

# 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

# Disclaimer

Web	Java	RPG
		CBL

iSeries AD, IBM Toronto

## 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.



- What is in WDS Sc 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 WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Design time controls (DTCs)
  - ▶ Web Interaction
  - ▶ Publishing
- Conclusion

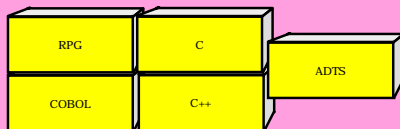


# WDS Sc 5.0 Standard!

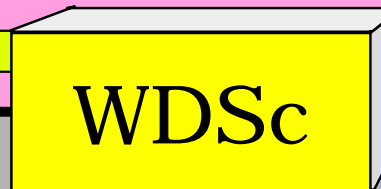


iSeries AD, IBM Toronto

WebSphere Development Studio

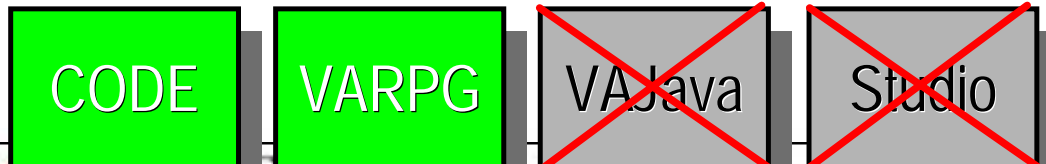
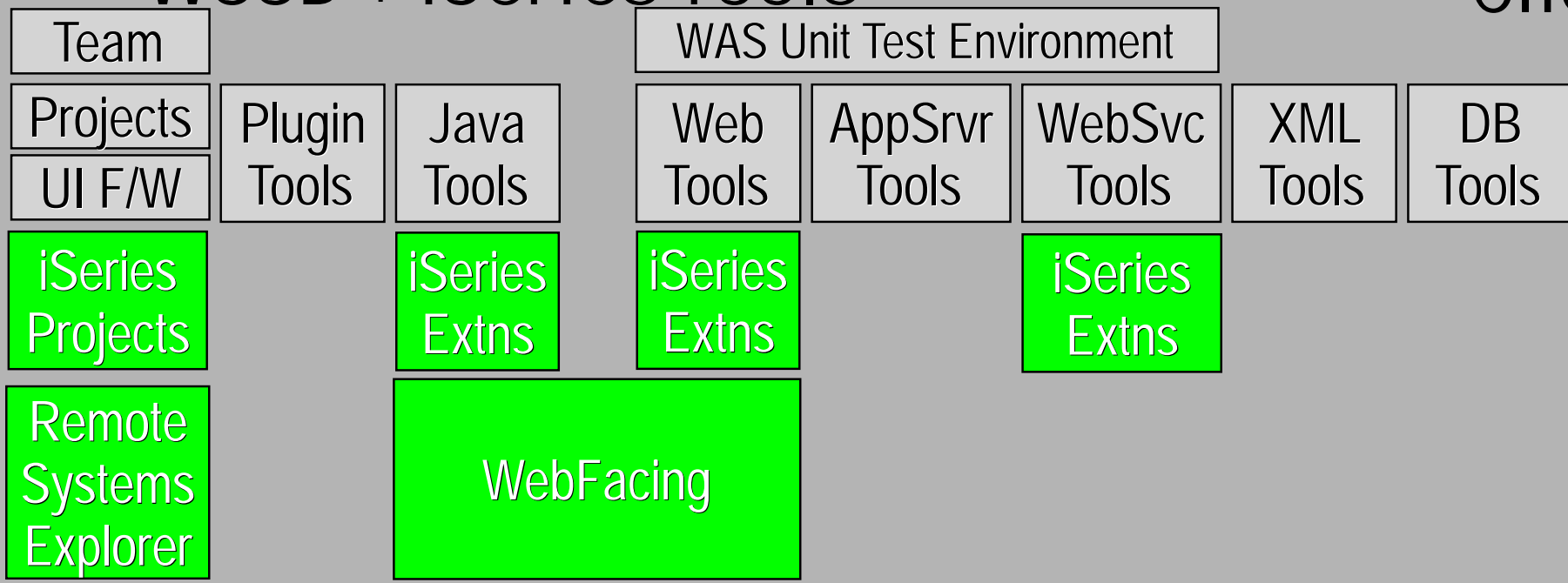


Host



## WSSD + iSeries Tools

## Client



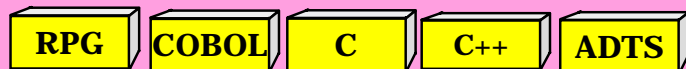


# WDS Sc 5.0 Advanced!



iSeries AD, IBM Toronto

WebSphere Development Studio Advanced Edition



Host

WDS Sc Advanced

WebSphere Studio Application Developer + iSeries Tools

WAS Unit Test Environment

Team											
Projects	Plugin Tools	Java Tools	Web Tools	Struts Tools	EJB Tools	AppSrvr Tools	WebSvc Tools	XML Tools	DB Tools	Profiling Tools	Tracing Tools
UI F/W											

iSeries Projects

iSeries Extns

iSeries Extns

iSeries Extns

Remote System Explorer

WebFacing Tool  
Struts compliant runtime

Client

CODE

VARPG

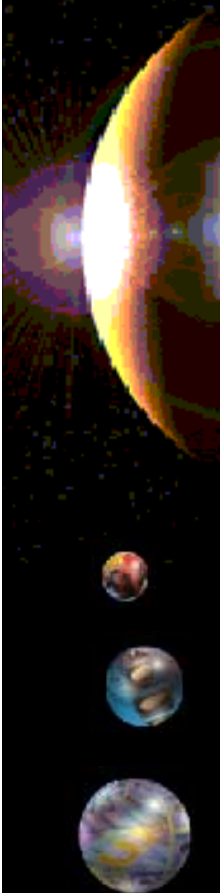
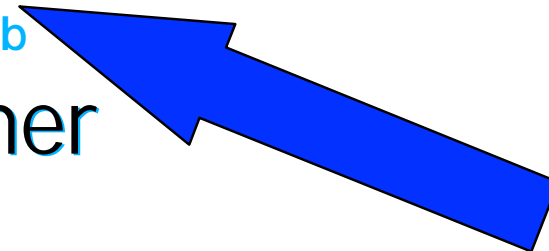
~~VJava~~

~~Studio~~

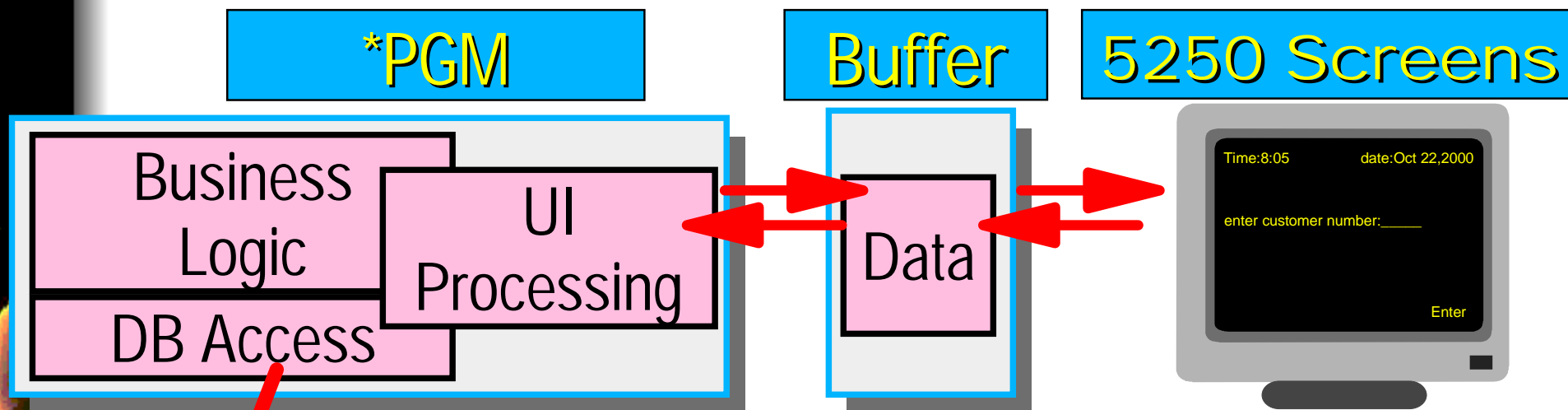
IBM @server. For the next generation of e-business.



- What is in WDS Sc 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 WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Design time controls (DTCs)
  - ▶ Web Interaction
  - ▶ Publishing
- Conclusion



