

IBM



Web Tools for IBM System i5 developers

ibm.com
the power of one

IBM System i5 Technical conference 2006

Claus Weiss
weiss@ca.ibm.com

WebTools agenda

- ▶ WDS overview and AD roadmap 
- ▶ Website creation
- ▶ Webpage templates
- ▶ Designer for static WebPages
- ▶ Designer for Web objects
- ▶ Tool for Cascading Style Sheets
- ▶ Tools to create WebApplications
- ▶ Designer for dynamic WebPages
- ▶ Tools to create WebServices

WebSphere Development Studio

Current 5722-WDS customers with software subscription for V5R3, to upgrade to WDSc V6.0 use feature #: 2656
Available after GA

RPG	COBOL
C/C++	PDM SEU SDA RLU

Upgrade from WDS 6.0 to 6.0.1 using Rational Product updater

Unlimited Licenses

New WDS Lite
Technology preview

iSeries	iSeries	iSeries	iSeries	Web Facing WDHT support	iSeries Projects	+CODE +VisualAge RPG	
Java™	Debug	Struts Web	Web Service		RSE		
JSF	EGL Java generation	Trace	Profiling	DB	XML	App Server	HATS Toolkit

www.ibm.com/software/awdtools/iseries

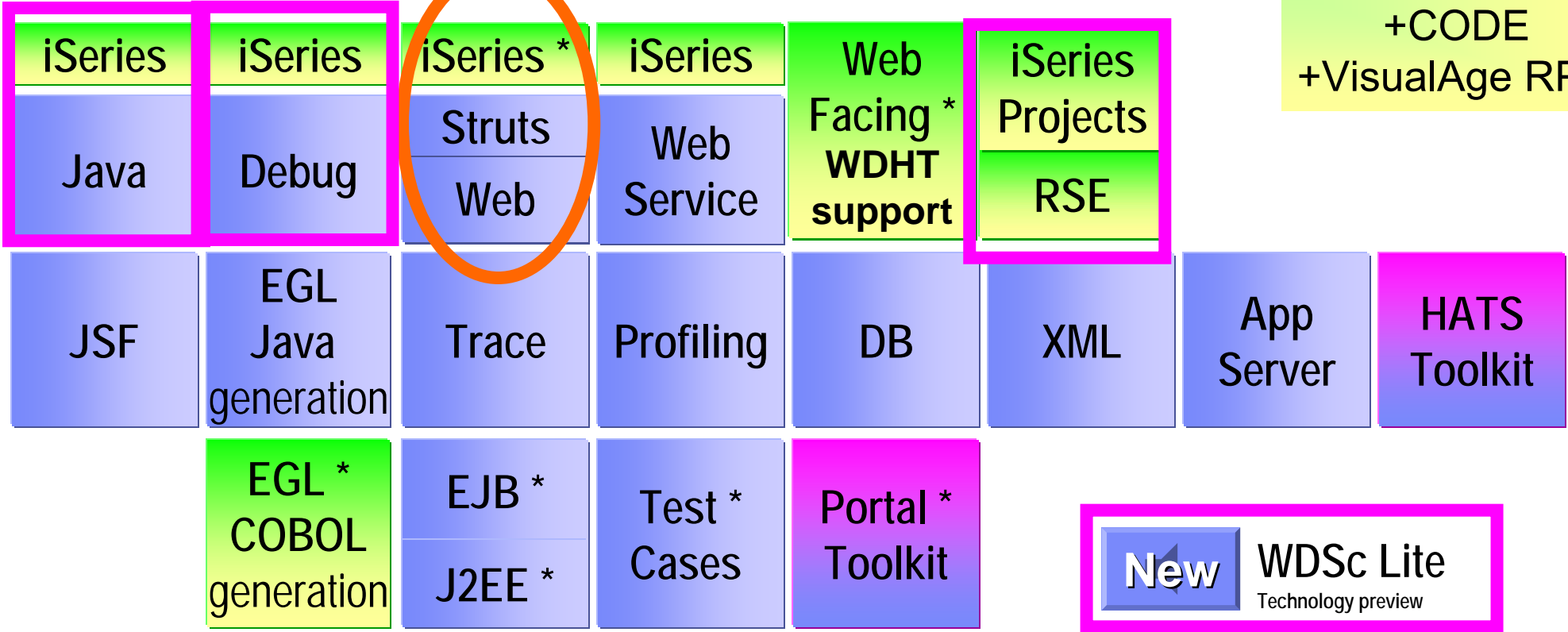
WebSphere Development Studio Client V6.0.1 based on RWD V6

WebSphere Development Studio Client Advanced Edition 6.0.1

*Workstation License
order through Passport Advantage
http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home*

Upgrade from WDSC 6.0 to 6.0.1 using Rational Product updater

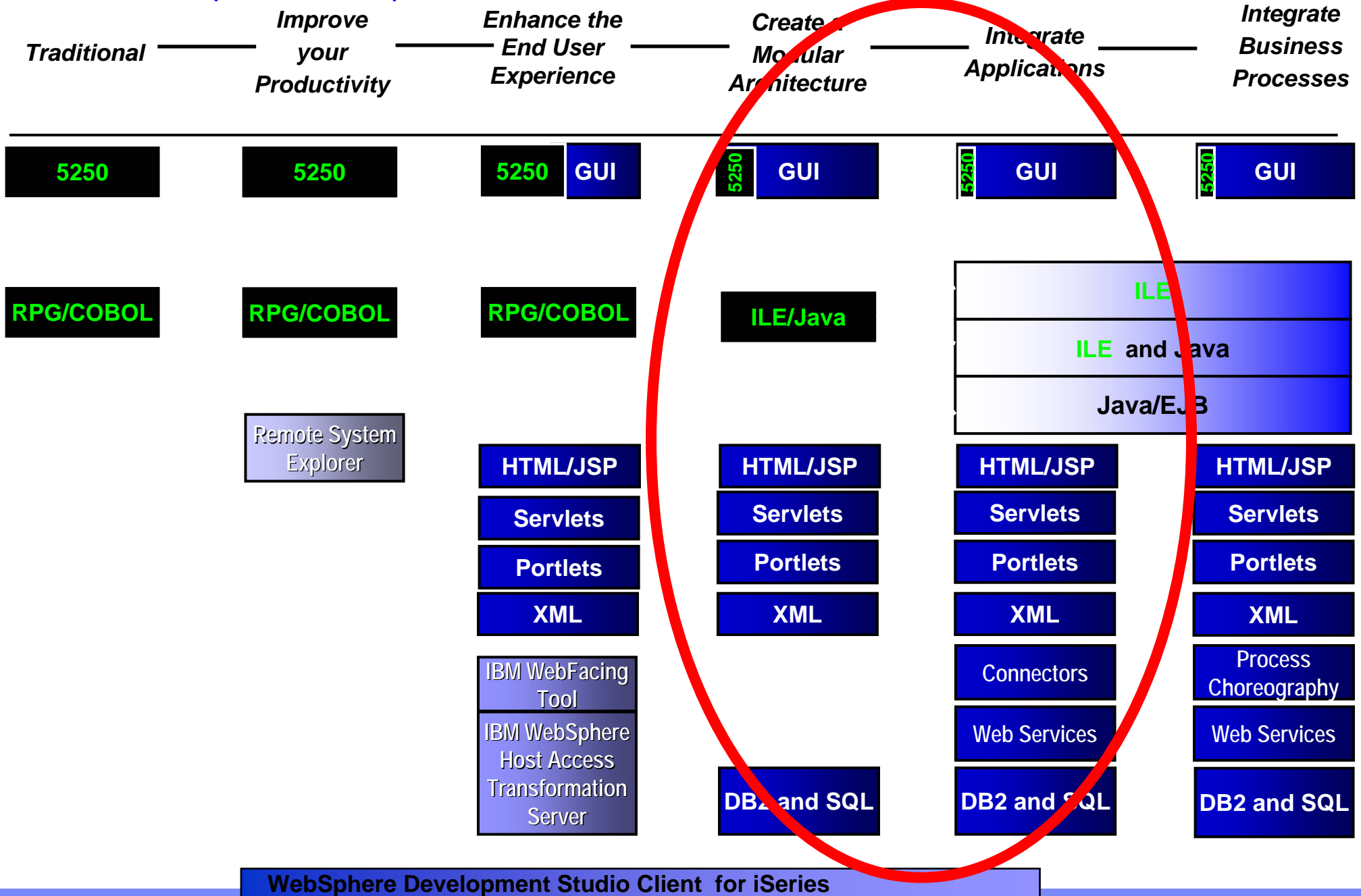
**+CODE
+VisualAge RPG**



www.ibm.com/software/awdtools/iseries

WebSphere Development Studio Client V6.0.1 based on RAD V6

iSeries Developer Roadmap - Architecture



WebSphere Development Studio Client for iSeries

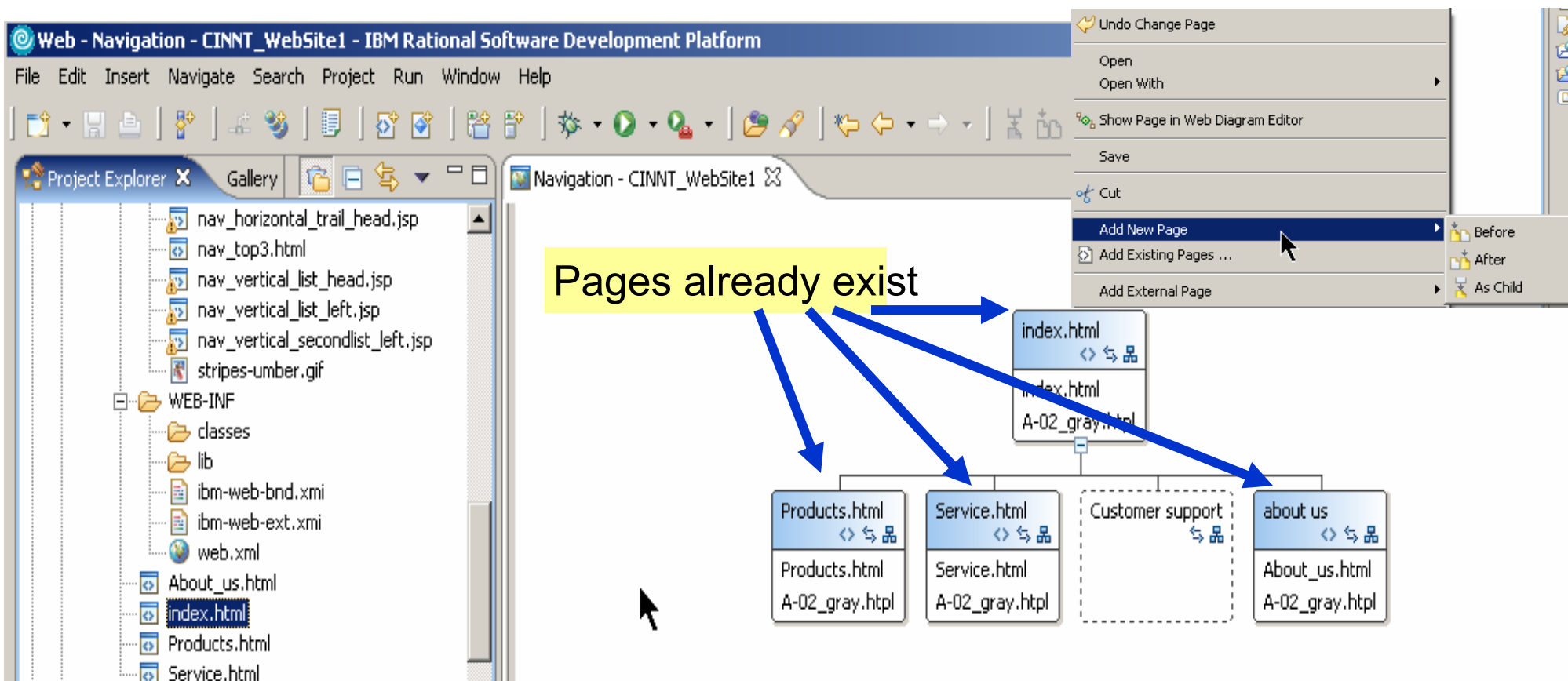
WebTools agenda

- ▶ WDS overview and AD roadmap
 - ▶ Website creation
 - ▶ Webpage templates
 - ▶ Designer for static WebPages
 - ▶ Designer for Web objects
 - ▶ Tool for Cascading Style Sheets
 - ▶ Tools to create WebApplications
 - ▶ Designer for dynamic WebPages
 - ▶ Tools to create WebServices
- 

WebSite designer

Design page relations

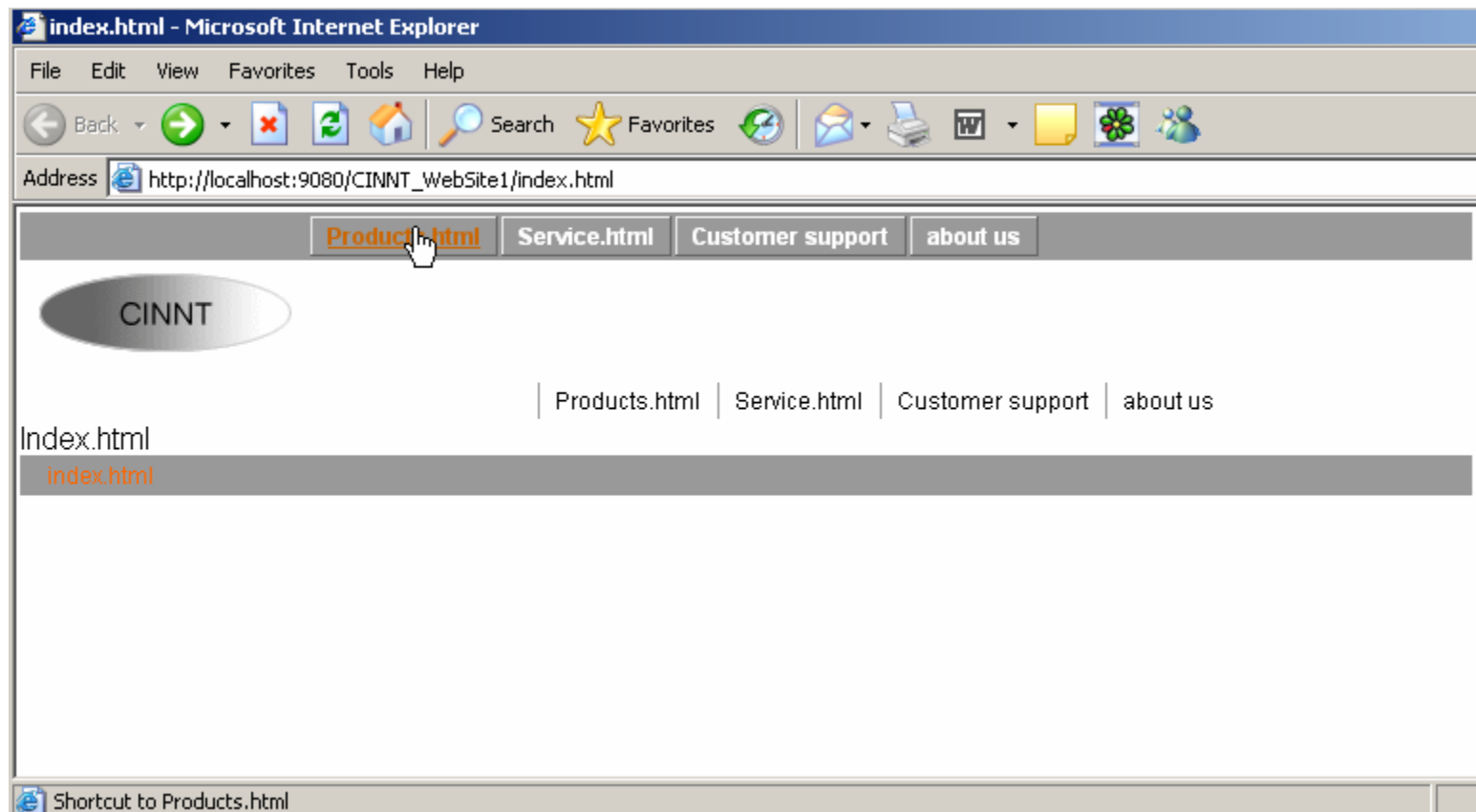
Bind to real Webpages later



WebPage templates

Template used with Navigation bar

Navigation bar automatically updated from WebSite designer



Create a new page template

New Page Template File

Specify a name and location for the new Page Template file

Folder: /CINNT_WebSite

File Name: C_Template

Markup Language: HTML

Options: Use XML Styl Create from

Model: Template contain
Generate a new

Configure advanced options

New Page Template File

Page Template File Selection

Select a page template to use when creating this file.

