

## Web Tools for IBM System i5 developers

**ibm.com**  
the power of one

### IBM System i5 ITSO Technical Forum 2006


Vadim Berestetsky  
berestet@ca.ibm.com



## AGENDA

iSeries AD, IBM Toronto

- WebSphere Development Studio WDS V5R4
- e-business primer
  - AD Model, traditional and web
  - Web Applications
  - Mixing Java and RPG
- Introducing WebTools for IBM System i5
  - Web projects
  - Interaction wizard
  - Page designer
  - Web Service wizard
  - JCA wizard
  - JSF (JavaServer Faces)

ibm.com/redbooks | International Technical Support Organization 

## WebSphere Development Studio

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

**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

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation

ibm.com/redbooks | International Technical Support Organization 

## WebSphere Development Studio Client Advanced Edition 6.0.1

**Workstation License order through Passport Advantage**  
[http://www.lotus.com/service/passport.nsf/WebDocs/Passport\\_Advantage\\_Home](http://www.lotus.com/service/passport.nsf/WebDocs/Passport_Advantage_Home)

**Upgrade from WDS 6.0 to 6.0.1 using Rational Product updater**

**+CODE +VisualAge RPG**

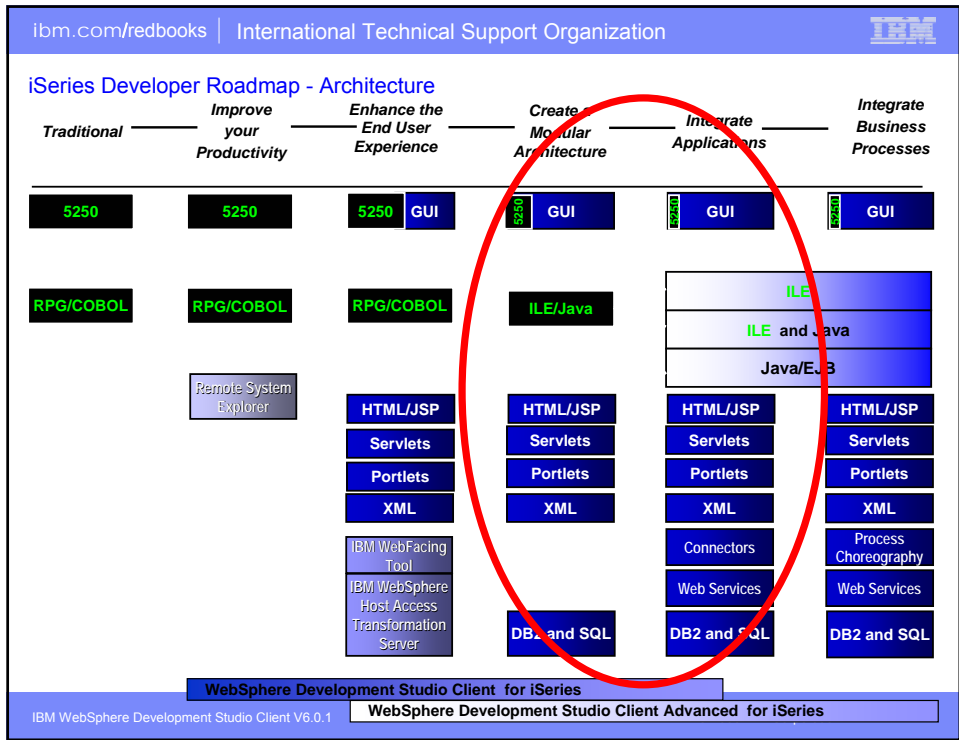
iSeries	iSeries	iSeries *	iSeries	Web Facing * WDHT support	iSeries Projects	+CODE +VisualAge RPG	
Java	Debug	Struts Web	Web Service		RSE		
JSF	EGL Java generation	Trace	Profiling	DB	XML	App Server	HATS Toolkit
	EGL * COBOL generation	EJB * J2EE *	Test * Cases	Portal *			

**New WDS Lite**  
Technology preview

www.ibm.com/software/awdtools/iseries

WebSphere Development Studio Client V6.0.1 based on RAD V6

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation



ibm.com/redbooks | International Technical Support Organization

## AGENDA

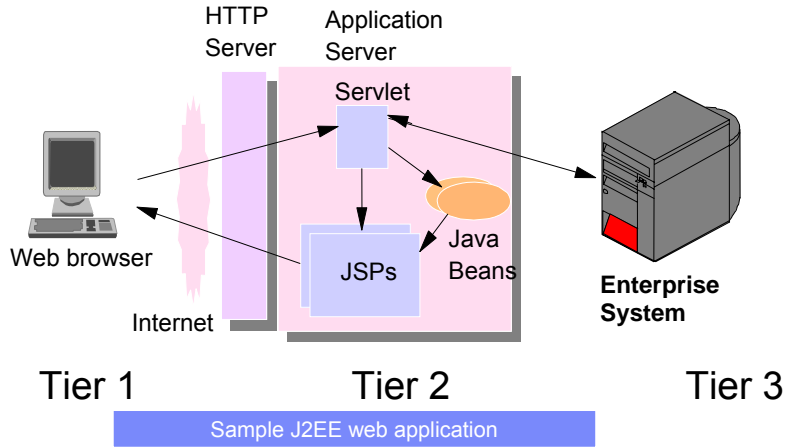
iSeries AD, IBM Toronto

- WebSphere Development Studio WDS V5R4
- e-business primer
  - AD Model, traditional and web
  - Web Applications
  - Mixing Java and RPG
- Introducing WebTools for IBM System i5
  - Web projects
  - Interaction wizard
  - Page designer
  - Web Service wizard
  - JCA wizard

←

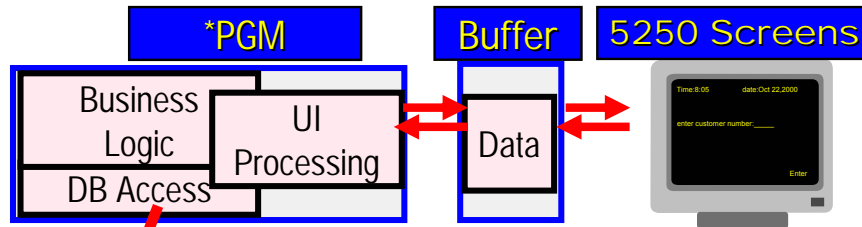
IBM WebSphere Development Studio Client V6.0.1 | © 2006 IBM Corporation

# Modern App Architecture



# Today's Model

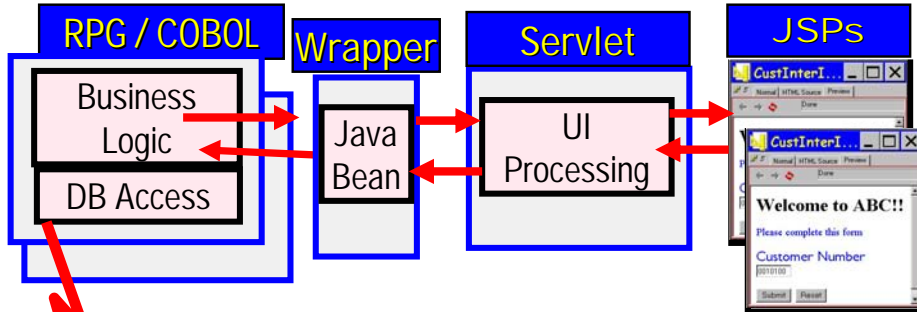
iSeries AD, IBM Toronto



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

# eBusiness Application

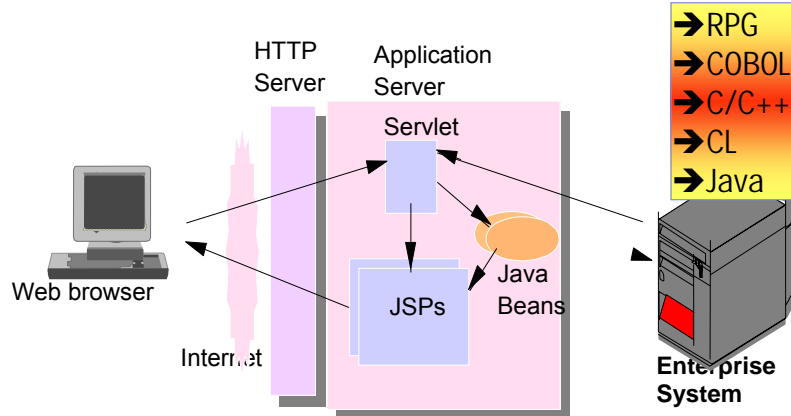
iSeries AD, IBM Toronto



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

DB2 UDB

## Tooling for building modern applications



**WDS**

→ iSeries webtooling

→ iSeries 3 GL tooling

→ Pagedesigner

→ Remote system explorer

→ Interaction wizard

→ LPEX source editor

→ WebServices wizard

→ Integrated Debugger

## What Are Servlets?

iSeries AD, IBM Toronto

- ▶ Servlets are . . .
  - Java classes (programs written in Java) that run . . .
    - On an application server (eg, WAS)
- ▶ Servlets are called . . .
  - By your HTTP Server software
  - From other Servlets
- ▶ The input to Servlets is . . .
  - User-entered data from a Web page
- ▶ The output of a Servlet is . . .
  - Java Bean, passed to a JavaServer Page

