COP 4610L: Applications in the Enterprise Spring 2006

Introduction to Servlet Technology– Part 1

Instructor : Mark Llewellyn
markl@cs.ucf.edu
CSB 242, 823-2790
http://www.cs.ucf.edu/courses/cop4610L/spr2006

School of Electrical Engineering and Computer Science
University of Central Florida

Client-Server Relationship Revisited

- In a client-server relationship, the client requests that some action be performed and the server performs the action and responds to the client.
- This request-response model of communication is the foundation for the highest-level view of networking in Java – servlets and JavaServer Pages (JSP).
- A servlet extends the functionality of a server, such as a Web server that serves Web pages to a user's browser using the HTTP protocol. A servlet can almost be thought of as an applet that runs on the server side--without a face. Java servlets make many Web applications possible.
- Packages `javax.servlet` and `javax.servlet.http` provide the classes and interfaces to define servlets. Packages `javax.servlet.jsp` and `javax.servlet.jsp.tagext` provide the classes and interfaces that extend the servlet capabilities for JSP.

Client-Server Relationship Revisited (cont.)

- Using special syntax, JSP allows Web-page implementers to create pages that encapsulate Java functionality and even to write scriplets of actual Java code directly into the page.
- A common implementation of the request-response model is between Web browsers and Web servers. When a user selects a Web site to browse through the browser (the client application), a request is sent to the appropriate Web server (the server application). The server normally responds to the client by sending the appropriate XHTML Web page.
- Servlets are effective for developing Web-based solutions that help provide secure access to a Web site, interact with databases on behalf of a client, dynamically generate custom XHTML documents to be displayed by browsers and maintain unique session information for each client.

Static and Dynamic Web Content

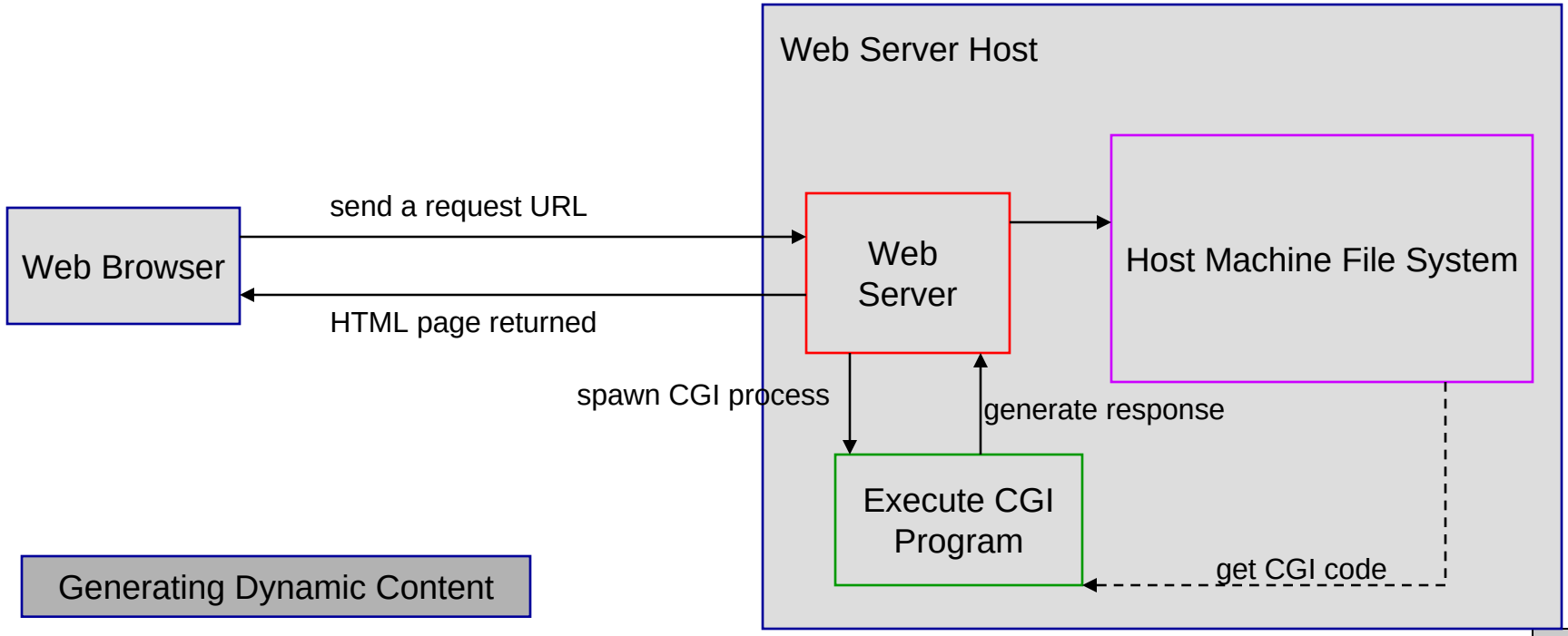
- Consider how a web page is displayed by a browser.
  - Typically, the web page is created using HTML and stored as a file on the web server. A user enters a URL for the file from a web browser. The browser contacts the web server and requests the file. The server finds the file and returns it to the browser. The browser then displays the file for the user.
- Static information is stored in HTML files. The HTML files can be updated, but at any given time, every request for the same file returns exactly the same content. The contents do not change regardless of who requested the file.

