



IBM Software Group

# WebSphere Development Studio Client for iSeries


## Web Tools for iSeries developers

| September, 2005 | WebSphere Development Studio Client V6.0

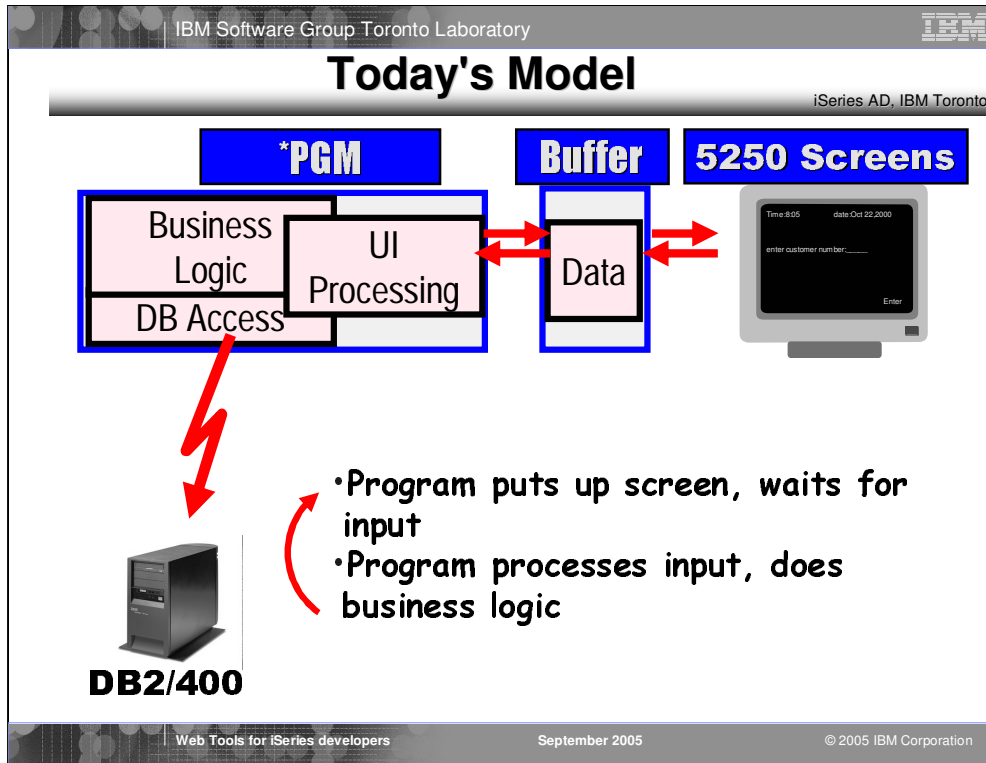
© 2005 IBM Corporation

## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
    - AD Model, traditional
    - e-Business Application, Web Model
  - Introducing Web Tools for iSeries
    - J2EE Enterprise Application
    - Web Tools At A Glance
    - Dynamic Web Project
    - Web Interaction wizards
    - Runtime Configuration wizard
    - iSeries Web Components
  - Demo: Using iSeries Web Tools
  - Advanced Topics
    - Mixing Java with RPG
    - Struts
- 

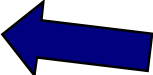
Next couple slides compare how many iSeries ILE applications look today vs. what the typical web application looks like

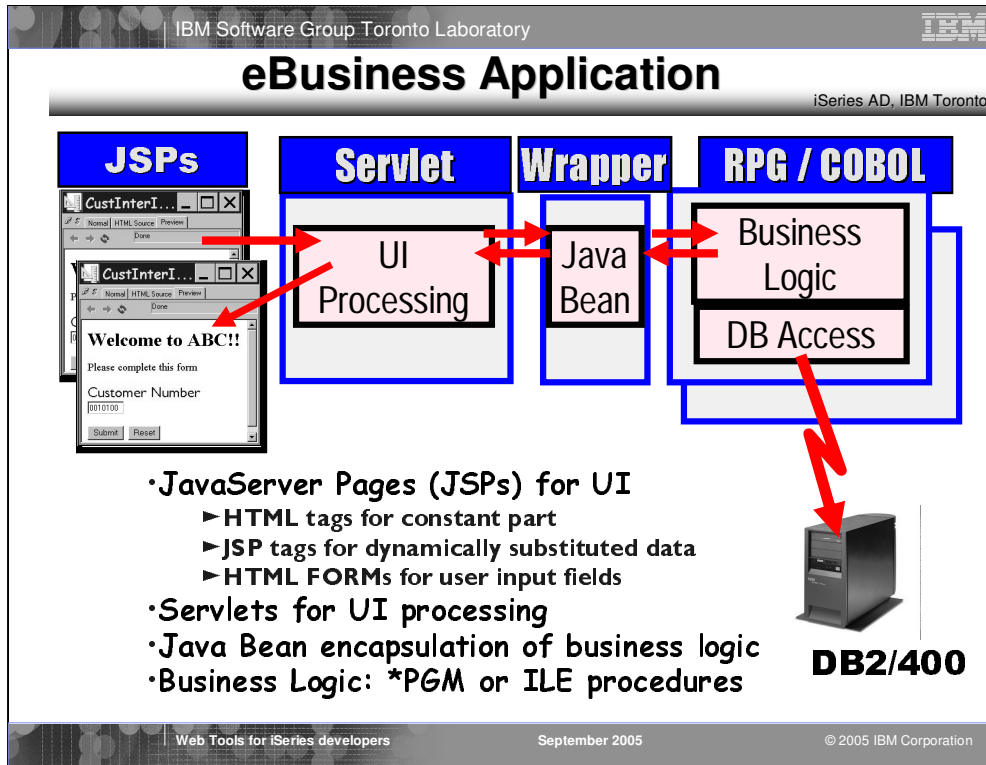


The iSeries ILE program (e.g. RPG \*PGM), performs the business logic, accesses program data (File I/O, or DB2 installed on iSeries for example). PGM does UI processing, outputs UI information to a buffer. 5250 reads the buffer to display UI.

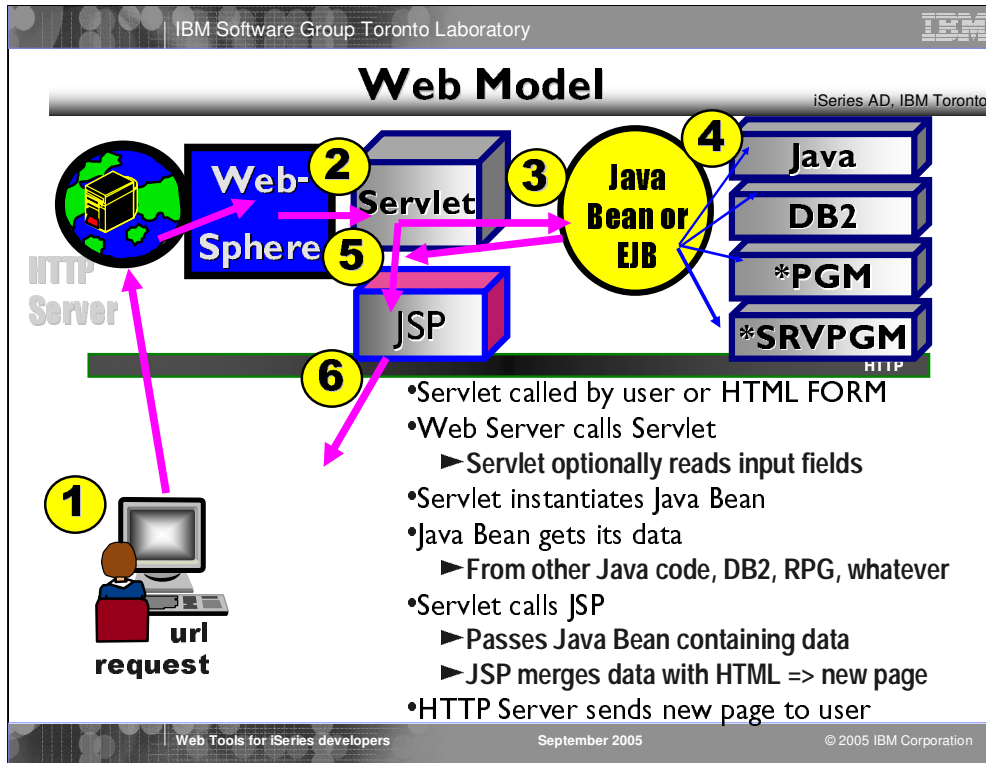
## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model 
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts

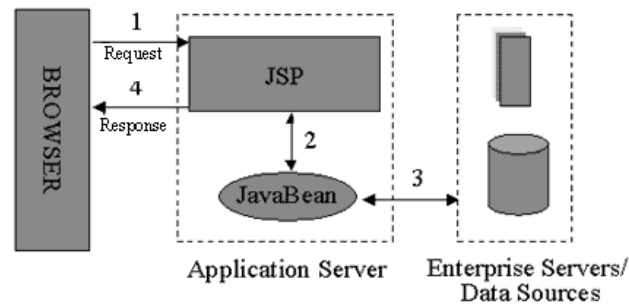


Typical e-Business application (JSP Model 2 architecture).



Above is typical Model 2 JSP architecture

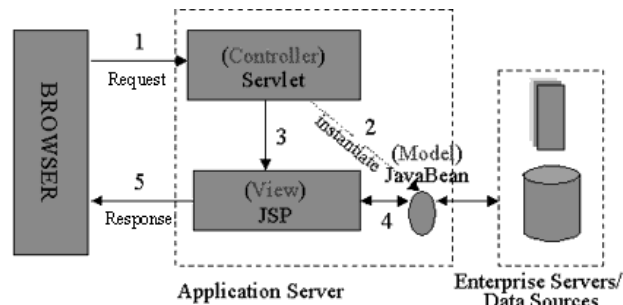
## JSP Model 1 architecture



Two approaches for building web applications using JSP technology- Model 1 & Model 2. The goal of a well designed web application (or any application) is to separate data presentation from content (or the model (business logic & data) from the view).

-M1 & M2 differ essentially in where the bulk of the processing is performed. In Model 1 the JSP alone is responsible for taking incoming requests and replying back to client. Results in lots of scriptlets and Java Code in the JSP.

## JSP Model 2 architecture



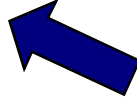
It takes advantage of the predominant strengths of both technologies, using JSP to generate the presentation layer and servlets to perform process-intensive tasks (controller). Here, the servlet acts as the *controller* and is in charge of the request processing and the creation of any beans or objects used by the JSP, as well as deciding, depending on the user's actions, which JSP page to forward the request to. Note particularly that there is no processing logic within the JSP page itself; it is simply responsible for retrieving any objects or beans that may have been previously created by the servlet, and extracting the dynamic content from that servlet for insertion within static templates. This approach typically results in the cleanest separation of presentation from content.



# AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts



## Terms: Web Application

iSeries AD, IBM Toronto

### ► Web App folder structure:

```
+Web application folder (root folder)
+JavaSource
  -all non-deployed files (java)
+WebContent
  -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
```

collectively known as "Web Resources"

maps dependent jar files in other Web apps

Web application deployment descriptor:

► identifies servlets, security, env vars, mime types, key pages, external references and session configuration info

J2EE  
1.4

## Terms: WAR Files

iSeries AD, IBM Toronto

### ► Web Archive Files (WAR)

**J2EE  
1.4**

- **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)
+JavaSource
  -all non-deployed files (java)
+WebContent
  -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

If you want to distribute and/or deploy a Web application, you package it as a WAR file (similar to a .jar file used to distribute java class libraries). J2EE specification defines what the WAR can contain → top-level WEB-INF folder contains web.xml and lib, class folders.

## Terms: EAR Files

iSeries AD, IBM Toronto

### ► Enterprise Archive Files (EAR)

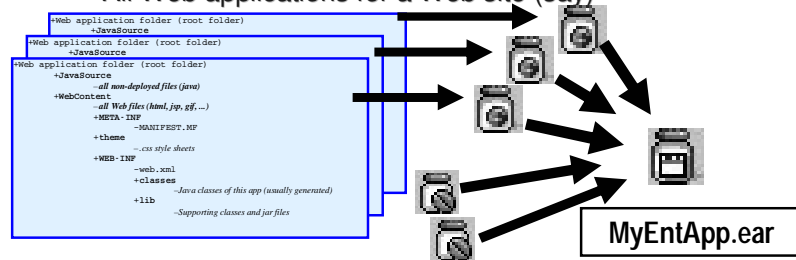
J2EE  
1.4

- One file containing:

- ▶ Zero or more Web Archive (war) files
- ▶ Zero or more EJB jar files
- ▶ Zero or more JCA resource adapter modules (rar) 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)



Publish the EAR to the app server

# AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance 
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts



# WDS Sc Web Tools At A Glance

iSeries AD, IBM Toronto

- ▶ **Web projects**
  - ✓ Dynamic Web project - 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 with content assist, and debugging
  - ✓ Including WYSIWYG PageDesigner
  - ✓ Custom JSP tags (taglib) support
  - ✓ Images and animation
  - ✓ Cascading Style Sheets (CSS)
  - ✓ Web Diagram Editor
    - ✓ To visualize and change the flow of a Struts-based application
- ▶ **WAR file import, export, and validation**
- ▶ **Integration with WebSphere Test Environment**

# Web Perspective with Page Designer

iSeries AD, IBM Toronto

**Project Explorer**

- oorder.jsp
- orderaccepted.jsp
- scripts
- theme
- WEB-INF
- changePassword.jsp
- error.jsp
- index.jsp
- login.jsp
- messages.jsp
- OrderEntryExample
- timeout.jsp
- wdk400br.js
- acceptorder.wit
- addItem.wit
- cancelorder.wit
- changeorder.wit

**Palette**

- HTML Tags
- Form Tags
- JSP Tags
- Struts HTML Tags
- iSeries Web Compon...
- Form
- Simple Table
- Button
- Check Box
- Combo Box
- Hyperlink
- Image
- Image Button
- Label
- Radio Button Group
- Selection Box
- Table
- Text Area
- Text Entry
- Struts Bean Tags
- Struts Logic Tags
- Page Template
- Data

**Page Designer**

index.jsp - Order Entry Example - Welcome

## WDCS Order Entry Example

Welcome

WebSphere Development Studio Client for iSeries, Order Entry sample. This sample is a Web entry application.

