

Web Tools in WebSphere Development Studio Client 5.0

*George Farr, Claus Weiss,
Phil Coulthard
IBM Toronto Laboratory*

WDS for iSeries

New World

New Servers

New Tools

- ▶ This presentation looks at the application models today in the 5250 environment and in the Web environment. It then looks at the user interface definitions for the Web. Instead of DDS, HTML files or Java server pages to define the user interface are used. To call programs on the iSeries from a Java program, an XML language called PCML is used to describe the interface between the Java program and the iSeries program. The presentation looks at the WebTools provided in WDS and walks through a scenario that describes the steps to create a simple interaction, that has an input screen asking for a customer number and that fetches the data from the iSeries data base by calling an RPG program.

Disclaimer

Acknowledgement:

- This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:
 - ▶ *Phil Coulthard, George Farr, Claus Weiss, Don Yantzi*

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.



AGENDA



iSeries AD, IBM Toronto

- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**

Eclipse

IBM @server. For the next generation of e-business.

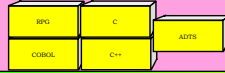


WDS 5.0 Standard!



iSeries AD, IBM Toronto

WebSphere Development Studio

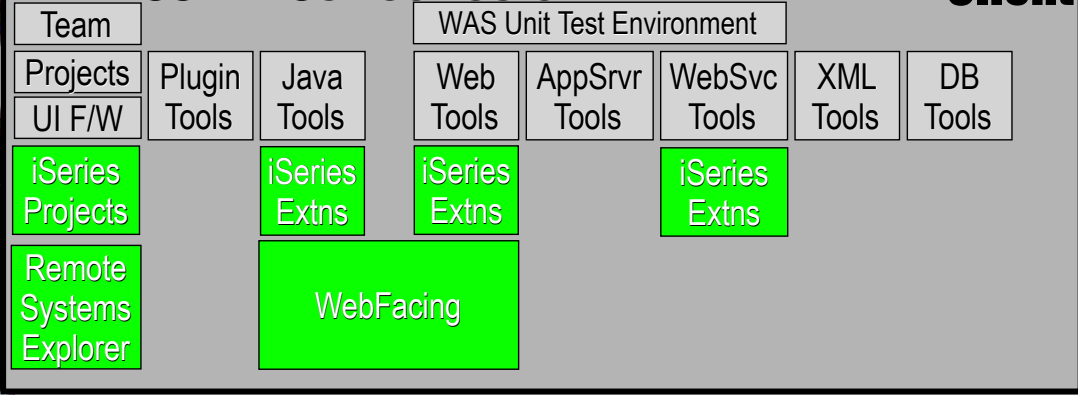


Host

WDS

WSSD + iSeries Tools

Client



CODE

VARPG

~~VJava~~

~~Studio~~

Eclipse

IBM eServer. For the next generation of e-business.

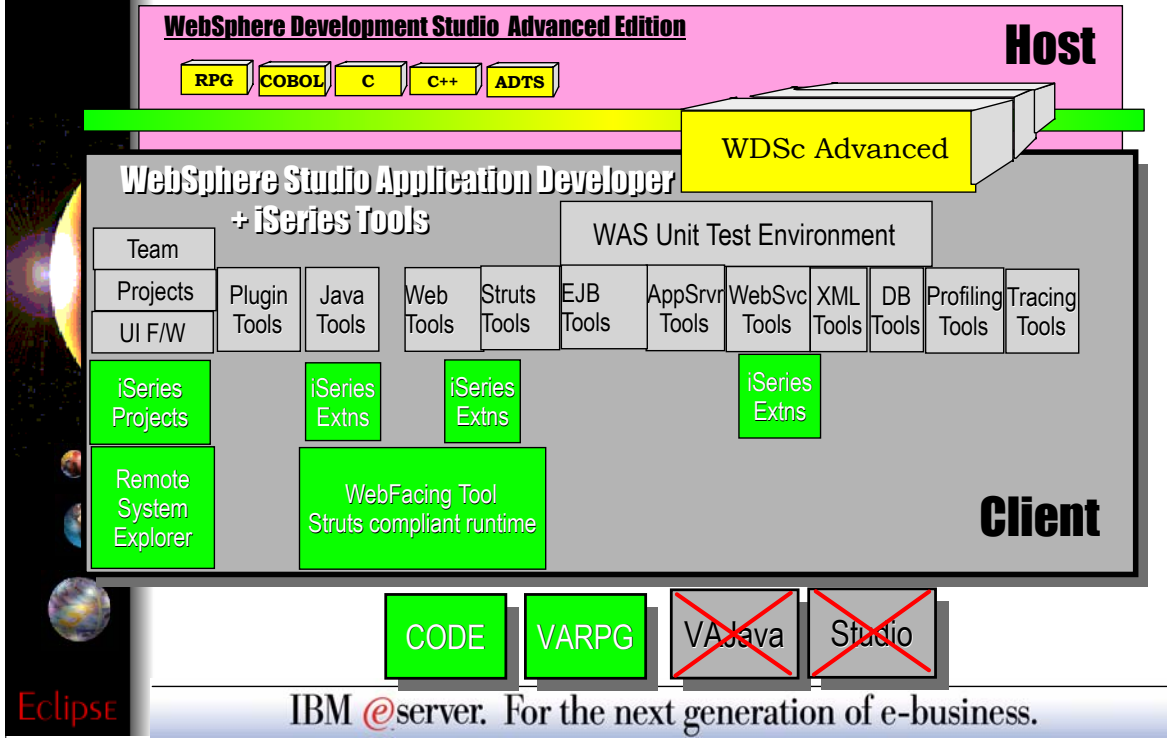
- ▶ The next generation of workstation development tools to create iSeries e-business applications that leverage the iSeries and the power of WebSphere Application Server
- ▶ Inherits and extends the WebSphere Studio Site Developer's robust, easy-to-use development environment for creating, building, and maintaining dynamic Web sites, or developing Java, Web and Web services applications. It also includes a robust, easy-to-use development environment for creating, building and maintaining iSeries RPG, COBOL, C, C++ applications, and Web-enabled applications using the IBM WebFacing Tool. The Remote Systems Explorer also makes it easy to access UNIX, Windows, and Linux application resources.



WDSc 5.0 Advanced!



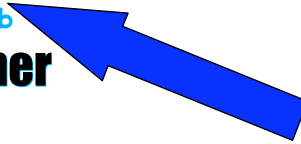
iSeries AD, IBM Toronto



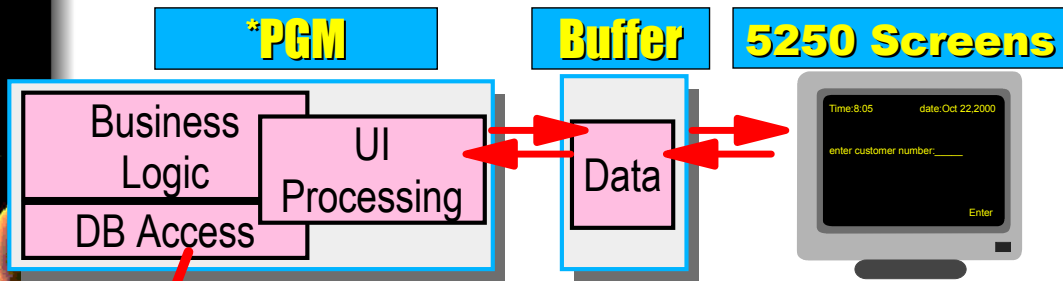
- ▶ The next generation of advanced workstation development tools to create iSeries e-business applications that leverage the execution capabilities of WebSphere Application Server
- ▶ Inherits and extends the application development environment for the creation and maintenance of J2EE and Web services applications from WebSphere Studio Application Developer V5.0. It also includes a robust, easy-to-use development for creating, building, and maintaining iSeries RPG, COBOL, C, C++ applications, and Web-enabled applications using the IBM WebFacing Tool. The Remote Systems Explorer also makes it easy to access UNIX, Windows, and Linux application resources.



- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**



Today's Model



1. Program puts up screen, waits for input
2. Program processes input, does business logic

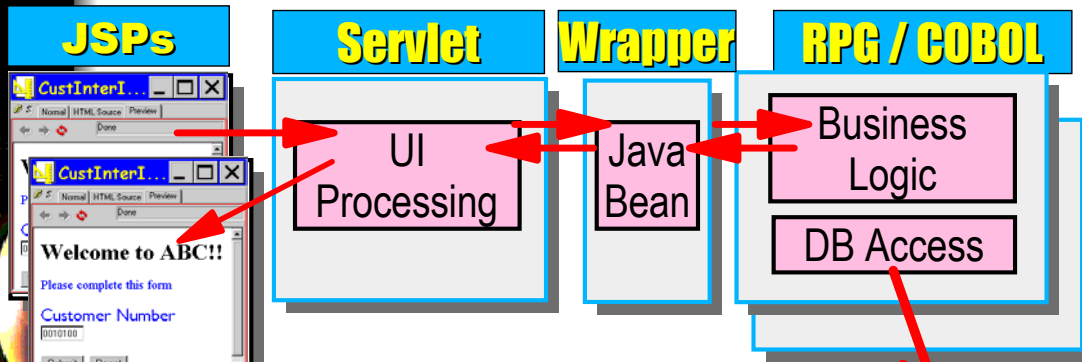
- ▶ in today's application model, the program drives the user interface, the program waits for input, after the input is received it presents a new panel and waits for input again.



eBusiness Application



iSeries AD, IBM Toronto



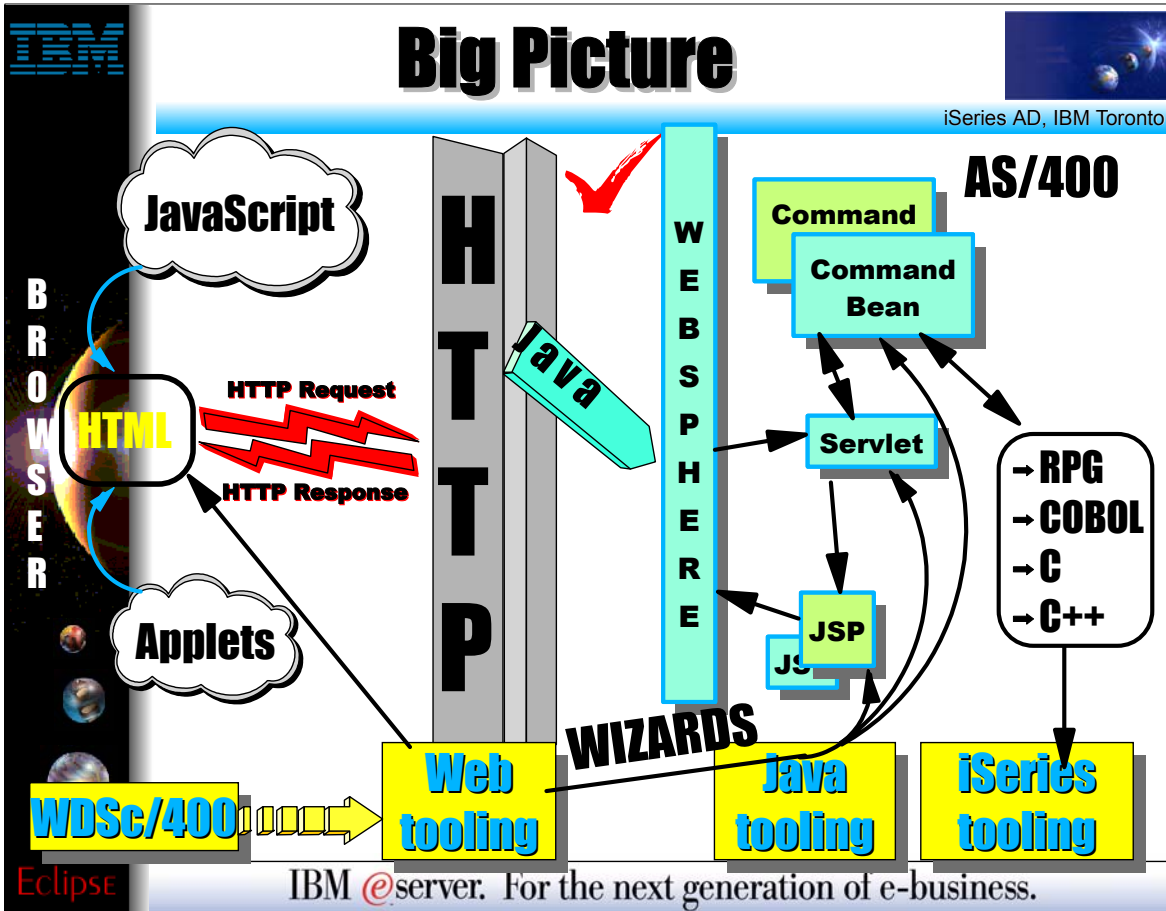
1. **JavaServer Pages (JSPs) for UI**
 - ▶ **HTML tags for constant part**
 - ▶ **JSP tags for dynamically substituted data**
 - ▶ **HTML FORMs for user input fields**
2. **Servlets for UI processing**
3. **Java Bean encapsulation of business logic **DB2/400****
4. **Business Logic: *PGM or ILE procedures**



Eclipse

IBM @server. For the next generation of e-business.