Static and Dynamic Web Content (cont.)

- Not all information, however, is static in nature. Recall, the first programming assignment. The HTML pages that generated this information must be generated dynamically.
- Dynamic web pages are generated by web server. The web server will execute certain programs to process user requests from browsers in order to produce a customized response.
- The Common Gateway Interface (CGI) was proposed to generate dynamic web content. The interface provides a standard framework for web servers to interact with external program known as CGI programs.

CGI Programming

- When a web server receives a request from a browser it passes it to the CGI program. The CGI program processes the request and generates a response at runtime. CGI programs can be written in any language, but Perl is the most popular choice.



The GET and POST Methods

- The two most common HTTP requests, also known as methods, are `GET` and `POST`.
- The web browser issues a request using a URL or an HTML from to trigger the web server to execute a CGI program. (We'll deal with forms later.) When issuing a CGI request directly from a URL, the `GET` method is used.
- This form of a URL is known as a query string. The URL query string consists of the location of the CGI program, parameters, and their values.
- When issuing a request from an HTML form, either a `GET` or `POST` method can be used.

The GET and POST Methods (cont.)

- The form explicitly specifies which of the two is used.
- If the GET method is used, the data in the form are appended to the request string as if they were submitted using a URL.
- If the POST method is used, the data in the form are packaged as part of the request file. The server program obtains the data by reading the file.

The GET and POST methods both send requests to the web server. The POST method always triggers the execution of the corresponding CGI program. The GET method may not cause the CGI program to be executed if the previous same request is cached in the web browser. Browsers often cache web pages so that the same request can be quickly responded to without contacting the web server. The browser checks the request sent through the GET method as a URL query string. If the results for the exact same URL are cached on a disk, then the previous web page for the URL may be displayed. To ensure that a new web page is always displayed, use the POST method.

From CGI To Java Servlets

- CGI provides a relatively simple approach for creating dynamic web applications that accept a user request, process it on the server side, and return responses to the user's browser.
- However, CGI is extremely slow when handling a large number of requests simultaneously, because the web server must spawn a process for executing each CGI program.
- Java servlets were developed to remedy the performance problem of CGI programs. Java servlets are basically Java programs that behave like CGI programs.

Java Servlets

- Java servlets are executed upon request from a web browser.
- All servlets execute inside a servlet container, also referred to as a servlet server or a servlet engine.
- A servlet container is a single process that runs a JVM (Java Virtual Machine). The JVM creates a thread to handle each servlet (recall that threads have considerably less overhead than full-blown processes). All the threads share the same memory allocated to the JVM. Since the JVM persists beyond the lifecycle of a single servlet execution, servlets can share objects already created in the JVM.
  - For example, if multiple servlets access the same database, they can share the connection object.

Thin Clients

- Servlets are the ideal solution for database-intensive applications that communicate with thin clients.
  - Thin clients are applications that provide presentation but do not process data, thus requiring few computing resources.
- The server is responsible for database access. Clients connect to the server using standard protocols available on most client platforms. The presentation-logic code for generating dynamic content can be written once and reside on the server for access by clients, to allow programmers to create efficient thin clients.

Apache Tomcat Server

- Sun Microsystems, through the Java Community Process is responsible for the development of the servlet and JSP specifications.
- To run Java servlets, you need a servlet container. While many servlet containers are available, the reference implementation of both these standards developed by the Apache Software Foundation (www.apache.org) is known as Tomcat.
- Tomcat was developed as part of the Jakarta Project. The Jakarta Project contains many subprojects designed to help commercial server-side developers.
- Tomcat became a top-level project at Apache in early October 2005.
- Tomcat is the official reference implementation of the JSP and servlet standards. Tomcat can be used standalone as a web server or plugged into a web server like Apache, IIS, etc.. The current stable implementation is Tomcat 5.5.15 (as of March 2006).