[Click here to start the sample.](#)

**Properties View**


Type:	Heading 1
Paragraph:	Alignment: (Auto)

**Outline View**

- jsp:directive.taglib
- jsp:directive.taglib
- tpl:insert
- DOCTYPE:HTML
- html
- head
- body
- table
- tbody
- tr
- tr
- tr
- td

## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project 
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts



IBM Software Group Toronto Laboratory

## Dynamic Web projects

iSeries AD, IBM Toronto

**File->New->Dynamic Web Project**

- Can contain dynamic J2EE resources such as servlets, JSP files, HTML files, images
- Also creates an Enterprise Application (EAR) project if it does not already exist

■ Specify project name  
■ Servlet 2.4 = J2EE 1.4  
■ Select Target Server  
■ Automatically associates with a supplied EAR file for easy auto config of Application Server.

Click Next

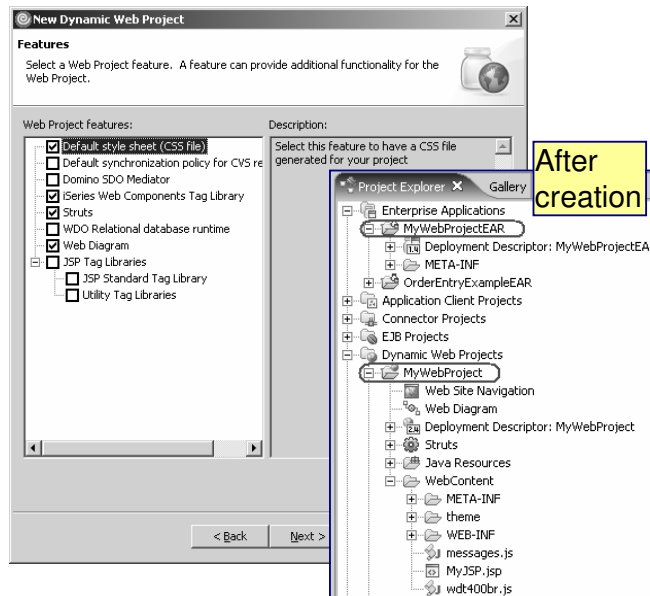
Web Tools for iSeries developers September 2005 © 2005 IBM Corporation

- **Special type of project**
- **Created with J2EE folder layout**
- **Created with simple web.xml file**
  - ▶ Automatically updated as resources are created
  - ▶ Has specialized web.xml editor

# Dynamic Web projects

iSeries AD, IBM Toronto

- Optionally creates a CSS style for whole Web app
- Add iSeries Web Components Tag Library
- Add Struts feature
- Add Web Diagram feature



## ► In Dynamic Web Project:

### ► JavaSource

- for Java classes and resources

### ► WebContent

- for deployed files
- .jspx and .html files go here

### ► ../theme

- for style sheets

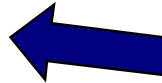
### ► ../WEB-INF

- run-time dependencies
- jar files go here

## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts





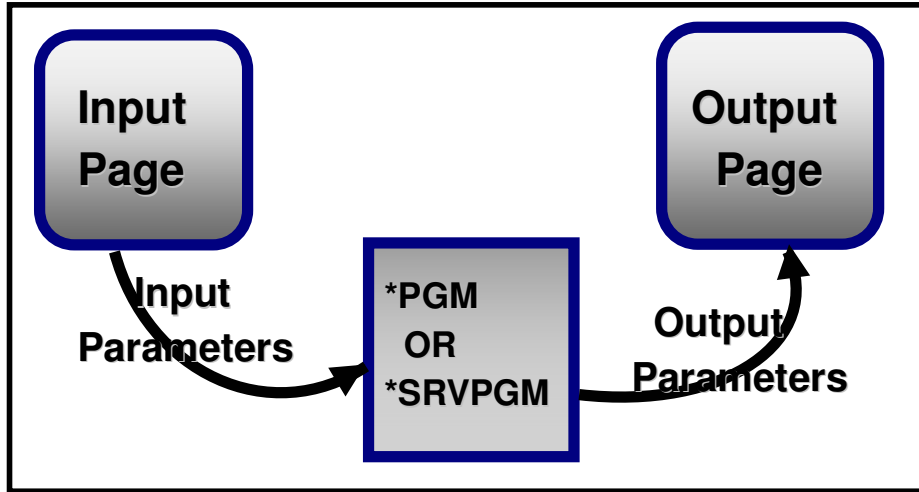
## Web Interaction Wizard

iSeries AD, IBM Toronto

- ✚ Wizard to help create iSeries RPG/COBOL web applications
  - To build a Web interaction from iSeries RPG/COBOL business logic, or with Java bean methods

# A Web Interaction

iSeries AD, IBM Toronto





# Build simple interaction

iSeries AD, IBM Toronto

http://localhost:9080/MyFirstWebApplication/CustomInquiry.jsp.do

### Input Form

custno: 0010100

http://localhost:9080/MyFirstWebApplication/CustomInquiry.do

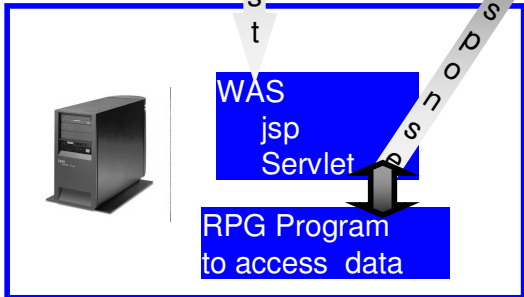
### Customer detail

CUSTNO: 0010100  
CUSTNA: Meriden Electronics Limited  
REPNO: 43443  
CONTACT: Alfredo Bayogno  
CPHONE: 206-865-4034  
CFAX: 206-865-4034  
CADDR: 10423 S.E. 30th Place  
CCITY: Bellevue, WA  
CCOUNT: U.S.A.  
CZIP: 98007  
CZIPLO: 1

Date:

R  
e  
q  
u  
e  
s  
t

R  
e  
s  
p  
o  
n  
s  
e





## Web Interaction Wizard

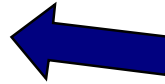
iSeries AD, IBM Toronto

- ▶ 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 provide 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
- ▶ Can optionally define message handling for Web pages, and specify flow control of pages
- ▶ STRUTS based application

## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts





## Runtime Configuration Wizard iSeries AD, IBM Toronto

### Define Runtime configuration

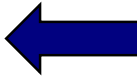
- To specify the authentication types
  - Sign on with specified values (server name, user ID, password)
  - Prompt for user ID and password
  - JCA connection (Advanced Edition)
  - Single Sign On (Advanced Edition)
- Library list setup
- Initial command when sign on
- Display detailed run-time errors
- Session timeout options
- Message Handling

Only prompted to  
signon once across multiple Web  
applications

## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts

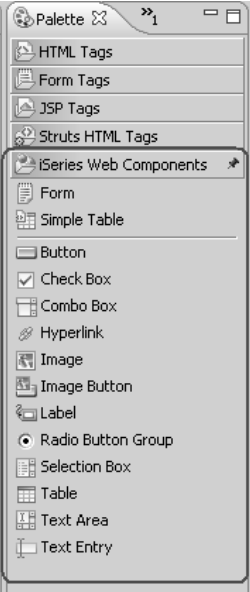


IBM Software Group Toronto Laboratory

## iSeries Web components

iSeries AD, IBM Toronto

- **Web UI components that allow visualization in Page Designer**
- **With iSeries built-in features**
- **Allows typical iSeries data formatting, e.g. edit code, edit word**
- **Table with Subfile capabilities to display a list of records from an iSeries database**



Web Tools for iSeries developers      September 2005      © 2005 IBM Corporation

- WC's implemented as VCT's so can be used in Page designers and viewed/edited in one of 3 views (design/source/preview)
- Formatting and validation defined for the WC in attribute view (e.g. format data as a date, numeric, currency etc. and specify that locale sensitive information, e.g. decimal symbol, currency symbol, to come from either the web client or the iSeries host defined in web.xml)

## AGENDA


iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools 
- Advanced Topics
  - Mixing Java with RPG
  - Struts

## Using iSeries Web Tools

iSeries AD, IBM Toronto

To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program 
- ✦ Create a Dynamic Web project
- ✦ Create Runtime configuration
- ✦ Design with Web Diagram Editor
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application

IBM Software Group Toronto Laboratory

# RPG Program

iSeries AD, IBM Toronto

**NOTE: Parameters to be passed**

```

Line 1      Column 1      Replace
.....1.....2.....3.....4.....5.....6.....7.....8.
000100      *
000200      *
000300      *   Sample RPGIV program to access file CUSTOML3
000400      *   Used in WDT hands on LAB
000500      *   Created by Claus Weiss
000600      *
000700      * F spec for file CUSTOML3 keyed by customer number
000800      FCUSTOML3 IF E           K DISK
000900      F*
001000      D*Input parameter from Web page
001100      DCustom01           S           like(CUSTNO)
001200      D* Data structure to specify output structure to return to Servlet
001300      D CSTRUC           E DS           extname(custom13:custom01)
001400      D feedback         S           20
001500      C *entry         plist
001600      C                  parm         custnoi
001700      C                  parm         cstruc
001800      C                  parm         feedback
002100      C                  eval         feedback=*blank
002101      C
002400      C*   Insert code here ----->
002600      C   CUSTNO1     CHAIN(e)     CUSTOML3
004200      C               IF          not %found(CUSTOML3)
004201      C               EVAL       FEEDBACK='CUS0001' + CUSTNO1
004202      C               ELSE
004203      C               EVAL       FEEDBACK='0'
004204      C               ENDIF
004600      C*   Done with inserting code ----->
004700      C*
004800      C               return
004900      C*
  
```

IBM Corporation

custnoi is customer number


cstruc is customer informaton

feedback is program return code

## Using iSeries Web Tools

iSeries AD, IBM Toronto

To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program
- ✦ Create a Dynamic Web project 
- ✦ Create Runtime configuration
- ✦ Design with Web Diagram Editor
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application

IBM Software Group Toronto Laboratory

# Create a Dynamic Web project

iSeries AD, IBM Toronto

File > New > Dynamic Web Project

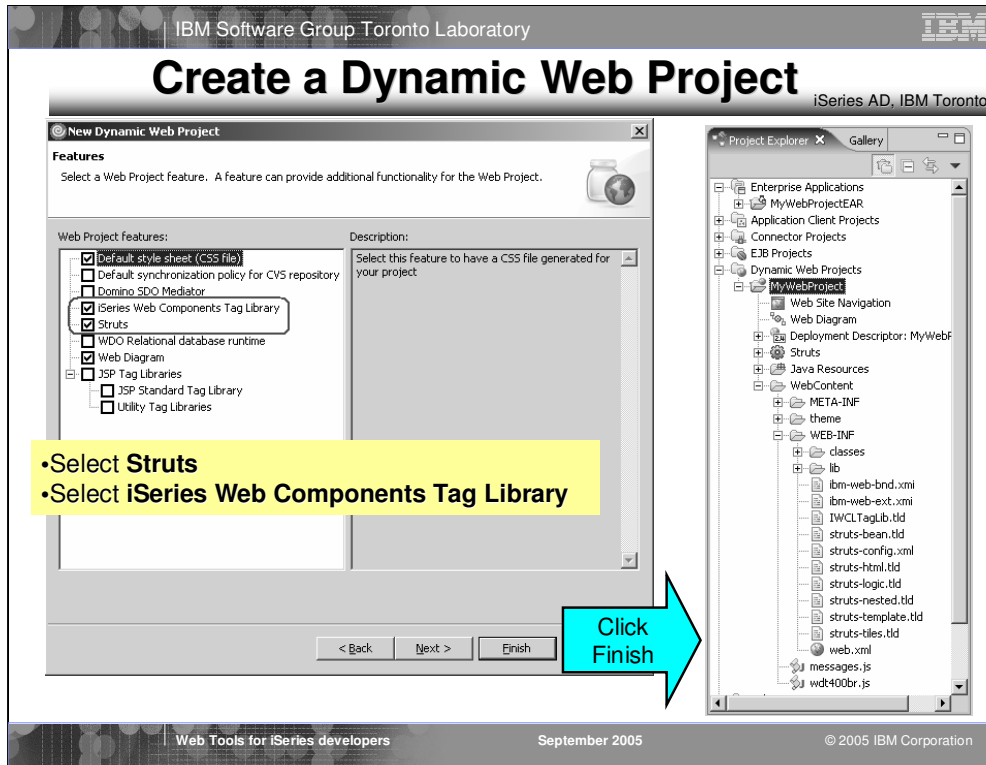
Web Tools for iSeries developers

September 2005

© 2005 IBM Corporation

- Servlet 2.4 = J2EE 1.4
- Servlet 2.3 = J2EE 1.3
- Servlet 2.2 = J2EE 1.2
- creates an Enterprise Application (EAR) project if it does not already exist






- Select **Struts** for a Struts-based application
- Select **iSeries Web Component Tag Library** to add the tag library to the project

## Using iSeries Web Tools

iSeries AD, IBM Toronto

To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program
- ✦ Create a Dynamic Web project
- ✦ Create Runtime configuration 
- ✦ Design with Web Diagram Editor
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application

IBM Software Group Toronto Laboratory

# Runtime Configuration Wizard

iSeries AD, IBM Toronto

- Select the project "MyWebProject" in **Project Explorer**
- Click Runtime Configuration icon

• Enter the name of the server that has the program or procedure in **Host name**

• Enter your user ID in **User ID**

Click Next

Web Tools for iSeries developers September 2005 © 2005 IBM Corporation

JCA Connector and Single Signon are available in advanced version.

