Servlets on the AS/400

IBM Toronto Development Laboratory 2000

Alison Butterill
butteril@ca.ibm.com

IBM®

[tm] Java is a trademark of Sun Microsystems, Inc

Acknowledgement and Disclaimer

Acknowledgement:
Thanks goes to the original creators of this presentation:
Phil Coulthard - IBM Toronto Development Laboratory
George Farr - IBM Toronto Development Laboratory

Disclaimer:
The information contained in this document has not been submitted to any formal IBM test and is distributed on an as is basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customers' ability to evaluate and integrate them into the customers' operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

Reproduction:
The base presentation is the property of IBM Corporation. Permission must be obtained PRIOR to making copies of this material for any reason.

Copyright IBM Corporation, 1999, 2000

Agenda

- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools

Copyright IBM Corporation, 1999, 2000

User Interface Evolution

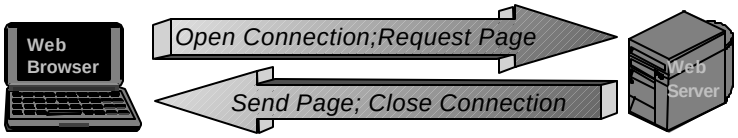
- UI Evolution
  - Early days — Enter-key Driven
    - 5250 Display File "green screens"
- GUI Evolution
  - First wave — Event Driven
    - Client/Server applications, with thick **GUI**
    - Problems? Distribution and maintenance
  - Second wave — Event Driven
    - Web-distributed Java **Applets**
    - Problems? Download time, Browser Java support
  - Third wave — SUBMIT-button Driven
    - Web **HTML** interfaces, generated dynamically
    - Possibly containing one or two small applets
  - Fourth wave ....

Each Option Has Its Place!

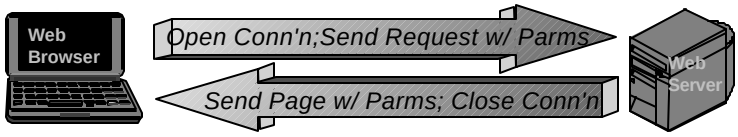
Copyright IBM Corporation, 1999, 2000

User Interface on the 'Net?

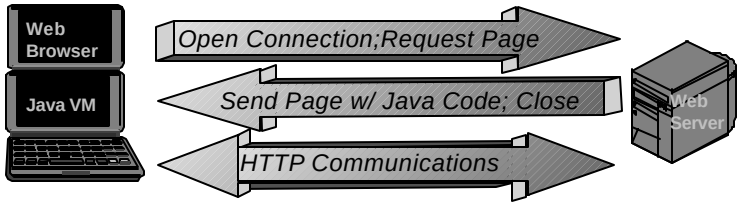
Simple Web:
- Static HTML



Interactive Web:
- CGI-BIN on server



Distributed Web:
- Java applets talk to server



META Group Web Application Models
- *Source: META Group*

Copyright IBM Corporation, 1999, 2000

Applet Vs Application

APPLETS	APPLICATIONS
• Run only in web browser pages – HTML "APPLET" tag – Entry point is "**init**" method	• Run from command line – java MyApp – Entry point is "**main**" method
• Use real estate of web browser	• Use own frame and dialog windows
• Have security restrictions ("sandbox")	• Have no security restrictions

Copyright IBM Corporation, 1999, 2000

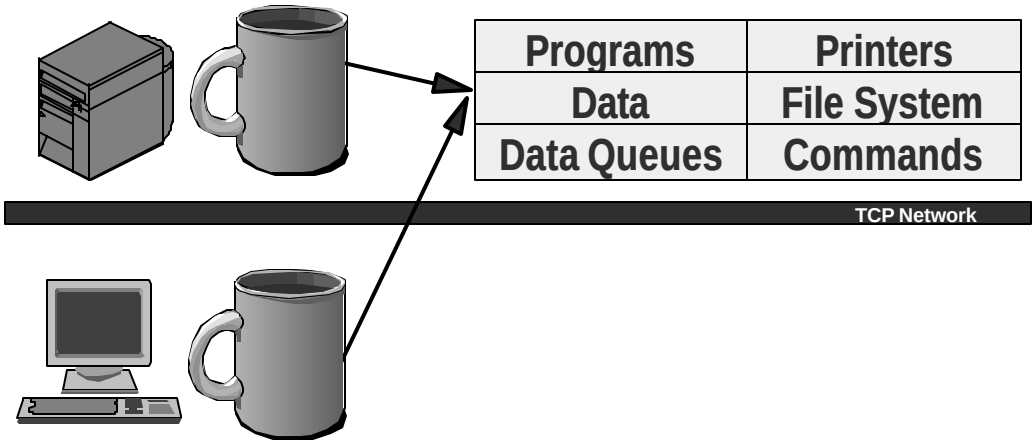
Java and the Web

- HTML, image and Java class files go on server, like your AS/400
- Served "on demand" to Web browsers by HTTP Server software
- For AS/400, use Integrated File System
  - See "OS/400 TCP/IP Configuration and Reference V4R2" manual.
    - http://as400bks.rochester.ibm.com
- Need to work with ISP to put onto Web

Copyright IBM Corporation, 1999, 2000

Where Java Runs

- Java can be run on the AS/400 itself or
- Java can be run on a client like Windows or the web and "talk" to the AS/400



Copyright IBM Corporation, 1999, 2000

Why Servlets?

- Servlets are about . . .
  - User Interface!
- What User Interface?
  - Not 5250 display files
  - Not Client/Server Graphical User Interfaces (GUIs)
  - But... Web Browser User Interfaces
- Servlets generate Web Browser Web-Pages
  - Page is created "on the fly"
  - Can access database, RPG programs, whatever to get content (data) for page

Copyright IBM Corporation, 1999, 2000

What About Applets?

- Applets. . .
  - get too big if entire UI is applet
    - Long time to download
  - require recent versions of Browsers
    - Sometimes need to get Java Activator plug-in
      - Brings Web Browser Java-support up to date

        `www.java.sun.com/products`
- However. . .
  - Applets work fine in Intranets
    - Control over desktop requirements
    - Fast network connection
  - Applets can help in Internets
    - When used to augment, not replace, HTML

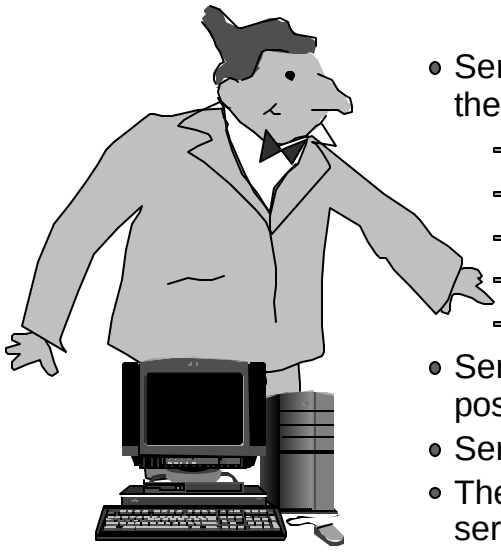
Copyright IBM Corporation, 1999, 2000

What Are Servlets?