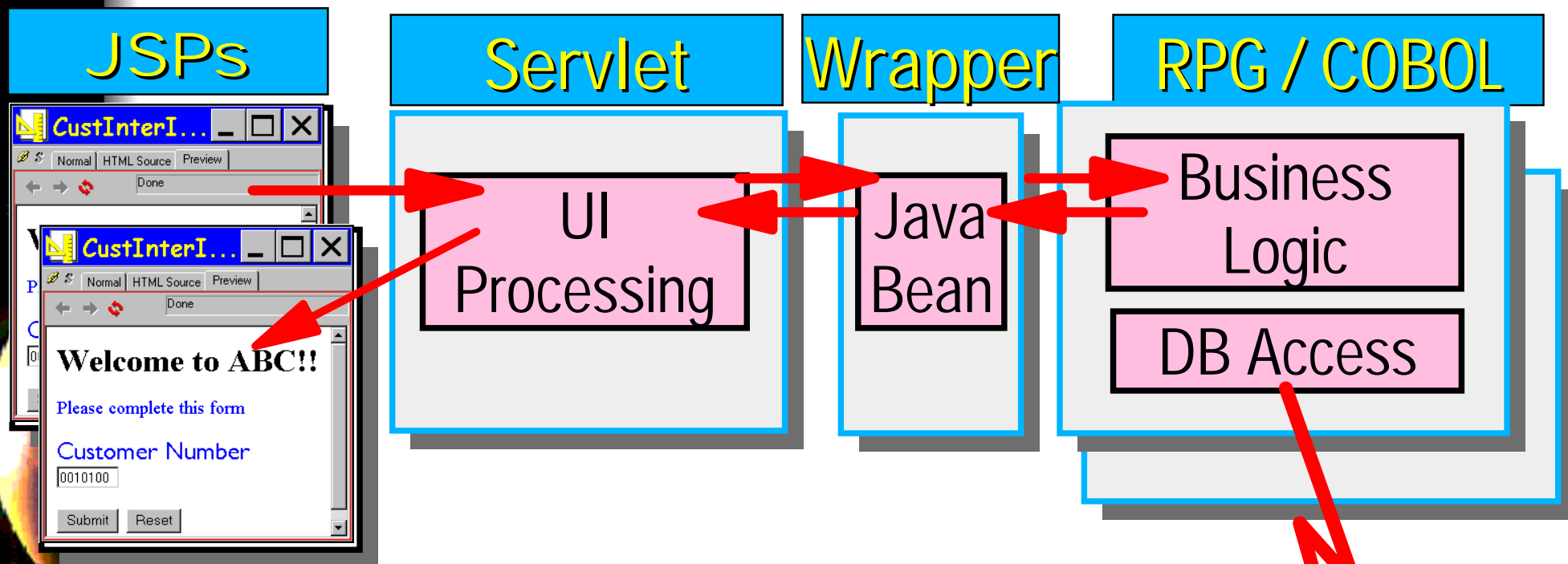
# Today's Model



B2/400

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





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



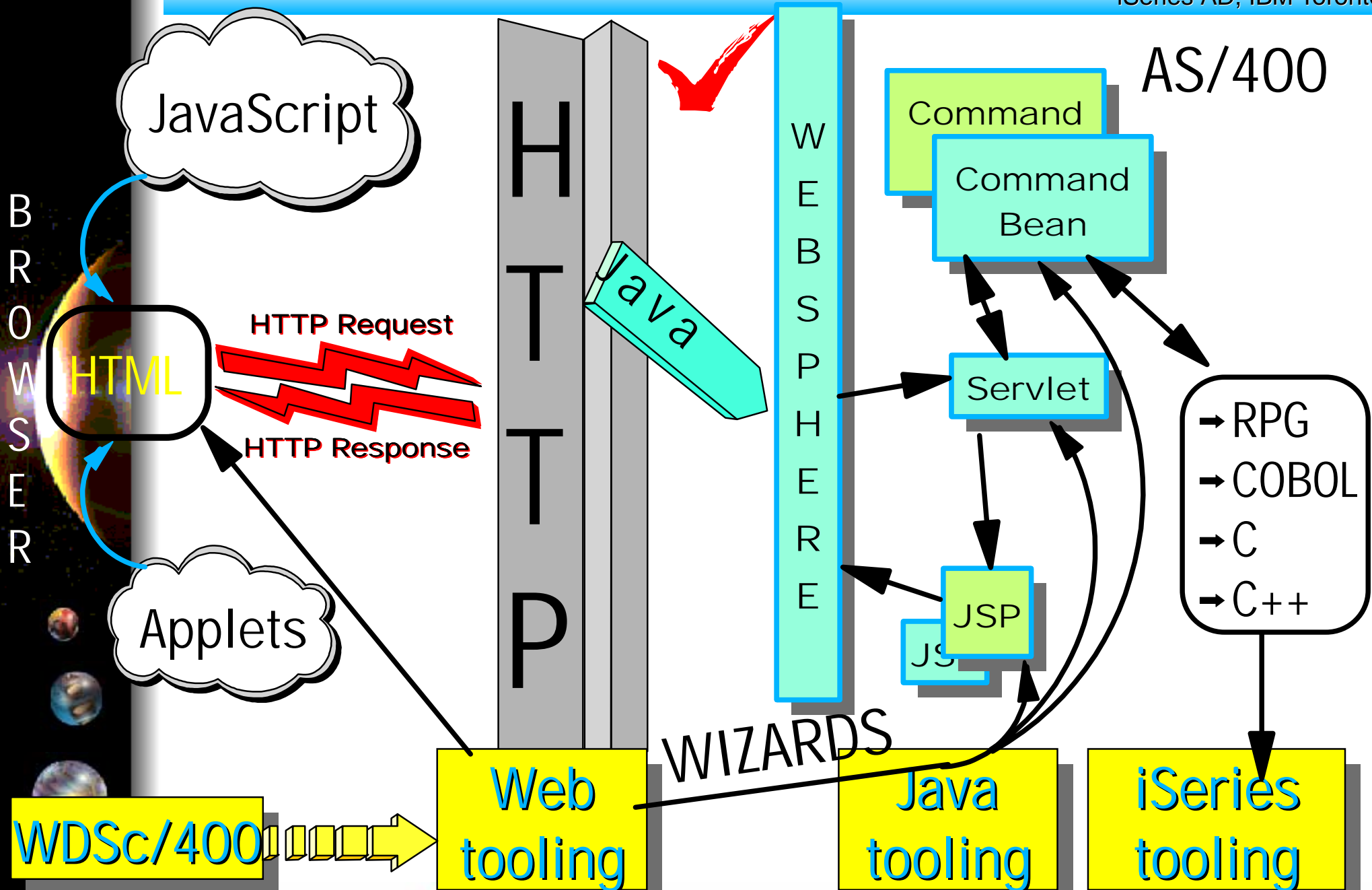




# Big Picture



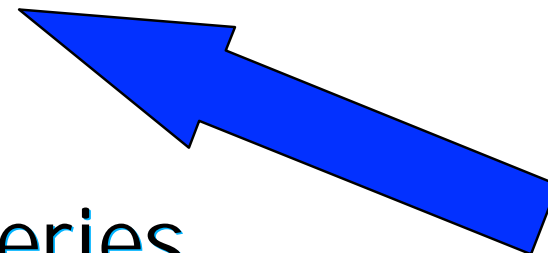
iSeries AD, IBM Toronto



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

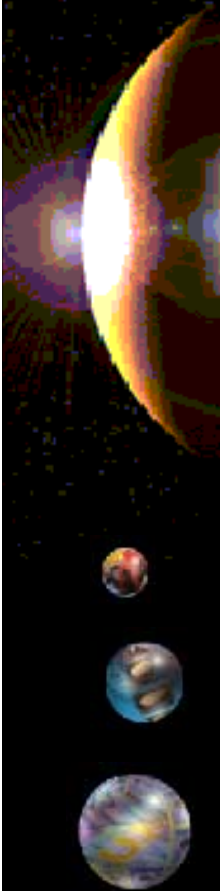


- What is in WDS Sc 5.0?
- e-business Primer
  - ▶ AD Model, traditional and web
- Web Application Primer
  - ▶ JSPs, Servlets, forms, etc
- Mixing Java and RPG
- What is Web Tool for iSeries
- Introducing WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Design time controls (DTCs)
  - ▶ Web Interaction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





- ▶ 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



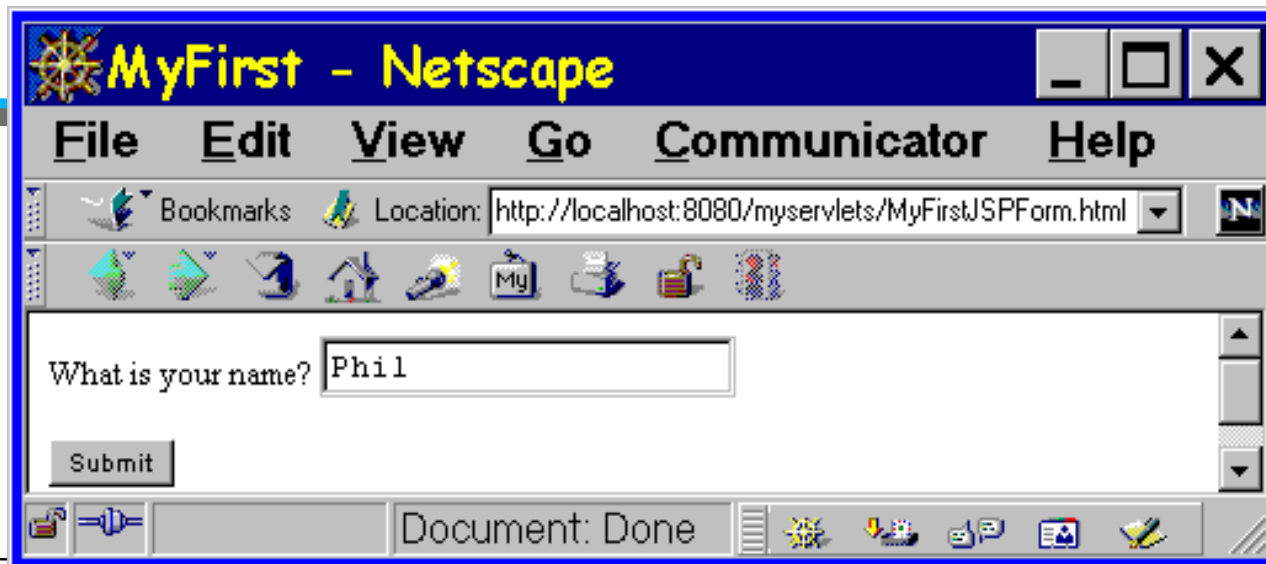
# JSP example ...