Page template type

Sample page template User-defined page template

Page Template File

Location: Browse... ▼

File name: JSP-A-03_gray.jtpl

Thumbnail:

JSP-C-02_blue.jtpl JSP-C-02_olive.jtpl JSP-C-02_umber.jtpl

< Back Next > Finish Cancel

Create new customized template

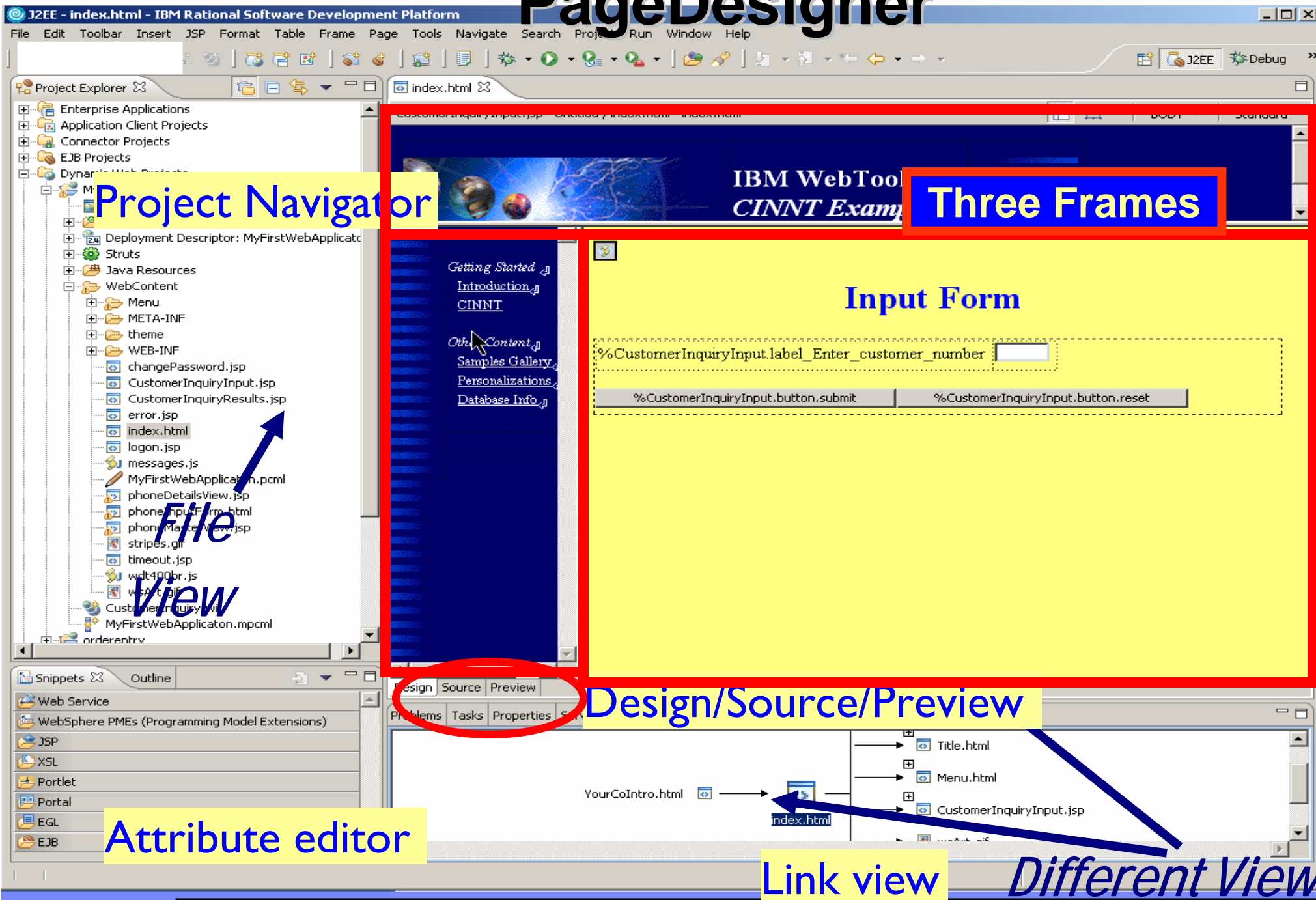
Based on sample template

Change page template

The screenshot shows the IBM Rational Software Development Platform interface. The main window displays a web page template with a yellow header and a content area. The text "Change to your companies needs" is highlighted in a yellow box. The interface includes a Project Explorer on the left, a Palette on the right, and a Properties panel at the bottom. The Properties panel shows the content area name as "bodyarea".

Change to your companies needs

PageDesigner



Frame tools

Set attributes of selected Frameset

Frame Attributes

Frame tree:

- FrameSet
 - Title.html
 - FrameSet
 - Menu.html
 - YourCoIntro.html

Size - cols:

Value	Unit
160	pixels
1	*

Border

Show border:

Border width: 1

Set attributes of selected frame

Frame Attributes

Frame tree:

- FrameSet
 - Title.html
 - FrameSet
 - Menu.html
 - YourCoIntro.html

URL: Menu/Menu.html

Frame name: menu

No resize on border

Scrollbar: (Auto)

Horizontal margin: 0 pixels

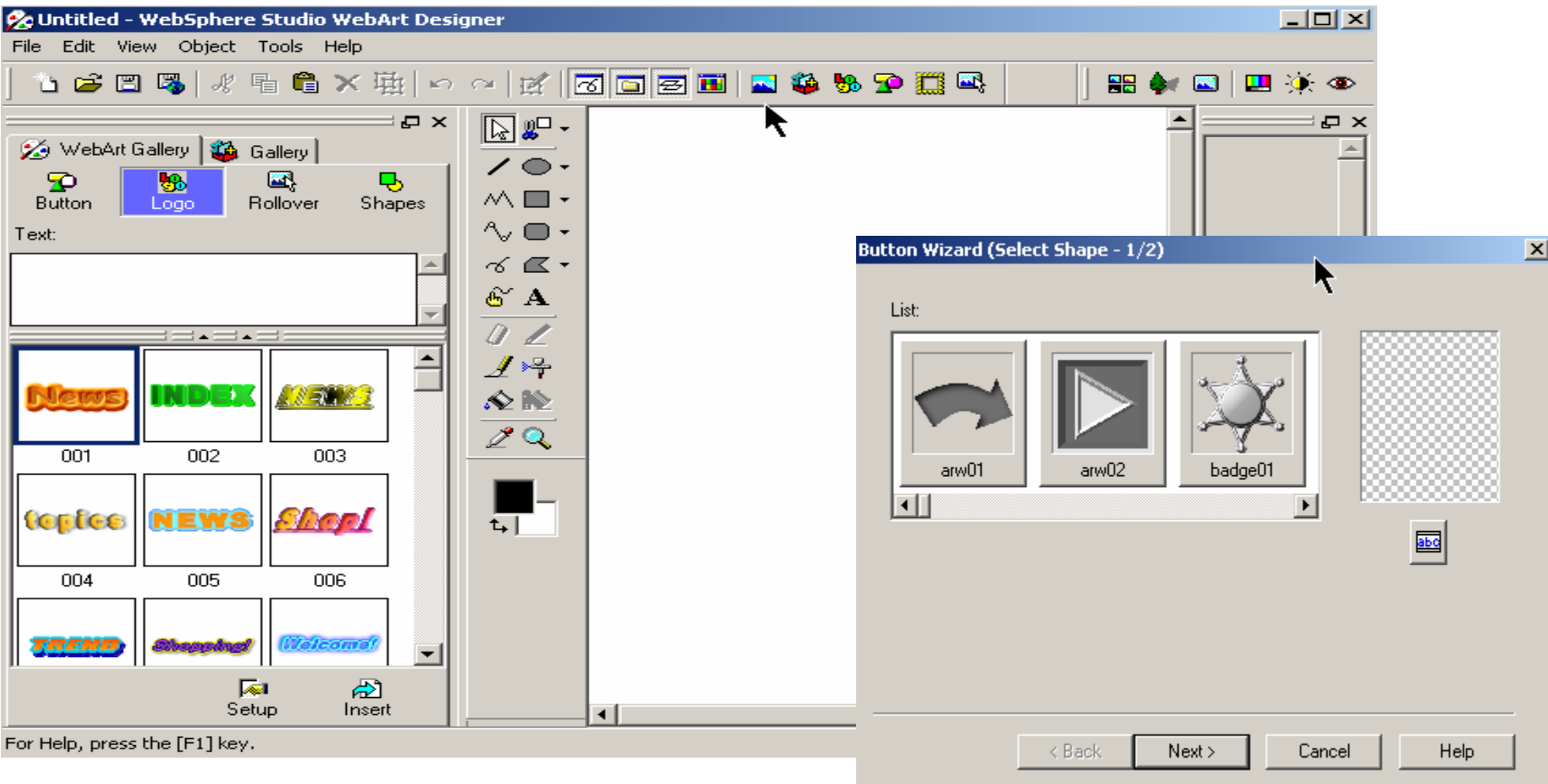
Vertical margin: 0 pixels

Apply

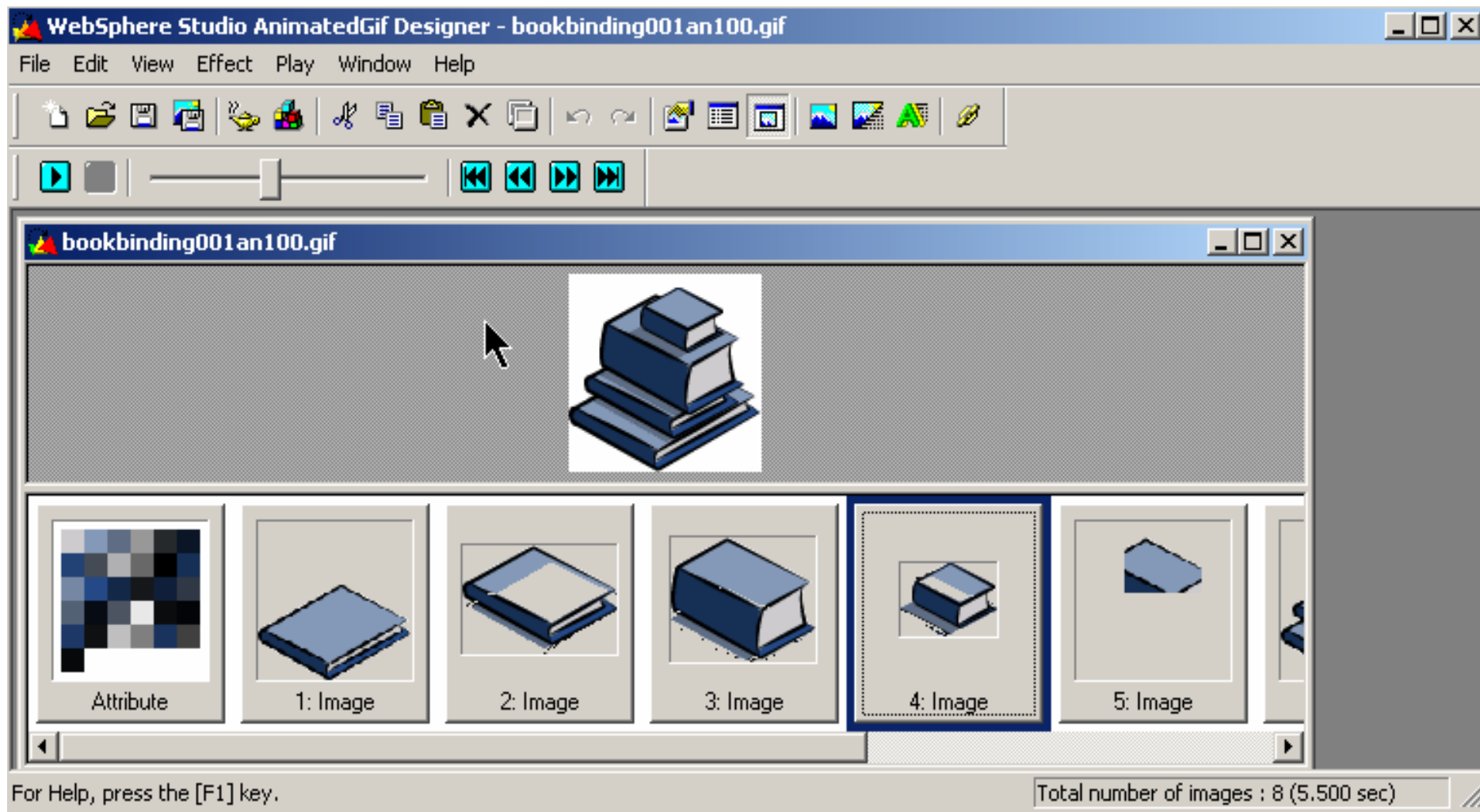
OK Cancel

WebArt Designer

Wizards for Logo/Button/PhotoFrame/Rollover



Animation GIF designer



Running the application

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "index.html - Microsoft Internet Explorer". The address bar contains "http://localhost:9080/MyFirstWebApplication/index.html". The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The toolbar features Back, Forward, Stop, Refresh, Home, Search, Favorites, Media, History, Print, Stop, and Go buttons. The main content area displays the "IBM WebTooling CINNT Example" application. The page has a dark blue header with a globe image and the text "IBM WebTooling CINNT Example". Below the header is a yellow section titled "Customer details" in blue text. The details include: Customer number: 0010100, Customer name: Meridien Electronics Limited, Rep number: 43443, Contact name: Alfredo Bayonne, Customer phone: 206-865-4027, Customer fax: 206-865-4037, Country: U.S.A., Zip code: 98007, and Zip code location: 1. At the bottom of the details section are "Submit" and "Reset" buttons. On the left side of the page, there is a navigation menu with links for "Getting Started" (Introduction, CINNT) and "Other Content" (Samples Gallery, Personalizations, Database Info). The status bar at the bottom shows "Done" and "Local intranet".

index.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media History Print Stop Go Links >>

Address <http://localhost:9080/MyFirstWebApplication/index.html>

IBM WebTooling
CINNT Example

Customer details

Customer number: 0010100
Customer name: Meridien Electronics Limited
Rep number: 43443
Contact name: Alfredo Bayonne
Customer phone: 206-865-4027
Customer fax: 206-865-4037
Country: U.S.A.
Zip code: 98007
Zip code location:

Submit Reset

Getting Started
[Introduction](#)
[CINNT](#)

Other Content
[Samples Gallery](#)
[Personalizations](#)
[Database Info](#)

Done Local intranet

Stylesheet editor

Instant feedback how it will look on Webpage

The screenshot displays the IBM Rational Software Development Platform's Stylesheet Editor. The interface is divided into several panes:

- Project Explorer:** Shows the project structure for 'orlando02', including folders like 'WebContent', 'META-INF', and 'WEB-INF', and various JSP files.
- Stylesheet Editor:** The main area showing the CSS code for 'Master.css'. It includes a 'Selected Style' section with a preview of 'Style Of H2' and a 'Standard HTML Elements' section with preview text for H1 through H6. A red circle highlights the H2 style rule in the code editor, which is defined as:


```

      H2
      {
      COLOR: #6666CC;
      FONT-FAMILY: 'Times New Roman';
      TEXT-TRANSFORM: capitalize;
      }
      
```
- Page Data:** A table at the bottom left showing the mapping of CSS classes to HTML elements:

Name	Description	M...
BODY	Page properties	
H1	Heading 1	
H2	Heading 2	
H3	Heading 3	
H4	Heading 4	
- Toolbar:** Located at the bottom, it contains buttons for 'Add...', 'Edit...', 'Save', and 'Format'. A red circle highlights the 'Add...' button.

A context menu is open over the H2 style rule, showing options like 'Undo Add style rule', 'Revert File', 'Cut', 'Copy', 'Paste', 'Cleanup Document...', 'Format', 'Run to Line', 'Preferences...', 'Add to Snippets...', 'Properties', and 'Save'.

Creating your own style classes

The screenshot illustrates the process of creating a new style class in the IBM WebSphere IDE. The main workspace shows a preview of the selected style, which is a blue-bordered box containing the text "Style of .myclass". The CSS code for this style is visible in the editor on the right, showing properties like color, font-family, and text-transform.

Two dialog boxes are open in the foreground:

- Set Selector of New Style:** This dialog is used to define the class name. The "Class" radio button is selected, and the class name ".myclass" is entered in the "Class name" field.
- Add Style:** This dialog is used to define the style properties. The "Font" category is selected, and the "Font" sub-category is expanded. The "Font Family" is set to "Comic Sans MS", the "Color" is set to "Blue", and the "Size" is set to "36". The "Sample" field shows the text "AaBbYyZz" in the selected font and color.

The IDE interface also shows the Project Explorer on the left, displaying the project structure for "orlando02", and a menu on the right with the "Add..." option circled in red.

WebTools agenda

- ▶ WDS overview and AD roadmap
- ▶ Website creation
- ▶ Webpage templates
- ▶ Designer for static WebPages
- ▶ Designer for Web objects
- ▶ Tool for Cascading Style Sheets
- ▶ Tools to create WebApplications 
 - ▶ Designer for dynamic WebPages
- ▶ Tools to create WebServices

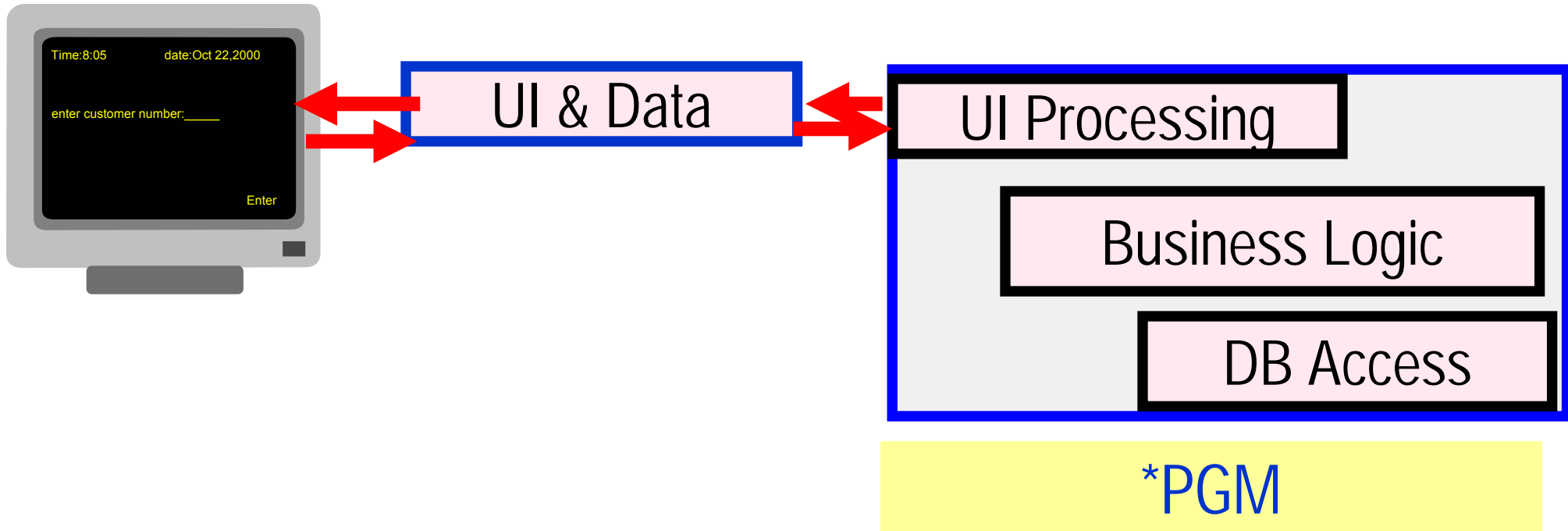
Web applications with dynamic webpages

- **Content of Webpages gets assembled at runtime**
 - 1. Applets (application running in a browser)**
 - 2. generate HTML at runtime on the fly**
 - 3. Use predefined HTML with leaving holes for runtime information**
 - **Substitution variables**
 - **Java Server Page (jsp) standard way in J2EE (Java code on server fills holes)**
 - **Jsp with Struts framework (controller logic strictly separated)**
 - **Java Server Face (jsf) framework that also addresses UI)**



Today's Model

5250 Screens



- Program puts up screen, waits for input
- Program processes input, does business logic

