Plugin Guide

{scrollbar}

Apache Geronimo v2.1 is now assembled completely out of plugins including the server configuration files, and maven now builds a plugin whenever you "predeploy" a module with a geronimo plan. We'll explain the plugin system features so you can see how to specify installation activities, customization points, and when your plugin will be started, as well as how to access these features in a maven build.

Plugin system overview

A geronimo plugin consists of a classloader specification, optional classes, optional service or component configuration, and information about how to install it in Geroninimo. The classloader specification and service configuration is specified in a Geronimo plan (and possibly other plans such as a javaee spec DD or annotations). The information about how to install the plugin is provided in the **`META-INF/geronimo-plugin.xml`** file. This file includes details such as the category and description, dependency information showing what else needs to be installed with this plugin, information on files to be unpacked on installation, and configuration information showing how and when the plugin will be started. Before looking at the more complicated aspects of the plugins lets look at a simple example of **`geronimo-plugin.xml`**. Here is an example for the jetty web container:

First we see fairly obvious cataloging information:

xmlsolid <?xml version="1.0" encoding="UTF-8" standalone="yes"?> <geronimo-plugin xmlns="http://geronimo.apache.org/xml/ns/plugins-1.3" xmlns:ns2="http://geronimo.apache.org/xml/ns/attributes-1.2"> <name>Geronimo Configs :: Jetty 6</name> <category>Jetty</category> <description>Geronimo Jetty Web Server integration.</description> <url>http://geronimo.apache.org/</url> <author>The Apache Geronimo development community</author> <license osi-approved="true">The Apache Software License, Version 2.0</license>

Each **`geronimo-plugin.xml`** can specify information for many versions of the "same" plugin, so the **plugin-artifact** element specifying info for one version can occur multiple times. Here there is just one. First we see the plugin **moduleId** and the list of dependencies that will be installed if not already present.

xmlsolid <plugin-artifact> <module-id> <groupId>org.apache.geronimo.configs</groupId> <artifactId>jetty6</artifactId> <version>2.1-SNAPSHOT</version> <type>car</type> </module-id> <geronimo-version>2.1-SNAPSHOT</geronimo-version> <jvm-version>1.5</jvm-version> <dependency start="true"> <groupId>org.apache.geronimo.configs</groupId> <artifactId>j2ee-server</artifactId> <version>2.1-SNAPSHOT</version> <type>car</type> </dependency> <dependency start="true"> <groupId>org.apache.geronimo.configs</groupId> <artifactId>server-security-config</artifactId> <version>2.1-SNAPSHOT</version> <type>car</type> </dependency> <dependency start="true"> <groupId>org.apache.geronimo.configs</groupId> <artifactId>transaction</artifactId> <version>2.1-SNAPSHOT</version> <type>car</type> </dependency> <dependency start="true"> <groupId>org.apache.geronimo.modules</groupId> <artifactId>geronimo-jetty6</artifactId> <version>2.1-SNAPSHOT</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.apache.geronimo.configs</groupId> <artifactId>clustering</artifactId> <version>2.1-SNAPSHOT</version> <type>car</type> </dependency> <dependency start="true"> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.4.3</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.slf4j</groupId> <artifactId>slf4j-jcl</artifactId> <version>1.4.3</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.mortbay.jetty</groupId> <artifactId>jetty</artifactId> <version>6.1.5</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.mortbay.jetty</groupId> <artifactId>jetty-ajp</artifactId> <version>6.1.5</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.mortbay.jetty</groupId> <artifactId>jetty-sslengine</artifactId> <version>6.1.5</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.mortbay.jetty</groupId> <artifactId>jetty-util</artifactId> <version>6.1.5</version> <type>jar</type> </dependency> <dependency start="true"> <groupId>org.apache.geronimo.configs</groupId> <artifactId>webservices-common</artifactId> <version>2.1-SNAPSHOT</version> <type>car</type> </dependency>

Now we see the list of repositories this plugin is expected to be available from. We normally include the local maven repository to make developing plugins easier.

xmlsolid <source-repository>~/.m2/repository/</source-repository> <source-repository>http://repo1.maven.org/maven2/</source-repository> <source-repository>http://people.apache.org/repo/m2-snapshot-repository/</source-repository> <source-repository>http://people.apache.org/repo/m2-incubating-repository/</source-repository>

Here we see the prototype for plugin customization. The **`config.xml`** file has a section for each module or plugin it knows about and the contents of the **config-xml-content** is copied into such an element. Note the use of substitution variables such as **${ServerHostname}**.

