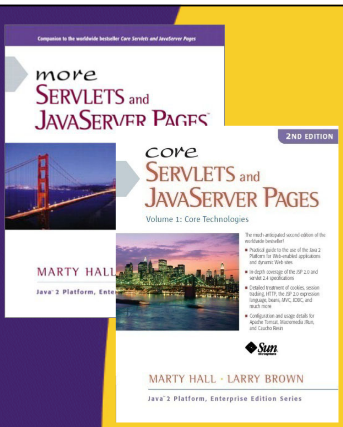
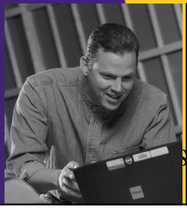
An Overview of Servlet & JSP Technology

Originals of Slides and Source Code for Examples:
http://courses.coreservlets.com/Course-Materials/csajsp2.html





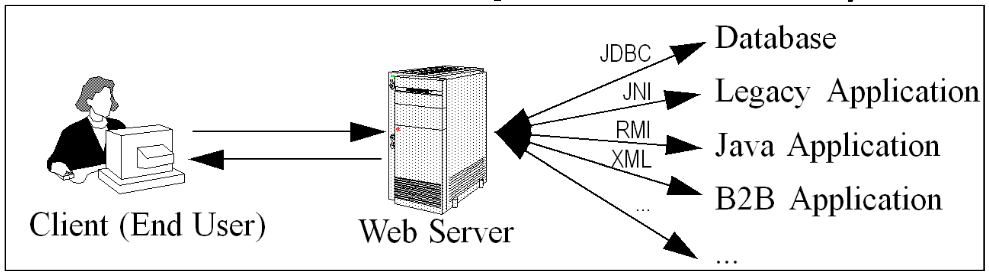
Agenda

- **What servlets and JSP are all about**
  - Understanding the role of servlets
  - Building Web pages dynamically
  - Evaluating servlets vs. other technologies
  - Understanding the role of JSP
- **Testing with Eclipse**
  - Basic server
  - HTML/JSP
  - Servlets
- **Testing manually**
  - Basic server
  - HTML/JSP
  - Servlets

What Servlets and JSP are All About

A Servlet's Job

- **Read explicit data sent by client (form data)**
- **Read implicit data sent by client (request headers)**
- **Generate the results**
- **Send the explicit data back to client (HTML)**
- **Send the implicit data to client (status codes and response headers)**



Why Build Web Pages Dynamically?

- **The Web page is based on data submitted by the user**
  - E.g., results page from search engines and order-confirmation pages at on-line stores
- **The Web page is derived from data that changes frequently**
  - E.g., a weather report or news headlines page
- **The Web page uses information from databases or other server-side sources**
  - E.g., an e-commerce site could use a servlet to build a Web page that lists the current price and availability of each item that is for sale.

The Advantages of Servlets Over “Traditional” CGI

- **Efficient**
  - Threads instead of OS processes, one servlet copy
- **Convenient**
  - Lots of high-level utilities
- **Powerful**
  - Sharing data, pooling, persistence
- **Portable**
  - Run on virtually all operating systems and servers
- **Inexpensive**
  - There are plenty of free and low-cost servers
- **Secure**
  - No shell escapes, no buffer overflows
- **Mainstream**
  - See next page

Mainstream

- **Popular:**
  - The single most common use of Java technology
  - The leading technology for medium/large Web applications
    - Google reports over 500 million Web pages using JSP
- **Supported by:**
  - Apache, Oracle, IBM, Sybase, BEA, Macromedia, Caucho, Sun/iPlanet, New Atlanta, ATG, Fujitsu, Lutris, Silverstream, the World Wide Web Consortium (W3C), and many others
  - Plugins for IIS and Zeus
- **Runs on:**
  - Windows, Unix/Linux, MacOS, VMS, and IBM mainframe OSs
- **Used for:**
  - Airline companies, hotels, e-commerce sites, search engines, banks, financial sites, etc., etc., etc.

