

# Little-G - minimal-tomcat-server

In this article you will find instructions for building and managing Apache Geronimo **Little-G** *minimal-tomcat-server* as well as example commands for deploying and managing applications.

This article is organized in the following sections:

- [#Installation](#)
- [#Start the server](#)
- [#Stop the server](#)
- [#Configuring log level](#)
- [#Changing default ports](#)
- [#Installing applications](#)
- [#Uninstalling applications](#)
- [#Updating applications](#)
- [#Starting and stopping applications](#)
- [#Additional resources](#)

## Installation

Currently, the only way to access this Geronimo configuration is by building it directly from the source, in this particular case from the **Trunk**. This section will show you how to download it using **SVN** and build it with **Maven**.

### Prerequisites

As a minimum you will need the following software already installed and configured.

- Apache Maven  
Maven 1.1-beta-2 will be used to build Geronimo and the sample application. Refer to the following URL for details on how to download and install Maven: <http://maven.apache.org>
- SubVersion  
SubVersion is used to retrieve the most updated version of the Geronimo source code. Refer to the following URL for details on how to download and install SubVersion: <http://subversion.tigris.org>
- J2SE 1.4.2  
Geronimo is currently using the Sun ORB for RMI so it is required to use SUN JDK. Using a different JDK or different version of the JDK may result in compilation errors. Refer to the following URL for details on how to download and install J2SE 1.4.2: <http://java.sun.com>

### Download the source code

Once you have installed and configured all the prerequisite software you can download Geronimo. This section provides the necessary instructions and commands for downloading and building Geronimo from the Apache versioned repository source code.

To download Geronimo the first thing you have to do is to checkout the Geronimo source by typing the following command:

```
svn checkout http://svn.apache.org/repos/asf/geronimo/trunk <trunk_home>
```

where <trunk\_home> is the name of the directory that will be created to download the source code.

Depending on you network connection speed this process may take several minutes.

### Build from the source

Since we are only interested in building Little-G we will position directly in that directory to run the necessary commands. After downloading the source from the repository a directory structure was created. Change to the following directory:

```
<trunk_home>\assemblies\minimal-tomcat-server
```

From that directory, run Maven with no additional arguments to build Little-G. Simply type the following command:

```
maven
```

This will download all the packages needed and compile the source code. Running this command may take several minutes depending on you connection speed.

Once the build finished you should see a new subdirectory **target**. This directory will contain the compressed (zip and tar gz) versions of Little-G as well as an uncompressed version in the **geronimo-1.2-SNAPSHOT** directory. You may want to move and/or rename this directory, throughout the rest of this article we will refer to this directory as <geronimo\_home>.

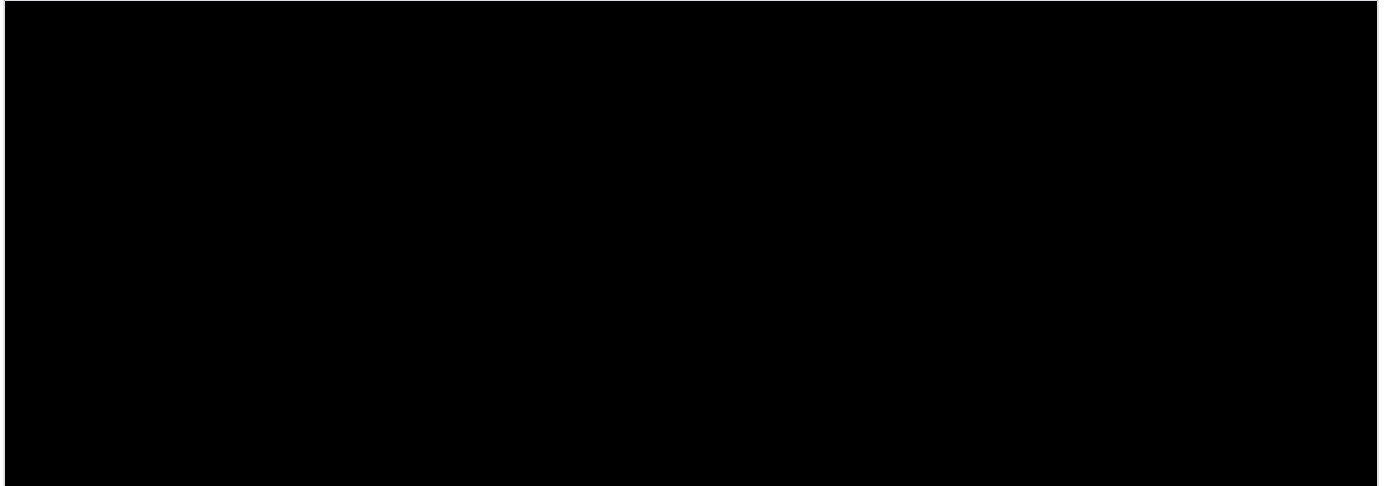
[Back to Top](#)

