

WebSphere Studio for AS/400

IBM Toronto Development Laboratory

Claus Weiss weiss@ca.ibm.com

WDT/400

New World

New Servers

New Tools

**Common
October 2001**

www.ibm.com/software/ad/wdt400



Disclaimer

Web	Java	RPG
		CBL

AS/400 AD, IBM Toronto

Acknowledgement:

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

Disclaimer:

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

Reproduction:

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

AGENDA

Web	Java	RPG
		CBL

AS/400 AD, IBM Toronto

- **What is WDT/400?**
- **AD Models**
 - ▶ Traditional
 - ▶ Web
- **Web Stuff**
 - ▶ HTML
 - ▶ JSPs
- **Mixing Java and RPG**
- **Introducing WDT/400 : Studio/400**
 - ▶ RPG Example
 - ▶ AS/400 Design time controls (DTCs)
 - ▶ Web Interaction
 - ▶ Publishing
 - ▶ Remote debug
- **Conclusion**

WebSphere Development Tools for AS/400

Web	Java	RPG CBL
-----	------	------------

AS/400 AD, IBM Toronto

Includes

CODE/400

✓ For RPG, COBOL, CL, C, C++, DDS

VisualAge RPG

✓ For RPG GUI

VisualAge for Java for AS/400

✓ Pro plus Enterprise Toolkit for AS/400

WebSphere Studio For AS/400

WebSphere Studio For AS/400

✓ Standard plus Affinity Toolkit for AS/400

WebFacing (the new tool on the block)

www.ibm.com/software/ad/wdt400

WebSphere Development Tools/400



**WebSphere
Studio**

**VisualAge for
Java**

CODE/400

**VisualAge
RPG**

Common Installation, JDK, Communications, Documentation

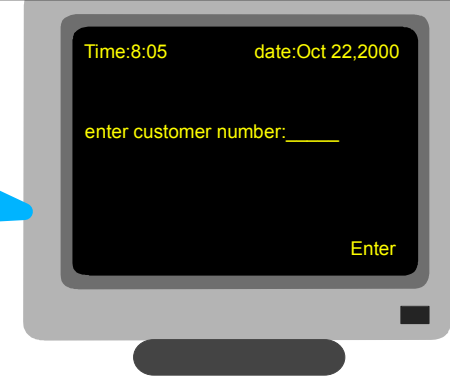
RPG Model Today



RPG Source

5250 Screens

```
***** Beginning of data *****
D*-----
C  *LIKE      DEFINE    FIRST      LAST
C             EVAL      LAST=' FARR '
C             READ      RECORD
C             :
C             MOVE      'AGE WAS---->' AGETEXT
C             EXFMT     FMT5250
C             :
C             MOVE      *ON          *INLR
C*-----
O             :
```



AS/400 DB

1. Create your DB files
2. Create your screens (i.e UI)
3. Create your RPG code to drive DB file and record formats

Web Model

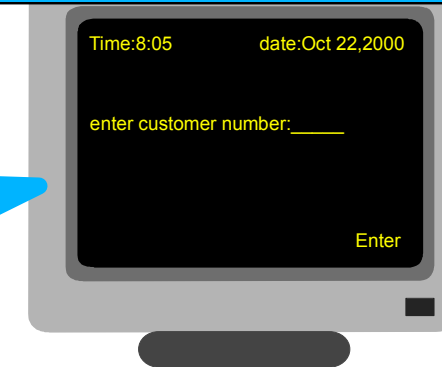
Web Java RPG
CBL

AS/400 AD, IBM Toronto

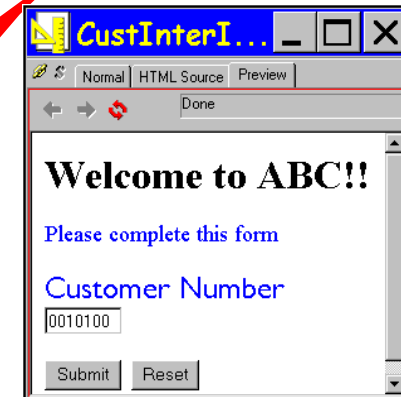
**RPG
IV**

RPG Source

5250 Screens



HTML & JSPs



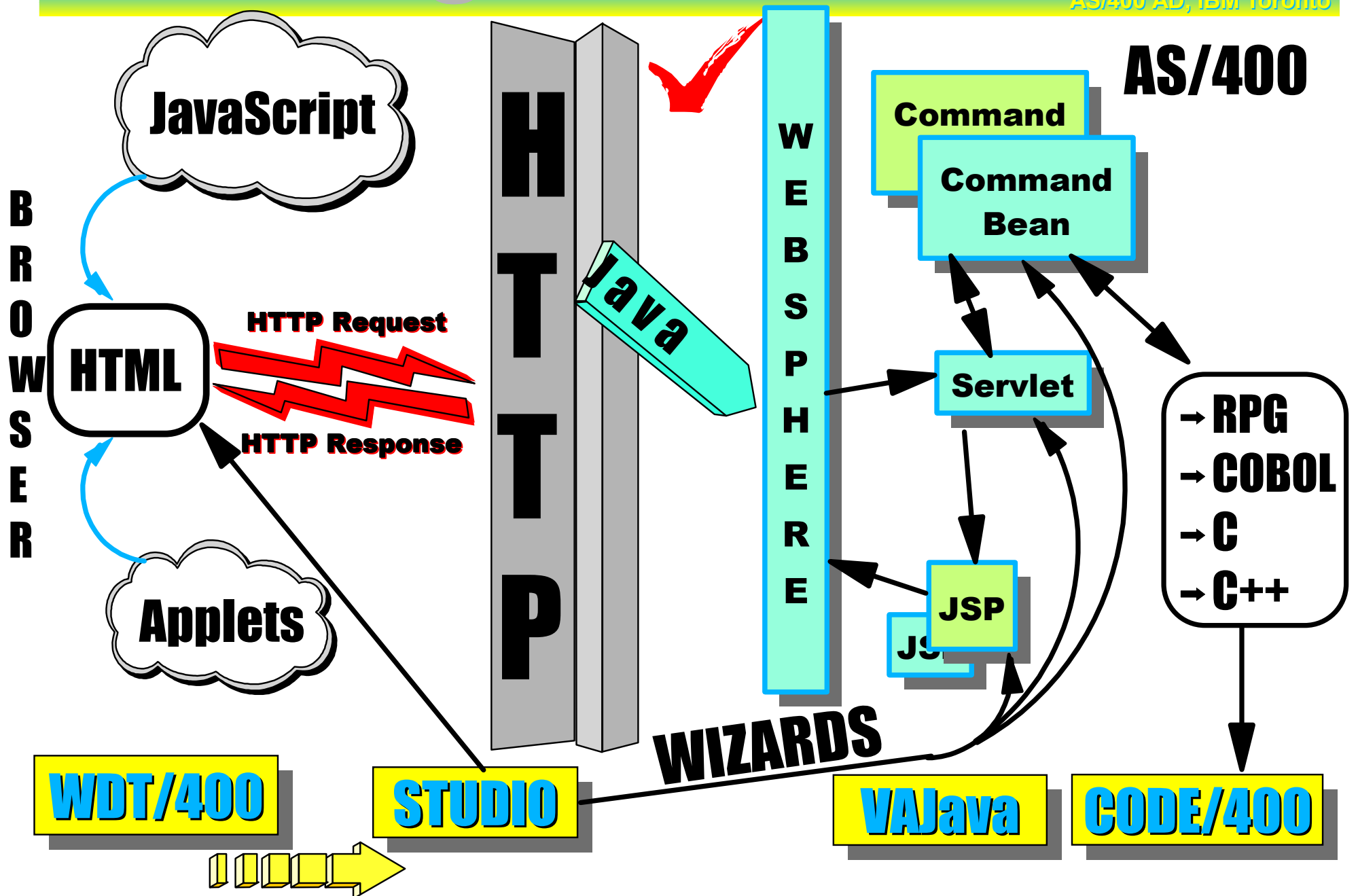
```
***** Beginning of data *****
D*-----
C  *LIKE      DEFINE    FIRST      LAST
C              EVAL     LAST=' FARR'
C              READ     RECORD
C              :
C              MOVE     'AGE WAS---->' AGETEXT
C              EXFMT    FMT5250
C              :
C              MOVE     *ON              *INLR
C*-----
O              :
```

1. Use Web Browser for UI
2. Replace 5250 with HTML & JSPs
3. Use servlet to drive UI and access DB/400 or RPG/COBOL logic

Big Picture

Web	Java	RPG
		CBL

AS/400 AD, IBM Toronto



HTML FORMS

HTML and JSP Forms

Web	Java	RPG
		CBL

AS/400 AD, IBM Toronto

- **Allows you to solicit user input**
- **Can build HTML document that contains:**
 - checkboxes
 - radio buttons
 - push buttons
 - entry fields
 - selectable lists
 - ... others
- **Collect data from user**
 - ▶ **send to server program**
 - The **FORM** tag specifies the name of the program
 - Historically a CGI-bin program
 - New option is a Java Servlet

Anatomy of a FORM

Anatomy of a FORM

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>A Simple Input Form</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

server-side CGI-bin program or Java servlet to call when
SUBMIT button pressed

```
<FORM action="http://localhost/servlet/MyServlet"  
method="post">
```

how to call server when SUBMIT pressed

body: one or more input controls plus regular html

```
<INPUT type="submit"> "SUBMIT" button  
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Example of a Form