# Runtime Configuration Wizard

iSeries AD, IBM Toronto

**iSeries Web Tools Run-time Configuration**

**Configure Run Time**  
Specify the run-time configuration for program call of iSeries Web development tools

Run-time library list

Library:  Add

Library	Library Position
WSSLABXX	*LAST

Change  
Remove  
Move Up

Current library: \*USRPRF

Initial command:

Display detailed run-time errors

Session timeout procedure

Library:  Program object:

Entry point:  Program type: \*SRVPGM

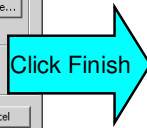
Message handling

Message resource:  Browse...

Use iSeries message file(s)

< Back Next > Finish Cancel


- Add any runtime library that are required to run the host program or procedure
- Enter library "wsslabxx" in **Library**
- Click **Add**
- Select **Display detailed run-time errors**



## Using iSeries Web Tools

iSeries AD, IBM Toronto

To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program
- ✦ Create a Dynamic Web project
- ✦ Create Runtime configuration
- ✦ Design with Web Diagram Editor 
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application

# Web Diagram Editor

iSeries AD, IBM Toronto

A Web diagram helps you to design the flow of a Web application. It consists of *nodes* and *connections* between nodes. A node is an icon that represents a resource such as a Web page, Java bean.

**Web Diagram**

**Web Diagram Palette**

A Web diagram is a file that helps you visualize the application flow of a Web application.

Add Nodes from the Palette to the free-form surface (click on Palette, then click on diagram).

Connect nodes by clicking Connection in Palette, then click on the two nodes to connect.

Double click on any new node in the diagram to launch a wizard to create the underlying object.

This note will disappear once a node is placed onto the free-form surface. For additional help, press F1.

Double click **Web Diagram** inside your Web project to open the Web Diagram Editor

# Web Diagram Editor

iSeries AD, IBM Toronto

Web - diagram.gph - IBM Rational Software Development Platform

File Edit View Navigate Search Project Run Window Help

Project Explorer: Enterprise Applications, Application Client Projects, Connector Projects, EJB Projects, Dynamic Web Projects, MyWebProject, Web Site Navigation, Web Diagram, Deployment Descriptor: MyW, Struts, Java Resources, WebContent, Other Projects, Web Services, Databases, Database Servers

Web Diagram Editor workspace:

A Web diagram is a file that helps you visualize the application flow of a Web application.

Add Nodes from the Palette to the free-form surface (click on Palette, then click on diagram).

Connect nodes by clicking on the connection in Palette on the two nodes.

Double click on the wizard icon in the diagram to launch a wizard to create a new object.

This note will disappear once a node is placed onto the free-form surface. For additional help, press F1.

Palette:

- Select
- Connection
- Note
- Struts Parts
- Struts Module
- Form Bean
- Action Mapping
- Web Interaction Action Mapping
- Web Parts
- Web Application
- Web Page**
- Faces Parts
- Faces Action

**Drag and drop Web Page node onto the Web Diagram Editor**

# Web Diagram Editor

iSeries AD, IBM Toronto

Web - diagram.gph - IBM Rational Software Development Platform

File Edit View Navigate Search Project Run Window Help

100%

Project... x1

Enterprise Applications  
Application Client Projects  
Connector Projects  
EJB Projects  
Dynamic Web Projects  
MyWebProject  
Web Site Navigation  
Web Diagram  
Deployment Descrip  
Struts  
Java Resources  
WebContent  
Other Projects  
Web Services  
Databases  
Database Servers

Web  
Remote System...

Select  
Connection  
Note  
Struts Parts  
Struts Module  
Form Bean  
Action Mapping  
Web Interaction  
Action Mapping  
Web Parts  
Web Application  
Web Page  
Faces Parts  
Faces Action

Page Data x4

Change the name of the Web page to "custInquiryInput.jsp"

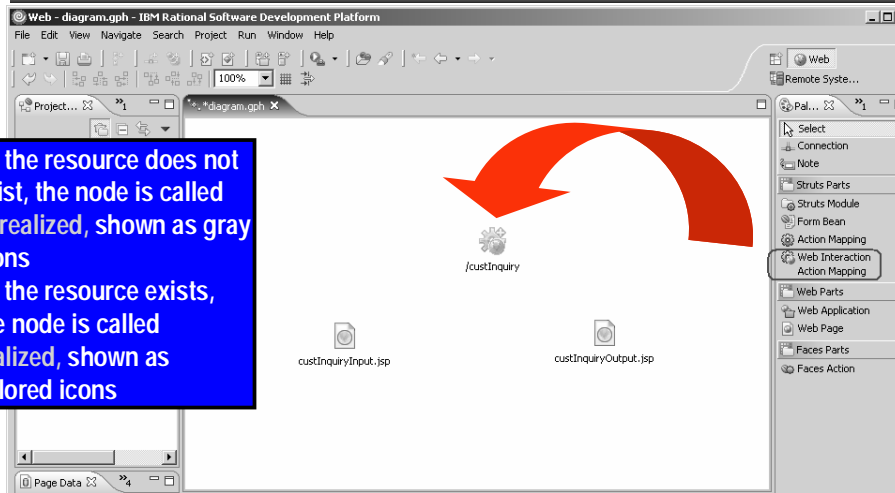


# Web Diagram Editor

iSeries AD, IBM Toronto

▪if the resource does not exist, the node is called unrealized, shown as gray icons

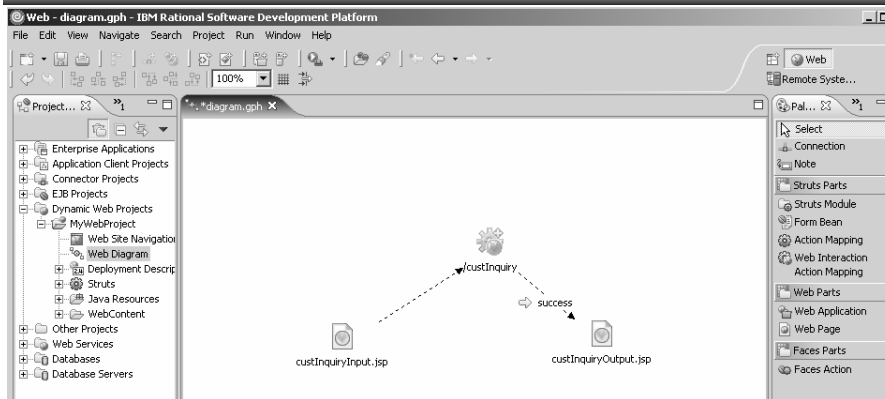
▪If the resource exists, the node is called realized, shown as colored icons



- Drop a second **Web Page** node and name it as “custInquiryOutput.jsp”
- Then drag and drop a **Web Interaction Action Mapping** node to represent the interaction, and name it as “custInquiry”

# Web Diagram Editor

iSeries AD, IBM Toronto



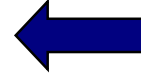
- Right click on custInquiryInput node, select **Connection**
- Draw a connection line from custInquiryInput node to custInquiry node
- Right click on custInquiry node, select **Connection**
- Draw a connection line from custInquiry node to custInquiryOutput node
- Give a name "success" to the local forward connection
- A local forward connection represents the possibility that this action can go to a new destination, "custInquiryOutput.jsp" through a forward

## Using iSeries Web Tools

iSeries AD, IBM Toronto

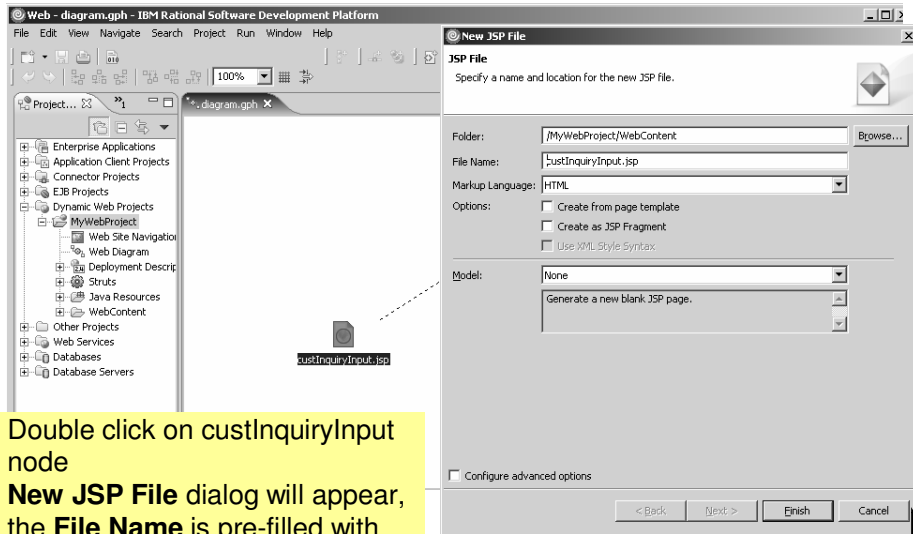
To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program
- ✦ Create a Dynamic Web project
- ✦ Create Runtime configuration
- ✦ Design with Web Diagram Editor
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application



# Create Web Page

iSeries AD, IBM Toronto



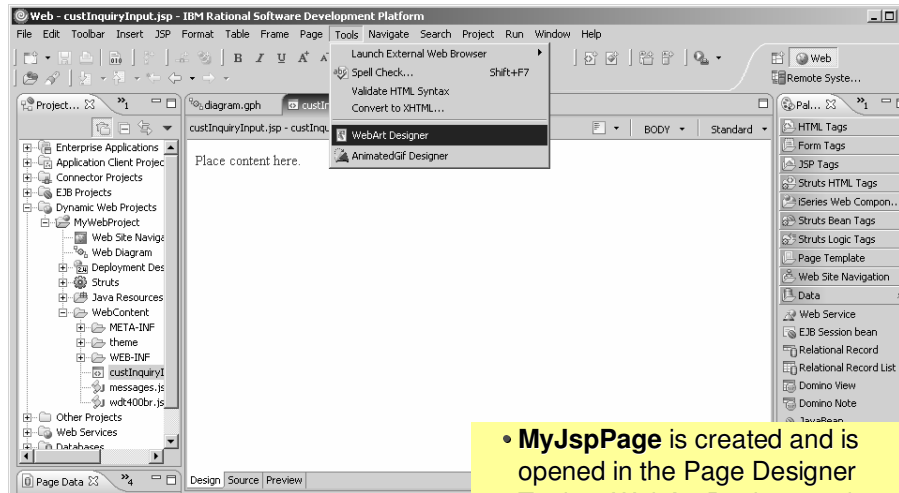
- Double click on custInquiryInput node
- **New JSP File** dialog will appear, the **File Name** is pre-filled with the name of the node
- Click **Finish**