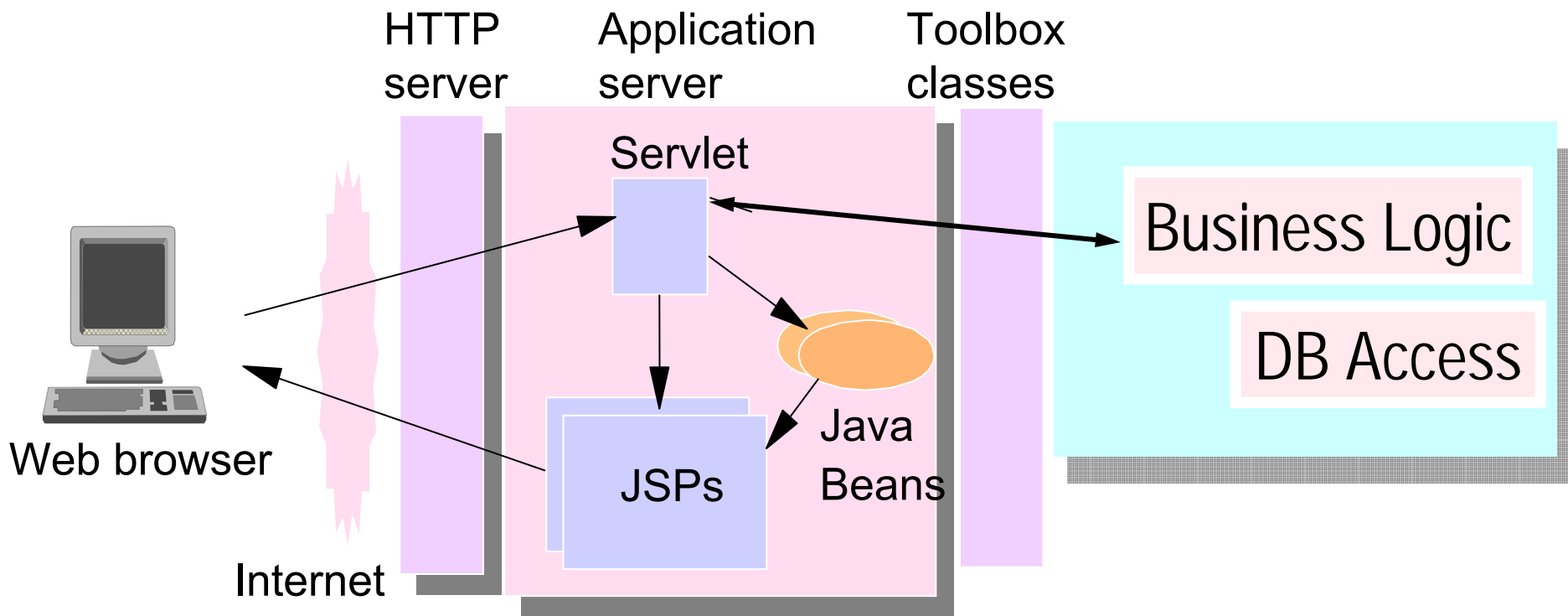


Modern Web App Architecture

Tier 1

Tier 2

Tier 3

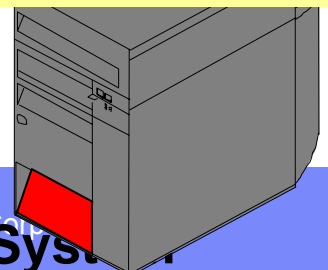


UI & Data

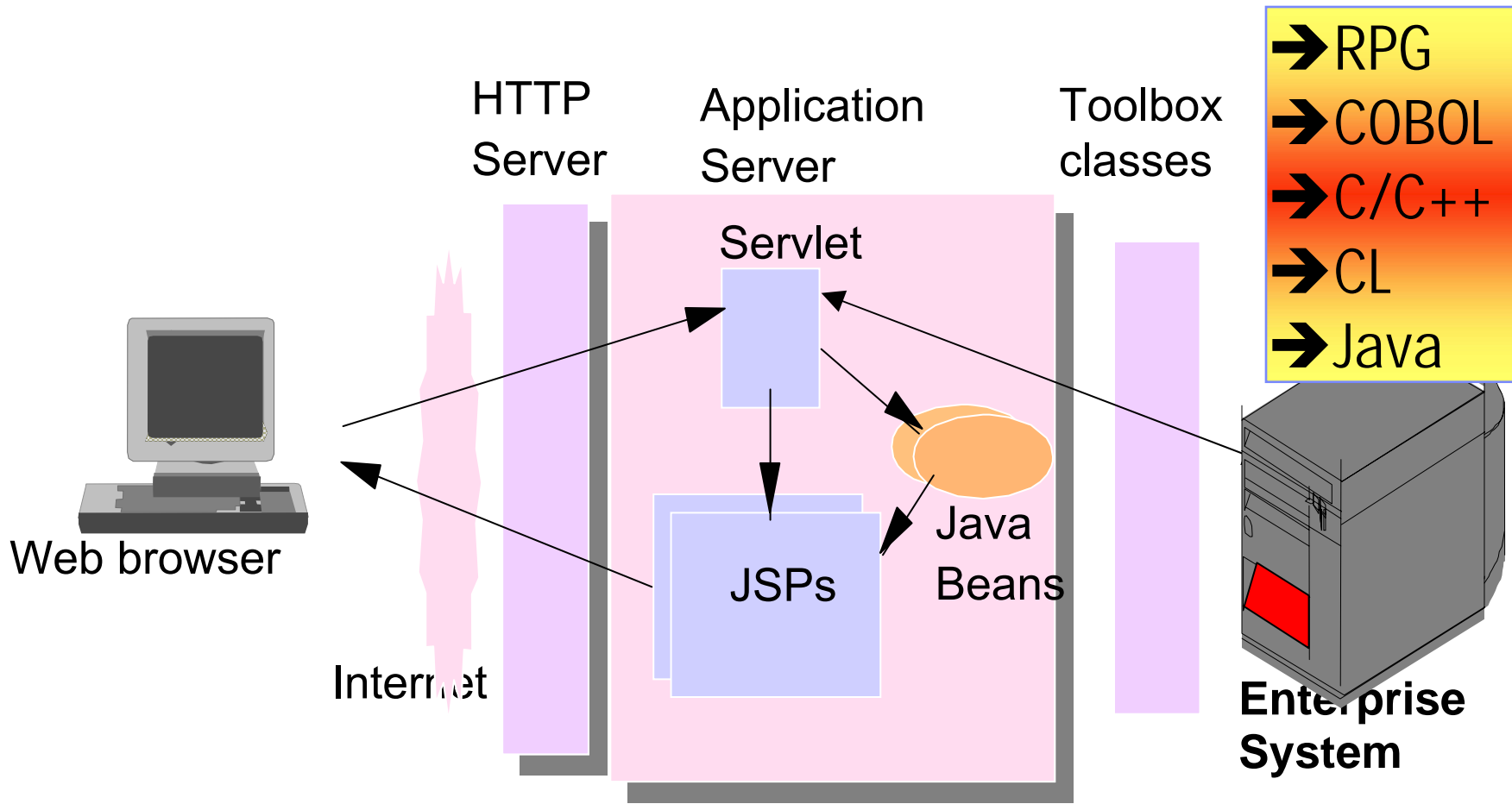
UI Processing

*PGM

Sample J2EE web application



Tooling for modern web applications



- WDS** → iSeries webtooling
- Pagedesigner
 - Jsf tooling
 - Interaction wizard
 - WebServices wizard

- iSeries 3 GL tooling
- Remote system explorer
 - LPEX source editor
 - Integrated Debugger

Calling native programs from Java

- A brief look at: **IBM System i5 Java toolbox**
- Java toolbox is included in WDS*c*
- Lets look at: **Using Program Call Markup Language (PCML)**

ILE RPG and COBOL compilers can create PCML for you

→ OS/400 V5R2 and higher:

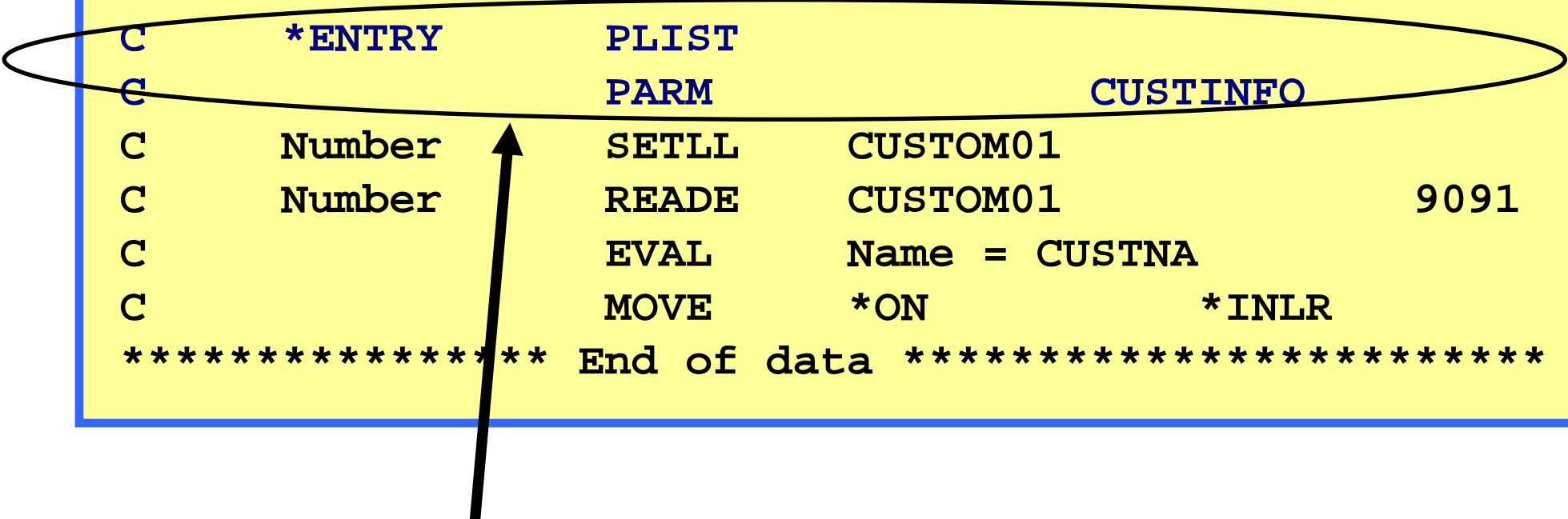
-New parameters in CRTxxxMOD and CRTBNDxxx

The screenshot shows the 'Create Bound RPG Program (CRTBNDRPG)' dialog box. The 'Create program interface' field is highlighted with an orange oval and contains '*PCML'. The 'Program interface stream file' field contains 'payroll.pcm|'. Other fields include 'Language identifier' (*JOBRUN), 'Replace program' (> *YES), 'User profile' (*USER), 'Authority' (*LIBCRTAUT), 'Truncate numeric' (*YES), 'Fix numeric' (empty), 'Target release' (*CURRENT), 'Allow null values' (*NO), 'Define condition names' (empty), 'Enable performance collection' (*PEP), 'Profiling data' (*NOCOL), 'Licensed Internal Code options' (empty), 'Include directory' (empty), 'Preprocessor options' (empty), 'Output source file' (empty), 'Library' (*CURLIB), 'Output source member' (*PGM), and 'Output stream file' (empty). The 'Advanced(5)' checkbox is unchecked, 'All Parameters(6)' is checked, and 'Keywords(7)' is unchecked. The bottom of the dialog shows the command: CRTBNDRPG PGM(RSELABXX/PAYROLL) SRCFILE(RSELABXX/QRPGLESRC) SRCMBR(PAYROLL) OPTION(*EVENTF) DBGVIEW(*SOURCE) REPLACE(*YES) PGMINFO(*PCML) INFOSTMF(payroll.pcm|)

Java Calling RPG

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.

Use PCML to describe program interface

RPG/COBOL compilers create PCML

No need for you to write PCML

```
<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>R
```