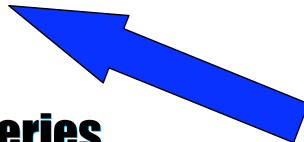
- ▶ In the Web the user interface is presented by a browser not a 5250 device.
- ▶ The user requests services from the server.
- ▶ The server program responds to the various requests.
- ▶ There is 3 tiered application divided in:
 - ▶ User Interface (jsp's)
 - ▶ Controller logic running in an application server (servlets)
 - ▶ Business logic running on a business server RPG/COBOL/C**/JAVA running on iSeries



- ▶ On the bottom of this slide the tools to develop the different pieces of a 3 tiered application are listed. All these tools are included in the WDS_c package.
- ▶ WebTooling addresses the Web user interface and the deployment of applications
- ▶ Java tooling is to write the middle tier Java servlets
- ▶ iSeries RPG/COBOL/CL/DDS tooling is to help create the business logic



- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**





What Are JSPs?



iSeries AD, IBM Toronto

▶ **JavaServer Pages (JSPs) are ...**

- **.jsp files**
 - ▶ containing html tags plus JSP tags

▶ **JSP tags ...**

- **Allow dynamic data to be inserted into the static HTML**
 - ▶ Data is extracted from Java Beans passed to the JSP

▶ **JSPs are called ...**

- **By your servlet**
- **The input to JSPs are ...**
 - ▶ Java Beans passed from your Servlet
- **The output of a JSP is ...**
 - ▶ A full Web page, displayed to user

Eclipse

IBM @server. For the next generation of e-business.

- ▶ Java Server Pages are the way to define the dynamic Web user interface.
- ▶ Historically, the first step was to use CGI bin programs to serve dynamic web pages which have a limit in scalability and are not very straight forward to use to define the user interface.
- ▶ The first attempt to move away from CGI bin programs was to move to servlets and mix the user interface definition and controller logic in the servlet.
- ▶ It turned out not to be a very good way to work because of the mix of UI definition and logic.
- ▶ In today's environment, the user interface definition is moved out of the logic and stored in separate JSP files, these files contain html tags to describe the user interface and they also contain small pieces of Java code to add dynamic content to the user interface.
- ▶ The jsp's get converted to servlets at runtime and the Java code gets executed on the server before the resulting html tags get sent to the browser.

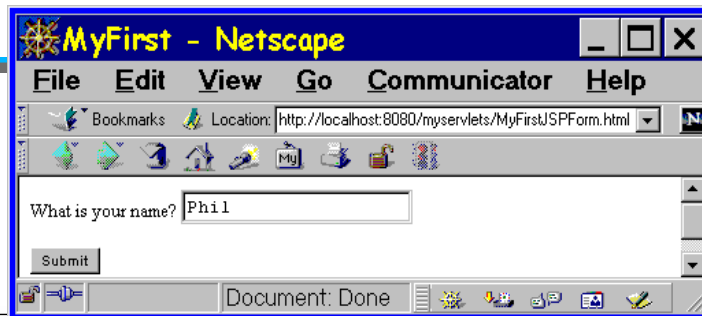
\myservlets\MyFirstJSPForm.html

```

<HTML>
<HEAD><TITLE> MyFirst </TITLE></HEAD>
<BODY>
<FORM METHOD=POST
  ACTION="http://localhost/myservlets/MyFirstjsp.jsp">
What is your name?
<INPUT TYPE="text" NAME="name"><P>
<INPUT TYPE="submit" VALUE="Submit">
</FORM>
</BODY>
</HTML>

```

Call the JSP
when SUBMIT
pressed



- ▶ Shows an example of an html file that contains a form that invokes a jsp on the server.
- ▶ The form tag in html is used to pass data from the Web page to the server. It is also used to indicate that a program should be invoked on the server instead of serving just a simple request for a new Web page.
- ▶ The result of this simple html file being rendered in the browser is shown in slide.
- ▶ Note an entry field "type text", which has the Name name and the push button "type submit" to submit the request to invoke the MyFirstjsp.jsp on the server and pass the data in the entry field to the server.



JSP Example ...



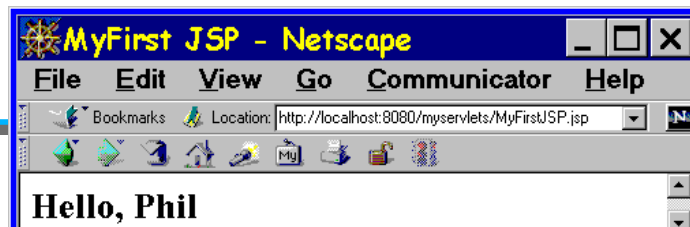
IBM Toronto

\myservlets\WEB-INF\jsp\MyFirstJSP.jsp

```
<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 Java code! Java code is run on the server, not the client. The resulting all-html file is sent to the client

You don't have to compile!



Eclipse

IBM @server. For the next generation of e-business.

- ▶ Shows the jsp that will get invoked from the form on the previous slide.
- ▶ There are normal html tags and there is some Java code included inside the <% %> Java tags
- ▶ If the entry field is empty when the request gets submitted, "Hello World" gets displayed in the resulting Web page
- ▶ If there is data in the entry field "Hello and the value in the entry field" are displayed



What Are Servlets?

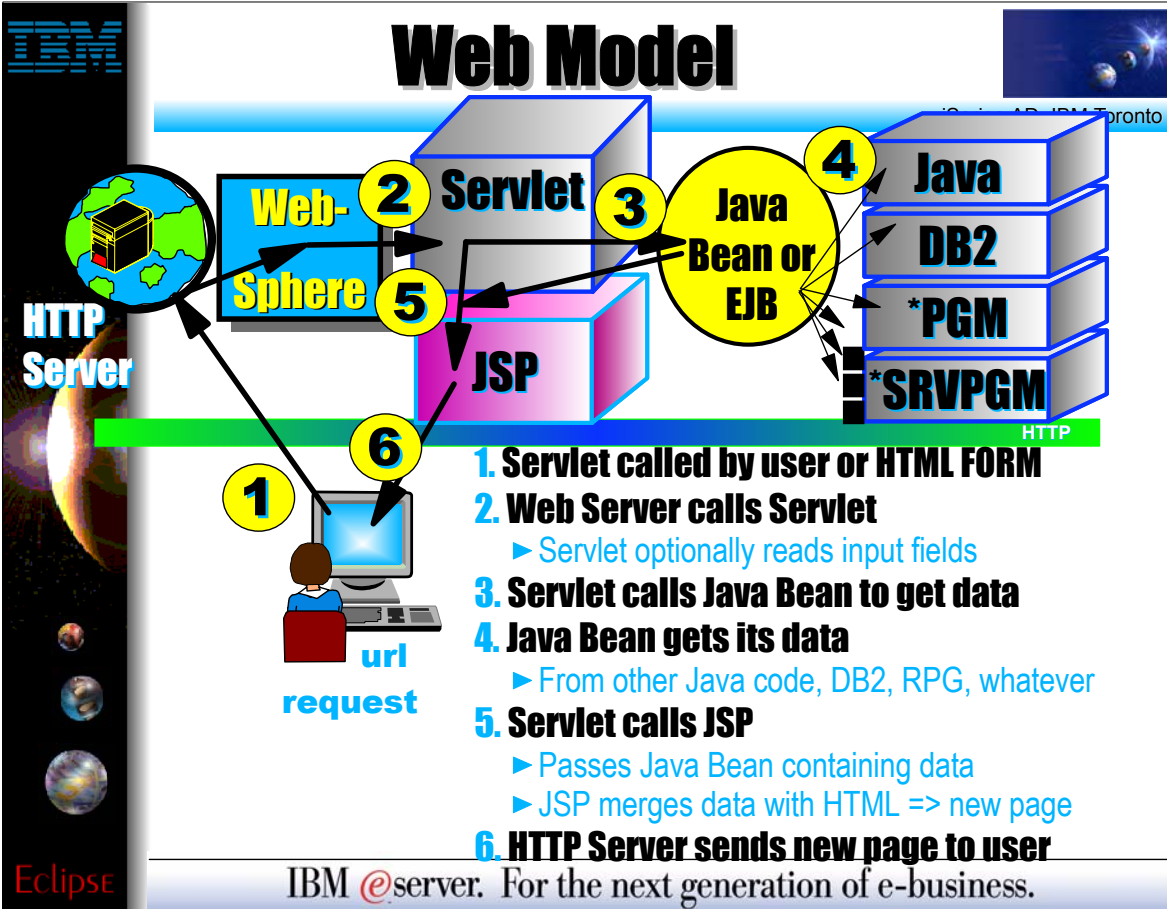


iSeries AD, IBM Toronto

- ▶ **Servlets are ...**
 - Java classes (programs written in Java)
- ▶ **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 ...**
 - Java Bean, passed to a JavaServer Page

Eclipse

IBM @server. For the next generation of e-business.





- ▶ **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



```
<HTML>
<HEAD>
  <TITLE>A Simple Input Form</TITLE>
</HEAD>
<BODY>
  <FORM action="http://localhost/servlet/MyServlet"
        method="post">
    <INPUT type="submit"> "SUBMIT" button
  </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



```
<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/>

Example of a FORM



A Simple Input Form ...

File Edit View Favorites Tools

Back Forward Stop Refresh Home

Address Go Links

Name entry field

Age

Country selectable list

e-mail

Male Female radio buttons

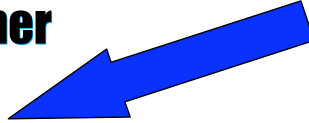
e-mail me check box

RESET button

My Computer



- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**



- ▶ Now you understand how jsp's and servlets work. The next slides define the interface (signature) of an iSeries program to make it easy to call it from Java.



Say we have the following RPG code ...