Click Finish

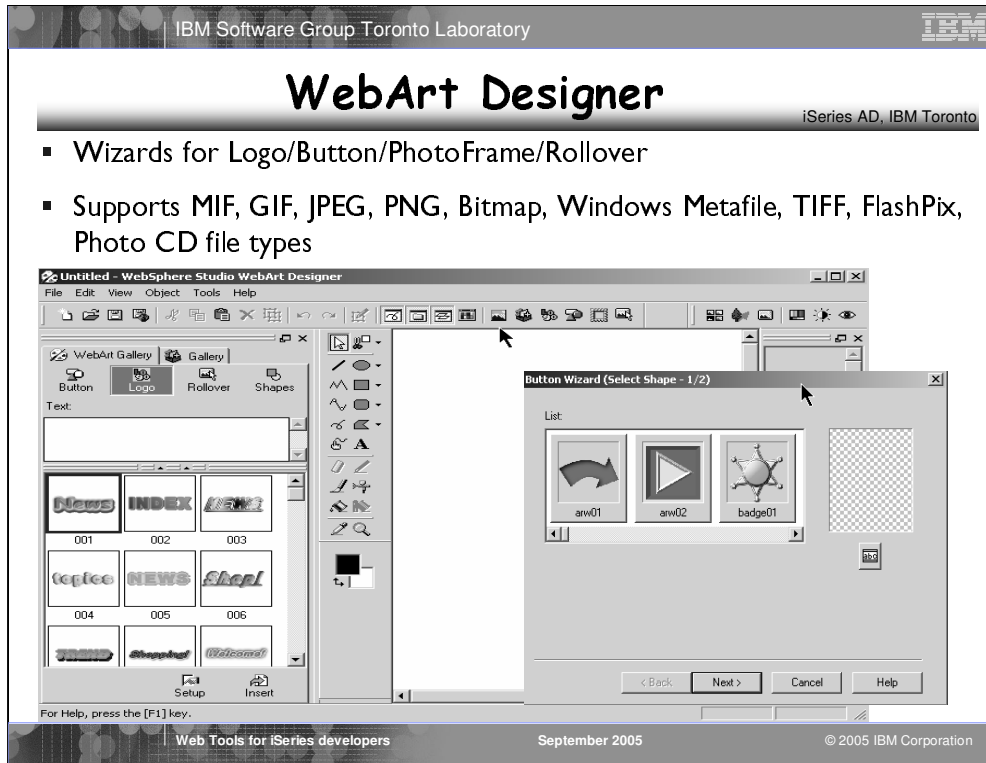


# Design Web Page

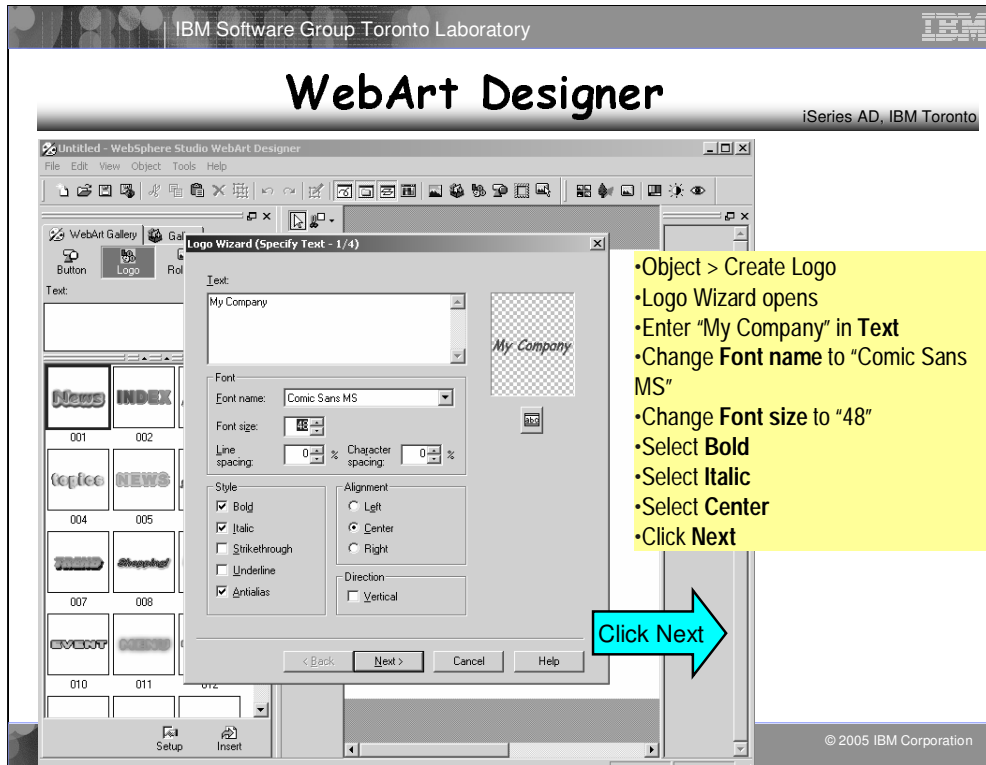
iSeries AD, IBM Toronto



- **MyJspPage** is created and is opened in the Page Designer
- Tools > WebArt Designer to launch the WebArt Designer to design a logo



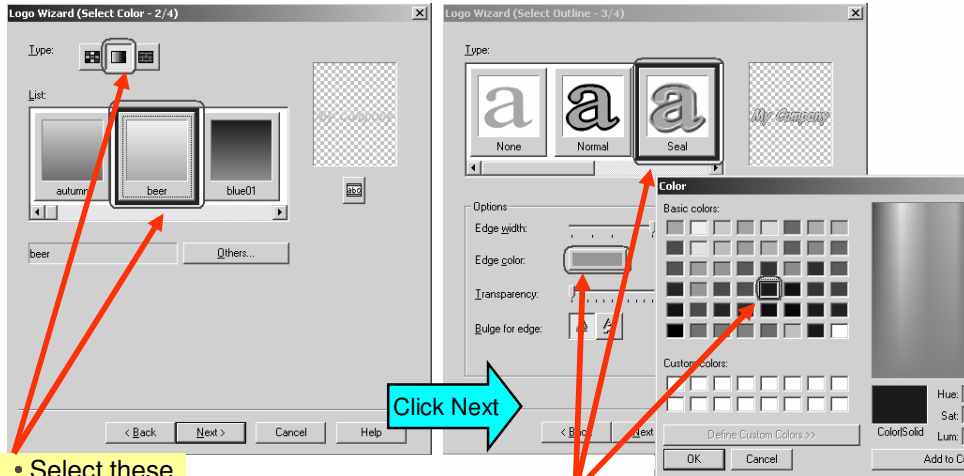
- WebArt Designer assists you in creating, manipulating, and editing graphics and image files that you can import to Page Designer.
- Use this application to edit the color tone and contrast of imported photographs



To create a logo

# WebArt Designer

iSeries AD, IBM Toronto



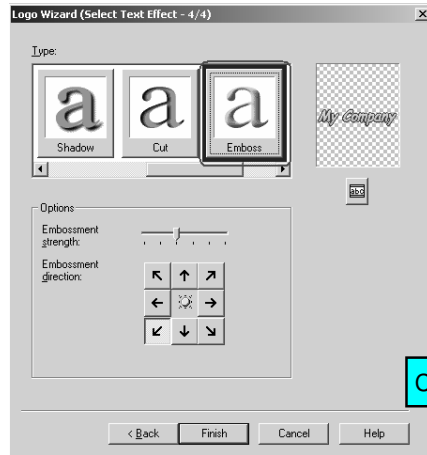
• Select these

- Select **Seal**
- Click **Edge color**
- Select blue color
- Click **OK**
- Click **Next**



# WebArt Designer

iSeries AD, IBM Toronto

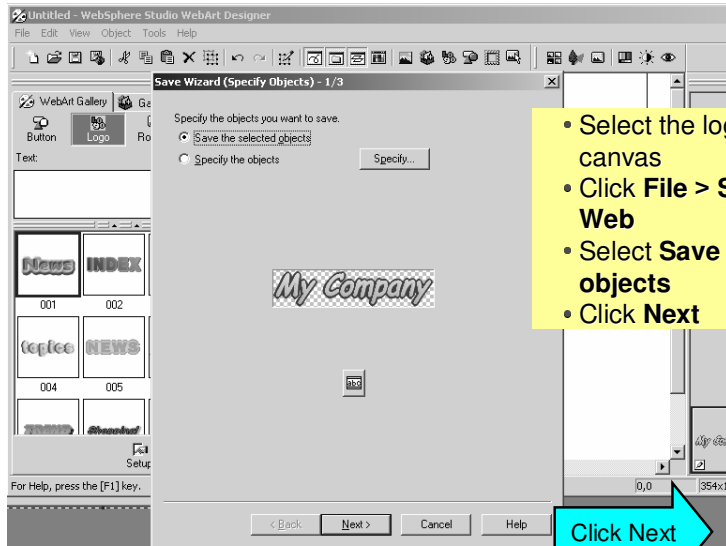


- Select **Emboss**
- Click **Finish**



# WebArt Designer

iSeries AD, IBM Toronto



- Select the logo object on the canvas
- Click **File > Save Wizard for Web**
- Select **Save the selected objects**
- Click **Next**

Click Next

# WebArt Designer

iSeries AD, IBM Toronto

**Save Wizard (Select File Format) - 2/3**

Select a format for the image file.

**GIF** This format can handle only 256 colors. By saving an image as a GIF file, you can reduce its file size. This format is suitable for images that do not require many colors, such as line drawings or graphs. You can make part of an GIF image transparent.

**JPEG** This format is in general use on the Internet because of its small file size and inconspicuous image degradation. This format is suitable for full-color images, such as photos taken with a digital camera. You cannot make a part of a JPEG image transparent.

**PNG** This format maintains the quality of an image, but the file size becomes larger than that of a JPEG. This format is suitable for full-color images whose quality is important, such as computer graphics. You can make a part of a PNG image transparent; however, only a few browsers can handle it correctly.

< Back   Next >   Cancel   Help

**Save Wizard (GIF Format) - 3/3**

[Original image]   [Image to be saved]

File size before: 105336 bytes  
File size after: 9684 bytes

Palette  
 Use the palette  
 Use the global palette  
Colors: 256  
Dither: Yes

**Save As**

Save in: My Documents

My Pictures

File name: MyCompanyLogo   Save

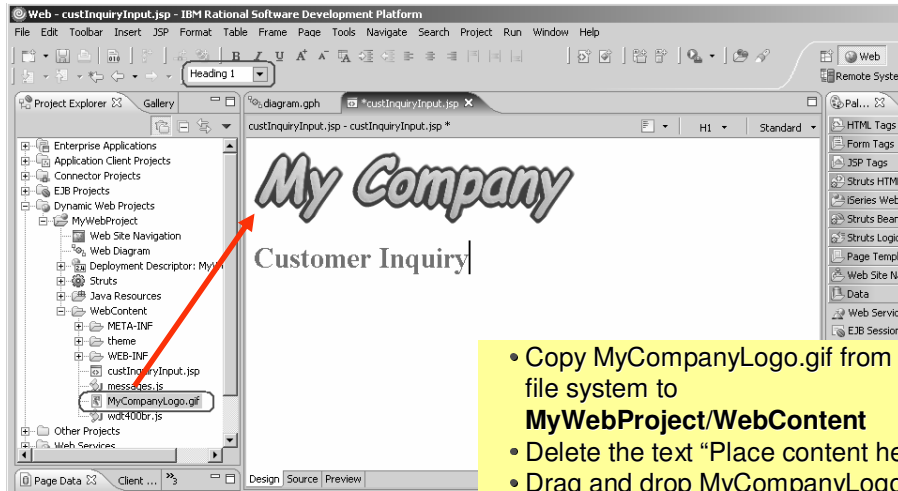
Save as type: GIF files (\*.gif)   Cancel



- Select **GIF**
- Click **Next**
- Click **Finish**
- Save to **MyCompanyLogo.gif**

# Design Web Page

iSeries AD, IBM Toronto



- Copy MyCompanyLogo.gif from your file system to **MyWebProject/WebContent**
- Delete the text "Place content here"
- Drag and drop MyCompanyLogo.gif to the canvas
- Place cursor under the logo
- Select **Heading 1** from drop down
- Type "Customer Inquiry"

IBM Software Group Toronto Laboratory

# Design Web Page

iSeries AD, IBM Toronto

Web Tools for iSeries developers

- Drag and drop **Form** from **iSeries Web Components** drawer to a location under the page heading
- A dialog will appear, click **Yes**

- When you create a JSP page, the default workbench encoding is used for the page.

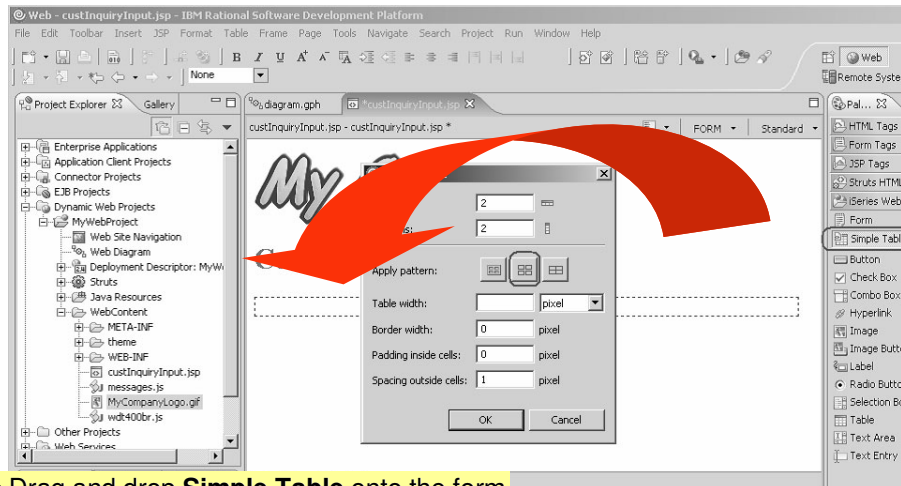
- When you insert an iSeries Web Component the very first time, it will prompt you this message, “It is recommended to use UTF-8 for encoding. Do you want to change it?”