# Start the server

To start Geronimo type the following command:

```
<geronimo_home>\bin\geronimo run
```

Depending on the operating system you are using, you should see a screen similar to this:



Keep in mind that this Geronimo configuration has no sample applications pre-deployed, to test the server you will have to manually deploy an application. Deploying applications will be covered later in this article.

In addition to this command there are a couple of options to start the server depending on the command you use and the parameters you specify. There are basically two commands to start the server, these commands are **startup** and **geronimo run** (which we just covered).

Independently on the command you use to start the server, you will still have the same set of parameters available. The available parameters are:

## **--quiet**

Suppress the normal startup progress bar. This is typically used when redirecting console output to a file, or starting the server from an IDE or other tool.

## **--long**

Write startup progress to the console in a format that is suitable for redirecting console output to a file, or starting the server from an IDE or other tool (does not use linefeeds to update the progress information that is used by default if you do not specify **--quiet** or **--long**).

## **-v --verbose**

Sets the console log level to INFO, resulting in more console output than is normally present.

## **-vv --veryverbose**

Sets the console log level to DEBUG, resulting in even more console output.

## **-override [configId]**

Overrides the configurations in `<geronimo_home>/var/config.list` such that only the configurations listed on the command line will be started. Note that many J2EE features depend on certain configs being started, so you should be very careful what you omit. Any arguments after **-override** are assumed to be configuration names.

As a difference, **geronimo run** will *run* on the same window where you typed the command, **startup** will open a new window. The information displayed on screen will be slightly different too.

[Back to Top](#)

# Stop the server

To stop the server you can simply press the **Ctrl+C** key sequence on the window where Geronimo is running.

As an alternative to stop the Geronimo server you have the **shutdown**. This command has the following syntax:

```
<geronimo_home>/bin/shutdown [options]
```

The available options are:

## **--user [user\_name]**

Specifies the user name with the authority to stop the server. By default you would normally use system as the user name.

`--password [password]`

Specifies the password for the user name you just entered. By default you would normally use manager as the password.

`--port [port_number]`

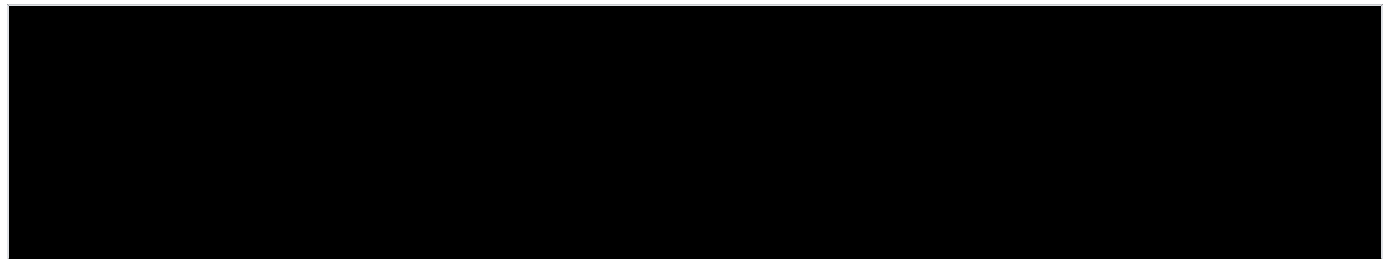
Specifies the RMI naming port to connect to the server (for example JMX connection port). By default you would normally use port 1099.