Program Call Markup Language (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.gotback.Name");
                System.out.println("Customer name: " + value);
            }
    } catch (PcmlException exc) {
        System.out.println("*** Call to getcust failed. ***");
        System.exit(0);
    }
    System.exit(0);
} // end main method
```

Using PCML to invoke native system I programs

1. First lets look at WebProjects
2. Interaction wizard
3. Jsf tooling
4. Web Services tooling

J2EE Web Application

▶ Standard Web App folder structure:

+Web application folder (root folder)

+**source**

–*all non-deployed files (java)*

+**WebContent**

–*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 application deployment descriptor:

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

+**WEB-INF**

–web.xml

+**classes**

–*Java classes of this app (usually generated)*

+**lib**

–*Supporting classes and jar files*

J2EE
1.3

Terms: Web Application

▶ Example Web Application

+accounts

+**source**

+**WebContent**

-index.html

www.mydomain.com/accounts

+receivable

-page1.html

www.mydomain.com/accounts/receivable/page1.html

+payable

-page1.html

www.mydomain.com/accounts/payable/page1.html

+**META-INF**

-MANIFEST.MF

+**theme**

-corporate.css

+**WEB-INF**

-web.xml

+**classes**

▶ Web Archive Files (WAR)

J2EE
1.3

- **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)
+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

▶ Enterprise Archive Files (EAR)

J2EE
EJB
Spec

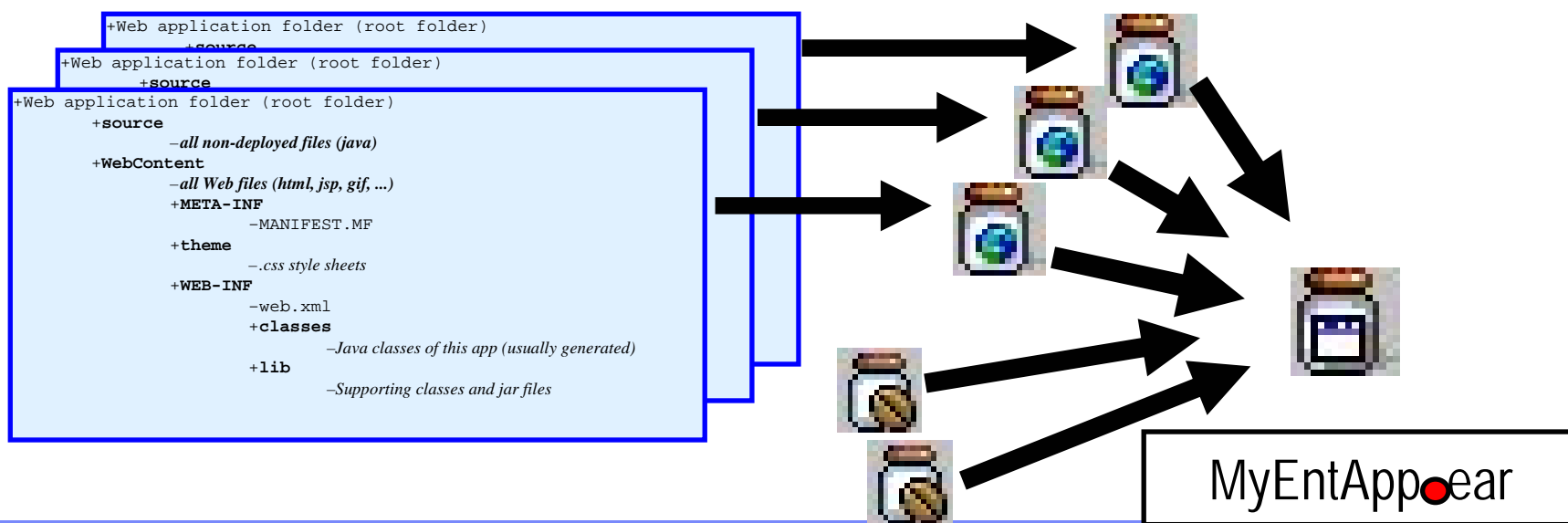
- **One file containing:**



- ▶ Zero or more Web Archive (war) files
- ▶ Zero or more EJB jar files
- ▶ A J2EE deployment descriptor

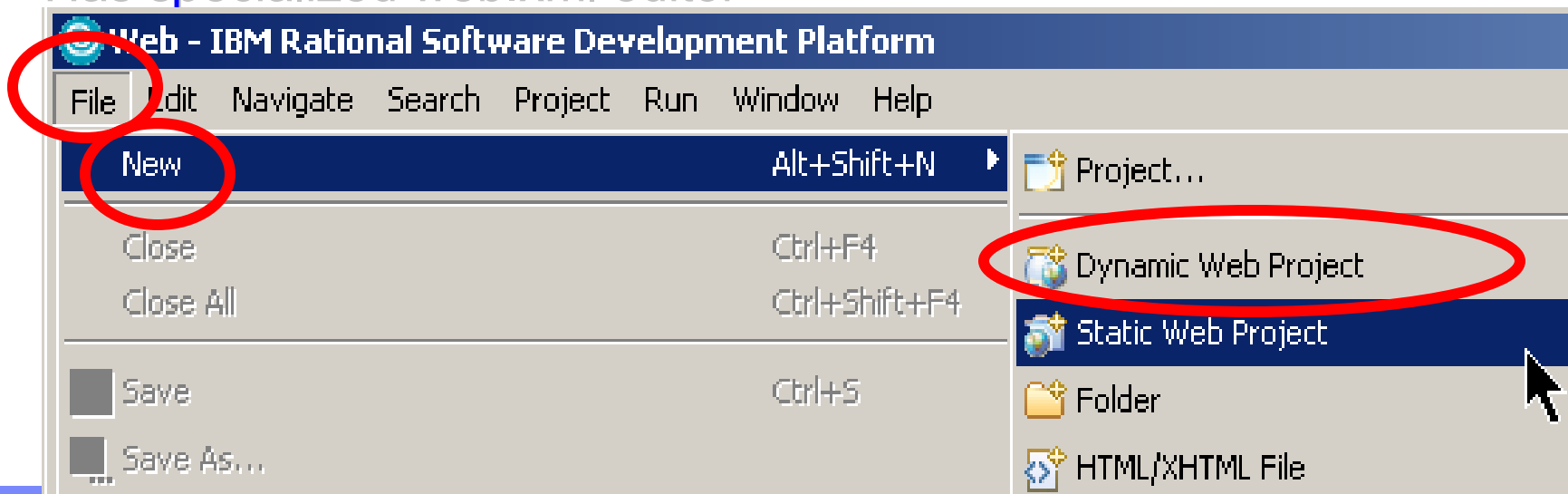
- **Used to install and configure:**

- ▶ All pieces of a J2EE Enterprise Application
 - ✓ Web application plus EJBs plus EJB clients
- ▶ All Web applications for a Web site (say)



Web Projects

- **Create new web applications in a web project**
- **One of the project types in WDS**
 - ▶ With its own web perspective
 - ▶ With its own "new" wizard
 - ▶ With its own set of tools
- **Created with J2EE folder layout**
- **Created with simple web.xml file**
 - ▶ Automatically updated as resources are created
 - ▶ Has specialized web.xml editor



Web projects

New Dynamic Web Project wizard

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

Name:

Project location:

Servlet version:

Target server:

Add module to an EAR project.

EAR project:

Context Root:

Add support for annotations

Functionality for the Web Project.

...this feature to have a CSS file generated for the project

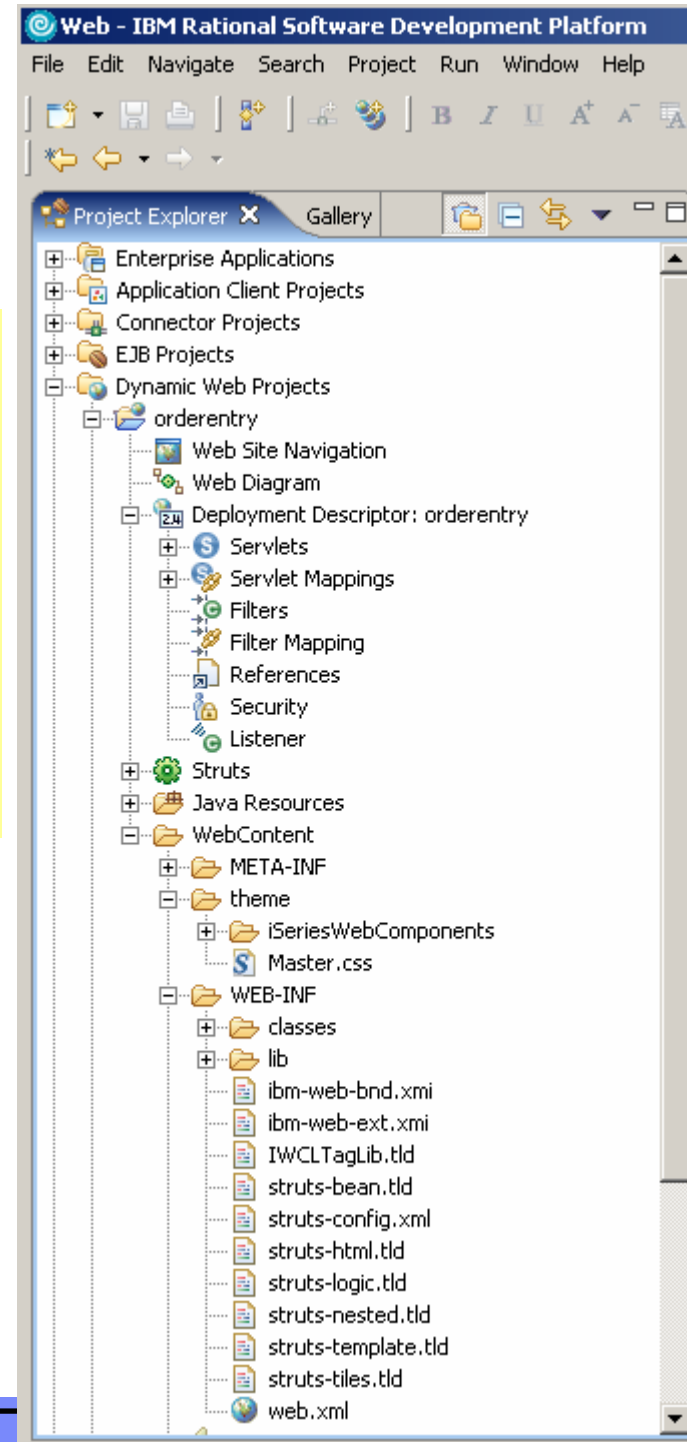
Callout 1: Automatically associates with a supplied EAR file for easy auto-cfg of Application Server

Callout 2: Select features being used in this web application

Callout 3: Select template for web pages

Web projects

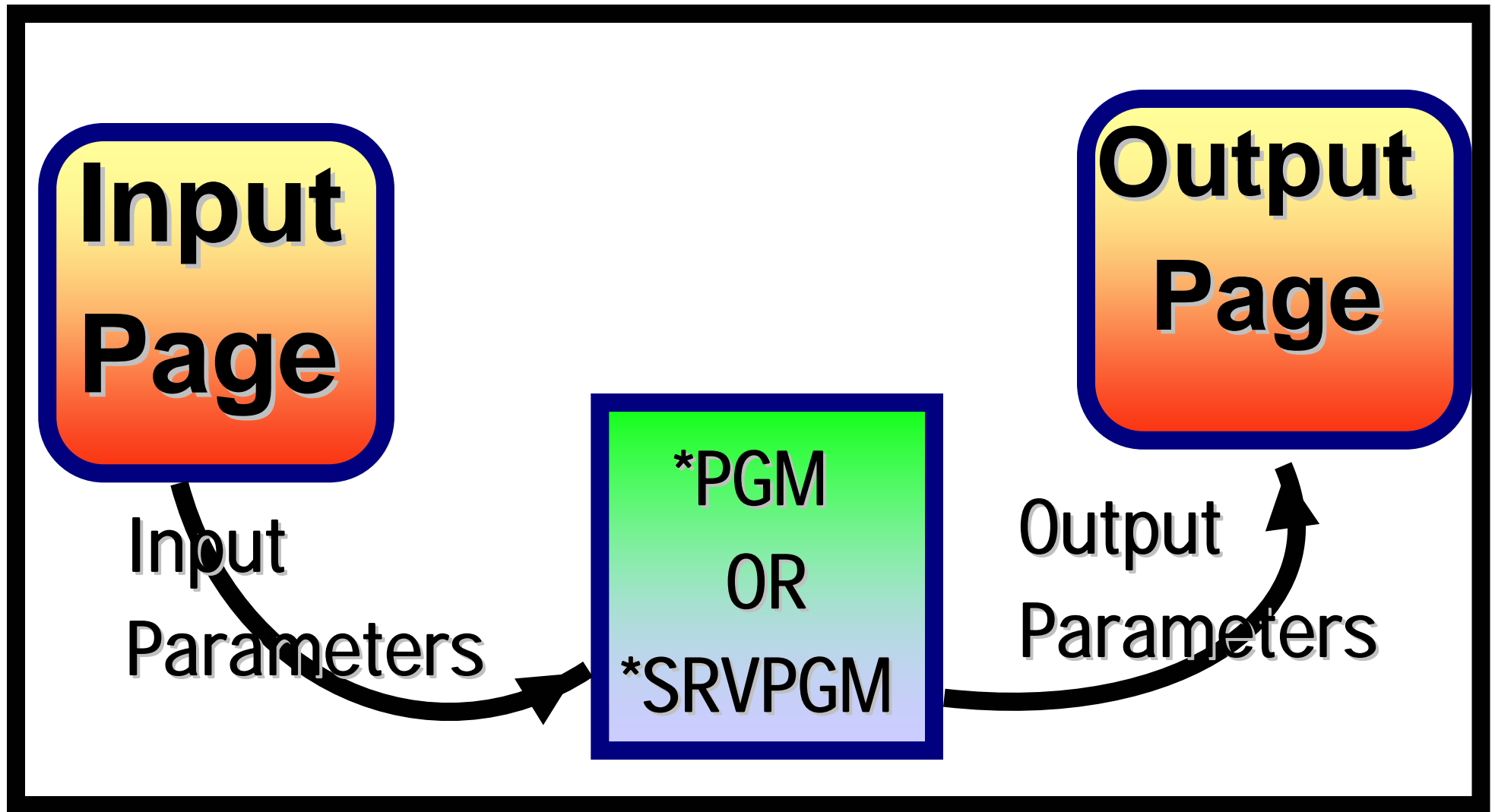
Web application structure is created from information provided in New Dynamic Web Project Wizard



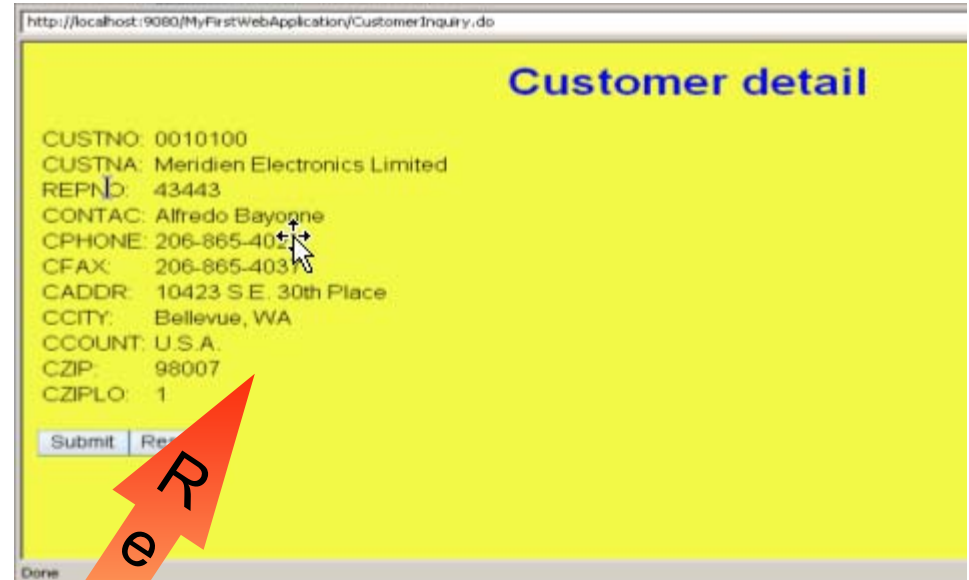
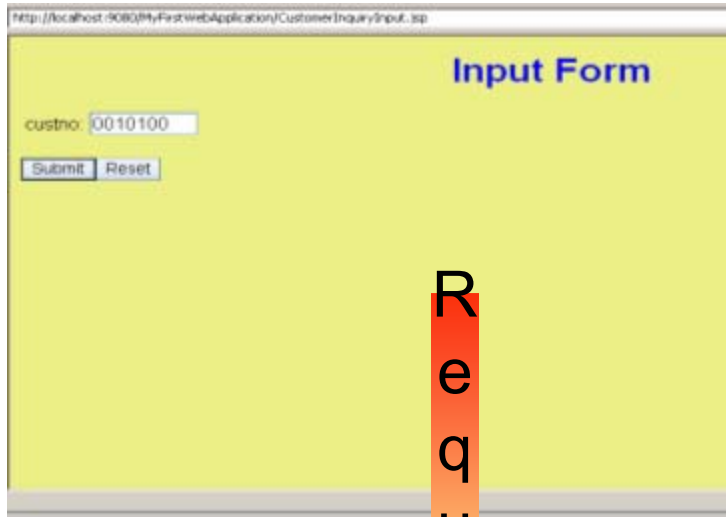
Web Tool for iSeries

- ✚ Wizards to help create iSeries RPG/COBOL web applications
 - Interaction wizard to build web application
Struts based
 - JSF tooling
 - WebService wizard to build an iSeries WebService
 - JCA wizard to build an iSeries Java Connector Architecture (JCA) connector

An interaction

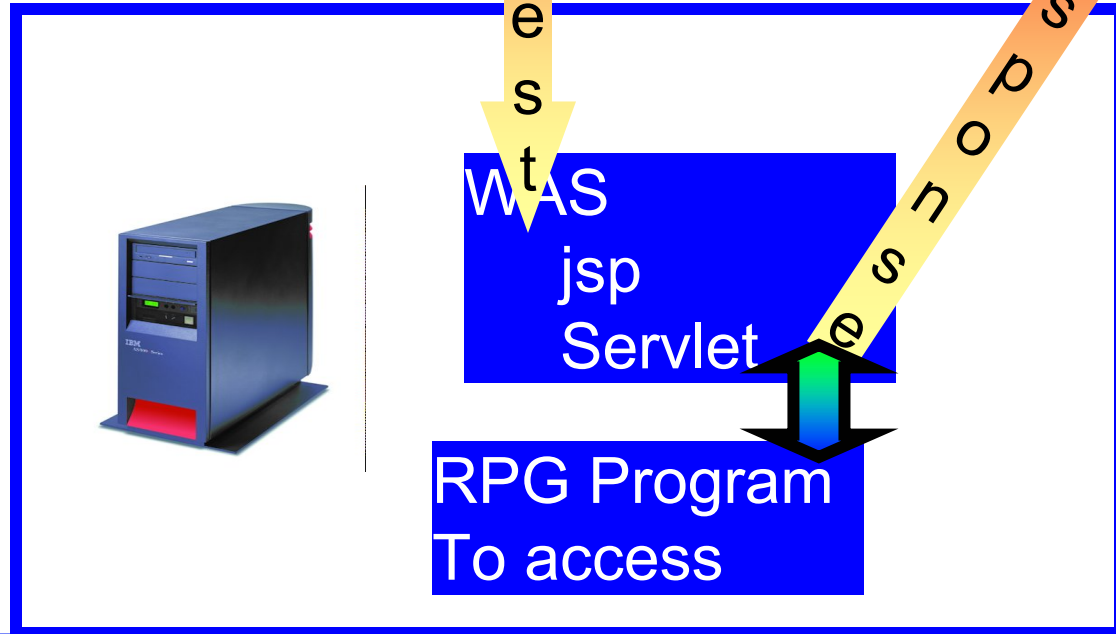


Build this simple interaction



Request

Response



data

Using the Interaction wizard

- ✚ Interaction wizard specific for iSeries development (not extendable)
- ✚ Steps involved
 - ✚ Create a dynamic web project
 - ✚ Create Runtime configuration
 - iSeries server
 - User id/password
 - Library list setup
 - ✚ Create Interaction
 - ✚ Input screen
 - ✚ Program interface (PCML)
 - ✚ Output screen

New standard framework java server faces (jsf)

- **Wait something better than STRUTS is here**
- A new standard framework fully supported in WDS*c*
 - Tool support for jsf
 - Page designer
 - Binding support of data beans and UI controls
 - Web diagram editor for jsf
 - Quick edit for small pieces of Java code

Java Server Faces (jsf)

- A specification and reference implementation for web applications
 - Components
 - Events
 - Validators & converters
 - Navigation
 - Back-end-data integration

- Standard

- Allows for extended tool support

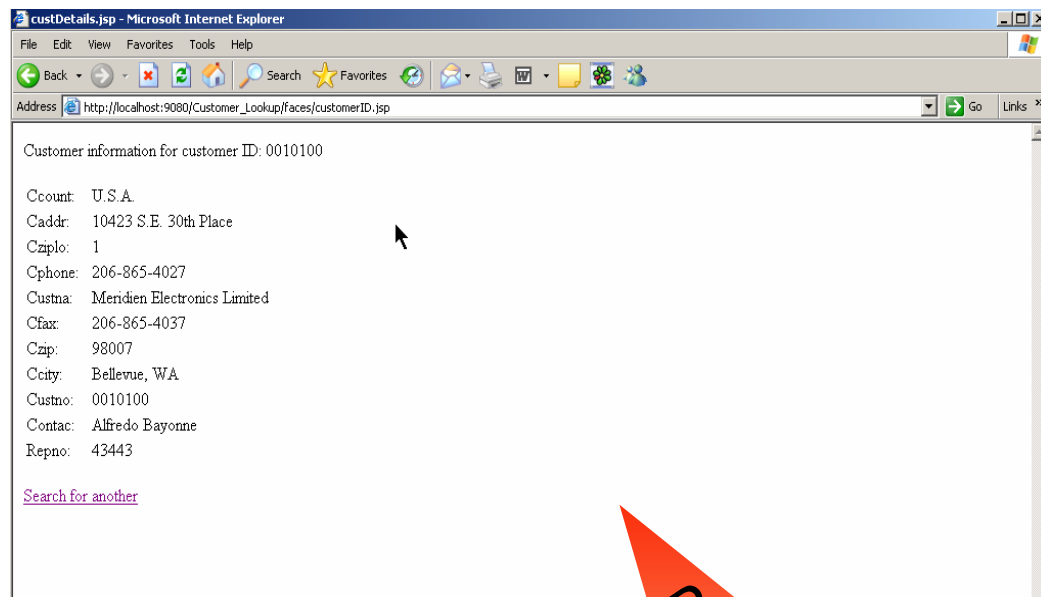
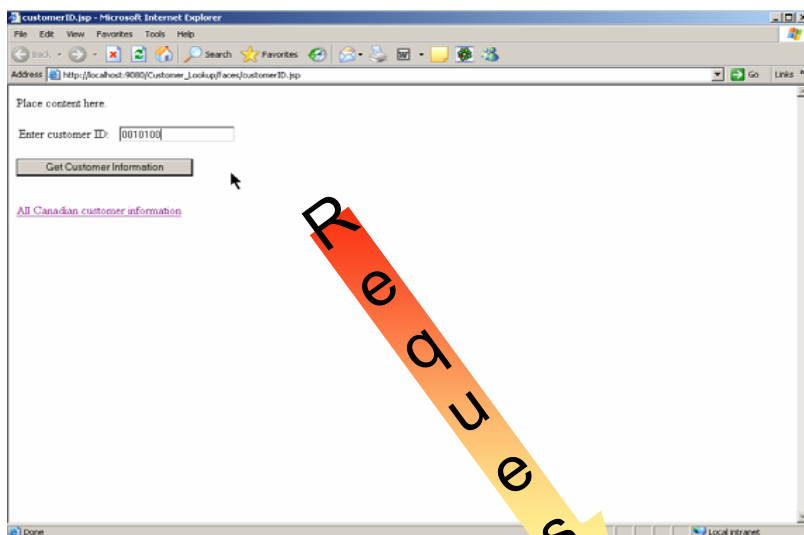
Web Page Development

- JSF contains a collection of UI components to make Web page development easier
 - Each component has extensive properties for customizing look and behavior of component
 - Components can be bound to variables for automatically:
 - Displaying the value of variable
 - Assigning value to a variable on page submit
 - Components have associated “events” which you can write code for
 - JavaScript for client side events
 - onclick, ondblclick, onkeypress, ...
 - Java code for server side events
 - Value Changed, Command, ...

Creating a jsf based web application

- Using
 - Web diagram editor
 - jsf controls and associated tools
 - Program call wizard
 - Jsf Data tools

Build this jsf web application



Request

Response

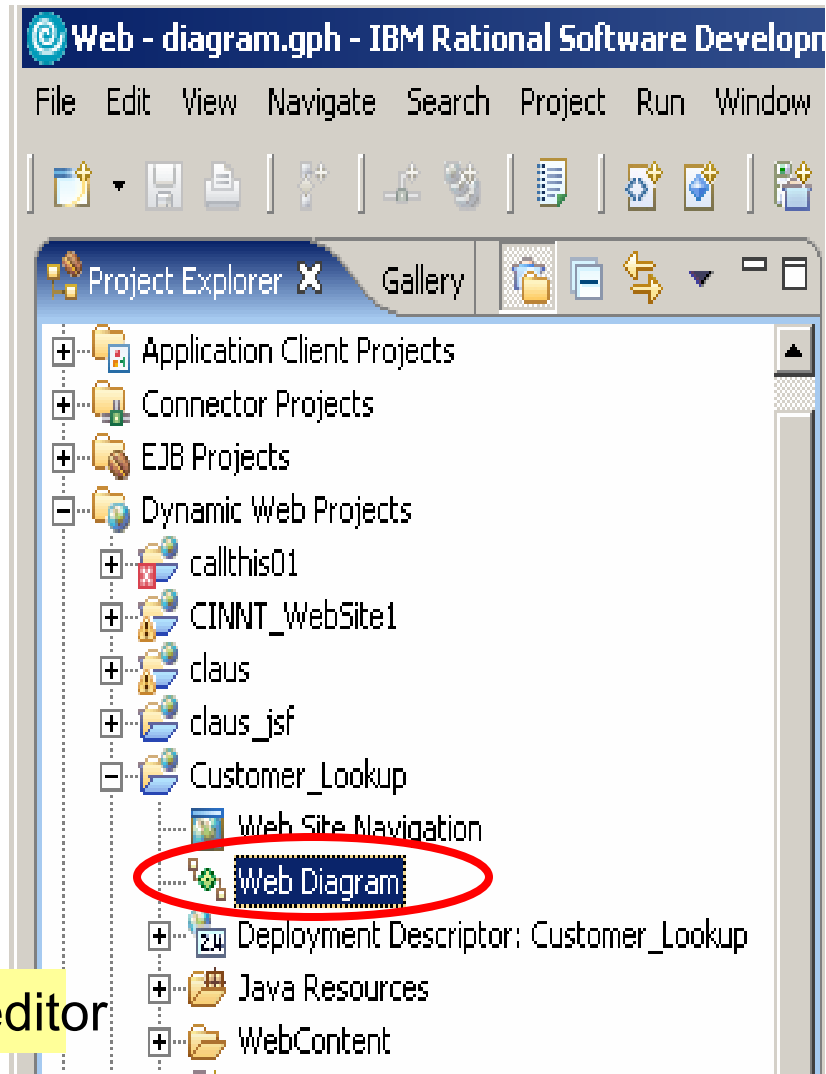
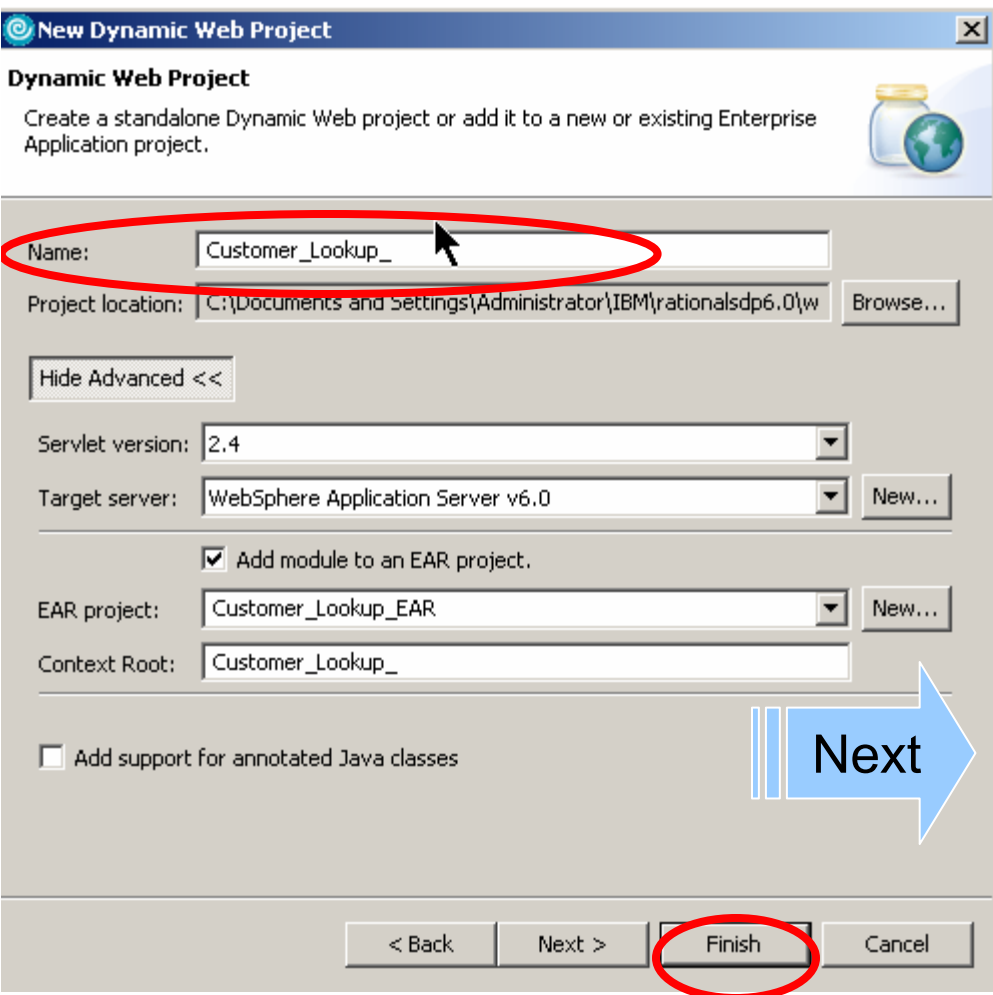


RPG Program
To access

jsf
Validation
Conversion and put values from UI control
into Program call bean
Use Program call bean to invoke program
Get return values from Program call bean
into bound UI controls

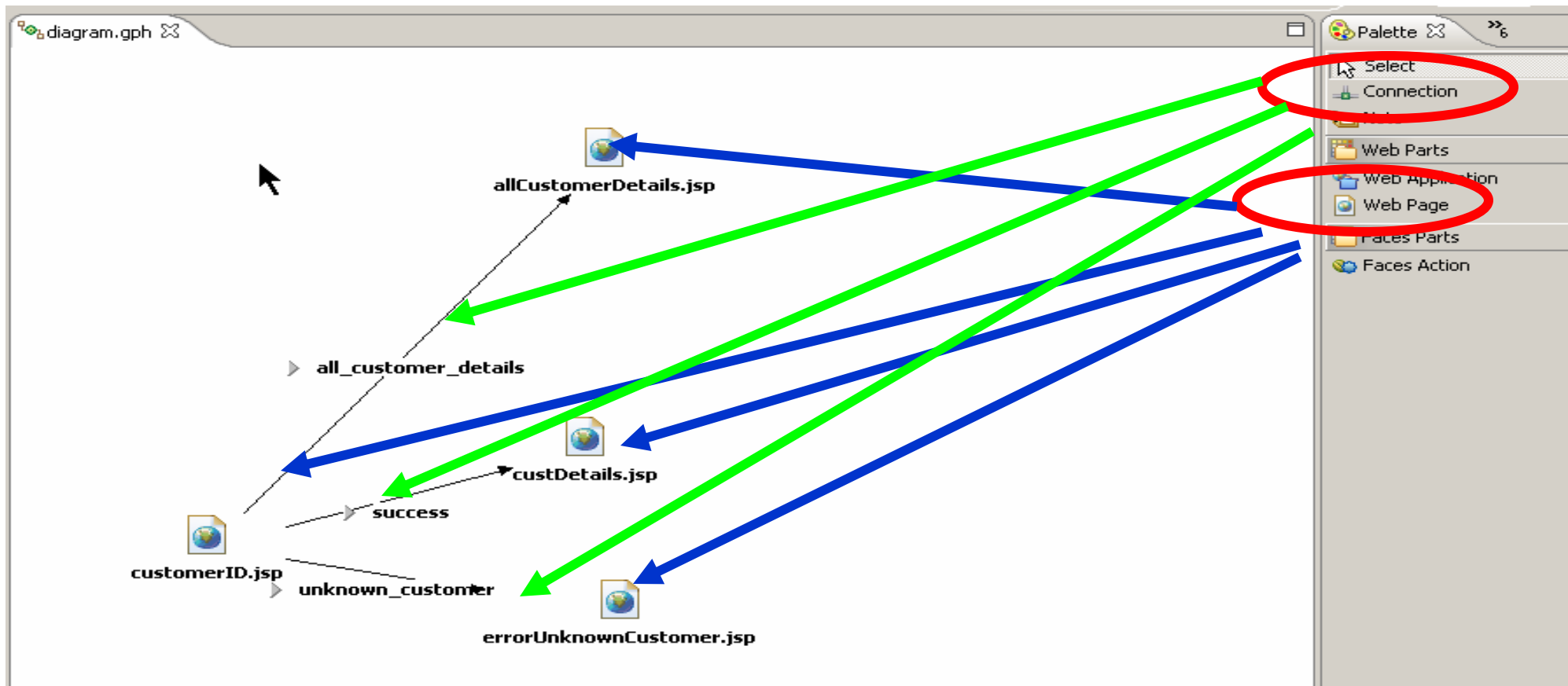
data

Create a Web project



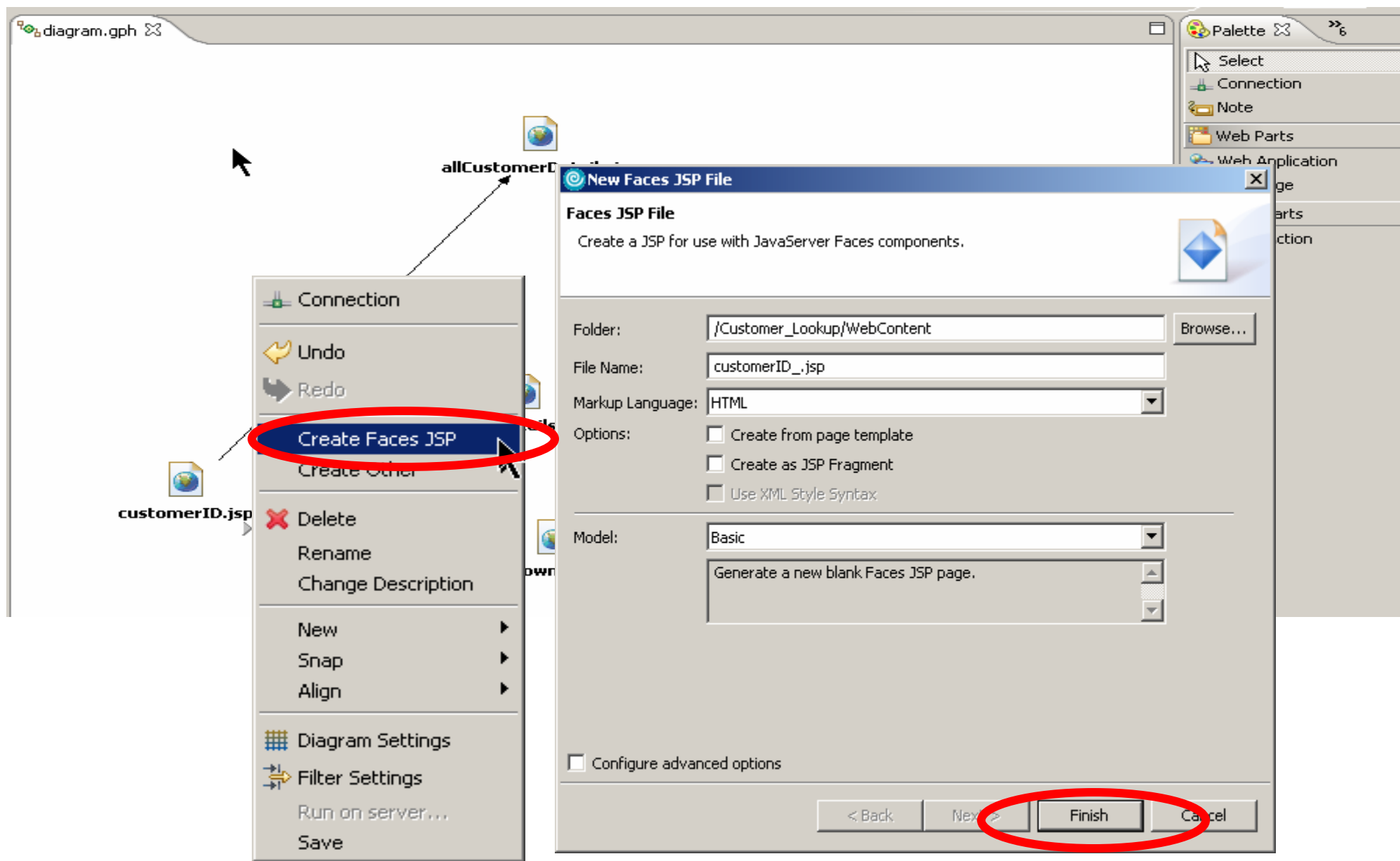
Create project and invoke WebDiagram editor

WebDiagram editor

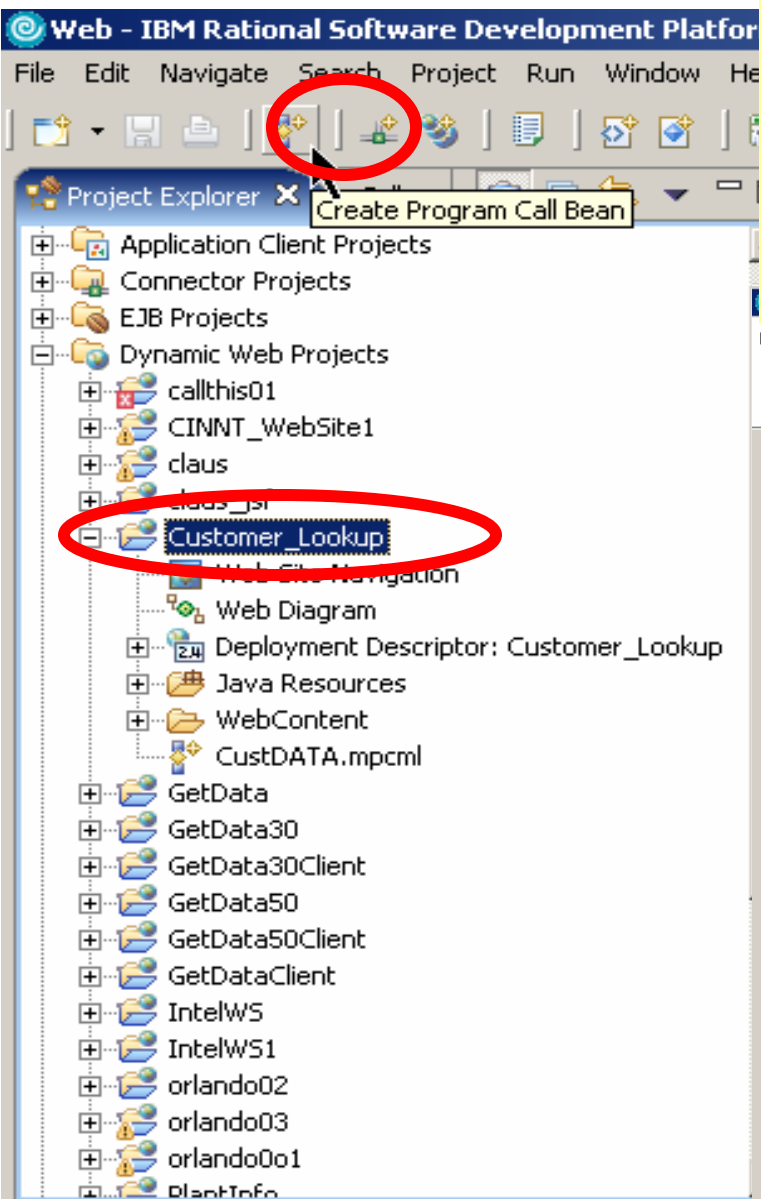


Create design of application layout
web pages and connections between pages with conditions

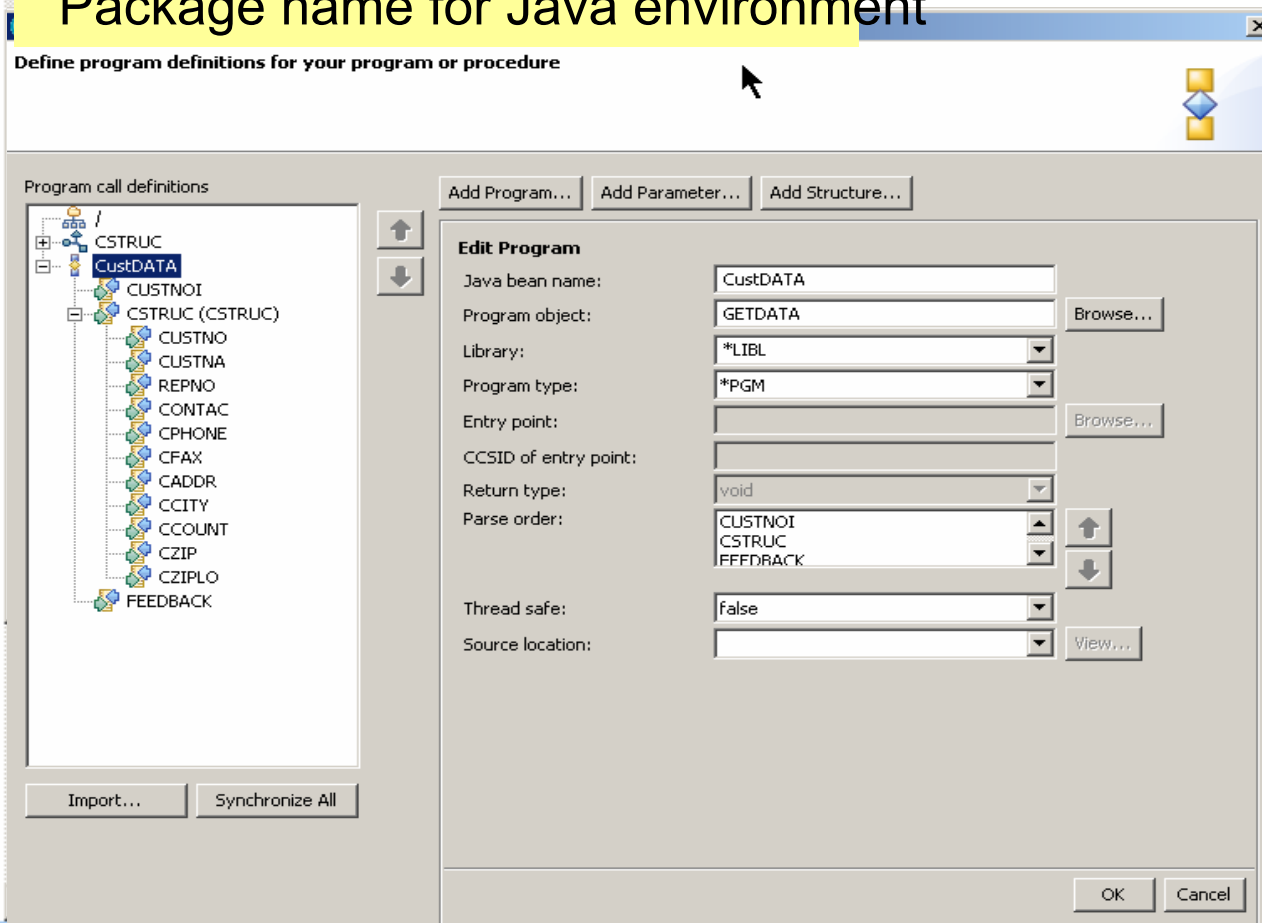
Create webpages from diagram as .jsp files



Create program call bean



Select source member
Also create connection info
