

# Plugin and plugins group

Apache Geronimo is now assembled completely out of plugins including the server configuration files, in this section we will explain the overview of plugin system and introduce concept of plugin groups betaken starting from v2.2.

- [Plugin basics](#)
- [Plugins group](#)

## Plugin basics

A geronimo plugin consists of a classloader specification, optional classes, optional service or component configuration, and information about how to install it in Geronimo. 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 about 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:

```
<?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 information 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.

```
<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.

```

<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` in `geronimo-plugin.xml` are copied into such an element in `config.xml`. Note the use of substitution variables such as `$(ServerHost)`.

```

<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.

```

<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 `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>`

```

</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:

```

<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.

```
<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>
```

## Plugins group

A plugins group is basically a set of plugins for users to easily understand and consume.

[Customizing server assemblies](#) will be easier and a plugins group can be installed as a regular plugin by using [deploy](#) command. There are 8 groups for different distribution with Web Container such as Tomcat and Jetty. We take Tomcat as an example here and introduce each plugin group and what plugins are included:

### 1. Geronimo Plugin Group :: Framework

- Description: This plugin group provides the framework functionality. It is required to build a working server.
- Plugin included:

Plugin name	ModuleId	description
Geronimo Assemblies :: Boilerplate Minimal	org.apache.geronimo.assemblies/geronimo-boilerplate//car	Provides boilerplate files for all Geronimo Server assemblies.
Geronimo Framework, Configs :: GShell Framework	org.apache.geronimo.framework/gshell-framework//car	Provides the GShell Framework used by the Geronimo Server.
Geronimo Framework, Configs :: GShell Geronimo	org.apache.geronimo.framework/gshell-geronimo//car	Provides the GShell commands to operate on the Geronimo Server.
Geronimo Framework, Configs :: GShell Remote	org.apache.geronimo.framework/gshell-remote//car	Provides the GShell commands to operate on a remote Geronimo Server.
Geronimo Framework, Configs :: J2EE System	org.apache.geronimo.framework/j2ee-system//car	Provides base geronimo server that sets up the kernel.
Geronimo Framework, Configs :: Client System	org.apache.geronimo.framework/client-system//car	Provides initial configuration for a demo Geronimo server.
Geronimo Framework, Configs :: RMI Naming	org.apache.geronimo.framework/rmi-naming//car	Provides fundamental geronimo services including the plugin installer.
Geronimo Framework, Configs :: Plugin Management	org.apache.geronimo.framework/plugin//car	Provides plugin functionality.
Geronimo Framework, Configs :: J2EE Security	org.apache.geronimo.framework/j2ee-security//car	Provides basic Geronimo server security infrastructure.
Geronimo Framework, Configs :: Server Security Configuration	org.apache.geronimo.framework/server-security-config//car	Provides sample security configuration for a demo geronimo server. Not suitable for production use.
Geronimo Framework, Configs :: Shutdown	org.apache.geronimo.framework/shutdown//car	Provides stop server functionality.
Geronimo Framework, Configs :: JSR-88 CLI	org.apache.geronimo.framework/jsr88-cli//car	Provides command line implementation of JSR-88 specification.

### 2. Geronimo Plugin Group :: WADI Clustering Tomcat

- Description: This plugin group provides WADI Clustering functionality for Tomcat.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Plugins, Clustering :: Farming	org.apache.geronimo.configs/farming//car	Provides basic farming support.
Geronimo Plugins, Clustering :: WADI	org.apache.geronimo.configs/wadi-clustering//car	Provides Geronimo WADI integration.
Geronimo Plugins, Tomcat :: Clustering Builder for WADI	org.apache.geronimo.configs/openejb-clustering-builder-wadi//car	Provides openejb and WADI integration.
Geronimo Plugins, Tomcat :: Clustering over WADI	org.apache.geronimo.configs/tomcat6-clustering-builder-wadi//car	Provides deployment for WADI clustering on Tomcat 6.

### 3. Geronimo Plugin Group :: ejb

- Description: This plugin group provides EJB functionality.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Plugins, Corba :: Deployer	org.apache.geronimo.configs/openejb-corba-deployer//car	Provides the Geronimo deployer for corba security configurations for openejb.
Geronimo Plugins, Corba :: Yoko ORB	org.apache.geronimo.configs/j2ee-corba-yoko//car	Provides Yoko Orb server setup with naming service, protected orb, and unprotected orb.