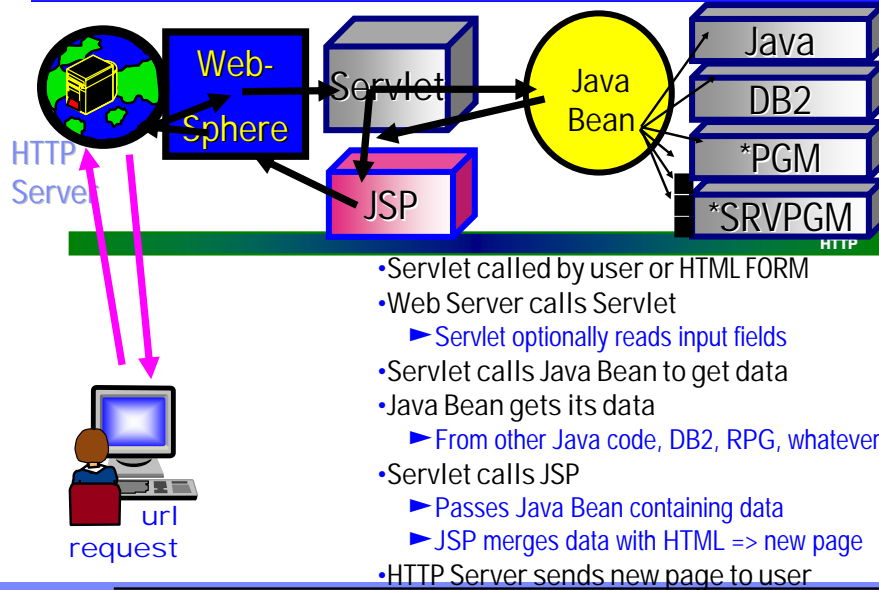
## What Are JSPs?

iSeries AD, IBM Toronto

- ▶ A way of describing dynamic web pages
- ▶ JavaServer Pages (JSPs) are . . .
  - .jsp files
    - ▶ containing html tags plus JSP tags
- ▶ JSP tags . . .
  - Allow dynamic data to be inserted into the static HTML
- ▶ JSPs are invoked . . .
  - By a servlet
  - The input to JSPs are . . .
    - ▶ Java Beans passed from your Servlet
  - The output of a JSP is . . .
    - ▶ A full Web page, displayed to user

## Web Model

iSeries AD, IBM Toronto



## Calling native programs from Java

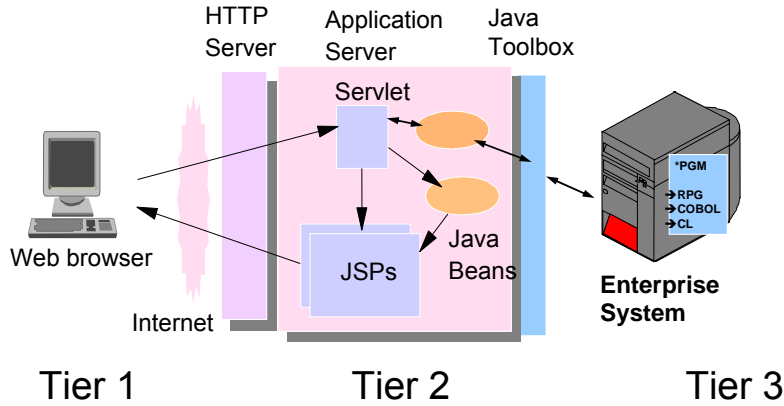
iSeries AD, IBM Toronto

A brief look at:

IBM System i5 Java toolbox

- Using Program Call Markup Language (PCML)

## Calling native programs from Java



Using the iSeries tool box classes to access native objects

## 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      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

iSeries AD, IBM Toronto

### 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

iSeries AD, IBM Toronto

```

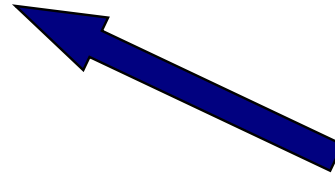
public static void main(String[] argv)
{
  AS400 as400System = new AS400();
  ProgramCallDocument pcml = null;
  String msgId, msgText;
  Object value = null;
  try {
    System.out.println(
      "Creating ProgramCallDocument for GetCust pgm.");
    pcml = new ProgramCallDocument(as400System, "GETCUST");
    boolean ok = pcml.callProgram("getcust");
    System.out.println(" rc is---> " + 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

```

# AGENDA

iSeries AD, IBM Toronto

- WebSphere Development Studio WDS V5R4
- e-business primer
  - AD Model, traditional and web
  - Web Applications
  - Mixing Java and RPG
- Introducing WebTools for IBM System i5
  - Web projects
  - Interaction wizard
  - Page designer
  - Web Service wizard
  - JCA wizard



## Web Application

iSeries AD, IBM Toronto

### ▶ Standard Web App folder structure:

```

+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

```

collectively known as "Web Resources"

maps dependent jar files in other Web apps

J2EE  
1.3

Web application deployment descriptor:  
▶ identifies servlets, security, env vars, mime types, key pages, external references and session configuration info

# Terms: Web Application

iSeries AD, IBM Toronto

## ▶ Example Web Application

```

+accounts
  +source
  +WebContent
    -index.html
    +receivable
      -page1.html
    +payable
      -page1.html
  +META-INF
    -MANIFEST.MF
  +theme
    -corporate.css
  +WEB-INF
    -web.xml
  +classes
+lib
    
```

www.mydomain.com/accounts

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

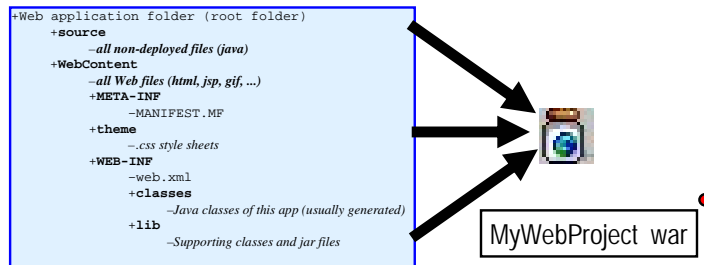
www.mydomain.com/accounts/payable/page1.html

iSeries AD, IBM Toronto

## ▶ Web Archive Files (WAR)

J2EE  
1.3

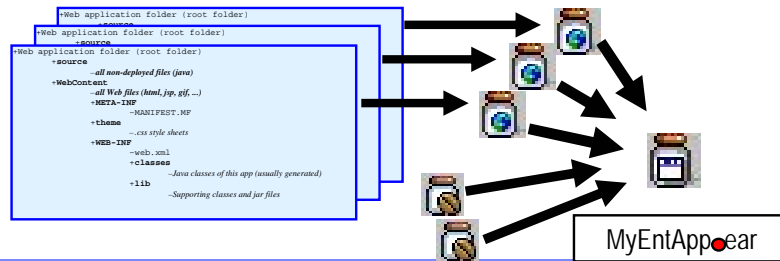
- **One file containing**
  - ▶ Whole folder structure of Web application
  - ▶ Including web.xml file
  - ▶ Optionally including source
- **Used to**
  - ▶ Install and configure Web application in an application server