Ten Most Popular Web Sites (Alexa.com, Fall 2008)

1. **Google**
   – Custom technology, some Java
2. **Yahoo**
   – PHP and Java
3. **MySpace**
   – ColdFusion (Java "under the hood")
4. **YouTube**
   – Flash, Python, Java
5. **Facebook**
   – PHP
6. **Windows Live Search**
   – .NET
7. **MSN (Microsoft Network)**
   – .NET
8. **Wikipedia**
   – PHP
9. **Ebay**
   – Java
10. **AOL**
    – Java

Extending the Power of Servlets: JavaServer Pages (JSP)

- **Idea:**
  – Use regular HTML for most of page
  – Mark dynamic content with special tags
  – Details in second half of course

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD>
<BODY>
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<%= coreservlets.Utils.getUserNameFromCookie(request) %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A></SMALL>
<P>
Regular HTML for  rest of on-line store's Web page
</BODY></HTML>
```

Accessing the Online Documentation

- **Servlets and JSP**
  - http://java.sun.com/products/servlet/2.5/docs/servlet-2_5-mr2/
  - http://java.sun.com/products/jsp/2.1/docs/jsp-2_1-pfd2/
  - http://tomcat.apache.org/tomcat-5.5-doc/servletapi/
  - http://tomcat.apache.org/tomcat-5.5-doc/jspapi/
- **Java 6 or Java 5**
  - http://java.sun.com/javase/6/docs/api/
    - Class uses Java 6 and Tomcat 6
  - http://java.sun.com/j2se/1.5.0/docs/api/
- **Advice**
  - If you have a fast and reliable internet connection, bookmark these addresses
  - If not, download a copy of the APIs onto your local machine and use it

Setting Up Tomcat on Your PC

- **With regular Eclipse**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/eclipse.html
    - More details in next section of this tutorial
- **With MyEclipse**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/myeclipse.html
- **For manual execution**
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/