Servlet Overview and Architecture

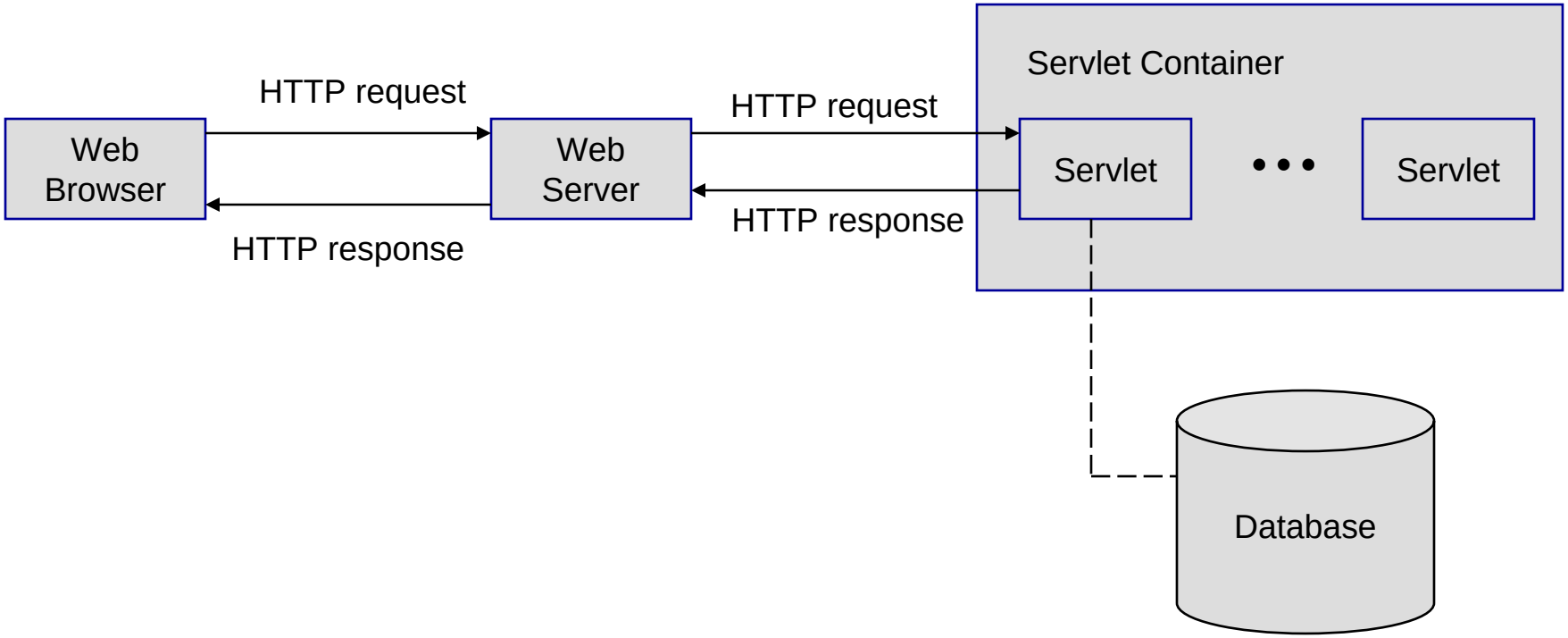
- The Internet offers many protocols. The HTTP (Hypertext Transfer Protocol) that forms the basis of the WWW uses URLs (Uniform Resource Locators) to locate resources on the Internet.
- URLs can represent files or directories and can represent complex tasks such as database lookups and Internet searches.
- JSP technology, basically an extension of servlet technology, simplifies the process of creating pages by separating presentation from content.
- Typically, JSPs are used when most of the content sent to the client is static text and markup, and only a small portion of the content is generated dynamically with Java code.

Servlet Overview and Architecture (cont.)

- Servlets are more commonly used when a small portion of the content sent to the client is static text or markup. In fact, some servlets do not produce content. Rather, they perform a task on behalf of the client, then invoke other servlets or JSPs to provide a response.
- Note that in most cases servlet and JSP technologies are interchangeable.
- The server that executes a servlet is referred to as the servlet container or servlet engine.
- Servlets and JSP have become so popular that they are now supported directly or with third-party plug-ins by most major Web servers and application servers (servers that execute applications to generate dynamic Web pages in response to requests).

Servlet Overview and Architecture (cont.)

- We'll look at servlets that implement the request-response model between clients and servers using the HTTP protocol. This architecture is shown in the diagram below.



Servlet Overview and Architecture (cont.)

Explanation of the architecture diagram on previous page

- A client application sends an HTTP request to the server.
- The servlet container receives the request and directs it to be processed by the appropriate servlet.
- The servlet does its processing, which may include interacting with a database or other server-side components, such as other servlets or JSPs.
- The servlet returns its results to the client – normally in the form of an HTML, XHTML, or XML document to display in a browser.