xmlsolid <config-xml-content> <ns2:gbean name="JettyWebConnector"> <ns2:attribute name="host">$\{ServerHostname\}</ns2:attribute> <ns2:attribute name="port">$\{HTTPPort + PortOffset\}</ns2:attribute> <ns2:attribute name="redirectPort">$\{HTTPSPortPrimary + PortOffset\}</ns2:attribute> </ns2:gbean> <ns2:gbean name="JettyAJP13Connector"> <ns2:attribute name="host">$\{ServerHostname\}</ns2:attribute> <ns2:attribute name="port">$\{AJPPort + PortOffset\}</ns2:attribute> <ns2:attribute name="redirectPort">$\{HTTPSPortPrimary + PortOffset\}</ns2:attribute> </ns2:gbean> <ns2:gbean name="JettySSLConnector"> <ns2:attribute name="host">$\{ServerHostname\}</ns2:attribute> <ns2:attribute name="port">$\{HTTPSPort + PortOffset\}</ns2:attribute> </ns2:gbean> </config-xml-content>

Here we see the default values of the substitution variables. These are copied into the **`config-substitutions.properties`** file; you are expected to modify these by hand as necessary.

xmlsolid <config-substitution key="HTTPPort">8080</config-substitution> <config-substitution key="AJPPort">8009</config-substitution> <config-substitution key="HTTPSPort">8443</config-substitution> <config-substitution key="ServerHostname">0.0.0.0</config-substitution> <config-substitution key="webcontainer">JettyWebContainer</config-substitution> <config-substitution key="webcontainerName">jetty6</config-substitution>

Missing from this example is the **<artifact-alias>** element which can be used to replace one plugin by another. For instance you can switch databases by deploying e.g **postgres-system-database** and specifying **<artifact-alias key="org.apache.geronimo.configs/system-database/2.1-SNAPSHOT/car" >org.apache.geronimo.configs/postgres-system-database/2.1-SNAPSHOT/car</artifact-alias>**

xmlsolid </plugin-artifact> </geronimo-plugin>

One of the more obvious parts of Geronimo is the repository which contains jars as well as plugins. However the plugins by themselves don't do anything; we need some information about which ones to start and how to customize them in order to get a functioning server. This kind of information is normally stored in configuration files in the var/config directory such as **`config.xml`**, **`config-substitutions.properties`** and **`artifact_aliases.properties`**. There are several "servers" you can start in a normal Geronimo installation, such as the "server", the app client container, the deployer, and the jsr88 tool. The plugin system abstracts this idea of a "server instance" with a **ServerInstance** gbean that specifies the attribute store (relating to the `config.xml` and `config-substitutions.xml` files) and artifact resolver (relating to the `artifact_aliases.properties` file). So the plugin system requires that you set up **ServerInstances** for all the kinds of servers you expect to start in a Geronimo installation. For instance, the normal Geronimo setup includes these:

xmlsolid <gbean name="DefaultServer" class="org.apache.geronimo.system.plugin.ServerInstance"> <attribute name="serverName">default</attribute> <reference name="PluginAttributeStore"> <name>AttributeManager</name> </reference> <reference name="ArtifactResolver"> <name>ArtifactResolver</name> </reference> </gbean> <gbean name="Offline" class="org.apache.geronimo.system.plugin.ServerInstance"> <attribute name="serverName">offline</attribute> <reference name="PluginAttributeStore"> <name>OfflineAttributeManager</name> </reference> <reference name="ArtifactResolver"> <name>ArtifactResolver</name> </reference> </gbean> <gbean name="OfflineAttributeManager" class="org.apache.geronimo.system.configuration.LocalAttributeManager"> <reference name="ServerInfo"> <name>ServerInfo</name> </reference> <attribute name="readOnly">true</attribute> <attribute name="configFile">var/config/offline-deployer-config.xml</attribute> <attribute name="substitutionsFile">var/config/config-substitutions.properties</attribute> <attribute name="substitutionPrefix">org.apache.geronimo.config.substitution.</attribute> </gbean> <gbean name="Client" class="org.apache.geronimo.system.plugin.ServerInstance"> <attribute name="serverName">client</attribute> <reference name="PluginAttributeStore"> <name>AttributeManager</name> </reference> <reference name="ArtifactResolver"> <name>ClientArtifactResolver</name> </reference> </gbean> <gbean name="ClientArtifactResolver" class="org.apache.geronimo.system.resolver.ExplicitDefaultArtifactResolver"> <reference name="ArtifactManager"> <name>ArtifactManager</name> </reference> <reference name="Repositories"></reference> <attribute name="versionMapLocation">var/config/client_artifact_aliases.properties</attribute> <reference name="ServerInfo"> <name>ServerInfo</name> </reference> </gbean> <gbean name="Jsr88" class="org.apache.geronimo.system.plugin.ServerInstance"> <attribute name="serverName">jsr88</attribute> <reference name="PluginAttributeStore"> <name>Jsr88AttributeManager</name> </reference> <reference name="ArtifactResolver"> <name>ArtifactResolver</name> </reference> </gbean> <gbean name="Jsr88AttributeManager" class="org.apache.geronimo.system.configuration.LocalAttributeManager"> <reference name="ServerInfo"> <name>ServerInfo</name> </reference> <attribute name="readOnly">true</attribute> <attribute name="configFile">var/config/jsr88-configurer-config.xml</attribute> <attribute name="substitutionsFile">var/config/config-substitutions.properties</attribute> <attribute name="substitutionPrefix">org.apache.geronimo.config.substitution.</attribute> </gbean>