`\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



# JSP Example ...

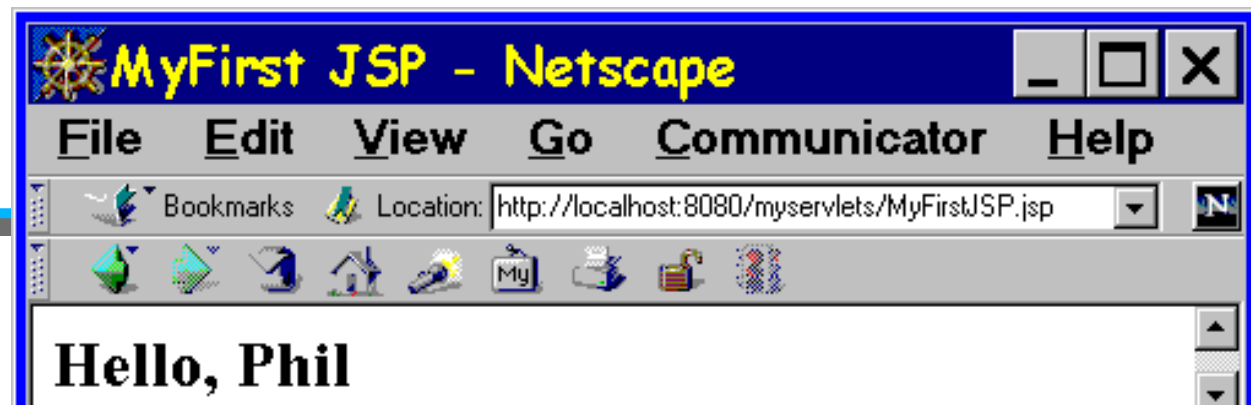


`\myservlets\WEB-INF\jsp\MyFirstJSP.jsp`

```
<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!



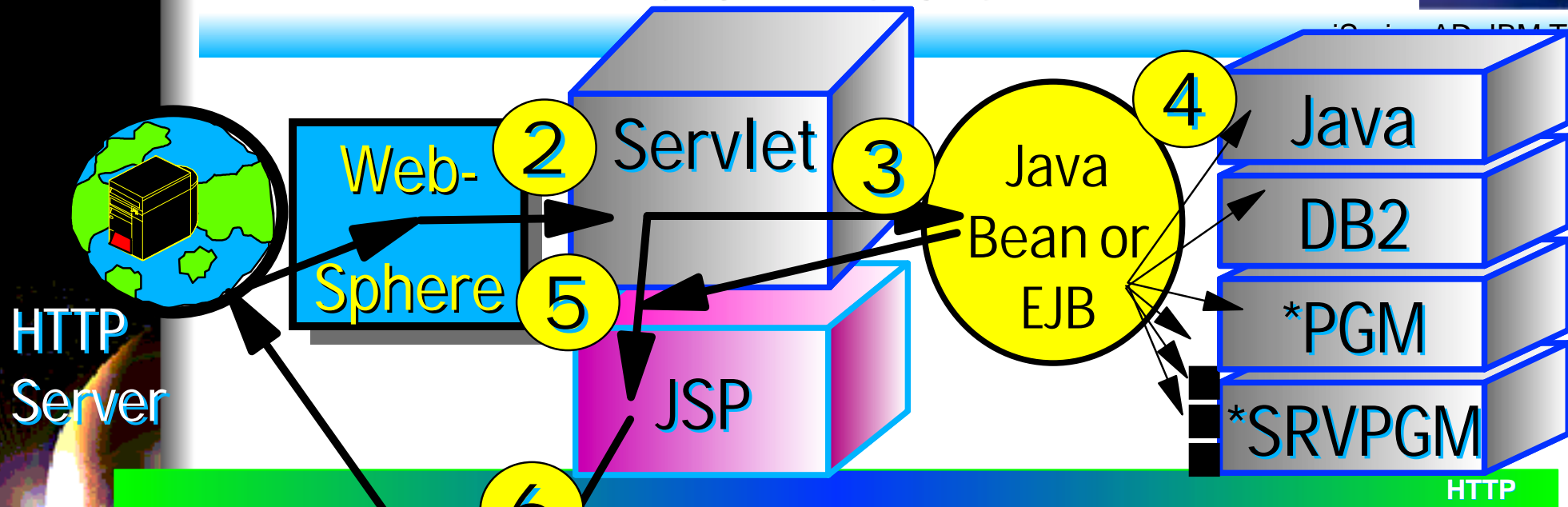
# What Are Servlets?



- ▶ 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**



# Web Model



1. Servlet called by user or HTML FORM
2. Web Server calls Servlet
  - ▶ Servlet optionally reads input fields
3. Servlet calls Java Bean to get data
4. Java Bean gets its data
  - ▶ From other Java code, DB2, RPG, whatever
5. Servlet calls JSP
  - ▶ Passes Java Bean containing data
  - ▶ JSP merges data with HTML => new page
6. HTTP Server sends new page to user





- ▶ 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



# Anatomy of a FORM



iSeries AD, IBM Toronto

```
<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



# Example of a FORM



iSeries AD, IBM Toronto

```
<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



The screenshot shows a web browser window with the following elements:

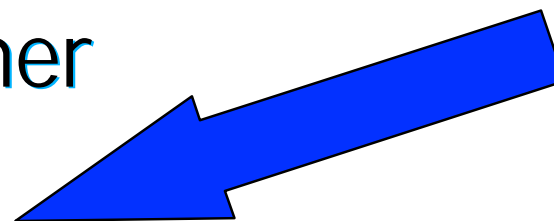
- Title Bar:** "A Simple Input Form ..." with standard window controls.
- Menu Bar:** File, Edit, View, Favorites, Tools.
- Navigation Bar:** Back, Forward, Stop, Refresh, Home.
- Address Bar:** "Address" followed by a dropdown menu showing "E:\WebSphere\St" and a "Go" button.
- Form Fields:**
  - Name:** An empty text input field labeled "entry field".
  - Age:** An empty text input field.
  - Country:** A dropdown menu showing "Canada" labeled "selectable list".
  - e-mail:** An empty text input field.
  - Gender:** Radio buttons for "Male" (selected) and "Female" labeled "radio buttons".
  - Subscription:** A checked checkbox for "e-mail me" labeled "check box".
  - Buttons:** "Register" and "Reset" buttons. The "Reset" button is labeled "RESET button".
- Taskbar:** Shows "My Computer" icon.

SUBMIT button





- What is in WDS Sc 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 WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Design time controls (DTCs)
  - ▶ Web Interaction
  - ▶ Publishing
- Conclusion





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.



# 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>
```





# 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.getback.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



# Mixing Java and RPG



iSeries AD, IBM Toronto

## Results . . .

The screenshot shows a Windows Command Prompt window with a blue title bar labeled "Command Prompt". The text inside the window is as follows:

```
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>
```

Overlaid on the bottom right of the Command Prompt is a "Signon to AS/400" dialog box. It has a blue title bar with the AS/400 logo and the text "Signon to AS/400". The dialog contains the following fields and options:

- System: TORASB5D
- User ID: FARR
- Password: \*\*\*\*
- Default User ID
- Save password
- OK button
- Cancel button



- What is in WDS Sc 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 WDS Sc for iSeries
    - ▶ RPG Example
    - ▶ iSeries Design time controls (DTCs)
    - ▶ Web Interaction
    - ▶ Publishing
  - Conclusion
- 



## ▶ 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





## ▶ Example Web Application

+accounts

+source

+webApplication

-index.html

[www.mydomain.com/accounts](http://www.mydomain.com/accounts)

+receivable

-page1.html

[www.mydomain.com/accounts/receivable/page1.html](http://www.mydomain.com/accounts/receivable/page1.html)

+payable

-page1.html

[www.mydomain.com/accounts/payable/page1.html](http://www.mydomain.com/accounts/payable/page1.html)

+META-INF

-MANIFEST.MF

+theme

-corporate.css

+WEB-INF

-web.xml

+classes

+lib



## ▶ Web Archive Files (WAR)

J2EE  
Servlet  
Spec  
2.2

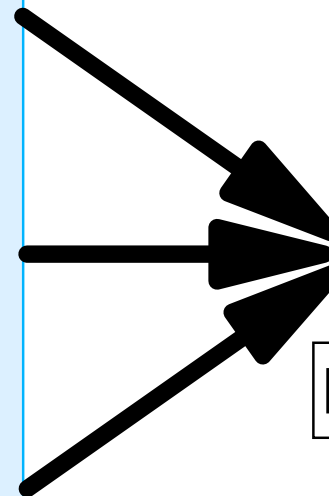
- **One file containing**

- ▶ 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



J2EE  
EJB  
Spec  
1.1

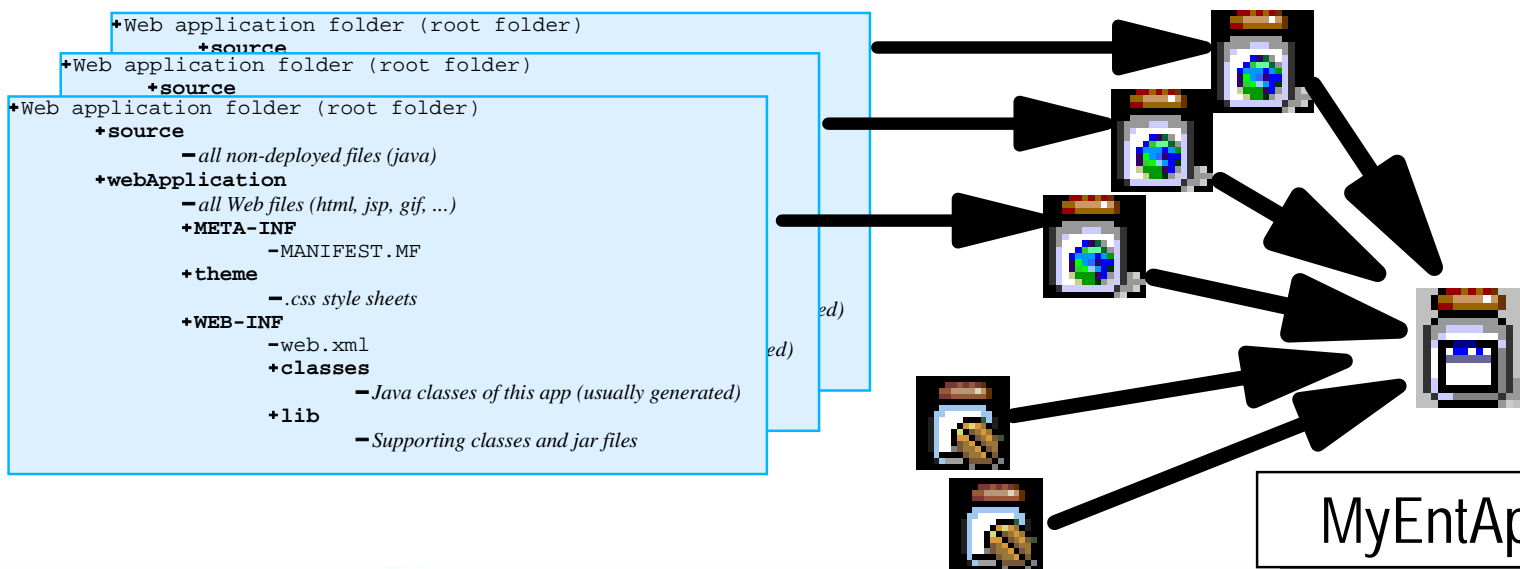
## ▶ Enterprise Archive Files (EAR)

- **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)





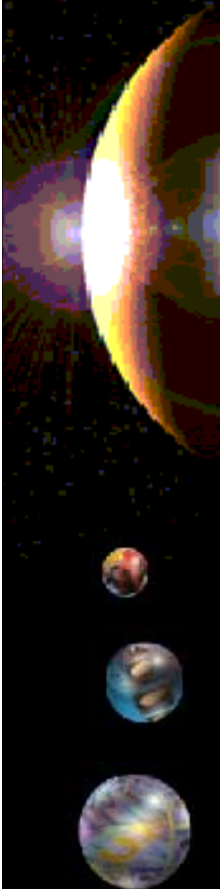


## ▶ WDS Sc Web Tools At A Glance:

- **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**



- ▶ **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**





# 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:

Use default location

Location:

Enterprise Application project name:

Context root:

Create CSS file

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

**Create a Web Project**

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

Project name:

Enterprise Application project name:

Available dependent JARs:

JAR/Module	Project

Manifest Class-Path:

Identify runtime dependent jar/war files

Optionally creates a CSS style for whole Web app

# Web Projects



**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

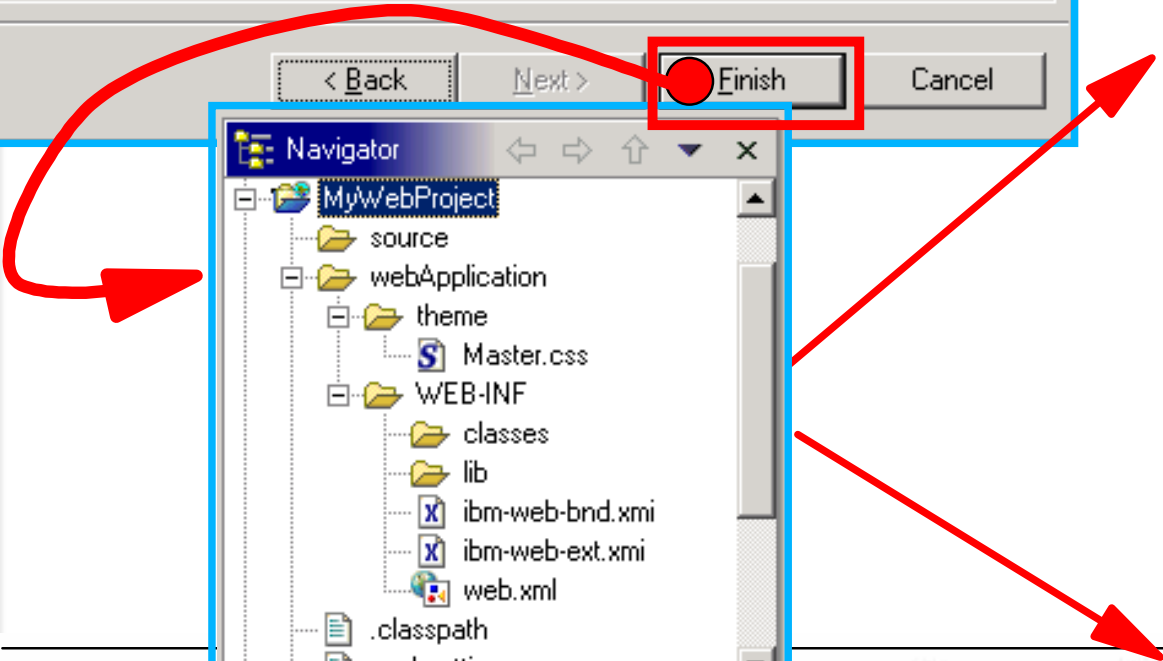
- ▶ for deployed files
- ▶ .jsp and .html files go here

.../theme

- ▶ for style sheets

../WEB-INF

- ▶ run-time dependencies
- ▶ jar files go here





## ▶ WDSsc 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 Style Sheets (CSS)

### ▶ Import/Export from/to a variety of sources

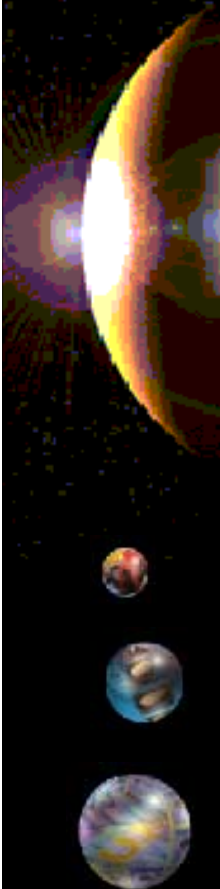
- ✓ HTTP/FTP/WAR

### ▶ 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



# Web tooling Components

Web	Java	RPG
		CBL

iSeries AD, IBM Toronto







# WebTooling Perspective!



The screenshot displays the Eclipse IDE in the WebTooling Perspective. The main window is titled "Web - Development Studio Client" and contains a menu bar (File, Edit, Perspective, Project, Toolbar, Insert, JSP, Format, Table, Frame, Page, Tools, Window, Help) and a toolbar with various icons. The central workspace is divided into several panels:

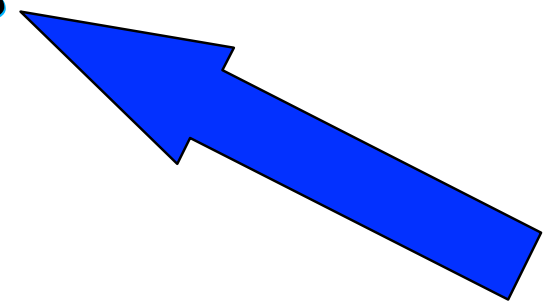
- Navigator:** Shows a project tree with folders like "source", "webApplication", and "WEB-INF". A blue arrow points to the "WEB-INF" folder, with the text "File View" written next to it.
- Gallery:** Lists various web components like Image, Wallpaper, Webart, Sound, Style Sheet, and Script.
- Main Design View:** Displays a web page design for "MyFirstJSP.jsp". The page has a blue background and features the text "Customer File" in a large, orange, 3D-style font. Below it is a form with the label "Enter Customer Number:" and a text input field. At the bottom of the form is an "ok" button. A blue arrow points to this design view, with the text "Different Views" written next to it.
- Tasks Panel:** Located at the bottom, it shows a list of tasks with columns for "Description", "Resources", and "In Folder".

At the bottom of the IDE, the address bar shows the path: "MyFirstWebProject/webApplication/MyFirstJSP.jsp".



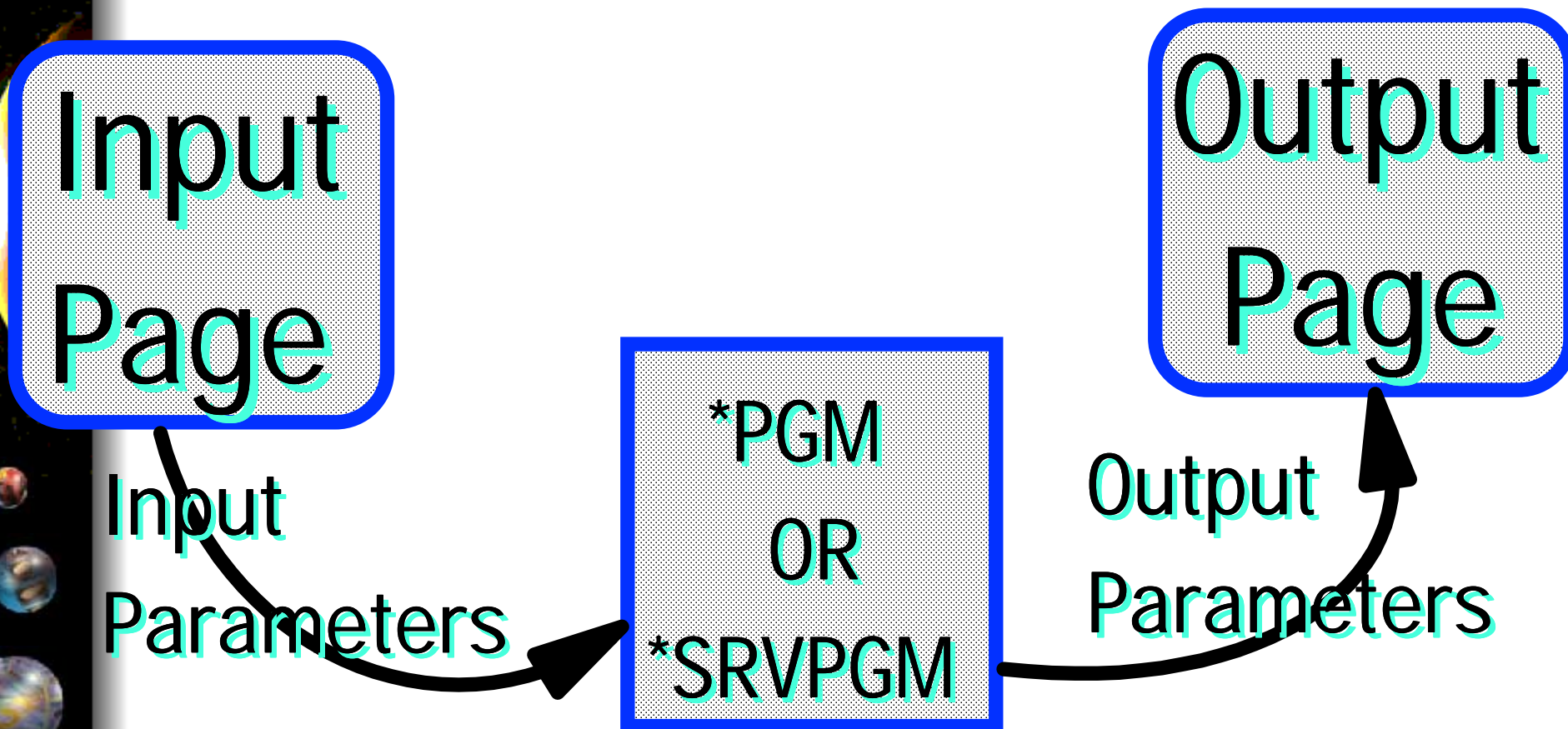


- What is in WDS Sc 5.0?
- e-business Primer
  - ▶ AD Model, traditional and web
- Web Application Primer
  - ▶ JSPs, Servlets, forms, etc
- Mixing Java and RPG
- What is Web Tool for iSeries
- Introducing WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Design time controls (DTCs)
  - ▶ Web Interaction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





## An interaction





# Create a Web project



iSeries AD, IBM Toronto

The screenshot shows the WebSphere Development Studio Client Advanced Edition for iSeries interface. The 'New Project' dialog box is open, displaying a project gallery with the following categories: EJB, iSeries, J2EE, Java, Plug-in Development, Server, Simple, Web (selected), WebFacing, and Examples. The 'Web' project type is highlighted in blue. The 'Next' button at the bottom of the dialog is highlighted in green. A green arrow points from the 'Next' button to the text 'Next page'.

Next page



# Create a Web Project



iSeries AD, IBM Toronto

Struts Based

Create a Web Project

Project name: MyFirstWebApplication

Use default

New project location: C:\Documents and Settings\farf\My Document

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 .cvsignore file
- Create a default CSS file
- Include Tag Libraries for accessing JSP ot
- Include Tag Libraries for database access
- Include Tag Libraries for internationalizati

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



# 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:

Enterprise application project name:

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:

Default Java package prefix:

Create a Resource Bundle for the Struts Project

Resource bundle

Java package:

Resource bundle name:

Done!

< Back   Next >   Finish   Cancel

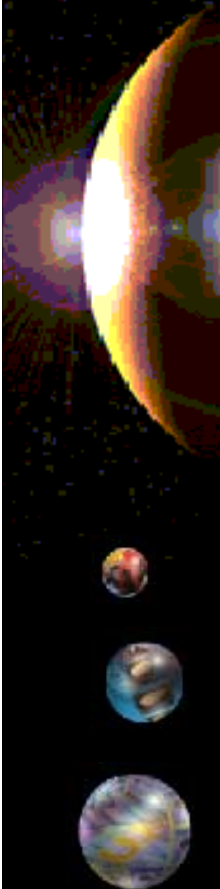


# Server Information



The screenshot shows the WebSphere Development Studio Client Advanced Edition for iSeries interface. A blue box with the text "iSeries RunTime configuration!" has two blue arrows pointing to the iSeries RunTime configuration icon in the top toolbar and the corresponding icon in the J2EE Navigator tree. The main window displays "Development St...s, Version 5.0" with a "Getting Started" section containing links to "Product Overview" and "online help". The console window at the bottom shows system output logs.

iSeries  
RunTime  
configuration!







## ▶ 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**

**iSeries Web Tools Run-time Configuration**

Specify the run-time configuration for program call of iSeries Web development tools

iSeries host name:

User ID:

Password:

Runtime library list

Farr	
Coulthard	
Weiss	

Display detailed runtime errors

Use program call JCA connector

Specify the JNDI name:

Tip: be sure to press Enter after typing library name!!

What is JCA connector?

