- This presentation looks at the application models today in the 5250 environment and in the Web environment. It then looks at the user interface definitions for the Web. Instead of DDS, HTML files or Java server pages to define the user interface are used. To call programs on the iSeries from a Java program, an XML language called PCML is used to describe the interface between the Java program and the iSeries program. The presentation looks at the WebTools provided in WDSc and walks through a scenario that describes the steps to create a simple interaction, that has an input screen asking for a customer number and that fetches the data from the iSeries data base by calling an RPG program.

Disclaimer

Acknowledgement:

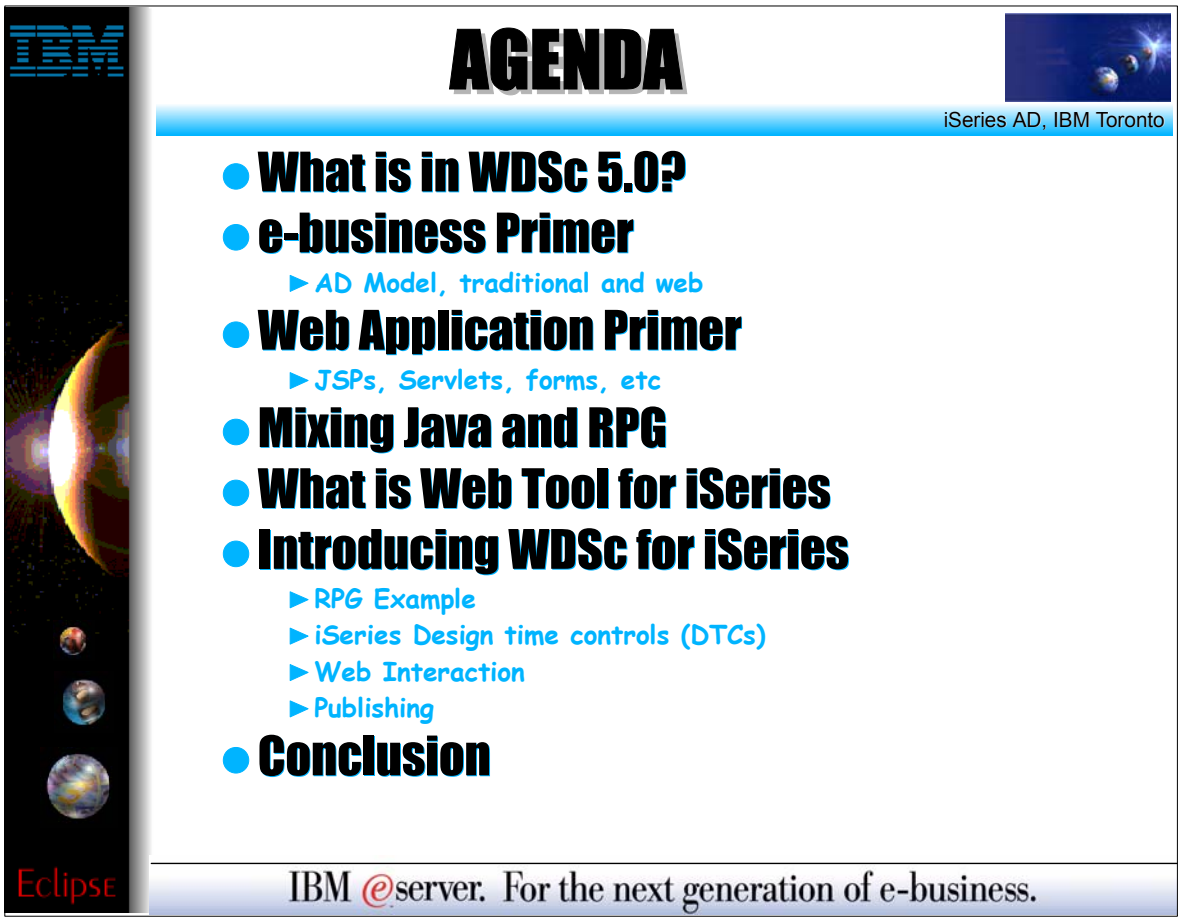
- This presentation is a collaborative effort of the IBM Toronto AS/400 Application Development presentation team, including work done by:
  - *Phil Coulthard, George Farr, Claus Weiss, Don Yantzi*