```

FCUSTOML3  IF    E              K DISK
DCUSTINFO              DS
D Number              1        7A
D Name                8        47A
C      *ENTRY        PLIST
C      PARM          CUSTINFO
C      Number        SETLL    CUSTOM01
C      Number        READE    CUSTOM01          9091
C                      EVAL    Name = CUSTNA
C                      MOVE    *ON              *INLR
***** End of data *****
    
```

Pass in Customer ID and receive back customer name.

- ▶ Here, the RPG program to call expects one parameter, that is defined as a data structure. The data structure contains 2 fields: Number and Name



Create XML required tags



iSeries AD, IBM Toronto

```
<pcml version="1.0">

  <!-- Create a Data Structure -->
  <struct name="custinfo">
    <data name="Number" type="char" length="7"
      usage="inputoutput" init="0014400"> </data>
    <data name="Name" type="char" length="40"
      usage="inputoutput" init=" "> </data>
  </struct>

  <!-- Program getcust -->
  <program name="getcust"
    path="/QSYS.lib/FARR.lib/GETCUST.pgm">
    <data name="gotback" type="struct"
      usage="inputoutput" struct="custinfo"> </data>
  </program>
</pcml>
```

Eclipse

IBM @server. For the next generation of e-business.

- ▶ This slide shows the definition of this interface in PCML.
- ▶ First on top is how the structure is defined
- ▶ It has a name "custinfo" in this case and 2 fields are defined inside the structure
- ▶ These are the fields Number and Name they have the same data definition as in the RPG program on the previous slide.
- ▶ Also the number field gets initialized with a value
- ▶ At the bottom of the slide the name of the program and where it is located on the iSeries is defined. Also defined is to pass the structure as a parameter



Call RPG from Java Servlet



```
public static void main(String[] argv)
{
    AS400 as400System = new AS400();
    ProgramCallDocument pcml = null;
    String msgId, msgText;
    Object value = null;

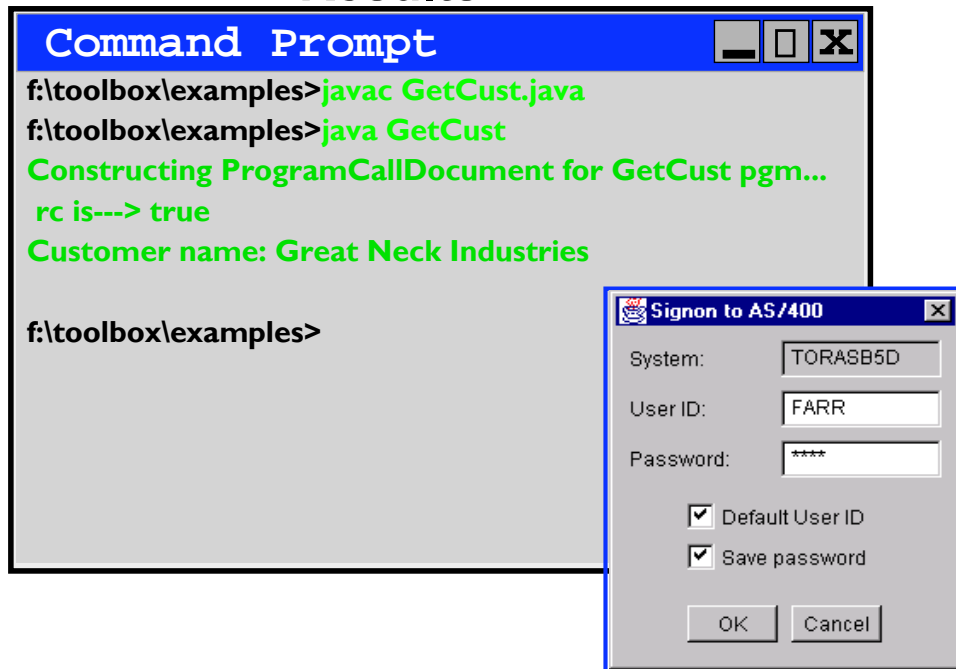
    try {
        System.out.println(
            "Creating ProgramCallDocument for GetCust pgm.");
        pcml = new ProgramCallDocument(as400System, "GETCUST");
        boolean ok = pcml.callProgram("getcust");
        System.out.println(" rc is----> " + rc);
        if (!ok)
            { /* Retrieve list of AS/400 messages & display them */ }
        else
            {
                value = pcml.getValue("getcust.gotback.Name");
                System.out.println("Customer name: " + value);
            }
    } catch (PcmlException exc) {
        System.out.println("*** Call to getcust failed. ***");
        System.exit(0);
    }
    System.exit(0);
} // end main method
```

File: GetCust.java
Class: GetCust

Eclip

- ▶ Here is the sample code in Java that now allows the use of the PCML interface description of our RPG program.
- ▶ In Java the interface description in the PCML file is used and along with a call the program described in the PCML file.
- ▶ This code will call the RPG program, it will pass a customer number to the RPG program which will fetch the corresponding customer name from the database

Results . . .



```
Command Prompt
f:\toolbox\examples>javac GetCust.java
f:\toolbox\examples>java GetCust
Constructing ProgramCallDocument for GetCust pgm...
rc is---> true
Customer name: Great Neck Industries

f:\toolbox\examples>
```

Signon to AS/400

System: TORASB5D

User ID: FARR

Password: ****

Default User ID

Save password

OK Cancel

- ▶ Here is an example of running the example and getting a customer name for the default customer number back. Note that to access the iSeries program requires the user to logon to the iSeries with a normal authentication dialog.



- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **WebTooling primer**
- **What is Web Tools for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**



- ▶ Next is Web applications, since we now understand how to define the interface of an iSeries program in XML and how to use the XML definition in Java



Terms: Web Application



iSeries AD, IBM Toronto

▶ Web App folder structure:

- +Web application folder (root folder)
 - +source
 - all non-deployed files (java)
 - +webApplication
 - all Web files (html, jsp, gif, ...) collectively known as "Web Resources"
 - +META-INF
 - MANIFEST.MF maps dependent jar files in other Web apps
 - +theme
 - .css style sheets
 - +WEB-INF
 - web.xml ← Web application deployment descriptor:
▶ identifies servlets, security, env vars, mime types, key pages, external references and session configuration info
 - +classes
 - Java classes of this app (usually generated)
 - +lib
 - Supporting classes and jar files

J2EE
Servlet
Spec
2.2



Eclipse

IBM @server. FOR THE NEXT GENERATION OF e-BUSINESS.

- ▶ Here is the Web Application folder structure.
- ▶ The Web application will run in an application server.
- ▶ The application server expects the application to conform to certain standards in the way its files are stored.
- ▶ The slide shows the standard for the J2EE Spec 2.2
- ▶ WAS 5.0 implements this standard and the Web tools in WebSphere Studio will generate and work with Web applications conforming to this standard.

▶ Example Web Application

```
+accounts
  +source
  +webApplication
    -index.html      www.mydomain.com/accounts
  +receivable
    -page1.html      www.mydomain.com/accounts/receivable/page1.html
  +payable
    -page1.html      www.mydomain.com/accounts/payable/page1.html
  +META-INF
    -MANIFEST.MF
  +theme
    -corporate.css
  +WEB-INF
    -web.xml
  +classes
  +lib
```

- ▶ Shows an example of the Web application folder structure and where in this folder structure the files of a Web application have to be stored

▶ Web Archive Files (WAR)

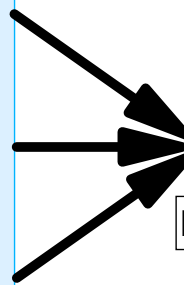
J2EE
Servlet
Spec
2.2

- **One file containing**
 - ▶ Whole folder structure of Web application
 - ▶ Including web.xml file
 - ▶ Optionally including source
- **Used to**
 - ▶ Install and configure Web application in an application server

