Getting Started with Geronimo

The goal of this chapter is to get Geronimo up and running for your own system in the shortest possible time. The basic system requirements, on both Unix and Windows systems will be revealed. Step-by-step installation is provided. Potential problems encountered during Geronimo installation and setup are pointed out, and we will share some of our solutions with you.

In this chapter, you will:

- ❑ Discover where to download Geronimo releases and updates
- ❑ Learn about which distribution to download and why
- ❑ Ascertain that your system is capable of running Geronimo and learn where to find the additional system requirements if needed
- ❑ Install a Geronimo server on your system
- ❑ Successfully start up and test the server

By the end of this chapter, you will have the information and confidence that you need to install and set up Geronimo servers in your own work environment. You will be ready to start deploying applications on your server.

Where to Find Geronimo

Geronimo is a top-level project at the Apache Software Foundation Web site. The headquarters for all your Geronimo needs is located at the following URL:

```
http://geronimo.apache.org
```

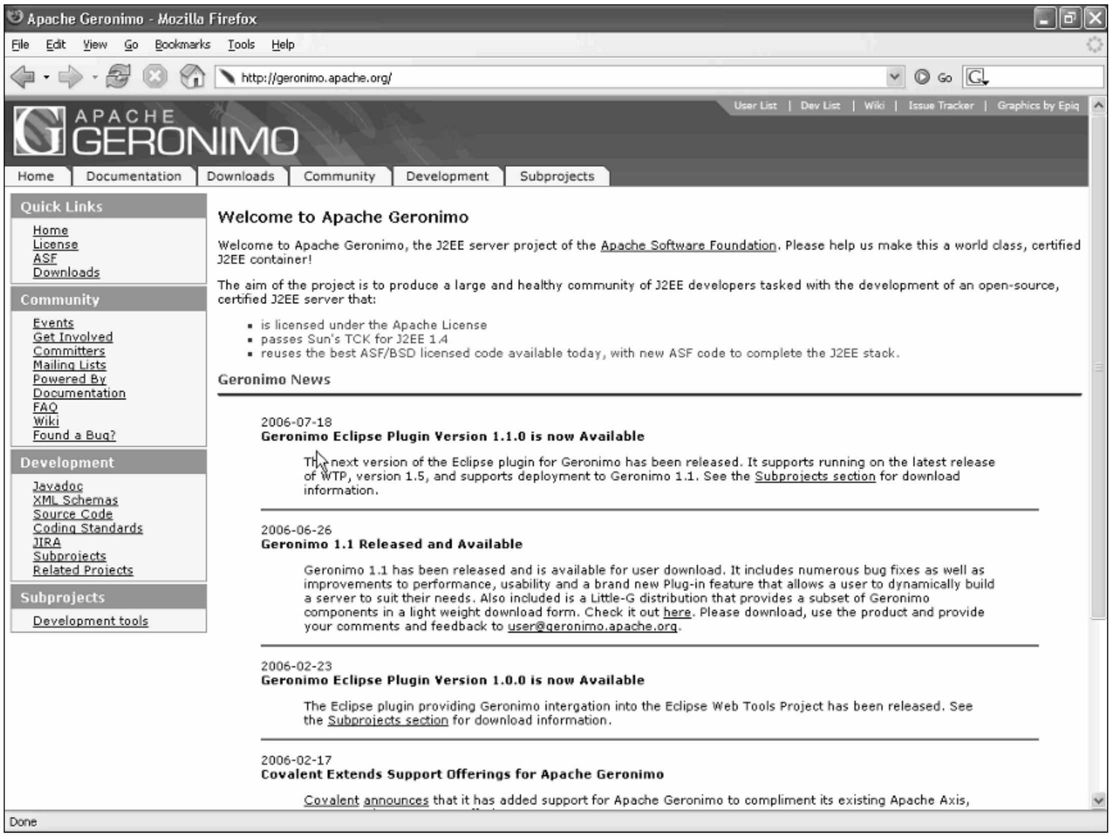
On this site, you will find the following:

- ❑ The latest Geronimo binaries
- ❑ The latest Geronimo source
- ❑ Geronimo documentation
- ❑ The latest Geronimo news and resources
- ❑ An active Geronimo developer community

The remainder of this chapter shows how to leverage the use of many of these resources.

Figure 1-1 shows the home page of the Geronimo project.

As you can see in Figure 1-1, you can find links to the latest software download and documentation by selection one of the tabs at the top of the home page. When you click on the Download tab, there are numerous download choices. Figure 1-2 shows some of the available download choices.