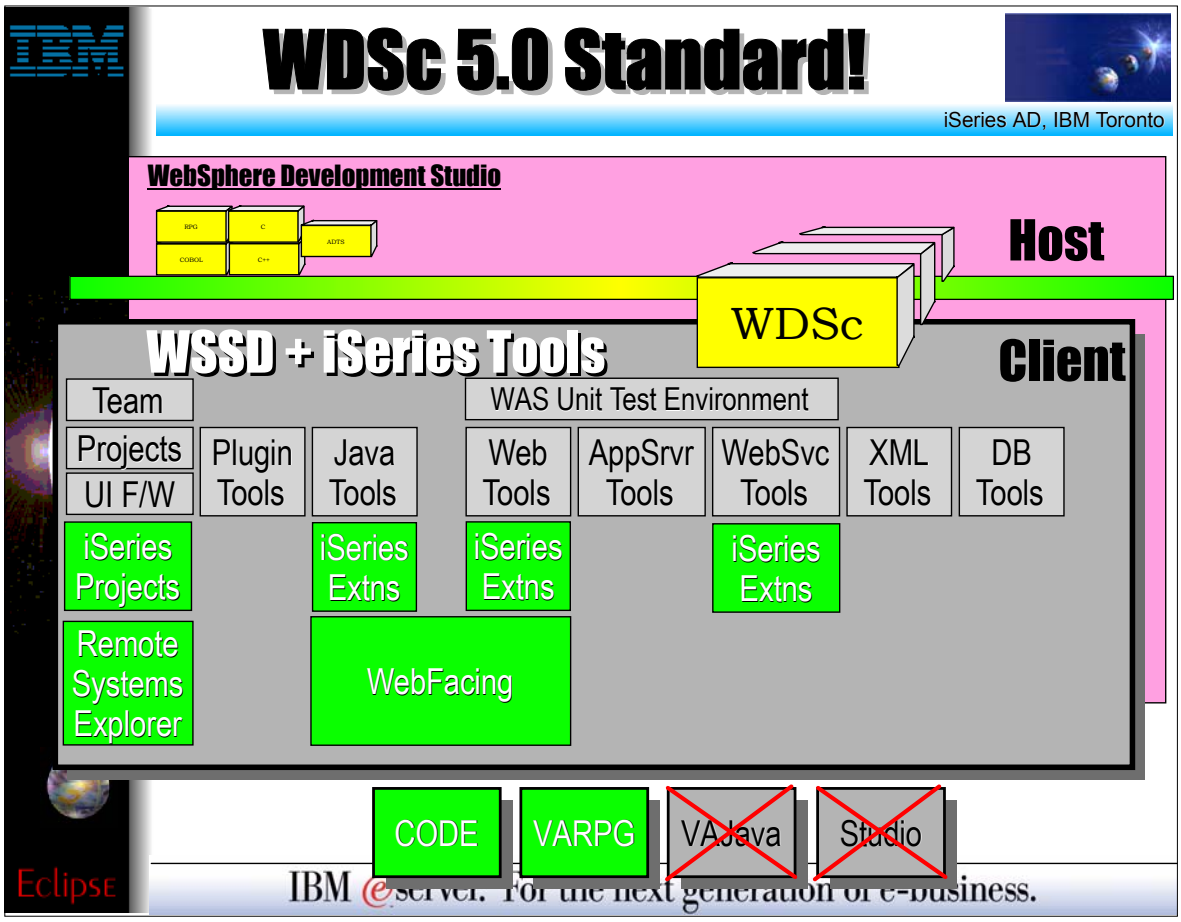
Disclaimer:

- 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.

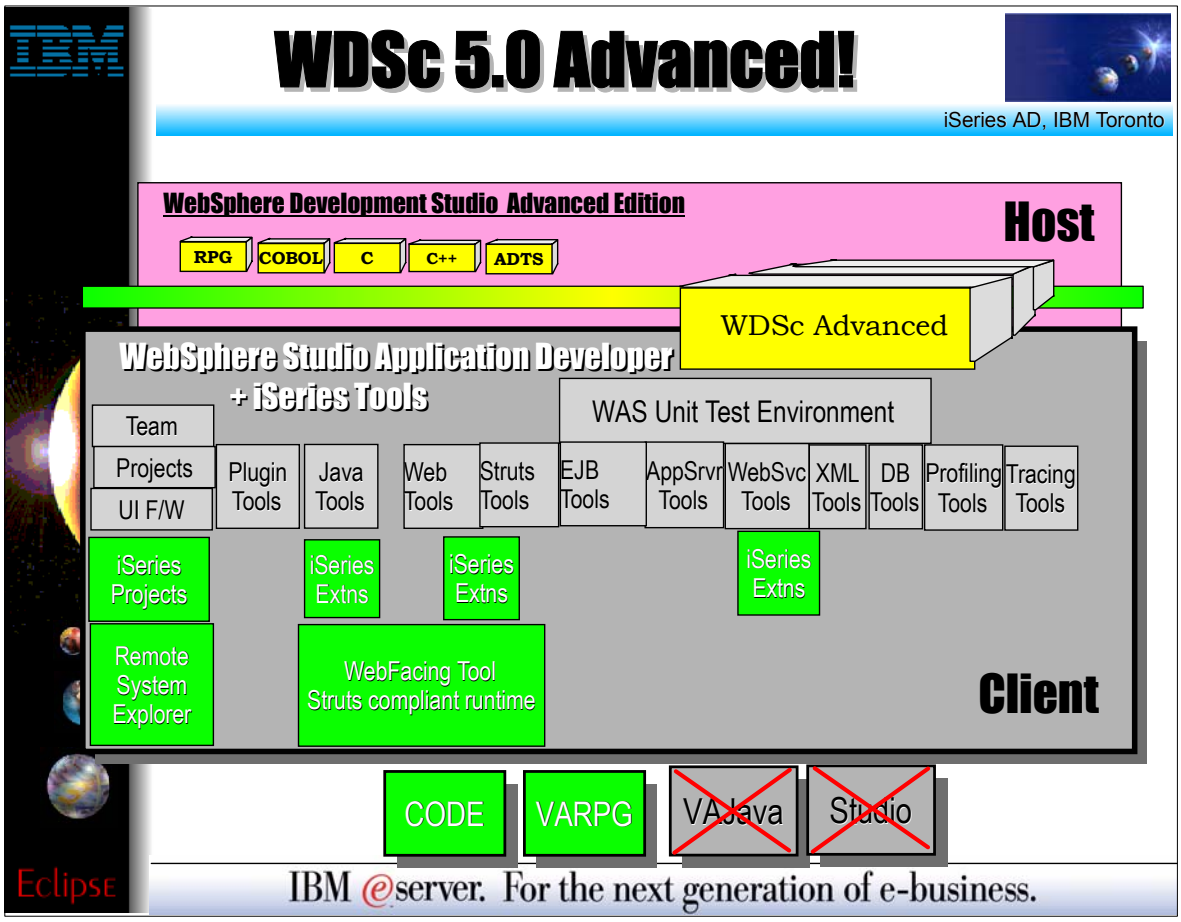
Reproduction:

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


IBM
AGENDA
iSeries AD, IBM Toronto
What is in WDSc 5.0?
e-business Primer
AD Model, traditional and web
Web Application Primer
JSPs, Servlets, forms, etc
Mixing Java and RPG
What is Web Tool for iSeries
Introducing WDSc for iSeries
RPG Example
iSeries Design time controls (DTCs)
Web Interaction
Publishing
Conclusion
Eclipse
IBM @server. For the next generation of e-business.