Interface Servlet and the Servlet Lifecycle

- Architecturally speaking, all servlets must implement the `Servlet` interface of package `javax.servlet`.
- The methods of interface `Servlet` are invoked by the servlet container. This interface declares five methods which deal with the execution of a servlet. These methods are shown on the next page. For the details see: www.java.sun.com/j2ee/1.4/docs/api/javax/servlet/Servlet.html
- A servlet's life cycle begins when the servlet container loads it into memory – normally, in response to the first request for the servlet.
- Before the servlet can handle that request, the container invokes the servlet's `init` method.

Methods of the Servlet Interface

Method	Description
destroy( )	Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.
getServletConfig( )	Returns a ServletConfig object, which contains initialization and startup parameters for this servlet.
getServletInfo( )	Returns information about the servlet, such as author, version, and copyright.
init( )	Called by the servlet container to indicate to a servlet that the servlet is being placed into service.
service( )	Called by the servlet container to allow the servlet to respond to a request.

The Servlet Lifecycle

- After `init` completes execution, the servlet can respond to its first request.
- All requests are handled by the a servlet's `service` method, which receives the request, processes it and sends a response to the client.
- During the servlet's lifecycle, the method `service` is invoked once per request. Each new request is typically handled in a separate thread of execution (managed by the servlet container) in which method `service` executes.
- When the servlet container terminates the servlet (whenever the servlet needs more memory or when it is shutdown), the servlet's `destroy` method is invoked to release servlet resources.

Setting Up Tomcat

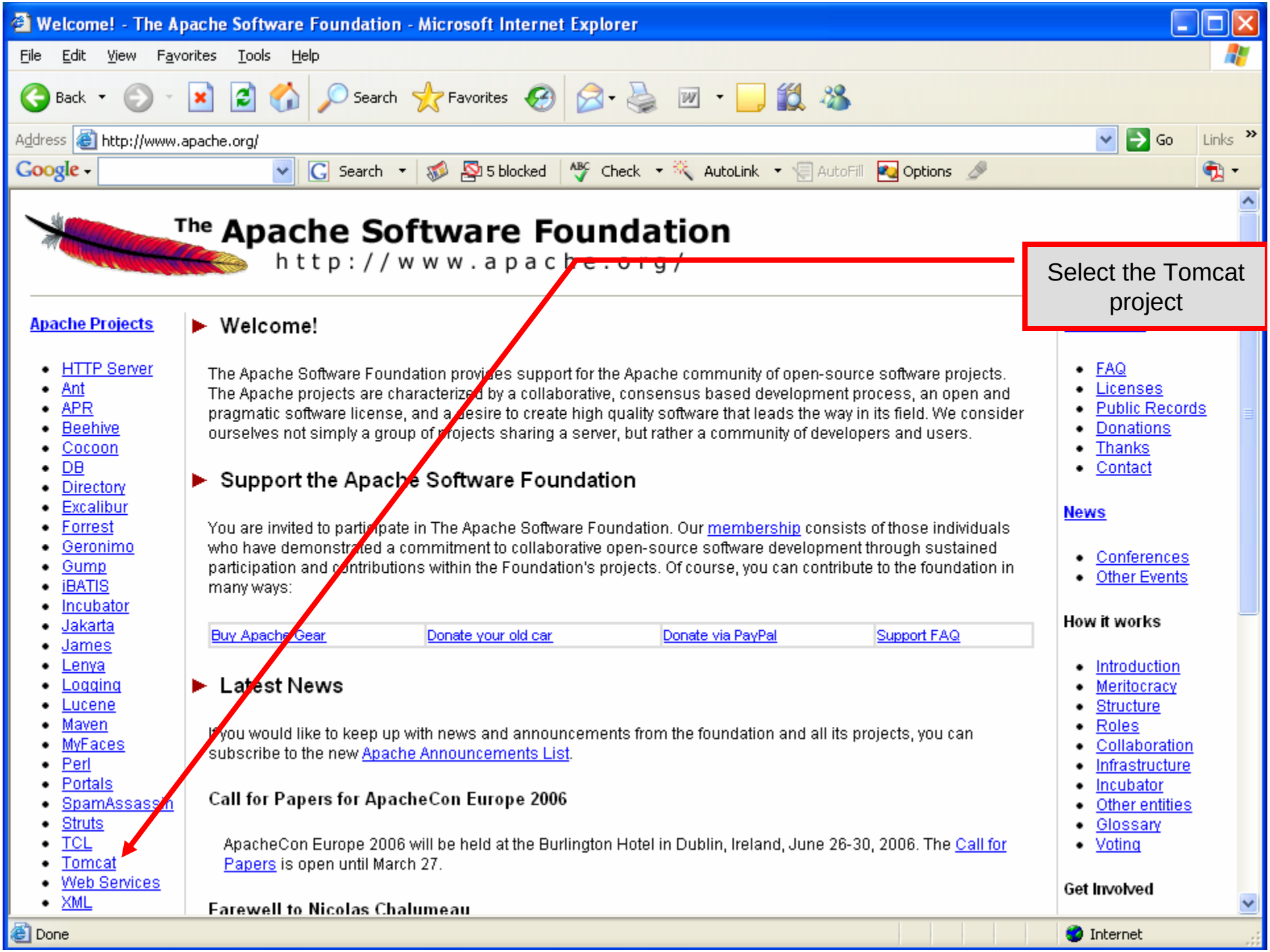
- Tomcat is a fully functional implementation of servlets and JSP. It includes a Web server, so it can be used as a standalone test container for servlets and JSPs.