Welcome to Apache Geronimo

Welcome to Apache Geronimo, the J2EE server project of the Apache Software Foundation. Please help us make this a world class, certified J2EE container!

The aim of the project is to produce a large and healthy community of J2EE developers tasked with the development of an open-source, certified J2EE server that:

- is licensed under the Apache License
- passes Sun's TCK for J2EE 1.4
- reuses the best ASF/BSD licensed code available today, with new ASF code to complete the J2EE stack.

Geronimo News

2006-07-18
Geronimo Eclipse Plugin Version 1.1.0 is now Available

The next version of the Eclipse plugin for Geronimo has been released. It supports running on the latest release of WTP, version 1.5, and supports deployment to Geronimo 1.1. See the Subprojects section for download information.

2006-06-26
Geronimo 1.1 Released and Available

Geronimo 1.1 has been released and is available for user download. It includes numerous bug fixes as well as improvements to performance, usability and a brand new Plug-in feature that allows a user to dynamically build a server to suit their needs. Also included is a Little-G distribution that provides a subset of Geronimo components in a light weight download form. Check it out here. Please download, use the product and provide your comments and feedback to user@geronimo.apache.org.

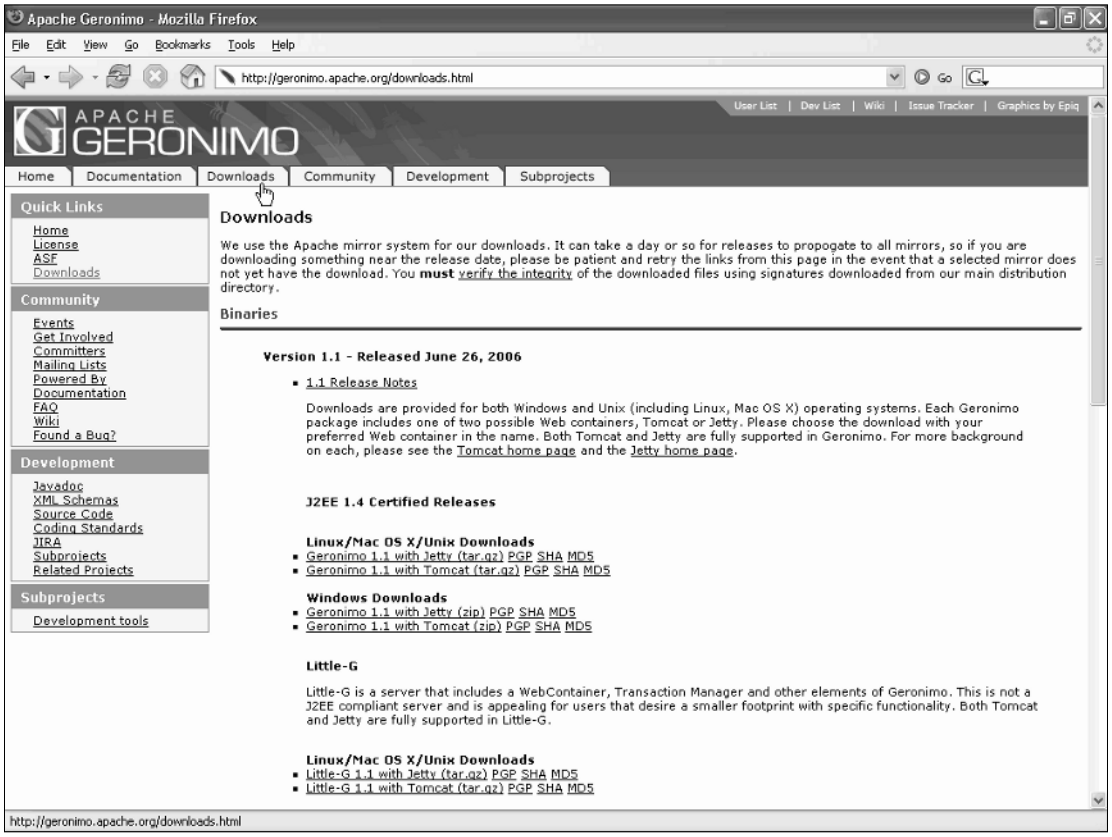
2006-02-23
Geronimo Eclipse Plugin Version 1.0.0 is now Available

The Eclipse plugin providing Geronimo intergation into the Eclipse Web Tools Project has been released. See the Subprojects section for download information.