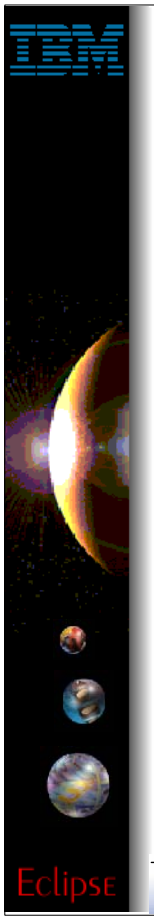
- The next generation of workstation development tools to create iSeries e-business applications that leverage the iSeries and the power of WebSphere Application Server
- Inherits and extends the WebSphere Studio Site Developer's robust, easy-to-use development environment for creating, building, and maintaining dynamic Web sites, or developing Java, Web and Web services applications. It also includes a robust, easy-to-use development environment for creating, building and maintaining iSeries RPG, COBOL, C, C++ applications, and Web-enabled applications using the IBM WebFacing Tool. The Remote Systems Explorer also makes it easy to access UNIX, Windows, and Linux application resources.

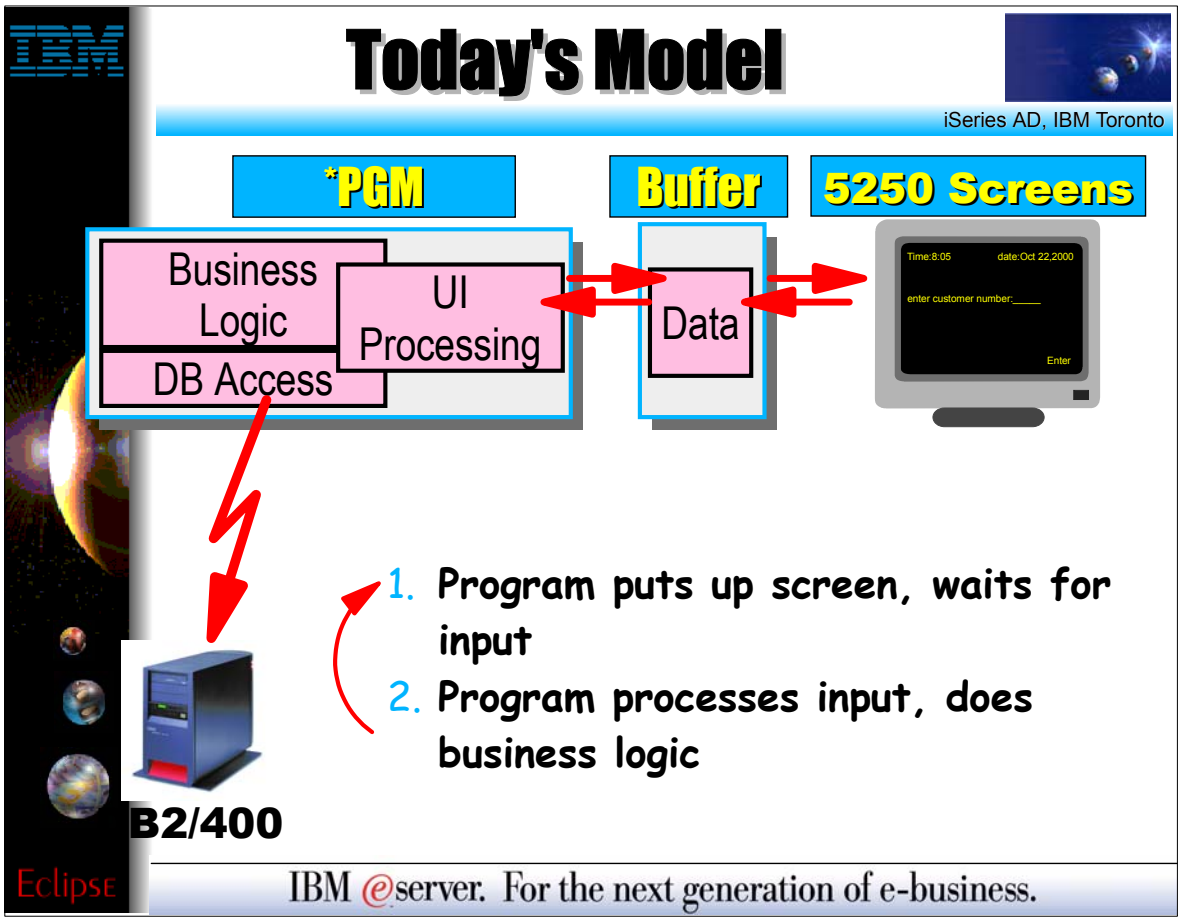


- The next generation of advanced workstation development tools to create iSeries e-business applications that leverage the execution capabilities of WebSphere Application Server
- Inherits and extends the application development environment for the creation and maintenance of J2EE and Web services applications from WebSphere Studio Application Developer V5.0. It also includes a robust, easy-to-use development for creating, building, and maintaining iSeries RPG, COBOL, C, C++ applications, and Web-enabled applications using the IBM WebFacing Tool. The Remote Systems Explorer also makes it easy to access UNIX, Windows, and Linux application resources.