iSeries name
User id/password
Package name for Java environment



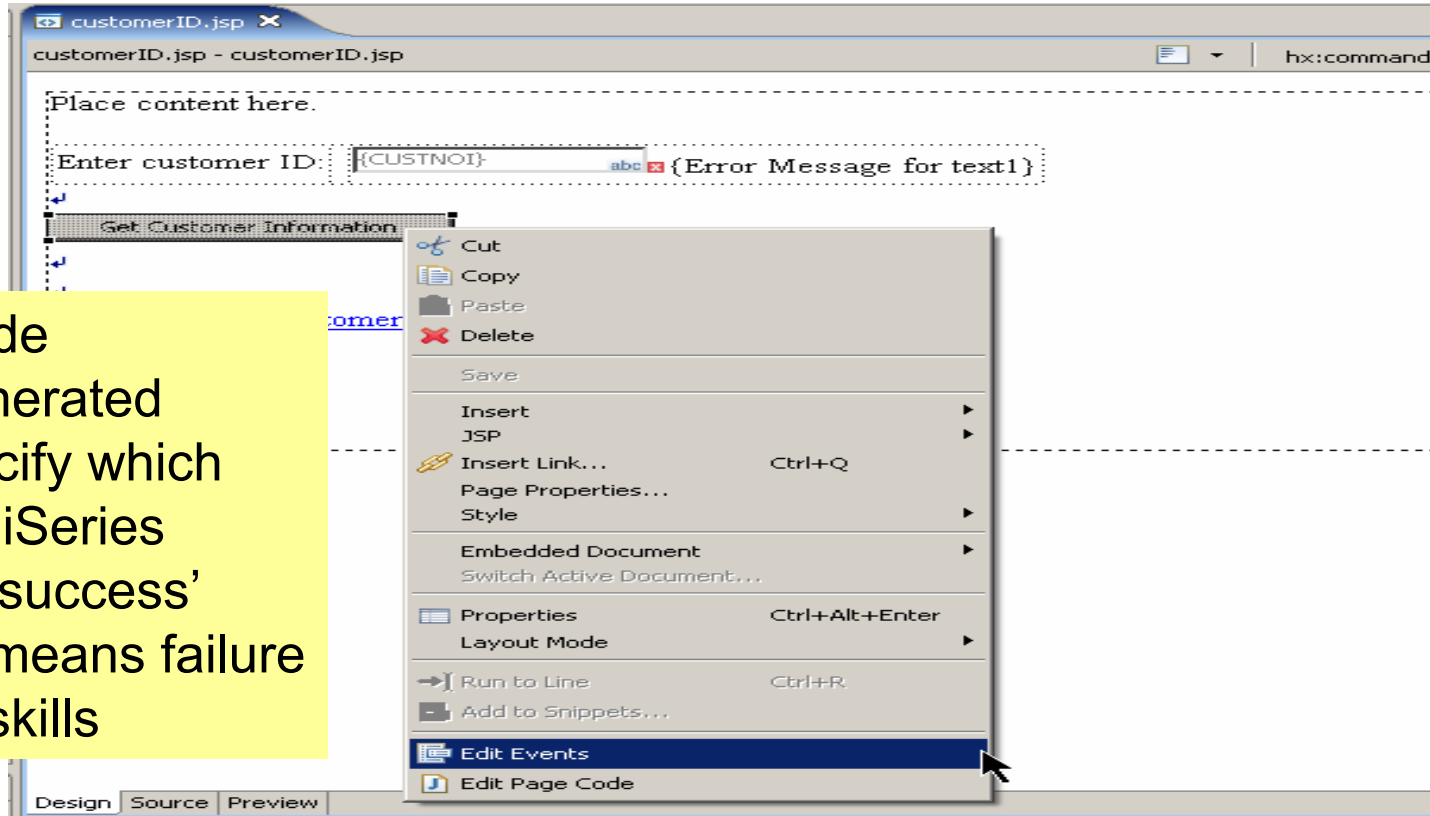
Create Web page design and binding

Created by Program call wizard

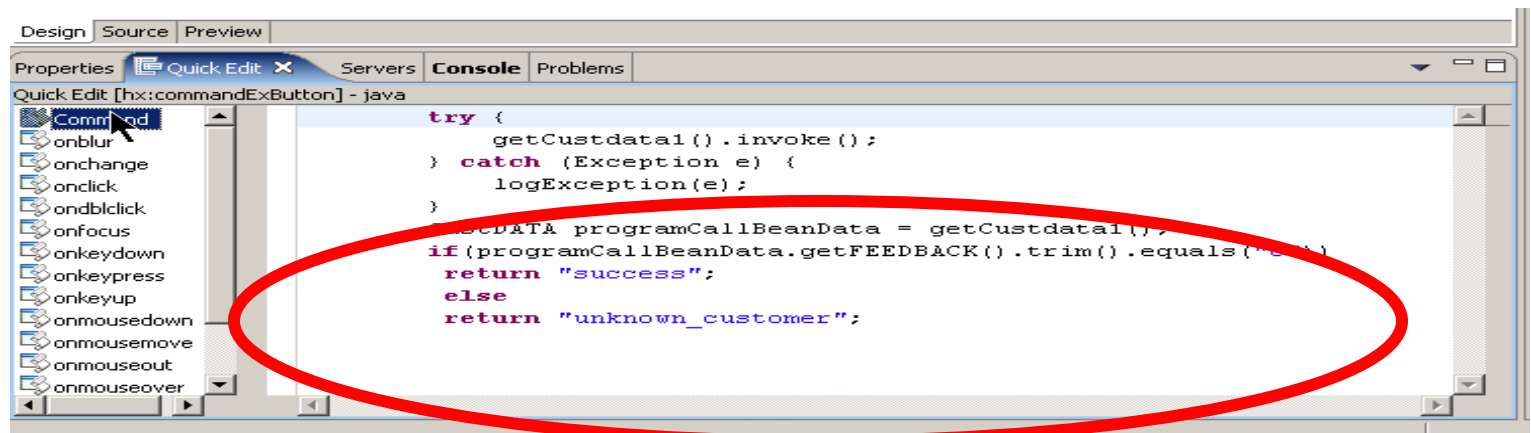
The screenshot displays the IBM Rational Software Development Platform interface. On the left, the Project Explorer shows a web project with various files and folders. The main workspace is divided into several panels. The 'Select JavaBean' dialog is open, showing options to 'Add new JavaBean' or 'Add existing reusable JavaBean'. The 'CustDATA' class is selected in the list. The 'Scope' is set to 'session'. The 'Options' dialog is also open, showing the 'Advanced' tab selected. The 'Submit button' checkbox is checked, and the label text is 'Get Customer Information'. At the bottom left, the 'Faces Managed Beans' window shows the 'custdata (pcwbeans.CustDATA)' bean.

Add program call bean for binding to UI
Creates input field bound to CUSTNOI
Submit button to invoke program call

Add rules for submit button and map return code to forward condition



Beginning of code automatically generated
But have to specify which return code from iSeries program means 'success'
And which one means failure
Very little Java skills required



Bind UI and java bean properties

The screenshot shows the IBM Rational Software Development Platform interface. The main editor displays a web page with a form titled "Get Customer Information" containing an input field for "Enter customer ID:". Below the form is a link labeled "All Canadian customer information".

The Project Explorer on the left shows a project structure with folders like "Customer_Lookup" and "JavaSource", and files like "CustDATA.java" and "CustDATA.pcm1".

The Properties window at the bottom is open to the "Properties" tab. It shows a list of properties for the selected element. The property `CUSTNOI (java.lang.String)` is circled in red. Other visible properties include `FEEDBACK (java.lang.String)`, `initialCommand (java.lang.String)`, `inputData (java.util.Vector)`, `libraryList (java.lang.String[])`, `libraryListPos (java.lang.String[])`, `path (java.lang.String)`, and `PCML (com.ibm.as400.data.ProgramCallDocume`.

The "Insert JavaBean" dialog box is open, showing the "Configure Data Controls" section. The "Inputting" radio button is selected. The "Fields to display" section has a table with the following entries:

Field Name	Method
<input checked="" type="checkbox"/> CUSTNOI	<code>invoke() (void) - pcwbeans.GETDATA</code>

The "Select Method" dialog box is also open, showing a list of methods for the selected bean. The `invoke() (void) - pcwbeans.GETDATA` method is circled in red. Other methods include `initInput() (void) - pcwbeans.GETDATA`, `getCUSTNOI() (String) - pcwbeans.GETDATA`, `setCUSTNOI(java.lang.String) (void) - pcwbeans`, `getCSTRUC() (GETDATA$CSTRUC_Struct_inpu`, `getFEEDBACK() (String) - pcwbeans.GETDATA`, and `setFEEDBACK(java.lang.String) (void) - pcwbea`.

Jsf entry field validation

The screenshot displays the IBM WebSphere Development Studio Client V6.0.1 interface. The main workspace shows a JSP page named `customerID.jsp` in design view. A text input field is visible, with a red circle highlighting the validation error message "(Error Message for text1)" next to it. Below the input field is a button labeled "Get Customer Information" and a link labeled "All Canadian customer information".

The Properties view at the bottom shows the configuration for the `h:inputText` component. A red circle highlights the "Validation" section, which includes the following settings:

- Validation is required
- Minimum length: (characters)
- Maximum length: (characters)
- Constraint:
- Click to create/edit custom validation code

Additionally, a red circle highlights the "Display validation error messages in an error message" checkbox, which is checked. Below this checkbox, the text reads: "When value is invalid, redisplay page with error".

Creating the error page

Sorry customer ID, {CUSTNOI} abc, we not found in the database

Try again abc

Output

Output - Formatted Text

Image

Label

Page Template

Web Site Navigation

Data

Scripting Variables

custdata (pcwbeans.CustDATA)

AS400Object (com.ibm.as400.access.AS400)

connectionData (com.ibm.as400.access.AS400)

CSTRUC (pcwbeans.CustDATA.CSTRUC_Struct_)

CUSTNOI (java.lang.String)

errPa

FEED

initial

input

librar

librar

path (java.lang.String)

PCML (com.ibm.as400.data.ProgramCallDocumen

recordName (java.lano.String)

Bind to 'text3'

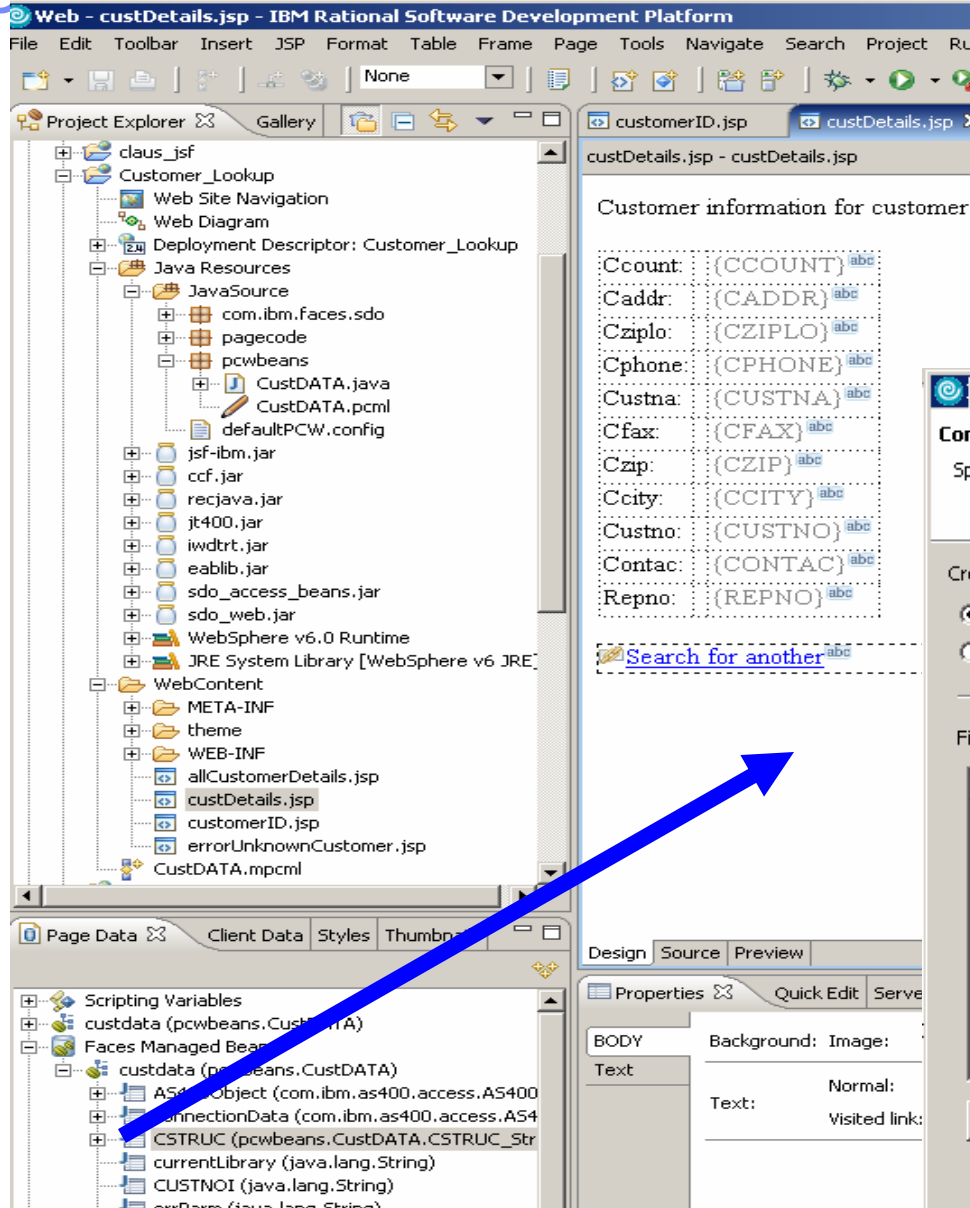
Add New JavaBean Method

Delete

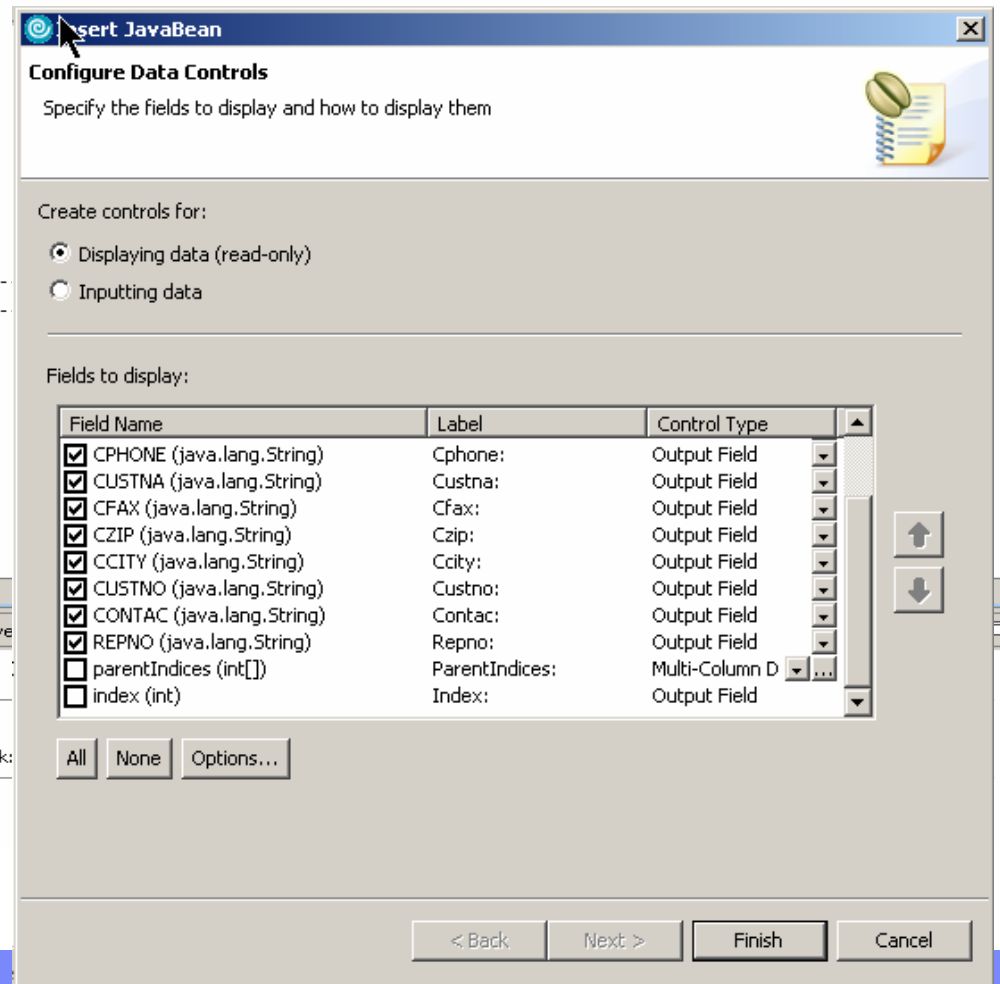
Configure

Create an output text control in the error message
Then bind it to the custnoi field so it gets filled at runtime with its value

Creating the customer detail page



Drag structure from bean to designer
 Select fields to be shown
 Fields get added to page and are bound to bean



Run it

The screenshot displays the IBM Rational Software Development Platform (RSDP) on the left and a Microsoft Internet Explorer browser window on the right. The RSDP interface includes a Project Explorer on the left side showing a project structure with folders like 'Customer' and 'Web C'. The main editor area shows a web page with a form containing several input fields: Ccount, Caddr, Czplo, Cphone, Custna, Cfax, Ccap, and Ccity. A red circle highlights the 'Run' menu in the IDE, with a sub-menu '1 Run on Server...' also circled. Another red circle highlights the 'customerID.jsp' file in the Project Explorer. The browser window shows the URL 'http://localhost:9080/Customer_Lookup/faces/customerID.jsp'. The page content includes a text input field with the value '0010100', a 'Get Customer Information' button, and a link labeled 'All Canadian customer information'. The browser's status bar at the bottom indicates 'Local intranet'.

Result Detail and error pages