If you do not specify any parameters, this command will prompt you for a user name and password and will assume port 1099 by default and will not prompt for any port.

To shutdown the server run the following command:

```
<geronimo_home>\bin\shutdown --user system --password manager
```

Depending on the operating system you are using, you should see a screen similar to this:



[Back to Top](#)

## Configuring log level

Geronimo uses log4j for logging, log properties files and logs are located in the `<geronimo_home>\var\log` directory. To configure the log level settings edit the `server-log4j.properties` file. The purpose of this section is not to teach you how to use **log4j** but to help you understand how it is implemented in Geronimo and where in the properties file make the setting changes so

By default the log level is set to **DEBUG**, the available values are **All**, **DEBUG**, **INFO**, **WARN**, **ERROR**, **FATAL**, **TRACE** and **OFF**.

Edit the `server-log4j.properties` file and look for the following entry `log4j.rootCategory=DEBUG`, changing this value will affect the level of detail of the info stored in the log file.

In this properties file you can also set the log file name and maximum size. These values are set by default to `geronimo.log` and `10MB` respectively.



We just mentioned that the log level is set to **DEBUG** by default. When you start Geronimo with this log level setting you will see an SSL implementation error, here is an excerpt:

```
DEBUG [SSLImplementation] Error loading SSL Implementation org.apache.tomcat.util.net.puretls.
PureTLSImplementation
java.lang.ClassNotFoundException: COM.claymoresystems.ptls.SSLContext
...
```

This is more a debug message than an error, Tomcat supports two different SSL implementations and one of those is TLS. This message is displayed when Tomcat loads Sun's SSL implementation after attempting and failing to load TLS. This is a message generated by Tomcat, not Geronimo and it is not an error so it can be ignored. Refer to [GERONIMO-1048](#) for tracking this issue.

[Back to Top](#)

## Changing default ports

You can see the default ports in use when you first start Apache Geronimo. The ports are assigned as follows:

Port	Description
1099	RMI Naming
4242	Remote Login Listener
8009	Tomcat Connector AJP
8080	Tomcat Connector HTTP
8443	Tomcat Connector HTTPS

If you need to change any of these ports you can edit the `config.xml` file located in the `<geronimo_home>\var\config` directory. In the `config.xml` file you will find the name of the network listener, host and port clearly identified. Here is an example:

#### Excerpt from config.xml

```
<gbean name="TomcatWebSSLConnector">
  <attribute name="host">0.0.0.0</attribute>
  <attribute name="port">8443</attribute>
</gbean>
```

[Back to Top](#)

## Installing applications

As mentioned earlier, this minimal configuration of Apache Geronimo does not include any sample application. For the purposes of showing the deployment and testing of applications there are two sample applications available for download with this article. These sample applications are provided by Apache Tomcat.

File	Sample application
<a href="#">geronimo-jsp-examples-tomcat-5.5.12-1.war</a>	JSP sample application
<a href="#">jsp-examples-tomcat-plan.xml</a>	Geronimo specific deployment plan
<a href="#">geronimo-servlet-examples-tomcat-5.5.12-1.war</a>	Servlet sample application
<a href="#">servlets-examples-tomcat-plan.xml</a>	Geronimo specific deployment plan

As we just mentioned, these sample applications are provided by Apache Tomcat and are already packaged in a **war** file. In addition, we have provided a Geronimo specific deployment plan for each of these sample applications.

Create a sample applications directory and download applications and deployment plans there, from now on this directory will be referred as **<apps\_home>**

Apache Geronimo provides the **Deployer tool** for administering applications. This tool allows you, among other things, to install, remove, list, start and stop applications.