Next page



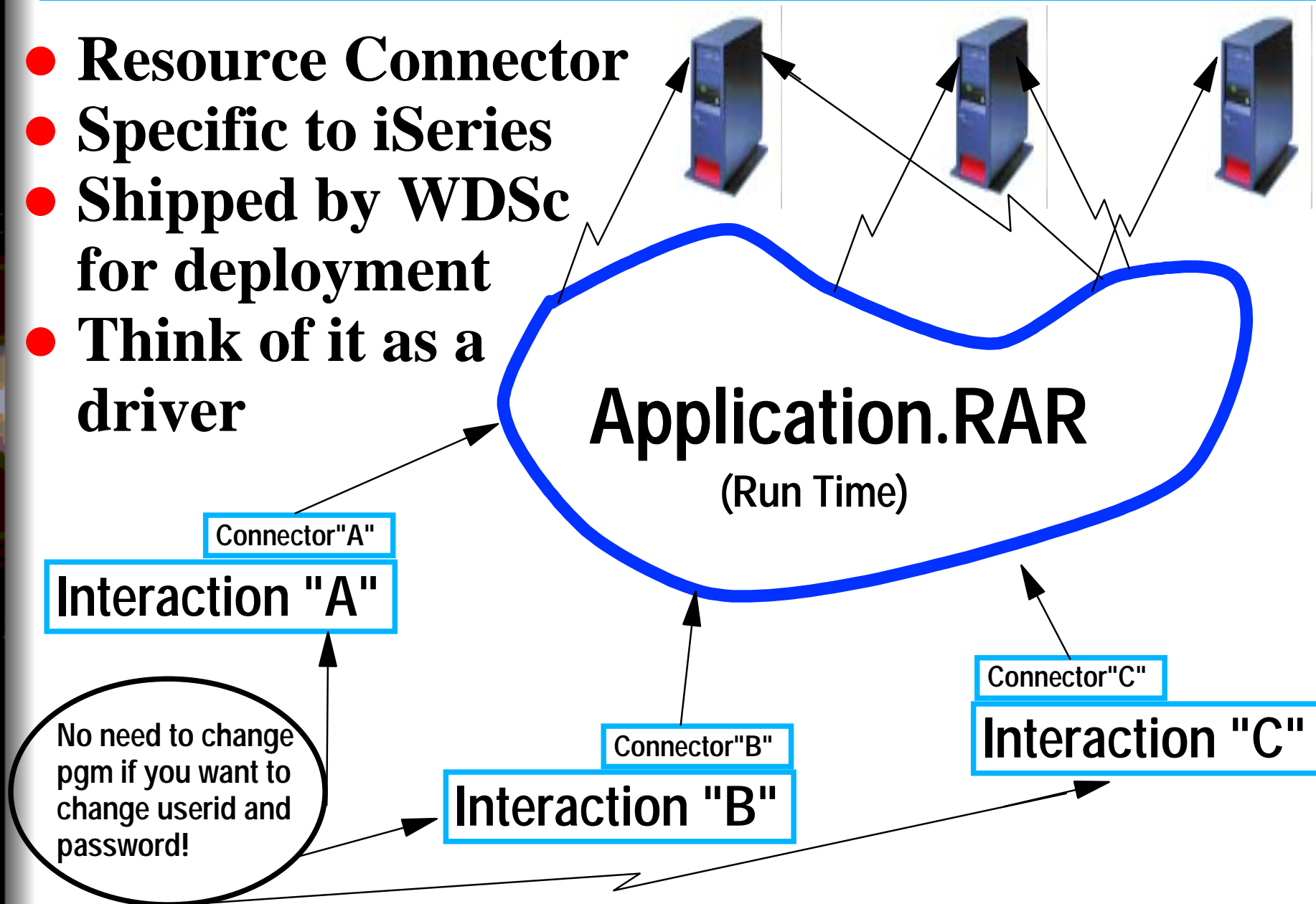


# What is JCA Connectors?



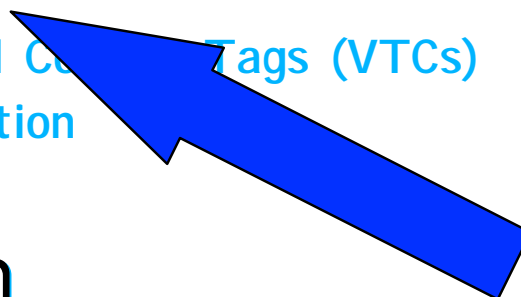
iSeries AD, IBM Toronto

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





- What is in WDS Sc 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 WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Visual C... Tags (VTCs)
  - ▶ Web Interaction
  - ▶ Publishing
- Conclusion

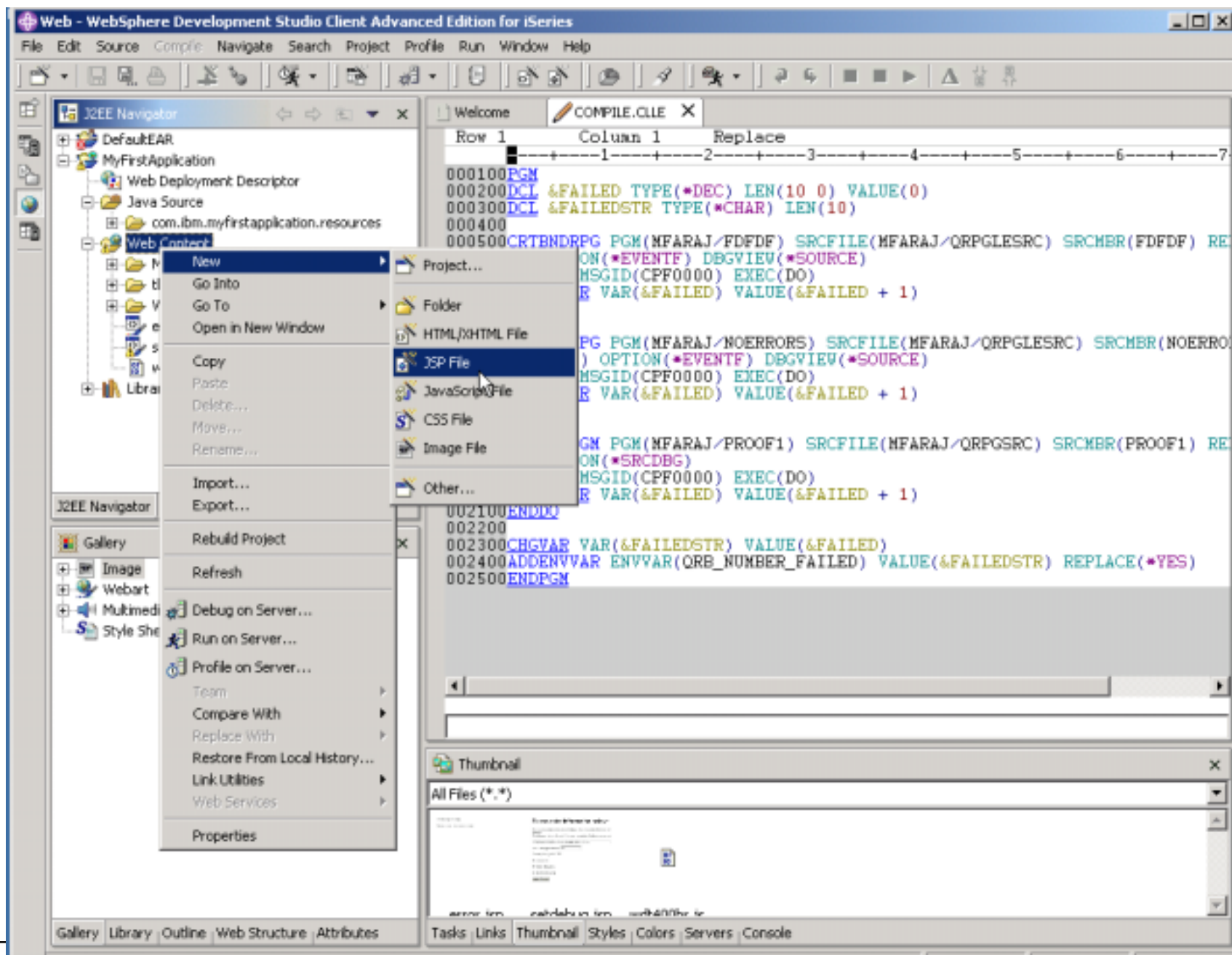




# Create your own pages . . .



iSeries AD, IBM Toronto



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

# First Screen



IBM Toronto

The screenshot shows the Web - Development Studio Client interface. The main window displays a web form titled "CustomerInfo.jsp - Customer Inquiry". The form has a blue background with the text "Customer File" in red. Below this, there is a text input field labeled "Enter Customer Number:" and an "OK." button. The interface includes a Navigator pane on the left showing a project structure with folders like "source" and "webApplication", and a Gallery pane below it with categories like "Image", "Webart", and "Style Sheet". A Tasks pane at the bottom shows a list of tasks with columns for "C", "!", "Description", "Resource", and "In Fold".

C	!	Description	Resource	In Fold
!		/servlet/com.ibm.webtools.as400.GenericDebu...	setdebug.jsp	MyFirs
!		JavaCompile The method getValue(java.lang.Str...	setdebug.jsp	MyFirs
!		com.ibm.webtools.as400.SFHandler- Broken Link.	web.xml	MyFirs