The image shows two overlapping Microsoft Internet Explorer browser windows. The top-left window, titled "custDetails.jsp - Microsoft Internet Explorer", displays a successful customer lookup result for ID 0010100. The top-right window, titled "errorUnknownCustomer.jsp - Microsoft Internet Explorer", displays an error message for ID 0010109, indicating it was not found in the database.

custDetails.jsp - Microsoft Internet Explorer
Address: http://localhost:9080/Customer_Lookup/faces/customerID.jsp

Customer information for customer ID: 0010100

Ccount: U.S.A.
Caddr: 10423 S.E. 30th Place
Cziplo: 1
Cphone: 206-865-4027
Custna: Meridien Electronics Limited
Cfax: 206-865-4037
Czip: 98007
Ccity: Bellevue, WA
Custno: 0010100
Contac: Alfredo Bayonne
Repno: 43443

[Search for another](#)

errorUnknownCustomer.jsp - Microsoft Internet Explorer
Address: http://localhost:9080/Customer_Lookup/faces/customerID.jsp

Sorry customer ID,0010109 , we not found in the database

[Try again](#)

Display list of records using Relational record list control

The screenshot displays the IBM WebSphere Development Studio interface. A yellow callout box at the top right contains the text: "Drag Relational record list" and "Fill out wizard pages".

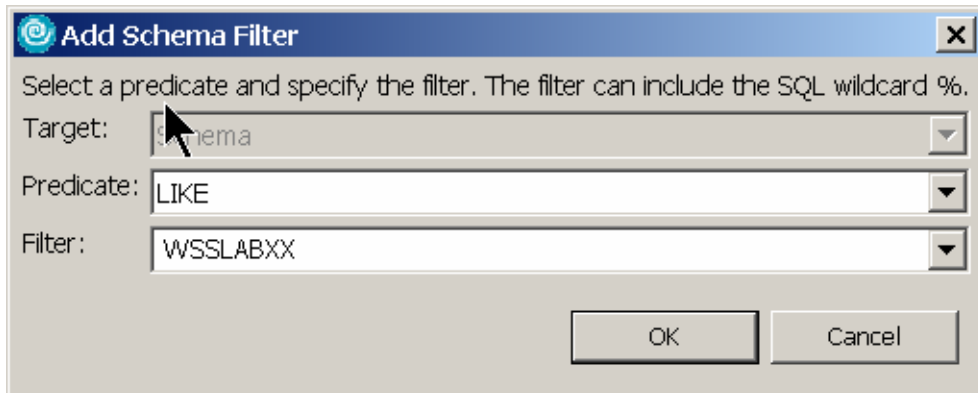
The main workspace shows the "Add Relational Record List" wizard. The "Name" field is set to "canadianCustomers". The "Scope" is "session" and the "Key" is "canadianCustomers". The "Add data controls" checkbox is checked. Below the wizard, the "Choose the database to connect" section shows the connection name "s400aDataBaseConnection" and the "Use Live Connection" radio button selected.

The "New Database Connection - s400aDataBaseConnection" dialog is open, showing the "Specify connection parameters" section. The "Database manager" is "AS/400 Toolbox for Java". The "JDBC driver" is "AS/400 Toolbox for Java". The "Connection URL details" section shows the "Database" field, "Host" set to "s400a", "Port number" field, "JDBC driver class" set to "com.ibm.as400.access.AS400JDBCDriver", "Class location" set to "D:/ProgramFiles/IBM/Rational/SDP/t", and "Connection URL" set to "jdbc:as400:s400a;prompt=false". The "Specify user information" section shows "User ID" set to "wslabxx" and "Password" set to "***".

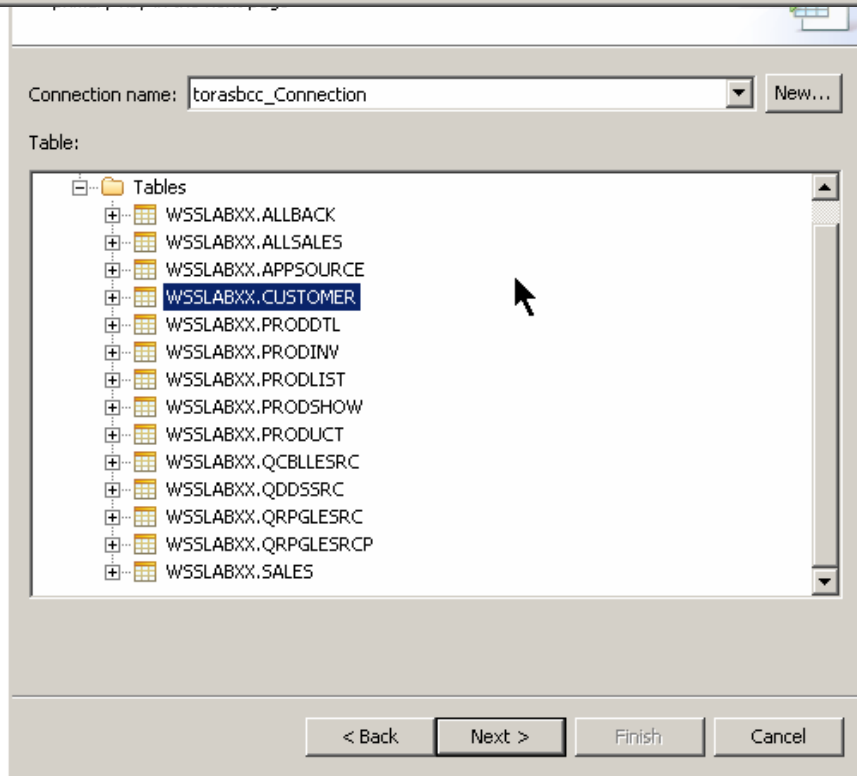
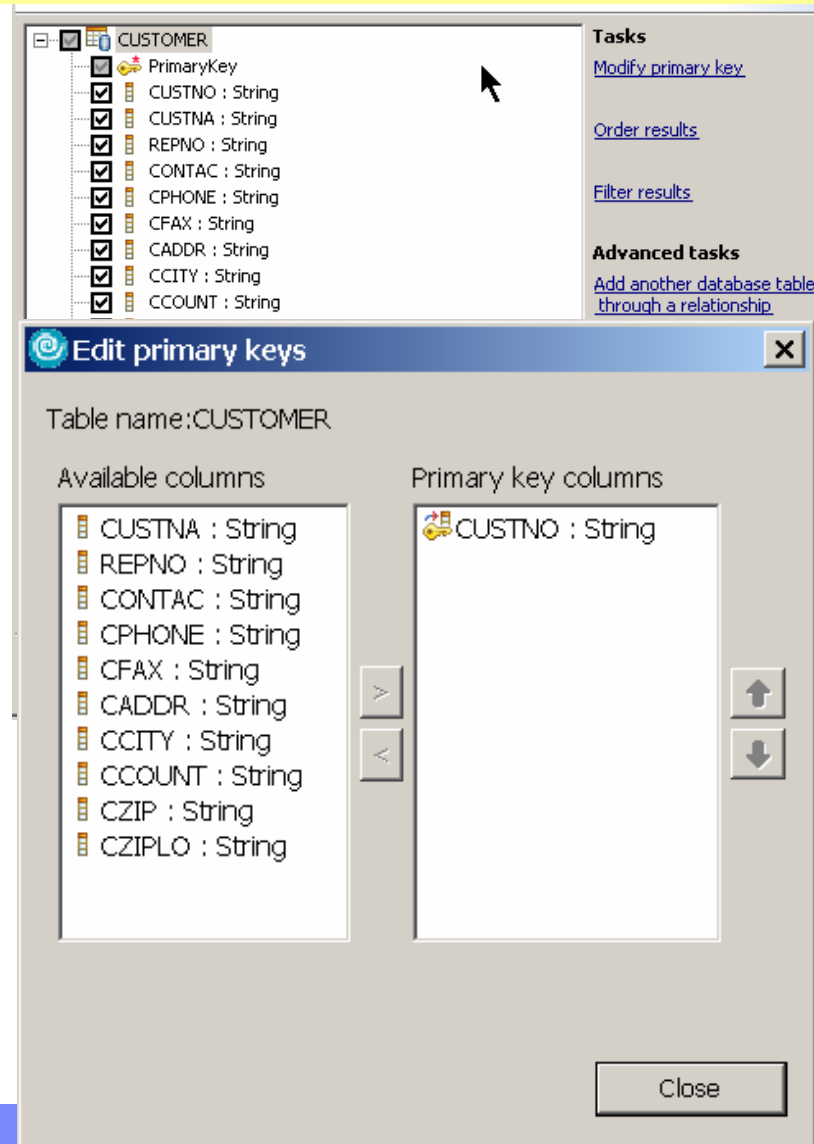
A blue arrow points from the "Test Connection" button in the "New Database Connection" dialog to the "Add Relational Record List" wizard, indicating the flow of the configuration process.

The right-hand side of the interface shows the "HTML Tags" palette, which includes various controls such as "Data Table", "Panel - Group Box", "Panels - Tabbed", "Panel - Menu Bar", "Command - Button", "Command - Hyperlink", "Link", "Input", "Input - Text Area", "Rich Text Area", "Input - Password", "Check Box", "Check Box Group", "Radio Button Group", "Combo Box", "List Box - Single Select", "List Box - Multiple Select", "File Upload", "Output", "Output - Formatted Text", "Image", "Label", "Page Template", "Web Site Navigation", "Data", "Web Service", "EJB Session bean", "Relational Record", and "Relational Record List".

Identify iSeries data base file



Filter for library
 Select database file
 Specify primary key
 Almost done



Specify the data to be shown

List page has been created
 Specify SQL Query conditions
 Specify sort etc
 The run it

The screenshot shows the IBM Rational Software Development Platform interface. The main window displays a web page titled 'allCustomerDetails.jsp' with a table of customer information. The table has columns: Custno, Custna, Repno, Contac, Cphone, Cfax, Caddr, Ccity. Below the table, there is a 'Conditions' dialog box with the following settings:

- Condition: By columns
- Set a condition where: CCOUNT LIKE '%Canada%'

The 'Order by' dialog box is also open, showing the following settings:

- Available columns: CUSTNO, REPNO, CONTAC, CPHONE, CFAIX, CADDR, CCITY, CCOUNT, C7ID
- Order by: CUSTNA
- Sort order: Ascending

The 'Page Data' panel at the bottom left shows the 'canadianCustomers (Service Data Object)' selected, with a 'Configure' button highlighted.

Running the query

allCustomerDetails.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail New Tab

Address http://localhost:9080/Customr_Lookup/faces/customerID.jsp

The following information for Canadian customers was retrieved:

Custno	Custna	Repno	Contac	Cphone	Cfax	Caddr	Ccity	Ccount	Czip	Cziplo
0011300	Burnham Trading Inc	41480	Efrem Helassie	613-225-0753	613-225-0753	91 Baseline Road	Trenton, Ontario	Canada	C6B 9S3	2
0013400	Communications Fournier	13497	Larry Schweyer	514-534-2953	514-534-2977	3162 Rue Ste Catherine	Montreal, Quebec	Canada	E6P 2S3	1
0012200	Communications-R-U's Corporation	22004	Jack Deauson	416-695-3455	416-695-3458	82 Royal York Road	Toronto, Ontario	Canada	M7G 5C1	2
0012300	Danton Industries	98989	Jack Vance	905-316-2245	905-316-2255	78 John Street	Markham, Ontario	Canada	M0J 8G4	1
0012100	DaSilva Novelities	17506	Henry Goldsmith	416-484-2766	416-484-2795	926 Danforth Avenue	Toronto, Ontario	Canada	M5R 6C1	2
0012500	Diplomat Sales and Liquidations	00488	Jim Doe Wan	905-369-1234	416-343-1027	Gerrard Square, 54 Gerrard St.	Toronto, Ontario	Canada	M3S 6F0	1
0012600	Down Under Electronics Supplies	42167	John Riley	416-694-1459	416-694-1469	206 Queen Street East	Toronto, Ontario	Canada	M3S 6G7	2
0012700	Dunsten Electronics Inc.	60585	Kersi Copper	416-633-1027	416-633-1029	1298 Victoria Park Avenue	North York, Ontario	Canada	M0K 8F4	2
0012800	Elite Communications	26489	Marilyn Myerson	416-343-2077	416-343-3079	257 Queen Street East	Toronto, Ontario	Canada	M8Y 3K9	2
0013000	Fantastic Future Supplies	63445	Mary Goodbar	905-316-4545	905-316-4572	Markham Place, 500 Steeles Ave E.	Markham, Ontario	Canada	M9U 6M1	2
0013500	Gallery Sales and Liquidations	64616	Mostafa Egoury	416-961-2722	416-961-2752	73 University Avenue	Toronto, Ontario	Canada	M9K 6G7	1
0015300	Holiday Sales	14915	William Riso	905-891-5677	905-891-5697	305 Appleby Line	Mississauga, Ontario	Canada	M9B 7F5	1