## ▶ 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)



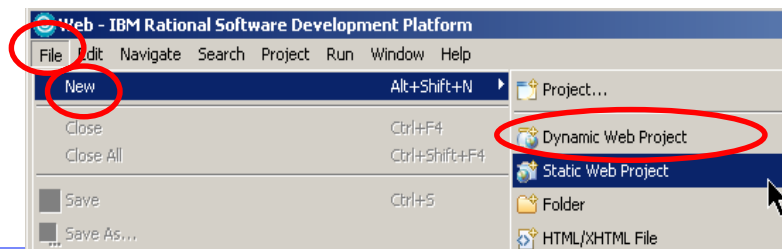
## WDS Sc Web Tools At A Glance

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

# Web Projects

iSeries AD, IBM Toronto

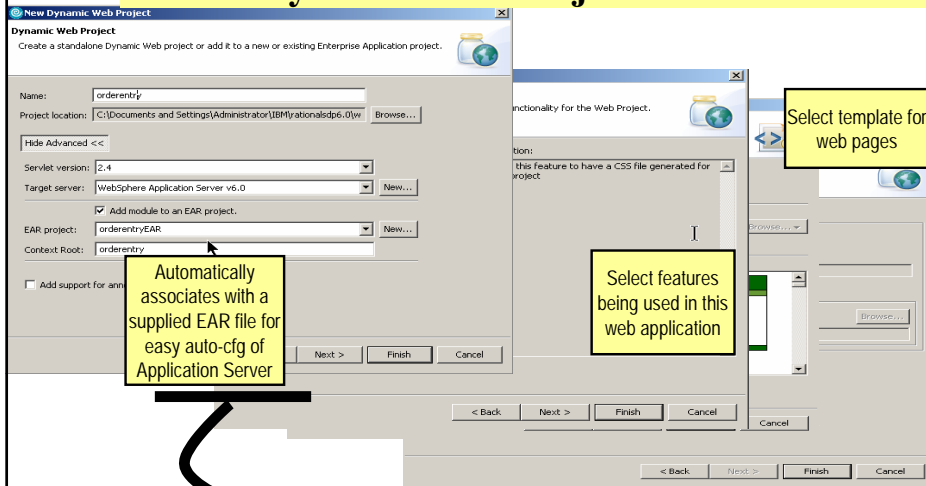
- **One of the project types in WDS**
  - ▶ With its own perspective
  - ▶ With its own "new" wizard
  - ▶ With its own set of tools
- **Created with J2EE folder layout**
- **Created with simple web.xml file**
  - ▶ Automatically updated as resources are created
  - ▶ Has specialized web.xml editor



# Web projects

iSeries AD, IBM Toronto

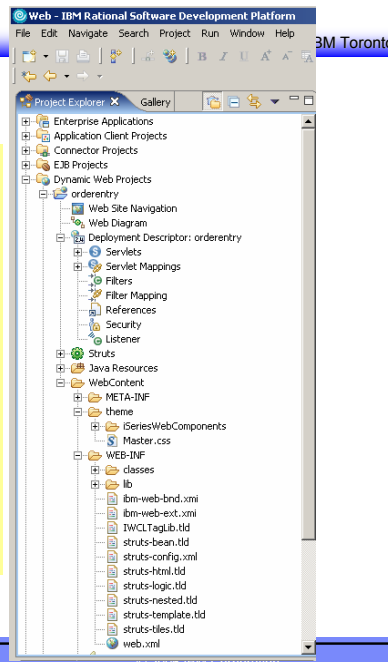
## New Dynamic Web Project wizard



# Web projects

**To create a new web project  
Select New Dynamic  
Web Project in Web  
perspective**

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



# AGENDA

- WebSphere Development Studio WDS V5R4
- e-business primer
  - AD Model, traditional and web
  - Web Applications
  - Mixing Java and RPG
- Introducing WebTools for IBM System i5
  - Web projects
  - Interaction wizard
  - Page designer
  - Web Service wizard
  - JCA wizard



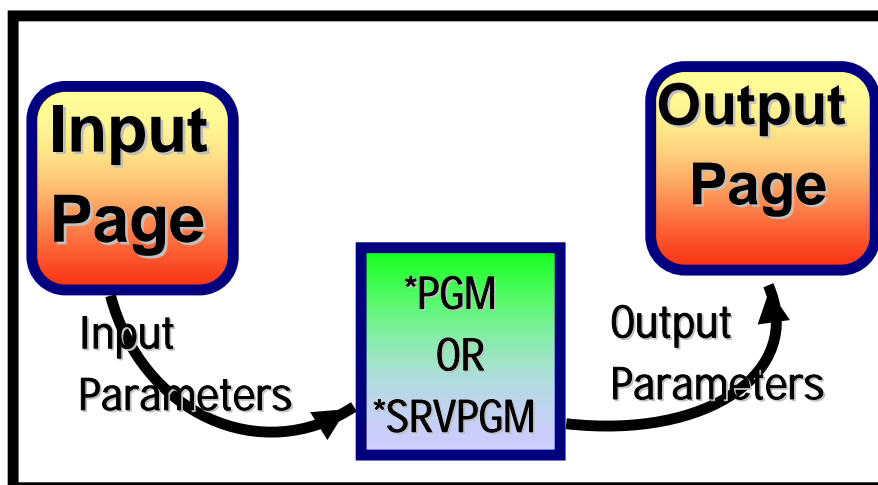
## Web Tool for iSeries

iSeries AD, IBM Toronto

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

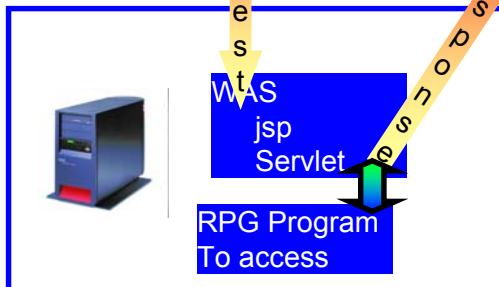
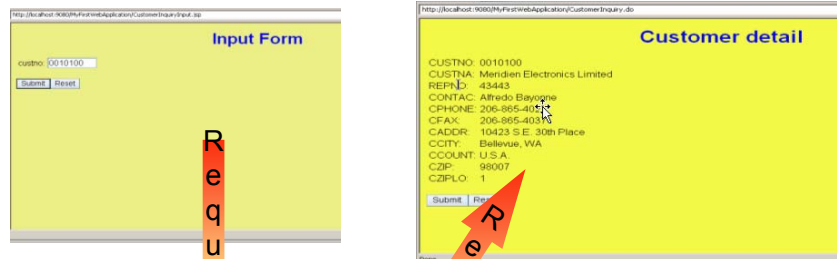
## An interaction

iSeries AD, IBM Toronto



## Build this simple interaction

iSeries AD, IBM Toronto



## Using the Interaction wizard

iSeries AD, IBM Toronto

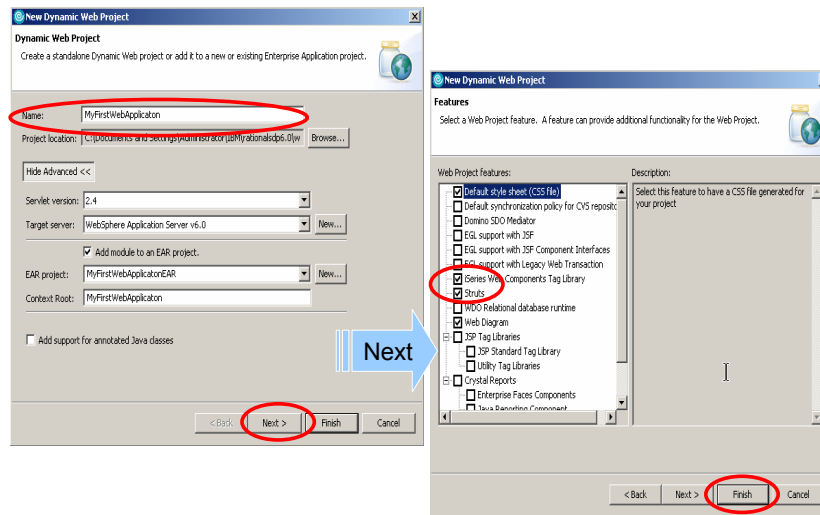
### Steps involved

- + 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



# Create a Web project

iSeries AD, IBM Toronto



# Using the Interaction wizard

iSeries AD, IBM Toronto

- ✓ Create a Webproject
- ✚ Create Runtime configuration ←
- ✚ iSeries server
- ✚ User id/password
- ✚ Library list setup
- ✚ Create Interaction
- ✚ Input screen
- ✚ Program interface
- ✚ Output screen