2006-02-17
Covalent Extends Support Offerings for Apache Geronimo

Covalent announces that it has added support for Apache Geronimo to compliment its existing Apache Axis,

Figure 1-1: Geronimo project home page


Apache Geronimo - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://geronimo.apache.org/downloads.html
User List | Dev List | Wiki | Issue Tracker | Graphics by Epiq
APACHE GERONIMO
Home Documentation Downloads Community Development Subprojects
Quick Links
Home
License
ASF
Downloads
Community
Events
Get Involved
Committers
Mailing Lists
Powered By
Documentation
FAQ
Wiki
Found a Bug?
Development
Javadoc
XML Schemas
Source Code
Coding Standards
JIRA
Subprojects
Related Projects
Subprojects
Development tools
http://geronimo.apache.org/downloads.html

Downloads

We use the Apache mirror system for our downloads. It can take a day or so for releases to propogate to all mirrors, so if you are downloading something near the release date, please be patient and retry the links from this page in the event that a selected mirror does not yet have the download. You **must** verify the integrity of the downloaded files using signatures downloaded from our main distribution directory.

Binaries

Version 1.1 - Released June 26, 2006

- 1.1 Release Notes

Downloads are provided for both Windows and Unix (including Linux, Mac OS X) operating systems. Each Geronimo package includes one of two possible Web containers, Tomcat or Jetty. Please choose the download with your preferred Web container in the name. Both Tomcat and Jetty are fully supported in Geronimo. For more background on each, please see the Tomcat home page and the Jetty home page.

J2EE 1.4 Certified Releases

Linux/Mac OS X/Unix Downloads
- Geronimo 1.1 with Jetty (tar.gz) PGP SHA MD5
- Geronimo 1.1 with Tomcat (tar.gz) PGP SHA MD5

Windows Downloads
- Geronimo 1.1 with Jetty (zip) PGP SHA MD5
- Geronimo 1.1 with Tomcat (zip) PGP SHA MD5

Little-G

Little-G is a server that includes a WebContainer, Transaction Manager and other elements of Geronimo. This is not a J2EE compliant server and is appealing for users that desire a smaller footprint with specific functionality. Both Tomcat and Jetty are fully supported in Little-G.

Linux/Mac OS X/Unix Downloads
- Little-G 1.1 with Jetty (tar.gz) PGP SHA MD5
- Little-G 1.1 with Tomcat (tar.gz) PGP SHA MD5

Figure 1-2: Geronimo download page

The next section will help you decide on what to download.

Geronimo Build and Releases

Geronimo is a very large project that is created by a geographically diverse team of international Open Source developers. Because of time zone differences, development continues around the clock. The software team creates releases of software on a regular basis to synchronize work. To track all the development changes, a version control system called Subversion is used. As a result, you will frequently hear about *builds* and *releases* at the Geronimo Web site.

Builds can be considered as trial releases of the software. A build is a compilation of the project at a specific time. With the Geronimo project, a build occurs every evening, 365 days a year. As a result, you will find that build numbers can become very large.

Releases are certain builds that are considered major deliverables in the Geronimo project. It takes great effort from the development team to coordinate a release of the software. There are only a few releases of the Geronimo software thus far (1.0, 1.1, and so on).

Users are typically concerned only with releases. However, to get the latest feature or bug fix, you may have to download and install the latest build between releases. From time to time, the Geronimo team may elect to make certain builds available for download. In general, the most important builds of Geronimo are summarized in Table 1-1.

Table 1-1: Geronimo Builds

Build	Description
Latest unstable build	This is the last successful daily build. It always contains the latest features and bug fixes. However, it is not of release quality and has not been extensively tested. Use these builds at your own risk.
Latest milestone build	Before the availability of the first major release of Geronimo, a set of product features are made available to early-adopter users via milestone builds. Milestone builds are well-tested releases of the software but with only a subset of features of major releases. You will not see Geronimo milestone builds available for download anymore. This description is provided for historical reasons, and you may see some Geronimo literature referring to milestone builds.

The first Java 2 Enterprise Edition (J2EE) 1.4–complaint release of Geronimo is release 1.0. However, since the release of 1.0, Geronimo users all over the world have discovered and reported various bugs. In addition, the Geronimo team made significant configuration and architecture improvements to the server. All of this work cumulates in the release of Geronimo 1.1 — the first stable version of Geronimo ready for production deployment. This book covers the features of Geronimo 1.1 extensively.

Since Geronimo has reached release 1.1 level, you may no longer see any downloads for Geronimo builds on the Geronimo site. This is normal, because the Geronimo team has elected to produce frequent releases instead of making intermediate builds available.

You should always download a Release 1.1 or later from the Geronimo Web site. Table 1-2 shows some of the current and prior Geronimo releases.