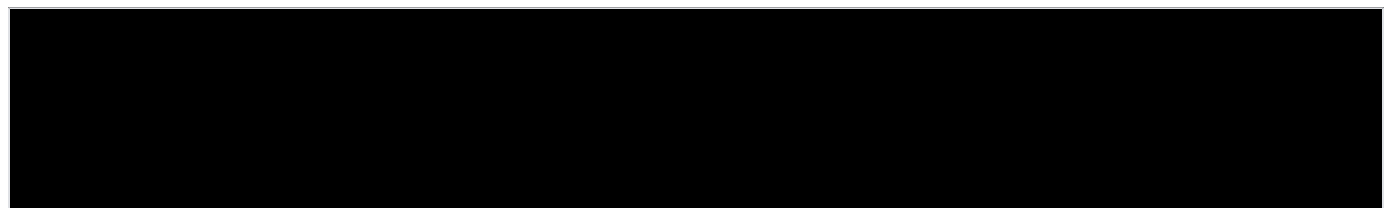


In order to deploy applications the Apache Geronimo server has to be running.

To deploy the JSP sample application run the following command:

```
<geronimo_home>\bin\deploy --user system --password manager deploy <apps_home>\geronimo-jsp-examples-tomcat-5.5.12-1.war <apps_home>\jsp-examples-tomcat-plan.xml
```

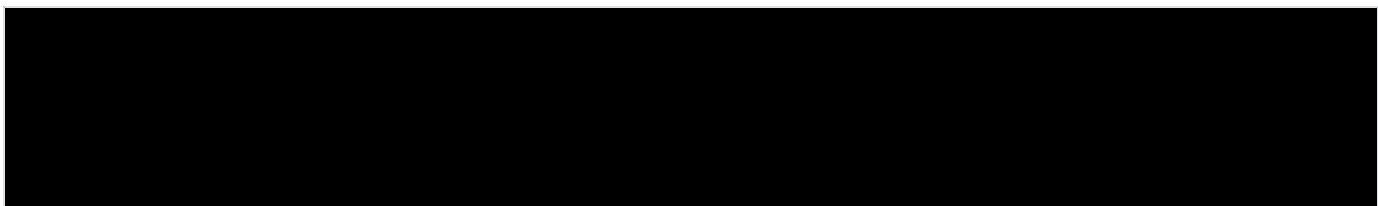
Once the application is successfully deployed you will get a confirmation message including the URL where the application has been deployed and is ready to be tested.



You would normally have the `geronimo-web.xml` deployment plan already packaged in `WEB-INF` directory along with the `web.xml`. Since these samples are from Tomcat, they do not include any Geronimo specific deployment plans but the **Deployer tool** allows you to specify an external deployment plan at deployment time.

Similarly to deploying the JSP application, type the following command to deploy the Servlet sample application:

```
<geronimo_home>\bin\deploy --user system --password manager deploy <apps_home>\geronimo-servlet-examples-tomcat-5.5.12-1.war <apps_home>\servlets-examples-tomcat-plan.xml
```



When you install an application with the **Deployer tool**, it will deploy the application and then start it.

Test the sample applications by accessing the following URLs respectively:

<http://localhost:8080/jsp-examples/>

<http://localhost:8080/servlets-examples/>

Navigate through the samples.