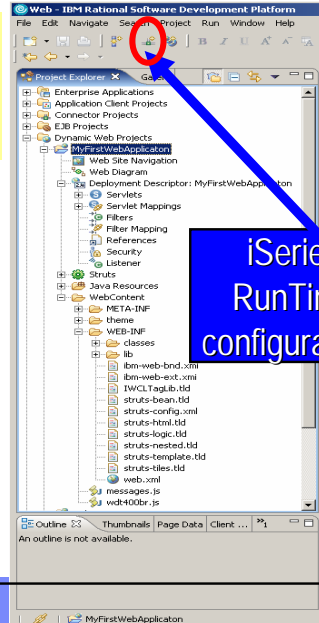
# Runtime configuration

iSeries AD, IBM Toronto

Created Dynamic Web project:  
→ MyFirstWebapplication  
Now create the runtime environment

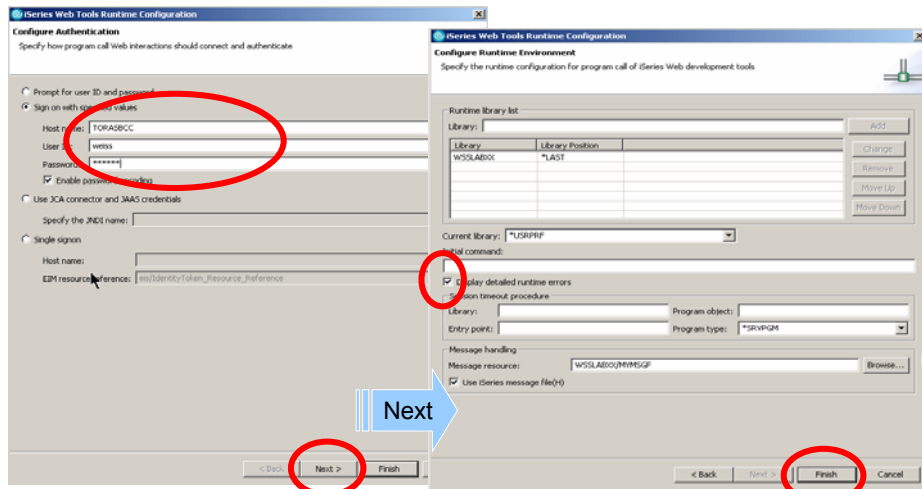
• Information used at runtime by all generated Web servlets in this Web project

- ▶ Name of host containing \*PGM or ILE Procedure
- ▶ User ID and password for signing on
- ▶ Library list to set for \*PGM/Proc job



# iSeries Web Tools

## ▶ iSeries Web Host Information Wizard



# Using the Interaction wizard

iSeries AD, IBM Toronto

- ✓ Create a Webproject
- ✓ Create Runtime configuration
  - ✓ iSeries server
  - ✓ User id/password
  - ✓ Library list setup
- ✚ Create Interaction ←
- ✚ Input screen
- ✚ Program interface
- ✚ Output screen

# WDS*c*: iSeries Web Tools

iSeries AD, IBM Toronto

## ► iSeries Web Interaction Wizard

```

To enable the iSeries Webtooling detailed runtime
<param-name>WDT_FLGERRORDetails</param-name>
<param-value>true</param-value>
</context-param>
<description>
Program type:</description>
<param-name>WDT_PROGRAMTYPE</param-name>
<param-value>*SRVPGM</param-value>
</context-param>
<description>
Message resource for message
<param-name>WDT_MSORESOURCE</param-name>
<param-value>WSSLABXX/MYMSGF
</context-param>
<description>
Use iSeries Message File</de
<param-name>WDT_USEISERIESMS
<param-value>true</param-value>
</context-param>
<description>
iSeries program call host name</description>
<param-name>WDT_HOSTNAME</param-name>
<param-value>TOPASBCC</param-value>
</context-param>
</context-param>

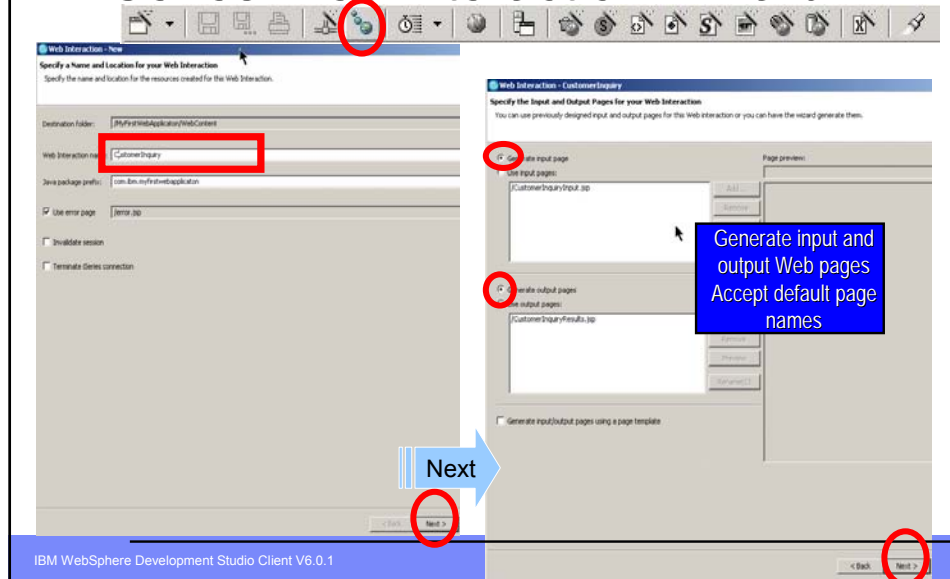
```

web.xml content changed:  
Added iSeries server  
information

# WDS: iSeries Web Tools

iSeries AD, IBM Toronto

## ► iSeries Web Interaction Wizard



# Web Interaction Wizard

iSeries AD, IBM Toronto

- Two modes to interaction wizard:
  - **Generate input/output Web pages**
    - **Given the parameter description of the API to call** (In this sample we are using this mode)
  - **Generate mappings**
    - **Given the input/output pages**
    - **Given the parameter description of the API to call**
    - **Given the mappings**
      - ✓ between input parms & input fields
      - ✓ between output parms & output fields