- By selecting Yes will change the page encoding and the charset values to UTF-8.

- UTF-8 encoding enables you to have globalization support in your Web application.

# Design Web Page

iSeries AD, IBM Toronto



- Drag and drop **Simple Table** onto the form to insert a table for alignment
- A dialog will appear, select the button in the centre of **Apply pattern**
- click **OK**

September 2005

© 2005 IBM Corporation

- The button in the centre of **Apply pattern** means no border for the table, that is the table is invisible.

IBM Software Group Toronto Laboratory

# Design Web Page

iSeries AD, IBM Toronto

Web - custInquiryInput.jsp - IBM Rational Software Development Platform

File Edit Toolbar Insert JSP Format Table Frame Page Tools Navigate Search Project Run Window Help

Project Explorer

- Enterprise Applications
- Application Client Projects
- Connector Projects
- EJB Projects
- Dynamic Web Projects
- MyWebProject
  - Web Site Navigation
  - Web Diagram
  - Deployment Descriptor: MyW...
  - Struts
  - Java Resources
  - WebContent
  - META-INF
  - theme
  - WEB-INF
    - custInquiryInput.jsp
    - messages.js
    - MyCompanyLogo.gif
    - web400hr.ic

Design Source Preview

Properties

TBODY	Name: LABEL1
TR	Initial value: Customer Number:
TD	

iwcd:WLabel

- Data
- Styles

HTML Tags

- Form Tags
- JSP Tags
- Struts HTML Tags
- iSeries Web Comp...
- Form
- Simple Table
- Button
- Check Box
- Combo Box
- Hyperlink
- Image
- Image Button
- Label
- Radio Button Group
- Selection Box
- Table
- Text Area
- Text Entry

Web Tools for iSeries developers

- Drag and drop **Label** onto row 1 column 1 of the table
- Type "Customer Number:" in properties view

•As you enter the **Initial Value** field, the value is shown on the Page Designer right away.

# Design Web Page

iSeries AD, IBM Toronto

Web - custInquiryInput.jsp - IBM Rational Software Development Platform

File Edit Toolbar Insert JSP Format Table Frame Page Tools Navigate Search Project Run Window Help

Project Explorer Gallery %diagram.gch \*custInquiryInput.jsp \*  
custInquiryInput.jsp - custInquiryInput.jsp \* iwd:WButton Standard

Enterprise Applications  
Application Client Projects  
Connector Projects  
EJB Projects  
Dynamic Web Projects  
MyWebProject  
Web Site Navigation  
Web Diagram  
Deployment Descriptor: MyW  
Struts  
Java Resources  
WebContent  
META-INF  
theme  
WEB-INF  
custInquiryInput.jsp  
miscellaneous

HTML Tags  
Form Tags  
JSP Tags  
Struts HTML Tags  
iSeries Web Comp...  
Form  
Simple Table  
Button  
Check Box  
Combo Box  
Hyperlink  
Image  
Image Button  
Label  
Radio Button Group  
Selection Box  
Table  
Text Area  
Text Entry

Struts Bean Tags  
Struts Logic Tags  
Page Template  
Web Site Navigation  
Data

Preview  
Quick Edit Servers Console Problems  
Name: BUTTON1  
Label: Submit  
Button type: Submit  
Initial state:  Disabled  Hidden  
Tool tip:  
Access key:

- Drop **Text Entry** onto row 1 column 2 of the table
- Drop **Button** onto row 2 column 1 of the table
- Type "Submit" in **Label**
- Save the file

Web Tools for iSeries developers September 2005 © 2005 IBM Corporation

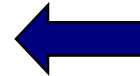


## Using iSeries Web Tools

iSeries AD, IBM Toronto

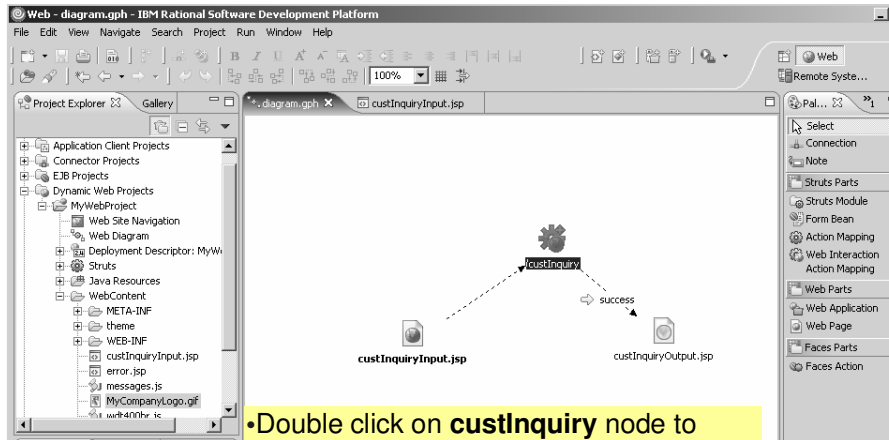
To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program
- ✦ Create a Dynamic Web project
- ✦ Create Runtime configuration
- ✦ Design with Web Diagram Editor
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application



# Web Interaction Wizard

iSeries AD, IBM Toronto



- Double click on **custInquiry** node to open Web Interaction wizard
- Note: **custInquiryInput.jsp** node is now shown as color for it is *realized*

# Web Interaction Wizard

iSeries AD, IBM Toronto

Web Interaction - New

**Specify a Name and Location for your Web Interaction**  
Specify the name and location for the resources created for this Web Interaction.

Destination folder:

Web Interaction name:

Java package prefix:

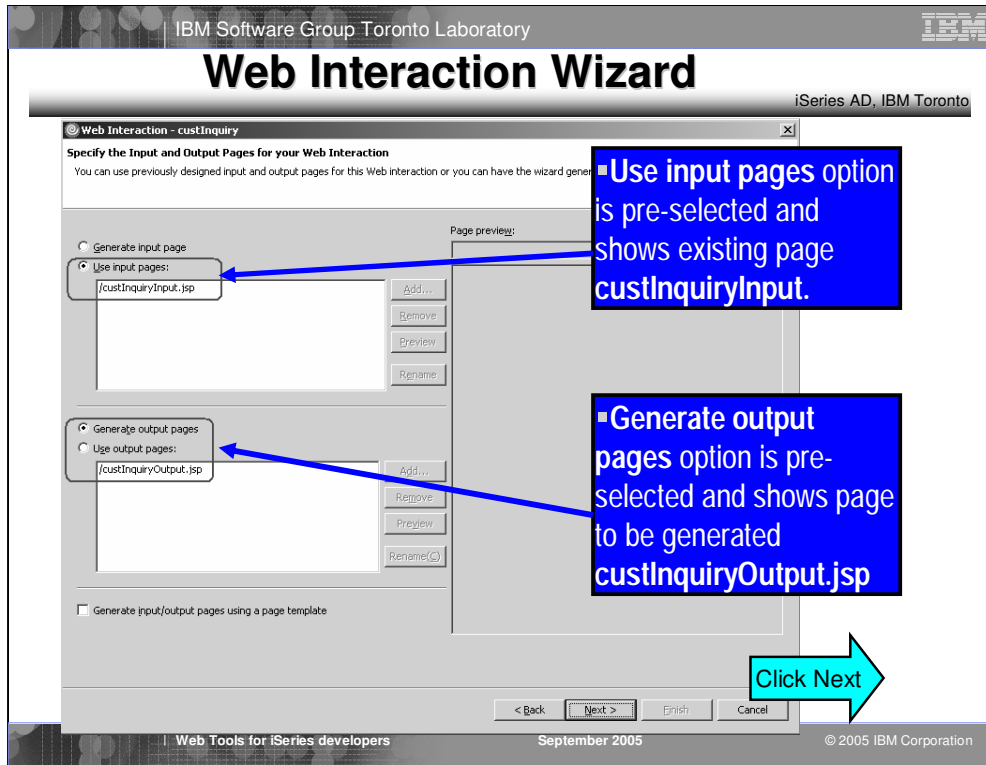
Use error page

Invalidate session after the interaction occurs

Terminate iSeries connection

The Web Interaction name is pre-filled with "custInquiry"





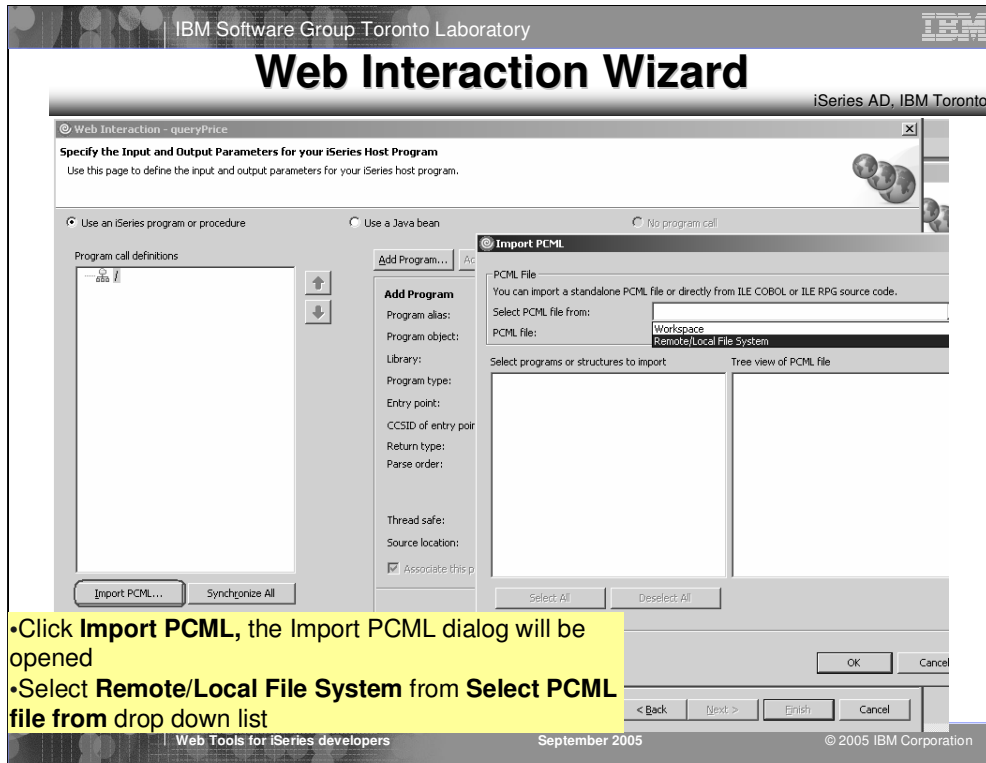
To define input and output pages.

2 modes:

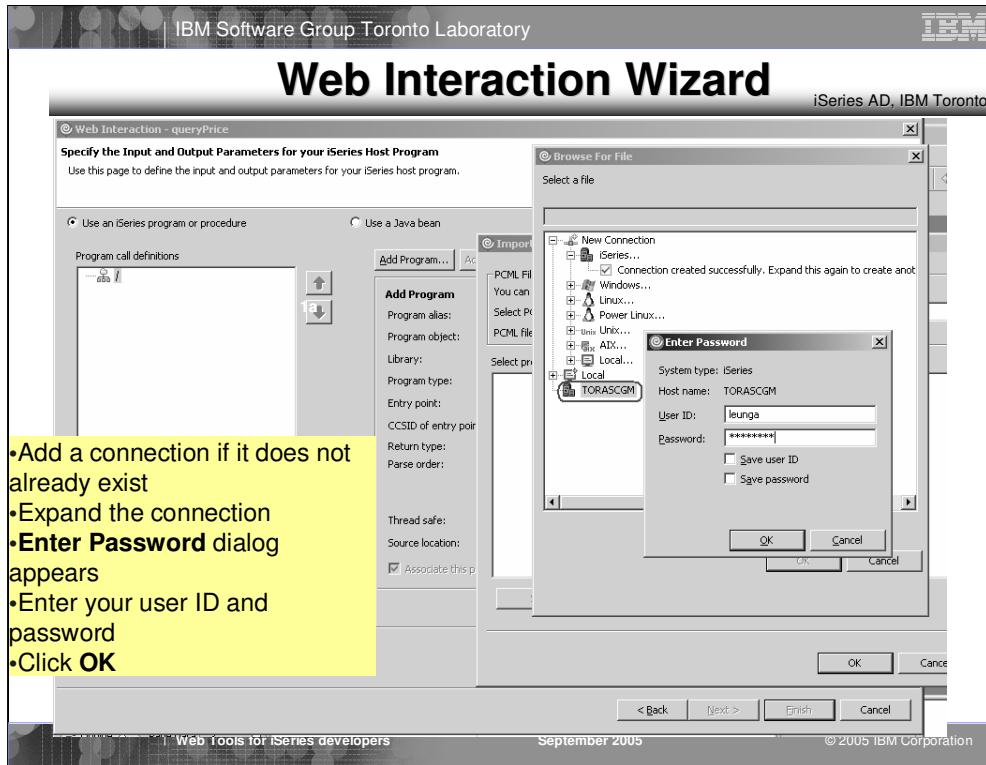
- Let wizard to generate the Web pages for you
- You specify existing pages

You can specify multiple **output** pages or the wizard can generate multiple **output** pages for you.