Table 1-2: Available Geronimo Releases

Release	Description
1,1	A much improved version of 1.1. Other than fixing major bugs, this release also has enhanced features such as support for plug-ins, support for a small footprint Little-G distribution. The module configuration notation has changed significantly in 1.1. With the availability of release 1.1, there is very little reason to work with 1.0 of the server.
1.0	The first fully J2EE 1.4–compliant release. Extensively tested.

Release	Description
M5 - Milestone 5	A release candidate milestone build. All of the features in release 1.0 are included. This build is extensively tested. M5 is almost release quality and was used in limited production before release 1.0 is available.
M4 - Milestone 4	First usable build for general users. Extensively tested. Can be used to deploy most enterprise applications. Useful for early-adopter users.
M3 and before	These builds are missing major features. They were used mainly to provide development users with early access to the software base. Do not use these builds for production under any circumstances.

Binary versus Source Download

Once you have decided on a release, you can select from the type of distributions available:

- Binary
- Source code

A *binary distribution* contains all the executables you need to run a Geronimo server. Binary distributions are typically packaged in ZIP format for Windows and tar-gzip (`tar.gz`) format for Unix operating systems.

The *source code distribution* is typically of interest only to Geronimo system developers who want to modify the source code.

Advanced user of Geronimo may sometimes use the source code distribution to build their own customized version of the Geronimo or to customize the capabilities of the server. In fact, we will share some of our own Geronimo customization tips in Bonus Downloadable Chapter 18, The Essence of Geronimo: Flexible Assemblies, at `www.progeronimo.com`**.**

Jetty versus Tomcat Binary Downloads

When you arrive at the Geronimo download page, you will be provided with a choice of downloading a binary distribution of Geronimo with either the Jetty Web-tier container or the Tomcat Web-tier container. You only need to choose one of these downloads.

Geronimo as a J2EE server supports two alternative Web-tier containers: Jetty and Tomcat. The latest available version of Jetty and Tomcat, at the time of release, is typically included with the distribution. For example, Geronimo 1.1 includes Tomcat 5.5.15 or Jetty 5.1.10.

Both Jetty and Tomcat are matured Open Source containers for running servlets and Java Server Pages (JSPs). Chances are that you or your company is already using one of these Web-tier application containers, and the choice will be clear.

If you want to find out more about these Web-tier containers, the following is the link to Jetty:

```
http://jetty.mortbay.org
```

The link to Tomcat, another Apache project, is at the following URL:

```
http://tomcat.apache.org
```

Full Installation versus Minimal Little-G Server

Release 1.1 of Geronimo supports two special very small footprint versions of the server, known among the developers as *Little-G*:

- ❑ Little-G 1.1 with Tomcat
- ❑ Little-G 1.1 with Jetty

Each of these Little-G servers is stripped down to the bare minimum. For example, the support for Enterprise JavaBeans (EJBs), for the Active MQ message broker, for internal relational database, for CORBA, and all sample applications are removed. This leaves only the container services (security, transaction, deployment, and so on) and a Web-tier server (either Tomcat or Jetty). The idea behind these minimal server configurations is to enable the easy construction of customized servers. Users can start with one of the Little-G servers and add just enough support for the application that they will be running.

If you are deploying applications that only contain servlets and JSPs, Little-G may be the right download for you. Applications using lightweight frameworks (such as Spring 2) can also run in Little-G. However, if you are deploying J2EE 1.4 applications that use EJB or message queues, then you will need to download the full binary distribution.

The focus of this book is on the J2EE 1.4–compatible personality of Geronimo. You will be working mostly with the full binary distribution of Geronimo. These larger server distributions contain a superset of software components and features when compared to Little-G.

All the techniques you learn in this book will apply equally to the regular full-sized distributions, as well as the Little-G minimal server distributions.

Before Installing Geronimo

Ensure that you have met all the hardware and software requirements before you install Geronimo. This section can help you to plan your systems for wide deployment of Geronimo.

Hardware Platform Requirements

Geronimo has been tested and runs well on any x86 platform. From our experience, you should really have a processor with speed of 2 GHz and above for serious production deployment. Being a Java-based server, you are not tied to x86 processors. Geronimo has been tested on PowerPC (Mac OS), Sparc (SUN Solaris), as well as PA-RISC (HP-UX) processors.

As with all J2EE 1.4–compatible servers, the compatibility testing is performed on a highly version-specific software stack (that is, a certain operating system release level with certain Java VM release level, and so on). If you are required to run only on 100 percent J2EE 1.4 certified platform, you must duplicate the certified stack and acquire your hardware based on it. See the following URL for more information on J2EE 1.4 compatibility: http://java.sun.com/j2ee/compatibility_1.4.html

A basic Geronimo configuration (such as the Little-G servers) can be started on systems with as little as 256MB of RAM. To get reasonable performance, it is recommended that you start your production Geronimo server at 1GB of RAM.