```
<FORM action="http://localhost/servlet/MyServlet"
  method="post">
  Name <INPUT type="text" name="name"><BR>
  Age <INPUT type="text" name="age" size="3" maxlength="3"><BR>
  Country <SELECT name="country">
    <OPTION selected>Canada</OPTION>
    <OPTION>Mexico</OPTION>
    <OPTION>United States</OPTION>
  </SELECT> <BR>
  e-mail <INPUT size="30" type="text" name="email"><BR>
  <BR>
  <INPUT type="radio" name="sex" value="M" checked>Male
  <INPUT type="radio" name="sex" value="F">Female<BR><BR>
  <INPUT type="checkbox" name="mail" checked>e-mail me<BR><BR>
  <INPUT type="submit" value="Register">
  <INPUT type="reset" value="Reset">
</FORM>
```

entry field

selectable list

entry field

radio
buttons

check box

SUBMIT button

RESET button

<http://www.w3.org/TR/REC-html40/>

Example of a Form

SUBMIT button.
This will indicate
which Servlet to
call on the
server!

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

- Browser Title:** A Simple Input Form ...
- Menu Bar:** File, Edit, View, Favorites, Tools
- Navigation Bar:** Back, Forward, Stop, Refresh, Home
- Address Bar:** Address E:\WebSphere\St
- Form Fields:**
 - Name: (labeled as **entry field**)
 - Age:
 - Country: (labeled as **selectable list**)
 - e-mail:
- Form Controls:**
 - Gender: Male Female (labeled as **radio buttons**)
 - Subscription: e-mail me (labeled as **check box**)
 - Buttons: Register, Reset (labeled as **RESET button**)
- Taskbar:** My Computer

Java Server Pages

JSPs: What are they?

Web	Java	RPG CBL
-----	------	------------

AS/400 AD, IBM Toronto

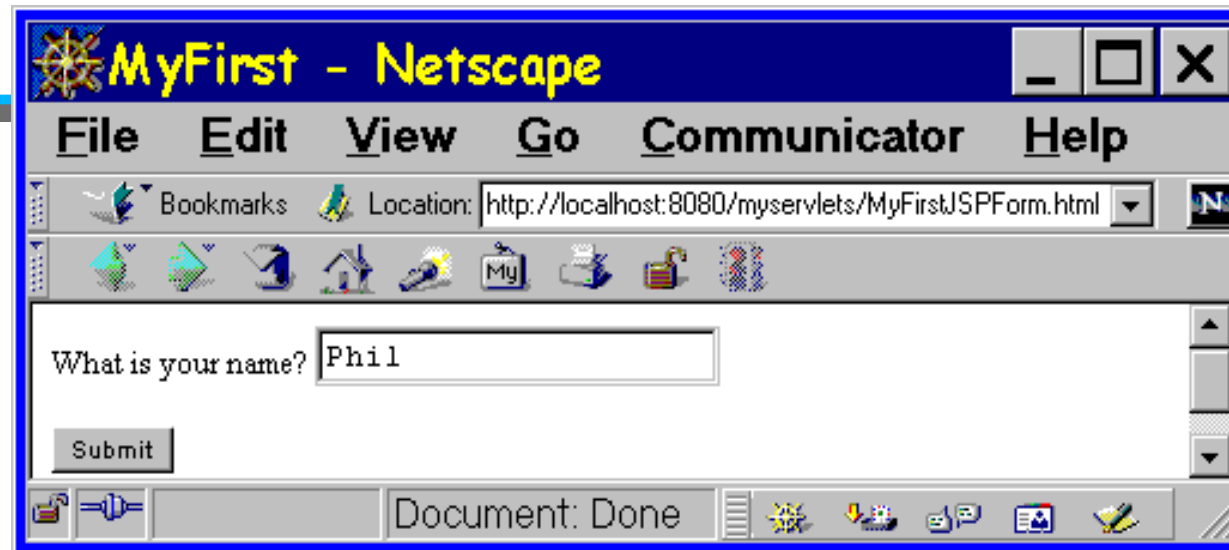
- **Not a good idea to merge Servlet Java code with HTML tags**
 - ▶ **Mixing skills**
 - ▶ **Can't use HTML WYSIWYG tools**
- **JavaServer Pages to the rescue!**
 - ▶ **Contain HTML, embedded Java, & JSP tags**
 - ▶ **Contain "holes" to be filled in at runtime**
- **New way to use servlets**
 - ▶ **Are compiled into servlets themselves**
 - ▶ **Are callable by other servlets**
- **App Servers must support them**
 - ▶ **IBM's WebSphere App Server does**

JSP Example

\myservlets\MyFirstJSPForm.html

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

Call the JSP when
SUBMIT pressed



JavaServer Pages Example

Web	Java	RPG
		CBL

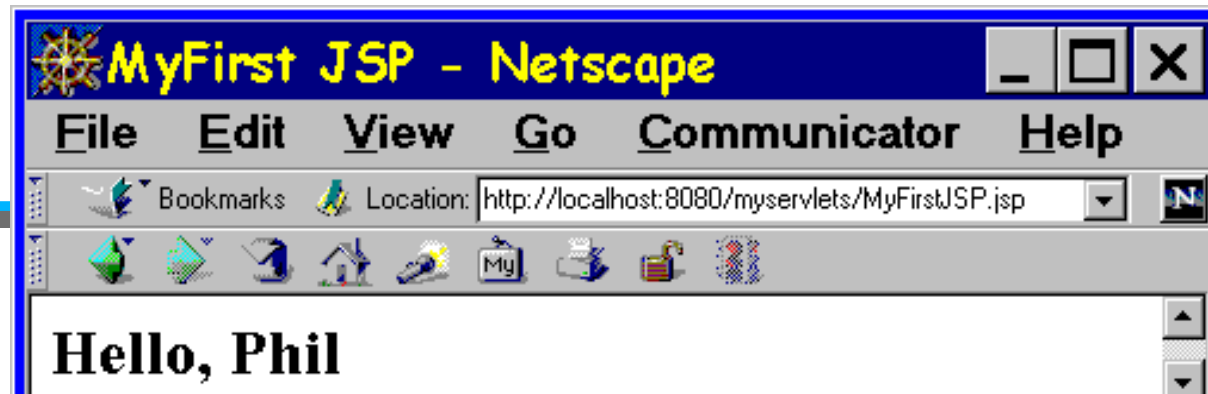
\myservlets\WEB-INF\jsp\MyFirstJSP.jsp

IBM Toronto

```
<HTML>
<HEAD><TITLE> MyFirst JSP </TITLE></HEAD>
<BODY>
<h1>
<% if
  (request.getParameter("name")
   == null) {%>
Hello World
<%} else {%>
Hello,
<%= request.getParameter("name")%>
<%}%>
</H1>
</BODY>
</HTML>
```

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

You don't have to compile!



Java and RPG, perfect match!

Web	Java	RPG
		CBL

AS/400 AD, IBM Toronto

**Mixing
Servlets
with RPG!**

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.

First create XML required tags

Name and describe Data Structure

```
<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>
```

Pass DS as input and output

Call RPG From Java servlet

Web

Java

RPG

CBL

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

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

File: GetCust.java

Class: GetCust

Name of PCML file

Retrieve
Name

Mixing Java and RPG

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

f:\toolbox\examples>
```

Signon to AS/400

System: TORASB5D

User ID: FARR

Password: ****

Default User ID

Save password

OK Cancel

Introducing WDT/400: Studio/400 to the rescue!

What is Studio/400

Web	Java	RPG CBL
-----	------	------------

AS/400 AD, IBM Toronto

- **Part of the newly announce WDT/400**
- **Allows you to create web based applications**
- **User Interface web support**
 - JSPs
 - HTML
- **AS/400 affinity is tightly integrated in Studio/400**
- **AS/400 publishing**
- **Specific AS/400 controls**
- **Direct access to external data descriptions on the AS/400**
- **Easy way to calling AS/400 '*PGMs'**

Studio Components

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

**Page
Designer
For JSPs &
HTML**

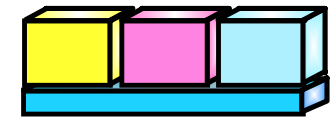
WorkBench

**AS/400
Tight
Integration**

**Many
Wizards**



How to Start Studio/400



AS/400 AD, IBM Toronto

Start

Programs

IBM WebSphere development tools for AS/400

IBM WebSphere studio for AS/400

IBM WebSphere studio v3.5

New Project

Project Name:
farr

Project Folder:
F:\StudioProjects\farr\ Browse...