```

+Web application folder (root folder)
+source
  - all non-deployed files (java)
+webApplication
  - all Web files (html, jsp, gif, ...)
+META-INF
  -MANIFEST.MF
+theme
  - .css style sheets
+WEB-INF
  -web.xml
+classes
  - Java classes of this app (usually
    generated)
+lib
  - Supporting classes and jar files

```



MyWebProject.war

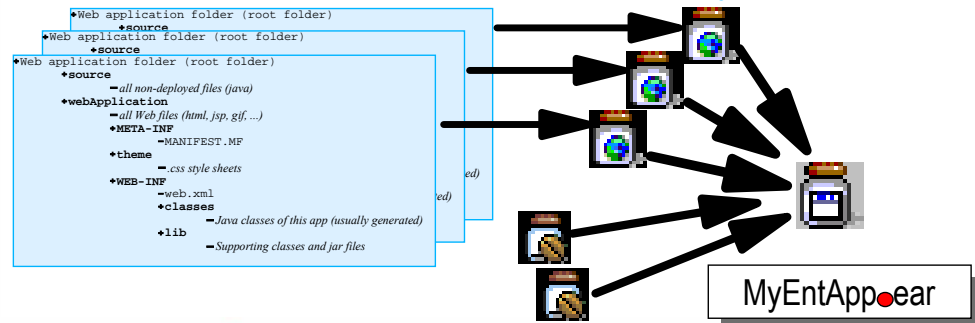
- ▶ A Web Archive (WAR) file, is a file that can store a full web application in a single file similar to a ZIP file, storing the web application makes it easy to move it around and deploy it to different servers.
- ▶ The application server has the capability to take the WAR file and create a web application inside the application server from it.



J2EE
EJB
Spec
1.1

▶ Enterprise Archive Files (EAR)

- **One file containing:**
 - ▶ Zero or more Web Archive (war) files
 - ▶ Zero or more EJB jar files
 - ▶ A J2EE deployment descriptor
- **Used to install and configure:**
 - ▶ All pieces of a J2EE Enterprise Application
 - ✓ Web application plus EJBs plus EJB clients
 - ▶ All Web applications for a Web site (say)



- ▶ An EAR file takes the concept of a WAR file one step further and stores multiple WAR files (Web applications) in one EAR file
- ▶ EAR files can also directly be installed in WebSphere Application Server 5.0



▶ **WDS Sc Web Tools At A Glance:**

- **Web projects**
- **Web Editors**
 - ▶ For JSP and HTML files
 - ▶ For Cascading Style Sheets
 - ▶ For logos, images, and animation
- **Link viewing and management**
- **Import/Export**
 - ▶ Numerous formats
- **Wizards**
 - ▶ for servlets, Web pages from DB or JavaBean
- **WebSphere 4.0 Built-in Test Environment**
- **WebSphere Local/Remote Server support**
- **iSeries special support**

- ▶ Now that we know what it takes to build a Web application let's look at what the WDS Sc tooling can do to make the creation of Web applications easier
- ▶ Not to worry if this all sounds too complicated, the WDS Sc tools help by creating most of the pieces of a Web application.



▶ **Web Projects**

- **Special type of project**
 - ▶ With its own "new" wizard
 - ▶ With its own perspective
 - ▶ With its own tools
- **Created with J2EE folder layout**
- **Created with simple web.xml file**
 - ▶ Automatically updated as resources are created
 - ▶ Has specialized web.xml editor

▶ **File->New->Other...**

- **->Web->Web Project**

- ▶ In WebSphere Studio, the Web developer has a full suite of tools to support him
- ▶ When creating a Web project, it is created with the correct folder structure to comply to the J2EE standard
- ▶ Also there are specific iSeries tools provided to help with creating the interface description of programs or service programs on the iSeries.
- ▶ The tools also allow to reference iSeries data base fields and to use these in the web

Web Projects

iSeries AD, IBM Toronto

Create a Web Project

Define the Web Project
Create a Web project and add it to a new or existing Enterprise Application project.

Project name: MyWebProject

Use default location

Location: E:\WDSC\WSSD\workspace\MyWebProject

Enterprise Application project name: DefaultEAR

Context root: MyWebProject

Create CSS file

Automatically associates with a supplied EAR file for easy auto-cfg of Application Server

Optionally creates a CSS style for whole Web app

Create a Web Project

Module Dependencies
Select dependent JARs for the module within the Enterprise Application project.

Project name: MyWebProject

Enterprise Application project name: DefaultEAR

Available dependent JARs:

JAR/Module	Project

Manifest Class-Path:

Identify runtime dependent jar/war files

Eclipse

IBM @server. For the next generation of e-business.



Create a Web Project

Define Java Build Settings

Specify the source and output folders, as well as the Java classpath in the appropriate order.

Source Projects Libraries Order

Use the project as source folder

Use source folders contained in the project

/MyWebProject/source

Create New Folder... Remove

Build output folder:
/MyWebProject/webApplication/WEB-INF/classes

< Back Next > **Finish** Cancel

Identify build-time dependent jar files

Navigator

- MyWebProject
 - source
 - webApplication
 - theme
 - Master.css
 - WEB-INF
 - classes
 - lib
 - ibm-web-bnd.xmi
 - ibm-web-ext.xmi
 - web.xml
 - .classpath
 - .websettings

- source
 - ▶ for non-deployed files
- webApplication
 - ▶ for deployed files
 - ▶ .jsp and .html files go here
- .../theme
 - ▶ for style sheets
- ../WEB-INF
 - ▶ run-time dependencies
 - ▶ jar files go here



WDS*c*: Web Tools



iSeries AD, IBM Toronto