- Servlets are . . .
  - Java classes
- Servlets run . . .
  - On the server (eg, AS/400)
- Servlets are called . . .
  - By your HTTP Server software
  - When a user goes to your Web page
- The input to Servlets are . . .
  - User-entered data from a Web page
- The output of a Servlet is . . .
  - Another Web page, displayed to user

Copyright IBM Corporation, 1999, 2000

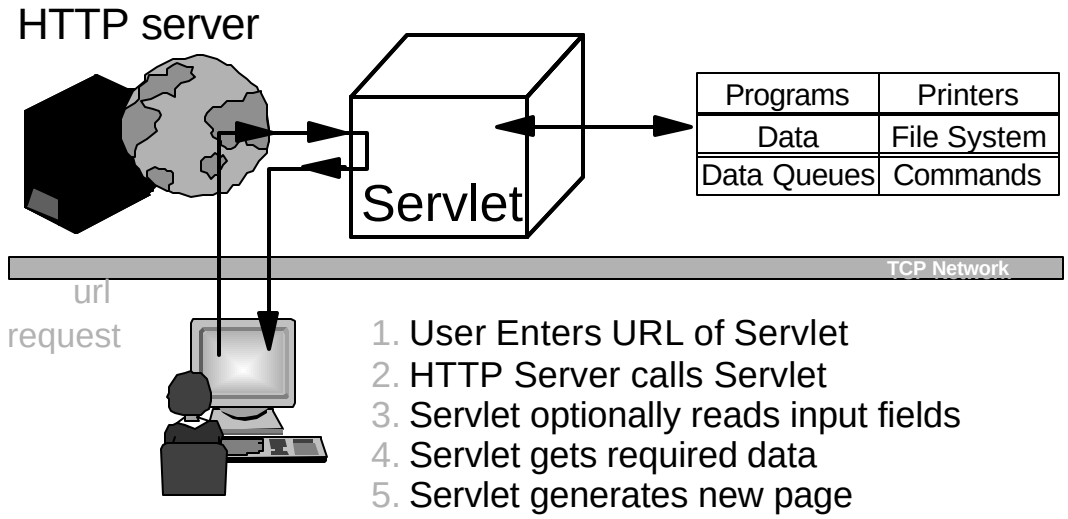
Fat Clients and Thin Clients



- Servlets harness the full power of the core Java APIs:
  - Networking
  - URL access
  - MultiThreading
  - Image Manipulation
  - Data Compression
- Servlets make 'thin' clients possible!
- Servlets are fast and scalable
- They are portable across web servers
- Data in servlets is persistent

Copyright IBM Corporation, 1999, 2000

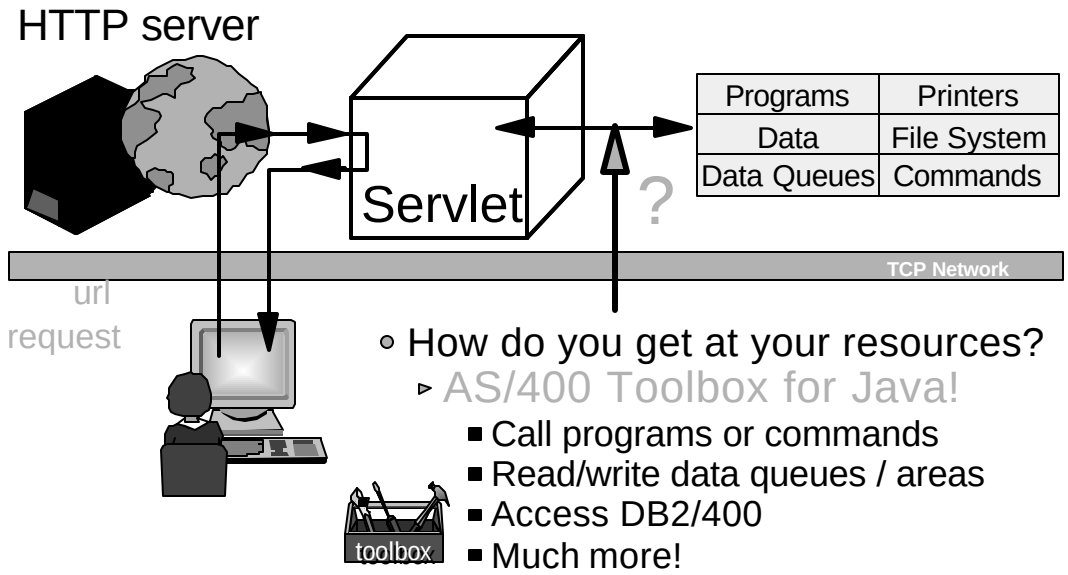
Servlet Architecture



1. User Enters URL of Servlet
2. HTTP Server calls Servlet
3. Servlet optionally reads input fields
4. Servlet gets required data
5. Servlet generates new page
6. HTTP Server sends new page to user

Copyright IBM Corporation, 1999, 2000

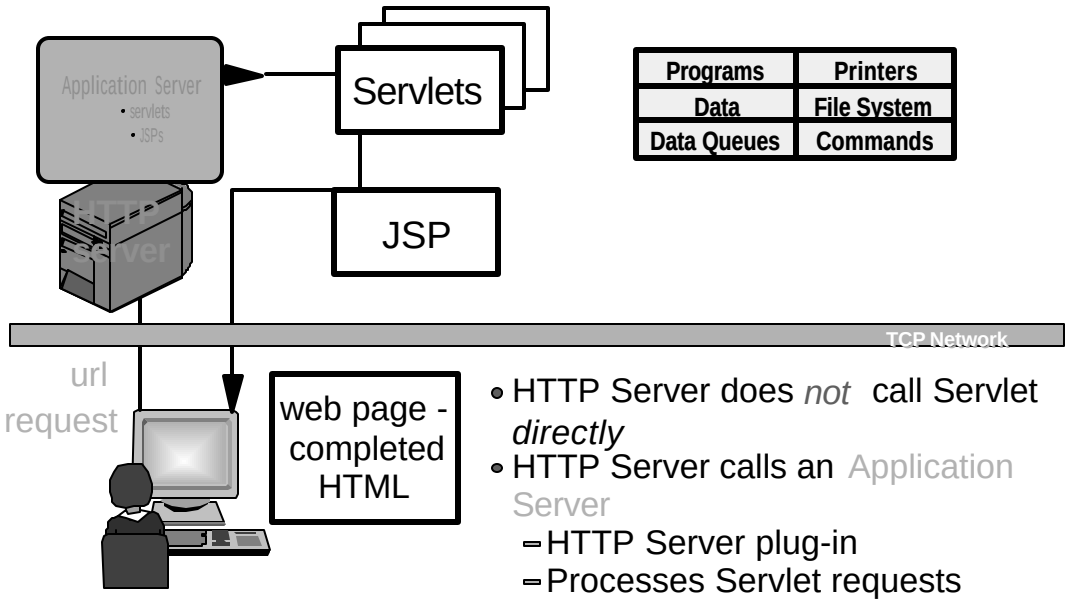
Servlets Calling World



- How do you get at your resources?
  - AS/400 Toolbox for Java!
    - Call programs or commands
    - Read/write data queues / areas
    - Access DB2/400
    - Much more!