Create using
Default Template

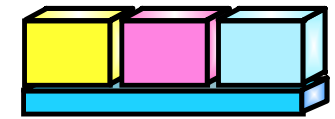
OK Cancel Help

*Specify
any
Directory*



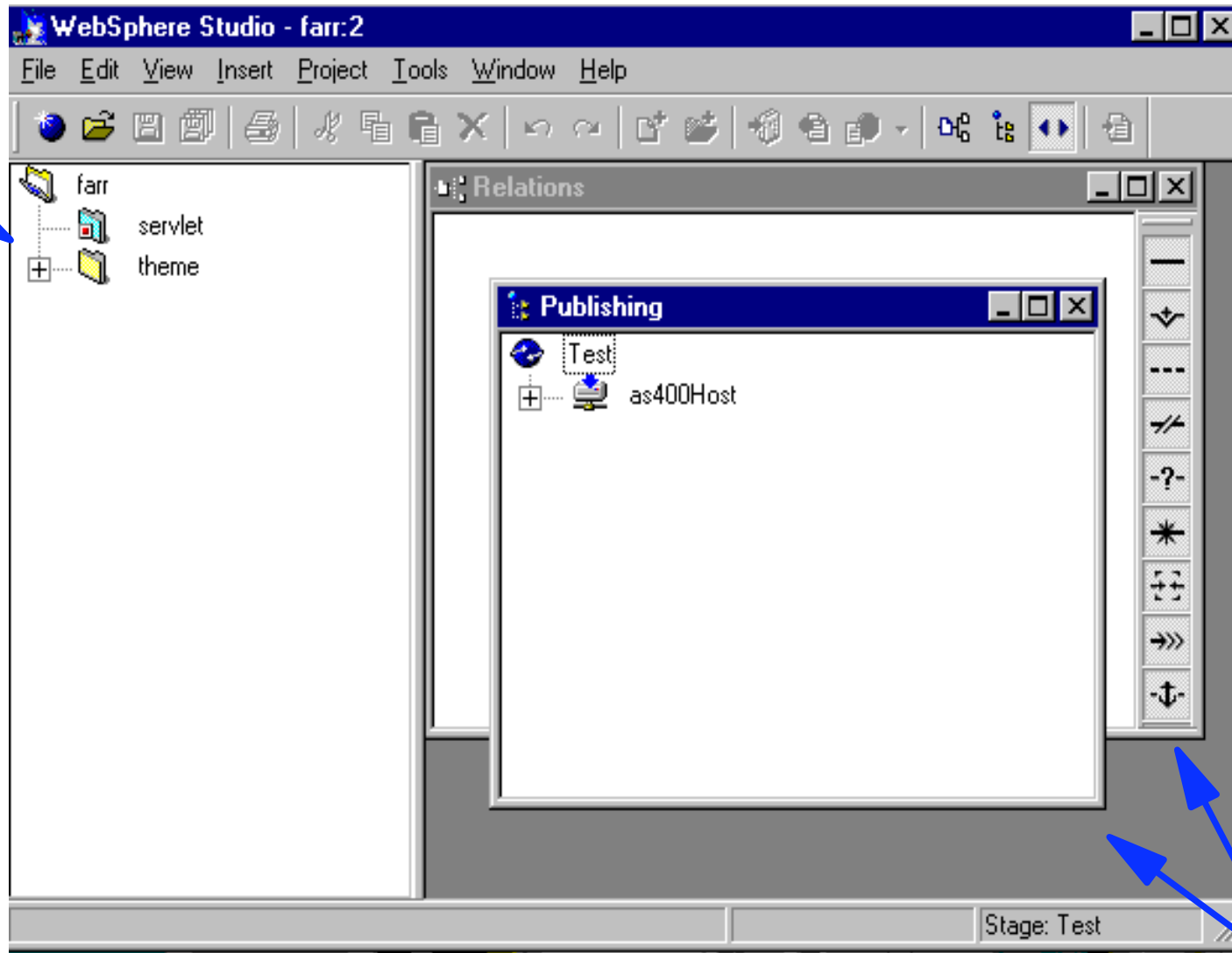


Studio/400 WorkBench



AS/400 AD, IBM Toronto

File View



Different Views



Relations View

Web

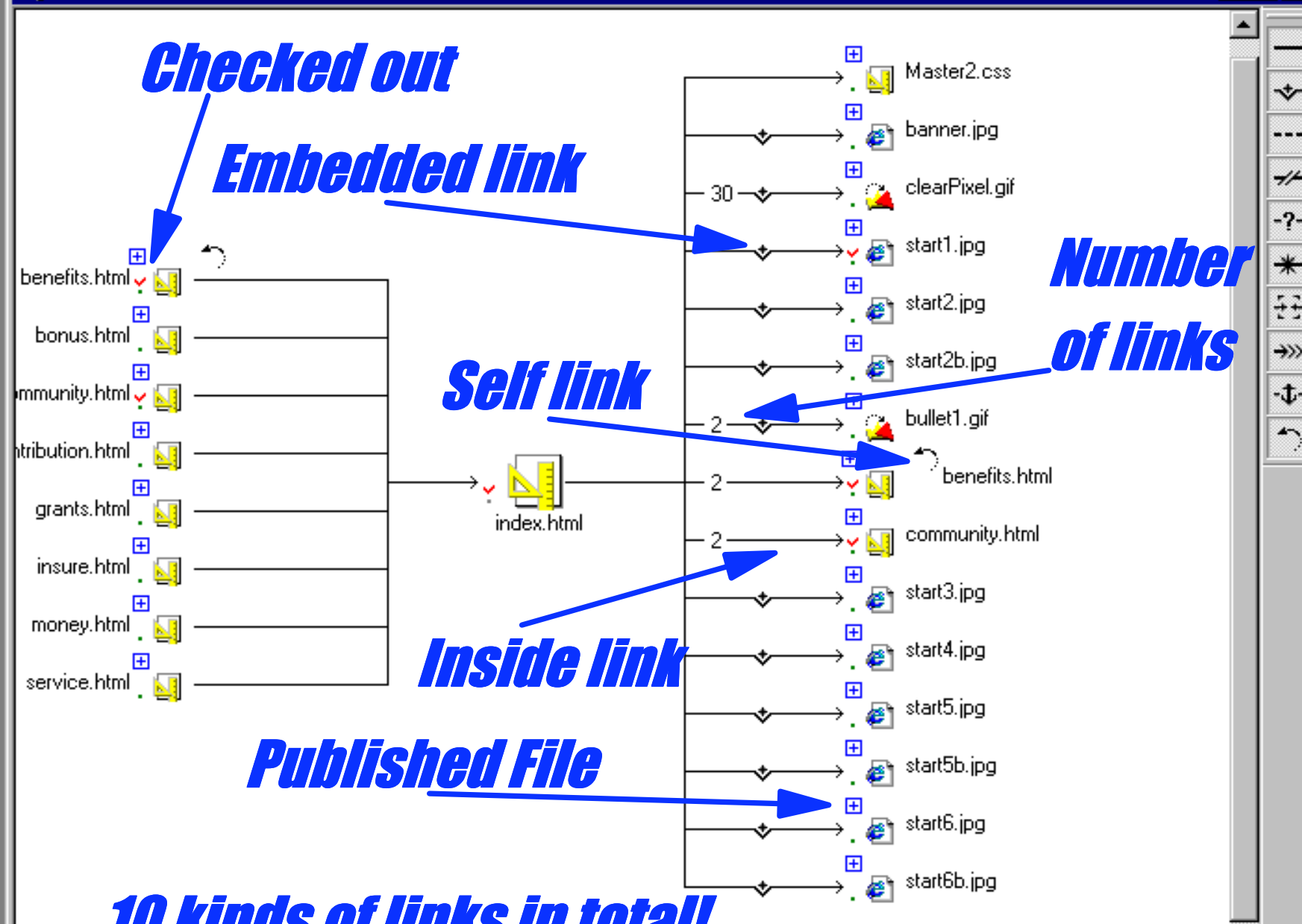
Java

RPG

CBL

AS/400 AD IBM Toronto

Relations

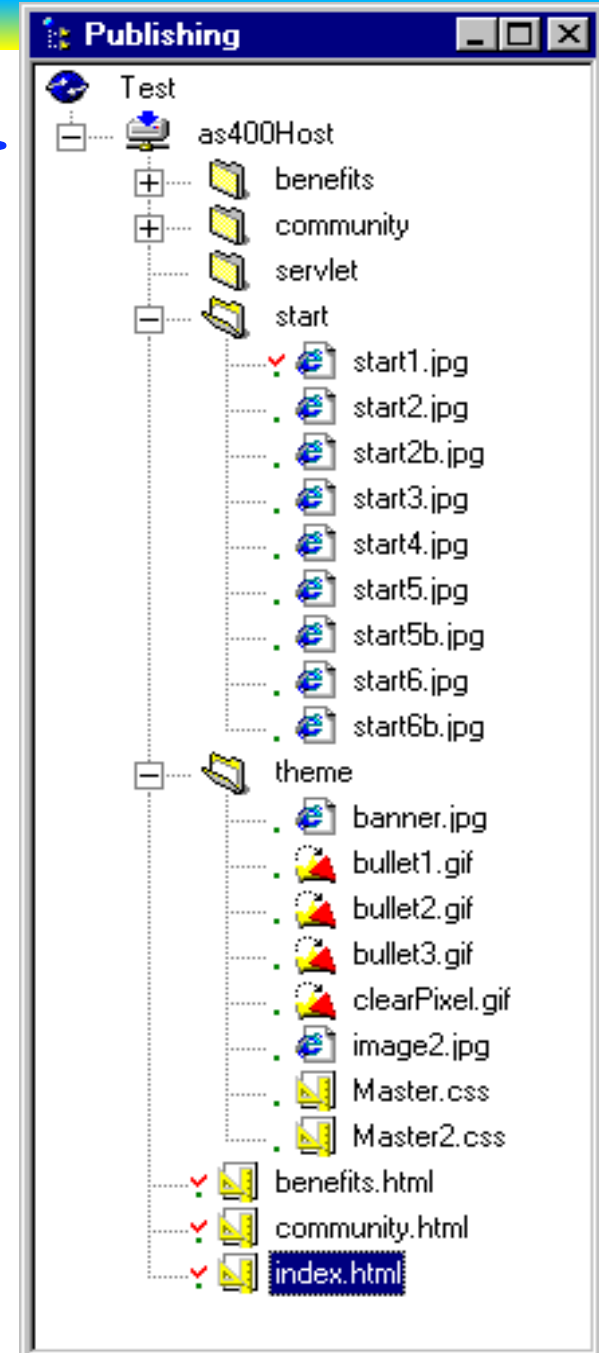


... 10 kinds of links in total!

Publishing View

Server to publish to!

- Supports staging and publishing to different servers
- Local and remote publishing
- Publishing view can differ from file view
- You can split files across servers
- To not publish a file you just remove it



Wizards!

Generates:

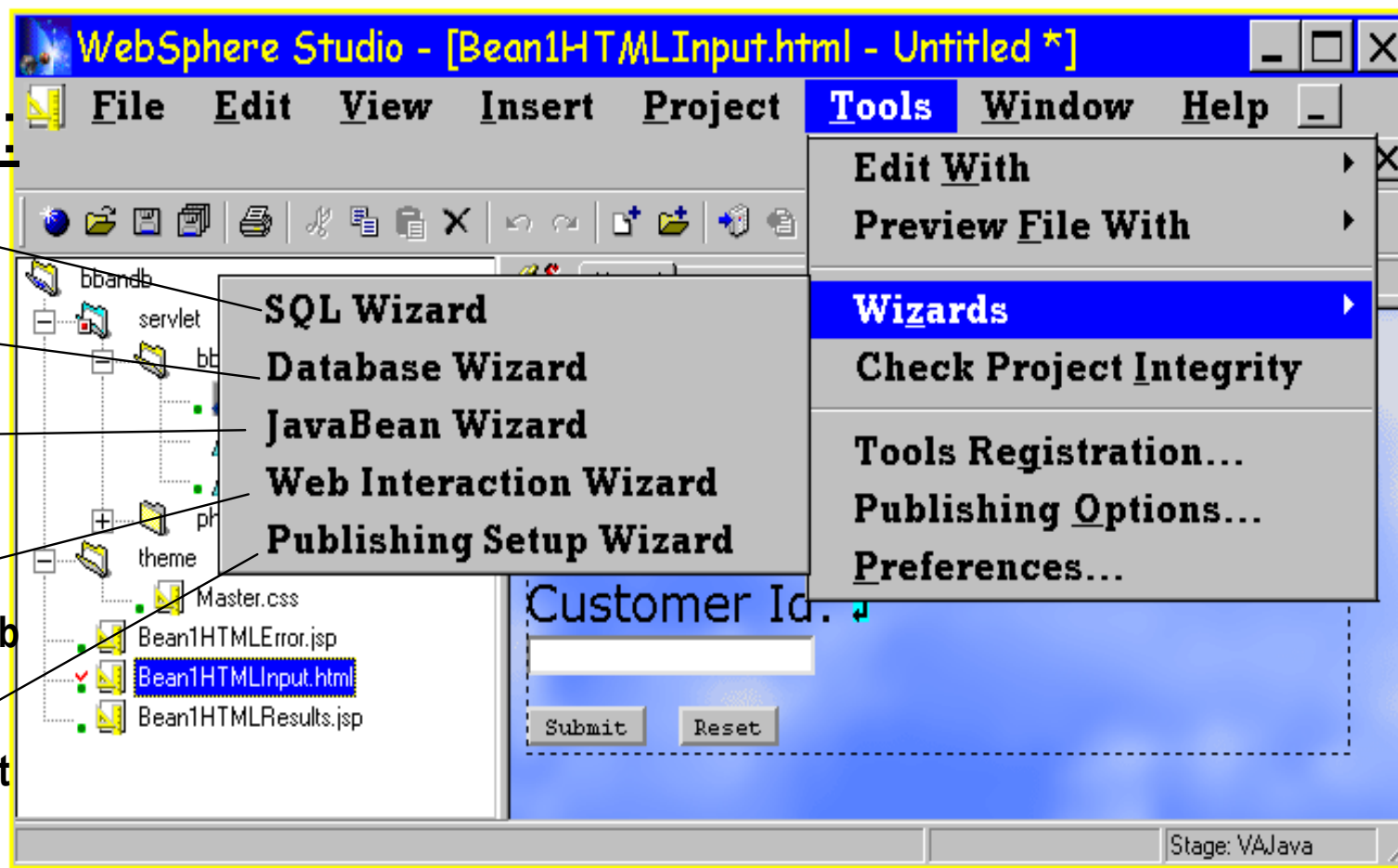
SQL Statement

DB Access Web Pages

Bean Access Web Pages

AS/400 *PGM or *SRVPGM access Web Pages

AS/400 Publish Target Properties



WebSphere Studio 3.5 Wizards

Example ...

Web

Java

RPG

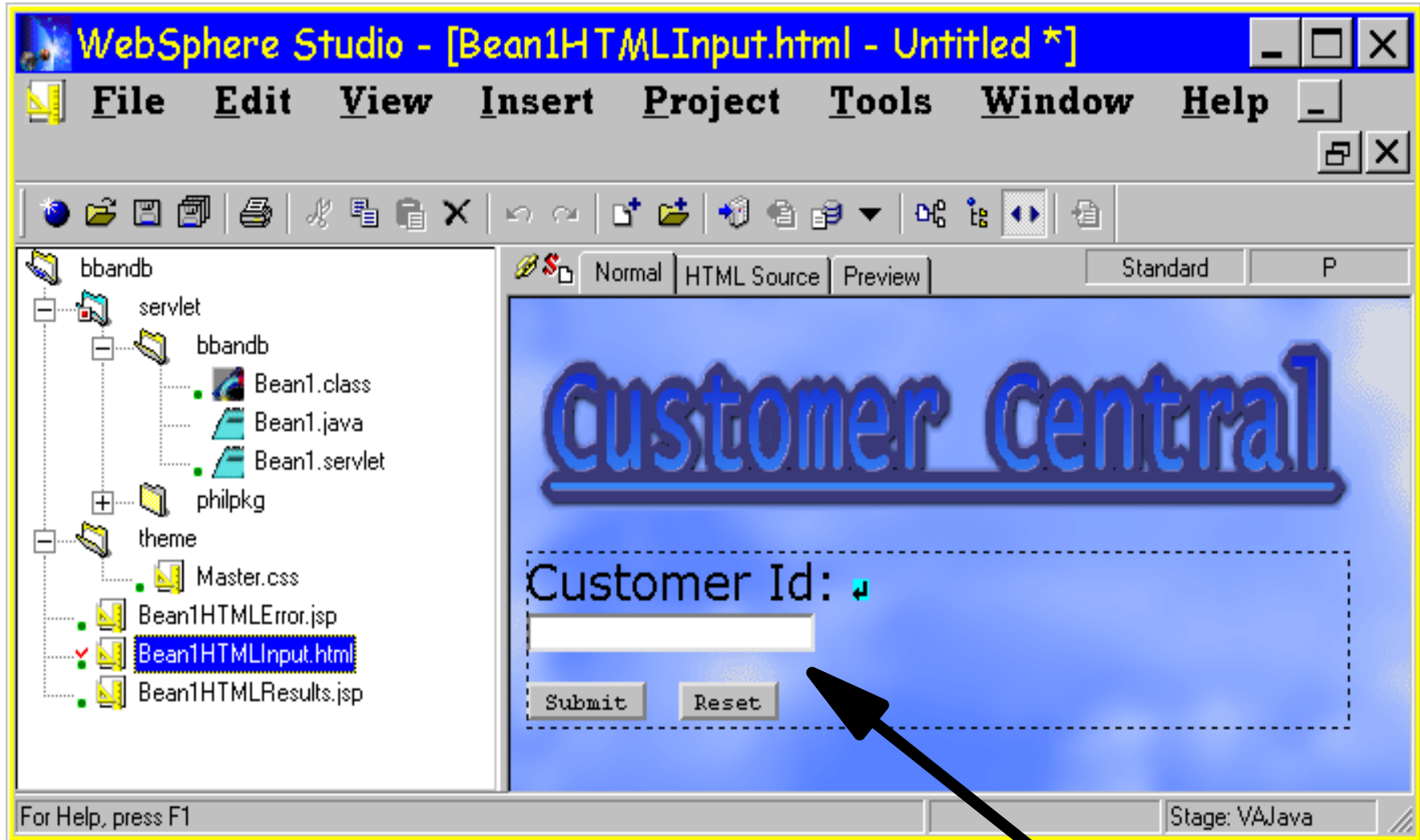
CBL

AS/400 AD, IBM Toronto

RPG

Example !