    - More details in last section.
    - Eclipse or another IDE strongly recommended over manual usage

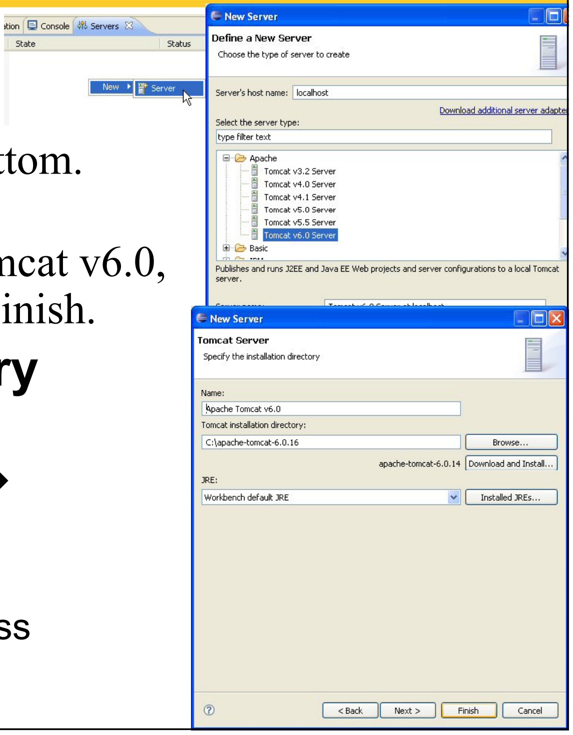
Testing with Eclipse

Installing Eclipse

- **Overview**
  - Eclipse is a free open-source development environment with support for Java and many other languages
- **Downloading**
  - http://www.eclipse.org/downloads/
    - Choose “Eclipse IDE for Java EE Developers”
    - As of 4/2009, version 3.4, called Eclipse Ganymede
- **Installing**
  - Unzip into directory of your choice
  - Put shortcut to eclipse.exe on your desktop
- **Integrating Tomcat in Eclipse**
  - http://www.coreservlets.com/ Apache-Tomcat-Tutorial/eclipse.html

Configuring Eclipse

- **Make sure Eclipse knows about Tomcat**
  - Click on Servers tab at bottom. R-click in window.
  - New, Server, Apache, Tomcat v6.0, Next, navigate to folder, Finish.
- **Suppress unnecessary compiler warnings**
  - Window → Preferences → Java → Compiler → Errors/Warnings
    - Change “Serializable class without ...” to “Ignore”



Define a New Server

Choose the type of server to create

Server's host name: localhost

Select the server type: type filter text

Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Tomcat Server

Specify the installation directory

Name: Apache Tomcat v6.0

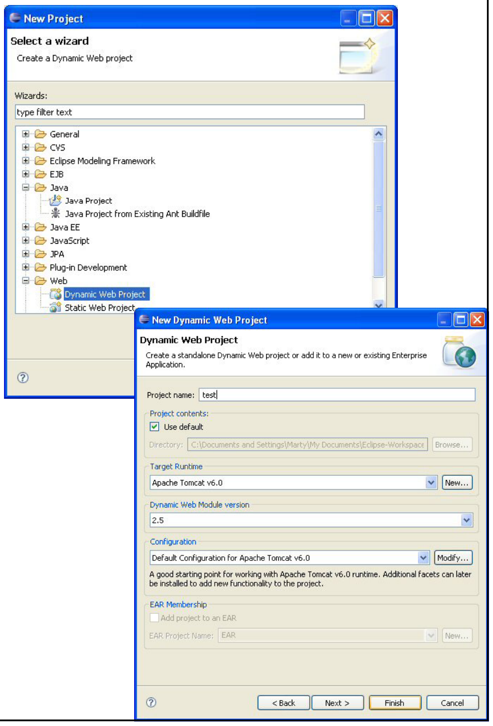
Tomcat installation directory: C:\apache-tomcat-6.0.16

apache-tomcat-6.0.14

JRE: Workbench default JRE

Making Web Apps in Eclipse

- **Make empty project**
  - File → New → Project → Web → Dynamic Web Project
  - Give it a name (e.g., “test”)
  - Accept all other defaults
- **Shortcut**
  - If you have made Dynamic Web Project recently in workspace, you can just do File → New → Dynamic Web Project



Select a wizard

Create a Dynamic Web project

Wizards: type filter text

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: test