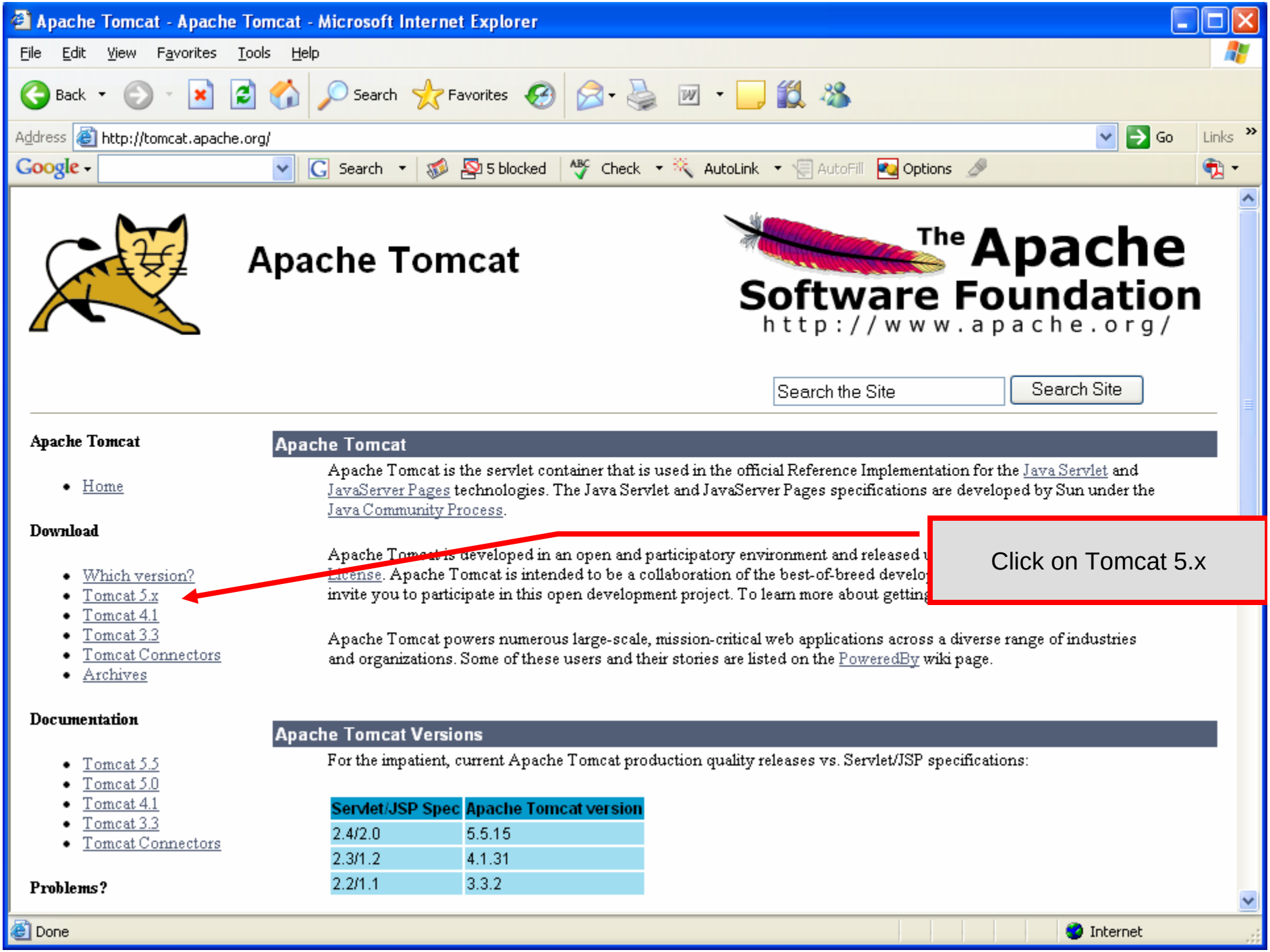
- The current stable version is 5.5.15 available from www.apache.org. This version was declared stable on January 4, 2006.

