

Deployment

{scrollbar}

What changed ??

From a deployment perspective there has been four significant enhancements to the 2.1 version of the Geronimo server:

1. The first is deployment using **GShell**. GShell is a command-line processing environment that can be used for the execution of Geronimo commands. GShell is an extensible environment and includes support for editing, command history, and tab completion. Deployment using **GShell** is described here: [GShell](#).
2. The second is deployment to a clustered farm of Geronimo servers. WADI can now be used to support clustering of web applications for Geronimo configurations which use the Tomcat Web Container (WADI support for Jetty was in previous releases). Applications can be deployed to administratively-defined groups of Geronimo servers. Deployment to a clustered farm is described here: [Farming](#).
3. Geronimo now provides a [Plan Creator](#) that automatically generates the necessary deployment plans based on the standard deployment descriptors and annotations, and allows deployment of the corresponding module from which the plans are generated.
4. Finally, starting with Geronimo 2.1 the servers are assembled entirely out of plugins. Plugins can now be used to extend the functionality of Geronimo itself, install applications that have been packaged as plugins, create custom assemblies of the Geronimo server using Maven or extracting the plugins from an existing server. Plugin information is described here: [Plugin infrastructure](#).

Deployment Options

Standard command line

The Geronimo command line can be used to invoke the deployer tool, which is a Java application that manages Java EE artifacts, GBean components, and plugins in the Geronimo server. If Geronimo is running, it will connect to the server and perform its action through the server's deployment service. If it cannot find a running server, it will throw an error stating it could not connect to the server or the server is unavailable. More Geronimo deployer information and details can be found here:

- [Deploy](#)

GShell command line

GShell is a command-line processing environment that can be used for the execution of Geronimo commands. GShell is an extensible environment and includes support for editing, command history, and tab completion. A number of Geronimo administrative commands have been implemented using GShell. The old administrative scripts (e.g. `geronimo.sh/.bat`, `deploy.sh/bat`, etc) are still available. However, over time, we will be moving our administrative command support to be solely GShell-based. GShell can also be used to install plugins to the Geronimo server. More GShell information and details can be found here:

- [GShell](#)

Clustered deployment

WADI can now be used to support the replication of HTTP Session state among multiple Geronimo servers. In previous releases of Geronimo, WADI could only be used along with the Jetty configuration of Geronimo. WADI can now be used with Tomcat configurations of Geronimo. In addition to the new session replication support, applications can now be deployed to administratively-defined groups of Geronimo servers. This makes it easier to manage a single application across a number of Geronimo servers. More information and details about clustered deployment can be found here:

- [Farming](#)

Plugins

Starting with Geronimo 2.1 the servers are assembled entirely out of plugins. Plugins can now be used to extend the functionality of Geronimo itself, install applications that have been packaged as plugins, create custom assemblies of the Geronimo server using Maven or extracting the plugins from an existing server. Deployment and installation of plugins is now available from the Geronimo command line, GShell, and the Geronimo Admin Console. More information can be found below:

- [Tools and commands](#)
- [GShell](#)
- [Geronimo Administration Console](#)

Console

The Geronimo Administration Console can also be used to deploy and undeploy new applications and plugins. More information and details can be found here:

- [Geronimo Administration Console](#)

GEP - Eclipse

The Geronimo Eclipse Plugin can also be used to deploy and undeploy new applications to a Geronimo server, which may be running either inside or outside of the Eclipse environment. More information and details can be found here:

- [Quick start - fast and easy development](#)
- [Development environment](#)

Hot deployment

The directory structure of an installed Geronimo server includes a `<geronimo_home>/deploy` subdirectory. This subdirectory is where the Geronimo "hot deployer" will regularly check for new artifacts to be deployed, modified artifacts to be redeployed, or deleted artifacts to be undeployed. Modules copied to the hot deploy directory will be automatically deployed to the Geronimo server, modules updated in the hot deploy directory will be redeployed to the server, and any modules deleted from the hot deploy directory will be undeployed from the server. Note that there are a couple of stipulations when using Geronimo hot deployment capability:

1. Geronimo-specific deployment plans must be packaged with the module to be deployed; they cannot exist external to the module.
2. The Geronimo server must be running when modules are deleted from the hot deploy directory; otherwise they will **not** be deleted when the server is restarted.

Plan Creator

Geronimo provides a Plan Creator wizard that automatically generates the necessary deployment plans based on the standard deployment descriptors and annotations, and also allows the deployment of the corresponding module from which the plans are generated. More information, details, and detailed examples can be found here:

- [Plan Creator](#)

Annotations

Every module that you install in Geronimo, whether it is a service, application, resource, etc., can be configured via a deployment plan. These deployment plans are XML files based on XML Schemas containing the configuration details for a specific application module or component. The Java EE 5 specification defines standard deployment descriptors such as `web.xml`, `application.xml`, etc. The Java EE 5 specifications also allows the usage of annotations directly in the Java code to mitigate the development burdens associated with creating Java EE 5 applications.

Annotations in the Java EE specifications is a new metadata facility where many java entities such as classes, fields, packages, methods, etc. can be decorated with annotations. These declarations can reduce, and in many cases eliminate entirely, the necessity of using deployment plans. When both annotations and a declarations in a deployment plan are provided the whe declarations in the deployment plna will take precedence. This allows changes to the deployment characteristics of Java EE 5 modules without having to change the code.

Geronimo as a Java EE 5 certified application server supports all the annotations in the Java EE 5 specifications. These annotations are listed and described below, along with the corresponding Java EE 5 specification, and the deployment plan XML element(s) they can replace (if applicable):

- [Annotations](#)

Deployment Plans

Deployment plans

- - [Plan Creator](#)
 - `geronimo-web.xml`
 - `geronimo-application.xml`
 - `geronimo-ra.xml`
 - `geronimo-application-client.xml`
 - `openejb-jar.xml`
 - `datasources`
 - `messaging`
 - `security`

Content:

true