Copyright IBM Corporation, 1999, 2000

Application Servers



- HTTP Server does *not* call Servlet *directly*
- HTTP Server calls an Application Server
  - HTTP Server plug-in
  - Processes Servlet requests
  - Written in Java (usually)
  - Uses same JVM for Servlet

Copyright IBM Corporation, 1999, 2000 IBM Toronto

IBM's Application Server

- IBM's App Server is WebSphere Application Server
  - Available on many platforms
    - OS/400, OS/390, NT, AIX, . . .

    www.ibm.com/software/webservers
  - Three flavours

    cumulative function
    - Standard.....: Servlets and JavaServer Pages
    - Advanced....: Enterprise Java Beans
    - Enterprise...: Managed Objects, C++ support
  - Standard is FREE on OS/400
    - As of V4R3, via TCP/IP PTFs
    - A separate but free PID in V4R4
  - Advanced is coming to OS/400
    - Early 2000
    - Not free

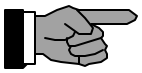
Copyright IBM Corporation, 1999, 2000 IBM Toronto

How To Call A Servlet

- Servlets are invoked by users . . .
  1. By typing Servlet name directly
     - `www.mycompany.com/servlet/myservlet`
     - Use /servlet/ to tell server you want to call a servlet
     - Specify name of servlet last (no extension)
  2. By typing Web page .html file name
     - that has been redirected to a servlet
     - must configure HTTP Server to map files to servlets
- Servlets are invoked by other Web Pages
  1. By using URL links
     - like a regular Web page link via anchor <A> tag
  2. By specifying a Servlet on a form
     - versus a CGI-BIN program

Copyright IBM Corporation, 1999, 2000

Agenda



- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools

Copyright IBM Corporation, 1999, 2000

HTML Forms

- Allows you to solicit user input
- Can build HTML document that contains:
  - checkboxes
  - radio buttons
  - push buttons
  - entry fields
  - selectable lists
  - ... others
- Collect data from user
  - send to server program
    - The FORM tag specifies the name of the program
    - Historically a CGI-bin program
    - New option is a Java Servlet

Copyright IBM Corporation, 1999, 2000

CGI-Bin

- Traditionally, FORMs specify a CGI-bin program
  - Common Gateway Interface
- Any server side program can be a CGI-bin pgm
  - Identified by FORM tag in HTML
  - Called by HTTP Server when user presses "SUBMIT" button on FORM
- CGI-bin programs
  - Use HTTP Server supplied APIs to
    - Read input from FORM fields
    - Write output HTML for new page
  - Pre-date Servlets!
    - No Application Server required for CGI-bin

Copyright IBM Corporation, 1999, 2000

CGI-Bin Versus Servlets

- Similar
  - Identified in HTML FORM
    - Called by HTTP Server when SUBMIT pressed
    - Builds new page dynamically
    - Returned to user's browser by HTTP Server
- But different; servlets are gaining ground because
  - More portable
    - standard definition of Servlets
    - written in ultra-portable Java
  - More efficient
    - called in same JVM as App Server (**vs** separate process)
    - remain in memory after each call (vs new process)
  - More state
    - easier to remember information between calls

Copyright IBM Corporation, 1999, 2000

Servlets Are Good

- Servlets are. . .
  - Powerful. Full power of Java!
    - Classes that are JDK-supplied, IBM-supplied, 3rd party, your own
  - Safe. Full safety of Java!
    - Strongly typed
    - No pointer manipulation
    - Bytecode verification
    - Memory management (garbage collection)
  - Integrated. With the Web Server.
    - Servlets cooperate with Web Server in ways CGI cannot
  - Today's Coolest Trend
    - Huge industry momentum
    - Multi vendor, Multi system, Multi Web Server.

Copyright IBM Corporation, 1999, 2000

Anatomy of a FORM

```
<HTML>
 <HEAD>
  <TITLE>A Simple Input Form</TITLE>
 </HEAD>
 <BODY>

  <FORM action="http://localhost/servlet/MyServlet"
        method="post">
    <INPUT type="submit">
  </FORM>

 </BODY>
</HTML>
```

server-side CGI-bin program or Java servlet to call when SUBMIT button pressed

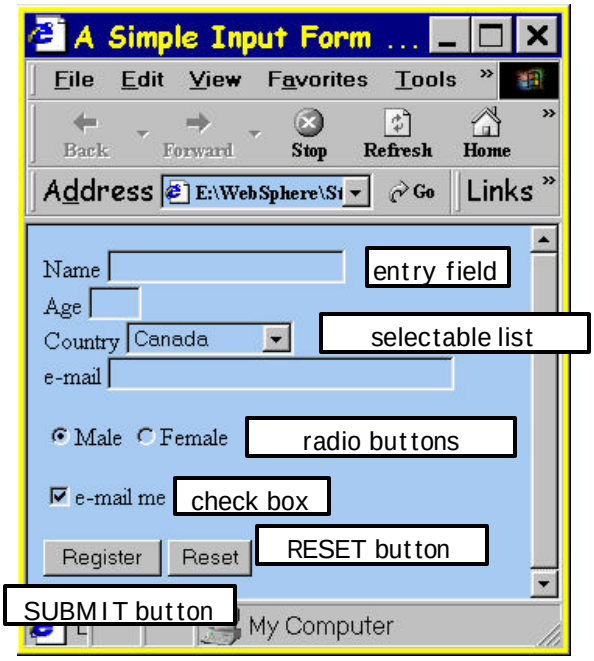
how to call server when SUBMIT pressed

body: one or more input controls plus regular html

"SUBMIT" button

Copyright IBM Corporation, 1999, 2000

Example of a FORM



Copyright IBM Corporation, 1999, 2000

Example of a FORM

```
<FORM action="http://localhost/servlet/MyServlet"
      method="post">
  Name <INPUT type="text" name="name"><BR>
  Age <INPUT type="text" name="age" size="3" maxlength="3"><BR>
  Country <SELECT name="country">
             <OPTION selected>Canada</OPTION>
             <OPTION>Mexico</OPTION>
             <OPTION>United States</OPTION>
           </SELECT> <BR>
  e-mail <INPUT size="30" type="text" name="email"><BR>
  <BR>
  <INPUT type="radio" name="sex" value="M" checked>Male
  <INPUT type="radio" name="sex" value="F">Female<BR><BR>
  <INPUT type="checkbox" name="mail" checked>e-mail me<BR><BR>

  <INPUT type="submit" value="Register">
  <INPUT type="reset"  value="Reset">
  </FORM>
```

entry field

selectable list

entry field

radio buttons

check box

SUBMIT button

RESET button

http://www.w3.org/TR/REC-html40/

Copyright IBM Corporation, 1999, 2000

Forms: Get vs Post Methods

- Clients connect to servers
  - Client example: Web Browser
  - Server: Web Server ("HTTP Server")
- Clients make requests to servers
  - Requests are called "methods"
  - Most frequently used methods are:
    - get : Get a document
    - post: Same as get, but slightly different protocol
    - post is usually preferred over get
- Example:
  - GET /testpage.html HTTP/1.0