iSeries push  
button

iSeries Entry Field

# No More DTCs !!!

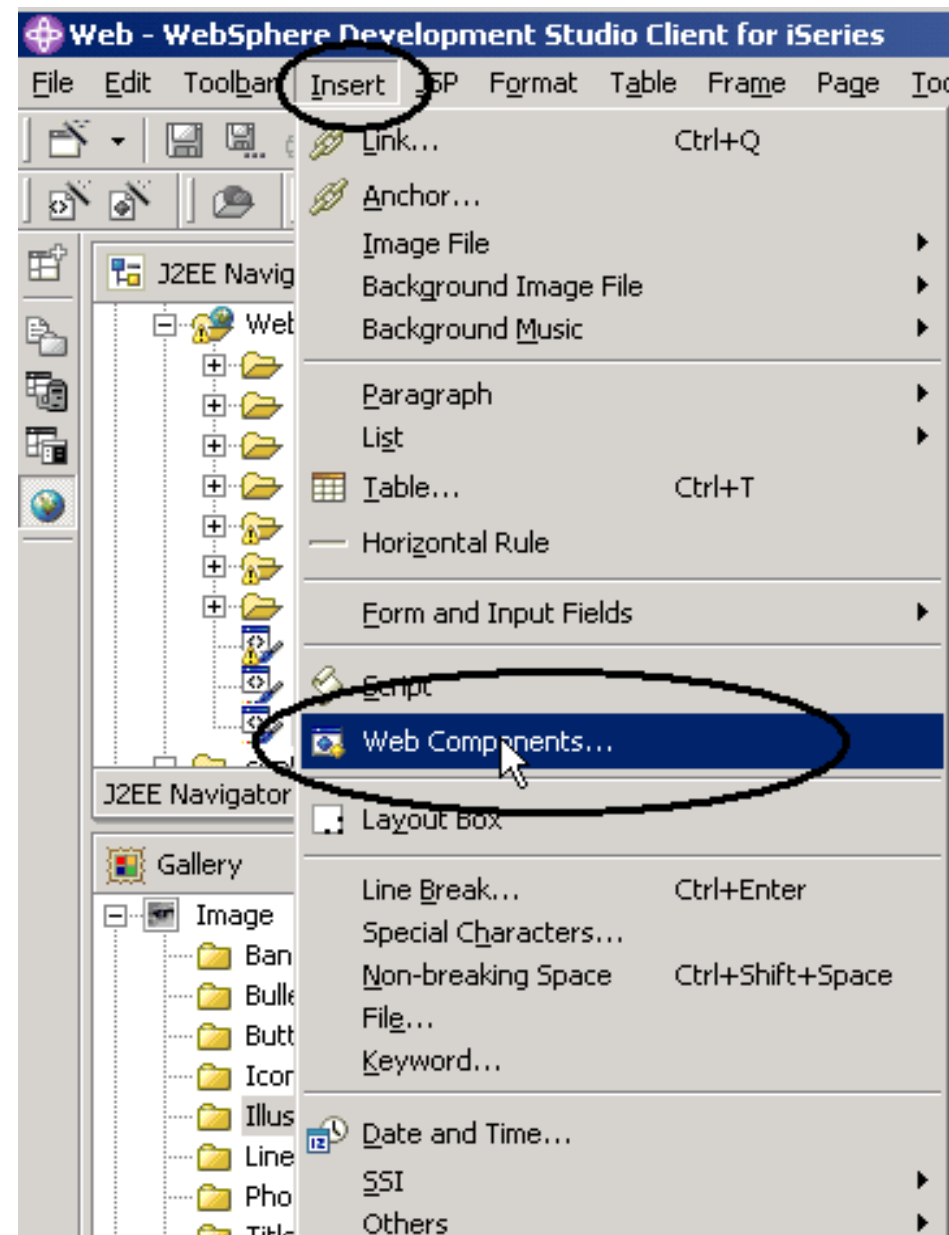
The screenshot displays the IBM Web Development Studio Client interface. A large red 'X' is drawn across the entire scene. In the foreground, the 'AS/400 Entryfield DTC Properties' dialog box is open, showing the 'General' tab. The 'Field name' is 'Entryfield400', 'Field type' is 'Normal', 'Visible width' is '20', and 'Alignment' is 'left'. Below, 'Field data' includes a 'DB Reference ...' button, 'Length' (0), 'Data type' (CHARACTER), and 'Formatting' (None). The 'Directionality' is set to 'NONE'. In the background, the 'Design-time Control' dialog box is open, listing various DTCs: AS/400 Checkbox DTC (checked), AS/400 Combobox DTC, AS/400 Entryfield DTC, AS/400 Label DTC, AS/400 Listbox DTC, AS/400 MLE DTC, AS/400 Pushbutton DTC, AS/400 Radiobutton DTC, AS/400 Subfile DTC, and AS/400 Table DTC. The file path 'E:\WDSC\WSSD\p...ins\com.ibm.etools.iserie...\CheckBox400.ocx' is visible. The main IDE window shows a project structure with 'lancelInput.jsp' and 'CustomerBalances.wlet'.



# VTCs ... No more DTCs

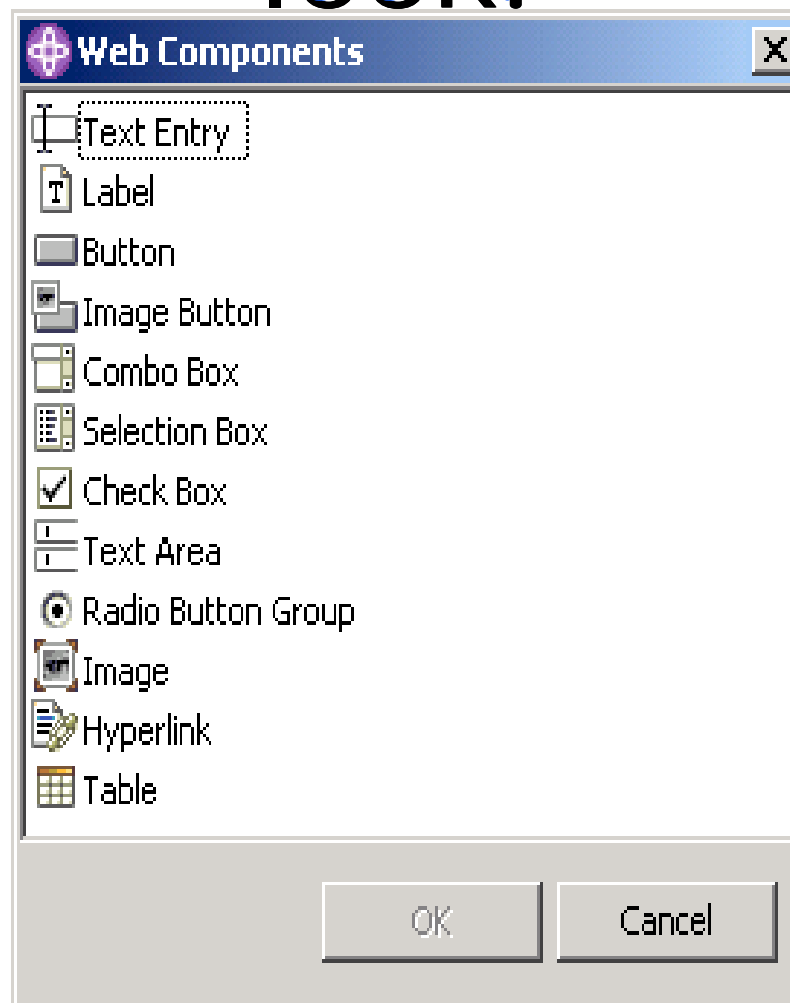


- **VTC: Visual Control Tags**
  - ▶ Emerging technology
  - ▶ Blessed by IBM and SUN
- **Tag library defined by:**
  - ▶ [www.sun.com](http://www.sun.com)
- **Extendable**
- **DTCs: Old fashion**
  - ▶ ActiveX technology



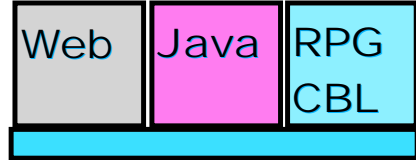
# VTCs ... No more DTCs

## Same selection ... Better look!

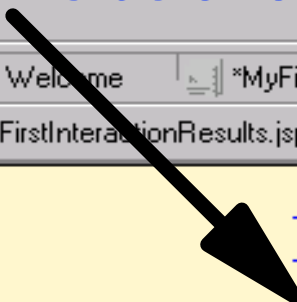




# Customer Detail Screen



Screen generated by WebTools



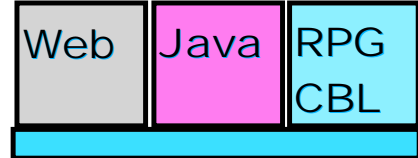
A screenshot of the 'Web - Development Studio Client' IDE. The main window shows a JSP page in design view titled 'MyFirstInteractionResults.jsp - Result Form'. The page content consists of a vertical list of labels: 'custno: Label400', 'msg: Label400', 'CUSTNO: Label400', 'CUSTNA: Label400', 'REPNO: Label400', 'CONTAC: Label400', 'CPHONE: Label400', and 'CFAX: Label400'. The IDE interface includes a menu bar (File, Edit, Perspective, Project, Toolbar, Insert, JSP, Format, Table, Frame, Page, Tools, Window, Help), a toolbar, a Navigator pane on the left showing a project tree, and a bottom status bar with tabs for Design, Source, and Preview. A 'Tasks' pane is also visible at the bottom.

# 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
```

# Files Used



iSeries AD, IBM Toronto

## A\* Logical file description

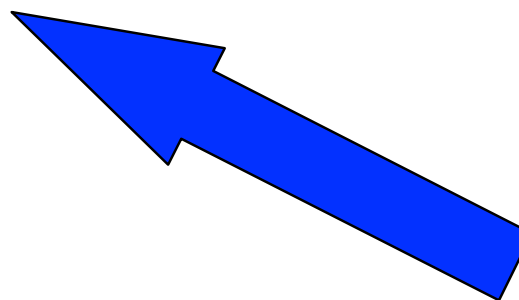
```
A          R CUSTOM01          PFILE(CUSTOMER)
A          K CUSTNO
```

## 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                                     VALUES('1' '2' '3')
A          K CUSTNO
```



- What is in WDS Sc 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 WDS Sc for iSeries
  - ▶ RPG Example
  - ▶ iSeries Visual Control Tags (VTCs)
  - ▶ Web Interaction
  - ▶ Publishing
- Conclusion





## ▶ 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

# Interaction Wizard



**Web Interaction**

Specify a Name and Location for your Web Interaction

Destination folder

Web Interaction name

Java package

Use error page

Invalidate session after the interaction occurs

Just a name!!!  
Like a field name



## ▶ 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:

Use error page

Invalidate session after the interaction occurs

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

**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:

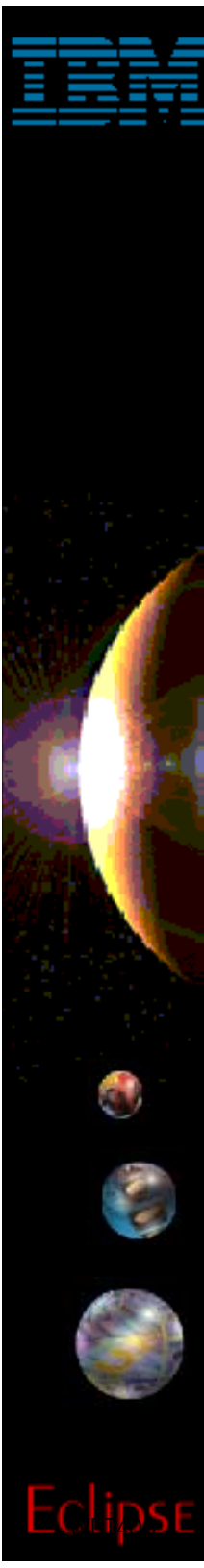
Mode 1: Generate input and output Web pages



# Or ... Create your own ...



The screenshot shows the Eclipse IDE's Web Development Studio Client. The title bar reads "Web - Development Studio Client". The menu bar includes "File", "Edit", "Perspective", "Project", "Toolbar", "Insert", "JSP", "Format", "Table", "Frame", "Page", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, editing, and development. A tooltip "Create or modify a web interaction" is visible over a specific toolbar icon. The Navigator on the left shows a project structure for "MyFirstWebProject" with folders "source" and "webApplication". The "webApplication" folder contains a "WEB-INF" folder and several files, including "MyFirstJSP.jsp". The main editor displays a web page design for "MyFirstJSP.jsp" with a blue background and the text "Customer File" in large, orange, 3D-style letters. Below this, there is a form with the label "Enter Customer Number:" and a text input field. An "ok" button is positioned below the input field. The editor has tabs for "Design", "Source", and "Preview", with "Design" selected. At the bottom, there is a "Tasks" panel showing 6 items and a "Properties" panel with tabs for "Tasks", "Colors", "Properties", "Styles", "Thumbnail", and "Links". The status bar at the bottom indicates the current file path: "MyFirstWebProject/webApplication/test.jsp".