By default, plugins are installed into the default server instance. If you need to install into a different instance you can specify this in the **config-xml-content**, **config-substitution**, and **artifact-alias** elements. Here's an example from **client-transaction**, showing how it redirects any dependencies to the server transaction plugin to itself.

xmlsolid <artifact-alias server="client" key="org.apache.geronimo.configs/transaction//car">org.apache.geronimo.configs/client-transaction/2.1-SNAPSHOT/car</artifact-alias> <artifact-alias server="client" key="org.apache.geronimo.configs/transaction/2.1-SNAPSHOT/car">org.apache.geronimo.configs/client-transaction/2.1-SNAPSHOT/car</artifact-alias>

Using maven to generate the Geronimo plugin descriptor

A lot of the information in **`geronimo-plugin.xml`** file is normally known to maven, such as the module Id and dependencies. Thus the **car-maven-plugin**, along with generating the car file, generates **`geronimo-plugin.xml`**. Much of the content is simply copied, but there are a few additional configuration choices. Lets look at the jetty web container plugin **`pom.xml`** file.

The **moduleId**, **name**, and **description** go into the **`geronimo-plugin.xml`**.

xmlsolid <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd"> <modelVersion>4.0.0</modelVersion> <parent> <groupId>org.apache.geronimo.configs</groupId> <artifactId>configs</artifactId> <version>2.1-SNAPSHOT</version> <relativePath>../pom.xml</relativePath> </parent> <artifactId>jetty6</artifactId> <name>Geronimo Configs :: Jetty 6</name> <packaging>car</packaging> <description>Geronimo Jetty Web Server integration.</description> <dependencies> <!-- dependencies ommitted --> </dependencies> <build> <plugins> <plugin> <groupId>org.apache.geronimo.plugins</groupId> <artifactId>car-maven-plugin</artifactId> <configuration>

For most plugins, all the dependencies that are plugins should be running. In this case we can use the maven dependencies unmodified as the Geronimo dependencies. For some plugins, typically deployers, some of the dependent plugins need to be loaded so their classes are available but they may not need to be started (for instance they might open ports that would conflict). In such cases you need to indicate not to use the maven dependencies and to specify them explicitly in this section. The dependencies here go both into the **`plan.xml`** file and the **`geronimo-plugin.xml`** file.

xmlsolid <useMavenDependencies> <value>true</value> <includeVersion>true</includeVersion> </useMavenDependencies> <category>Jetty</category>

The contents of the instance are copied pretty much unchanged into **`geronimo-plugin.xml`**. You can specify a **default-instance** element in a parent pom plugin configuration and that will be merged into the **`geronimo-plugin.xml`** of all the children. Note that due to some peculiarities of maven xml processing you have to use **#{foo}** instead of **${foo}** for substitution variables.

xmlsolid <instance> <plugin-artifact> <config-xml-content> <gbean name="JettyWebConnector"> <attribute name="host">#{ServerHostname}</attribute> <attribute name="port">#{HTTPPort + PortOffset}</attribute> <attribute name="redirectPort">#{HTTPSPortPrimary + PortOffset}</attribute> </gbean> <gbean name="JettyAJP13Connector"> <attribute name="host">#{ServerHostname}</attribute> <attribute name="port">#{AJPPort + PortOffset}</attribute> <attribute name="redirectPort">#{HTTPSPortPrimary + PortOffset}</attribute> </gbean> <gbean name="JettySSLConnector"> <attribute name="host">#{ServerHostname}</attribute> <attribute name="port">#{HTTPSPort + PortOffset}</attribute> </gbean> </config-xml-content> <config-substitution key="HTTPPort">8080</config-substitution> <config-substitution key="AJPPort">8009</config-substitution> <config-substitution key="HTTPSPort">8443</config-substitution> <config-substitution key="ServerHostname">0.0.0.0</config-substitution> <config-substitution key="webcontainer">JettyWebContainer</config-substitution> <config-substitution key="webcontainerName">jetty6</config-substitution> </plugin-artifact> </instance> </configuration> </plugin> </plugins> </build> </project>