Copyright IBM Corporation, 1999, 2000

Agenda

- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools

Copyright IBM Corporation, 1999, 2000

Coding Servlets

- To create a servlet you need. . .
  - To know Java
  - To have Java
    - OS/400: V4R3 or higher (for WebSphere App Server)
    - Windows: `www.java.sun.com/products`
  - To have the Java Servlet Devt Kit
    - Part of
      - IBM WebSphere Application Server
      - IBM VisualAge for Java
    - Also separately downloadable:
      - `www.java.sun.com/products/servlet`

Copyright IBM Corporation, 1999, 2000

JSDK

- Java Servlet Development Kit supplies:
  - Required APIs for servlets
    - Java packages you need to "import"
      - javax.servlet
      - javax.servlet.http
    - Java classes you need
      - HttpServlet, HttpServletRequest, HttpServletResponse
    - JavaDoc API documentation
  - Simple Web Server
    - Named "**Servlet Runner**"
    - For learning purposes only!
      - Not a full function HTTP server

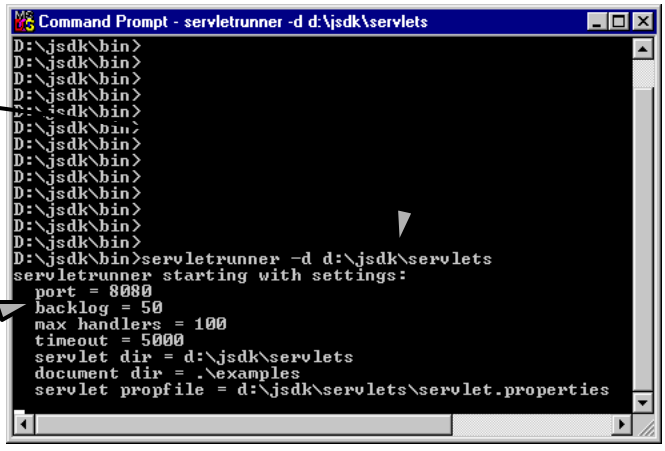
Copyright IBM Corporation, 1999, 2000

Servlet Runner on Windows

- Once you install the JSDK, look under the bin directory, you will see servletrunner.exe
- You must execute it to set up 'application server environment:

Directory where to find the servlet!

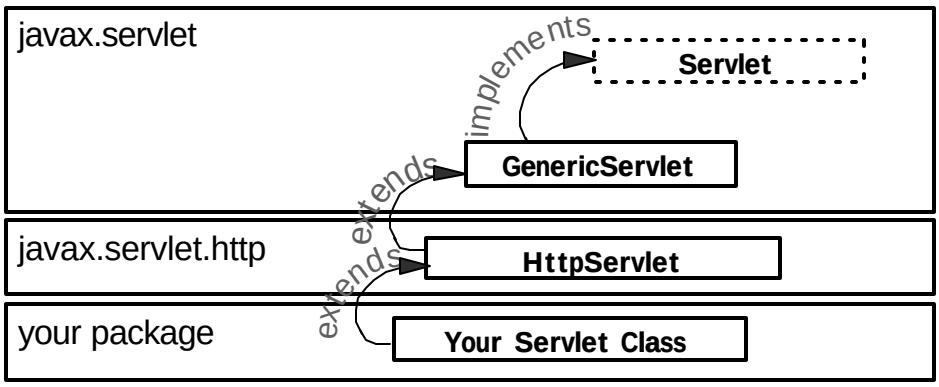
HTTP port is '80'



Copyright IBM Corporation, 1999, 2000

Servlet Classes

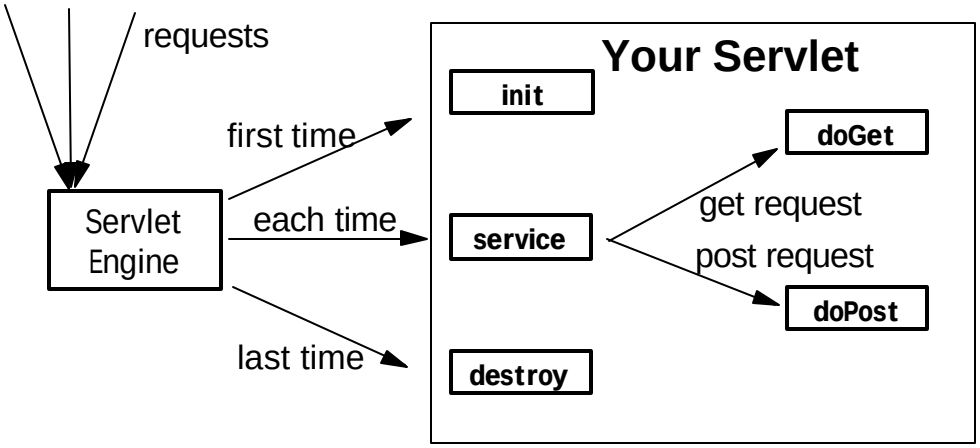
- javax.servlet package
  - classes to support generic, protocol-independent servlets
- javax.servlet.http package
  - adds HTTP-specific functionality



Copyright IBM Corporation, 1999, 2000

Servlet LifeCycle

- Servlet Engine
  - Servlet Runner or WebSphere
    - Calls specific methods in your servlet



Copyright IBM Corporation, 1999, 2000

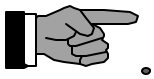
Servlet Methods

- HTTP servlet must define specific methods:
  - service( )
  - doPost( )
  - doGet( )
  - init( )
  - destroy( )
- The service( ) method accepts two parameters:
  - Request object: Tells the servlet about the request
  - Response object: It is used to return a response
- The default service( ) method of an HttpServlet handles the setup and calls either doPost or doGet for you
  - no need to code the service method!