# iSeries Web Interaction Wiz

iSeries AD, IBM Toronto

**Get call interface from source member**

- ▶ Press import
- ▶ Select member
- ▶ PCML interface description gets generated

# RPG IV Program GETDATA

iSeries AD, IBM Toronto

**NOTE: Parameters to be passed**

```

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

# iSeries Web Interaction Wiz

iSeries AD, IBM Toronto

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

Program call definitions

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

**Edit Program**

Program alias: GETDATA

Program object: GETDATA

Library: \*LIB

Program type: \*PGM

Entry point:    Browse...

CCSID of entry point:    Browse...

Return type: void

Parse order: CSTRUC FEEDBACK

Thread safe: False

Source location:    View...

Associate this program with the interaction

Changes applied.    OK    Cancel

IBM WebSphere Development Studio Client V6.0.1    © 2006 IBM Corporation

# iSeries Web Interaction Wiz

iSeries AD, IBM Toronto

Web Interaction - CustomerInquiry

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

Program call definitions

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

**Edit Parameter**

Parameter name: CUSTNO1

Data type: character

Structure name:

Length: 7

Precision:

Count:

Usage: Input

Initial value:

Advanced...

Specify database reference field    Specify...    Synchronize

Show database field definition    Show...

OK    Cancel

Next

IBM WebSphere Development Studio Client V6.0.1    © 2006 IBM Corporation

ibm.com/redbooks | International Technical Support Organization

# Input page

Select input parameters to prompt user for, in input page

Immediately see results of tailoring parameter and page attributes

Tailor attributes of generated prompt per parameter

Tailor attributes of overall input page

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation

ibm.com/redbooks | International Technical Support Organization

# Output page

iSeries AD, IBM Toronto

Select output parameters to display in output page

Immediately see results of tailoring parameter and page attributes

Tailor attributes of generated prompt per parameter

Tailor attributes of overall output page

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation

## You are done! What's Next?

iSeries AD, IBM Toronto

- **What did you do so far?**
  - ✓ Created your Files on the iSeries
  - ✓ Created your RPG Program
  - ✓ Created an 'iSeries Web Interaction' to link your program parameters to the input page and output page
    - ✓ The input page was generated for you
    - ✓ The output page was generated for you
- **Next?**
  - ▶ Run locally to test your application Or
  - ▶ Publish all your files to the iSeries server

## WDS*c*: Run On Server

iSeries AD, IBM Toronto

- ▶ **Run On Server**
  - **Now this is VERY COOL!**
- ▶ **When ready to test your Web app**
  - **Right click on initial html or jsp file**
  - **Select "Run... --> Run on Server"**
  - **Wait for the magic...**
- ▶ **Your Web application will run!**
  - **Opens Server perspective**
    - ▶ Publishes it to built-in copy of WAS
    - ▶ Starts built-in copy of WAS
    - ▶ Brings up a Web Browser
    - ▶ **Runs your application!!**



# iSeries Web Interaction Wiz

iSeries AD, IBM Toronto

Generated files

Ready to test

Use "Run on Server" to test

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation

# Testing the Interaction

iSeries AD, IBM Toronto

Input page in the built-in browser

Use Page Designer to finesse generated pages

Using WAS test environment Console entries can be viewed in console view

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation

# Testing the Interaction

iSeries AD, IBM Toronto

The screenshot displays the IBM WebSphere Development Studio Client V6.0.1 interface. On the left, a Navigator pane shows a project structure for 'MyFirstWebApplication'. The central Web Browser window displays a 'Customer detail' page with the following data:

- CUSTNO: 0010100
- CUSTNA: Meridien Electronics Limited
- REPNO: 43443
- CONTACT: Alfredo Bayonne
- CPHONE: 206-865-4027
- CFAX: 206-865-4037
- CADDR: 10423 S.E. 30th Place
- CCITY: Bellevue, WA
- CCOUNT: U.S.A.
- CZIP: 98007
- CZIPLO: 1

Below the data is a 'Submit' button and a 'Reset' button. A blue callout box points to the data, stating 'Application fetched data from database'. Another blue callout box points to the buttons, stating 'Output page in the built-in browser'. At the bottom, a console window shows server logs for 'myserver2'.

IBM WebSphere Development Studio Client V6.0.1 © 2006 IBM Corporation

# WebSphere Test Environment

iSeries AD, IBM Toronto

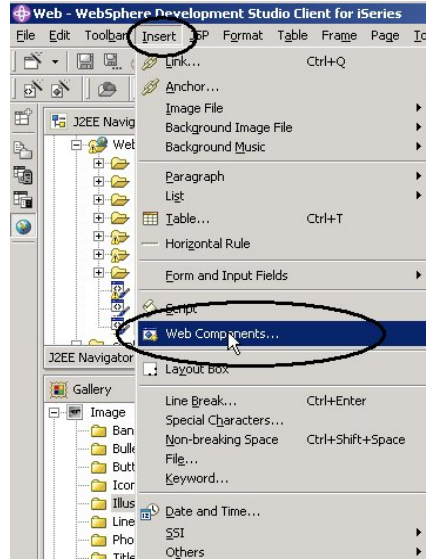
## A full copy of WAS 6.0 is embedded in the IDE

- ▶ 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

# Web UI with Web components

iSeries AD, IBM Toronto

- **Web components:**  
**Visual Control Tags (VCT)**
- **Tag library defined by:**  
▶ [www.sun.com](http://www.sun.com)
- **Extendable**



# Web components ...

iSeries AD, IBM Toronto

◆ **With iSeries built in capabilities!**

The screenshot displays two dialog boxes. On the left is the 'Web Components' dialog, showing a list of components: Text Entry, Label, Button, Image Button, Combo Box, Selection Box, Check Box, Text Area, Radio Button Group, Image, Hyperlink, and Table. A red arrow points from the 'Image Button' component to the 'Attributes' dialog. On the right is the 'Attributes' dialog for 'IWCL:WTEXTENTRY'. It has tabs for 'General', 'Data', 'Events', and 'Styles'. The 'General' tab is active, showing fields for Name, Input type, Size, Initial state, Orientation, Initial value, Tool tip, Access key, Tab index, Label, and Label position. A blue box labeled 'Text Entry attributes' is overlaid on the 'General' tab. Below it, another 'Attributes' dialog is shown, also for 'IWCL:WTEXTENTRY', with the 'Data' tab active. This dialog shows 'Length', 'Decimal places', 'Get data attribute from iSeries database' (checked), 'Database Reference', 'Get locale-sensitive values from', 'Decimal symbol', 'Formatting', and 'Validator type'. Red circles highlight the 'Get data attribute from iSeries database' option and the 'Validator type' dropdown menu.

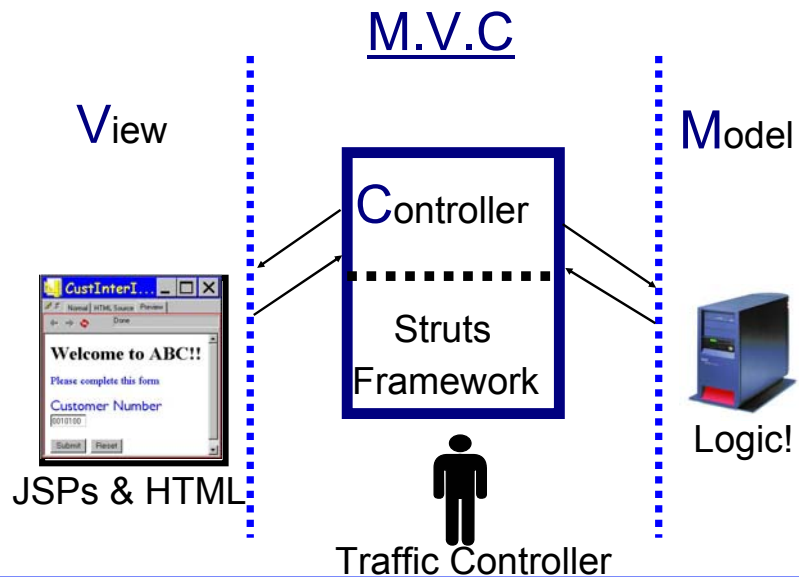
# Struts

iSeries AD, IBM Toronto

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

# Web Wizard Struts based ...

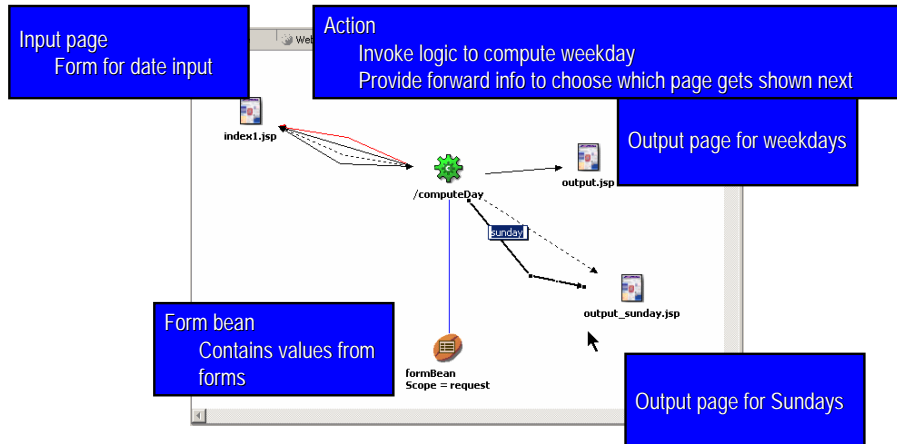
iSeries AD, IBM Toronto



# Simple Struts application

iSeries AD, IBM Toronto

## Web diagram of a simple Struts application



Input is date, output is day of week for that date  
or if the day of week is a **Sunday**, output is a special Sunday page

# 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">
  <forward name="success" path="/output.jsp">
  </forward>
  <forward name="failure" path="/index1.jsp">
  </forward>
  <forward name="sunday" path="/output_sunday.jsp">
  </forward>
</action-mappings>
    
```