Project contents: [✓ Use default]

Directory: C:\Documents and Settings\Marty\My Documents\Eclipse-Workspace

Target Runtime: Apache Tomcat v6.0

Dynamic Web Module version: 2.5

Configuration: Default Configuration for Apache Tomcat v6.0

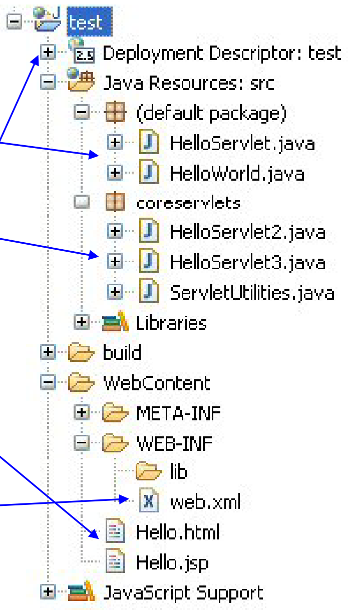
A good starting point for working with Apache Tomcat v6.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR Membership: Add project to an EAR

EAR Project Name: EAR

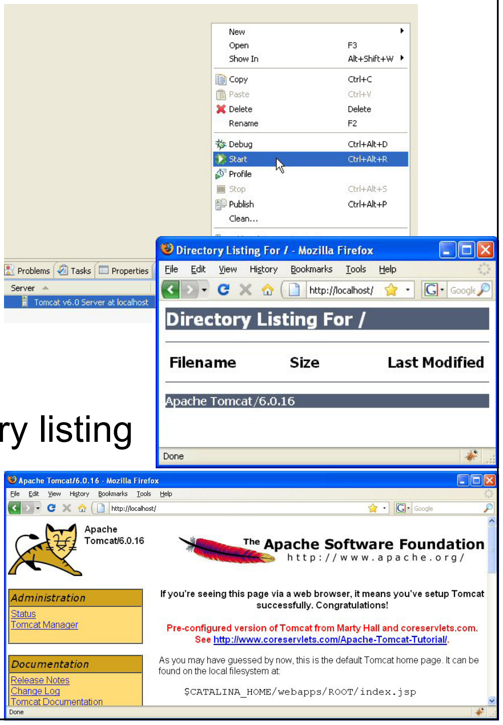
Adding Code to Eclipse Projects

- **Locations**
  - src
    - Unpackaged Java code
    - Packages strongly recommended
  - src/somePackage
    - Java code in somePackage package
  - WebContent
    - Web files (HTML, JavaScript, CSS, JSP, images, etc.)
  - WebContent/some-subdirectory
    - Web content in subdirectory
  - WebContent/WEB-INF
    - web.xml (will be discussed later)
    - Can also click on "Deployment Descriptor"
- **Note**
  - Can cut/paste or drag/drop files into appropriate locations



Starting Server in Eclipse

- **Start Tomcat**
  - Select "Servers" tab at bottom
  - R-click on Tomcat
  - Choose "Start"
- **Verify server startup**
  - Open browser
  - Enter http://localhost/
    - You should see blank directory listing
      - If you want pretty Tomcat welcome page, search for a folder called ROOT in your Eclipse workspace. Copy files from C:\tomcat-dir\webapps\ROOT to that folder



Directory Listing For /

Filename	Size	Last Modified

Apache Tomcat/6.0.16

Apache Tomcat/6.0.16

The Apache Software Foundation
http://www.apache.org/

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

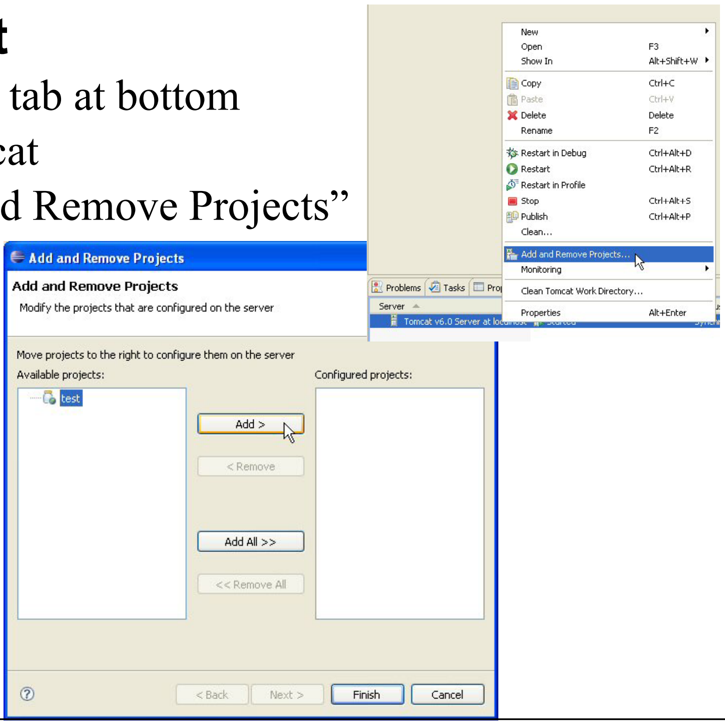
Pre-configured version of Tomcat from Marty Hall and coreservlets.com. See http://www.coreservlets.com/Apache-Tomcat-Tutorial/.

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
```

