For example, you can run the showcase example application from the command line:

Running Showcase in Linux

```
cd apps/showcase
mvn jetty:run
```

You could also use one of the Maven IDE plugins to run the plugin or simply execute the `mvn` or `mvn.bat` file as an external application within your IDE. A good tutorial on how to use the Maven 2 jetty plugin along with your IDE, including JPDA debugging support, can be found at the [Apache Wicket documentation](#).

Maven Tips

A few helpful tips for using Maven are provided:

Skip exporting pages from Confluence

The whole Struts2 documentation resides on Confluence and during normal build process is exported with SiteExporter tool locally to html files. To perform that, you must have an account on Confluence. In most cases you don't need to export the whole Confluence, so you can skip this, just type command:

```
mvn clean install -DskipWiki
```

Skip creating assemblies

During normal development cycle you don't need to create assembly packages which consist of all the libraries, example apps and docs - you can skip them as well:

```
mvn clean install -DskipAssembly
```

Offline mode

If you are disconnected from the Internet or simply wish to make your build faster, pass in the `-o` argument and Maven won't check for new modules to download.

Offline Mode

```
mvn -o
```

OutOfMemoryError

If you get an `OutOfMemoryError` while running the tests:

Memory Settings

```
// Bash
export MAVEN_OPTS=-Xmx512m

// Windows
set MAVEN_OPTS=-Xmx512m
```

Skipping test execution

Although this shouldn't ever happen, sometimes tests do fail and you need to build the framework anyway. If there's a problem, you can pass in the skip tests parameter.

Skip Tests

```
mvn -Dmaven.test.skip=true
```

Of course, if you find tests are failing, please submit a patch to fix them!

Debugging build failures

Struts 2 depends on the current snapshot artifact of XWork 2. Unfortunately, if XWork 2 is modified in a significant way, the Struts build doesn't check for and retrieve a new version of the XWork snapshot jar, resulting in compilation or unit test failures.

The solution is to clear out any stale XWork 2 jars in your local Maven repository, usually by removing the directory `~/.m2/repository/opensymphony`. This will force the Struts Maven build to pull down a fresh copy of XWork and hopefully resolve your problem.

Mirrors

The default mirror for Maven builds ("ibiblio") can be slow and unreliable. Maven lets you specify [alternative mirrors](#) so that you don't have to depend on ibiblio for everything.

You can add new mirrors through the Settings file (`~/.m2/settings.xml`).

Mirror, Mirror

```
<settings>

  <mirrors>
    <mirror>
      <id>dotsrc</id>
      <url>http://mirrors.dotsrc.org/maven2</url>
      <mirrorOf>central</mirrorOf>
    </mirror>
  </mirrors>

</settings>
```



Sometimes the alternative mirrors have problems too or aren't updated at the same frequency as the main mirror. If you have trouble building, try commenting out the alternative mirror.

First time building

In some cases it has been seen that Maven will complain if a module doesn't exist, even though it is part of the current build. Often, the missing module turns up when executing `mvn package`. A simple fix for this is to run `mvn install` instead. If you have to do this, it will probably only be a one time thing.

Next: [Creating and Signing a Distribution](#)