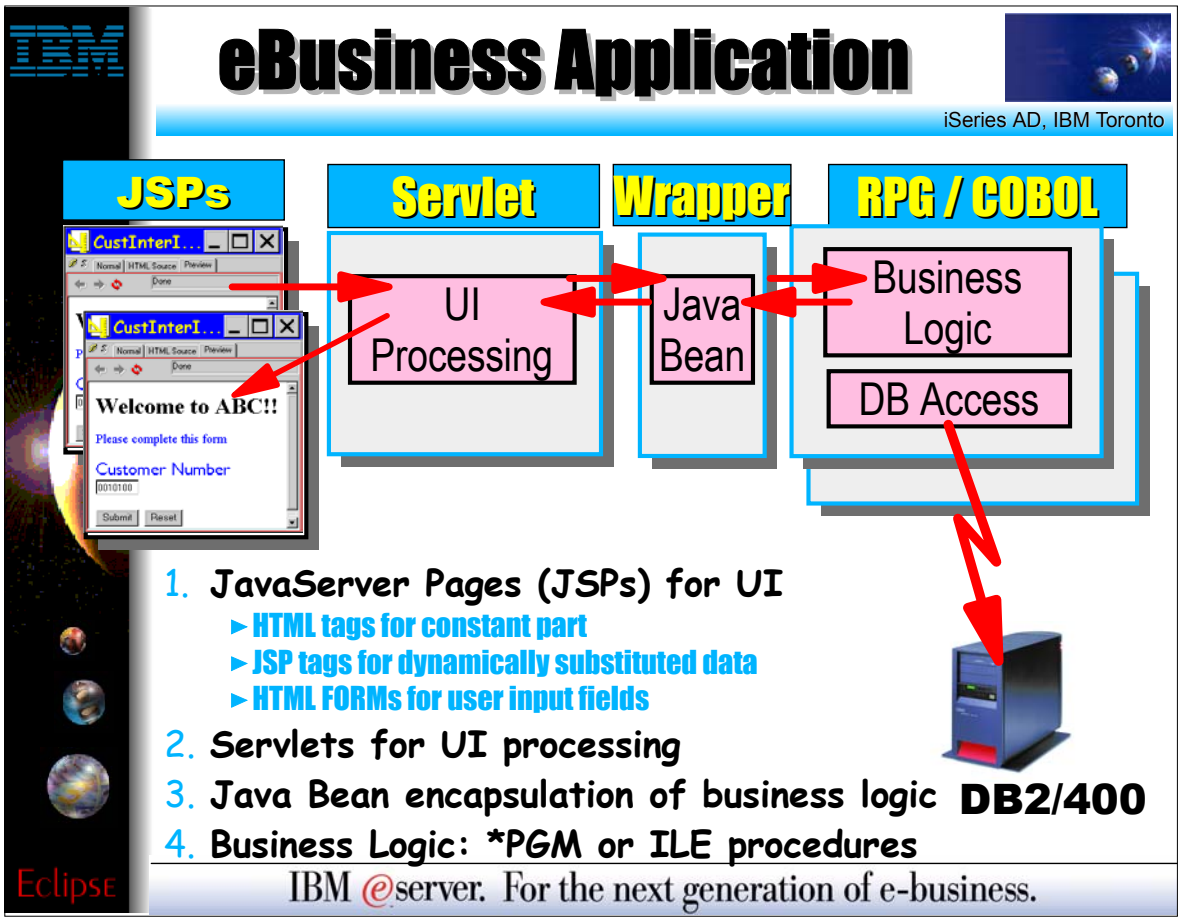
AGENDA

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

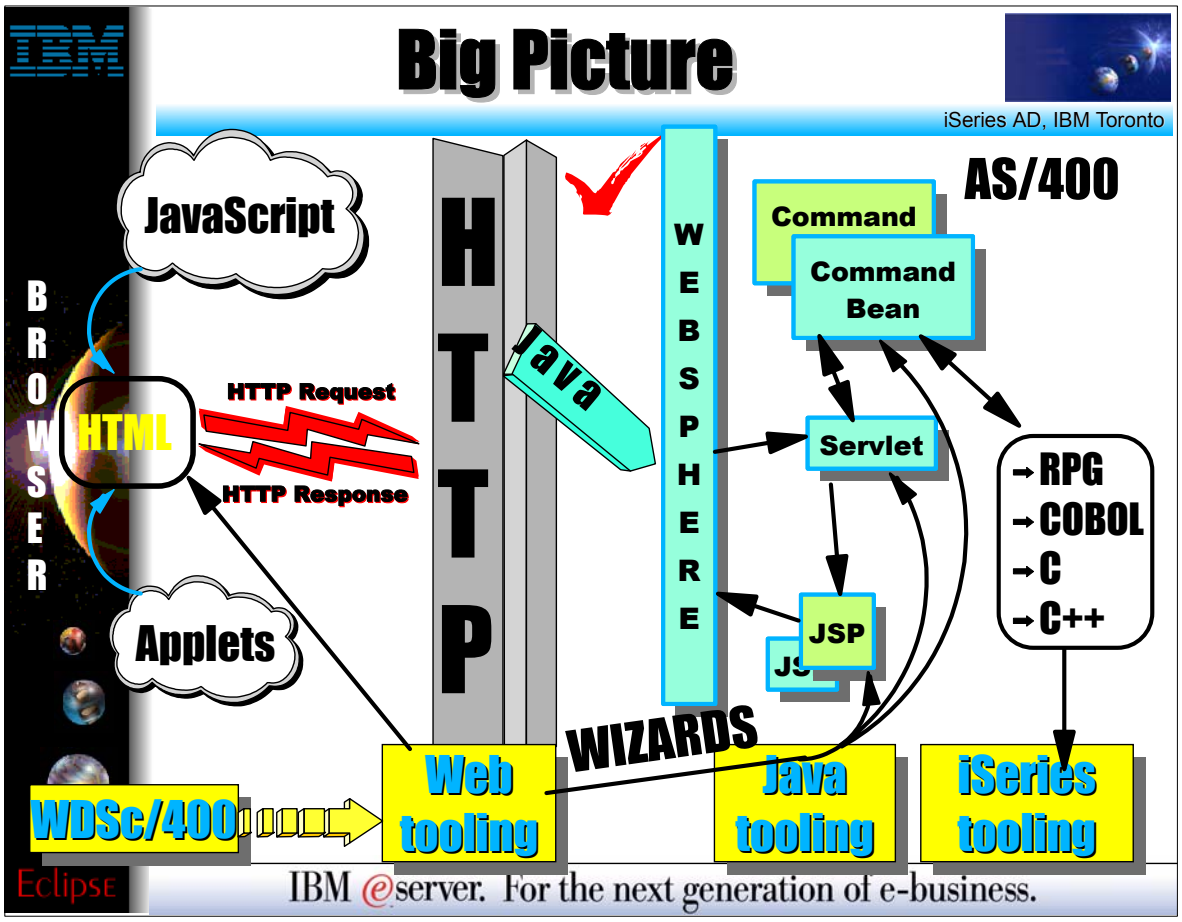




- in today's application model, the program drives the user interface, the program waits for input, after the input is received it presents a new panel and waits for input again.



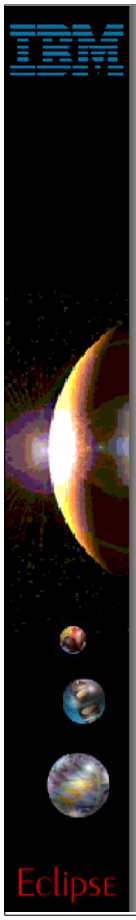
- In the Web the user interface is presented by a browser not a 5250 device.
- The user requests services from the server.
- The server program responds to the various requests.
- There is 3 tiered application divided in:
- User Interface (jsp's)
- Controller logic running in an application server (servlets)
- Business logic running on a business server RPG/COBOL/C/C**/JAVA running on iSeries

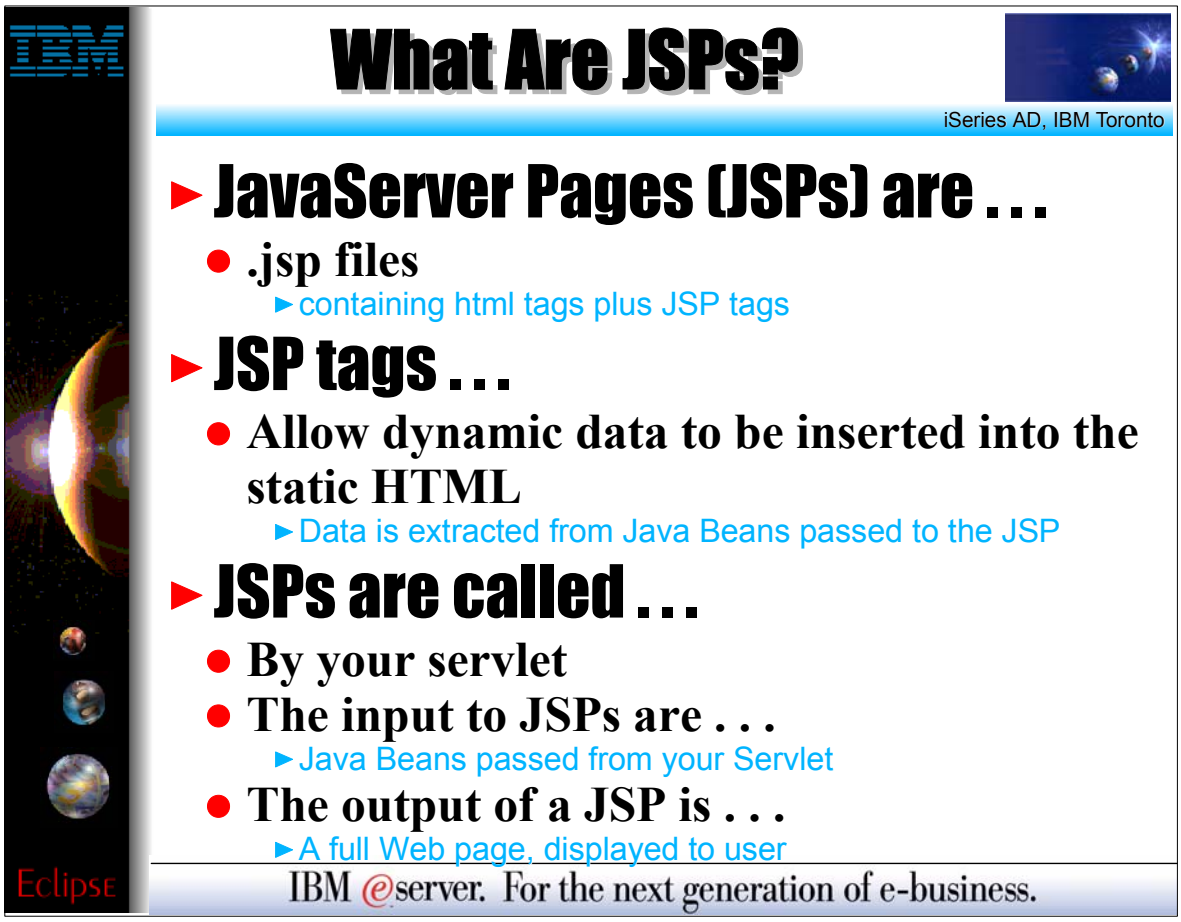


- On the bottom of this slide the tools to develop the different pieces of a 3 tiered application are listed. All these tools are included in the WDSc package.
- WebTooling addresses the Web user interface and the deployment of applications
- Java tooling is to write the middle tier Java servlets
- iSeries RPG/COBOL/CL/DDS tooling is to help create the business logic