0909090	I&B&M&A&B	12345	Mr. Customer	416-448-9809	416-448-4545	1150 E&A Ave	North York	Canada	M1M 2N2	1
0010400	ProLine Building Supplies	13300	Bud Dobbs	905-403-4055	905-403-4059	73 Marchwood Road	Burlington, Ontario	Canada	D8G 3V6	2
0010200	Royal Hardware Supplies	20527	Arnie Podell	905-619-2045	905-619-2073	Maple View Plaza, 256 New St.	Ajax, Ontario	Canada	L8D 4S6	2
0011600	Sudbury Radio and TV	01900	Gary Morehouse	705-522-5044	705-522-5047	7 North Road	Sudbury, Ontario	Canada	P7G 5A3	2


Done Local Intra

WebSphere Test Environment

A full copy of WAS 6.0 is embedded in the IDE

- ▶ Integrated with Server Tools to enable instant and dead-easy testing of Web projects within WDS.
- ✓ 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

WebTools agenda

- ▶ WDS overview and AD roadmap
- ▶ Website creation
- ▶ Webpage templates
- ▶ Designer for static WebPages
- ▶ Designer for Web objects
- ▶ Tool for Cascading Style Sheets
- ▶ Tools to create WebApplications
 - ▶ Interaction wizard
 - ▶ Java server faces tooling
- ▶ Tools to create WebServices 

What Are Web Services

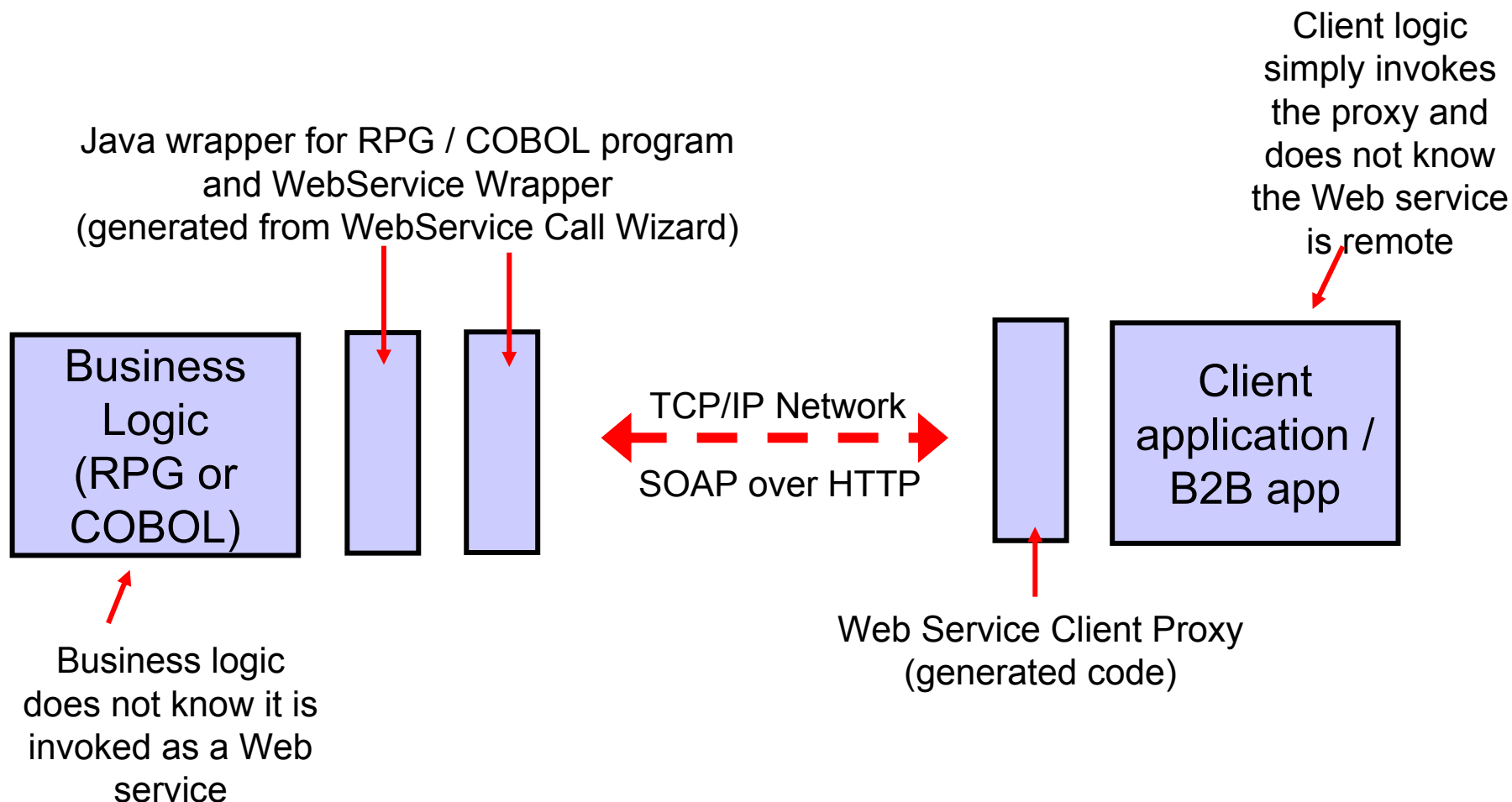
- Function that can be programmatically invoked over a network

- Basically remote procedure calls built on open standards and proven technologies
 - Lots of new standards around Web services to ensure interoperability in heterogeneous environments
 - Underlying implementations built on proven technologies like
 - XML
 - HTTP
 - Messaging middleware
 - Security standards

Why use Web Services

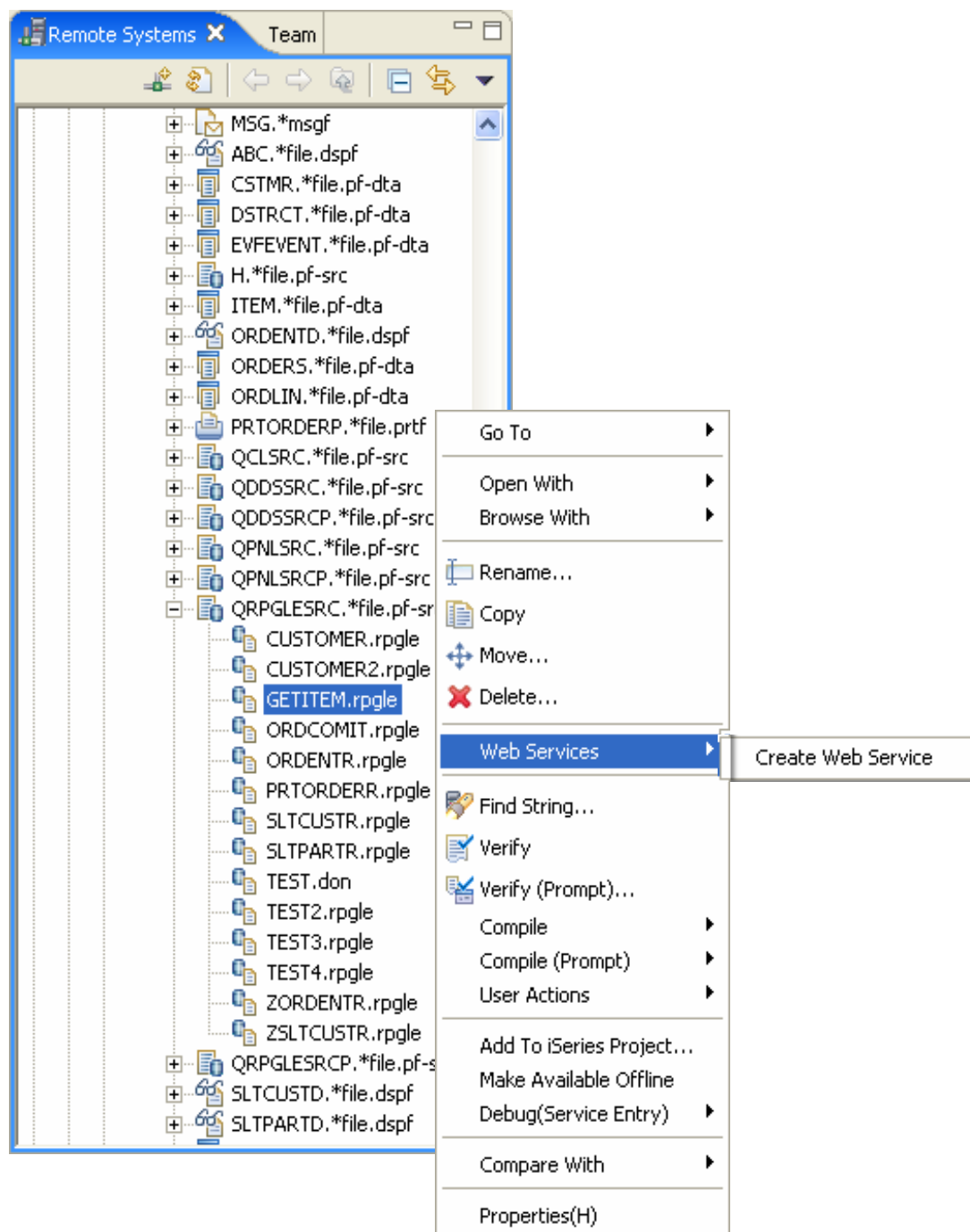
- **Because:**
 - They are the latest hype
 - They are hot
 - They are cool
 - They are useful
 - Somebody asks you to provide one for a certain task
 - Somebody provides one and asks you to use it

RPG/COBOL Program as Service Provider



System i5 Web Services Tools (new in V6.0.1)

- Single wizard to convert i5/OS program or service program procedure into a Web Service
 - Generates all required code for deploying Web Service to WebSphere Application Server
- Automatically determines program (or procedure) parameters when invoked from RSE
- Just point to RPGIV/ILE COBOL program source
- Go thru a couple of wizard pages and WSDL for the program or procedure will be generated



Tools

■ Web Services Tools in WDS

– Create

- Web services from:
 - ILE RPG/COBOL source
 - Java Beans
 - Enterprise JavaBeans (advanced edition only)
 - SQL Queries
 - DB2 Stored Procedures
 - DB2 XML Extender calls
- Web service client proxy given a WSDL document

– Test

- Test your Web services in the WebSphere test environment
- Generate test code to test generated Web service client proxies

– TCP/IP Monitor

- Useful for debugging SOAP messages between provider and consumer

WebService Wizard

Use Remote Systems Explorer to select member

Remote System Explorer - IBM Rational Software Development Platform

Edit Navigate Search Project Run Window Help

Remote Systems Team

Select member

From popup select WebService option

torasbcc

iSeries Objects

- Work with libraries...
- Work with objects...
- Work with members...
- Library list
 - QSYS.*lib.prod-sys
 - QSYS2.*lib.prod-sys
 - QHLPYSYS.*lib.prod-sys
 - QUSRSYS.*lib.prod-sys
 - WSSLABXX.*lib.prod-usr
 - DEMOR.*pgm.rpgle
 - DEMORES.*pgm.rpgle
 - GETDASOL.*pgm.rpgle
 - GETDASOLV.*pgm.rpgle
 - GETDATA.*pgm.rpgle
 - GETDSSOLC.*pgm.cbllc
 - READDTAQ.*pgm.rpgle
 - WRDTAQ.*pgm.rpgle
 - GETDATAS.*srvpgm.rpgle
 - GETDATASC.*srvpgm.cbllc
 - GETDSSOL.*srvpgm.rpgle
 - ORDENTR.*srvpgm.rpgle
 - GETDATAS.*module
 - GETDATASC.*module
 - GETDSSOL.*module
 - SLTPARTRFA.*module
 - CELDIAL.*msgf
 - INVMMSGF.*msgf
 - MYMSGF.*msgf
 - VRPGMSGF.*msgf
 - ALLBACK.*file.pf-dta
 - ALLSALES.*file.pf-dta
 - APPSOURCE.*file.pf-src
 - BYREGION.*file.if
 - CELDIAL.*file.dspf
 - CUSPRINT.*file.prtf
 - CUSTINQ.*file.dspf
 - CUSTOMER.*file.pf-dta
 - CUSTOML1.*file.if
 - CUSTOML3.*file.if
 - CUSTOMS1.*file.if
 - CUSTOMS2.*file.if
 - CUSTOMS3.*file.if
 - CUSTOMS4.*file.if
 - EVFEVENT.*file.pf-dta
 - INVPRINT.*file.prtf

Remote System Details Tasks iSeries Tab

File WSSLABXX/QRPGLESRC (8 Members)

Name	Type	Attribute	Properties	Last modified	Created
GETDASOL	RPGLE	SRC		October 5, 2003 11:...	April 22, 2005...
GETDASOLV	RPGLE	SRC		October 5, 2003 11:...	April 22, 2005...
GETDATA	RPGLE	SRC		October 17, 2003 4:...	October 17, 2003 4:...
GETDATAS	RPGLE	SRC	OK	April 10, 2004 3:55:...	October 17, 2003 4:...
GETDSSOL	RPGLE	SRC	OK	April 10, 2004 3:56:...	October 17, 2003 4:...
SUBFILEPR2	RPGLE	SRC	OK	February 28, 2005 ...	February 28, 2005 ...
WDSCSRV	RPGLE	SRC	OK	February 28, 2005 ...	February 28, 2005 ...
WDSCSRV2	RPGLE	SRC	OK	February 28, 2005 ...	February 28, 2005 ...

Web Service Wizard

- Click Finish & Done!
- Drop-Dead Simple!

Default is iSeries Web Service type

The Java bean proxy will provide a remote procedure call interface to the Web service

We will test the Web service after it is created

We will monitor the SOAP traffic for this Web service

Web Service

Web Services
Review your Web service options and make any necessary changes before proceeding to the next page.

Service

Web service type: iSeries Program Web Service

Start Web service in Web project

Launch the Web Services Explorer to publish this Web service to a UDDI Registry

Generate a proxy

Client proxy

Client proxy type: Java proxy

Test the Web service after it is created

Monitor the Web service SOAP traffic

< Back **Next >** Finish Cancel

We could select Finish here and test the Service
But lets specify some more details

Configuring the Web Service- Edit Program/Parameters

-File name shows the Program source file the wizard was launched from (valid types incl. ILE RPG, COBOL, PCML)

-**Browse files** to select a different file

-The default Runtime configuration is taken from the System i5 connection

-**Edit** to change the configuration

-**Browse** to choose an existing configuration (.config file)