# iSeries Web Interaction Wiz



iSeries AD, IBM Toronto

▶ 1a. Press "Add Program",  
▶ 1b. Enter program information  
▶ 1c. Press Ok

▶ For \*SRVPGM, enter procedure name



### Web Interaction

Specify the Input and Output Parameters for your iSeries Host Program

Use an iSeries ILE program     
  Use a Java bean

Program call definitions

```

getname
├── number
└── name
                    
```

Add Program... Add Parameter... Add Structure...

**Edit Parameter**

Parameter name:

Data type:

Structure name:

Length:

Precision:

Count:

Usage:

Initial value:

Advanced...

Specify database reference field

Show database field definition

Describe parameters

▶ 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

Reference DB fields!

▶ For arrays

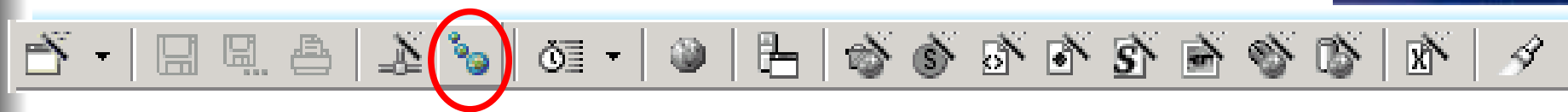
▶ Input: read by program

▶ Output: updated by program

▶ Input/Output: both



# iSeries Web Interaction Wiz



**Web Interaction Wizard**

**Design the Input Form**

Input parameters:

- carClass
- carMake
- carColor

Move up  
Move down  
All  
None

**Input Form**

carClass:

carMake:

carColor:

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

Page Fields

< Back   Next >   Finish   Cancel

Select input parameters to prompt user for, in input page

Tailor attributes of generated prompt per parameter

Table with 2 columns: Property, Value. Contains details for the carClass parameter.

Immediately see results of tailoring parameter and page attributes

Tailor attributes of overall input page



# iSeries Web Interaction Wiz



iSeries AD, IBM Toronto

**Web Interaction**

**Design the Input Form**

Input parameters:

- number

Move up  
Move down  
All  
None

**Input Form**

number:

Property	Value
Page Title	Input Form
Background Color	RGB {128, 255, 128}
Title Color	RGB {0, 0, 160}
Field Color	■ (0,64,128) ...

Page Fields

< Back   Next >   Finish   Cancel

Generated files

Use "Run on Server" to test

Use Page Designer to finesse generated pages





# iSeries Web Interaction Wiz



iSeries AD, IBM Toronto

**Web Interaction**

**Design the Result Form**

Output parameters:

name

Move up  
Move down  
All  
None

**Result Form**

name:

Property	Value
Page Title	Result Form
Background Color	RGB {128, 0, 255}
Title Color	■ (0,0,0)
Field Color	

Page Fields

< Back   Next >   Finish   Cancel

Select output parameters to display in output page

Tailor attributes of generated prompt per parameter

Immediately see results of tailoring parameter and page attributes

Tailor attributes of overall output page



# You are done! What's Next?

iSeries AD, IBM Toronto

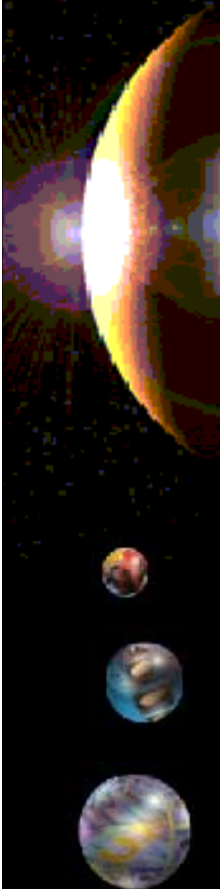
- What did you do so far?

- ▶ 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 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!





- ▶ **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

# Run it!



The screenshot shows the WebSphere Development Studio Client Advanced Edition for iSeries interface. The main window displays a JSP file named 'MyFirstInteractionInput.jsp' with the following code:

```

Row 1      Column 1      Replace
... ..FFilename++IPEASFRlen+LKlen+AIDevice+. Keyw
000100    FCHKENVFM  CF  E          Workstn
000200    *
000300    DQQFENV          PR          10I 0  Extp
000400    *
000500    Drc          S          10I 0
000600    *
000700    C          Eval          rc = QQFENV
*
C          Dow          NOT *IN03
*
C          If          rc = 1
C          Eval          *in01 = *on
*
C          Else
C          Eval          *IN01 = *off
C          EndIf
*
C          Exfmt          FMT01
C          EndDo
*
C          Eval          *inlr = *on
    
```

A context menu is open over the file 'MyFirstInteractionInput.jsp' in the J2EE Navigator, with the 'Run on Server...' option selected. The console window at the bottom shows the following output:

```

WebSphere v5.0 Test Environment (WebSphere v5.0)
:21:18:466 EST] 730b8488 SystemOut 0 ]
:21:21:000 EST] 730b8488 SystemOut 0 WI
:21:21:190 EST] 730b8488 SystemOut 0 F
    
```

# First Page



The screenshot displays the WebSphere Development Studio Client interface. The main window shows a web browser with the URL `http://localhost:9080/MyFirstApplication/Myf` and a page titled "Input Form". The form contains a text input field with the value "0010100" and two buttons labeled "Submit" and "Reset".

The J2EE Navigator on the left shows a project structure with files like `struts-template.tld`, `web.xml`, `error.jsp`, `getnameInput.jsp`, `getnameResults.jsp`, `Input`, `MyFirstApplication.pcm1`, `MyFirstInteractionInput.jsp`, `MyFirstInteractionResults.jsp`, and `output.jsp`.

The Console window at the bottom shows the following output:

```

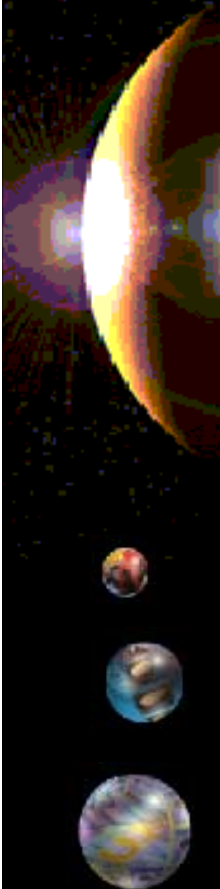
SystemOut      0 {name=[Ljava.lang.String:@2f9c848e}
SystemOut      0 ]
SystemOut      0 WDT400:QZRCRVS /QUSER /068458:Po
WebGroup       I SRVE0180I: [MyFirstApplication] [/MyFi
WebGroup       I SRVE0180I: [MyFirstApplication] [/MyFi
    
```

Make sure that server is ready for business!

# Output Page



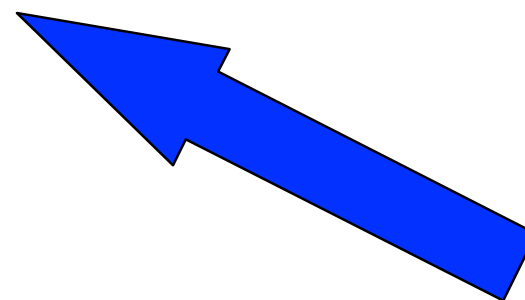
The screenshot shows the WebSphere Development Studio Client interface. The main window displays a web browser at the URL `http://localhost:9080/MyFirstApplication/Myf`. The browser content shows a "Result Form" with the text "name: Meridien Electronics Ltd". The IDE interface includes a J2EE Navigator on the left showing a project structure with files like `struts-template.tld`, `web.xml`, `error.jsp`, `getNameInput.jsp`, `getNameResults.jsp`, `Input`, `MyFirstApplication.pcm1`, `MyFirstInteractionInput.jsp`, `MyFirstInteractionResults.jsp`, and `output.jsp`. Below the navigator is an "Attributes" panel with the message "No attributes are available." At the bottom, a "Console" window shows system output logs with timestamps and process IDs.

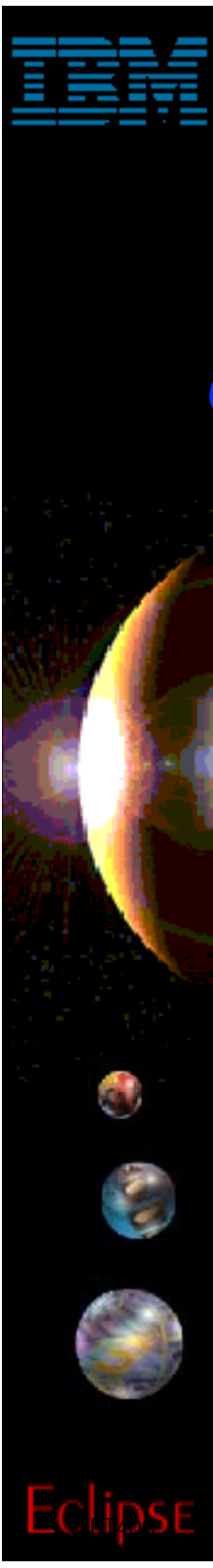






- Web Interaction - Struts based!





# Web Wizard Struts based ...



iSeries AD, IBM Toronto

Based on 1.02 and not 1.1

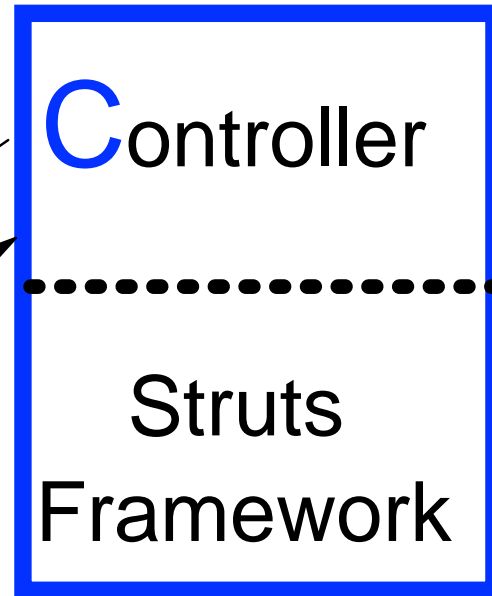
## M.V.C

View

Model



JSPs & HTML



Logic!