First Screen



AS/400 Entry Field

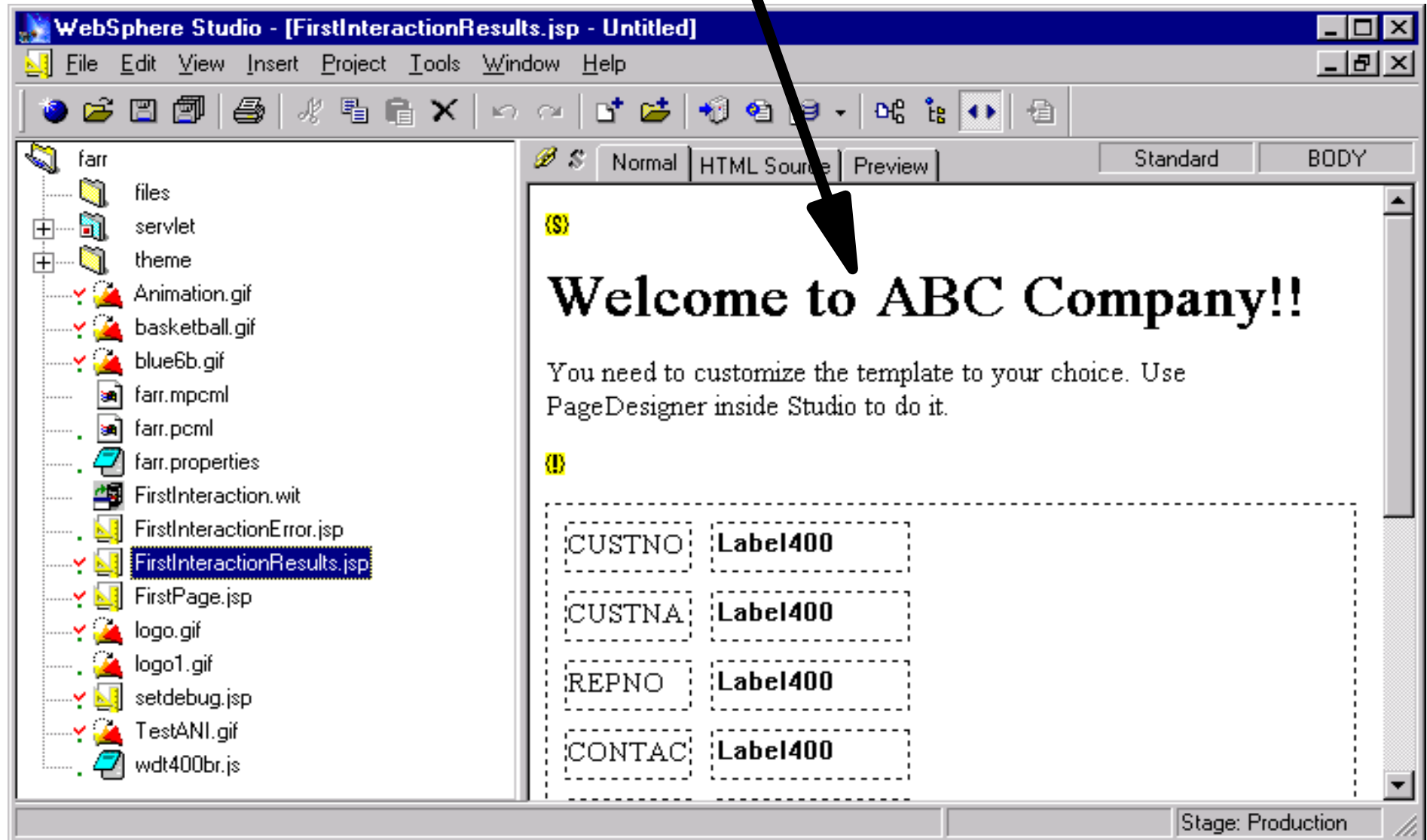
Entry Field

The screenshot displays the WebSphere Studio 3.5 AS/400 interface. On the left, the 'Insert' menu is open, with 'Design-time Control...' selected. The 'Design-time Control' palette is visible, listing various AS/400 DTCs, with 'AS/400 Entryfield DTC' highlighted. The 'AS/400 Entryfield DTC Properties' dialog is open, showing the 'General' tab. The 'Field name' is 'Entryfield400'. The 'Field type' is 'Normal'. The 'Visible width' is 20, and the 'Alignment' is 'left'. The 'Field data' section shows 'Length' as 0, 'Data type' as 'NUMERIC', and 'Formatting' as 'Edit code' with a value of 1. The 'Parameter' is 'NONE'. Red arrows point from the 'AS/400 Entryfield DTC' in the palette to the 'AS/400 Entryfield DTC Properties' dialog.

WebSphere Studio 3.5 AS/400 Parts

Customer Detail Screen

Screen generated by Studio/400



RPG IV Program

NOTE Parameters to be passed

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

Files Used

A* Logical file description

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

A* Physical file description - CUSTOMER FILE

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

**Other
AS/400
DTCS !**

AS/400 DTCs

The screenshot shows the WebSphere Studio interface with a 'Design-time Control' dialog box open. The dialog box lists several AS/400 DTCs, with 'AS/400 Checkbox DTC' selected. Arrows point from the selected DTC to a checkbox control in the design view, and from other DTCs to a combobox and a label control. The design view shows a form with a checkbox labeled 'Checkbox400', an empty text field, a combobox labeled 'Combobox', and a label labeled 'Label400'. The status bar at the bottom right indicates 'Stage: Test'.

AS/400 DTCs

The screenshot displays the WebSphere Studio interface with a 'Design-time Control' dialog box open. The dialog box lists several AS/400 DTCs, with 'AS/400 Checkbox DTC' selected. The path 'F:\WDT400\Studio35\bin\DTCs\CheckBox400.ocx' is shown below the list. The main editor area shows a 'Listbox', a 'Multi-line editor', a 'Radiobutton400', and a 'Pushbutton400'. Arrows point from the dialog box to these components in the editor. The status bar at the bottom indicates 'Stage: Test'.

WebSphere Studio - [Blank.jsp - Put Your Title Here *]

File Edit View Format Insert Table Frame Page Project Tools Window

farr

- files
- servlet
- theme
- Blank.jsp
- Master.css
- Animation.gif
- basketball.gif
- blue6b.gif
- farr.mpcml
- farr.pcml
- farr.properties
- FirstInteraction.wit
- FirstInteractionError.jsp
- FirstInteractionResults.jsp
- FirstPage.jsp
- logo.gif

Normal HTML Source

Listbox

Multi-line editor

Radiobutton400

Pushbutton400

Design-time Control

- AS/400 Checkbox DTC
- AS/400 Combobox DTC
- AS/400 Entryfield DTC
- AS/400 Label DTC
- AS/400 Listbox DTC
- AS/400 MLE DTC
- AS/400 Pushbutton DTC
- AS/400 Radiobutton DTC
- AS/400 Subfile DTC

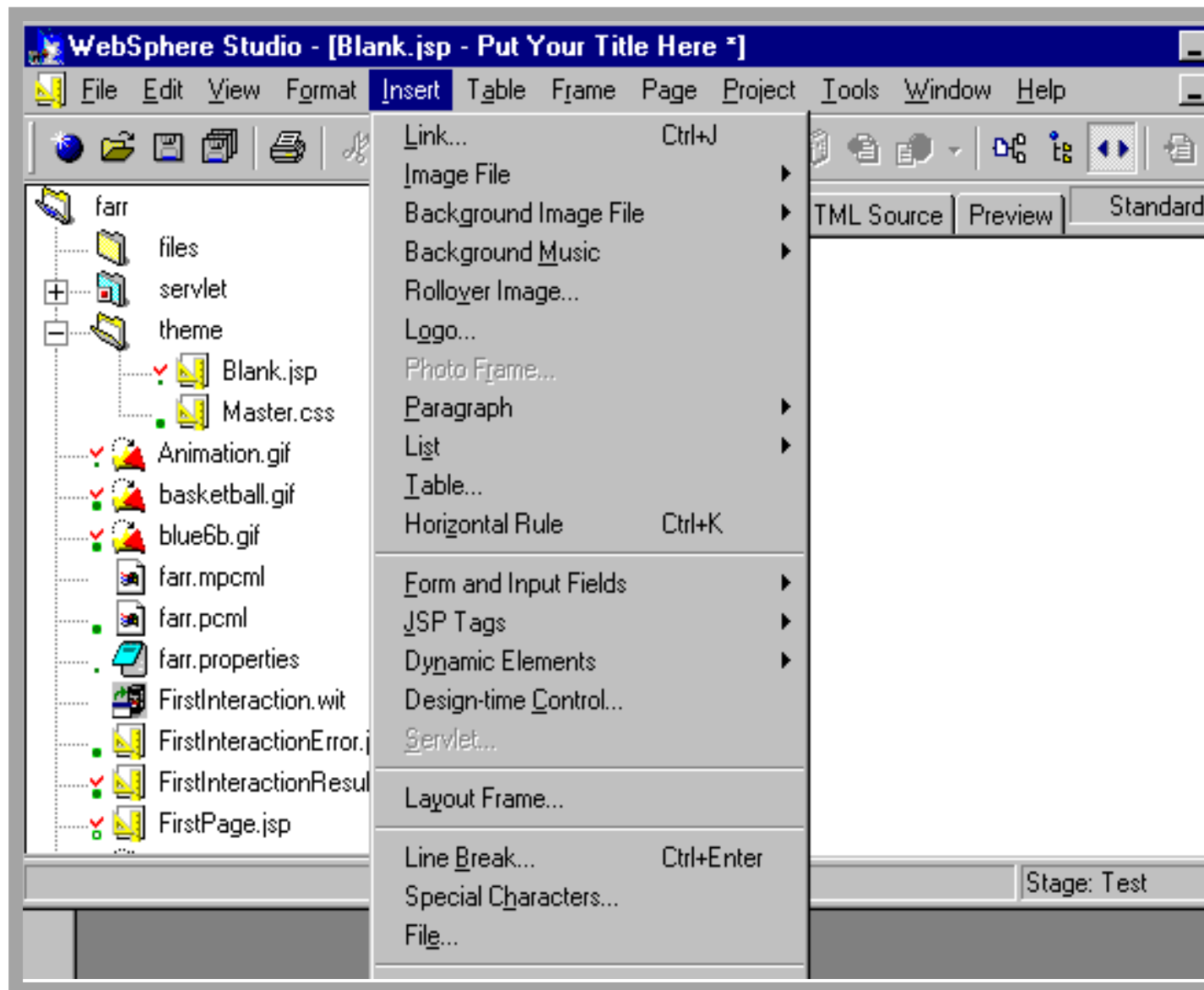
F:\WDT400\Studio35\bin\DTCs\CheckBox400.ocx

OK Cancel Help

For Help, press F1

Stage: Test

Other parts...