Deploying App in Eclipse

- **Deploy project**
  - Select “Servers” tab at bottom
  - R-click on Tomcat
  - Choose “Add and Remove Projects”
  - Choose project
  - Press “Add”
  - Click “Finish”
- **Restart Server**
  - R-click Tomcat at bottom
  - Restart



Add and Remove Projects

Modify the projects that are configured on the server

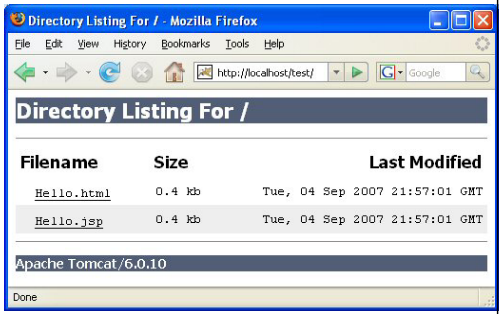
Move projects to the right to configure them on the server

Available projects:

Configured projects:

Testing Deployed Apps in Eclipse

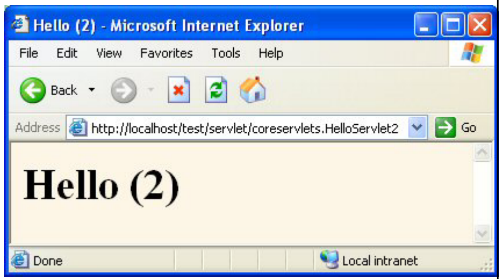
- **Start a browser**
  - Eclipse also has builtin browser, but I prefer to use Firefox or Internet Explorer
- **Test base URL**
  - http://localhost/test/
- **Test Web content**
  - http://localhost/test/Hello.html (case sensitive!)
  - http://localhost/test/Hello.jsp
  - If you used subdirectories
    - http://localhost/test/some-subdirectory/blah.html
- **Test servlets**
  - http://localhost/test/servlet/HelloServlet
  - http://localhost/test/servlet/coreservlets.HelloServlet2
    - Note: custom URLs discussed in next section


Directory Listing For / - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/test/
Google
Done

Directory Listing For /

Filename	Size	Last Modified
Hello.html	0.4 kb	Tue, 04 Sep 2007 21:57:01 GMT
Hello.jsp	0.4 kb	Tue, 04 Sep 2007 21:57:01 GMT

Apache Tomcat/6.0.10


Hello (2) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost/test/servlet/coreservlets.HelloServlet2
Go
Done
Local intranet

Hello (2)

Defining Custom URLs

- **Java code**

```
package myPackage;  ...
public class MyServlet extends HttpServlet { ... }
```

- **web.xml entry (in <web-app...>...</web-app>)**
  – Give name to servlet

```
<servlet>
  <servlet-name>MyName</servlet-name>
  <servlet-class>myPackage.MyServlet</servlet-class>
</servlet>
```

  – Give address (URL mapping) to servlet

```
<servlet-mapping>
  <servlet-name>MyName</servlet-name>
  <url-pattern>/MyAddress</url-pattern>
</servlet-mapping>
```

- **Resultant URL**
  – http://*hostname*/*webappPrefix*/MyAddress

Defining Custom URLs: Example (Assume Eclipse Project is "test")

