

# Quick start - Apache Geronimo for the impatient

If you are in a hurry to have Apache Geronimo up and running right away, this article will provide you with some basic steps for downloading, building (when needed) and run the server in no time.

All you need is 5 to 10 minutes and to follow these basic steps.

- [Getting the software](#)
  - [Prerequisites](#)
  - [Download binaries](#)
- [Start the server](#)
- [Create and deploy a sample application](#)
  - [Deploy and test the application](#)
- [Summary](#)

## Getting the software

There are currently two way to get the released software. You can either retrieve it from the Apache SVN repository or the fastest and easiest way is to get the binary directly from the Apache Geronimo site. Since this article is intended to help you to have Apache Geronimo up and running in the fastest possible way, only the binary download will be covered at this time.

## Prerequisites

Although this installation path is the fastest one, still you will need to install some software before installing Geronimo.

### J2SE 1.5

Apache Geronimo v2.0 now provides full support for Sun JDK 5.0+ (J2SE 1.5.0+). Refer to the following URL for details on how to download and install J2SE 1.5:

<http://java.sun.com>

## Download binaries

Depending on the platform you plan to install and run Apache Geronimo you will select the appropriate installation image. Open a Web browser and access the following URL, there you will find the available packages for download (binaries and source code).

<http://geronimo.apache.org/downloads.html>

Select the file compression format compatible with your system (zip, tar, gz) by clicking directly on the link, download it and expand the binary to your hard drive in a new directory. For example in `z:/geronimo`, from now on this directory will be referenced as **<geronimo\_home>**.

The installation of Apache Geronimo is as simple as uncompressing the .zip or .tar.gz files. The next step is to start the server.

## Start the server

With Apache Geronimo already installed, open a command line console and change directory to **<geronimo\_home>/bin** and run the following command:

```
geronimo run
```

This script starts the server on the same terminal you run the command. Once the server is started you should see a screen similar to this one.



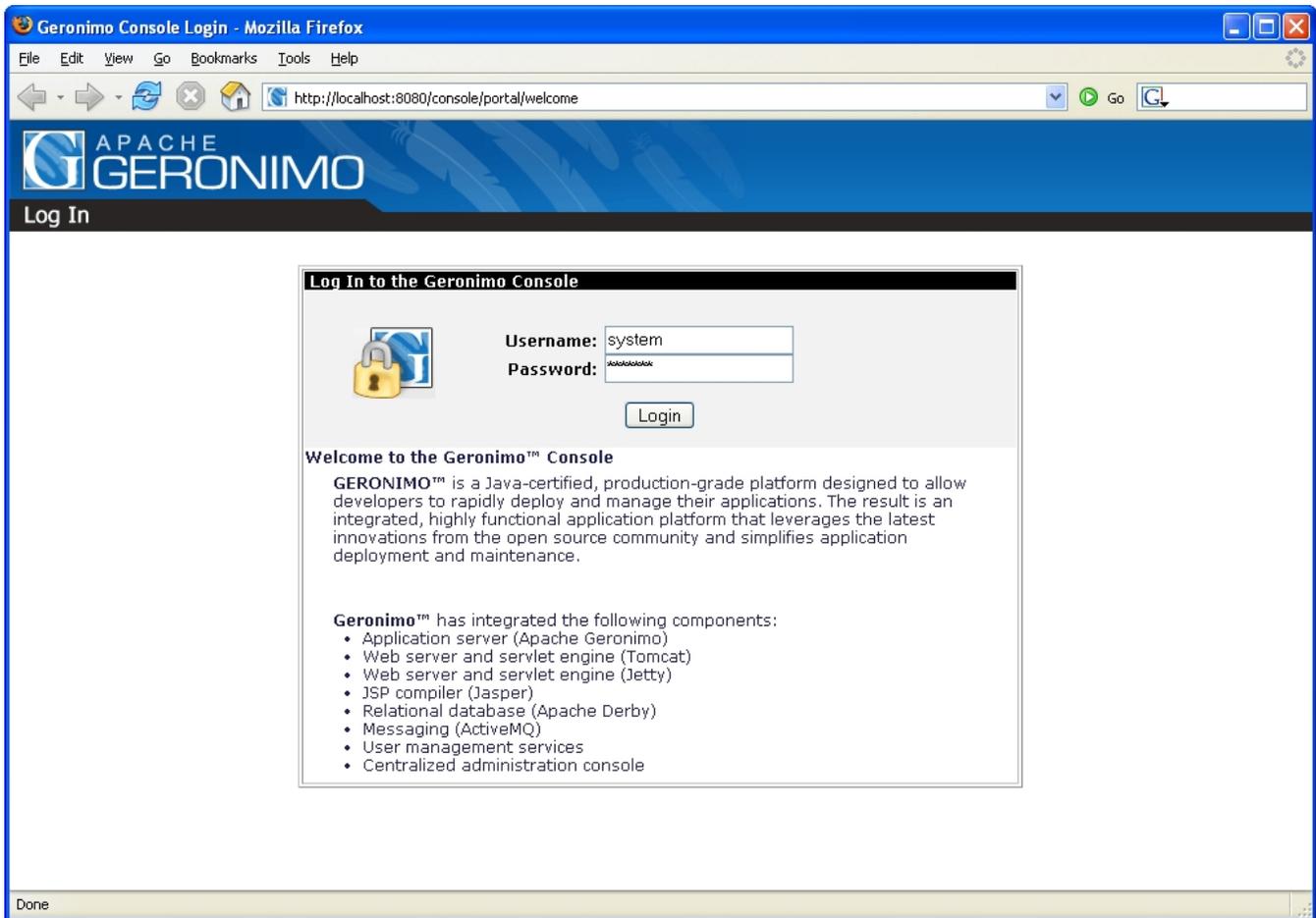
Note that some lines from the previous screenshot have been truncated for displaying purposes, instead this lines ended with "...".