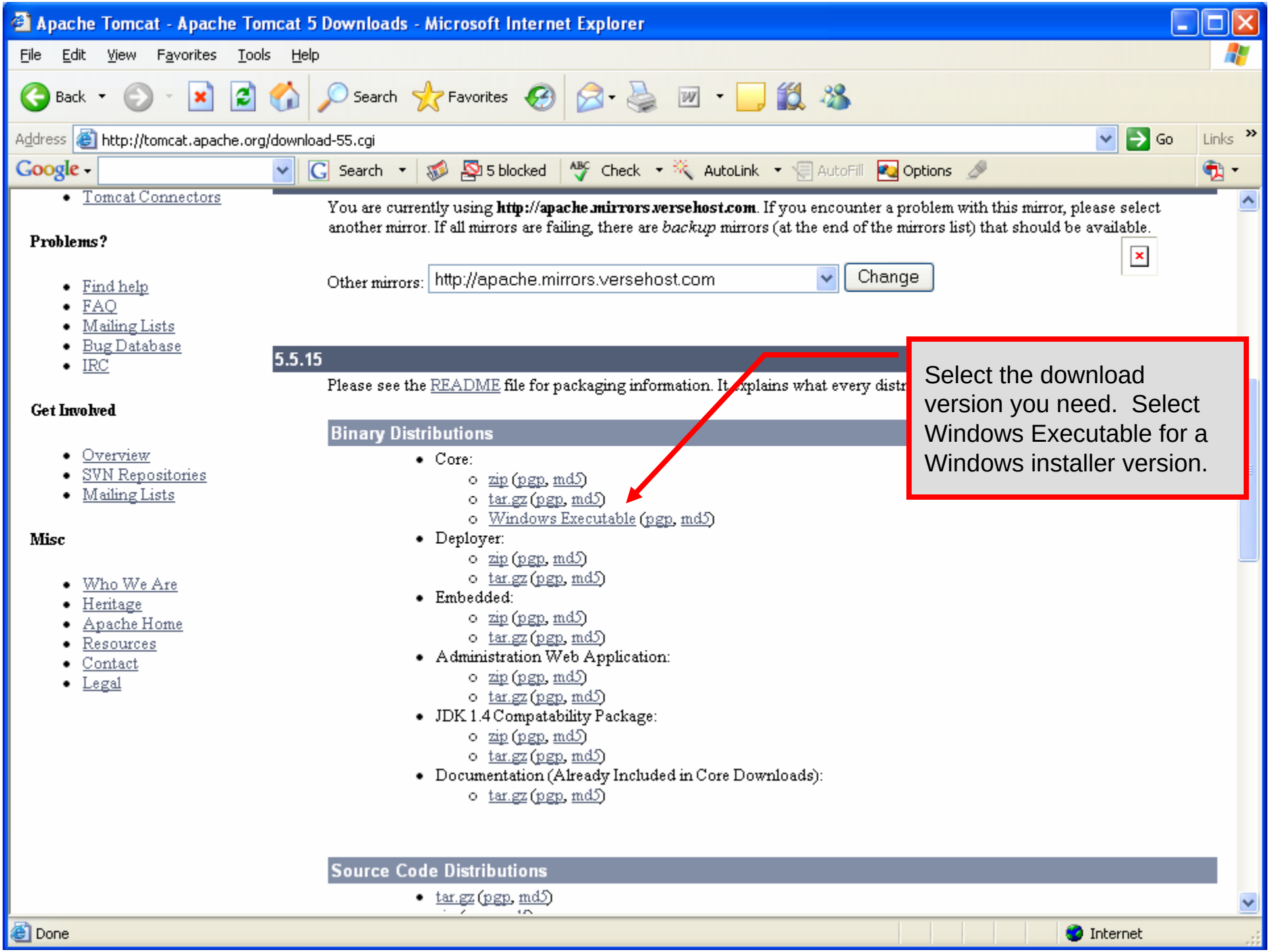
1. Select the Tomcat page from the menu on the left-hand side of the screen. As shown on page 21.