Needless to say, you will need a network connection on the server machine. Adapters for gigabits, 100 Mbps, or even 10 Mbps will work fine for most Geronimo configurations.

JVM Requirement

In general, you will require Java Development Kit (JDK) 1.4.2 to successfully run Geronimo. More particularly, you will need SUN's Java Virtual Machine (JVM) to run release 1.1 of Geronimo if you need to support Common Object Request Broker Architecture (CORBA). This restriction is because of some internal dependency of this release of Geronimo on the Object Request Broker (ORB) included with SUN's JDK.

To eliminate the stranglehold of the non–Open Source ORB, veterans from the original Geronimo team are busy working with on the Yoko project in the Apache incubator (the stage of development when startup Open Source projects go through initial incubation). The Yoko project aims to create a complete standards-compliant, Open Source CORBA ORB. When the ORB is finished, it will be integrated in Geronimo — eliminating the need for SUN's JDK. You can find out a lot more about the Yoko project at its own Apache project page:

```
http://incubator.apache.org/projects/yoko.html
```

If you do not use any CORBA features, you can run Geronimo on non–SUN JDK 1.5, as well as any compatible version of JDK 5.

Throughout this book, it is assumed that you are running Geronimo with the latest available release of JDK 1.4.2 or JDK 5. For the CORBA configuration and support example in Chapter 14, you are required to run under SUN's JDK 1.4.2. If you do not already have SUN's JDK 1.4.2 installed on your system, you can download it from the following URL:

```
http://java.sun.com/j2se/1.4.2/download.html
```

The latest version (as of this writing) is 1.4.2_12, and has been tested to fully support Geronimo.

If you are using Mac OSX, the latest Apple JVM 1.4.2 version on OSX should support Geronimo.

As a reminder, you can always find out what version of the Java VM you have installed using the following command at the console:

```
java -version
```

Installing the Geronimo Server

The exact installation procedure depends on which binary distribution you have downloaded and also on your host operating system. In all cases, you need to preselect the directory on your machine that you will install the server on.

For the following discussion, it is assumed that you are installing the server on:

- `c:\geronimo` — For the Windows operating system
- `/home/jeff/geronimo` — For an Unix system

If you have downloaded the ZIP file on a Windows machine, just unzip the distribution to your destination directory. This will create `c:\geronimo\geronimo-1.1` (assuming that you are running the 1.1 release).

On an Unix system, change into your installation directory and the use the following command to unarchive the file:

```
tar zxvf <downloaded tar.gz file name>
```

This command unarchives the server's binary into the `/home/jeff/geronimo-1.1` directory.

This completes the initial installation of your Geronimo server. This installation comes with a preconfigured server that is designed to run on most system configurations.

The next sections show you how to customize this default configuration.

Customizing Geronimo after Installation

The initial installation provides a starter configuration for your Geronimo server. You can readily modify the configuration if required. This can be accomplished by editing the `config.xml` file.

The config.xml File

The `config.xml` is located in the `var/config` directory of your Geronimo installation. This file contains a list of modules that will be loaded and started when the Geronimo server starts up. In fact, many operations performed via the Web console (covered in Chapter 8) involve changes in this configuration file.

In addition to controlling which module is started, the file can also be used to customize some attributes of Geronimo components that are installed. This will include changing the port and hostnames/IP address of the default installation.

Generally, the Geronimo server will examine the `config.xml` during startup to determine the following:

- Which modules to load and start
- The attributes of server components (called GBeans in Geronimo) that should be overridden

Note, however, that `config.xml` should never be modified while Geronimo is running. This is because Geronimo may write the changed module configurations back to `config.xml`. Any modifications you make while Geronimo is running will be lost.

In addition, since Geronimo automates the writing of the `config.xml` from its own internal module configuration structure, any comments you placed in the original `config.xml` will also be lost after Geronimo restart. Therefore, remember to only edit `config.xml` when Geronimo is not running — and to make frequent backup of this file.

Never edit `config.xml` while the Geronimo server is running. Make frequent backups of known-good version of this file.