To test the Apache Geronimo server you can just point your Web browser to the following URL to test the Geronimo Console. The Geronimo Administration Console is the first application you can test, and the good thing is, it is already deployed in the standard, full J2EE certified distribution.

<http://localhost:8080/console>

When prompted enter **system** as the user name and **manager** as the password.

The following figure illustrates the Web based console.



At this point you have Apache Geronimo up and running and the time spent has been minimal. The following section offers a very basic sample JSP and shows you how to deploy and run that sample application.

## Create and deploy a sample application

A **HelloWorld** like JSP is almost a default for these basic testing and everybody is already familiar with it. Create a new directory to hold all the application related files. From now on this directory will be referred as **<app\_home>**.

Create a plain text file in the **<app\_home>** directory named **HelloWorld.jsp** and copy the content of the following example.

### HelloWorld.jsp

```
<html>
  <head>
    <jsp:useBean id="datetime" class="java.util.Date"/>
    <title>
      Basic HelloWorld JSP
    </title>
  </head>
  <body bgcolor="#909DB8">
    <h1>
      <font face="tahoma" color="white">
        Hello world from Geronimo!
      </font>
    </h1>
    <font face="tahoma" color="white">on ${datetime}</font>
  </body>
</html>
```

We will now create a Geronimo specific deployment plan so you would get a first glimpse of the deployment plans. Create a directory named **WEB-INF** inside the `<app_home>` directory.

Create a plain text file in the `<app_home>/WEB-INF` directory named **geronimo-web.xml** and copy the content of the following example.

#### Geronimo deployment plan geronimo-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.2">
  <environment>
    <moduleId>
      <groupId>sample.applications</groupId>
      <artifactId>HelloWorldApp</artifactId>
      <version>2.0</version>
      <type>war</type>
    </moduleId>
  </environment>
  <context-root>/hello</context-root>
</web-app>
```

Create a plain text file in the `<app_home>/WEB-INF` directory named **web.xml** and copy the content of the following example.

#### web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"

  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

  <welcome-file-list>
    <welcome-file>HelloWorld.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

For your convenience, you can download the HelloWorld ([HelloWorld\\_2.0.zip](#) or [HelloWorld\\_2.0.war](#)) sample application provided with this article.

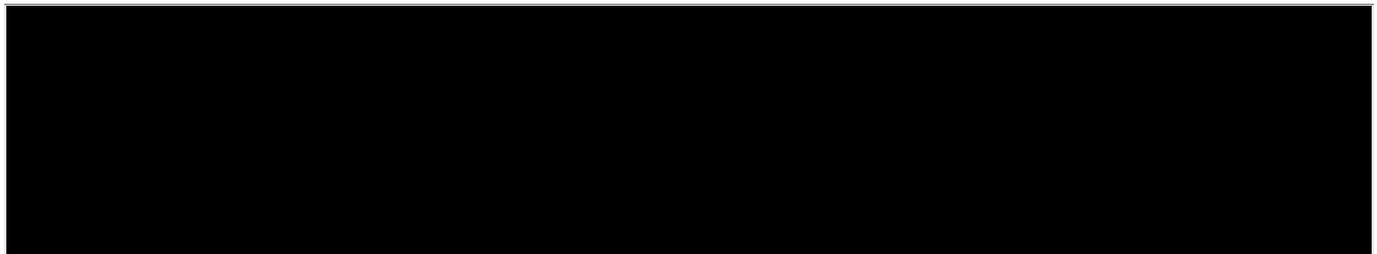
## Deploy and test the application

For this example we will not invest time in packaging the application, we will just use the **--inPlace** feature of the deployer tool which allows us to deploy an application from any directory.

Change directory to `<geronimo_home>/bin` and run the following command:

```
deploy --user system --password manager deploy --inPlace <app_home>
```

Once the application is successfully deployed you should see the following message:



Test the application by pointing a Web browser to the following URL:

<http://localhost:8080/hello>

## Summary

This article showed you how simple is to have Apache Geronimo up and running. The overall time for getting the server running should be less than five minutes if you have the prerequisites already installed. Additionally, this article also showed you how to create, deploy and test a very simple JSP in less than five minutes.