Copyright IBM Corporation, 1999, 2000

Agenda

- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools



Copyright IBM Corporation, 1999, 2000

My First Servlet

- You need to import JSDK packages:

```
import javax.servlet.*;
import javax.servlet.http.*;
```

- You need to extend HttpServlet class:

```
public class myFirstServlet extends HttpServlet
{

}
```

Copyright IBM Corporation, 1999, 2000

My First Servlet

- Finally you need to override appropriate methods

init: For handling initialization
doGet: For handling GET and HEAD requests
doPost: For handling POST requests
doPut: For handling PUT requests
doDelete: For handling DELETE requests
destroy: When removed from memory

Copyright IBM Corporation, 1999, 2000

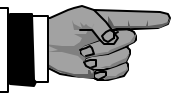
My First Servlet


```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;


public class myFirstServlet extends HttpServlet
{
  public void doGet(HttpServletRequest req, HttpServletResponse res)
                                    throws ServletException, IOException
  {
      // set header field first
      res.setContentType("text/html");

      // then get the writer and write the response data
      PrintWriter out = res.getWriter();
      out.println("<HEAD><TITLE> myFirstServletServlet Output</TITLE></HEAD><BODY>");
      out.println("<h1> myFirstServlet Output </h1>");
      out.println("<P>This output is from my First Servlet!");
      out.println("<p>Congratulations!!! $$$ coming your way");
      out.println("</BODY>");
      out.close();
  }
}
```

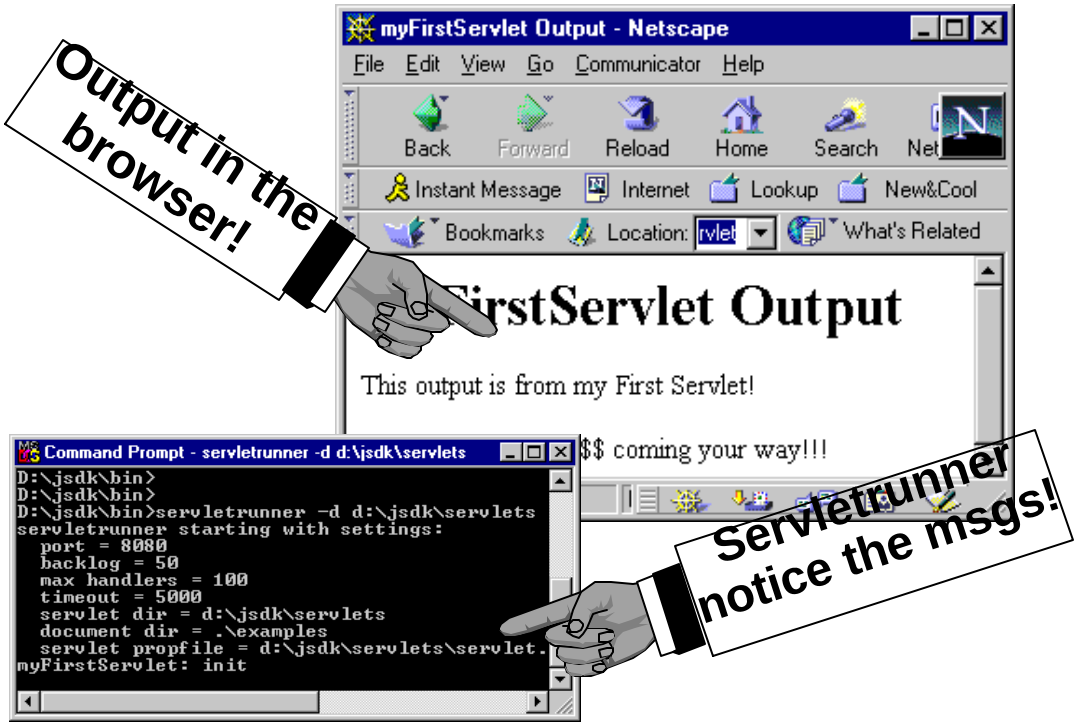
To invoke it do the following from your browser



`http://farr:8080/servlet/myFirstServlet`

Copyright IBM Corporation, 1999, 2000

My First Servlet



Copyright IBM Corporation, 1999, 2000

Example Two

Passing parameters to the servlet

```
<HTML>
<HEAD>
<TITLE> AS/400 Magazine </TITLE>
</HEAD>
<BODY>
<FORM METHOD=GET ACTION="http://localhost/servlet/HelloWorldWithParms">
What is your First name?
<Input TYPE=TEXT NAME="first"><P>
What is your last name?
<Input TYPE=TEXT NAME="last"><P>
and  yes,   what is your age?
<Input TYPE=TEXT NAME="age"><P>
<INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

Three input fields!

Copyright IBM Corporation, 1999, 2000

Example Two

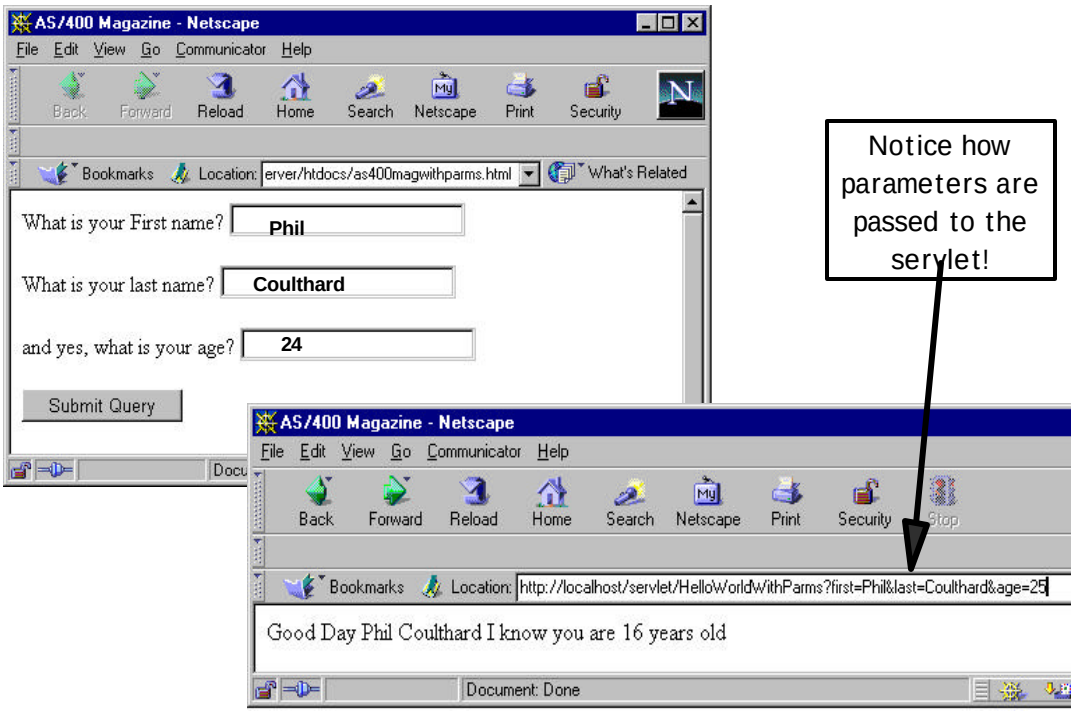
Content Type

Retrieve parameters

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorldWithParms extends HttpServlet
{
    public void doGet (HttpServletRequest req,
                       HttpServletResponse res)
                       throws ServletException, IOException
    {
        PrintWriter   out;
        res.setContentType("text/html");
        out = res.getWriter();
        String first  = req.getParameter("first");
        String second = req.getParameter("last");
        String age    = req.getParameter("age");
        out.println("<html>");
        out.println("<head><title>AS/400
Magazine</title></head>");
        out.println("<body>");
        out.println("Good Day " + first + " " + second);
        out.println("I know you are 16 years old");
        out.println("</body></html>");
    }
}
```

Copyright IBM Corporation, 1999, 2000

Example Result



Copyright IBM Corporation, 1999, 2000

Agenda

- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools

Copyright IBM Corporation, 1999, 2000

JavaServer Pages

- Not a good idea to merge Servlet Java code with HTML tags
  - Mixing skills
  - Can't use HTML WYSIWYG tools
- JavaServer Pages to the rescue!
  - Contain HTML and embedded Java
  - Contain "holes" to be filled in at runtime with data generated by servlets
    - These holes are Java Beans!