▶ WDS*c* Web Tools At A Glance:

- ▶ Web projects
 - ✓ Created with J2EE-defined folder structure for Web Applications
 - ✓ Superset of Java projects (so contain all Java Tool support too)
- ▶ Automatic creation/maintenance of web.xml file
- ▶ Editor support
 - ✓ JSP and HTML files
 - ✓ Support for creating, validating, editing and debugging
 - ✓ Including WYSIWYG PageDesigner
 - ✓ Custom JSP tags (taglib) support
 - ✓ based on the Sun Microsystems JSP 1.1 Specification
 - ✓ Images and animation
 - ✓ Cascading Syle Sheets (CSS)
- ▶ Import/Export from/to a variety of sources
 - ✓ HTTP/FTP/WAR
- ▶ Link viewing and management
 - ✓ Converting links, flagging broken links, and fixing up links as linked resources are moved or renamed
- ▶ Wizard for servlets, Web pages from DB or JavaBean
- ▶ Integration with WebSphere Unit Test Environment

Eclipse

IBM @server. For the next generation of e-business.

Web tooling Components



- ▶ This slide shows the different components that support the creation of Web Applications on an iSeries server.
- ▶ The workbench integrates all the different tools
- ▶ Page designer allows WYSIWYG user interface composition for both html or jsp definitions
- ▶ Wizards allow you to get started fast on your Web project
- ▶ Specific iSeries tools help you to exploit the content of your iSeries server

► Web projects

- File->New->Other->Web->Web Project

Define the Web Project
Create a Web project and add it to the current workspace.

Project name: MyWebProject

Use default location

Location: E:\WDS*c*\WSS

Module Dependencies
Select dependent JARs for the module within the Enterprise Application project.

Project name: MyWebProject

Enterprise Application project name: DefaultEAR

Define Java Build Settings
Specify the source and output folders, as well as the build order, for the project.

Source: [Source] [Projects] [Libraries]

Build output folder: [MyWebProject\webApplication\WEB-INF\classes]

Buttons: < Back, Next >, Finish, Cancel

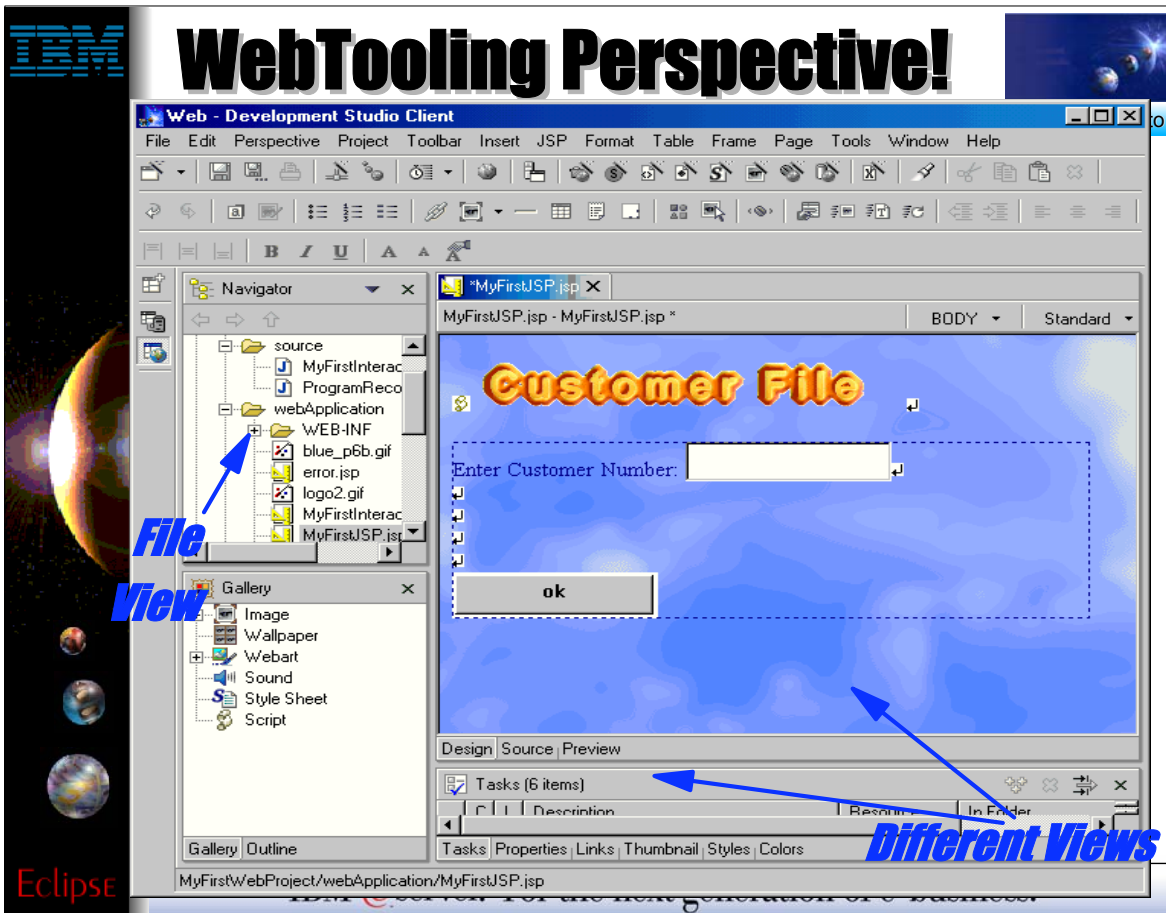
Annotations:

- Automatically associates with a supplied EAR file for easy auto-cfg of Application Server
- Optionally creates a CSS style for whole Web app
- Identify runtime dependent jar/war files
- Identify build-time dependent jar files

Project Structure (Navigator):

- **source**
 - for non-deployed files
- **webApplication**
 - for deployed files
 - .jsp and .html files go here
- **.../theme**
 - for style sheets
- **../WEB-INF**
 - run-time dependencies
 - jar files go here

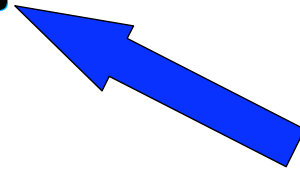
- Here is a sample of the Web project wizard
- It will guide the user through specifying the information necessary to create a Web project, based on this information a Web project infrastructure will be created.
- This base Web project can then be filled with content
- It can be tested directly in the Studio test environment or can be deployed in EAR files to remote servers.



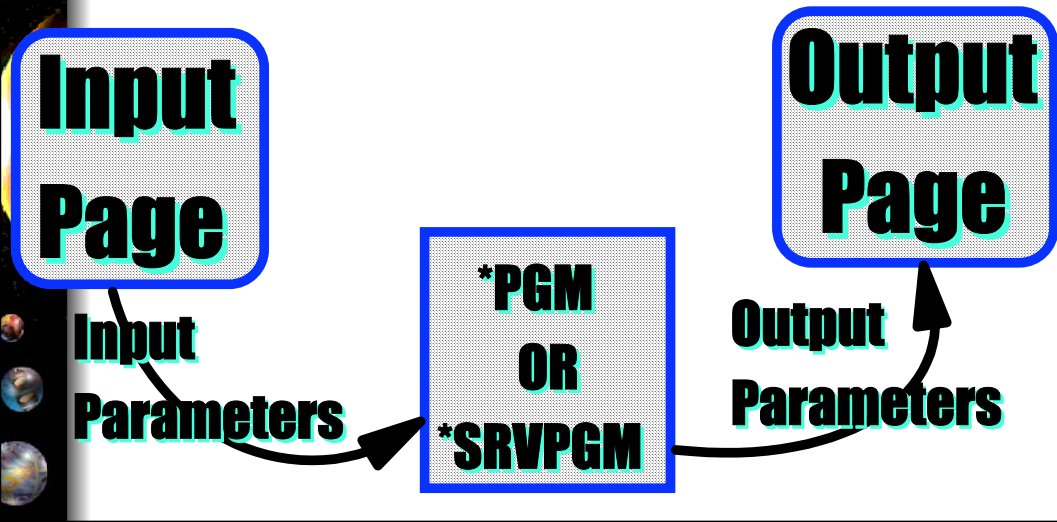
- ▶ Here is a sample of the Web Perspective which contains the tools a Web developer needs. In this example, the Page Designer tool , file view, and a view of multi media files shows.
- ▶ Perspective's can be tailored to individual needs.



- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**



An interaction



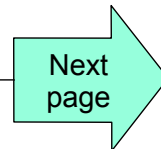
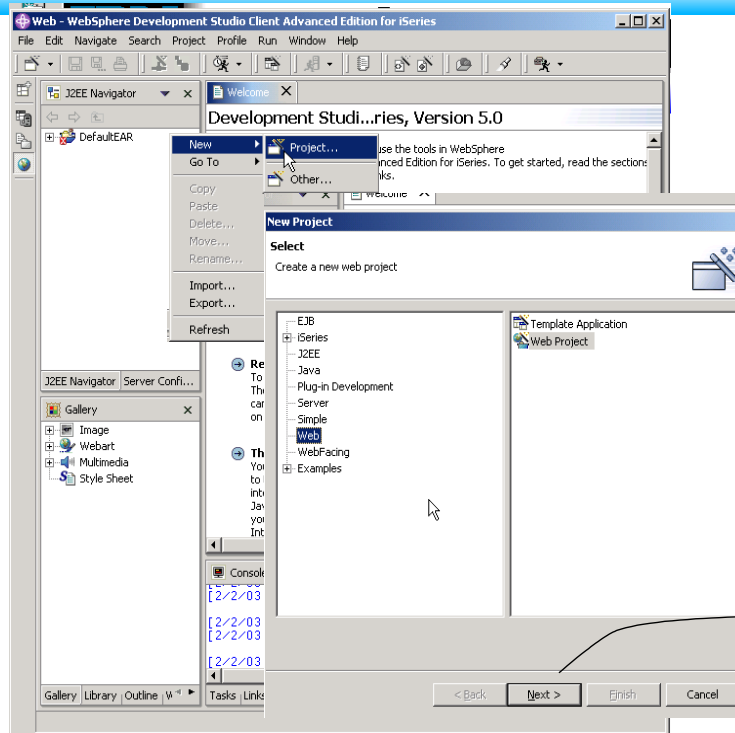
- ▶ An Interaction is defined as an Input Page which allows data input, this data is submitted to a server and processing of the request is done on the server. The server will then respond with serving a result page. The content of the result page is based on the data on the input page
- ▶ The combination of the Input Page and the Output page with the Server in the middle handling the dynamic requests is called an Interaction.



Create a Web project



iSeries AD, IBM Toronto



Eclipse

IBM @server. For the next generation of e-business.



Create a Web Project



iSeries AD, IBM Toronto

Create a Web Project

Struts Based

Project name: MyFirstWebApplication

Use default

New project location: C:\Documents and Settings\far\My Documents

J2EE Web Project Static Web Project

Description:

In a J2EE Web Project you will be able to create content serve HTTP server (HTML, JavaScript, images, text...) as well as cont by a J2EE Application Server (Servlets, JSPs, EJBs...)

Web Project features:

- Add Struts support
- Create a default .eclipseignore file
- Create a default CSS file
- Include Tag Libraries for accessing JSP ot
- Include Tag Libraries for database access
- Include Tag Libraries for internationalizat

Description:

Select this featur
Struts added to:

< Back Next >

Create a Web Project

J2EE Settings Page

Set the Enterprise Application project settings, context root, and J2EE level.

Enterprise application project: New Existing

Existing project name: DefaultEAR Browse...

Context root: MyFirstWebApplication

J2EE Level: 1.3

Description:

J2EE Level 1.3 includes a Servlet Specification level of 2.3 and a JSP Specification level of 1.2. Applications developed for this J2EE level typically target a WAS version 5.0 server.

< Back Next > Finish Cancel

Next page

Eclipse

IBM @server. For the next generation of e-business.

Create a Web Project

iSeries AD, IBM Toronto

Create a Web Project
Module Dependencies
Select dependent JARs for the module within the Enterprise Application project.

Project name: MyFirstWebApplication
Enterprise application project name: DefaultEAR

Available dependent JARs:

JAR/Module	Project

Manifest Class-Path:

< Back Next > Finish

Create a Web Project
Struts Settings
Select the initial settings for Struts

Override default settings

Settings:

Struts version: 1.0.2

Default Java package prefix: com.ibm.myfirstwebapplication

Create a Resource Bundle for the Struts Project

Resource bundle:

Java package: com.ibm.myfirstwebapplication.resourc Browse...

Resource bundle name: ApplicationResources

Done!

< Back Next > Finish Cancel

Eclipse

IBM @server. For the next generation of e-business.

- ▶ You can create a Web application based on a Struts infrastructure

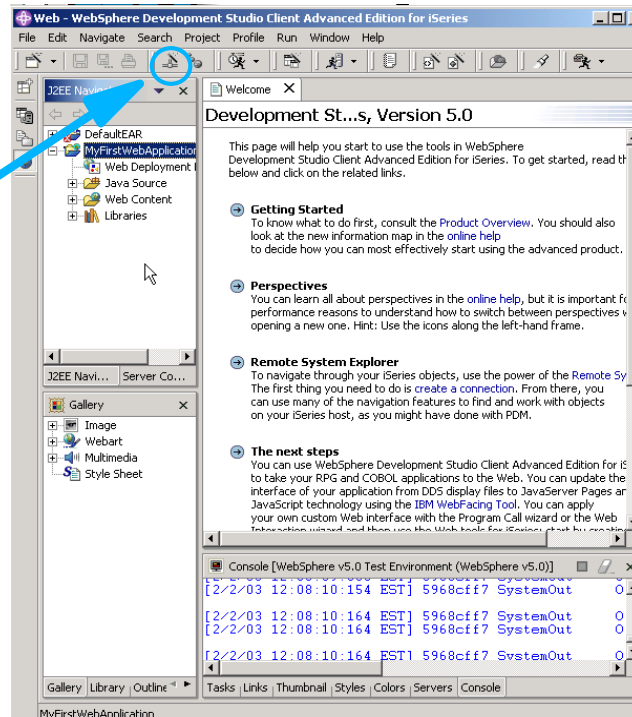


Server Information



iSeries AD, IBM Toronto

iSeries
RunTime
configuration!



Eclipse

IBM @server. For the next generation of e-business.

- ▶ Here we specify what server our Program or Server Program resides on
- ▶ Which Userid / Password we want to use to start a job on our server
- ▶ Also this dialog allows us to add specific libraries to the job environment if this is needed for the Interaction

iSeries Web Tools

▶ iSeries Web Host Information Wiz

- Information used at runtime by all generated Web servlets in this Web project
 - ▶ Name of host containing *PGM or ILE Procedure
 - ▶ User ID and password for signing on
 - ▶ Library list to set for *PGM/Proc job

IBM @server. For the next generation of e-business.

- ▶ The iSeries Web Tools Run-time Configuration wizard captures information such as server name, userid, password and library list used by Web applications

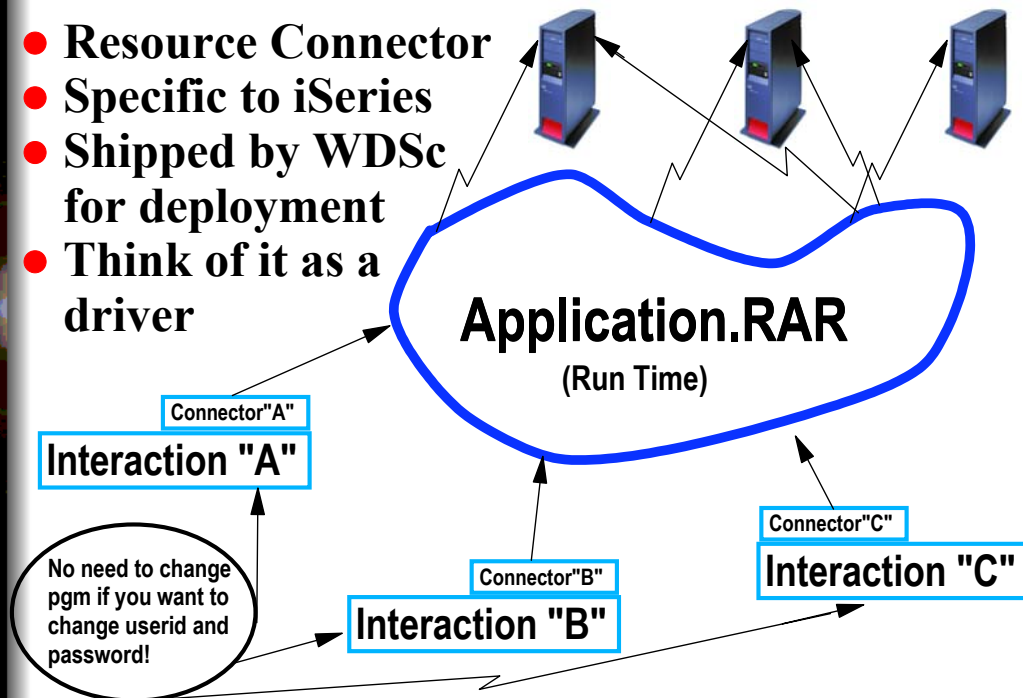


What is JCA Connectors?



iSeries AD, IBM Toronto

- Resource Connector
- Specific to iSeries
- Shipped by WDS*c* for deployment
- Think of it as a driver



Eclipse

IBM @server. For the next generation of e-business.

- ▶ JCA Program Call connector provides a standard interface to call any *PGM, based on the J2EE Connector Architecture

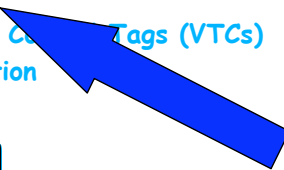


AGENDA



iSeries AD, IBM Toronto

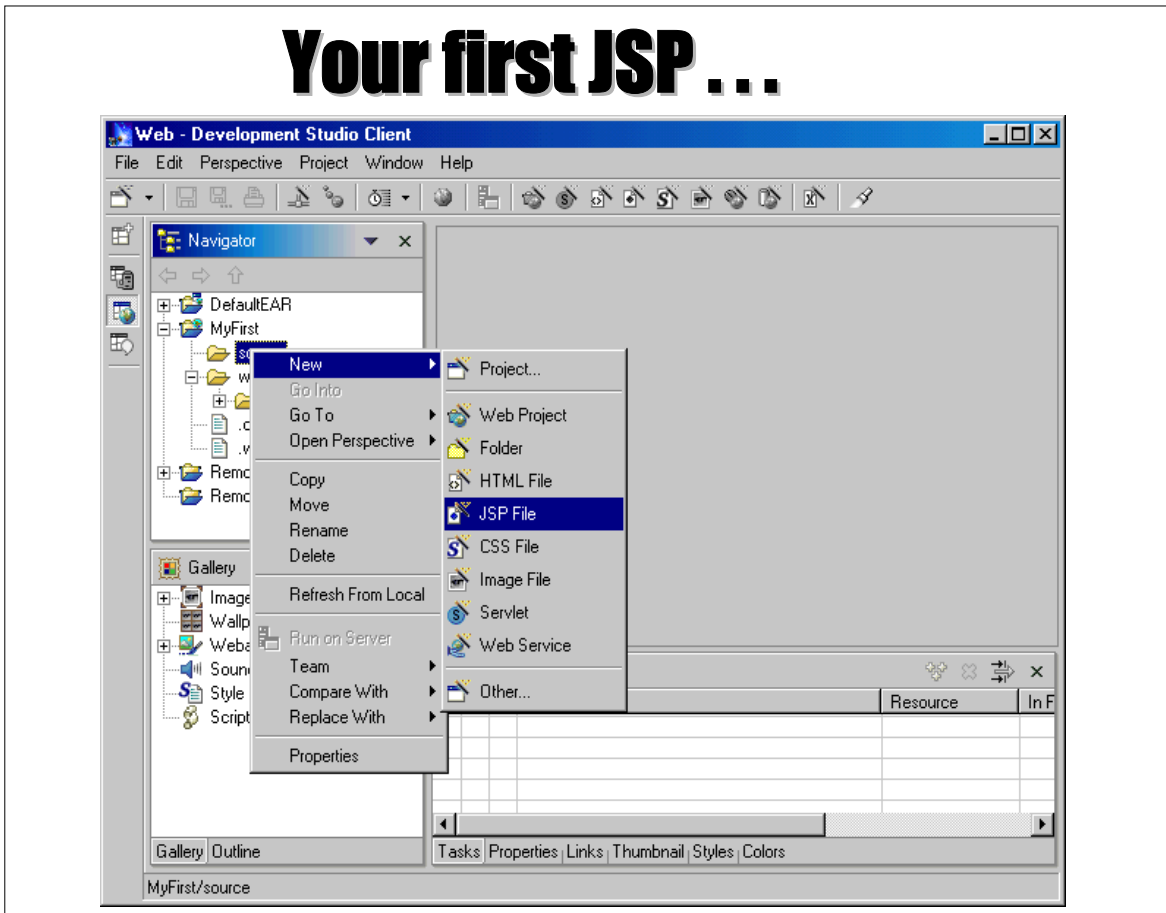
- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Visual C... Tags (VTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**



Eclipse

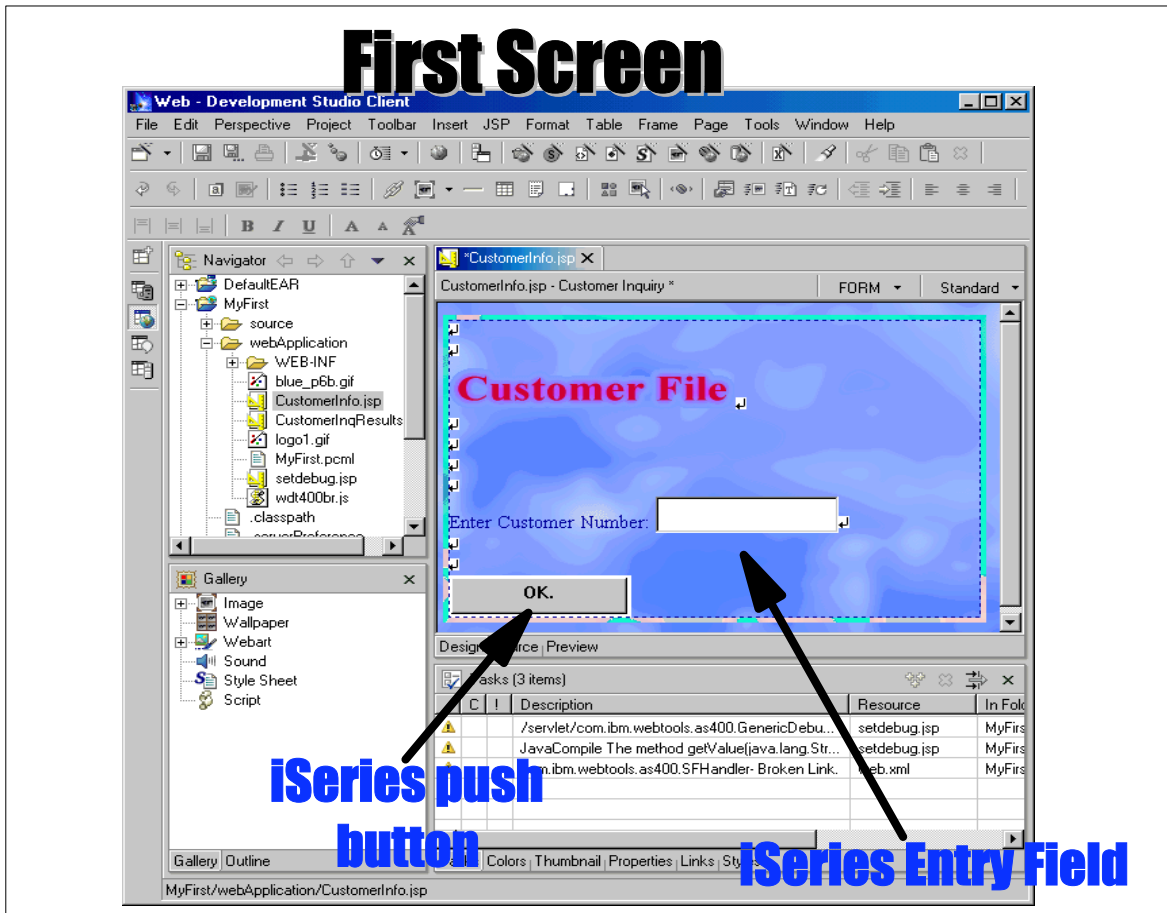
IBM @server. For the next generation of e-business.

Your first JSP ...

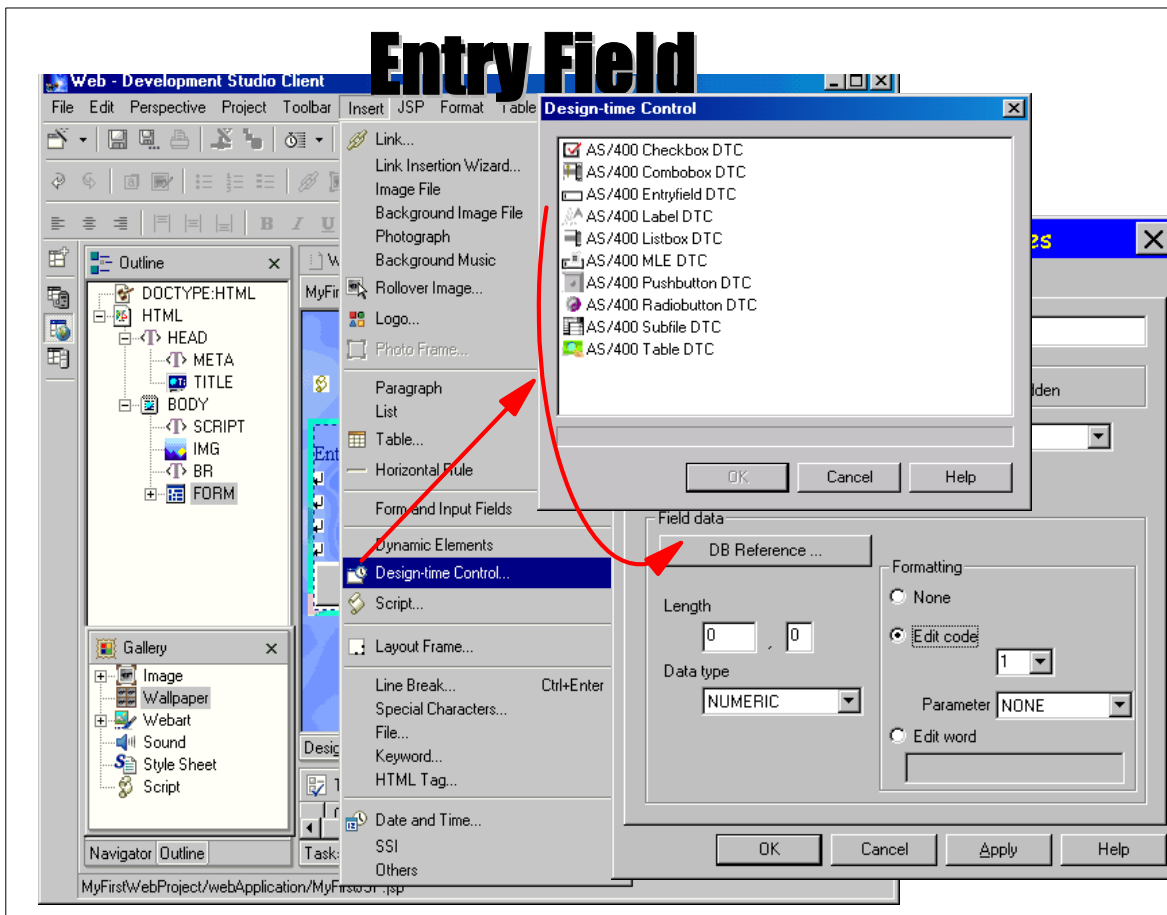


- ▶ First we want to build a user interface for a simple input page
- ▶ We select to create a new jsp file. This will automatically invoke the Page Designer

First Screen

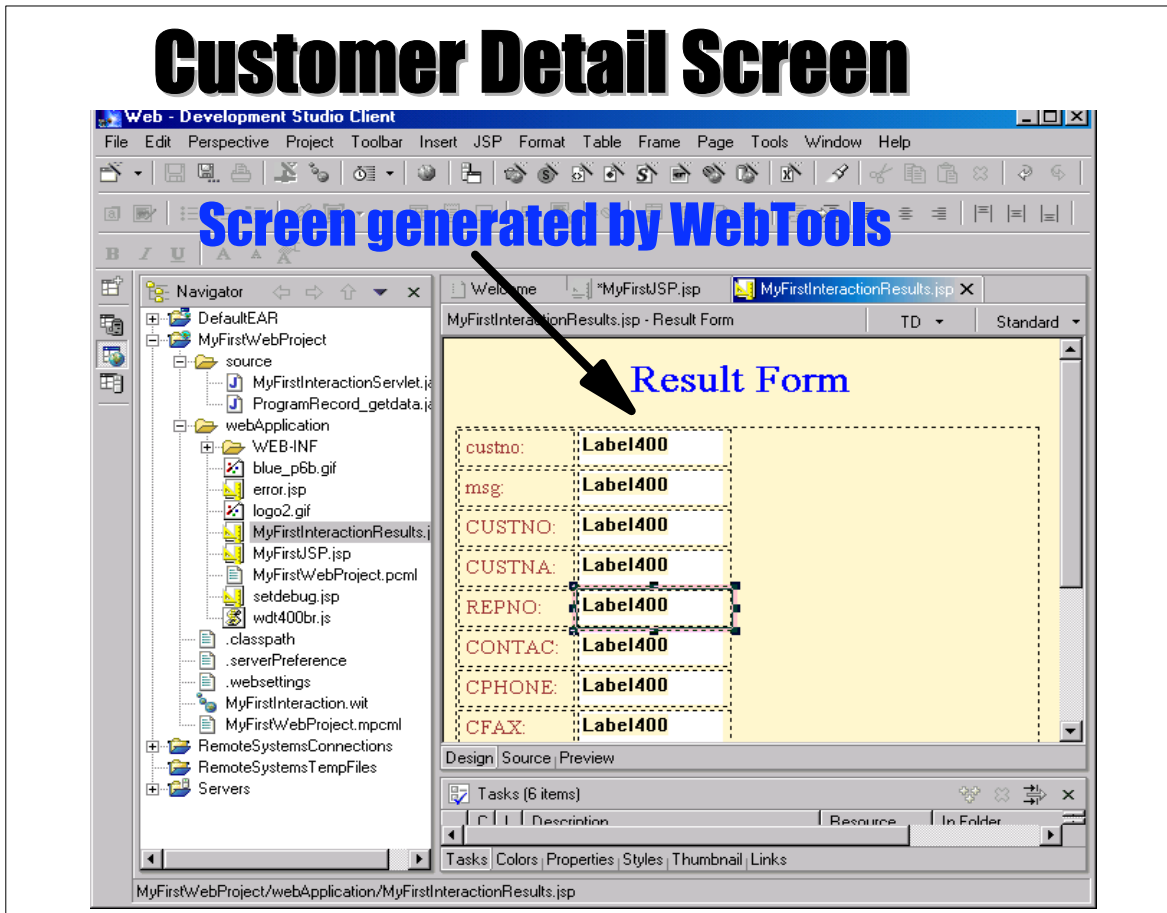


- ▶ This slide shows the Page Designer tool, which can be used to enhance the user interface.
- ▶ Putting a nice background to the page and changing some of the fonts will increase the look of the Web page.
- ▶ We also add a form tag to our page and an iSeries entry field and an iSeries push button.
- ▶ These iSeries controls are specific controls that inherit some of the capabilities of 5250 parts that normal html controls don't provide.
- ▶ For example the capability of referencing iSeries data base fields and the capability of specifying edit codes.



- ▶ Here is a list of all the iSeries special controls provided in the WDSC tools. Notice that even a subfile has been added to the list.

Customer Detail Screen



- ▶ We just looked at the input screen for our little application.
- ▶ This slide shows the Output or result screen.
- ▶ It uses reference fields from the iSeries database.
- ▶ The application will send data from the entry field on the input page to an RPG program, which will extract data from the data base and pass this data back
- ▶ The resulting data will be displayed on this page

RPG IV Program

NOTE Parameters to be passed

