

Runs a Grails application as a JAR file with an embedded Tomcat or Jetty server

Grails Standalone Plugin - Reference Documentation

Authors: Burt Beckwith

Version: 1.2

Table of Contents

- 1** Introduction to the Standalone Plugin
- 2** Running the application

1 Introduction to the Standalone Plugin

The Standalone plugin builds a runnable JAR file with an embedded war built from your application and an embedded Tomcat 7 or Jetty 7 instance. This allows you to build a single archive that can be run on any computer with Java 5 or higher by running `java -jar standalone.jar`. This can be convenient for demos or even very lightweight installs of low-traffic Grails applications.

Release History

- May 10, 2013
 - 1.2 release
- August 16, 2012
 - 1.1.1 release
- August 6, 2012
 - 1.1 release
- August 4, 2012
 - 1.0.1 release
- July 8, 2011
 - initial 1.0 release

2 Running the application

Building the jar

The first step is to run the [build-standalone](#) script, e.g.

```
grails prod build-standalone
```

or

```
grails -Dgrails.env=demo build-standalone our_cool_demo.jar
```

If you pass the `--jetty` flag the embedded server will be Jetty instead of the default Tomcat, for example

```
grails prod build-standalone --jetty
```

or

```
grails -Dgrails.env=demo build-standalone our_cool_demo.jar --jetty
```

You can change the default embedded server to Jetty by adding this to BuildConfig.groovy:

```
grails.plugin.standalone.useJetty = true
```

If you've set Jetty to the default, you can force a Tomcat build later with the `--tomcat` flag:

```
grails build-standalone --tomcat
```

Build Configuration

There are several configuration settings that you can use to change how the jar is built; add any of these to BuildConfig.groovy to override the defaults:

Property	Default Value	Meaning
grails.plugin.standalone.ecjDependency	'org.eclipse.jdt.core.compiler:ecj:3.7.1'	the dependency config for the ECJ jar
grails.plugin.standalone.extraDependencies	none	extra dependency jars to include in the standalone jar
grails.plugin.standalone.ivyLogLevel	'warn'	the Ivy log level
grails.plugin.standalone.jettyServletApiDependency	'javax.servlet:servlet-api:2.5' 'javax.servlet:javax.servlet-api:3.0.1'	or the Jetty servlet API jar dependency
grails.plugin.standalone.jettyVersion	'7.6.0.v20120127'	the version of Jetty to use
grails.plugin.standalone.tomcatVersion	'7.0.39'	the version of Tomcat to use
grails.plugin.standalone.tomcatDependencies	'tomcat-annotations-api', 'tomcat-catalina-ant', 'tomcat-coyote', 'tomcat-juli', 'tomcat-servlet-api', use 'tomcat-util'	'tomcat-api', 'tomcat-catalina', the Tomcat jars to use
grails.plugin.standalone.tomcatEmbedDependencies	'tomcat-embed-core', 'tomcat-embed-logging-juli', 'tomcat-embed-logging-log4j'	'tomcat-embed-jasper', the Tomcat embed jars to use

Running the server

As long as the target machine has Java 5 or higher available, all you need to do next is run

```
java -jar /path/to/jar_name.jar
```

There are several arguments you can pass to customize how the application runs, using name=value syntax:

1. `context`, the context name; if not specified it defaults to "" (the "root" context)
2. `host`, the host name; if not specified it defaults to "localhost"
3. `port`, the HTTP port; if not specified it defaults to 8080
4. `httpsPort`, the HTTPS port; there is no default for this, but if specified you can also specify the keystore path and password
5. `keystorePath` or `javax.net.ssl.keyStore`, the SSL keystore path; if not specified a temporary keystore will be generated
6. `keystorePassword` or `javax.net.ssl.keyStorePassword`, the SSL keystore password; required if an existing keystore path is specified
7. `truststorePath` or `javax.net.ssl.trustStore`, the SSL truststore path
8. `trustStorePassword` or `javax.net.ssl.trustStorePassword`, the SSL truststore password; required if an existing truststore path is specified
9. `enableClientAuth`, whether to enable client auth, defaults to false
10. `workDir`, the working directory where the war file is extracted, defaults to the system temp directory
11. `enableCompression`, whether to enable compression (Tomcat only)
12. `compressableMimeTypes`, a comma separated list of MIME types for which HTTP compression may be used; defaults to the Tomcat defaults, "text/html,text/xml,text/plain"
13. `sessionTimeout`, the session timeout in minutes; defaults to 30
14. `nio` or `tomcat.nio`, whether to use NIO; defaults to true

In addition, if you specify a value that is the name of a system property (e.g. 'home.dir'), the system property value will be used.

For example running

```
java -jar /path/to/jar_name.jar
```

will start a server at `http://localhost:8080/` and

```
java -jar /path/to/jar_name.jar context=cool_demo port=9000
```

will start a server at `http://localhost:9000/cool_demo`

```
java -jar /path/to/jar_name.jar context=cool_demo port=8080 httpsPort=8443
```

will start a server at `http://localhost:8080/cool_demo` and will also support SSL at `https://localhost:8443/cool_demo`