Select **Generate input/output pages using a page template** allows you to provide a page template, and the wizard will generate Web pages using the specified page template. This option is only available if **Generate input page** or **Generate output pages** is selected



- To identify the host program or Java bean to call for each interaction.
- You can import the definitions from a program source
- you can also specify program and parameter definitions manually if you do not have the source
- Or you can import a pcml file if it has already been generated by the RPG/COBOL compiler or Program Verifier



Look up the program source from the host

# Web Interaction Wizard

iSeries AD, IBM Toronto

•Right click on the connection

•Select **Properties**, Properties dialog appears

•Select **Subsystems** on the left pane

•Select **iSeries Commands** tab

•Enter library "wsslabxx" in **Library**

•Click **Add**

•Click **OK**

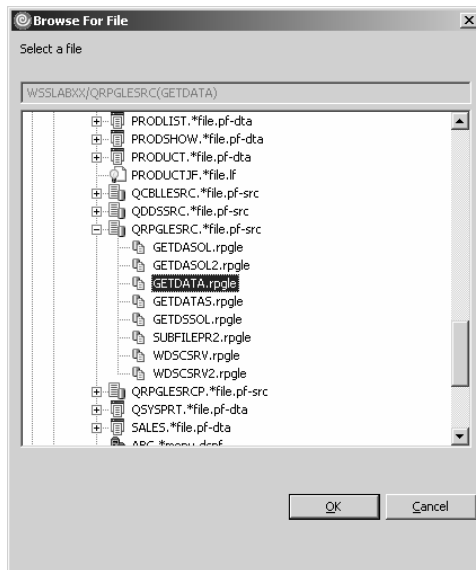
Web Tools for iSeries developers

September 2005

© 2005 IBM Corporation

# Web Interaction Wizard

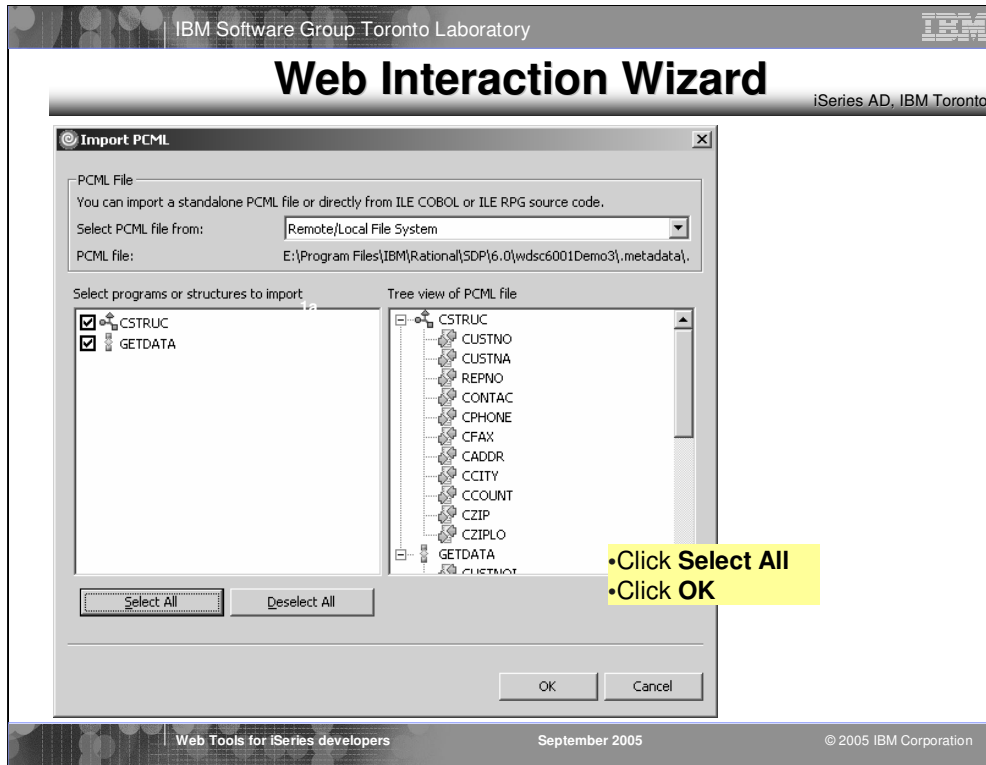
iSeries AD, IBM Toronto



- Expand **\*LIBL**
- Expand "wsslabxx"
- Locate the program source /QRPGLESRC.FILE/GETDATA.RPGLE
- Click **OK**

Can also import PCML, or import iSeries ILE source





A tree representation of program and parameter relationship

CSTRUC is a structure

GETDATA is a program

The tree of the right shows all the fields in the structure and the program.

# Web Interaction Wizard

iSeries AD, IBM Toronto

Web Interaction - custInquiry

**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.

**Describe program**

Use an iSeries program or procedure     Use a Java bean     No program call

Program call definitions

- GETDATA
- CSTRUC
- CUSTNOI
- CSTRUC (CSTRUC)
- FEEDBACK

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

**Edit Program**

Program alias: GETDATA  
Program object: GETDATA  
Library: \*LIBL  
Program type: \*PGM  
Entry point: GETDATA  
CCSID of entry point:  
Return type: void  
Parse order: CSTRUC, FEEDBACK  
Thread safe: false  
Source location: /New...

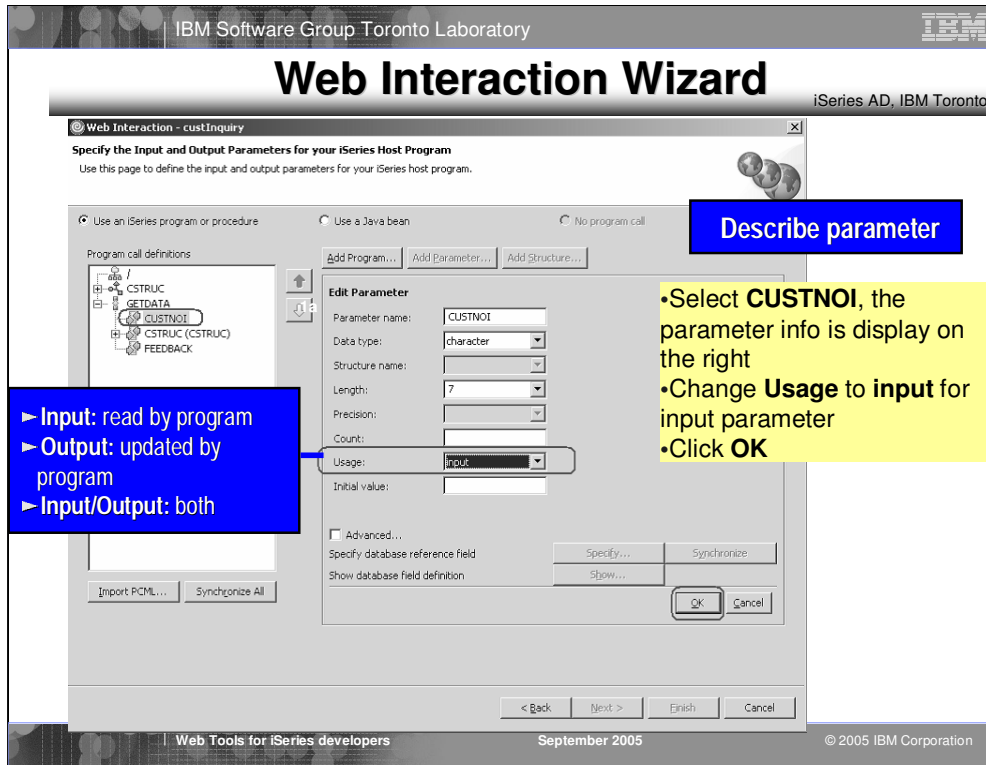
Associate this program with the interaction

Import PCML...    Synchronize All

OK    Cancel

< Back    Next >    Finish    Cancel

- Select **GETDATA** on the left pane, the program info is displayed on the right
- Select **Associate this program with the interaction**
- Click **OK**



Parameters imported from source are all default to “input/output” usage type.

IBM Software Group Toronto Laboratory

# Web Interaction Wizard

iSeries AD, IBM Toronto

Web Interaction - custInquiry

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 program or procedure  Use a Java bean  No program call

Program call definitions

- CSTRUC
- GETDATA
- CUSTINDI
- CSTRUC (CSTRUC)**
- FEEDBACK

**Describe parameter**

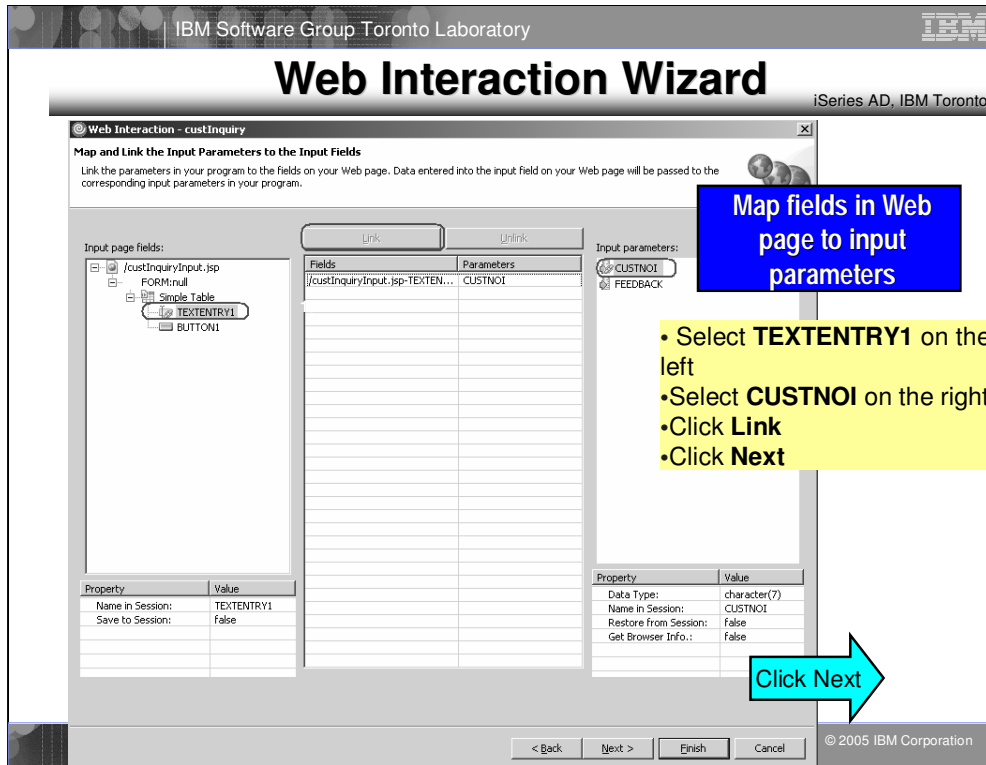
**Edit Parameter**

Parameter name: CSTRUC  
Data type: structure  
Structure name: CSTRUC  
Length:   
Precision:   
Count:   
Usage: **output**  
Initial value:   
 Advanced...  
Specify database reference field:   
Show database field definition:   
Specify... Synchronize  
Show...   
OK Cancel

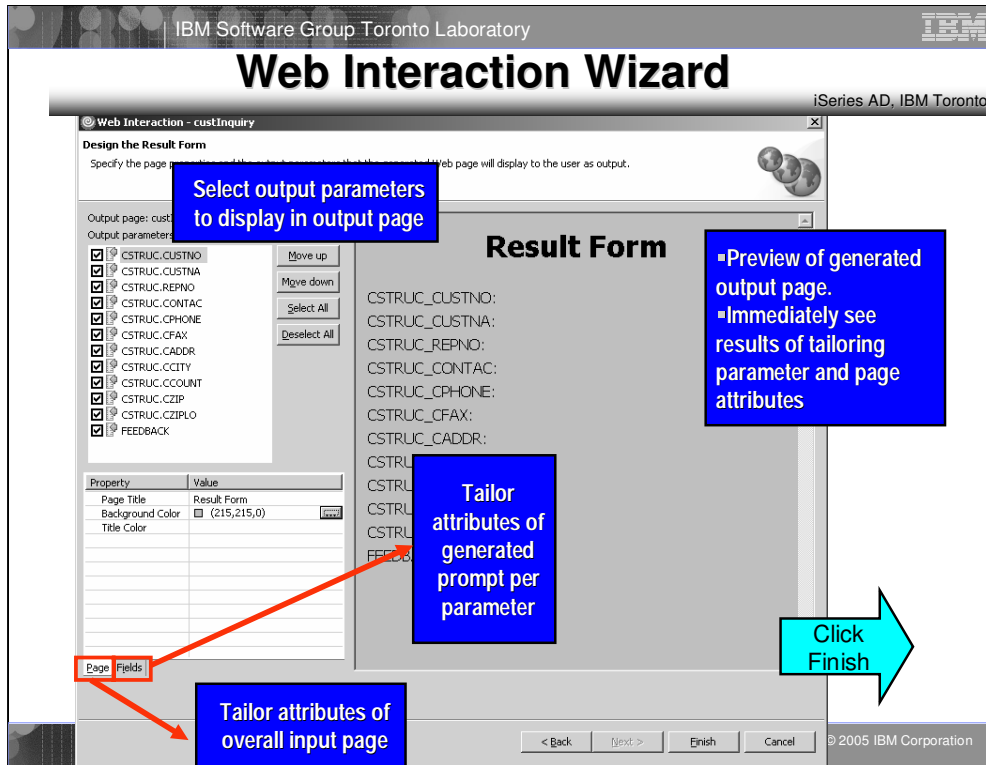
**Click Next**

Web Tools for iSeries developers September 2005 © 2005 IBM Corporation

CSTRUC is a structure



- The input page is created by user, the wizard has no clue of which field is for which parameter, so field mappings are required
- **TEXTENTRY1** is the name of the text entry field for Customer Number in `custInquiryInput.jsp`, it is mapped to **CUSTNOI** input parameter
- When the Web application is run, data from this **TEXTENTRY1** field will be passed to the RPG program as input parameter, and the host program will process it and return the data in the output parameters.