Define in the Struts config file

Which page to show  
The action just returns different

It Is A Sunday Wonderful

# Page designer tool

iSeries AD, IBM Toronto

## •Page Designer for JSPs & HTML

- Frame wizard
- WebArt designer
- AnimatedGIF designer
- Spell Checker
- HTML validator
- HTML to XHTML converter
- Attribute dialog
- Link utilities
  - o fix broken links
  - o Convert links

# PageDesigner

iSeries AD, IBM Toronto

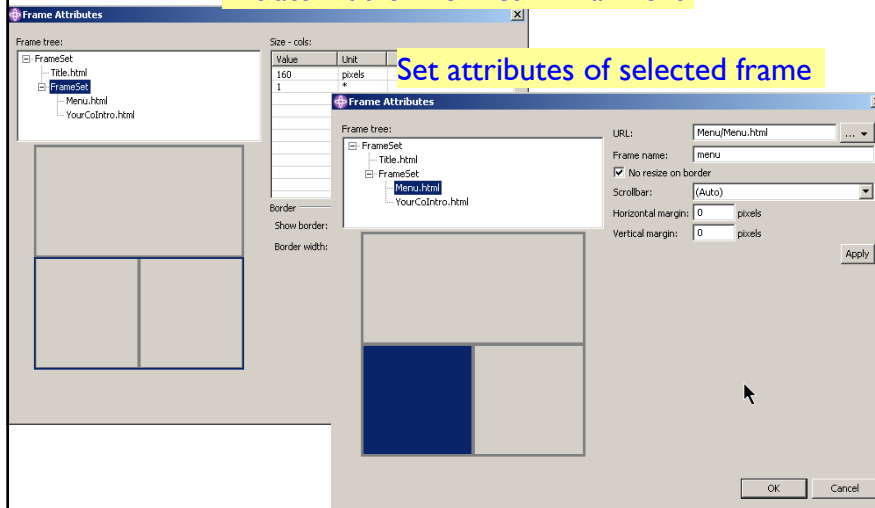
The screenshot shows the IBM WebSphere Page Designer interface. On the left is the **Project Navigator** showing a tree view of files and folders. The main workspace is titled **IBM WebTool CINNT Exam** and contains an **Input Form** with a text input field and a button. Below the workspace is a **Design/Source/Preview** tab bar. At the bottom, there is an **Attribute editor** and a **Link view** showing a **Link view** diagram. Annotations include a blue arrow pointing to the Project Navigator labeled **File View**, a red box around the workspace labeled **Three Frames**, and a blue arrow pointing to the Link view labeled **Different Views**.

# Frame tools

iSeries AD, IBM Toronto

Set attributes of selected Frameset

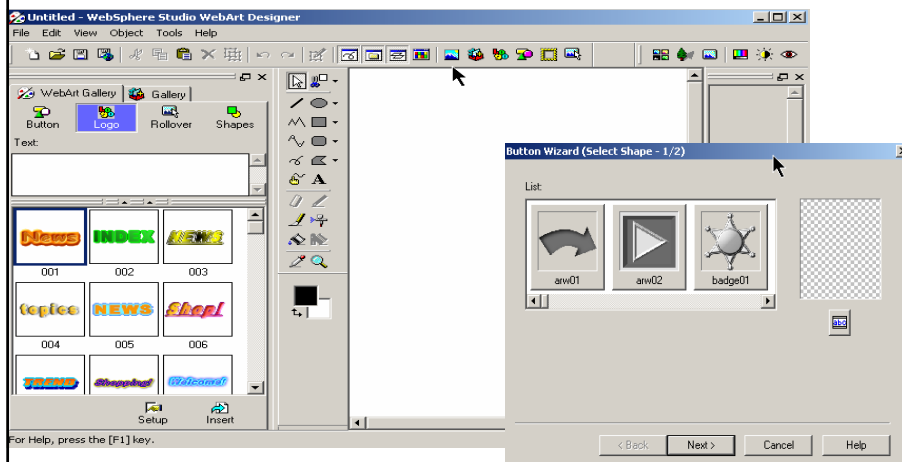
Set attributes of selected frame



# WebArt Designer

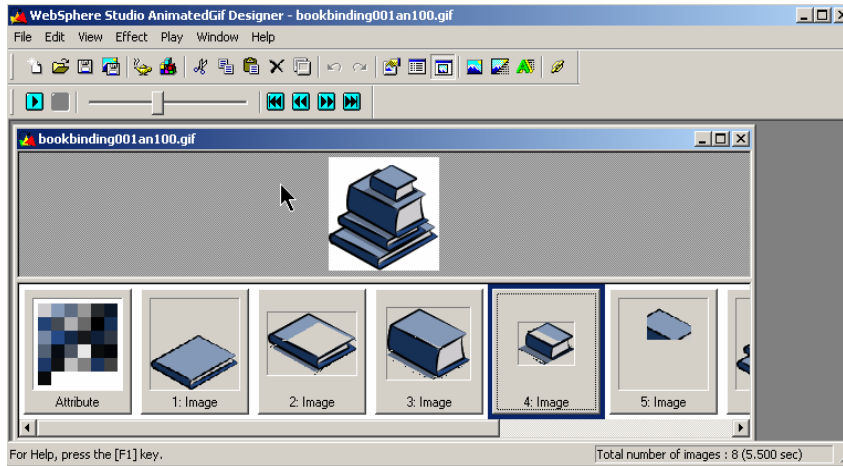
iSeries AD, IBM Toronto

Wizards for Logo/Button/PhotoFrame/Rollover



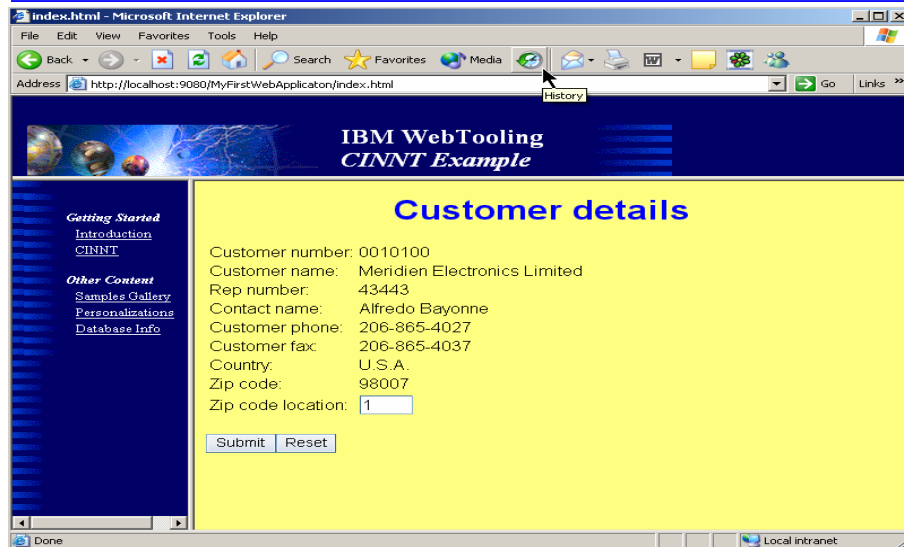
# Animation GIF designer

iSeries AD, IBM Toronto



# Running the improved application

iSeries AD, IBM Toronto





## AGENDA

iSeries AD, IBM Toronto

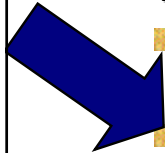
- WebSphere Development Studio WDS V5R4
- e-business primer
  - AD Model, traditional and web
  - Web Applications
  - Mixing Java and RPG
- Introducing WebTools for IBM System i5
  - Web projects
  - Interaction wizard
  - Page designer
  - Web Service wizard
  - JCA wizard