```
FCUSTOML3  IF  E          K Disk
DCustnoi   s          like(CUSTNO)
D*
D CSTRUC   E DS          extname(customl3:custom01)
d field2   10
D return   s          20
D*-----
c          *entry      plist
c          parm          custnoi
c          parm          cstruc
c          parm          return
c          eval         return=*blank
c          custnoi      chain   customl3          5050
c          if           *in50
c          eval         return='CUS0001 ' + CUSTNOI
c          else
c          eval         return='0'
c          endif
c          return
```

- ▶ This slide shows the RPG program that gets invoked
- ▶ A customer number is used to specify which customer data should be displayed. The first parameter custnoi will contain the customer number.
- ▶ The data structure CSTRUC will contain the full customer data passed back to the Servlet running in the Application server which will invoke the result jsp to show the data
- ▶ The parameter return, contains information whether the access to the data base was successful or not

Files Used

A* Logical file description

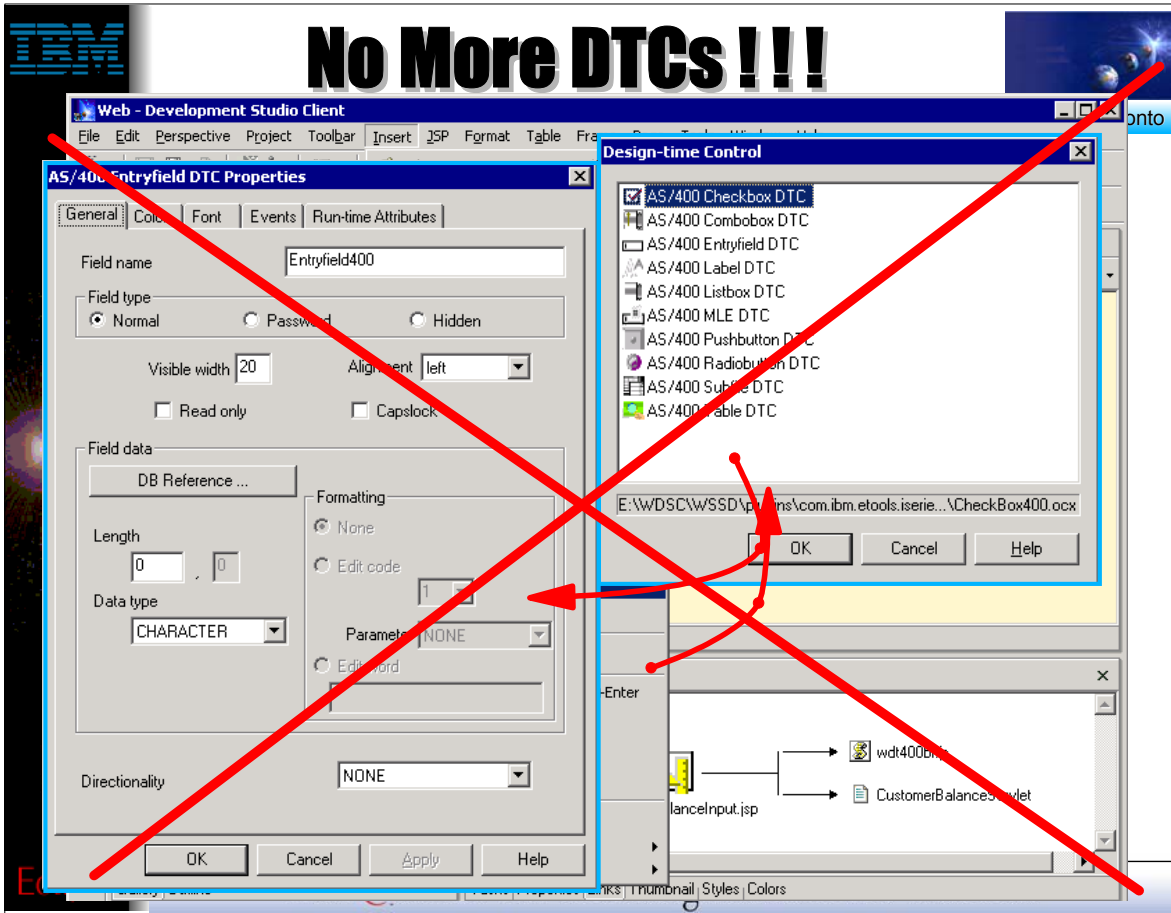
```
A          R CUSTOM01                PFILE(CUSTOMER)
A          K CUSTNO
```

A* Physical file description - CUSTOMER FILE