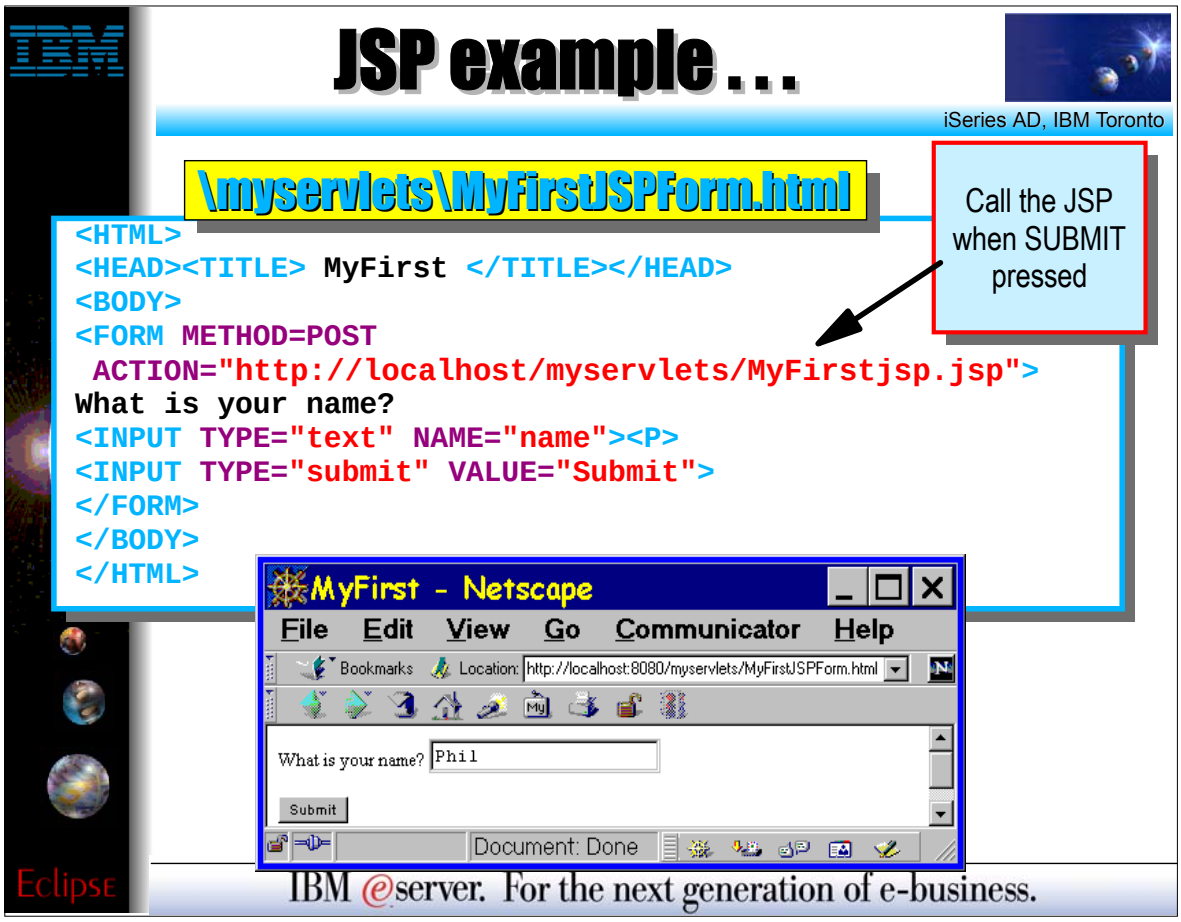
AGENDA

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

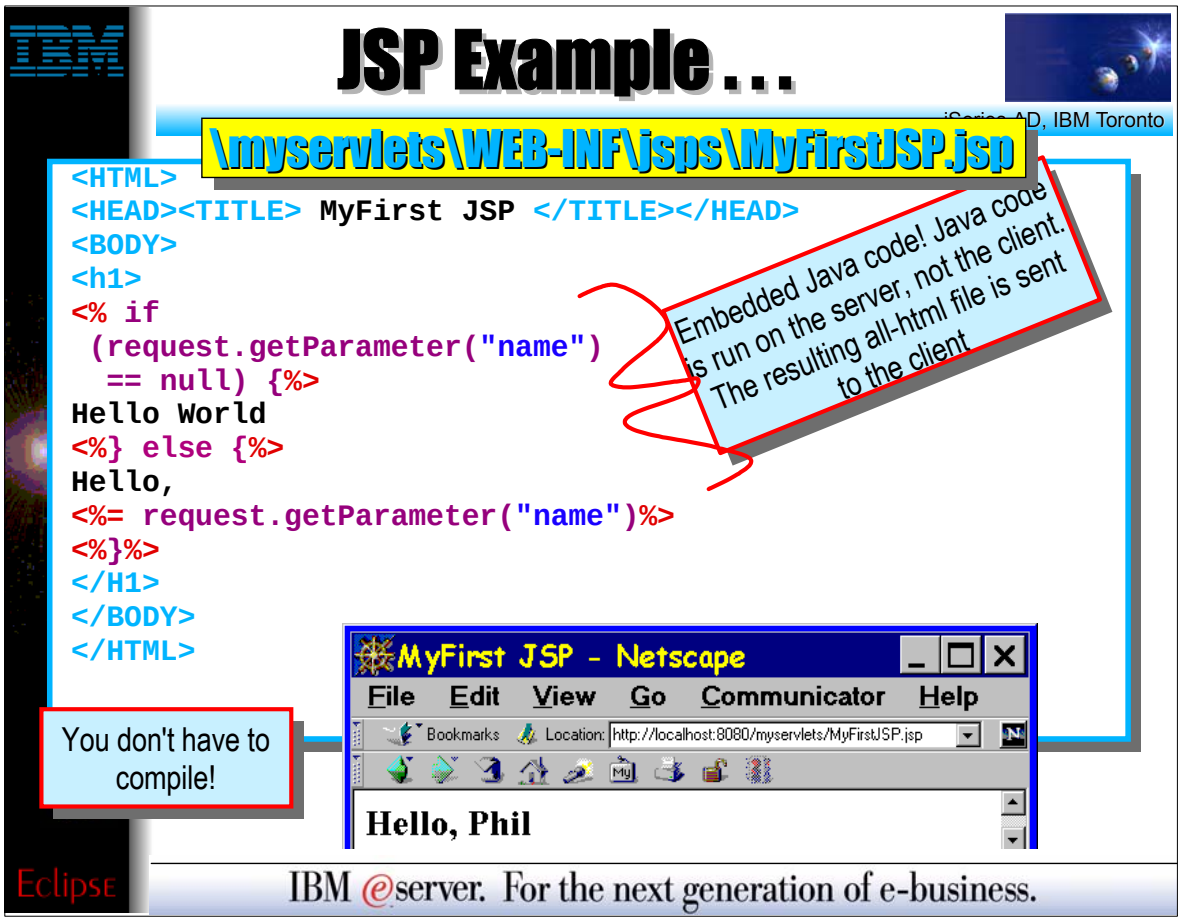




- Java Server Pages are the way to define the dynamic Web user interface.
- Historically, the first step was to use CGI bin programs to serve dynamic web pages which have a limit in scalability and are not very straight forward to use to define the user interface.
- The first attempt to move away from CGI bin programs was to move to servlets and mix the user interface definition and controller logic in the servlet.
- It turned out not to be a very good way to work because of the mix of UI definition and logic.
- In todays environment, the user interface definition is moved out of the logic and stored in separate JSP files, these files contain html tags to describe the user interface and they also contain small pieces of Java code to add dynamic content to the user interface.