2. Once in the Tomcat project, select Download binaries from the left-hand side of the screen as shown on page 22.

3. Once in the download binaries screen, select the option of your choice. This is shown on page 23.


Welcome! - The Apache Software Foundation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.apache.org/
Go Links
Google Search 5 blocked Check AutoLink AutoFill Options
The Apache Software Foundation
http://www.apache.org/
Apache Projects
HTTP Server
Ant
APR
Beehive
Cocoon
DB
Directory
Excalibur
Forrest
Geronimo
Gump
iBATIS
Incubator
Jakarta
James
Lenya
Logging
Lucene
Maven
MyFaces
Perl
Portals
SpamAssassin
Struts
TCL
Tomcat
Web Services
XML
Welcome!
The Apache Software Foundation provides support for the Apache community of open-source software projects. The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field. We consider ourselves not simply a group of projects sharing a server, but rather a community of developers and users.
Support the Apache Software Foundation
You are invited to participate in The Apache Software Foundation. Our membership consists of those individuals who have demonstrated a commitment to collaborative open-source software development through sustained participation and contributions within the Foundation's projects. Of course, you can contribute to the foundation in many ways:
Buy Apache Gear
Donate your old car
Donate via PayPal
Support FAQ
Latest News
If you would like to keep up with news and announcements from the foundation and all its projects, you can subscribe to the new Apache Announcements List.
Call for Papers for ApacheCon Europe 2006
ApacheCon Europe 2006 will be held at the Burlington Hotel in Dublin, Ireland, June 26-30, 2006. The Call for Papers is open until March 27.
Farewell to Nicolas Chalumeau
FAQ
Licenses
Public Records
Donations
Thanks
Contact
News
Conferences
Other Events
How it works
Introduction
Meritocracy
Structure
Roles
Collaboration
Infrastructure
Incubator
Other entities
Glossary
Voting
Get Involved
Done
Internet
Select the Tomcat project


Apache Tomcat - Apache Tomcat - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://tomcat.apache.org/
Go Links
Google Search 5 blocked Check AutoLink AutoFill Options
Apache Tomcat
The Apache Software Foundation
http://www.apache.org/
Search the Site
Search Site
Apache Tomcat
Home
Download
Which version?
Tomcat 5.x
Tomcat 4.1
Tomcat 3.3
Tomcat Connectors
Archives
Documentation
Tomcat 5.5
Tomcat 5.0
Tomcat 4.1
Tomcat 3.3
Tomcat Connectors
Problems?
Apache Tomcat
Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process.
Apache Tomcat is developed in an open and participatory environment and released
License. Apache Tomcat is intended to be a collaboration of the best-of-breed develo
invite you to participate in this open development project. To learn more about gettin
Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.
Click on Tomcat 5.x
Apache Tomcat Versions
For the impatient, current Apache Tomcat production quality releases vs. Servlet/JSP specifications:
Servlet/JSP Spec
Apache Tomcat version
2.4/2.0
5.5.15
2.3/1.2
4.1.31
2.2/1.1
3.3.2
Done
Internet


Apache Tomcat - Apache Tomcat 5 Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://tomcat.apache.org/download-55.cgi
Go Links
Google Search 5 blocked Check AutoLink AutoFill Options
Tomcat Connectors
Problems?
Find help
FAQ
Mailing Lists
Bug Database
IRC
Get Involved
Overview
SVN Repositories
Mailing Lists
Misc
Who We Are
Heritage
Apache Home
Resources
Contact
Legal
You are currently using http://apache.mirrors.versehost.com. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are backup mirrors (at the end of the mirrors list) that should be available.
Other mirrors: http://apache.mirrors.versehost.com Change
5.5.15
Please see the README file for packaging information. It explains what every distr
Binary Distributions
Core:
zip (pgp, md5)
tar.gz (pgp, md5)
Windows Executable (pgp, md5)
Deployer:
zip (pgp, md5)
tar.gz (pgp, md5)
Embedded:
zip (pgp, md5)
tar.gz (pgp, md5)
Administration Web Application:
zip (pgp, md5)
tar.gz (pgp, md5)
JDK 1.4 Compatability Package:
zip (pgp, md5)
tar.gz (pgp, md5)
Documentation (Already Included in Core Downloads):
tar.gz (pgp, md5)
Source Code Distributions
tar.gz (pgp, md5)
Select the download version you need. Select Windows Executable for a Windows installer version.
Done
Internet

Setting Up Tomcat

- Once you've downloaded and installed Tomcat you're ready to run a demonstration test that will tell you if you've got everything set-up properly.