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    ...
    version="2.5">

  <!-- Use the URL http://hostname/appname/hi instead
       of http://hostname/appname/servlet/HelloServlet -->
  <servlet>
    <servlet-name>Second Hello Servlet</servlet-name>
    <servlet-class>coreservlets.HelloServlet2</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Second Hello Servlet</servlet-name>
    <url-pattern>/hi2</url-pattern>
  </servlet-mapping>
</web-app>
```

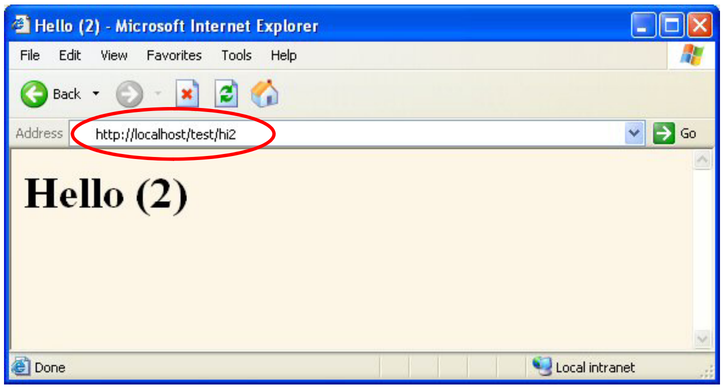
Don't edit this manually. Should refer to version 2.4 or 2.5 (Tomcat 6 only).

Fully qualified classname.

Any arbitrary name. But must be the same both times.

The part of the URL that comes after the app (project) name. Should start with a slash.

Defining Custom URLs: Result


Hello (2) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://localhost/test/hi2
Go
Done
Local intranet

Hello (2)

- **Eclipse details**
  - Name of Eclipse project is “test”
  - Servlet is in src/coreservlets/HelloServlet2.java
  - Deployed by right-clicking on Tomcat, Add and Remove Projects, Add, choosing test project, Finish, right-clicking again, Start

Testing without an IDE (Not Recommended)

Server Setup and Configuration

1. Download and install the Java Development Kit (JDK)
2. Download a server.
3. Configure the server
4. Set up your development environment
5. Test your setup
6. Establish a simplified deployment method
7. Create custom Web applications

- For very detailed coverage of these steps for Tomcat, see
  - http://www.coreservlets.com/Apache-Tomcat-Tutorial/

Download & Install Java JDK

- Recommended Java version
  - Java 6 (aka JDK 1.6) or Java 5 (aka JDK 1.5)
  - Obtain at http://java.sun.com/javase/downloads/
    - Get JDK, not just JRE. Don't get Java EE.
    - Set PATH variable as described in Java documentation
- Minimum supported Java version
  - Apache Tomcat 6.x
    - Java 1.5 or later
  - Servlets 2.3 and JSP 1.2 (standalone servers).
    - Java 1.2 or later.
  - J2EE 1.3 (which includes servlets 2.3 and JSP 1.2).
    - Java 1.3 or later.
  - Servlets 2.4 and JSP 2.0 (standalone servers).
    - Java 1.3 or later.
  - J2EE 1.4 (which includes servlets 2.4 and JSP 2.0).
    - Java 1.4 or later.

Download a Free Server for Your Desktop

- **Apache Tomcat**
  - http://tomcat.apache.org/
  - For installation and setup details, see http://www.coreservlets.com/Apache-Tomcat-Tutorial/
- **Macromedia JRun**
  - http://www.macromedia.com/software/jrun/
- **Caucho Resin**
  - http://caucho.com/products/resin/
- **New Atlanta ServletExec**
  - http://www.newatlanta.com/products/servletexec/
- **Jetty**
  - http://jetty.mortbay.org/jetty/

Configure the Server

- **Identify the JDK installation directory.**
  - For Tomcat: set JAVA_HOME
- **Specify the port.**
  - Change the port from default (usually 8080) to 80
- **Make server-specific customizations.**
  - For Tomcat:
    - Enable servlet reloading
    - Enable the ROOT context
    - Turn on the invoker servlet
    - These changes already done for class.
      To reproduce for home/office setup, see
      http://www.coreservlets.com/Apache-Tomcat-Tutorial/
      - Use preconfigured version. Set CLASSPATH and JAVA_HOME and you are done

Set Up Your Development Environment

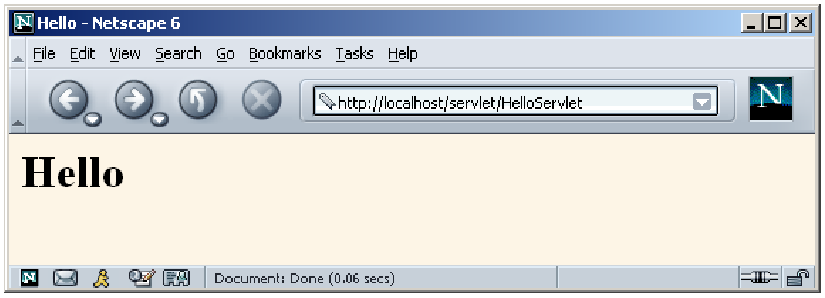
- **Create a development directory**
  - Choose a location in which to develop your servlets, JSP documents, and supporting classes (e.g., C:\Servlets+JSP)
- **Set your CLASSPATH**