## Web Tool for iSeries

iSeries AD, IBM Toronto

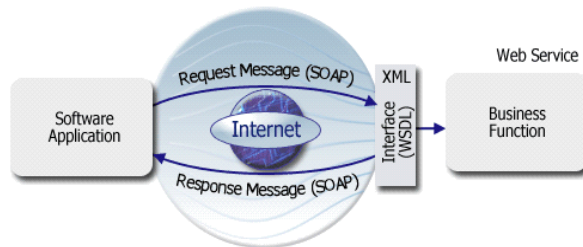
- ✚ Wizards to help create iSeries RPG/COBOL web applications
  - Interaction wizard to build an RPG web application
  - WebService wizard to build an iSeries WebService
  - JCA wizard to build an iSeries Java Connector Architecture (JCA) connector



# Webservice overview

iSeries AD, IBM Toronto

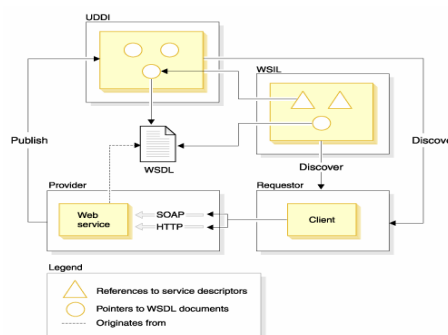
- Industry standard
- A way to interchange data in a non-proprietary way
- Self-contained software components, with a well-defined interface
- The interface describes a set of operations accessible over the internet
- UDDI Universal Description, Discovery and Integration
- WSIL Web Services Inspection Language
- WSDL Web Services Description Language
- SOAP Simple Object Access Protocol



# Web Service and IBM System i5

iSeries AD, IBM Toronto

- Describe the Program interface that provides the Service
- New in 6.0.1
- Web Service Wizard in WDSi generates
- for a given program source member on IBM System i5:
  - WSDL to describe the service interface
  - Test client to test the Web Service



# Create Web Service for RPG program

iSeries AD, IBM Toronto

→ In RSE:

- select member
- choose Web Service wizard

# Create Web Service for RPG program

iSeries AD, IBM Toronto

```

<program name="CUSTINFO" parseorder="OUT_STATUS 0_DS"
  path="/QSYS.LIB/1.LIB/1.LIB/CUSTINFO.PGM">
  <data length="15" name="IN_LNAME" type="char" usage="input"/>
  <data length="4" name="IN_CID" type="char" usage="input"/>
  <data length="7" name="OUT_STATUS" type="char" usage="output"/>
  <data name="0_DS" struct="OUT_DS" type="struct" usage="output"/>
</program>

```

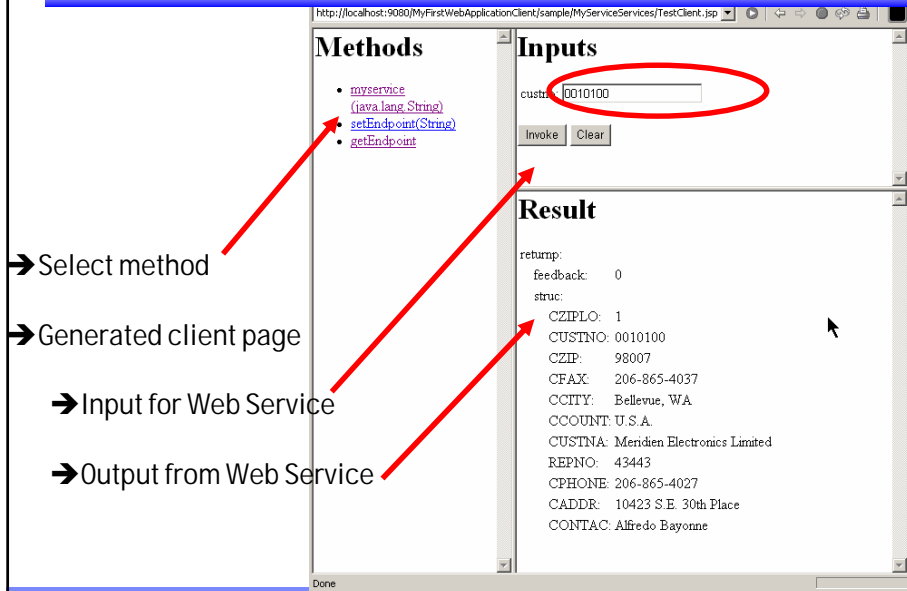
```

<wsdl:types>
  .....
  <complexType name="CUSTINFOInput">
  <sequence>
  <element name="IN_LNAME" nillable="true" type="xsd:string"/>
  <element name="IN_CID" nillable="true" type="xsd:string"/>
  </sequence>
  </complexType>
  <complexType name="OUT_DS">
  <sequence>
  <element name="OUT_CID" nillable="true" type="xsd:string"/>
  <element name="OUT_FNAME" nillable="true" type="xsd:string"/>
  <element name="OUT_LNAME" nillable="true" type="xsd:string"/>
  <element name="OUT_ADDRESS" nillable="true" type="xsd:string"/>
  <element name="OUT_CITY" nillable="true" type="xsd:string"/>
  .....
  </sequence>
  </complexType>
  .....
  </schemas>
</wsdl:types>

```

# JSP Test client for Web Service

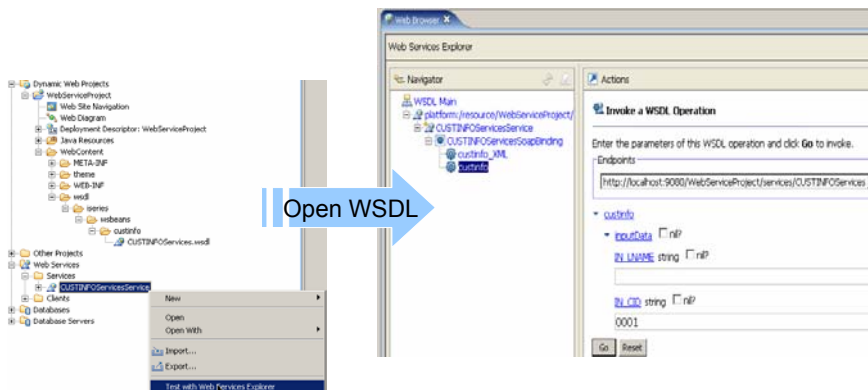
iSeries AD, IBM Toronto



- Select method
- Generated client page
- Input for Web Service
- Output from Web Service

## Test the Web service - Web Service Explorer

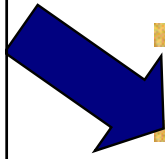
- Right-click on the WSDL document.
- Web Services explorer opens WSDL document.
- Select an operation from the Web Service (custinfo), enter program input, Go!



# Web Tool for iSeries

iSeries AD, IBM Toronto

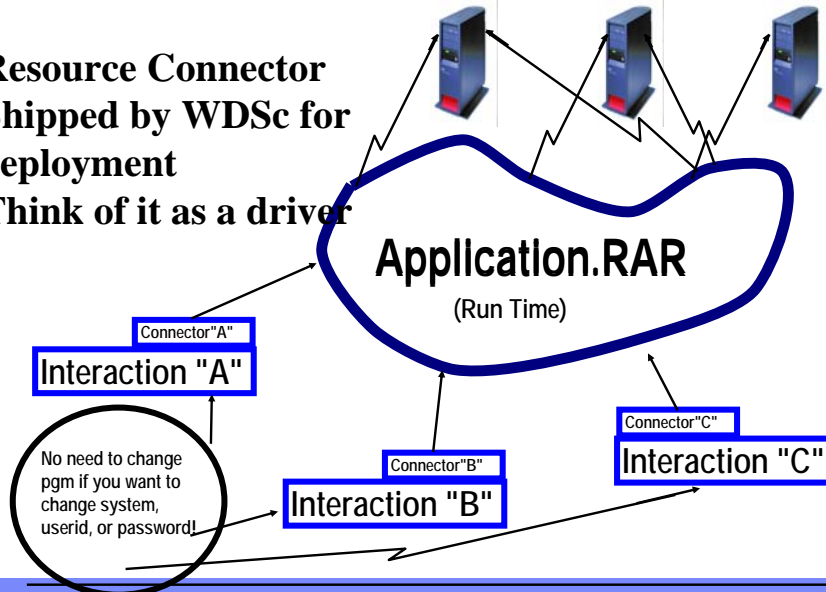
- ✚ Wizards to help create iSeries RPG/COBOL web applications
  - Interaction wizard to build an RPG web application
  - WebService wizard to build an iSeries WebService
  - JCA wizard to build an iSeries Java Connector Architecture (JCA) connector



# What is JCA Connectors?

iSeries AD, IBM Toronto

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



# Configuring a JCA connector

iSeries AD, IBM Toronto

- First you add the connector to the server configuration (all JCA info can be defined either through admin console)
- The userid(s) you want to use for program calls using JCA are defined via a JAAS authentication entry
- You map JAAS entry to a JNDI name.