As an example, the module configuration entry that configures the port and hostname of the Tomcat Web connectors is shown in the following listing:

```
<module name="geronimo/tomcat/1.1/car">
  <gbean name="TomcatResources"/>
  <gbean name="TomcatWebConnector">
     <attribute name="host">0.0.0.0</attribute>
     <attribute name="port">8080</attribute>
     <attribute name="redirectPort">8443</attribute>
   </gbean>
   <gbean name="TomcatAJPConnector">
      <attribute name="host">0.0.0.0</attribute>
      <attribute name="port">8009</attribute>
      <attribute name="redirectPort">8443</attribute>
   </gbean>
   <gbean name="TomcatWebSSLConnector">
      <attribute name="host">0.0.0.0</attribute>
      <attribute name="port">8443</attribute>
   </gbean>
   <gbean
name="geronimo/tomcat/1.1/car?ServiceModule=geronimo/tomcat/1.1/car,j2eeType=GBean,
name=TomcatWebContainer">
       <attribute name="catalinaHome">var/catalina</attribute>
   </gbean>
</module>
```

You can modify any of these values and start the server to have the host and port changed.

Any listed modules in the `config.xml` will be loaded and started by default. To prevent one or more of the modules from loading and starting by Geronimo, set the attribute `load="false"` in the `<module>` element. For example, if you look up the CORBA support module, with name `geronimo/j2ee-corba/1.1/car`, you will see the following module configuration:

```
<module load="false" name="geronimo/j2ee-corba/1.1/car">
<gbean name="NameServer">
<attribute name="dbDir">var/cosnaming.db</attribute>
```

```
<attribute name="port">1050</attribute>
</gbean>
<gbean name="Server">
<attribute name="args">-ORBInitRef,
NameService=corbaloc::localhost:1050/NameService</attribute>
</gbean>
<gbean name="UnprotectedServer">
<attribute name="args">-ORBInitRef,
NameService=corbaloc::localhost:1050/NameService</attribute>
</gbean>
</module>
```

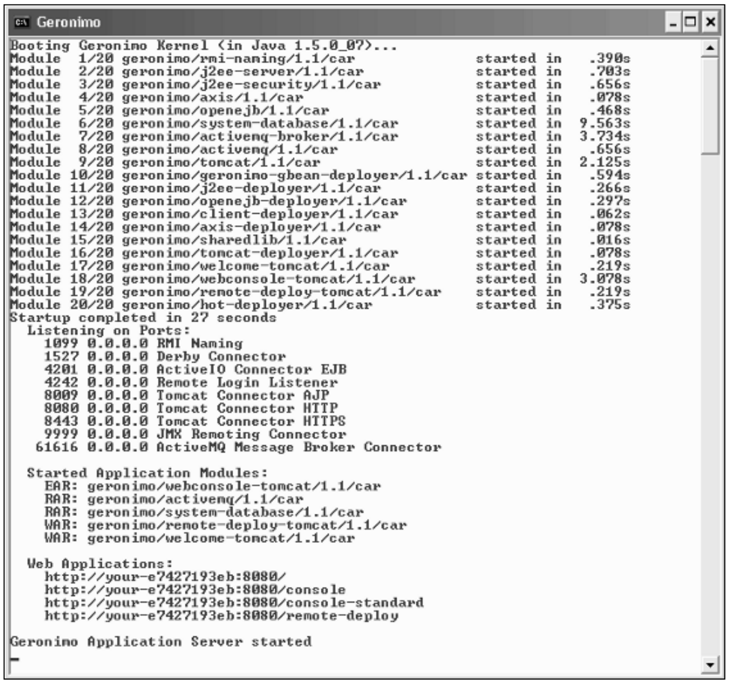
Note the `load` attribute of the module is set to `"false"`, telling the Geronimo server not to load the CORBA support module during startup. This is by default — and you definitely want to prevent the CORBA support module from loading if you are using the JDK 5 VM (Java SE 5).

Running the Geronimo Server

Once you've installed Geronimo using the binary archives, you can start the server. First, change directory to the installation directory that you have selected. From the installation directory, change directory to the `bin` subdirectory. From the `bin` subdirectory, type in the following command:

```
startup
```