- The jsp's get converted to servlets at runtime and the Java code gets executed on the server before the resulting html tags get sent to the browser.



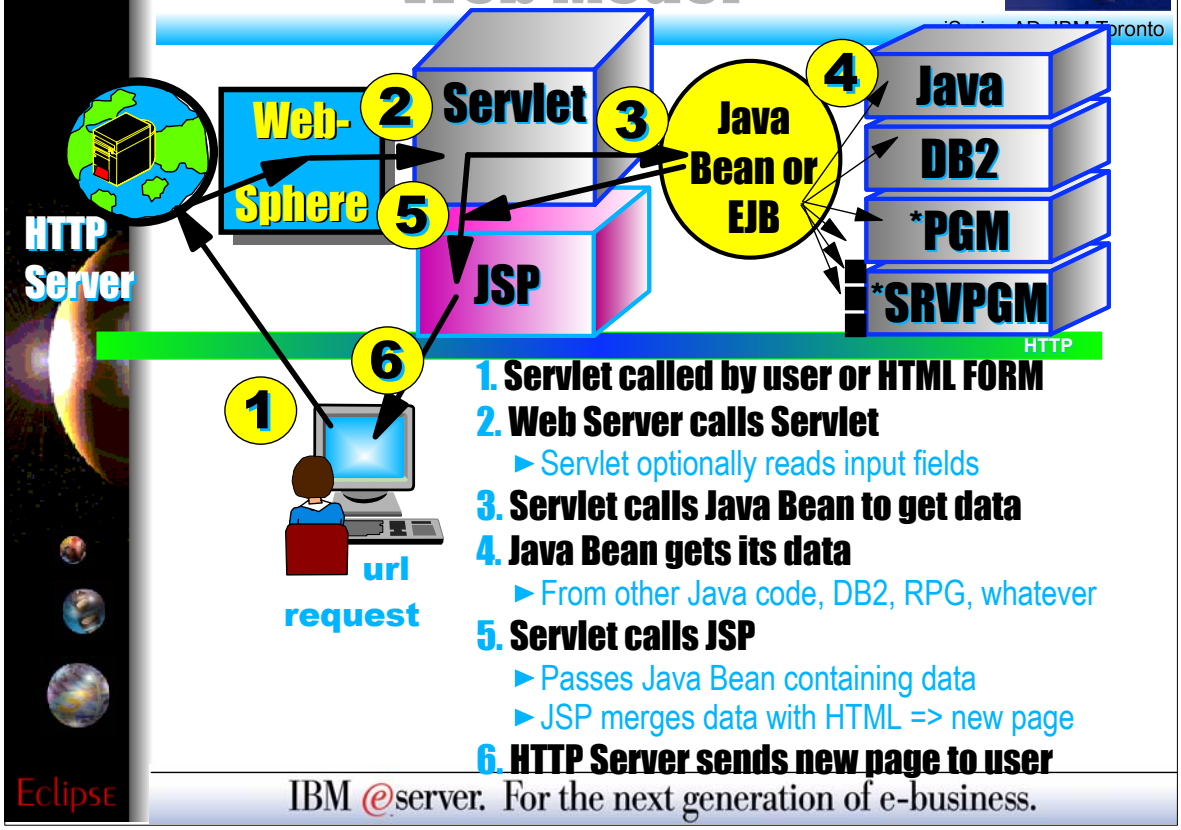
- Shows an example of an html file that contains a form that invokes a jsp on the server.
- The form tag in html is used to pass data from the Web page to the server. It is also used to indicate that a program should be invoked on the server instead of serving just a simple request for a new Web page.
- The result of this simple html file being rendered in the browser is shown in slide.
- Note an entry field "type text", which has the Name name and the push button "type submit" to submit the request to invoke the MyFirstjsp.jsp on the server and pass the data in the entry field to the server.