- Selecting the Program (**CUSTINFO**), you can change the Library, Program type, Program Object etc. (in most cases the defaults are sufficient)

- Expand the program to see program parameters

- Selecting a parameter (e.g. **IN_CID**), you can change Usage type, CCSID

Web Service

Object Selection Page

Object Selection Page

File name: :RVICE/QRPGLESRC(CUSTINFO) Browse files...

Runtime configuration: CUSTINFO Edit... Browse...

Program call definitions

- OUT_DS
 - CUSTINFO**
 - IN_LNAME
 - IN_CID
 - OUT_STATU
 - O_DS (OUT
 - OUT_CII
 - OUT_FN
 - OUT_LN
 - OUT_AC

Java bean name: CUSTINFO

Program object: CUSTINFO

Library: *LIBL

Program type: *SRVPGM

Entry point: CUSTINFO

CCSID of entry point:

Parse order: OUT_STATUS
O_DS

Thread safe: false

Configuring the Web Service- Runtime Configuration

Signon information is defaulted from System i5 connection

Alternatively, can use the Program Call Java Connection Architecture connector for signon and connection management (advanced product only)

iSeries Web Service Runtime Configuration

Configure Authentication

Specify how the program call runtime should connect and authenticate

Specify signon values

Host name: TORASCGM

User ID: ERICDP

Password: *****

Enable password encoding

Use program call JCA connector

JNDI name:

< Back Next > Finish Cancel

Testing the Web Service

After wizard completes, the generated Test Client JSPs are run on the server (selected option "Test the Web service" in the wizard)

Select one of the methods to test, then enter the input data, Customer ID "0001"

Click "Invoke" and the results of the Web service are displayed, Customer ID "0001" is associated with Annie O!

- As you will see shortly, there are many different ways to test the Web service

Web Browser x

http://CUSTOMER.Services.Pro//TestClient.jsp

Methods

- [custinfo \(iseries.wsbeans.c\)](#)
- [custinfo_XML \(iseries.wsbeans.c\)](#)

Inputs

inputData:

iN_LNAME:

iN_CID:

Invoke Clear

returnp:

oUT_STATUS:

o_DS:

rEC_FOUND: 1

oUT_STATE: IO

oUT_FNAME: Annie O

oUT_COMMENTS: 00001

WSDL editor in WDS Sc (graphical view)

The screenshot displays the WSDL editor interface within the IBM Rational Software Development Platform. The main workspace is titled 'Definition' and is divided into several panels:

- Imports:** An empty panel for external WSDL imports.
- Types:** A panel showing a reference to the namespace `http://ws`.
- Services:** Contains the `InventoryService` with a sub-element `Inventory` that has a `wsdlsoap:address` property.
- Bindings:** Shows the `InventorySoapBinding` for the `Inventory` service. It includes a `wsdlsoap:binding` and a `getProductInfo` operation with `wsdlsoap:input` and `wsdlsoap:output` elements.
- Port Types:** Shows the `Inventory` port type with a `getProductInfo` operation.
- Messages:** Shows two messages: `getProductInfoRequest` (containing `parameters (impl:getProductInfo)`) and `getProductInfoResponse`.

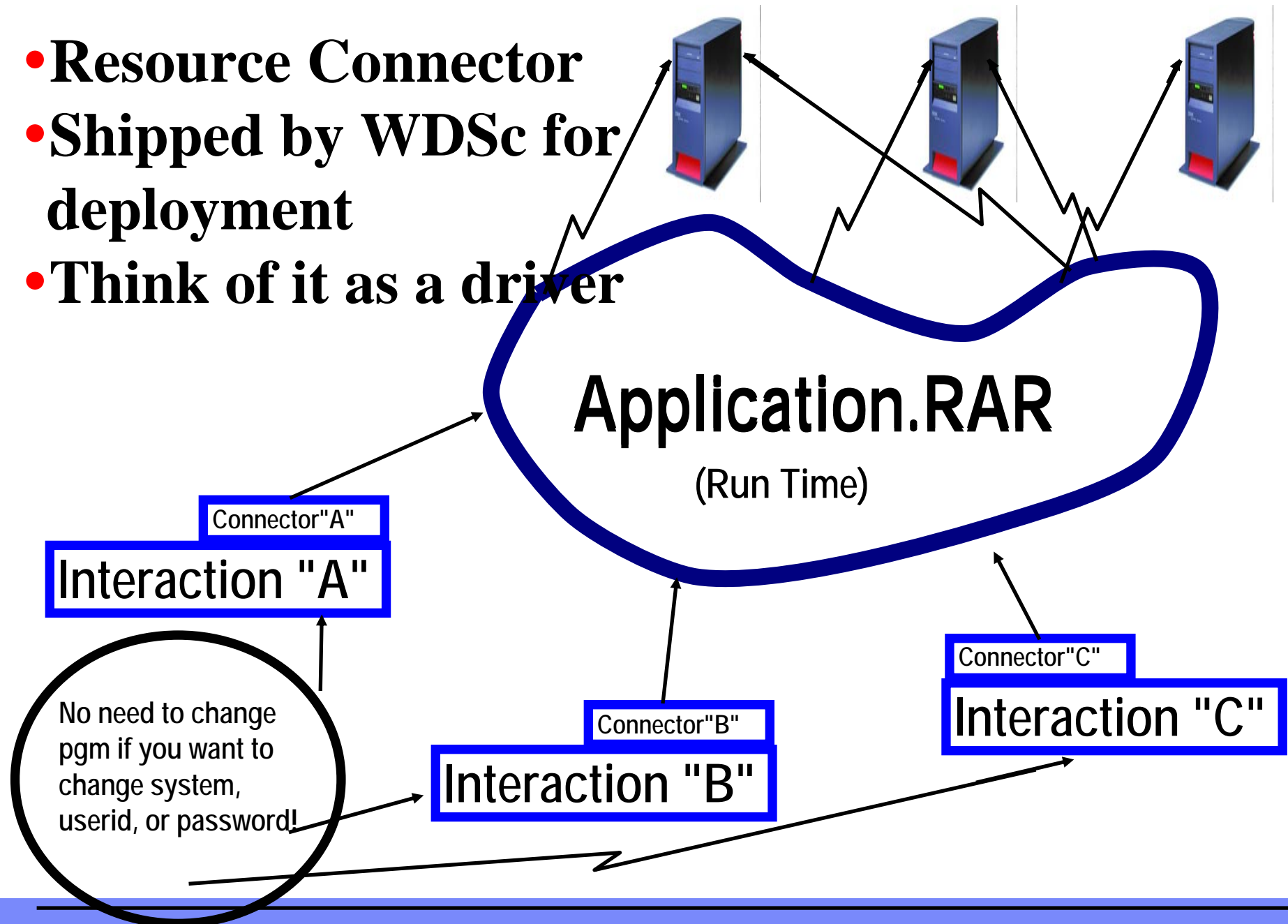
Arrows indicate the relationships between these elements. A detailed view of the `getProductInfo` operation is shown on the right, illustrating its signature: `getProductInfo` (type = <anonymous>) with a parameter `prodId` (type = xsd:string).

Web Tool for iSeries

- ✚ Wizards to help create iSeries RPG/COBOL web applications
 - Interaction wizard to build an RPG web application
 - WebService wizard to build an iSeries WebService
 - JCA wizard to build an iSeries Java Connector Architecture (JCA) connector

What is JCA Connectors?

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



Creating a connector

- Use runtime configuration dialog
- Check JCA checkbox
- Specify JNDI name

iSeries Web Tools Run-time Configuration

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

iSeries host name: torascgm

User ID: weiss

Password: *****

Prompt for iSeries user ID and password

Runtime library list:

Library	Library Position
WEISS	*LAST
GUIDESWSS	*LAST

Library: Add

Remove

Move Up

Move Down

Current library: *USRPRF

Initial command:

Display detailed runtime errors

Use program call JCA connector

Specify the JNDI name: MyFirstWebApplication

Finish Cancel

SUMMARY

- ▶ **System i5 web tools, at a glance**
 - **Tools optimized for System i5 developers!**
 - ▶ **Web Interaction wizard (deprecated using STRUTS framework)**
 - ▶ **Jsf tools with iSeries support**
 - ▶ **Use Program call bean**
 - ▶ **Use direct access to data base**
 - ▶ **New and enhanced Web Service wizard**
 - ✓ **Make your system i5 programs available as a Web Service**
 - ▶ **JCA wizard**
 - ✓ **Create connectors for you system i5 programs**

More Information?

▶ Information Sources

- **www.ibm.com/software/awdtools/iSeries**
 - ▶ For iSeries Application Development
- **www.eclipse.org**
 - ▶ Eclipse and information about eclipse
- **www.ignite400.org**
 - ▶ Introduction to eclipse article
- **www.ibm.com/software/info1/websphere/partners/series.jsp**
 - ▶ WebSphere on iSeries home page for BPs
- **eServer iSeries magazine, July 2002 issue**
 - ▶ 3 articles on WDS
- **www.ibm.com/websphere/developer**
 - ▶ WebSphere Developer Domain
 - ▶ Many articles and tutorials on technology and tools, including eclipse and WSWB and WebSphere Studio configurations

Disclaimer

Acknowledgement:

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

Disclaimer:

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

Reproduction:


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

Trademarks and Disclaimers

© IBM Corporation 1994-2006. 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 of International Business Machines Corporation in the United States, other countries, or both:

AS/400	e-business on demand	i5/OS
AS/400e	IBM	OS/400
eServer	IBM (logo)	System i5
	iSeries	

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.

Intel, Intel Logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or 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 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 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.