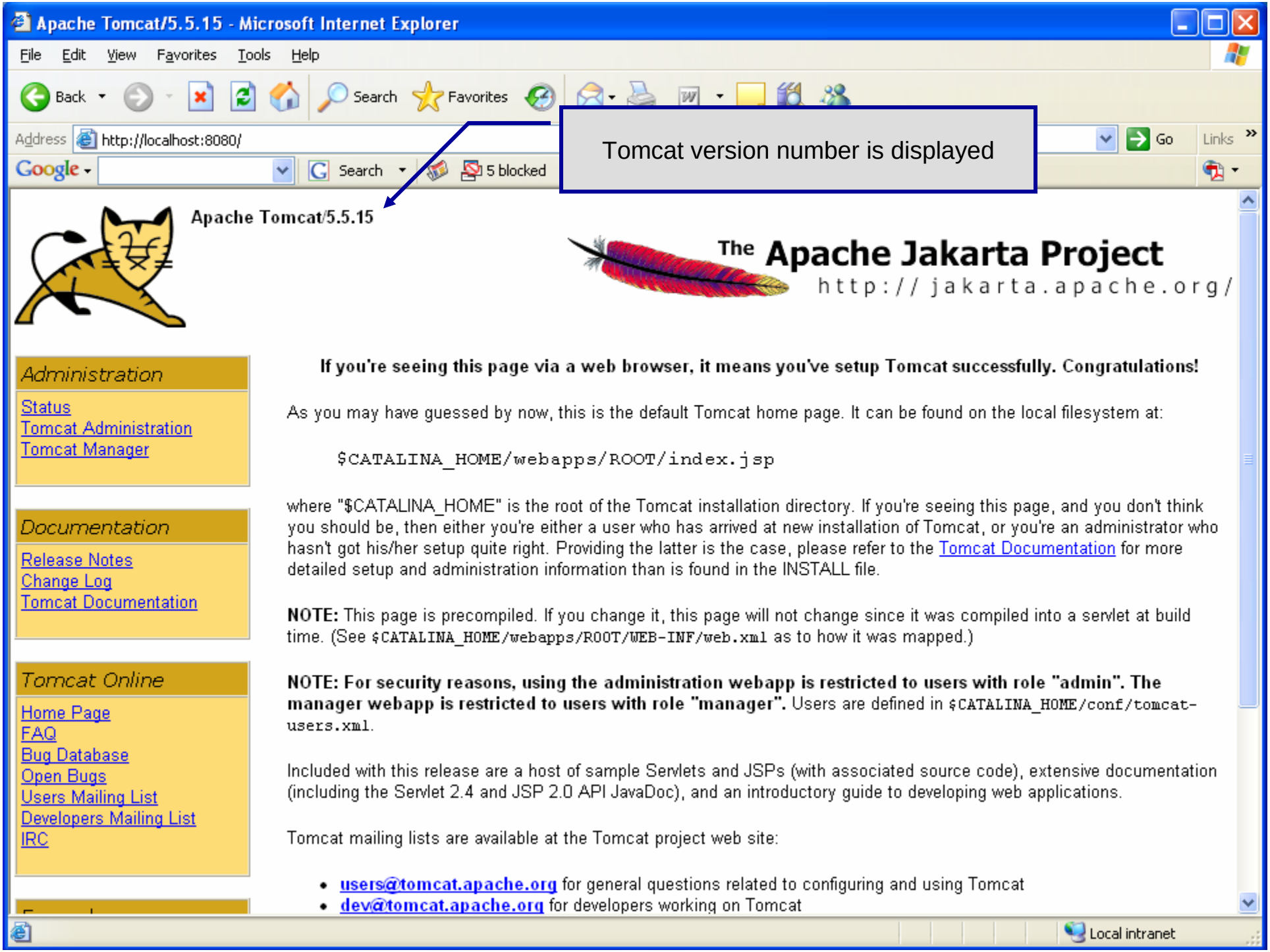
NOTE: During the install, Tomcat will ask you which TCP port Tomcat should run on. To avoid any conflict with standard Web servers which default to TCP port 80, Tomcat is set to default to TCP port 8080. If you have any other service running on this port change the port number at this time to one on which no conflict will occur.

In all subsequent examples, I'm running Tomcat on TCP port 8080.

Starting Up Tomcat

- Once Tomcat is installed, you need to start it as a service. On Windows machines, the current versions of Tomcat are installed as a service that will start when Windows starts. On Unix/Linux a startup.sh file is included so you just type startup (assuming you are in the bin directory where you located Tomcat).

1. Start Tomcat running.
2. Start your Web browser.
3. Enter URL: http://localhost:8080
4. You should see the screen on the following page if everything is set up ok.


Apache Tomcat/5.5.15 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/
Go Links
Google Search 5 blocked
Tomcat version number is displayed
Apache Tomcat/5.5.15
The Apache Jakarta Project
http://jakarta.apache.org/
Administration
Status
Tomcat Administration
Tomcat Manager
Documentation
Release Notes
Change Log
Tomcat Documentation
Tomcat Online
Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC
If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!
As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:
$CATALINA_HOME/webapps/ROOT/index.jsp
where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.
NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See $CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml as to how it was mapped.)
NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in $CATALINA_HOME/conf/tomcat-users.xml.
Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.
Tomcat mailing lists are available at the Tomcat project web site:
users@tomcat.apache.org for general questions related to configuring and using Tomcat
dev@tomcat.apache.org for developers working on Tomcat
Local intranet