  - Tell the compiler about the servlet and JSP JAR file and the location of your development directory.
  - *Setting this variable incorrectly is the single most common cause of problems for beginners.*
- **Make shortcuts to start and stop the server**
  - Make sure it is convenient to start and stop the server
  - Copy tomcat_dir/bin/startup.bat and tomcat_dir/bin/shutdown.bat and choose "Paste Shortcut"
  - Already done if you have preconfigured Tomcat version

Test Your Setup

- **Verify your Java installation**
  - Be sure that you get meaningful results for *both* of these:
    - `java -version`
    - `javac -help`
- **Check your basic server configuration**
  - Start server and access the server home page (http://localhost/)
  - Access a simple user-defined HTML page
    - Download Hello.html from book's source code archive
    - Put in *install_dir*/webapps/ROOT
    - Access with http://localhost/Hello.html
  - Access a simple user-defined JSP page
    - Download Hello.jsp and put in *install_dir*/webapps/ROOT
    - Access with http://localhost/Hello.jsp

Test Your Setup (Continued)

- **Compile and deploy a packageless servlet**
  - Download HelloServlet.java from source code archive
  - Place in development directory (e.g., C:\Servlets+JSP)
  - Compile (if errors, check CLASSPATH)
  - Move HelloServlet.class to *install_dir*/webapps/ROOT/WEB-INF/classes
  - Access with http://localhost/servlet/HelloServlet

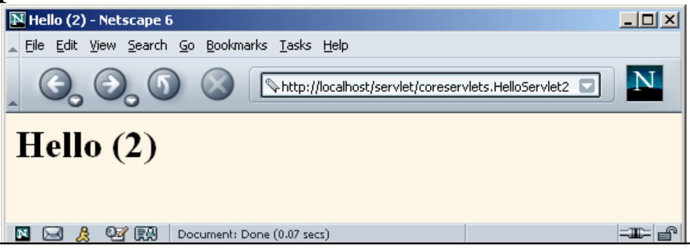

Hello - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://localhost/servlet/HelloServlet

Hello

Document: Done (0.06 secs)

Test Your Setup (Continued)

- **Compile and deploy a packaged servlet**
  - Download HelloServlet2.java from source code archive
  - Place in coreservlets subdirectory of development directory (e.g., C:\Servlets+JSP\coreservlets)
  - Compile (if errors, check CLASSPATH)
  - Move HelloServlet2.class to *install_dir*/webapps/ROOT/WEB-INF/classes/coreservlets
  - Access with http://localhost/servlet/coreservlets.HelloServlet2

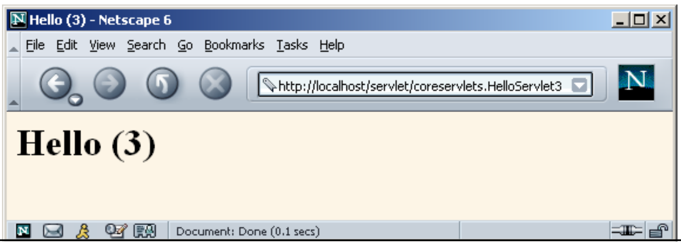

Hello (2) - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://localhost/servlet/coreservlets.HelloServlet2

Hello (2)

Document: Done (0.07 secs)

Test Your Setup (Continued)

- **Compile and deploy a packaged servlet that uses a helper class**
  - Download HelloServlet3.java *and* ServletUtilities.java
  - Place in coreservlets subdirectory of development dir
  - Compile (if errors, check CLASSPATH)
  - Move *both* class files to *install_dir*/webapps/ROOT/WEB-INF/classes/coreservlets
  - Access with http://localhost/servlet/coreservlets.HelloServlet3


Hello (3) - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://localhost/servlet/coreservlets.HelloServlet3
Document: Done (0.1 secs)

Hello (3)

Establish a Simplified Deployment Method

- **Let your IDE take care of deployment**
  - See Eclipse directions on earlier slides
- **Copy to a shortcut or symbolic link**
  - Make shortcut to *install_dir*/webapps/ROOT/WEB-INF/classes
  - For packageless servlets, copy .class file to this shortcut
  - For packaged servlets, copy entire directory to shortcut
    - This is simplest for beginners who don't have an IDE
- **Use the -d option of javac**
  - Lets you have source files in one location but automatically place .class files in another location
- **Use ant or a similar tool**
  - Ant is especially popular when using custom Web apps

Deploying to JHU

Setup

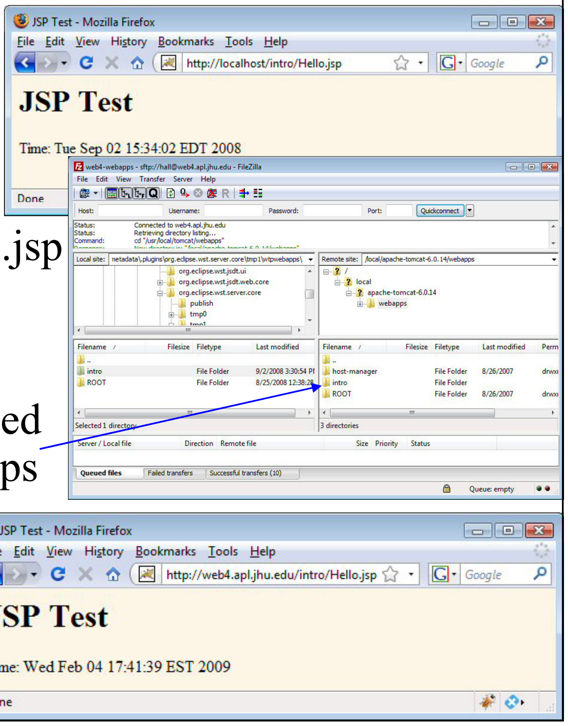
- **Install an sftp client**