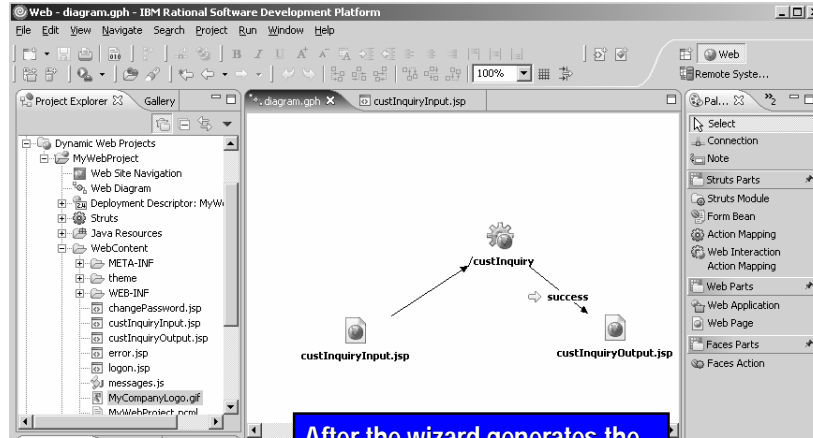
Page Tab – allows you change the page title, background color of the page, and title color

Fields Tab – allows you to the label, web component type, size of each field



# Web Diagram Editor

iSeries AD, IBM Toronto



After the wizard generates the Web pages, Java Action classes, icons are *realized* and are shown as colored

IBM Software Group Toronto Laboratory

# Web Interaction Wizard

iSeries AD, IBM Toronto

The screenshot shows a Project Explorer window for a Dynamic Web Project named 'MyWebProject'. The tree structure includes folders for 'Web Site Navigation', 'Web Diagram', 'Deployment Descriptor: MyWebProject', 'Struts', 'Java Resources', and 'WebContent'. Under 'Java Resources', there are sub-folders for 'actions', 'beans', 'forms', and 'resources', containing files like 'CustInquiryAction.java', 'CustInquiryInputForm.java', and 'CustInquiryOutputForm.java'. Under 'WebContent', there are folders for 'META-INF', 'theme', and 'WEB-INF', containing files like 'changePassword.jsp', 'custInquiryInput.jsp', 'custInquiryOutput.jsp', 'error.jsp', 'login.jsp', 'messages.js', 'MyCompanyLogo.gif', 'MyWebProject.pcm1', 'timeout.jsp', 'wdt400br.js', 'custInquiry.wit', and 'MyWebProject.mpcm1'. Two blue callout boxes with arrows point to specific files: one points to the 'actions' folder and its contents, and another points to the 'custInquiry.wit' file.

Files that are generated by the wizard: Java classes, properties files, JSP files, JavaScript files, PCML file

.wit file is generated per interaction for round-tripping purpose.

Web Tools for iSeries developers      September 2005      © 2005 IBM Corporation

You can double click on custInquiry.wit to reopen the Web Interaction wizard with custInquiry interaction information.



## What's Next?

iSeries AD, IBM Toronto

### ● What did you do so far?

- ▶ Created a Dynamic Web project
- ▶ Set runtime configuration to call your RPG Program remotely
- ▶ Design your Web application using Web Diagram Editor
- ▶ Design your input page
- ▶ Created an 'iSeries Web Interaction' to link your RPG program parameters to the input page and output page
  - ▶ The input page was created by you
  - ▶ The output page was generated for you


### ● Next?

- ▶ Run locally to test your application Or
- ▶ Publish all your files to the iSeries server

## Using iSeries Web Tools

iSeries AD, IBM Toronto

To create a Web-based front-end customer inquiry application that uses RPG business logic residing on an iSeries server

- ✦ Create or reuse RPG program
- ✦ Create a Dynamic Web project
- ✦ Create Runtime configuration
- ✦ Design with Web Diagram Editor
- ✦ Design a Web page
- ✦ Create Web Interaction
- ✦ Test Run the Web Application 



## WDSc: 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 Server"**
  - **Wait for the magic...**
- ▶ Your Web application will run!
  - **Opens Server perspective**
    - ▶ Publishes it to the built-in copy of WAS (and starts WAS)
    - ▶ Brings up a Web Browser
    - ▶ **Runs your application!!**
      - ✓ Tip: you can set breakpoints in your Java code!

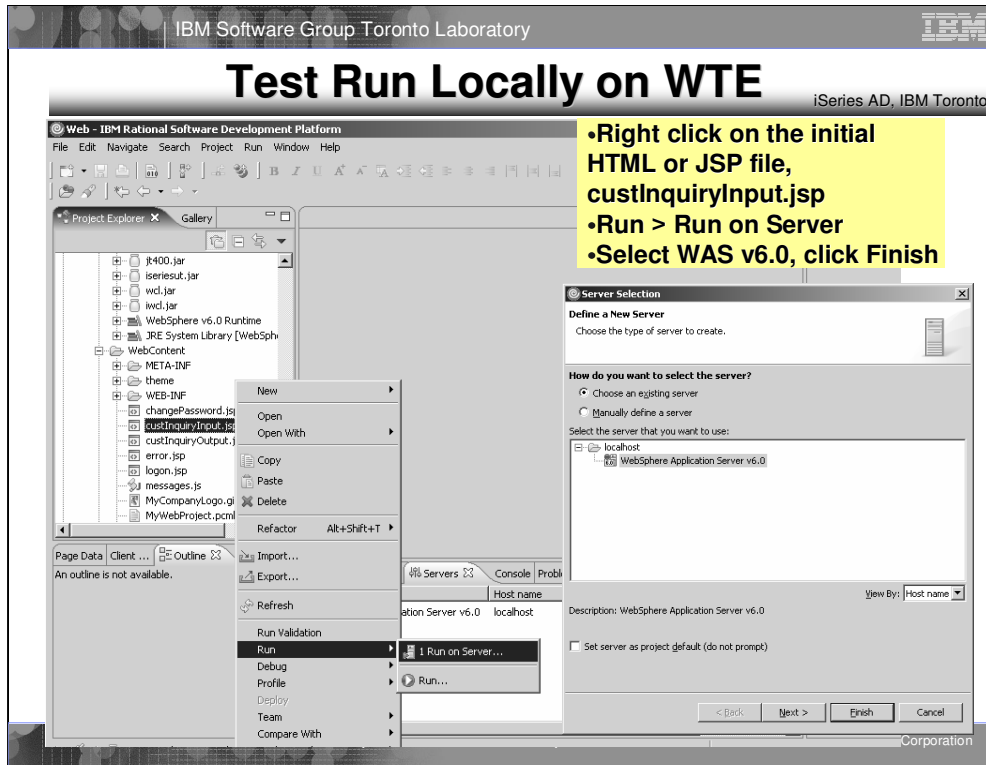
## WebSphere Test Environment iSeries AD, IBM Toronto

### **A local copy of WAS 6.0 Single Server Edition is embedded in the IDE**

- ▶ Integrated with Server Tools to enable instant and dead-easy testing of Web projects, EJB projects, Enterprise Application projects within WDS.
- ▶ Supports configuring multiple Web applications
- ▶ Supports multiple servers that can be configured and run at the same time
- ▶ Provides access to the WAS Administration Console

Can create a server instance for deploying e.g. EAR 1-5, another for EAR 1-4

Can create a server instance for deploying with security configuration “1”, another for security configuration “2”



You can select to test run on other versions of WAS if they are installed.

# Testing the Interaction

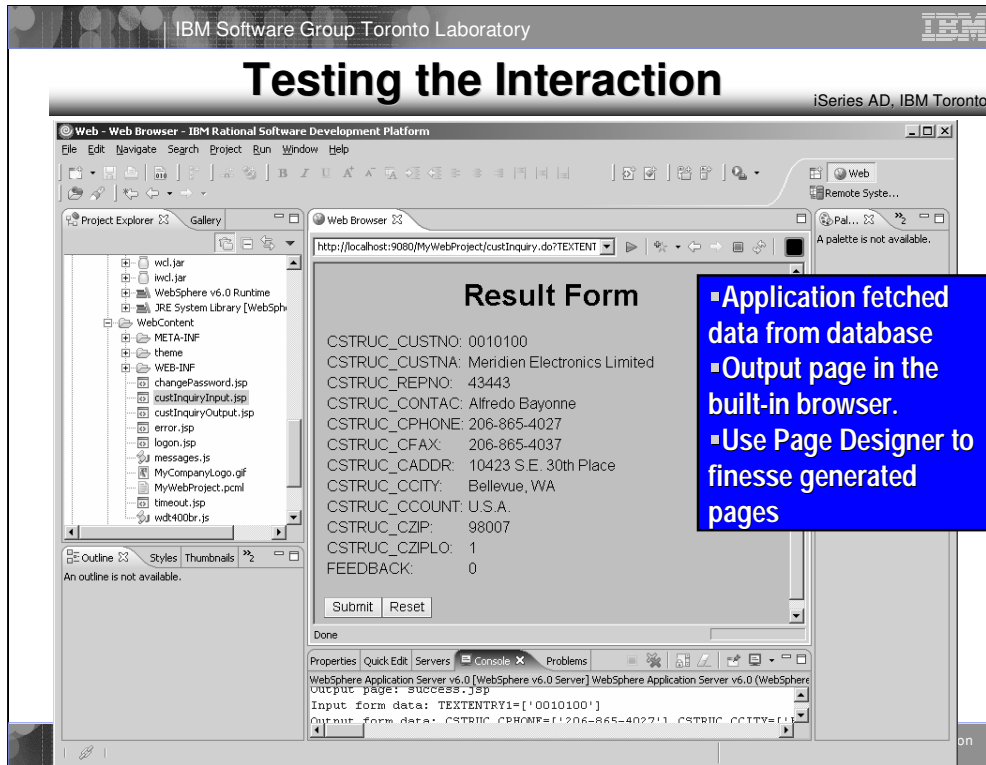
iSeries AD, IBM Toronto

The screenshot displays the IBM Rational Software Development Platform interface. On the left, the Project Explorer shows a file tree with various JSP files, including 'custInquiryInput.jsp'. The central Web Browser window shows a web page titled 'My Company' with a 'Customer Inquiry' form. The form includes a text input field for 'Customer Number' containing '0010100' and a 'Submit' button. At the bottom, the Console view shows server logs with timestamps and messages from 'FileRepositor A'. Three callout boxes provide additional information: a blue box points to the browser window, a yellow box lists instructions for testing the form, and another blue box points to the console view.

Input page in the built-in browser

- Enter "0010100" for **Customer Numer**
- Click **Submit**
- The action that is associated with this Submit button will invoke the RPG program, which will return customer data in the output parameter CSTRUC1 and display it on the output page

Console entries can be viewed in console view



You can customize the generated output page using the Page Designer.

## AGENDA

iSeries AD, IBM Toronto

- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts





## Java Calling RPG

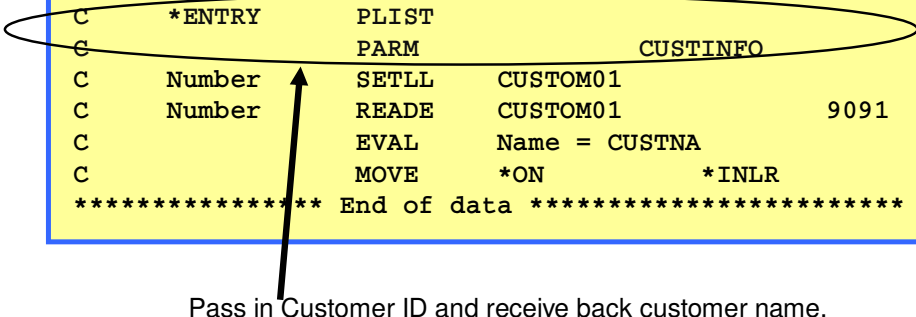
iSeries AD, IBM Toronto

### 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           READ  CUSTOM01           9091
C                 EVAL   Name = CUSTNA
C                 MOVE   *ON           *INLR
***** End of data *****

```



Pass in Customer ID and receive back customer name.

- F-spec defines the database File containing customer information records
- D-spec defines data structure CUSTINFO, containing two fields, Name, Number.
- C-spec defines one parameter, the struct- type inputoutput



## 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 Application

Series AD IBM Toronto

```
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--> " + ok);
        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

Toolbox classes used for making program calls.