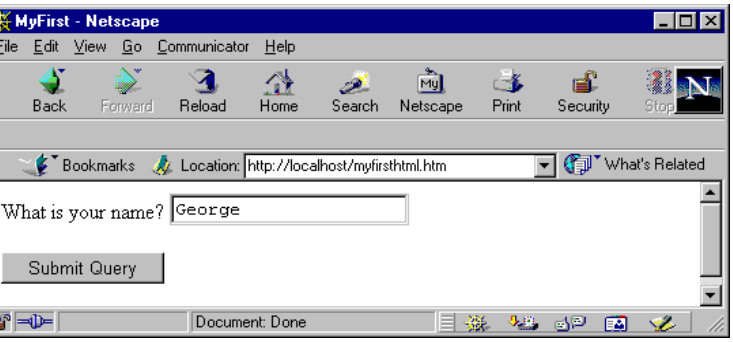
Copyright IBM Corporation, 1999, 2000

JavaServer Pages

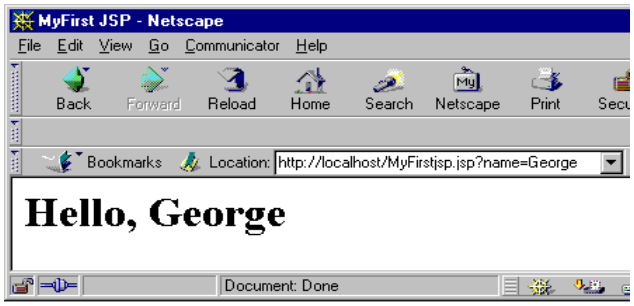
- Java Server Pages are
  - new way to use servlets
  - resemble Active Server Pages (ASPs)
  - HTML files plus embedded Java, JSP tags
  - Similar to server side includes
- App Servers must support them
  - JSDK simple servletrunner does not!
  - IBMs WebSphere application server does

Copyright IBM Corporation, 1999, 2000

JSP Example



Output...



Copyright IBM Corporation, 1999, 2000 IBM Toronto

JSP Example

```
<HTML>
<HEAD>
<TITLE> MyFirst </TITLE>
</HEAD>
<BODY>
<FORM METHOD=GET
ACTION="http://localhost/MyFirstjsp.jsp">
What is your name?
<Input TYPE=TEXT NAME="name"><P>
<INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

Load the JSP page

Copyright IBM Corporation, 1999, 2000 IBM Toronto

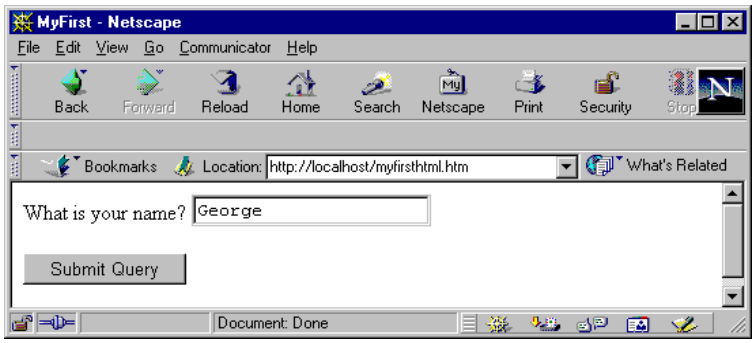
JSP Example

```
<HTML>
<HEAD>
<TITLE> MyFirst JSP </TITLE>
</HEAD>
<BODY>
<h1>
<% if (request.getParameter("name") == null) {%>
Hello World
<%} else {%>
Hello,
<%= request.getParameter("name")%>
<%}%>
</H1>
</BODY>
</HTML>
```

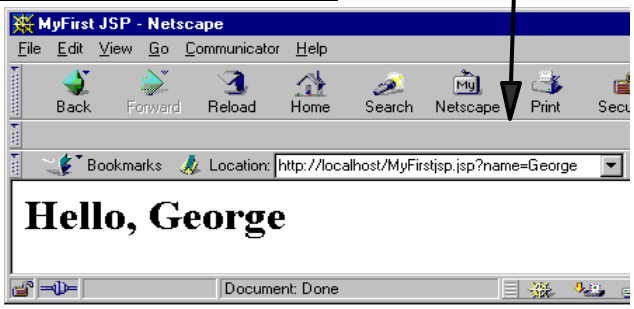
embedded servlet code!

Copyright IBM Corporation, 1999, 2000

JSP Example Result



Notice how parameters are passed to the JSP file!



Output...

Copyright IBM Corporation, 1999, 2000

How JSPs Work

- The server automatically does the following for the servlet:
  - creates
  - compiles
  - loads
  - run
- First time the JSP is accessed it will take more time
- In addition, each time the JSP is changed the web server automatically recompiles it.

Copyright IBM Corporation, 1999, 2000

Generated JSP Java Code

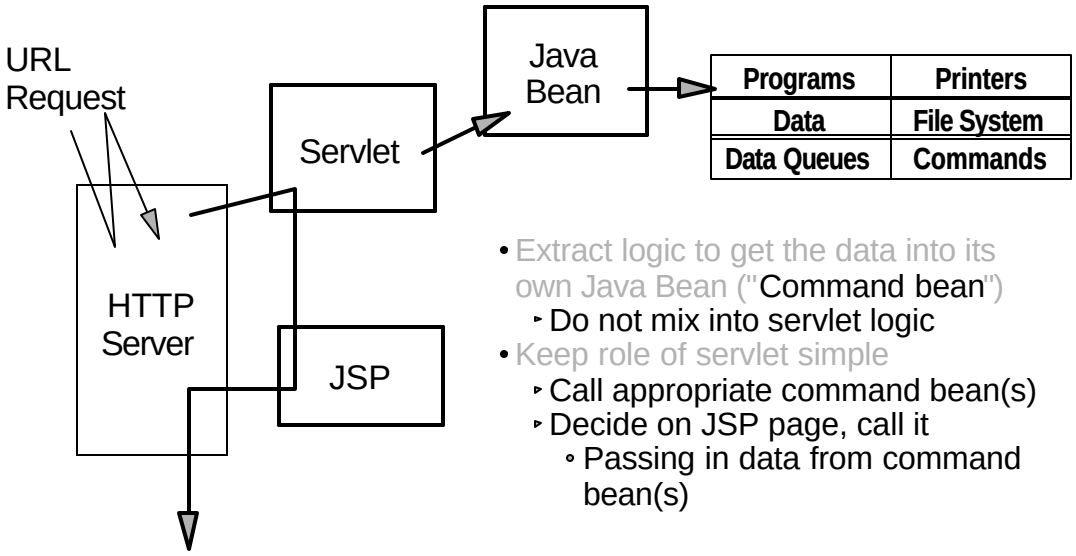
The generated Java code may look as follows:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorldWithParms extends HttpServlet
{
    public void doGet (HttpServletRequest req, HttpServletResponse res)
                      throws ServletException, IOException
    {
        PrintWriter      out;
        res.setContentType("text/html");
        out = res.getWriter();
        out.println("<html>");
        out.println("<head><title>MyFirst JSP</title></head>");
        out.println("<body>");
        out.println("<h1>");
             if (request.getParameter("name")==null)
            out.println("Hello World");
            else
            out.println("Hello," + request.getParameter("name");
            out.println("I know you are 16 years old");
            out.println("</H1></body></html>");
     }
}
```