Geronimo Plugins, MEJB :: Config	org.apache.geronimo.configs/mejb//car	MEJB plugin.

#### 4. Geronimo Plugin Group :: client

- Description: This plugin group provides Client functionality.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Plugins, Corba :: Client Yoko	org.apache.geronimo.configs/client-corba-yoko//car	Provides Geronimo deployer for corba security configurations for openejb.
Geronimo Plugins, Client :: Deployer	org.apache.geronimo.configs/client-deployer//car	Provides Geronimo deployer for standard JEE artifacts.
Geronimo Plugins, J2EE :: JSR-88	org.apache.geronimo.configs/jsr88-jar-configurer//car	Provides a base server that lets you implement offline JSR-88 APIs.

#### 5. Geronimo Plugin Group :: Persistence

- Description: This plugin group provides Persistence functionality.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Plugins, OpenJPA :: Deployer	org.apache.geronimo.configs/persistence-jpa10-deployer//car	Provides Geronimo Persistence Unit deployer.
Geronimo Plugins, OpenJPA :: OpenJPA	org.apache.geronimo.configs/openjpa//car	Provides OpenJPA plugin.

#### 6. Geronimo Plugin Group :: Web Little-G Tomcat

- Description: This plugin group provides the little-G Tomcat functionality.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Assemblies :: Boilerplate	org.apache.geronimo.assemblies/geronimo-boilerplate//car	Provides boilerplate files for all Geronimo Server assemblies.
Geronimo Framework, Configs :: CLI Upgrade	org.apache.geronimo.framework/upgrade-cli//car	Provides repository registration of a plugin. ( <i>obsolete</i> )
Geronimo Framework, Configs :: Offline Deployer	org.apache.geronimo.framework/offline-deployer//car	Provides offline deployer.
Geronimo Plugins, Hot Deploy :: Deployer	org.apache.geronimo.configs/hot-deployer//car	Provides a substitute for using explicit deploy commands from maven, the command line, or the console.
Geronimo Plugins, J2EE :: JSR-88 EAR Configurer	org.apache.geronimo.configs/jsr88-ear-configurer//car	Provides ear implementation of JSR-88 specification.
Geronimo Plugins, J2EE :: JSR-88 RAR Configurer	org.apache.geronimo.configs/jsr88-rar-configurer//car	Provides rar implementation of JSR-88 specification.
Geronimo Plugins, J2EE :: JSR-88 WAR Configurer	org.apache.geronimo.configs/jsr88-war-configurer//car	Provides war implementation of JSR-88 specification.
Geronimo Plugins, Remote Deploy :: Tomcat	org.apache.geronimo.configs/remote-deploy-tomcat//car	Provides a Geronimo remote deploy upload servlet (Tomcat).
Geronimo Framework, Configs :: GShell Geronimo	org.apache.geronimo.framework/gshell-geronimo//car	Provides the GShell commands to operate on the Geronimo Server.
Geronimo Framework, Configs :: GShell Remote	org.apache.geronimo.framework/gshell-remote//car	Provides the GShell commands to operate on a remote Geronimo Server.
Geronimo Plugins, Remote Deploy :: Tomcat	org.apache.geronimo.configs/tomcat6-deployer//car	Provides the Geronimo deployer for Tomcat web container.
Geronimo Plugins, Jasper :: Deployer	org.apache.geronimo.configs/jasper-deployer//car	Provides the deployer for Jasper jsps. Installs injection support components.
Geronimo Plugins, Shared Library	org.apache.geronimo.configs/sharedlib//car	Shared Library GBean.

#### 7. Geronimo Plugin Group :: Web Services Axis2

- Description: This plugin group provides Web Services Axis2 functionality.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Plugins, AXIS :: Deployer	org.apache.geronimo.configs/axis-deployer//car	Provides the web Services Deployer for Geronimo Axis 1 integration.
Geronimo Plugins, AXIS2 :: Deployer	org.apache.geronimo.configs/axis2-deployer//car	Provides the web Services Deployer for Geronimo Axis 2 integration.
Geronimo Plugins, AXIS2 :: EJB Deployer	org.apache.geronimo.configs/axis2-ejb-deployer//car	Provides the Geronimo JAX-WS EJB deployer for Apache Axis2.
Geronimo Plugins, JAXWS :: Tools	org.apache.geronimo.configs/jaxws-tools//car	Provides JAX-WS command line tools.
Geronimo Plugins, CXF :: Tools CLI	org.apache.geronimo.configs/cxf-tools//car	Provides CXF JAX-WS command line tools.

8. Geronimo Plugin Group :: Web Services CXF

- Description: This plugin group provides Web Services CXF functionality.
- Plugins included:

Plugin name	ModuleId	description
Geronimo Plugins, AXIS :: Deployer	org.apache.geronimo.configs/axis-deployer//car	Provides web Services Deployer for Geronimo Axis 1 integration.
Geronimo Plugins, CXF :: Deployer	org.apache.geronimo.configs/cxf-deployer//car	Provides Geronimo JAX-WS deployer for Apache CXF.
Geronimo Plugins, CXF :: EJB Deployer	org.apache.geronimo.configs/cxf-ejb-deployer//car	Provides Geronimo JAX-WS EJB deployer for Apache CXF.
Geronimo Plugins, JAXWS :: Tools	org.apache.geronimo.configs/jaxws-tools//car	Provides JAX-WS command line tools.
Geronimo Plugins, CXF :: Tools CLI	org.apache.geronimo.configs/cxf-tools//car	Provides CXF JAX-WS command line tools.

Look into [Administering plugins](#) for how to manage a plugin.