Traffic Controller



# Start Web Diagram ...

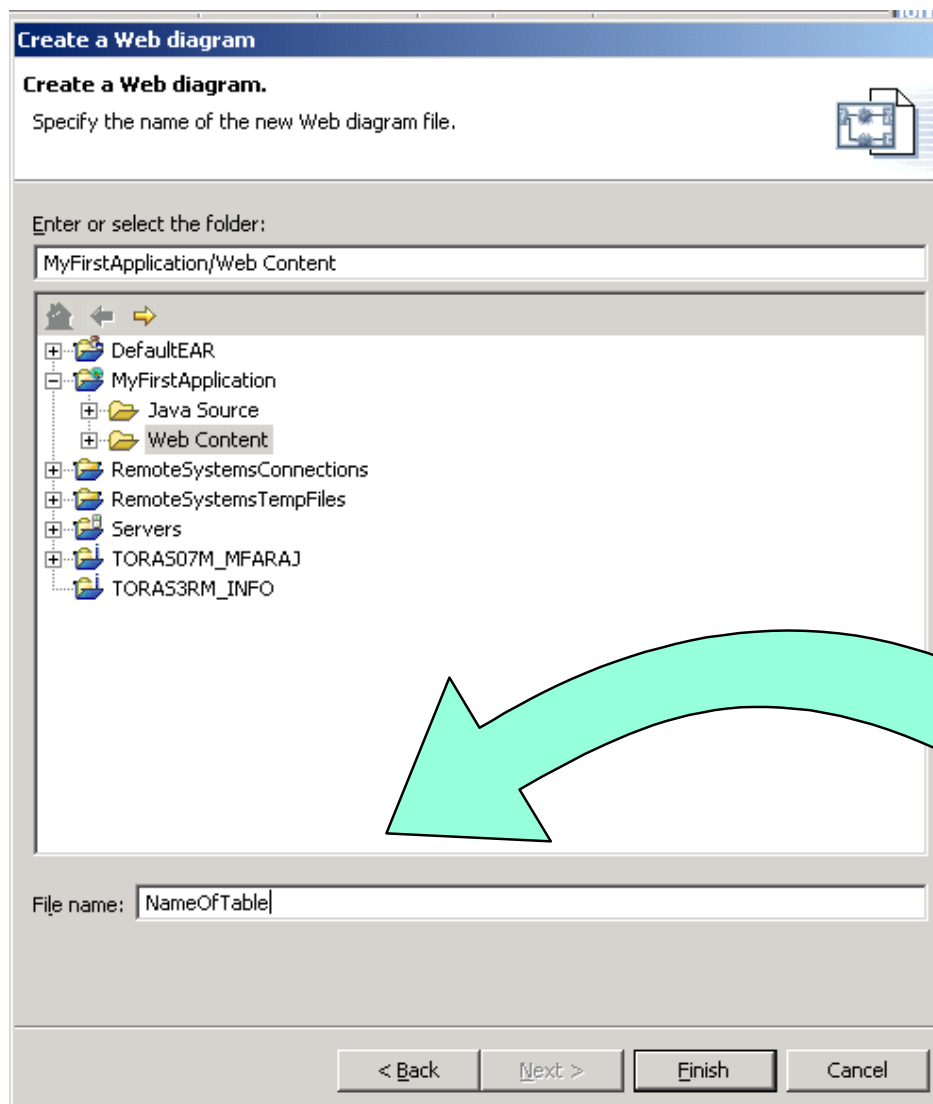


iSeries AD, IBM Toronto

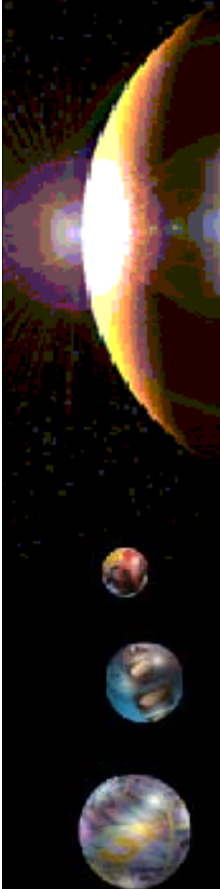
The screenshot shows the WebSphere Development Studio Client interface. The J2EE Navigator on the left displays a project named 'MyFirstApplication' with a 'Web Context' folder selected. A context menu is open over the 'Web Context' folder, with the 'New' option selected. The 'New' submenu is also open, showing various file types. The 'Other...' option is highlighted, and a green arrow points from it to the 'New' dialog box on the right.

The 'New' dialog box is open, showing a list of templates. The 'Web Diagram' template is selected. The dialog box has a 'Select' section with the text 'Create a new Web diagram'. The list of templates includes: Component Test, Data, EJB, Series, Java, Plug-in Development, Remote File Transfer, Server, Simple, Symptom Database, Web, Struts, WebFacing, Web Services, XML, and Examples. The 'Web Diagram' template is highlighted in blue. At the bottom of the dialog box, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

# Diagram Definition



Give the diagram a name!





# Draw the diagram!

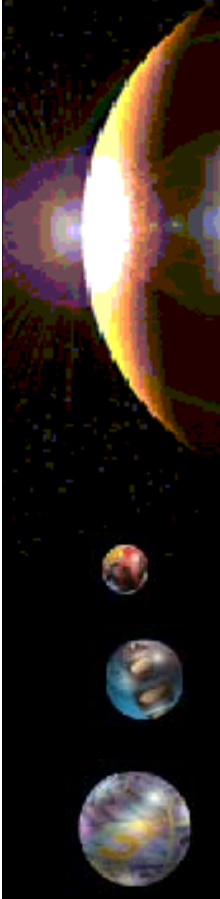


IBM Toronto

The screenshot shows the WebSphere Development Studio Client interface. On the left, the J2EE Navigator displays a project structure with files like `getnameInput.jsp`, `MyFirstApplication.pcm1`, and `NameOfTable.gph`. The main editor area is currently blank. A context menu is open over the editor, showing options: **Draw**, **New** (highlighted), **Undo**, **Redo**, and **Delete**. The **New** submenu is expanded, listing: **Action Mapping Node**, **Form Bean Node**, **Java Bean Node**, **Web Page Node**, and **Web Application Node**. Below the editor, the console window shows system output logs with timestamps and IP addresses.

Pgm or Service pgm use Action

For input and output page use Web page node







# Web Diagram Editor



iSeries AD, IBM Toronto

The screenshot displays the WebSphere Development Studio Client Advanced Edition for iSeries interface. The main window shows a web diagram with a central gear icon labeled "getname". A blue box labeled "\*PGM or \*SRVPGM" has an arrow pointing to the gear. Below the gear are two boxes: "MyFirstInteractionIn..." on the left and "MyFirstInteractionRe..." on the right. A blue box labeled "Input page" has an arrow pointing to the left box, and a blue box labeled "Output page" has an arrow pointing to the right box. The J2EE Navigator on the left shows a project structure with files like error.jsp, getnameInput.jsp, and MyFirstApplication.pcm1. The Console window at the bottom shows system output logs with timestamps and messages like "WDT400:QZRCRSRVS /QUSER".





# Invoke Interaction Wizard!



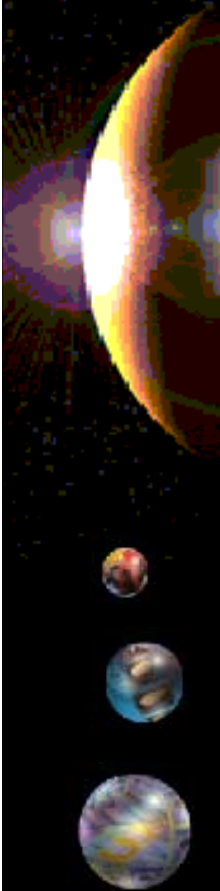
iSeries AD, IBM Toronto

The screenshot shows the WebSphere Development Studio Client Advanced Edition for iSeries. The J2EE Navigator on the left lists files like error.jsp, getnameInput.jsp, and MyFirstInteraction.wit. The main workspace shows a diagram with a component named 'MyFirstInteractionIn...' and a green gear icon labeled 'getnan'. A context menu is open over the gear icon, with the option 'Open iSeries web interaction wizard...' circled in black. A blue callout box points to this menu item with the text 'right mouse click'. Another blue callout box points to the gear icon with the text 'This invokes the same interaction wizard we talked about'. The console at the bottom shows system output logs.





- ▶ 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



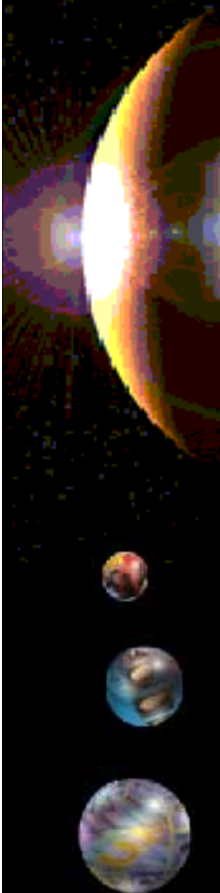


# Standards



iSeries AD, IBM Toronto

## ▶ Industry Standards



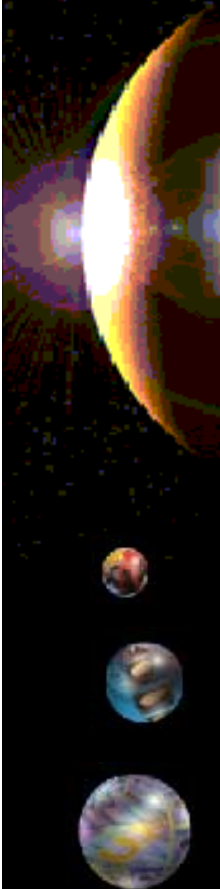
Eclipse



## ▶ Industry Standards Supported

### ● By WSSDa, WSAD and WDSa:

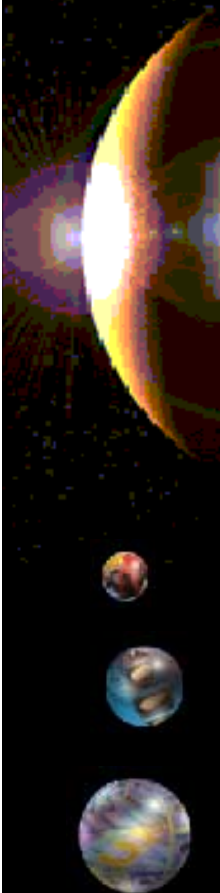
- ▶ EJB 1.1
- ▶ Servlet 2.2
- ▶ JSP 1.1
- ▶ JRE 1.3
- ▶ Web Services Definition Language (WSDL) 1.1
- ▶ Apache SOAP 2.1
- ▶ XML DTD 1.0 10/2000 Revision
- ▶ XML Namespaces 1/99 Version
- ▶ XML Schema 5/2001 Version
- ▶ HTML 4.01 (other levels should work)
- ▶ CSS2 (PageDesigner displays a subset)





## ▶ Information Sources

- **[www.ibm.com/software/ad/wds400](http://www.ibm.com/software/ad/wds400)**
  - ▶ As of June 4th
- **[www.eclipse.org](http://www.eclipse.org)**
  - ▶ Eclipse and information about eclipse
- **[www.ignite400.org](http://www.ignite400.org)**
  - ▶ Introduction to eclipse article
- **[www.ibm.com/software/info1/websphere/partners/iseriess.jsp](http://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](http://www.ibm.com/websphere/developer)**
  - ▶ WebSphere Developer Domain
  - ▶ Many articles and tutorials on technology and tools, including eclipse and WSWB and WebSphere Studio configurations





# 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.