Copyright IBM Corporation, 1999, 2000

JSP Preferred Usage

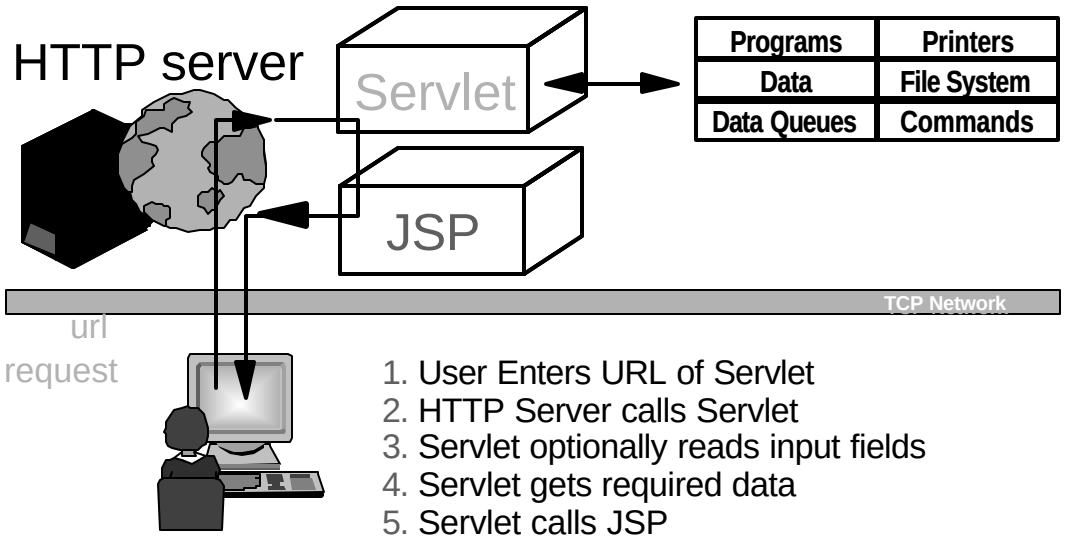
Following is recommended architecture for applications using Servers + JSPs for UI:



- Extract logic to get the data into its own Java Bean ("Command bean")
  - Do not mix into servlet logic
- Keep role of servlet simple
  - Call appropriate command bean(s)
  - Decide on JSP page, call it
    - Passing in data from command bean(s)

Copyright IBM Corporation, 1999, 2000 IBM Toronto

JSP Architecture

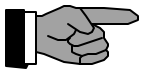


1. User Enters URL of Servlet
2. HTTP Server calls Servlet
3. Servlet optionally reads input fields
4. Servlet gets required data
5. Servlet calls JSP
   - data passed via beans
   - JSP merges data with HTML tags to produce new page
6. HTTP Server sends new page to user

Copyright IBM Corporation, 1999, 2000 IBM Toronto

Agenda

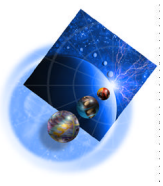
- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools



Copyright IBM Corporation, 1999, 2000

IBM Tools

- Choices (IBM) for Tools
  - VisualAge for Java for all your Java coding
    - classes, beans, servlets, command beans, applets, ...
  - WebSphere Studio for
    - WYSIWYG JSP development ("Top Page")
    - Web site management, deployment
    - Wizards to generate sample servlet, JSPs from given Java command bean
      - And for generating command bean for database access



Copyright IBM Corporation, 1999, 2000

WebSphere Studio 3.0

- A suite of tools that helps teams
  - Design, Develop, and Publish web sites
- Supports multidisciplinary team working together with a common view
  - Page designers, graphic artists, programmers, and others can all work on the same projects
  - Site is composed of independent parts, created and maintained by the appropriate people
  - Key tools are included, but any tool can plug in, so team members can work with their favorites
- Targeted to WebSphere App Server
  - HTML and other web technologies
  - Java Servlets and Java Server Pages (JSP)
  - Enterprise Java Beans

Copyright IBM Corporation, 1999, 2000

WebSphere Studio

Web Site Development

Web Master
- Uses **Site Analysis** to measure usage rate
- Uses **Performance Pack** for clustered environments

Develop

Deploy

Page Designer
- Uses **Studio** to create HTML pages
- Uses **TopPage, Fusion**, **ScriptBuilder** or other page editing tools

Artist or Writer
- Uses **Studio** to contribute content
- Uses **BeanBuilder** or other tools like ImageReady

Programmer
- Uses **Studio** to create servlets and JSP
- Uses **VisualAge** to create JavaBeans and for debugging

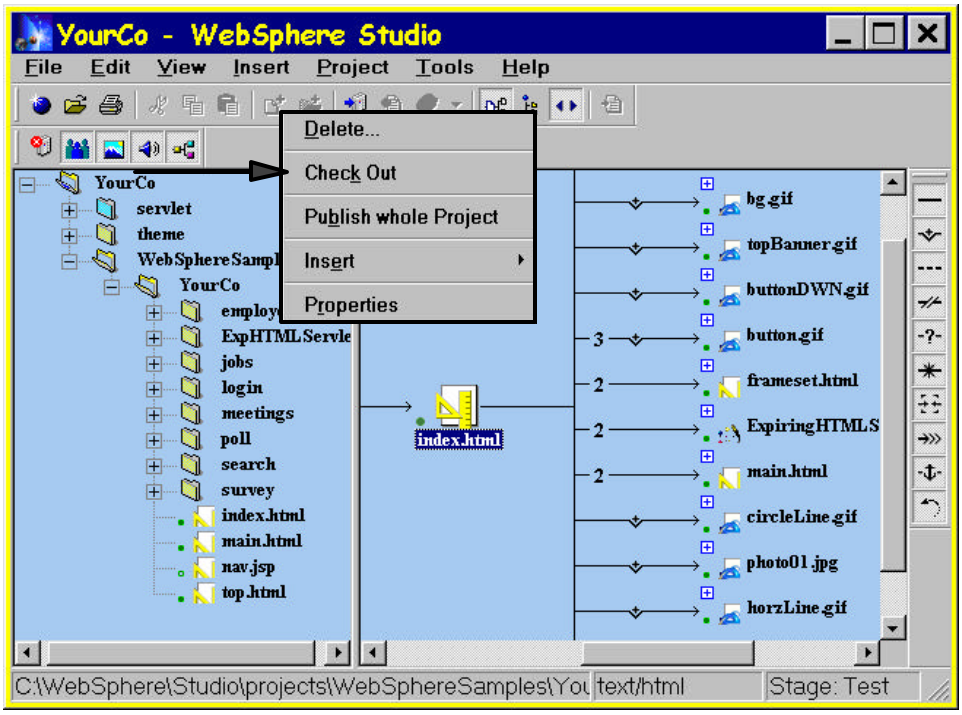
Creative Director
- Uses **Studio** to deploy the site
- Uses **Site Analysis** to find structural problems