On an Unix system, you run the `startup.sh` script using `sh startup.sh`. This will start the Geronimo server running the core J2EE services. The startup status is displayed on another console window, or in the backgroup with Unix systems. The name of each module is displayed as it is started, as well as the time it took in starting it. Once the server has started up completely, your console should look similar to Figure 1-3.



```
Booting Geronimo Kernel (in Java 1.5.0_07)...
Module  1/20 geronimo/rmi-naming/1.1/car             started in   .390s
Module  2/20 geronimo/j2ee-server/1.1/car            started in   .703s
Module  3/20 geronimo/j2ee-security/1.1/car          started in   .656s
Module  4/20 geronimo/axis/1.1/car                   started in   .078s
Module  5/20 geronimo/openejb/1.1/car                started in   .468s
Module  6/20 geronimo/system-database/1.1/car        started in  9.563s
Module  7/20 geronimo/activemq-broker/1.1/car        started in  3.734s
Module  8/20 geronimo/activemq/1.1/car               started in   .656s
Module  9/20 geronimo/tomcat/1.1/car                 started in  2.125s
Module 10/20 geronimo/geronimo-gbean-deployer/1.1/car started in   .594s
Module 11/20 geronimo/j2ee-deployer/1.1/car          started in   .266s
Module 12/20 geronimo/openejb-deployer/1.1/car       started in   .297s
Module 13/20 geronimo/client-deployer/1.1/car        started in   .062s
Module 14/20 geronimo/axis-deployer/1.1/car          started in   .078s
Module 15/20 geronimo/sharedlib/1.1/car              started in   .016s
Module 16/20 geronimo/tomcat-deployer/1.1/car        started in   .078s
Module 17/20 geronimo/welcome-tomcat/1.1/car         started in   .219s
Module 18/20 geronimo/webconsole-tomcat/1.1/car      started in  3.078s
Module 19/20 geronimo/remote-deploy-tomcat/1.1/car   started in   .219s
Module 20/20 geronimo/hot-deployer/1.1/car           started in   .375s
Startup completed in 27 seconds
  Listening on Ports:
    1099 0.0.0.0 RMI Naming
    1527 0.0.0.0 Derby Connector
    4201 0.0.0.0 ActiveIO Connector EJB
    4242 0.0.0.0 Remote Login Listener
    8009 0.0.0.0 Tomcat Connector AJP
    8080 0.0.0.0 Tomcat Connector HTTP
    8443 0.0.0.0 Tomcat Connector HTTPS
    9999 0.0.0.0 JMX Remoting Connector
   61616 0.0.0.0 ActiveMQ Message Broker Connector

  Started Application Modules:
    EAR: geronimo/webconsole-tomcat/1.1/car
    RAR: geronimo/activemq/1.1/car
    RAR: geronimo/system-database/1.1/car
    WAR: geronimo/remote-deploy-tomcat/1.1/car
    WAR: geronimo/welcome-tomcat/1.1/car

  Web Applications:
    http://your-e7427193eb:8080/
    http://your-e7427193eb:8080/console
    http://your-e7427193eb:8080/console-standard
    http://your-e7427193eb:8080/remote-deploy

Geronimo Application Server started
_
```

Figure 1-3: Startup screen for server

Table 1-3 describes some of the services that are started by Geronimo 1.1. . The port for each service will reflect any customization you may have made in `config.xml`. The list you see may be different if you are using versions beyond 1.1.

Table 1-3: Core J2EE Services Started by Geronimo

Service	Port
Geronimo Naming Service (RMI-based)	1099
OpenEJB EJB container (ActiveIO)	4201
Remote authentication (login) server	4242
Tomcat container connectors	8080 (HTTP) , 8443 (secured socket - HTTPS), and 8009 (AJP 13)
Jetty container connectors	8080 (HTTP) and 8443 (HTTPS) and 9090 (HTTP) and 8009 (AJP 13)
JMX Remoting (Management) Connector	9999
ActiveMQ	61616
Derby RDBMS system	1527

To shut down your Geronimo server run the `shutdown` (or `shutdown.sh`) script from the `bin` subdirectory. It will prompt you with userid (enter `system`) and password (enter `manager`). You can also shut down Geronimo from the Web console (covered in Chapter 8).

Verifying Your Installation of Geronimo

To verify that everything is running well with your Geronimo installation, you can use the deployer tool (a tool that you will use a lot, and find out a lot more about in Chapter 6) to list all the system configurations that are a built-in part of your server.

From the `bin` subdirectory under your installation directory, type the following command:

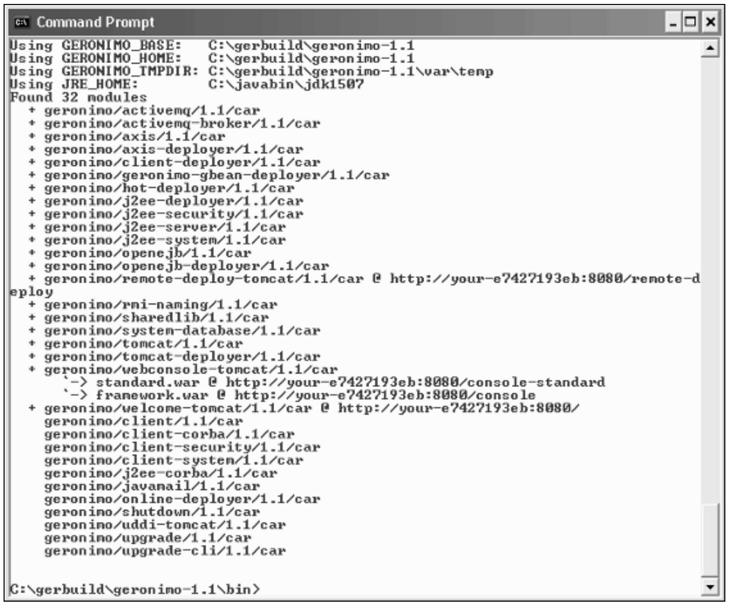
```
deploy --user system --password manager list-modules
```