Interaction ...

Web

Java

RPG

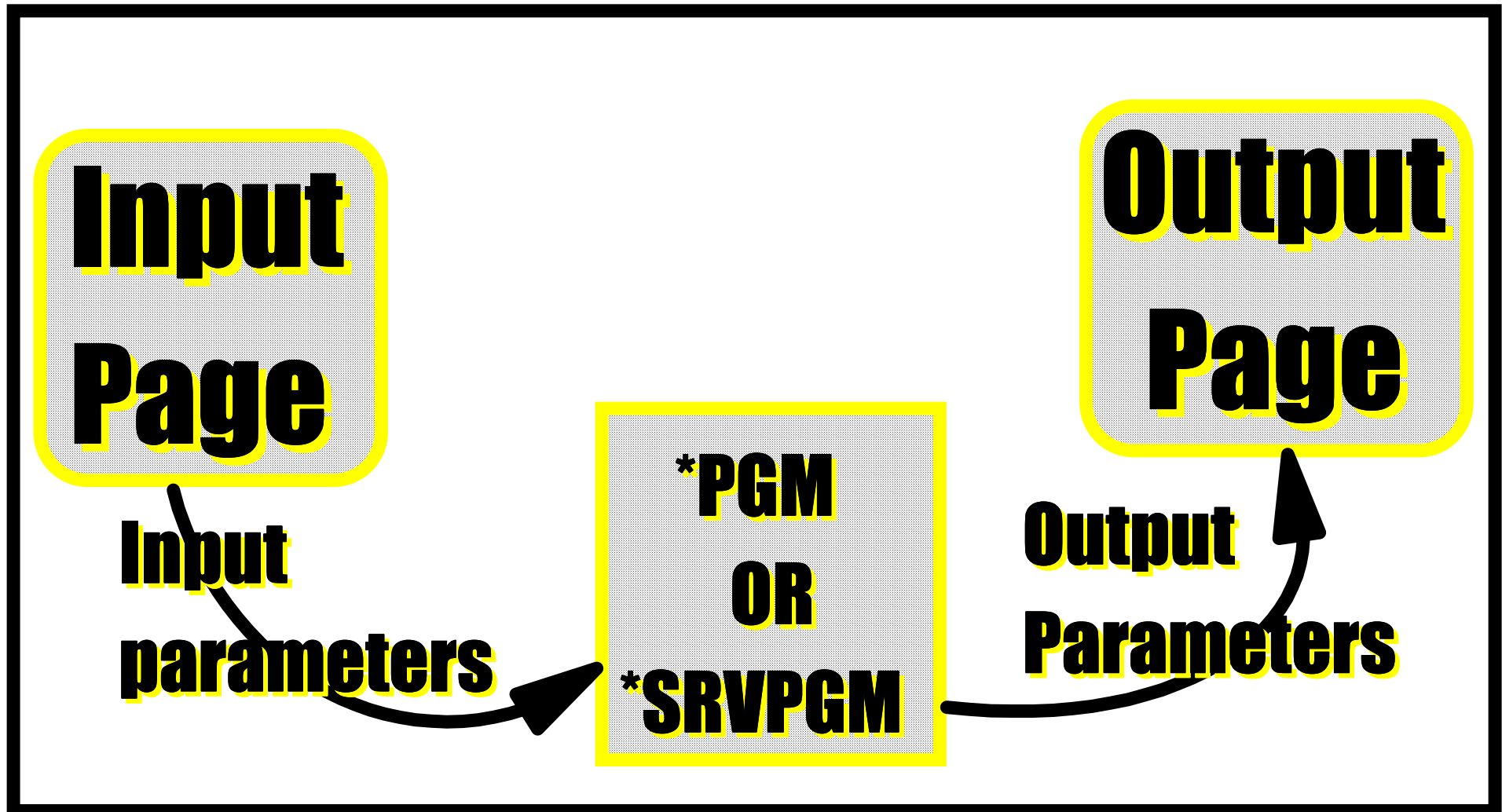
CBL

AS/400 AD, IBM Toronto

AS/400

Interaction !

An interaction



Web Interaction Wizard

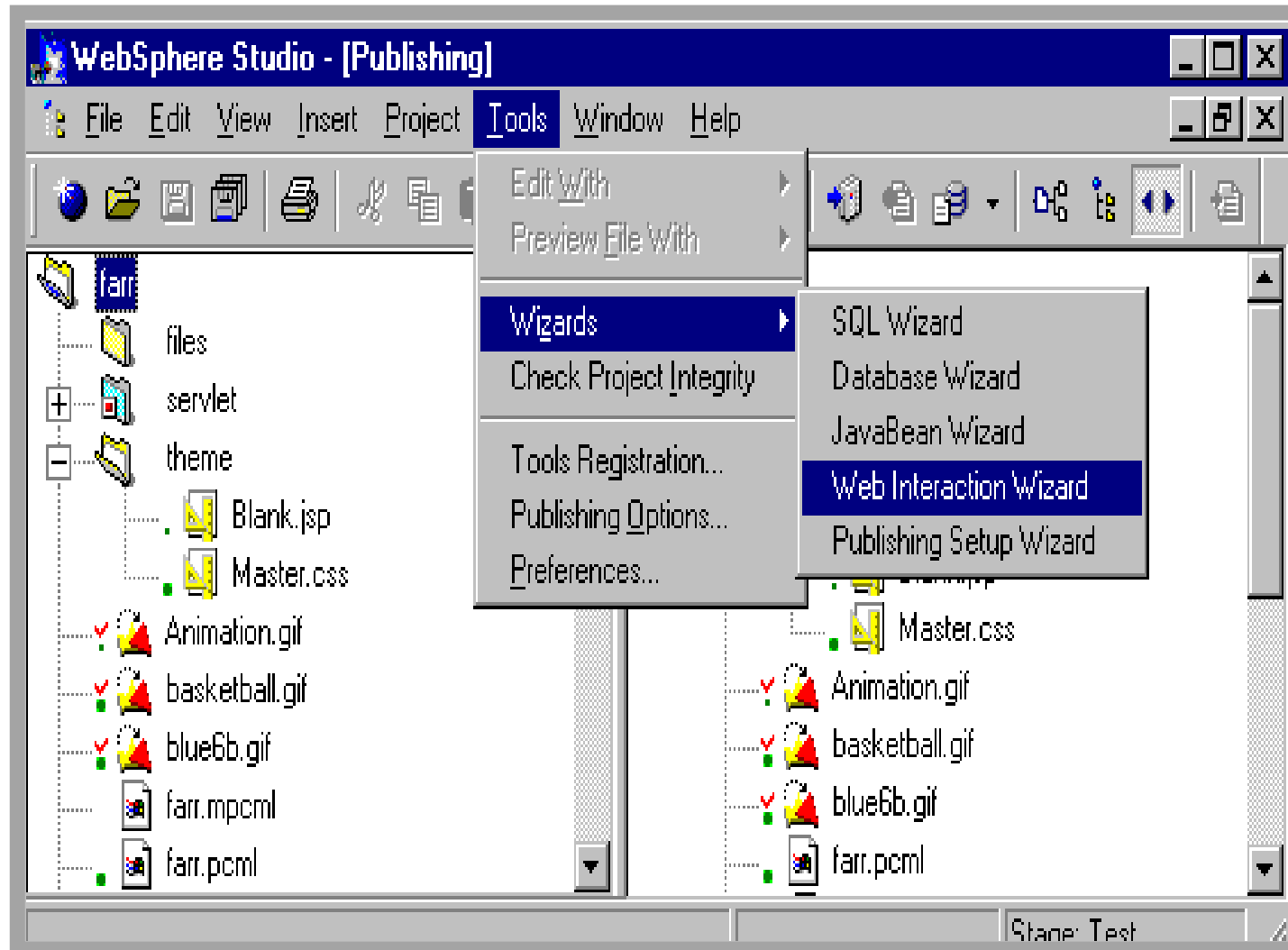
Web

Java

RPG

CBL

AS/400 AD, IBM Toronto



Interaction Wizzard

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

Web Interaction Wizard

Introduction

Welcome to the Web Interaction wizard. This wizard leads you through the following steps:

1. Specify an interaction name for the communications that occur between your Web pages and your ILE program.
2. Choose your input, results, and error Web pages.
3. Identify the ILE program and the input and output parameters called during the Web interaction.
4. Map the fields on your input page to your ILE program parameters.
5. Map the fields on your results page to your ILE program parameters.
6. Select an error page and define how program and input errors should be handled.

After completing the steps in the wizard, click Finish. A list will appear displaying all of the items to be generated, compiled, and checked into WebSphere Studio.

<Back

Next>

Finish

Cancel

Help

Web Interaction

Web Interaction Wizard [X]

Specify a name for your Web interaction

This wizard helps you define the communication that occurs between Web pages and the ILE program. This is called an interaction. Each page in this wizard lets you describe a different aspect of the Web interaction. Use this page to type a name that describes the interaction. Click Next to continue.

Interaction name

Create Edit existing

Invalidate session after the interaction occurs

No host program call

Servlet URL:

<Back Next> Finish Cancel Help

Specify Interaction name

Cleans up web session

Don't want to call a program

Define Pages

Define Input, Output, and error pages!

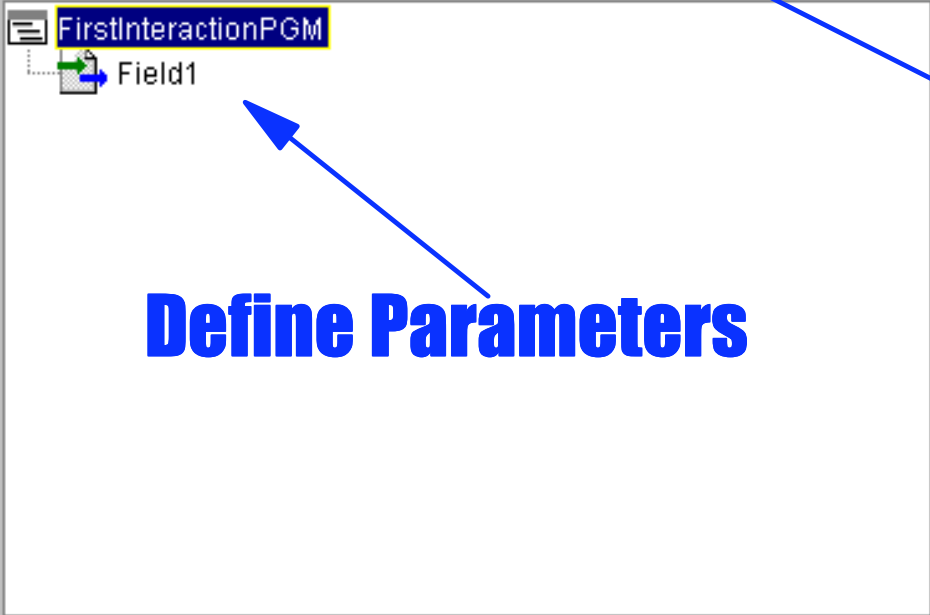
Define program call interface

Web Interaction Wizard

Specify the input and output parameters for your ILE program

Use this page to define the input and output parameters for your ILE program. The popup menu lets you add new fields or import field definitions from data descriptions of a database file. To view the program source in the CODE Editor, click on the Show Source push button. * denotes a mandatory field.