[Back to Top](#)

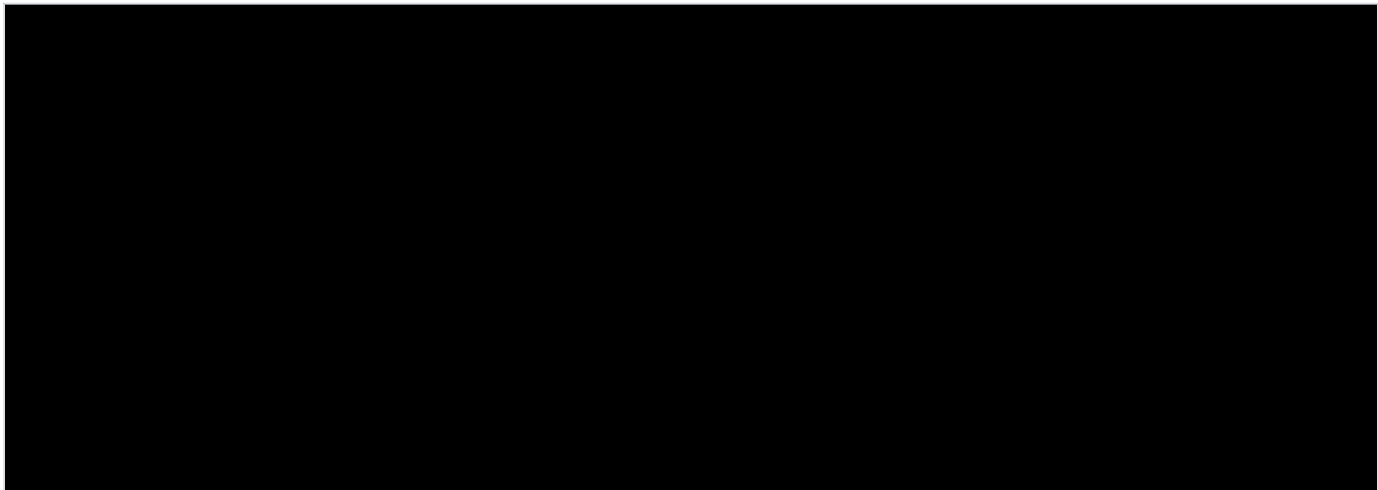
## Uninstalling applications

You can use the same **Deployer tool** to uninstall any application, in fact you can use it to uninstall any deployed module. To uninstall a module you first need to know the module's ID, if you do not know the module's ID you can list all the installed modules first to identify the application you want to uninstall.

To list the installed modules run the following command:

```
<geronimo_home>\bin\deploy --user system --password manager list-modules
```

Depending on the operating system you are using, you should see a screen similar to this:

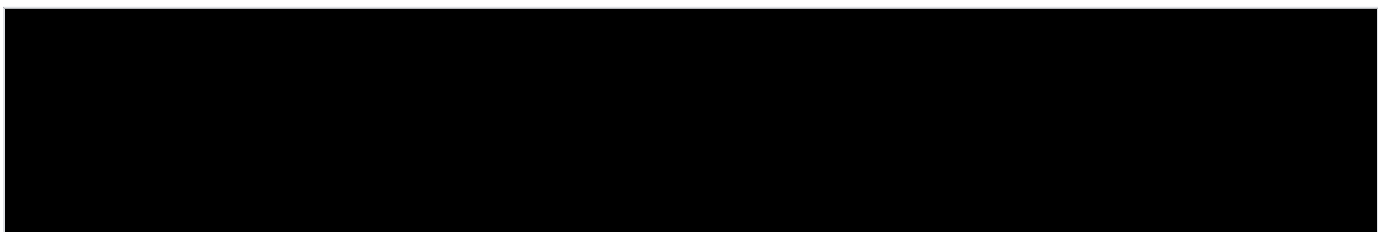


The modules that have a **+** listed on the left indicate that the module is started. The **list-modules** command can selectively display started or stopped modules depending on the additional parameters specified, if no additional parameter is specified the command will display both started and stopped modules.