# Mixing Java and RPG

iSeries AD, IBM Toronto

## 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

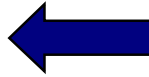
Save password

OK Cancel

## AGENDA

iSeries AD, IBM Toronto

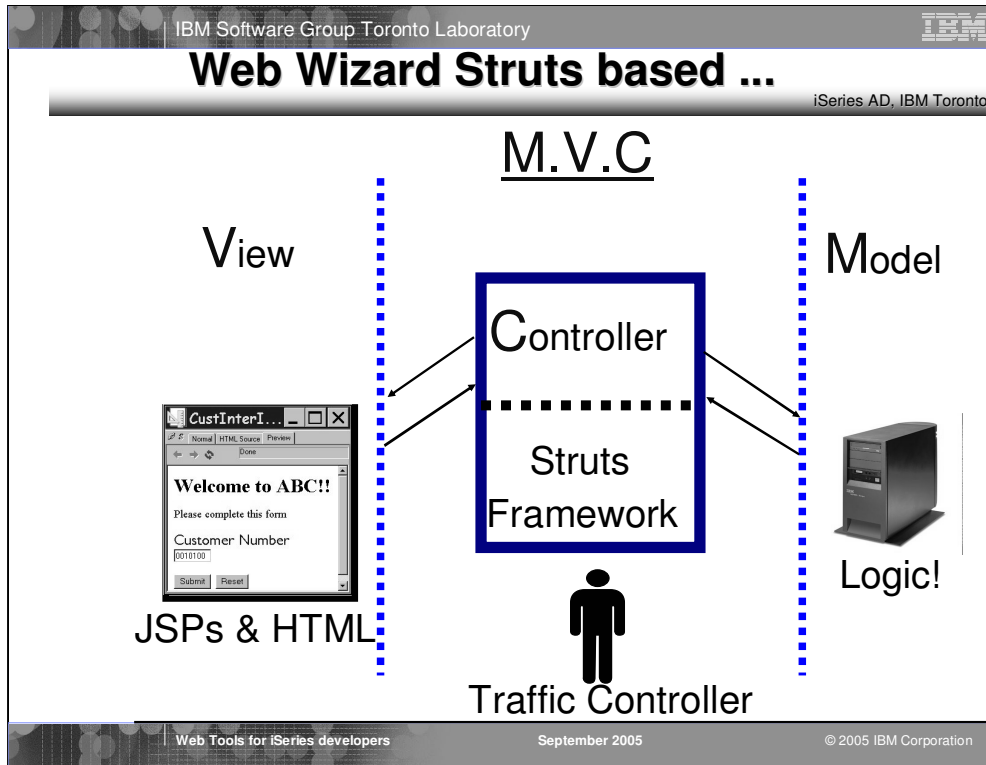
- e-business Primer
  - AD Model, traditional
  - e-Business Application, Web Model
- Introducing Web Tools for iSeries
  - J2EE Enterprise Application
  - Web Tools At A Glance
  - Dynamic Web Project
  - Web Interaction wizards
  - Runtime Configuration wizard
  - iSeries Web Components
- Demo: Using iSeries Web Tools
- Advanced Topics
  - Mixing Java with RPG
  - Struts



# Struts

iSeries AD, IBM Toronto

- **Web Interactions are Struts based!**
  - ± **Struts environment**
  - ± **Simple Struts application**
  - ± **iSeries Interaction and Struts**



**What is Struts? A development framework for Java servlet applications based upon the Model-View-Controller (MVC) design paradigm.**

# Simple Struts web application

iSeries AD, IBM Toronto

→ Determine weekday from date

→ One input page  
 → Two different Output pages  
 Struts-config file

```

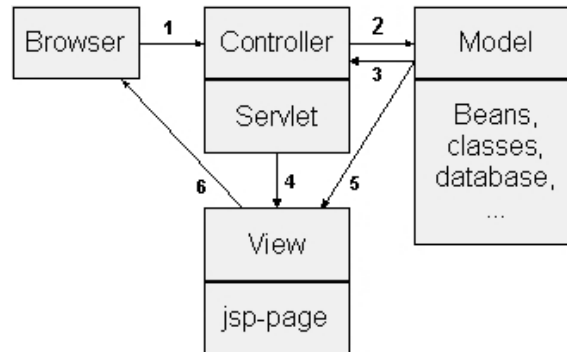
<action-mappings>
<action name="formBean" path="/computeDay"
  scope="request"
  type="com.ibm.dayofweek.actions.ComputeDayAction"
  input="index1.jsp">
</action>
<forward name="success" path="/output.jsp">
</forward>
<forward name="failure" path="/index1.jsp">
</forward>
<forward name="sunday" path="/output_sunday.jsp">
</forward>
</action-mappings>
  
```

It Is A Sunday Wonderful



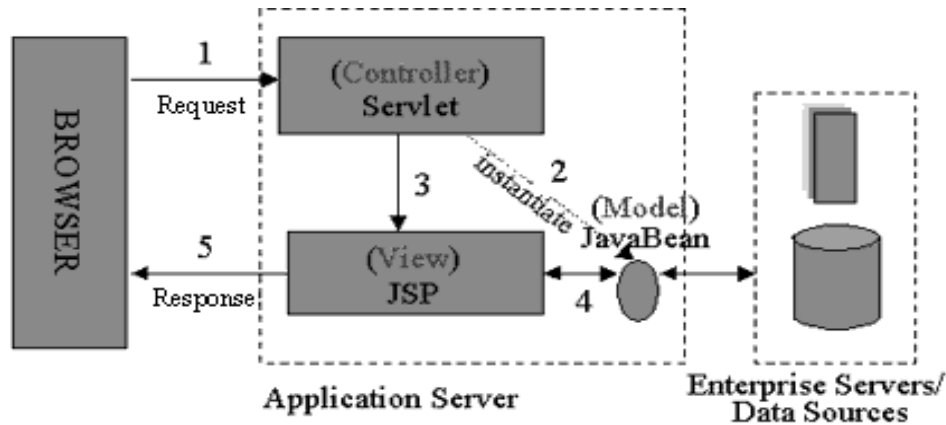
# Struts = MVC design = Model 2

iSeries AD, IBM Toronto



- the Struts “action” servlet acts as a common controller for the whole application
- Model includes Action classes, Form Beans
- The web.xml file contains: the definition of the Struts servlet named "ActionServlet", the URL mapping for the calls to this servlet (\*.do)

## JSP Model 2 architecture

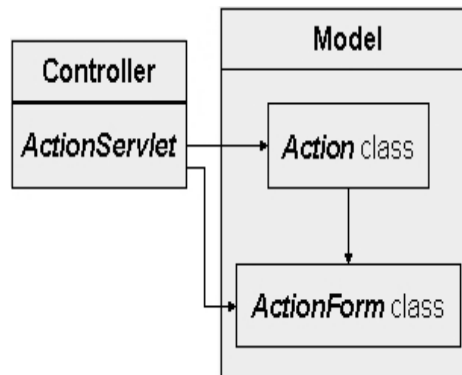


It takes advantage of the predominant strengths of both technologies, using JSP to generate the presentation layer and servlets to perform process-intensive tasks (controller). Here, the servlet acts as the *controller* and is in charge of the request processing and the creation of any beans or objects used by the JSP, as well as deciding, depending on the user's actions, which JSP page to forward the request to. Note particularly that there is no processing logic within the JSP page itself; it is simply responsible for retrieving any objects or beans that may have been previously created by the servlet, and extracting the dynamic content from that servlet for insertion within static templates. This approach typically results in the cleanest separation of presentation from content.

## What the “action” servlet does

iSeries AD, IBM Toronto

- a. Automatically transfers data from your form into a JavaBean (ActionForm)
- b. Calls the Action class you specify. This class may use the data in the ActionForm bean. When this class finishes it returns control to the ActionServlet, passing it a “forward”- as defined in the struts-config.xml- the next page to be displayed in the browser.



## Struts web.xml entries...

iSeries AD, IBM Toronto

```
<servlet-name>action</servlet-name>
<servlet-class> org.apache.struts.action.ActionServlet </servlet-class>
<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

- the Struts “action” servlet acts as a common controller for the whole application
- Model includes Action classes, Form Beans
- The web.xml file contains: the definition of the Struts servlet named "ActionServlet“, the URL mapping for the calls to this servlet (\*.do)

## struts-config.xml entries ...

iSeries AD, IBM Toronto

```
<form-beans>
  <form-bean name="submitForm"
    type="myPackage.SubmitForm"/>
</form-beans>
<action-mappings>
  <action path="/submit"
    type="myPackage.SubmitAction"
    name="submitForm"
    scope="request">
    <forward name="success" path="/submit.jsp"/>
    <forward name="failure" path="/submit.jsp"/>
  </action>
</action-mappings>
```

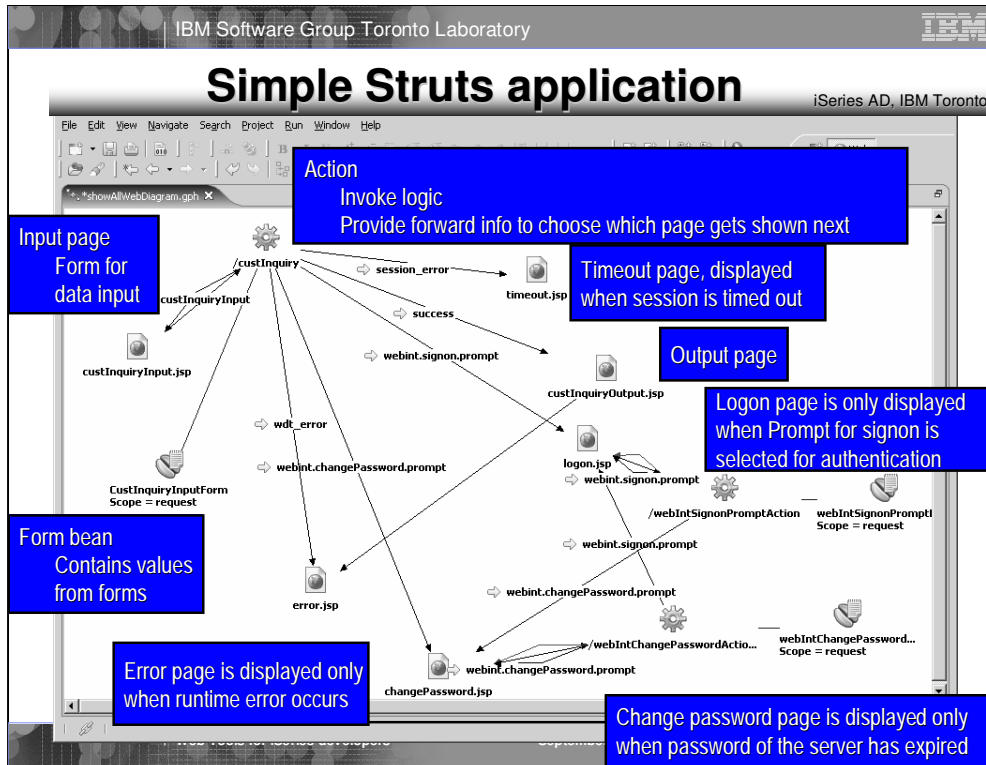
**path** - name of the request: "submit.do". You don't enter the ".do"-part here.

**type** - the path for the Action class file

**name** - is the logical name of the form bean (from the form-bean section)

**input** - validation errors should be shown on this page

**forward-** tag tells the "action" servlet where to go if it receives either "success" or "failure" from the Action class.



After Web Interaction wizard generates all the classes, SP files and connections, you can drop the initial JSP page (custInquiryInput.jsp) onto the web diagram and right click > Draw All, then the entire flow of the application is drawn for you. These information are based on struts-config.xml.



## More Information?

iSeries AD, IBM Toronto

### ► Information Sources

**[www-306.ibm.com/software/awdtools/series](http://www-306.ibm.com/software/awdtools/series)**

► For iSeries Application Development

**[www.eclipse.org](http://www.eclipse.org)**

► Eclipse and information about eclipse

**[www.ibm.com/software/info1/websphere/partners/series.jsp](http://www.ibm.com/software/info1/websphere/partners/series.jsp)**

► WebSphere on iSeries home page for BPs

**eServer magazine (iSeries edition)**

► Articles on WDS



# Disclaimer

iSeries AD, IBM Toronto

## Acknowledgement:

- This presentation is a collaborative effort of the IBM Toronto iSeries Application Development team.

## 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.





# Trademarks & Disclaimers

iSeries AD, IBM Toronto

© IBM Corporation 1994-2005. 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.



IBM Software Group

# WebSphere Development Studio Client for iSeries

## Web Tools for iSeries developers

| September, 2005 | WebSphere Development Studio Client V6.0

© 2005 IBM Corporation