Program call definition



Program element attributes

Attribute Name	value
*Program alias	FirstInteractionPGM
*Program object	
*Program library	
*Program type	*PGM
entrypoint	
Program source	

Show expert attributes

Buttons: Show Repository, Show Source

Program to call (points to FirstInteractionPGM)

Define Parameters (points to Field1)

Can be service program entry point! (points to entrypoint attribute)

Navigation: <Back, Next>, Finish, Cancel, Help

Interaction Wizard

Web Interaction Wizard

Specify the input and output parameters for your ILE program

Use this page to define the input and output parameters for your ILE program. The popup menu lets you add new fields or import field definitions from data descriptions of a database file. To view the program source in the CODE Editor, click on the Show Source push button. * denotes a mandatory field.

Program call definition

- FirstInteractionPGM
 - Field1

Field attributes

Program element attributes

Attribute Name	value
*Field name	Field1
*Data type	char
length	7
precision	
count	
usage	input
init	
struct	
outputsize	

Show expert attributes

Show Source

Show Respository

<Back Next> Finish Cancel Help

Data Structure

Web Interaction Wizard

Specify the input and output parameters for your ILE program

Use this page to define the input and output parameters for your ILE program. The popup menu lets you add new fields or import field definitions from data descriptions of a database file. To view the program source in the CODE Editor, click on the Show Source push button. * denotes a mandatory field.

Program call definition

- FirstInteraction
- Field

Program element attributes

Attribute Name	value
Program Name	FirstInteractionPGM
Function	getdata
Object	weiss
Library	
*Program type	*PGM
entrypoint	
Program source	

Context Menu:

- Add
- Delete
- Add Fields from DB Reference
- Sync Fields from DB Reference
- Add Struct from Repository
- Save to Repository
- Move Up
- Move Down
- Show All Programs
- Set Current Program

Structure Menu:

- Program
- Field
- Structure

Create a new structure dialog:

Select a struct from the list or type a new structure name here.

Buttons: Show Repository, Show Source, <Back, Next>, Cancel, Set

External Field Description

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

The screenshot shows a main dialog box titled "Add AS/400 DB Reference Field" with a close button (X) in the top right corner. At the top, there is a "Path" text box containing "<TORAS14M>" and a "Connect" button to its right. Below the path is a tree view under "Host" showing a folder icon and the text "TORAS14M".

In the center, a smaller dialog box titled "Signon to AS/400" is open, also with a close button (X) in its top right corner. It contains the following fields and options:

- "System:" text box with "TORAS14M" entered.
- "User ID:" text box with a vertical cursor.
- "Password:" text box.
- Two checked checkboxes: "Default User ID" and "Save password".
- "OK" and "Cancel" buttons at the bottom.

Below the tree view, there is a large empty rectangular area. At the bottom of the main dialog box, there are three buttons: "OK", "Cancel", and "Help".

On the right side of the main dialog box, there is a list of field attributes, each with an adjacent text box:

- Field
- Alias
- Length
- Type
- Usage
- Description
- Column headings
- Edit code or word
- Validity keywords

External Descriptions

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

Add AS400 DB Reference Field

Path: <TORAS14M>GUIDES4/CUSTOML3/CUSTOM01 Connect

Host

- TORAS14M
 - GUIDES4
 - ALLBACK
 - ALLSALES
 - BYREGION
 - CUSTOMER
 - CUSTOML1
 - CUSTOML3
 - CUSTOM01**
 - CUSTOMS1
 - CUSTOMS2
 - CUSTOMS3
 - CUSTOMS4
 - PRODDTL
 - PRODINV
 - PRODLIST
 - PRODSHOW
 - PRODUCT
 - PRODUCTJF
 - SALES

Fields:

- CUSTNO
- CUSTNA
- REPNO**
- CONTAC**
- CPHONE**
- CFAK
- CADDR

Field Details:

Field	
Alias	
Length	
Type	
Usage	
Description	
Column headings	
Edit code or word	
Validity keywords	

OK Cancel Help

Connecting Fields

Web Interaction Wizard

Map and link the input parameters to the input fields

This page shows the relationship between the parameters in your program and the fields on your Web page. Data entered into the input field on your Web page will be passed to the corresponding input parameters in your program. The table shows which parameters are linked to which fields.

Input parameters		Link		Input fields	
Field1		<input type="button" value="Link"/>	<input type="button" value="Unlink"/>	files/FirstPage.jsp	
DataS.CUSTNO		Parameters	Fields	FORM:	
DataS.CUSTNA		Field1	Custno	Custno <input type="text"/>	
DataS.REPNO				Inquiry <input type="checkbox"/>	
DataS.CONTAC					
DataS.CPHONE					
DataS.CFAX					
DataS.CADDR					
DataS.CCITY					
DataS.CCOUNT					
DataS.CZIP					
DataS.CZIPLO					
Return					
Session	Value			Session	Value
Restore from	<input type="checkbox"/>			Save to	<input type="checkbox"/>
Name	Field1			Name	Custno

Field1 from the input page is linked to Custno.

<Back Next> Finish Cancel Help

Output Parameters

Web

Java

RPG

CBL

Web Interaction Wizard

Select the output parameters to be included

The output parameters listed below are all selected for this interaction. If there are parameters that you do not want to include, then clear the Include check box. To clear all of the parameters, click on the Clear All push button and then you can individually select which parameters you want to include. You can modify the parameter's attributes as well as reorder the fields.

Output parameters to include

- ✓ DataS.CUSTNO
- ✓ DataS.CUSTNA
- ✓ DataS.REPNO
- ✓ DataS.CONTAC
- ✓ DataS.CPHONE
- ✓ DataS.CFAX
- ✓ DataS.CADDR
- ✓ DataS.CCITY
- ✓ DataS.CCOUNT
- ✓ DataS.CZIP
- ✓ DataS.CZIPLO
- Return**

Select All

Clear All

Include	<input type="checkbox"/>
Name	Return
Type	char
Caption	Return
Save to session	<input type="checkbox"/>
Name in session	Return

Generate output page with these fields!

<Back

Next>

Finish

Cancel

Help

Error Page

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

Web Interaction Wizard

Specify the layout and messages for the error pages

There are standard error pages that you can use for this Web interaction or you can customize your own error page and messages. You can customize the text for a standard error page or specify a unique message ID to a specific input field.

Standard error page

Use an existing page

yourCustomErrorPage.jsp

Create a new page

Include the following text in the error page

```
An error has occurred.  
Please contact the support center.
```

Define program and user error handling

Select an output parameter for retrieving message ID and substitution text

Return

Specify the parameter value if no error occurs

0

Specify the AS/400 message file name

Library

File name

FARR

MymsgF

Enter the message ID for the input field

Input Field	Message ID
Custno	

<Back

Next>

Finish

Cancel

Help

Web Interaction Wizard

Status →

Web Interaction Wizard

- ✓ Creating servlet and JSP...
- Compiling servlet and data bean...
- Checking generated servlet and JSP into Studio...
- Saving settings into metadata file...

Servlet URL:

Create Package

The files listed below will automatically be created in the package directory structure, then you must specify

Description	
Results page	FirstInteractionResults.jsp
Error page	FirstInteractionError.jsp
Servlet source	FirstInteractionServlet.java
Data bean	ProgramRecord_FirstInteractionPGM.java

**Optional
Package
Name!**

Package name

What's Next?

Web	Java	RPG CBL
-----	------	------------

AS/400 AD, IBM Toronto

● What did you do so far?

- ▶ You used the designer to create the input page
- ▶ The output page was generated for you
- ▶ Created you Files on the AS/400
- ▶ Created your RPG Program
- ▶ Created an 'AS/400 Web Interaction' Interaction to link your program parameters to the input page and output page

● Next?

- ▶ Publish all your files to the AS/400 server



Publishing ...

Web

Java

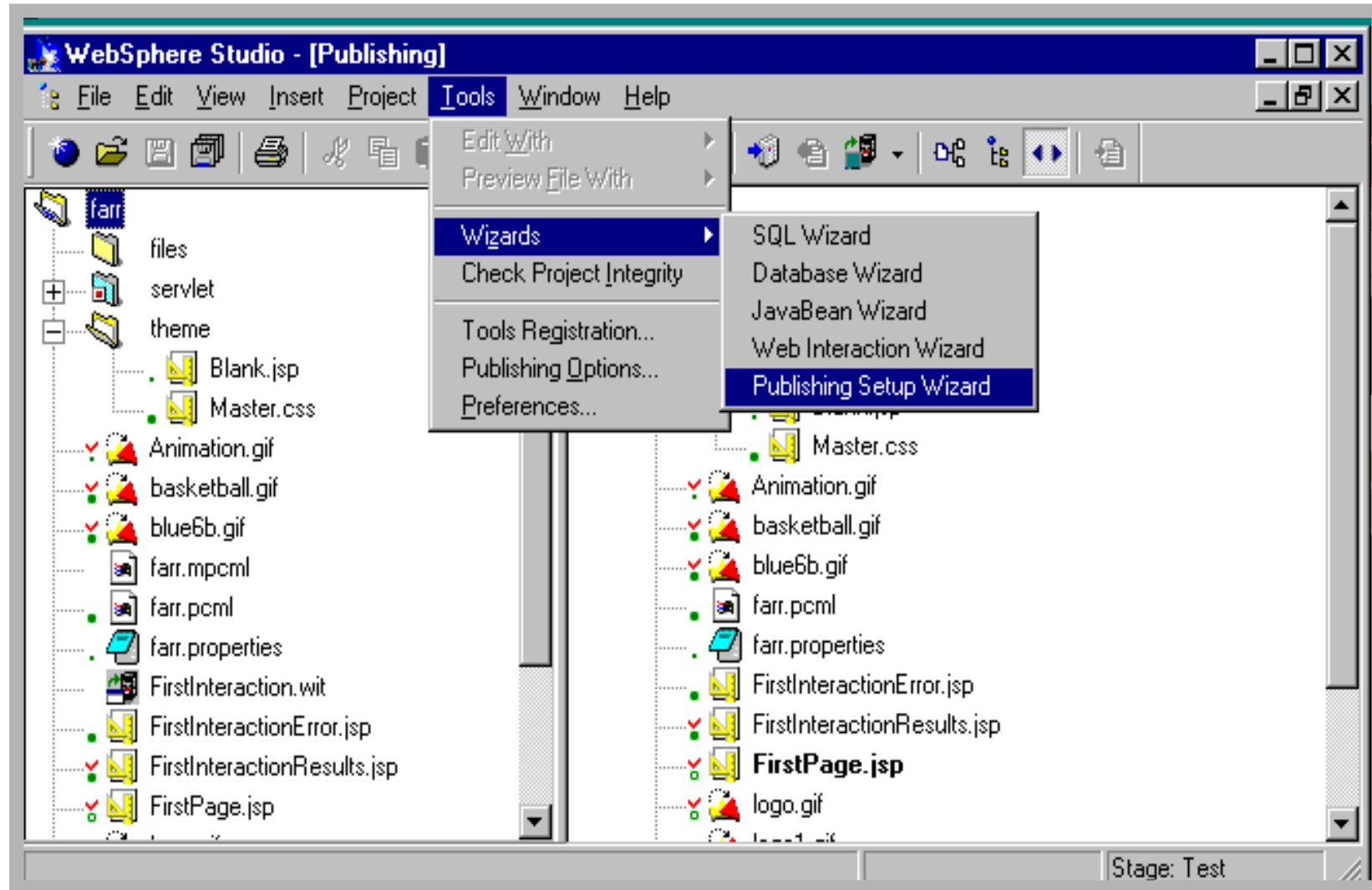
RPG

CBL

AS/400 AD, IBM Toronto

**Publishing
your app !**

Publishing



Publishing

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

Publishing Setup Wizard

Specify the location and publishing settings for the Web application

This wizard helps you define on which AS/400 host you want to publish the Web application. Each page in this wizard lets you describe a different aspect of the connection between your AS/400 host and your local machine. Use this page to set up the name of your AS/400 host, the Web address directory, and the servlet root directory and path. Click Next to continue.