**1**

Resource Adapter Name	Display Name
iSeriespgmcall	iSeries Program Call Resource Adapter

**2**

Alias	User ID
JAASAlias	ericdp
JAASAlias2	ericdp2

**3**

Name: \* JNDI1  
 JNDI name: \* eis/JNDI1  
 Container-managed authentication alias: JAASAlias  
 Component-managed authentication alias: JAASAlias2

# Using JCA connector

iSeries AD, IBM Toronto

- 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

Series host name: iSeriesAD  
 User ID: ericdp  
 Password: \*\*\*\*\*

Prompt for iSeries user ID and password

Runtime library list:

Library	Library Position
WL150	*LAST
GUIDCPW55	*LAST

Current library: \*USERPRF

Initial commands:

Display detailed runtime errors

Use program call JCA connector

Specify the JNDI name: MyFirstWebApplication

# AGENDA

iSeries AD, IBM Toronto

- WebSphere Development Studio WDS V5R4
- e-business primer
  - AD Model, traditional and web
  - Web Applications
  - Mixing Java and RPG
- Introducing WebTools for IBM System i5
  - Web projects
  - Interaction wizard
  - Page designer
  - Web Service wizard
  - JCA wizard
  - JSF - JavaServer Faces



## Java Server Faces Overview

### What JSF?

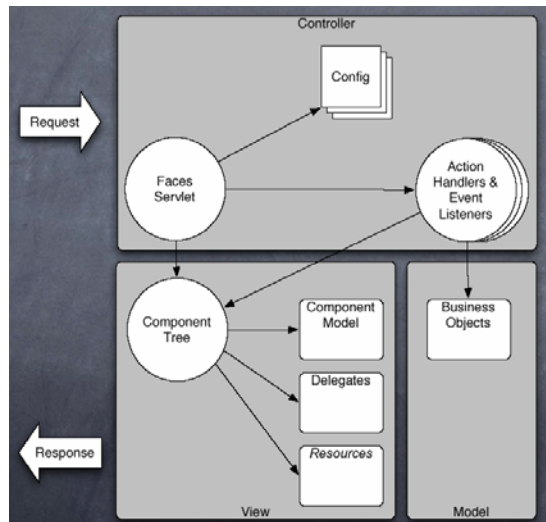
- ⦿ Defines the Programming Model
- ⦿ Reusable Components
- ⦿ Builds HTML from Components
- ⦿ Validation & Conversion
- ⦿ Component Tree

### Why JSF?

- ⦿ Ease/Simplify Development
- ⦿ Support from/by tools
- ⦿ Events easily tied to server side code
- ⦿ Ease of extension
- ⦿ Manage UI state



## Java Server Faces MVC on the Web



- Controller
  - FacesServlet
  - Lifecycle
  - Application
  - NavigationHandler
  - ActionListener
  - Conversion and Validation
- View
  - RenderKit
  - Renderers
- Model
  - JavaBeans
  - EJBs
- Extension Points
  - View & Controller are extensible



## Java Server Faces Custom Component Examples

Activity	Start	End
Project Summary	Jan 20, 2004	Feb 1, 2005
Call Requirements	Jan 10, 2004	Jan 10, 2004
Talk to customers	Jan 10, 2004	Jan 10, 2004
Contact customer	Jan 10, 2004	Jan 10, 2004
Contact customer	Jan 12, 2004	Jan 12, 2004
Write up requirements	Jan 10, 2004	Jan 10, 2004
Review requirements	Jan 22, 2004	Jan 22, 2004
Marketing Specification	Jan 27, 2004	Jan 27, 2004
First Draft Specification	Jan 27, 2004	Jan 27, 2004
Second Draft Specifi...	Jan 4, 2004	Jan 4, 2004
Engineering Review	Jan 9, 2004	Jan 9, 2004

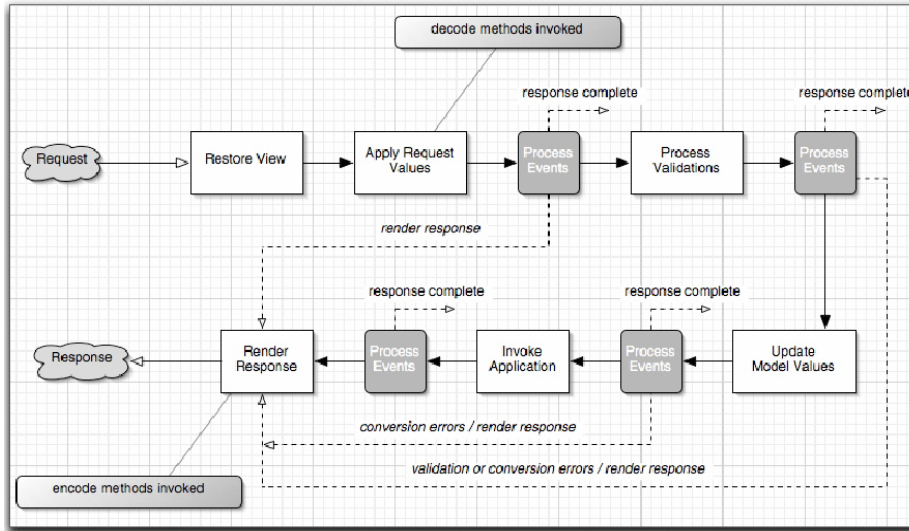
# of Items	Task Name
4	Customer Request
2	Automatic Processing
4	1st Level Support
16	Expert Support
2	Dispatching
0	Genie Support

Task Name	Count
Customer Request	4
Automatic Processing	2
1st Level Support	4
Expert Support	16
Dispatching	2
Genie Support	0

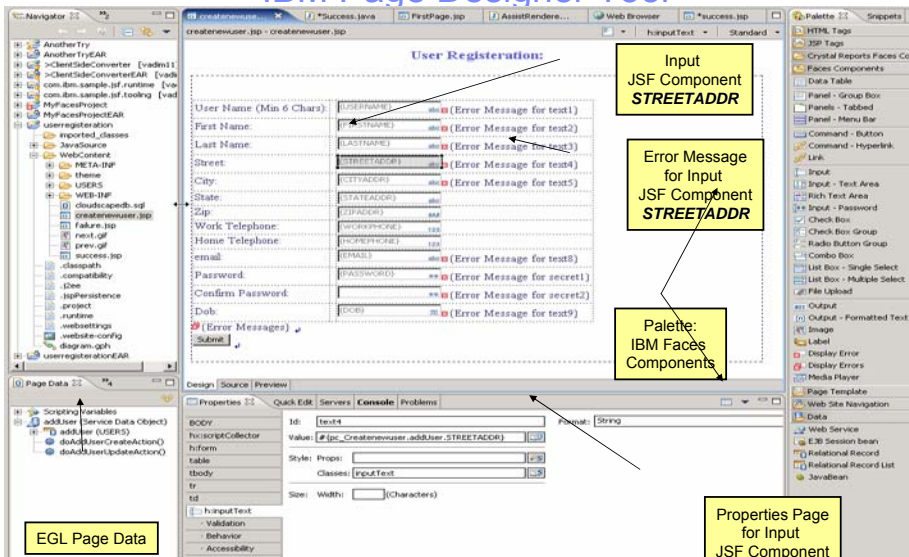




# Java Server Faces Life Cycle



# IBM Page Designer Tool



## SUMMARY

iSeries AD, IBM Toronto

### ▶ System i5 web tools, at a glance

- **Tools optimized for System i5 developers!**

- ▶ **Web Interaction wizard**

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

- ▶ **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?

iSeries AD, IBM Toronto

### ▶ Information Sources

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



# Disclaimer

iSeries AD, IBM Toronto

## Acknowledgement:

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

## 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
@server	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.