This is how you access the deployer tool. You must supply the username `system` and password `manager`. This is set up by the default configuration at installation.

After installation, if you need to change the password, edit the `users.properties` file under the `<installation directory>/geronimo-1.1/var/security` directory. Chapter 8 provides more details on how to use the Web console to add new administrative users.

Of course, you can save the repetitive step of changing directory to the `bin` subdirectory if you add that directory into your `PATH` environment variable.

The `list-modules` command of the deployer tool will display all the system configurations in the server. After an initial installation, your `list-modules` command output should be similar to Figure 1-4.



Command Prompt

```
Using GERONIMO_BASE:   C:\gerbuild\geronimo-1.1
Using GERONIMO_HOME:   C:\gerbuild\geronimo-1.1
Using GERONIMO_TMPDIR: C:\gerbuild\geronimo-1.1\var\temp
Using JRE_HOME:        C:\javabin\jdk1507
Found 32 modules
  + geronimo/activemq/1.1/car
  + geronimo/activemq-broker/1.1/car
  + geronimo/axis/1.1/car
  + geronimo/axis-deployer/1.1/car
  + geronimo/client-deployer/1.1/car
  + geronimo/geronimo-gbean-deployer/1.1/car
  + geronimo/hot-deployer/1.1/car
  + geronimo/j2ee-deployer/1.1/car
  + geronimo/j2ee-security/1.1/car
  + geronimo/j2ee-server/1.1/car
  + geronimo/j2ee-system/1.1/car
  + geronimo/openejb/1.1/car
  + geronimo/openejb-deployer/1.1/car
  + geronimo/remote-deploy-tomcat/1.1/car @ http://your-e7427193eb:8080/remote-d
eploy
  + geronimo/rmi-naming/1.1/car
  + geronimo/sharedlib/1.1/car
  + geronimo/system-database/1.1/car
  + geronimo/tomcat/1.1/car
  + geronimo/tomcat-deployer/1.1/car
  + geronimo/webconsole-tomcat/1.1/car
      `-> standard.war @ http://your-e7427193eb:8080/console-standard
      `-> framework.war @ http://your-e7427193eb:8080/console
  + geronimo/welcome-tomcat/1.1/car @ http://your-e7427193eb:8080/
    geronimo/client/1.1/car
    geronimo/client-corba/1.1/car
    geronimo/client-security/1.1/car
    geronimo/client-system/1.1/car
    geronimo/j2ee-corba/1.1/car
    geronimo/javamail/1.1/car
    geronimo/online-deployer/1.1/car
    geronimo/shutdown/1.1/car
    geronimo/uddi-tomcat/1.1/car
    geronimo/upgrade/1.1/car
    geronimo/upgrade-cli/1.1/car

C:\gerbuild\geronimo-1.1\bin>
```

Figure 1-4: Output from list-modules command

Each line in Figure 1-4 represents a module deployed in the Geronimo server. The ones with a "+" sign on the left are the ones that have been started, and are currently running. The URL to some of the Web-accessible modules is also shown.

Congratulations, you have just successfully installed and run Geronimo!

Summary

In this chapter, you have successfully installed and tested Geronimo on your own system.

At the official Geronimo Web site, a variety of builds and releases of Geronimo may be available for downloading. The latest builds can be useful for accessing the latest features or bug fixes. However, production purposes, you should stick with official releases.

Geronimo can run on almost any modern machine, including x86 machines running Windows or Unix, or Linux of many flavors. Mac OSX, and Sun Solaris also support the server. You must use SUN's JDK 1.4.2 i JVM for Geronimo if you need CORBA support. Otherwise, Geronimo works great on JDK 5 (Java SE 5).

Geronimo installation on both Windows and Unix machine can be as simple as unarchiving the binary distribution to a directory of your choice. In Chapter 2, you will see how easy it is to deploy an enterprise application to your working Geronimo server.