  - Google "free sftp client"
    - Eclipse also supports sftp via "Remote System Explorer" perspective. See http://www.eclipse.org/dsdp/tm/
  - I use filezilla, but there are many reliable free clients that support drag-and-drop from your PC to remote server
    - In FileZilla, File → Site Manager lets you save locations
    - If you are prompted, sftp port is 22
- **The Art of WAR: Learn to create WAR files**
  - R-click project, Export, WAR file (or Export, Web, WAR file)
  - You can deploy this WAR file to *any* Java-capable server
- **Or, find the location Eclipse uses for Web apps**
  - Deploy a project, go to eclipse-workspace/.metadata/ and search for a wtpwebapps in that project
    - On my system it is …\.metadata\.plugins\...\tmp1\wtpwebapps
    - Can deploy project folder from here <u>or</u> deploy WAR file

Sending Apps to Tomcat on web4.apl.jhu.edu

- **Make project starting with your name or ID**
  - E.g., name your Eclipse project “hall-intro”
  - Use same naming scheme all semester
- **Deploy from Eclipse and test at home**
  - http://localhost/hall-intro/Hello.jsp
  - http://localhost/hall-intro/servlet/coreservlets.HelloServlet2
- **Send app to web4**
  - Find project
    - Find deployed project folder (e.g., “wtpwebapps/hall-intro”)
    - Or, build WAR file (e.g., “hall-intro.war”)
  - Connect to web4.apl.jhu.edu
  - Copy project folder or WAR file to /usr/local/tomcat/webapps
    - This is the exact pathname. Do *not* replace /usr with your id
  - Test (only hostname changes!)
    - http://web4.apl.jhu.edu/hall-intro/Hello.jsp
    - http://web4.apl.jhu.edu/hall-intro/servlet/coreservlets.HelloServlet2

Example: “intro” project

- **On local PC**
  - R-click Servers, Add & Remove Projects, select intro, Restart
  - http://localhost/intro/Hello.jsp
- **Deploying to web4**
  - Started FileZilla
  - Created intro.war and copied to /usr/local/tomcat/webapps
    - Tomcat expands WAR
- **On web4**
  - http://web4.apl.jhu.edu/intro/Hello.jsp


JSP Test - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/intro/Hello.jsp
Google
Done
web4-webapps - sftp://hall@web4.apl.jhu.edu - FileZilla
File Edit View Transfer Server Help
Host: Username: Password: Port: Quickconnect
Local site: netadata\.plugins\org.eclipse.wst.server.core\tmp1\wtpwebapps\
Remote site: /local/apache-tomcat-6.0.14/webapps
Queued files
Failed transfers
Successful transfers (10)
Queue: empty
JSP Test - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://web4.apl.jhu.edu/intro/Hello.jsp
Google
Done

JSP Test

Time: Tue Sep 02 15:34:02 EDT 2008

```
Status:     Connected to web4.apl.jhu.edu
Status:     Retrieving directory listing...
Command:    cd "/usr/local/tomcat/webapps"
```

Filename	Filesize	Filetype	Last modified
..			
intro		File Folder	9/2/2008 3:30:54 PM
ROOT		File Folder	8/25/2008 12:38:28

Selected 1 directory

Filename	Filesize	Filetype	Last modified	Perm
..				
host-manager		File Folder	8/26/2007	drwx
intro		File Folder		
ROOT		File Folder	8/26/2007	drwx

3 directories

JSP Test

Time: Wed Feb 04 17:41:39 EST 2009

Summary

- **Servlets are efficient, portable, powerful, and widely accepted in industry**
- **Regardless of deployment server, run a free server on your desktop for development**
- **Using Eclipse greatly simplifies development and deployment**
- **Download existing servlet first time**
  – Start with HelloServlet from www.coreservlets.com
  – Click on "Servlet Tutorial" in top-left corner and you can get pre-made Eclipse projects

Questions?