```
A          R CUSTOM01
A          CUSTNO          7          COLHDG('Customer number')
A          CUSTNA         40          COLHDG('Company name')
A          REPNO           5          COLHDG('Rep identifier')
A          CONTAC         30          COLHDG('Name')
A          CPHONE         17          COLHDG('Telephone')
A          CFAX           17          COLHDG('Fax')
A          CADDR          40          COLHDG('Address')
A          CCITY          30          COLHDG('City')
A          CCOUNT       20          COLHDG('Country')
A          CZIP           10          COLHDG('Postal Code')
A          CZIPLO         1          COLHDG('PC location')
A
A          VALUES('1' '2' '3')
A          K CUSTNO
```

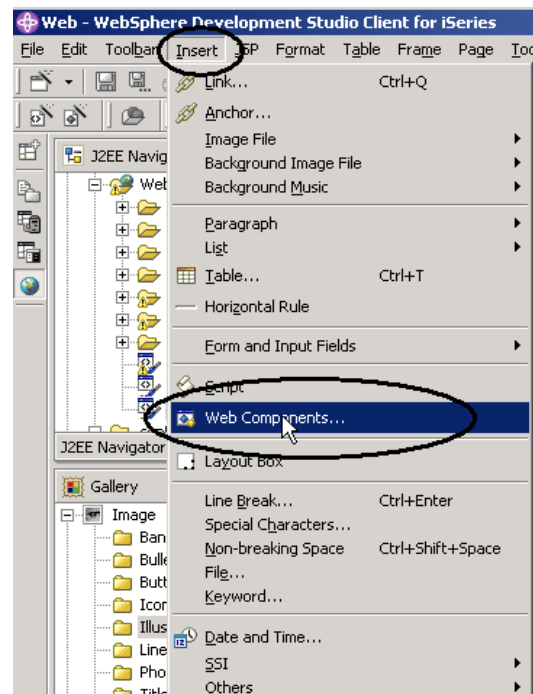
- Here is a description of the files used in this sample application.

No More DTCs !!!





- **VTC: Visual Control Tags**
 - ▶ Emerging technology
 - ▶ Blessed by IBM and SUN
- **Tag library defined by:**
 - ▶ www.sun.com
- **Extendable**
- **DTCs: Old fashion**
 - ▶ ActiveX technology



- ▶ VCTs replace the ActiveX Design-Time Controls (DTCs). VCTs can be used and rendered in the HTML Designer.

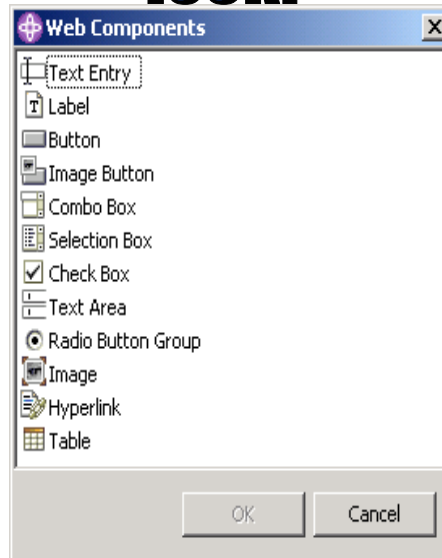


VTCs ... No more DTCs



iSeries AD, IBM Toronto

Same selection ... Better look!



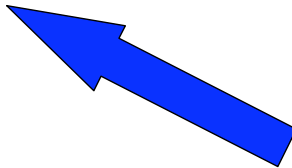
Eclipse

IBM @server. For the next generation of e-business.

- ▶ VCTs provide the controls such as textField, label, dropDown, pushButton, and so on.



- **What is in WSc 5.0?**
- **e-business Primer**
 - ▶ AD Model, traditional and web
- **Web Application Primer**
 - ▶ JSPs, Servlets, forms, etc
- **Mixing Java and RPG**
- **What is Web Tool for iSeries**
- **Introducing WSc for iSeries**
 - ▶ RPG Example
 - ▶ iSeries Visual Control Tags (VTCs)
 - ▶ Web Interaction
 - ▶ Publishing
- **Conclusion**





Web Interaction Wizard



iSeries AD, IBM Toronto

▶ Two modes to interaction wizard:

1. Generate input/output Web pages

- ▶ Given the parameter description of the API to call

2. Generate mappings

- ▶ Given the input/output pages
- ▶ Given the parameter description of the API to call
- ▶ Given the mappings
 - ✓ between input parms & input fields
 - ✓ between output parms & output fields

Eclipse

IBM @server. For the next generation of e-business.

- ▶ This slide shows that we have a choice of having the Interaction Wizard create our User Interface or design it ourselves.
- ▶ In this example we decide to design the page ourselves. use Page Designer to add a background and heading.
- ▶ We add a form tag to the Input page. The form contains an entry field for data entry and a submit push button to send a request with the data in the entry field to a server.

Interaction Wizard

Web Interaction Wizard

Specify a Name and Location for your Web Interaction

Define the name and location for the communication between Web pages and the chosen business process. The location must be the webApplication directory inside a project.

Destination folder: /MyFirst/webApplication [Browse...]

Web Interaction name: CustomerInquiry

Java package: [Browse...]

Use error page [Browse...]

Invalidate session after the interaction occurs

Just a name!!!
Like a field name

< Back Next > Finish Cancel

- ▶ After finishing our Input page, we are ready to start the Interaction wizard
- ▶ This slide shows that we need to specify the Interaction name and go on to the next page.



WDS*c*: iSeries Web Tools

iSeries AD, IBM Toronto

▶ iSeries Web Interaction Wizard

Web Interaction Wizard
Specify a Name and Location for your Web Interaction

Destination folder: /MyWebProject/webApplication

Web Interaction name: carRentTest1

Java package: **car.rent.test1**

Use error page

Invalidate session after the interaction occurs

< Back Next > Finish Cancel

Web Interaction Wizard
Specify the Input and Output Pages for your Web Interaction

You can use previously designed input and output pages for this Web interaction or you can have the wizard generate them.

Generate input JSP

Use input pages:

Generate output JSP

Use output pages:

Mode 1: Generate input and output Web pages

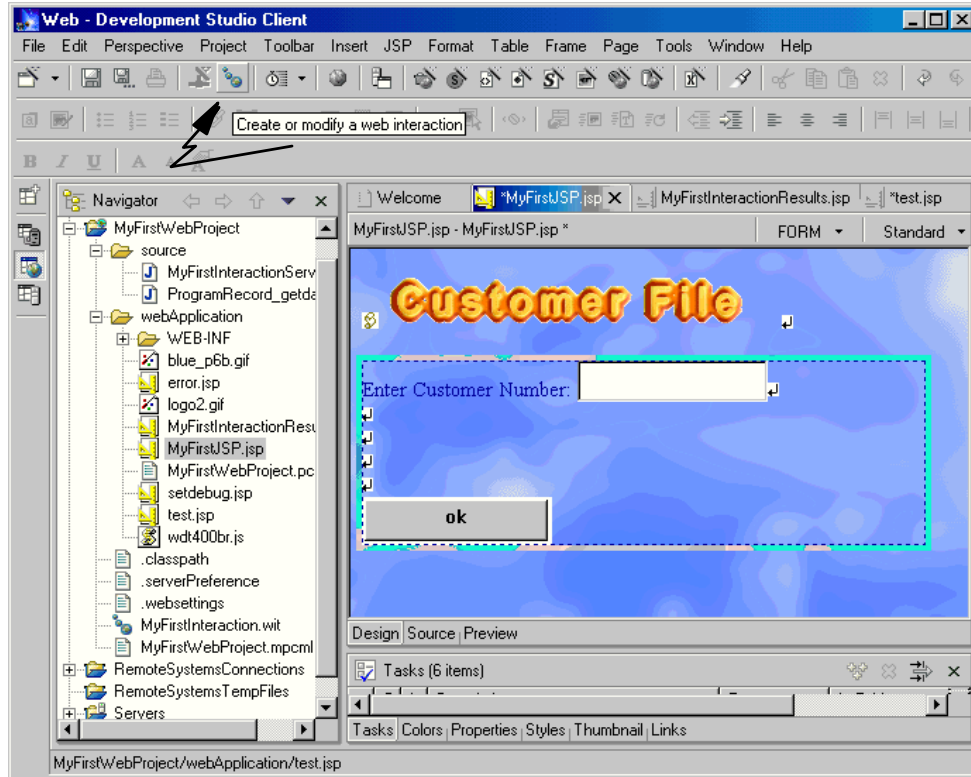
< Back Next > Finish Cancel

Tip: pre-create a Java package in your Web project, under source folder, and specify it here

Eclipse

IBM @server. For the next generation of e-business.

Or ... Create your own!



Web Interaction Wizard

Specify the Input and Output Parameters for your iSeries Host Program

The program to be invoked by this interaction.

Use an iSeries ILE program (selected) | Use a Java bean | No program call

Program call definitions: CarRent

Buttons: Add Program... (1a), Add Parameter..., Add Structure...

Edit Program

Program alias: CarRent

Program object: RENTCAR

Library: CARDEMO

Program type: *PGM

Entry point: (highlighted with red box)

Return type: void

Thread safe: false

Source location: (View...)

Associate this program with the interaction

Buttons: Import PCML..., Synchronize All, OK (1c), Cancel

Wizard navigation: < Back, Next >, Finish, Cancel

Annotations:

- Describe program / procedure
- 1a. Press "Add Program", 1b. Enter program information, 1c. Press Ok
- For *SRVPGM, enter procedure name

Eclipse | IBM @server. For the next generation of e-business.

- ▶ Shows the next page of the wizard. On this page we need to specify the interface to our program on the iSeries
- ▶ The wizard will generate PCML out of this information
- ▶ We need to specify the program to call, with information which library the program resides in
- ▶ We could also specify to call a procedure in a Service program.

IBM iSeries Web Interaction Wiz

Web Interaction Wizard
Specify the Input and Output Parameters for your iSeries Host Program
Use this page to define the input and output parameters for your iSeries host program.

Use an iSeries ILE program | Use a Java bean | No program call

Program call definitions: CarRent, carClass, carMake, carColor, retPlate

Edit Parameter
 Parameter name: carClass
 Data type: character
 Structure name: **3**
 Length: 10
 Precision:
 Count: **3**
 Usage: input
 Initial value: *ANY

2a, **2b**, **2c**

3

2a. Press "Add Parameter",
2b. Enter parameter information
2c. Press Ok
2d. Repeat for each parm

3a. Pre-define structures before referencing them for parms

For arrays

Input: read by program
Output: updated by program
Input/Output: both

Reference DB fields!

Next >

Eclipse IBM @server. For the next generation of e-business.

- ▶ Then we need to specify the parameters expected by the program
- ▶ If the parameters are data structures we can define these here as well
- ▶ All parameters can be defined by using reference fields

Web Interaction Wizard
Design the Input Form

Input parameters:

- carClass
- carMake
- carColor

Move up
Move down
All
None

Select input parameters to prompt user for, in input page

Property	Value
Label:	carClass
Data Type:	character(10)
Input Style:	Text
Input Choices:	...
Size:	20
Max Length:	40
Name in Session:	carClass
Restore from Session:	false
Save to Session:	false

Tailor attributes of generated prompt per parameter

Input Form

carClass:

carMake:

carColor:

Immediately see results of tailoring parameter and page attributes

Page Fields

Tailor attributes of overall input page

< Back Next > Finish Cancel

Eclipse

IBM @server. For the next generation of e-business.

- ▶ This page allows you to specify which entry field on the Input page contains data for which parameter
- ▶ Entry fields and parameters with the same name are linked together

Web Interaction Wizard
Design the Result Form

Output parameters:
 retPlate

Move up
Move down
All
None

Property	Value
Label:	retPlate
Data Type:	character(10)
Name in Session:	retPlate
Save to Session:	false

Page Fields

Result Form
retPlate: Immediately see results of tailoring parameter and page attributes

< Back Next > Finish Cancel

Eclipse

IBM @server. For the next generation of e-business.

- ▶ Now the result page has to be defined.
- ▶ Automatically all output parameters are shown in the result page, if the page is generated by the wizard
- ▶ To deselect fields from being included on the result page, just uncheck them in the list on the left side of the dialog



iSeries Web Interaction Wiz



Generated files

Use "Run on Server" to test

The screenshot shows the 'Web - Development Studio Client' interface. The main window displays a web form titled 'Input Form' with three dropdown menus labeled 'carClass', 'carMake', and 'carColor', all set to 'Combobox'. Below the form are 'Submit' and 'Reset' buttons. The Navigator pane on the left shows a project structure with folders like 'source', 'car', 'rent', and 'test1', and files like 'carRentTest1Input.jsp'. The Links pane at the bottom shows a diagram where 'carRentTest1Input.jsp' is linked to 'wdt400br.js' and 'carRentTest1Servlet'. The status bar at the bottom indicates the current file path: 'MyWebProject/webApplication/carRentTest1Input.jsp'.

Eclipse

IBM @server. For the next generation of e-business.

You are done! What's Next?

● What did you do so far?

- ▶ You used the designer to create the input page
- ▶ The output page was generated for you
- ▶ Created your Files on the iSeries
- ▶ Created your RPG Program
- ▶ Created an 'iSeries Web Interaction' Interaction to link your program parameters to the input page and output page

● Next?

- ▶ Run locally to test your application Or

- ▶ This slide shows what we just did and what gets generated:
- ▶ The output (result) page jsp
- ▶ The PCML to describe your iSeries program
- ▶ A servlet
- ▶ The Form tag on the input page gets updated with the information which servlet to invoke when the submit push button is pressed
- ▶ We only have to write the RPG program.



WDS*c*: Run On Server



iSeries AD, IBM Toronto

- ▶ **Run On Server**
 - Now this is **VERY COOL!**
- ▶ **When ready to test your Web app**
 - Right click on initial html or jsp file
 - ▶ or whole project, which implies the index.html file
 - Select "Run on Project"
 - Wait for the magic...
- ▶ **Your Web application will run!**
 - Opens Server perspective
 - ▶ Publishes it to built-in copy of WAS
 - ▶ Starts built-in copy of WAS
 - ▶ Brings up a Web Browser
 - ▶ **Runs your application!!**
 - ✓ Tip: you can set breakpoints in your Java code!

Eclipse

IBM @server. For the next generation of e-business.

- ▶ To test this, we can select our .jsp input page in our Web project in the Web perspective in the workbench
- ▶ Right mouse click on it and select Run on Server from the popup window
- ▶ The built-in WAS test environment will start
- ▶ The built-in browser will show the input page
- ▶ Key in a valid customer number
- ▶ Press the Submit push button
- ▶ The customer number will be sent to the RPG program
- ▶ If the number was valid the customer data stored in our data base will be displayed



WDS*c*: WAS Test Env



iSeries AD, IBM Toronto

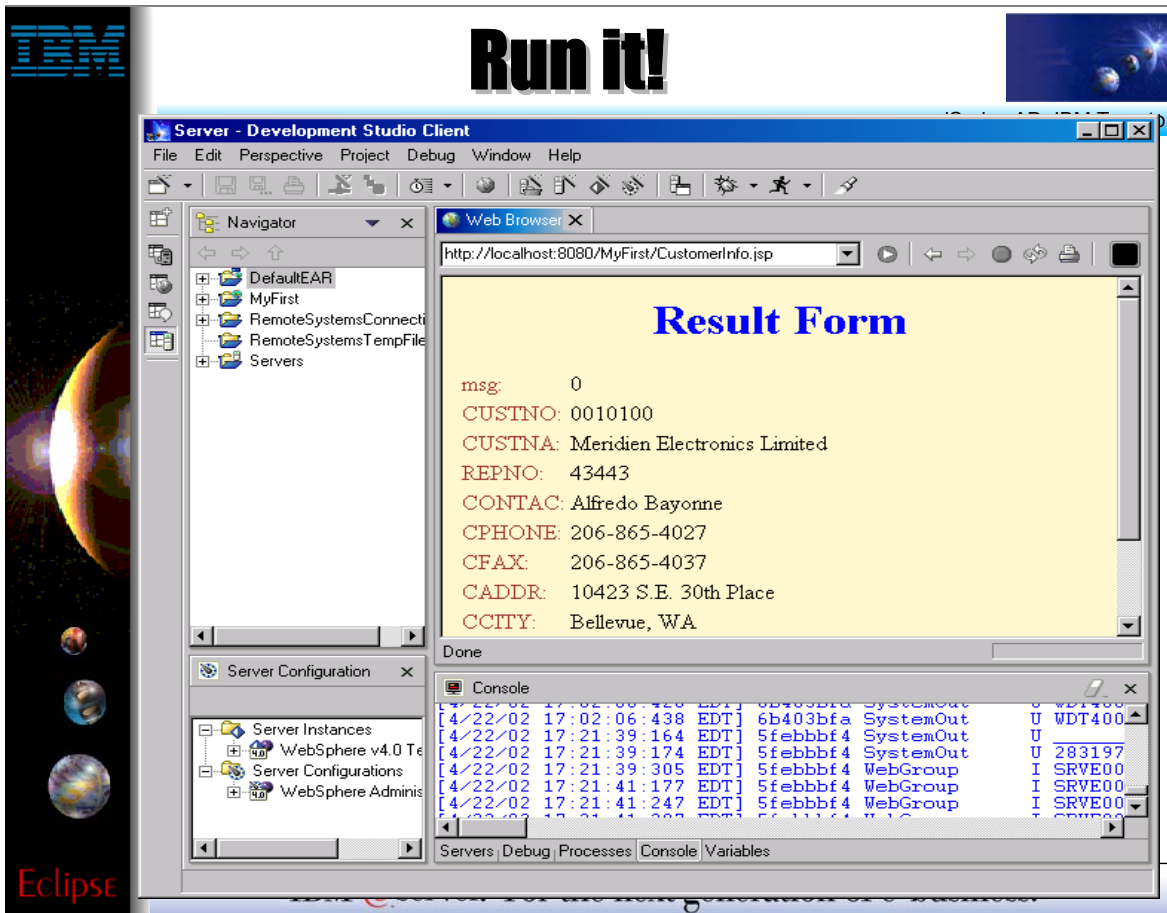
▶ **WebSphere Test Environment**

- **A full copy of WAS 4.0 Single Server Edition is embedded in the IDE**
 - ▶ Integrated with Server Tools to enable instant and dead-easy testing of Web projects within WDS*c*.
 - ✓ Standalone all-in-one testing
 - ✓ No dependency on WAS installation or availability
 - ✓ No dependency on an external database
 - ▶ Provides the ability to debug live server-side code
 - ▶ Supports configuring multiple Web applications
 - ▶ Supports multiple servers that can be configured and run at the same time
 - ▶ Provides access to the profiling feature that is available in the workbench
 - ▶ Provides the ability to version Server Tools server configurations
 - ▶ Provides access to the WAS Administration Client

Eclipse

IBM @server. For the next generation of e-business.

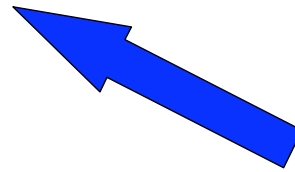
- ▶ The WebSphere test environment provides a fully functional test bed for Web applications without having to setup WAS on a remote server.



► Here we are, the Result page is displayed



- **Web Interaction - Struts based!**



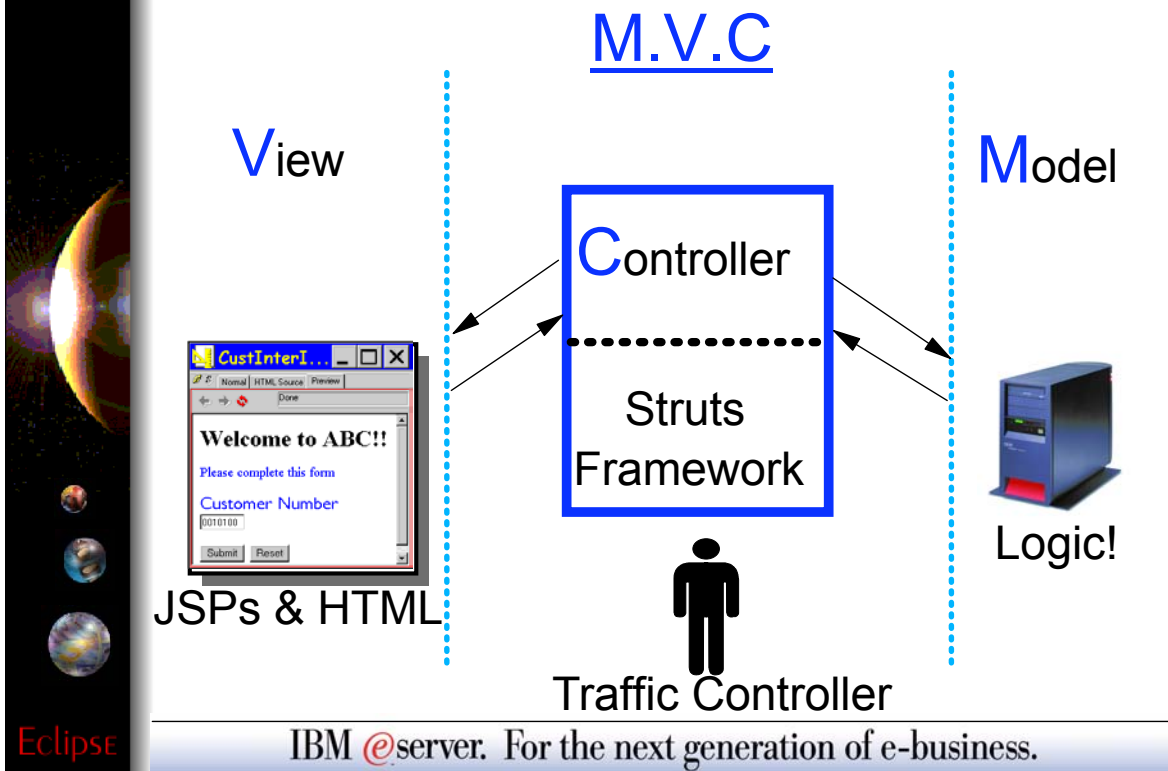
- ▶ Web Interaction wizard now generates a Web application based on a Struts runtime



Web Wizard Struts based ...



iSeries AD, IBM Toronto



- ▶ Struts-based Model View Controller that allows you to map out and visually construct Web-based applications



SUMMARY



iSeries AD, IBM Toronto

▶ **iSeries Web Tools, at a glance**

● **Tools optimized for iSeries developers!**

▶ **Web Interaction wizard**

- ✓ You define the parameters to a *PGM/*SRVPGM, wizard generates input JSP prompting for input parm, output JSP showing output parms, and all the glue in-between
- ✓ Or you pre-create the input and/or output pages, and map the input/output fields on the pages to the input/output parameters in the *PGM/*SRVPGM, and it generates the glue to bind them

▶ **Host Information wizard**

- ✓ Set runtime information such as library list and sign-on information, to be used by glue generated by all Web Interaction wizards for this Web project

▶ **Design Time Controls (Web GUI Widgets)**

- ✓ Web GUI Widgets that support DSPF-like attributes such as error checking by datatype, and edit-code and edit-word

Eclipse

IBM @server. For the next generation of e-business.

- ▶ Wizards make it easy to 'wrapper' existing iSeries programs as Java Beans and/or Web Services. This lets the iSeries developer access these programs from a Java application, a Web application, or include them as the base for a Web Service.
- ▶ Extensions to the Struts builder make it easy to visually construct Web applications based on the open-source Struts standard that accesses iSeries programs
- ▶ Visual Custom Tags bring some of the power of DDS to Web pages



Standards



iSeries AD, IBM Toronto

▶ Industry Standards

Eclipse

IBM @server. For the next generation of e-business.



Standards



iSeries AD, IBM Toronto

▶ **Industry Standards Supported**

● **By WSSDa, WSAD and WDSa:**

- ▶ EJB 1.1
- ▶ Servlet 2.2
- ▶ JSP 1.1
- ▶ JRE 1.3
- ▶ Web Services Definition Language (WSDL) 1.1
- ▶ Apache SOAP 2.1
- ▶ XML DTD 1.0 10/2000 Revision
- ▶ XML Namespaces 1/99 Version
- ▶ XML Schema 5/2001 Version
- ▶ HTML 4.01 (other levels should work)
- ▶ CSS2 (PageDesigner displays a subset)

Eclipse

IBM @server. For the next generation of e-business.

- ▶ WDSa is keeping up with industry standards, most of which IBM wrote or contributed heavily to



More Information?



iSeries AD, IBM Toronto

▶ Information Sources

- www.ibm.com/software/ad/iseriess
 - ▶ As of June 4th
- www.eclipse.org
 - ▶ Eclipse and information about eclipse
- www.ignite400.org
 - ▶ Introduction to eclipse article
- www.ibm.com/software/info1/websphere/partners/iseriess.jsp
 - ▶ WebSphere on iSeries home page for BPs
- **eServer iSeries magazine, July issue**
 - ▶ 3 articles on WDS*c*
- www.ibm.com/websphere/developer
 - ▶ WebSphere Developer Domain
 - ▶ Many articles and tutorials on technology and tools, including eclipse and WSWB and WebSphere Studio configurations

Eclipse

IBM @server. For the next generation of e-business.



Trademarks & Disclaimers



iSeries AD, IBM Toronto

© IBM Corporation 1994-2002. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation. Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both. ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC. Other company, product and service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Eclipse

IBM @server. For the next generation of e-business.