Analyst
- Uses **Site Analysis** to understand usage patterns to make the site more effective

Manage

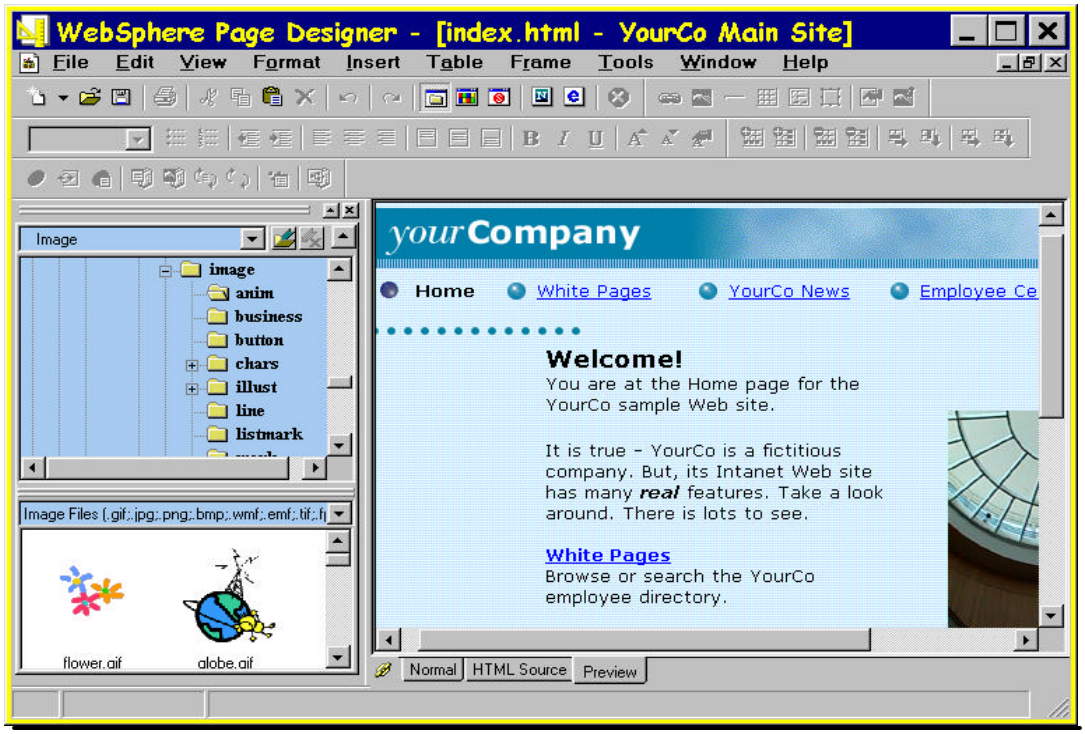
Copyright IBM Corporation, 1999, 2000

WebSphere Studio 3.0



Copyright IBM Corporation, 1999, 2000 IBM Toronto

Top Page



Copyright IBM Corporation, 1999, 2000 IBM Toronto

WebSphere Development Tools

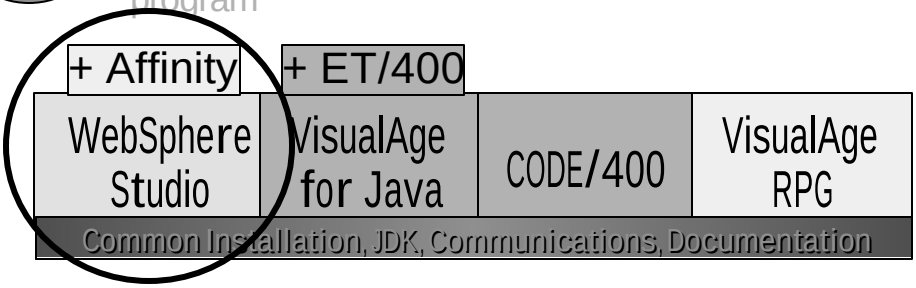
- Client Tools Together
  - CODE/400
  - VisualAge RPG
  - VisualAge for Java Professional + ET/400
  - WebSphere Studio + AS/400 "Affinity"

Oct 20, 2000

Optionally purchase VAJava EE for EJB Tooling

- Add Additional Functionality
  - Business Partner "Early Adopter" DSPF to Web conversion (WebFacing) program

4Q, 2000



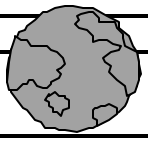
Copyright IBM Corporation, 1999, 2000

Summary

- Why?
  - What are servlets?
  - How do you call them?
  - What's a FORM?
  - What about CGI?
- What?
  - How to code one!
  - Example
- JavaServer Pages
- IBM Tools

Copyright IBM Corporation, 1999, 2000

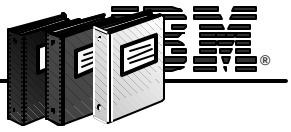
WebSites



Website URL	Description
www.software.ibm.com/ad/vajava	IBM VisualAge for Java
www.software.ibm.com/ad/varpg	IBM VisualAge for RPG and CODE/400
www.software.ibm.com/webservers	IBM WebSphere AS and Studio
www.ibm.com/java www.as400.ibm.com/java	IBM Java
www.java.sun.com/products	Sun Java
www.as400bks.rochester.ibm.com	IBM AS/400 online books and help
www.redbooks.ibm.com	IBM Redbooks
www.software.ibm.com/ad/services	IBM Toronto services. Want to run this course onsite? We can...

Copyright IBM Corporation, 1999, 2000

Books



Book, URL	By, ISBN
Java for RPG Programmers www.emergingskills.com	Phil Coulthard, George Farr. ISBN 1-889671-23-1
JAVA and the AS/400 www.29thStreetPress.com	Daniel Darnell. ISBN 1-58304-033-1
Experience RPG IV www.advice.com/ibm	Maha Masri, Julie Santilli, Heather Rogers. ISBN 1-889671-22-3
VisualAge for RPG by Example www.amazon.com	Brian Meyers and Jef Sutherland. ISBN 1882419839
Core Java series www.amazon.com	Now a series of 3 books. Horstmann, Cornell
Teach Yourself Java in 21 Days www.amazon.com	Rogers Cadenhead, Laura Lamay. ISBN 1575213907
Programming VisualAge for Java www.amazon.com	John Akerly, Antonello Parlavecchia

Copyright IBM Corporation, 1999, 2000