Host name	<input type="text" value="TORAS14M.TOROLAB.IBM.COM"/>
XML node name	<input type="text" value="TORAS14M"/>
WAS Bootstrap port	<input type="text" value="11050"/>
Web application name	<input type="text" value="farr"/>

**WAS Port,
use '900'
(default)**



Specify the portion of the URL that identifies your Web application's Web path on the WebSphere Application Server

Web application's Web path	<input type="text" value="/projects"/>
----------------------------	--

Specify the root directory for your HTML, JSP, and servlet files on the AS/400 host

Root directory	<input type="text" value="/FARR/WEB"/>
Class path	<input type="text" value="/FARR/SERVLETS"/>

Validate

IFS
Directories



<Back

Next>

Finish

Cancel

Help

User id and library list

Web Java RPG
CBL

AS/400 AD, IBM Toronto

Publishing Setup Wizard [X]

Specify the AS/400 environment where this Web application will run

This page allows you to specify the host name, user ID, passwords, and libraries that your local machine will copy the Web application

AS/400 environment settings

Host	TORAS14M.TOROLAB.IBM.COM												
User ID	FARR												
Password	*****												
Runtime library list	<table><thead><tr><th>Libraries</th><td>Up</td></tr></thead><tbody><tr><td>FARR</td><td>Down</td></tr><tr><td>WEISS</td><td>Refresh</td></tr><tr><td>COULTHARD</td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></tbody></table>	Libraries	Up	FARR	Down	WEISS	Refresh	COULTHARD					
Libraries	Up												
FARR	Down												
WEISS	Refresh												
COULTHARD													

<Back Next> Finish Cancel Help

Review publishing settings

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

Publishing Setup Wizard [X]

Review the publishing settings for the Web application

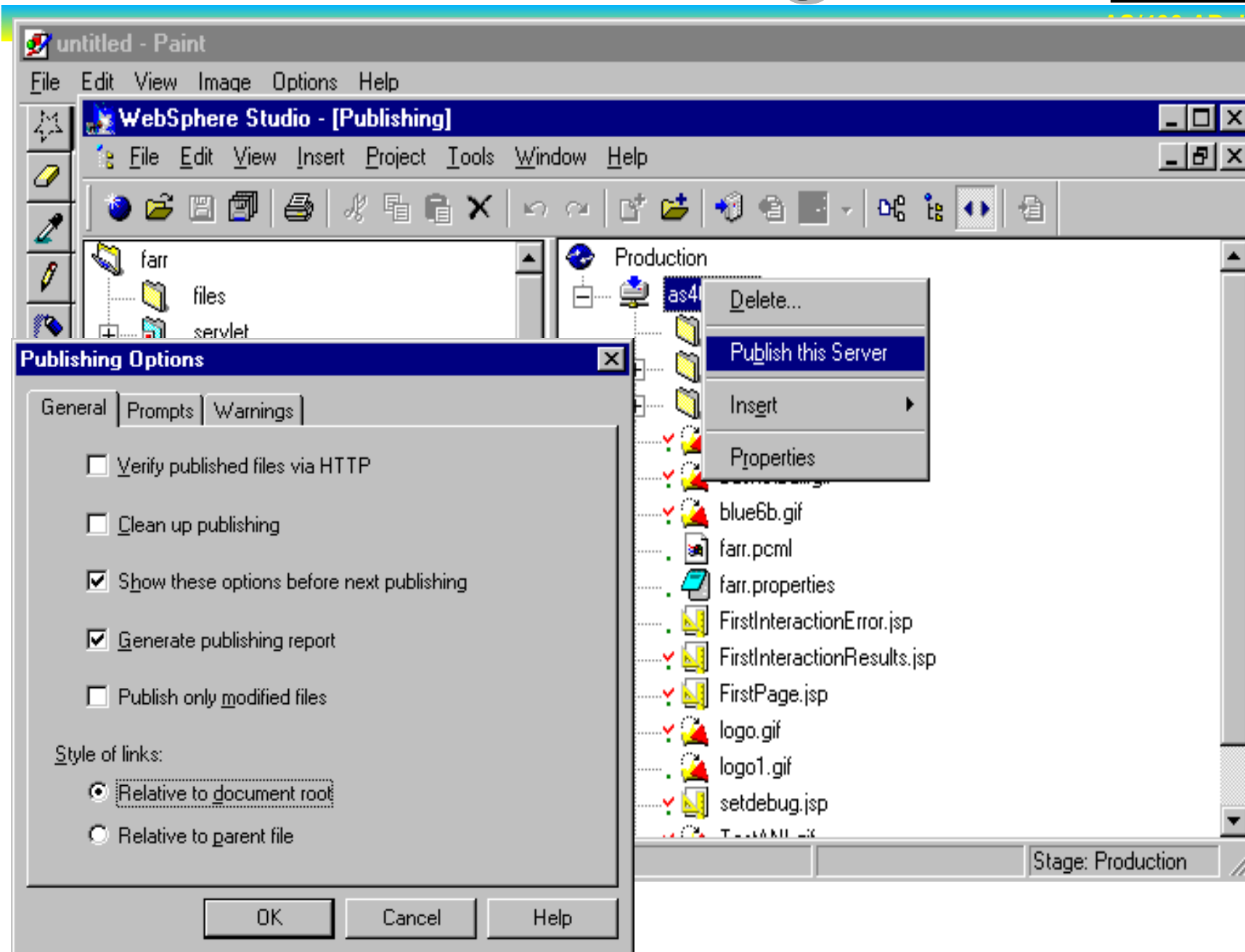
When you click Finish, the wizard will store the settings that you have defined. To change any of the information, go back to the previous page in the wizard. To publish the Web application, use the Publishing view in WebSphere Studio to copy the files and servlets to the AS/400 host.

Host name to publish to:	TORAS14M.TOROLAB.IBM.COM
Web application name:	farr
Web path for Web application	/projects
Document root for Web application	/FARR/WEB
Class path for Web application	/FARR/SERVLETS
Host name for application to run on	TORAS14M.TOROLAB.IBM.COM
User ID for application to run with	FARR
Runtime library list for application to run with	FARR WEISS COULTHARD

<Back Next> Finish Cancel Help

Publishing

IBM Toronto



Publishing

**HTTP port
Default port =
80**

**WAS
Configuration**

Set Publishing Options

AS400 Host: TORAS14M.TOROLAB.IBM.COM
WAS node: TORAS14M

View a web page after publishing

Page Name: Firstpage
Http Port: 11042

Display Detailed Errors
 Check Configuration

WAS 3.0
 WAS 3.5

Instance Name:

OK Cancel

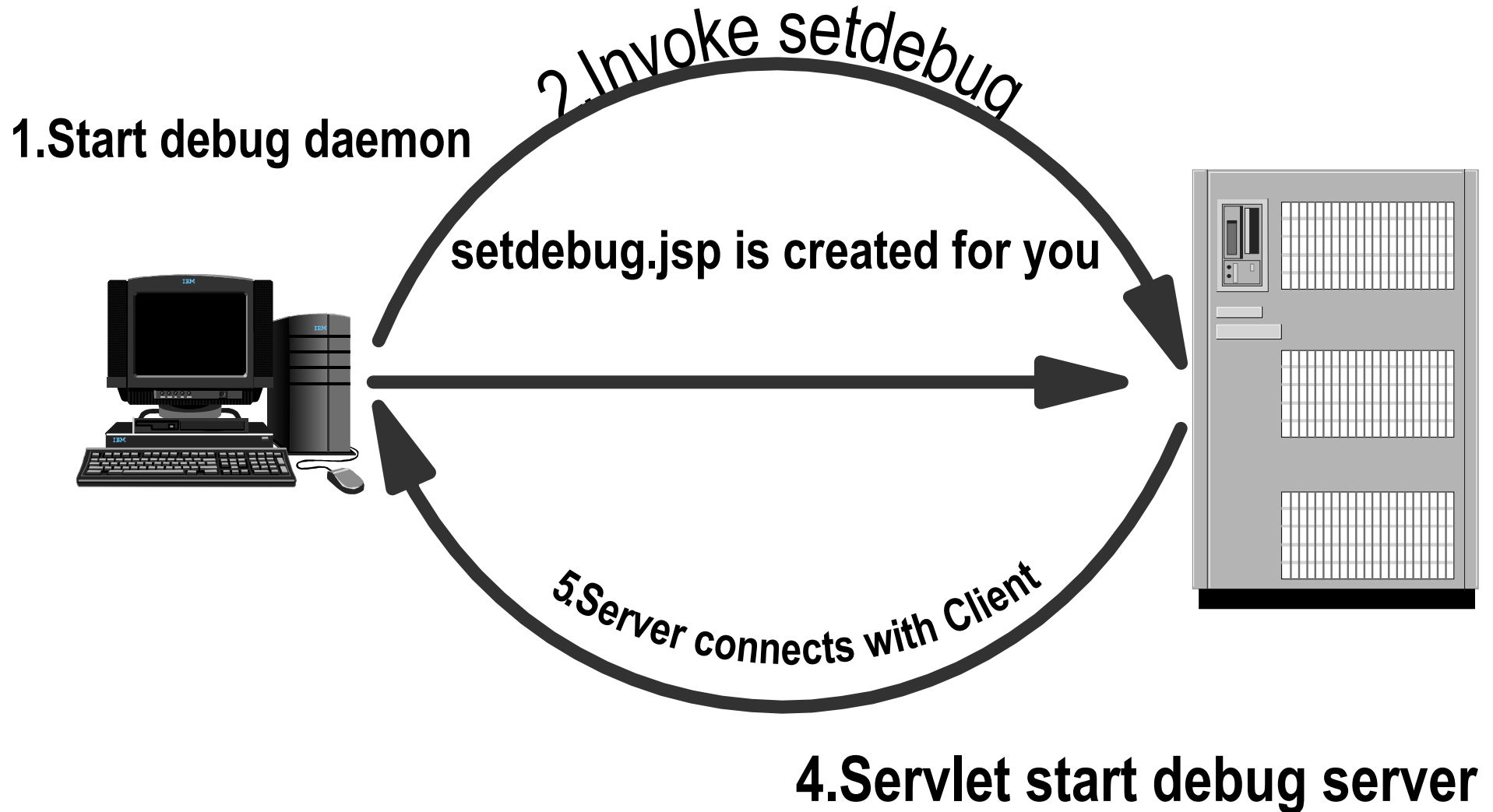
**First Page to
be displayed**

Remote Debug!

Debug, How does it work?