Run the following command to uninstall one of the sample applications you just installed, let's say the servlets example.

```
<geronimo_home>\bin\deploy --user system --password manager undeploy geronimo/servlets-examples/1.0/war
```

Depending on the operating system you are using, you should see a screen similar to this:



[Back to Top](#)

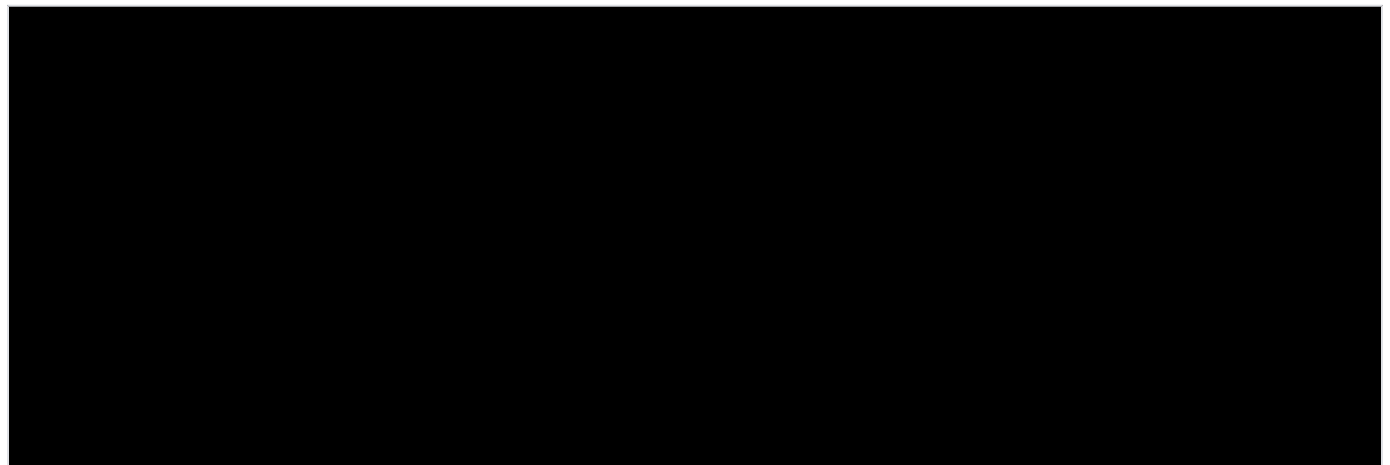
## Updating applications

To update a deployed application there is no need to stop it or undeploy it first, the **Deployer tool** provides the **redploy** command that will do the process automatically for you.

Let's say you need to update the JSP sample application deployed earlier, we assume that this application is deployed, started and working. To update the application run the following commands:

- First need to know the module ID, run the `list-modules` command to list the installed modules.  
`<geronimo_home>\bin\deploy --user system --password manager list-modules`
- Now that you have identified the module ID run the following command to update the application:  
`<geronimo_home>\bin\deploy --user system --password manager redploy \apps_home\geronimo-jsp-examples-tomcat-5.5.12-1.war \apps_home\jsp-examples-tomcat-plan.xml geronimo/jsp-examples/1.0/war`  
Note that for this command you specify:
  - Application (.war)
  - Deployment plan (.xml)
  - Module ID

Depending on the operating system you are using, you should see a screen similar to this:



[Back to Top](#)

## Starting and stopping applications

We already discussed about the `list-modules` command for listing installed modules and their status (started or stopped). You can manually start or stop a module if needed; let's go back to the JSP sample application.

When you installed the JSP sample application the **Deployer tool** deployed it and started it, if you need to stop this sample application you can run the following command:

```
<geronimo_home>\bin\deploy --user system --password manager stop geronimo/jsp-examples/1.0/war
```

You should get a confirmation message indicating the module is Stopped. If you now access the sample application now you will receive a **404 Resource not available** error.

Similarly you can run the following command to start the module:

```
<geronimo_home>\bin\deploy --user system --password manager start geronimo/jsp-examples/1.0/war
```

You should get a confirmation message indicating the module is started and the URL where is running.

[Back to Top](#)

## Additional resources

For additional details on the topics covered in this article visit the following URLs:

- [Apache Geronimo v1.0 - User's Guide](#)

- [Quick start - Apache Geronimo for the impatient](#)
- [Tools and commands](#)
- [Deployer tool](#)
- [Deployment plans](#)

[Back to Top](#)