Web	Java	RPG
		CBL

AS/400 AD, IBM Toronto



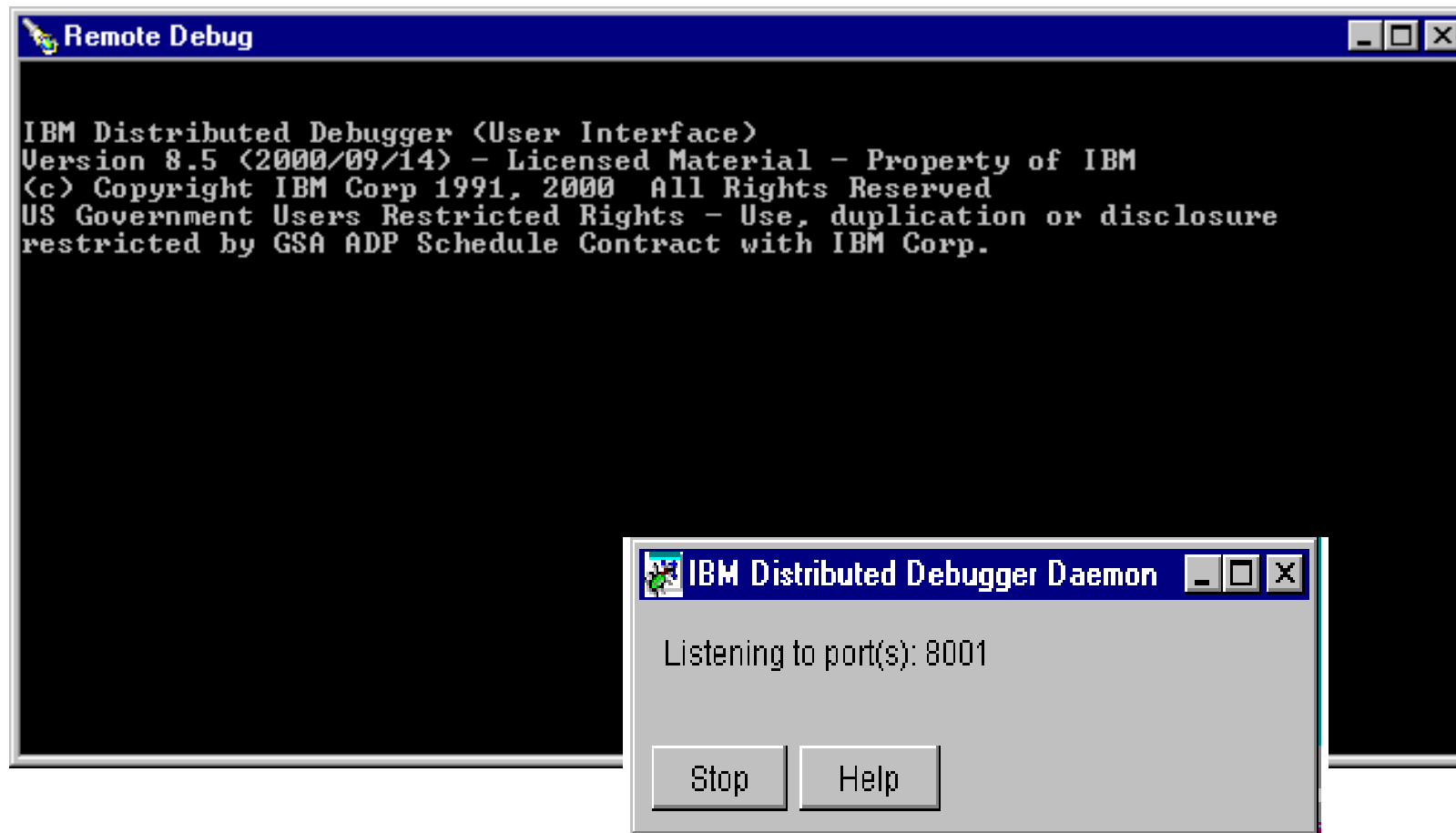
DEBUG

Web	Java	RPG CBL
-----	------	------------

AS/400 AD, IBM Toronto

Execute the command:

F:\IBMDebug\bin\idebug.exe -qdaemon -quiport=8001



setdebug JSP file

The screenshot displays the WebSphere Studio interface for a project named 'farr'. The main workspace shows a project tree with the following structure:

- farr
 - files
 - FirstPage.jsp
 - portlet
 - com
 - FirstInteractionServlet.class
 - FirstInteractionServlet.java
 - ProgramRecord_FirstInteractionPGM\$CustDS_Struct.class
 - ProgramRecord_FirstInteractionPGM.class
 - ProgramRecord_FirstInteractionPGM.java
 - theme
 - basketball.gif
 - blue6b.gif
 - farr.mpcml
 - farr.pcml
 - farr.properties
 - FirstInteractionError.jsp
 - FirstInteractionResults.jsp
 - logo.gif
 - setdebug.jsp
 - wdt400br.js
 - FirstInteraction.wit

A hand icon is pointing to the 'setdebug.jsp' file in the theme folder. To the right, a 'Publishing' window is open, showing the same project structure being deployed to a 'Test' environment. The 'setdebug.jsp' file is also visible in the published structure. The 'Stage: Test' indicator is visible at the bottom right of the publishing window.

Debug Trigger

Web Java RPG
CBL

AS/400 AD, IBM Toronto

Setting up debugger information - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Location: What's Related

Instant Message Internet Lookup New&Cool RealPlayer

Please enter information below:

If you use the default values in the following fields, the default behavior will be taken:
The debugger client will run on the same workstation this browser runs on

IP Address of machine where debugger client will run:

Port of debugger deamon:

Currently debug is OFF

Do you want to:

enable debugging

disable debugging

Document: Done

Page (68)

Enable debug trigger

Distributed Debugger

Web

Java

RPG

CBL

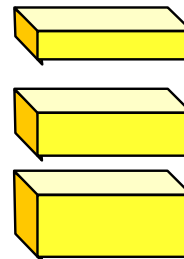
The screenshot shows the IBM Distributed Debugger interface. The main window displays the source code for a thread named *PGM FARR/GETDATA. The code is as follows:

```
1 FCUSTOML3 IF E K Disk
2 DCustnoi s like(CUSTNO)
3 D*
4 D CSTRUC E DS extrname(custom13:custom01
5 d field2 10
6 D return s 20
7 D*-----
8 c *entry plist
9 c parm custnoi
10 c parm cstruc
11 c parm return
12 c eval return=*blank
13 c custnoi chain custom13
14 c if *in50
15 c eval return='CUS0001 ' + CUSTNOI
16 c else
17 c eval return='0'
18 c endif
19 c return
```

The interface includes a menu bar (File, View, Selected, Debug, Source, Stacks, Monitors, Window, Help), a toolbar with various icons, and a status bar at the bottom that reads "Debugger ready/Daemon listening to port 8001...". On the left side, there are panels for "Stacks" (showing Thread 1 with sub-items GETDATA, _QRNP_PEP_GETDA, CallProgram, main, _C_pep) and "Monitors" (showing an expression CUSTNOI = 0010100).

Hardware & Software

	Memory	Disk
Hardware Requirement	96 MegaByte - Minimum 128 MegaByte is recommended	300 MegaByte Minimum 500 MegaByte is recommended



	Development time	Run time
Software Requirement	Windows 95/98/NT/2000; Internet Explorer 4.0, or later; OS/400 V4R4, or later; WebSphere Application server 3.5, or later;	OS/400 V4R4, or later; Netscape 4.7(provided), or later; Internet Explorer 5.0 for page detailer; WebSphere Application server V3.5, or later

Conclusion

Web

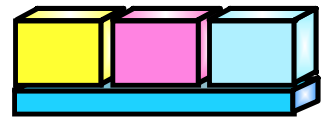
Java

RPG

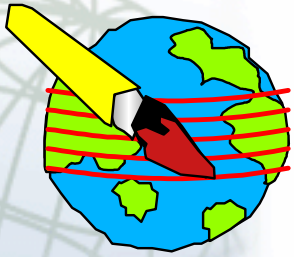
CBL

AS/400 AD, IBM Toronto

**Into the
Future with
WDT/400**

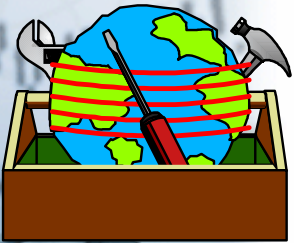


► Strategy goals:



CHANGE THE WORLD!

! Dramatic increase in eBusiness apps



TOOL THE WORLD!

! All tools available to all programmers



TRAIN THE WORLD!

! Train Business Partners on tech'g, tools



URLS

Web

Java

RPG

CBL

💧 www.ibm.com/as400/java

▶ **AS/400 Java homepage**

💧 www.ibm.com/software/ad/wdt400

▶ **WebSphere Development Tools for AS/400**

💧 www.ibm.com/software/ad/vajava

▶ **VisualAge for Java**

💧 www.ibm.com/software/ad/vadd

▶ **VisualAge Developer Domain**

💧 www.ibm.com/alphaworks

▶ **Lots of free Java stuff! Eg, Java Beans, XML tools**

Books

Web

Java

RPG

CBL

AS/400 AD, IBM Toronto

💧 www.29thStreetPress.com news/400

▶ **Java and the AS/400, Practical Examples Using VAJava**

💧 www.emergingskills.com

▶ **Java for RPG Programmers, Java for COBOL Programmers**

◆ Plus Student Workbook for Java for RPG and COBOL Programmers

💧 www.mc-store.com/java3.html midrange computing

▶ **Various books on Java and WebSphere**

💧 www.amazon/borders/fatbrain.com, ...

▶ **Tons of books on Java, VisualAge, WebSphere, etc**

💧 www.ibm.com/redbooks.com

▶ **Search for Servlets, JavaServer Pages**



IBM WebSphere Development Tools for AS/400: An Introduction, REDP0503

Abstract

IBM WebSphere Development Tools for AS/400 (5769-CL3) is a complete package of premium AS/400 and iSeries program development tools. These tools help you quickly develop iSeries and AS/400-based applications for e-business. You can do this by using your choice of traditional 5250 green screen interfaces, HTML browser interfaces, or GUI-based interfaces on workstations running the Windows operating system.

The features of WebSphere Development Tools allow you to:

- Generate Java to Web-enable your RPG applications.
- Write host or server code with the CODE/400 workstation-based client/server development environment.
- Develop and maintain applications in many AS/400 languages, including RPG and Java.

This IBM Redpaper redbook targets IBM marketing personnel, Business Partners, and AS/400e and iSeries customers who are looking to extend and expand their information server into e-business. It shows you how to install, tailor, and configure WebSphere Development Tools for AS/400.



Design and Implement Servlets, JSPs, and EJBs for IBM WebSphere Application Server, SG24-5754-00

Abstract

This IBM redbook provides design guidelines for developing e-business applications based on servlets, JavaServer Pages (JSP) and Enterprise JavaBean (EJB) technologies.

The
Ser
con
net
Ser



Domino and WebSphere Together, SG24-5955-00

Abstract

Lotus Domino, IBM WebSphere, and their related products can be used to deliver new value to customers in the form of "collaborative commerce services." In this redbook we explain how you can use the combined capabilities of Domino and WebSphere to provide a complete and integrated platform for collaborative commerce.

Using examples and scenarios, we show you how to install and configure Lotus Domino R5 and IBM WebSphere 3.02 on Windows NT. We then look at how the components in a WebSphere Web application (servlets, JavaServer Pages and Enterprise JavaBeans) can work together with the functionality of Domino.



Servlet and JSP Programming with IBM WebSphere Studio and VisualAge for Java, SG24-5755-00

Abstract

This IBM Redbook provides you with sufficient information to effectively use the WebSphere and VisualAge for Java environments to create, manage and deploy Web-based applications using methodologies centered around servlet, JavaServer Pages, and JavaBean architectures.

In Part 1 we describe the products used in our environment and provide instruction on product installation and configuration. Following this, we cover servlet and JSP programming, which provide you with both a theoretical and practical understanding of these components, together with working examples of the concepts described. For execution of the sample code, we provide information on configuring the WebSphere Application Server and deploying and running the sample Web applications in WebSphere. Using the knowledge developed in these chapters, we then provide detailed information on the development environments offered by VisualAge for Java and