- Shows the jsp that will get invoked from the form on the previous slide.
- There are normal html tags and there is some Java code included inside the <% %> Java tags
- If the entry field is empty when the request gets submitted, "Hello World" gets displayed in the resulting Web page
- If there is data in the entry field "Hello and the value in the entry field" are displayed


IBM
What Are Servlets?
iSeries AD, IBM Toronto
Servlets are . . .
Java classes (programs written in Java)
Servlets run . . .
On the server (eg, AS/400)
Servlets are called . . .
By your HTTP Server software
When a user goes to your Web page
The input to Servlets are . . .
User-entered data from a Web page
The output of a Servlet is . . .
Java Bean, passed to a JavaServer Page
Eclipse
IBM @server. For the next generation of e-business.

Web Model



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

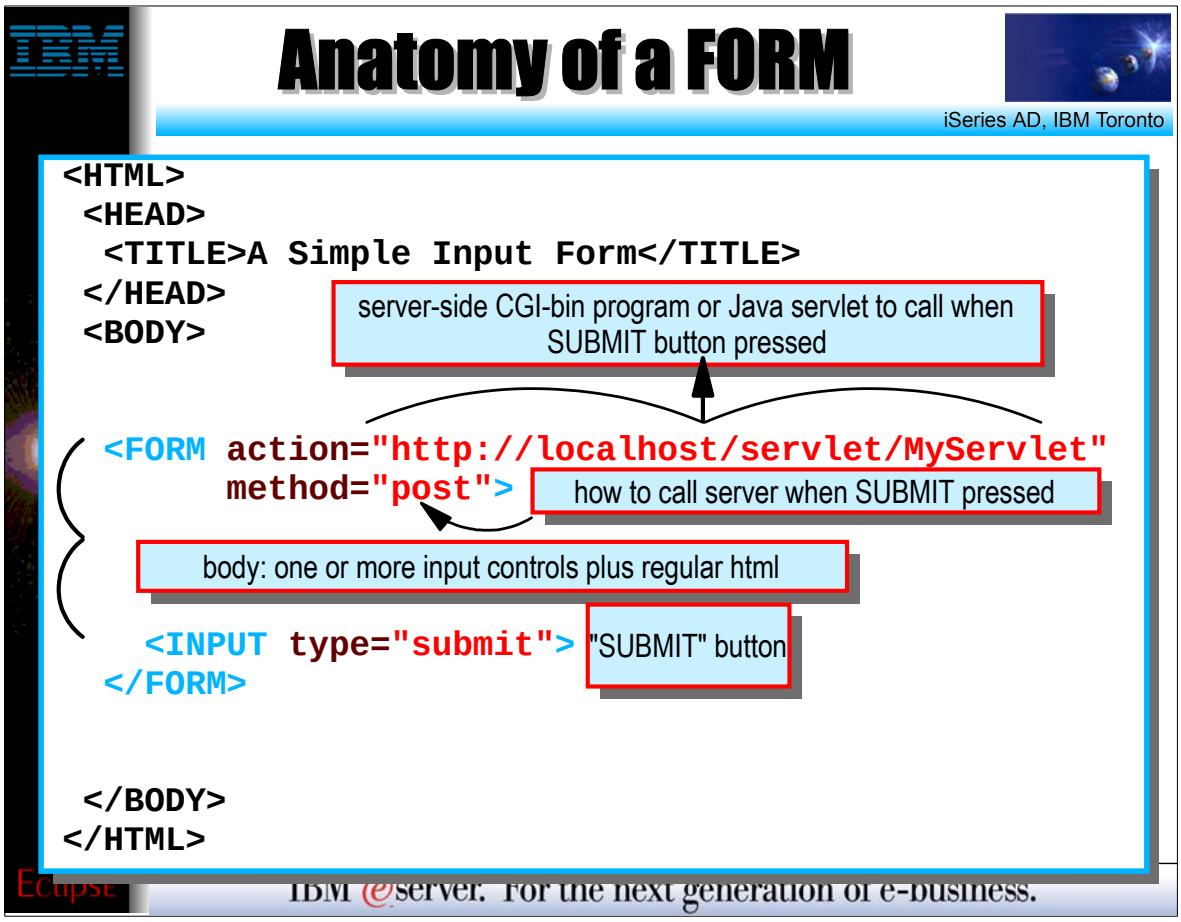
Eclipse

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

HTML Forms

- 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



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

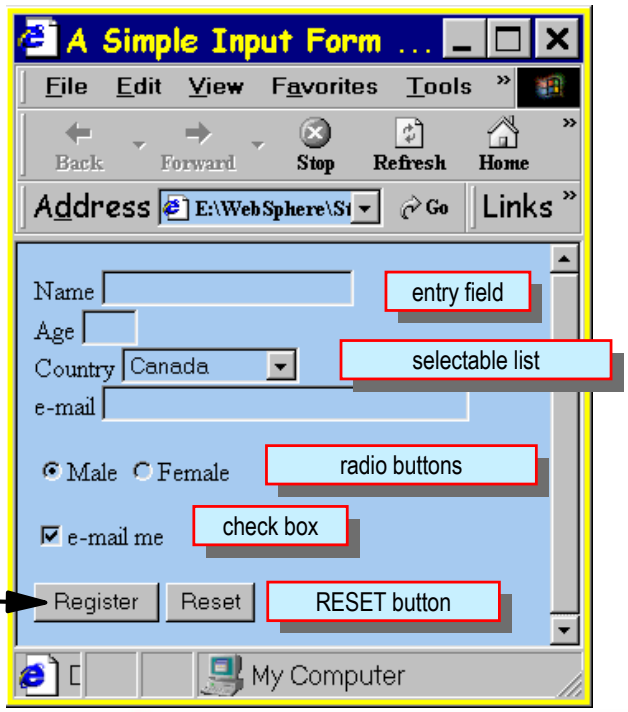
check box

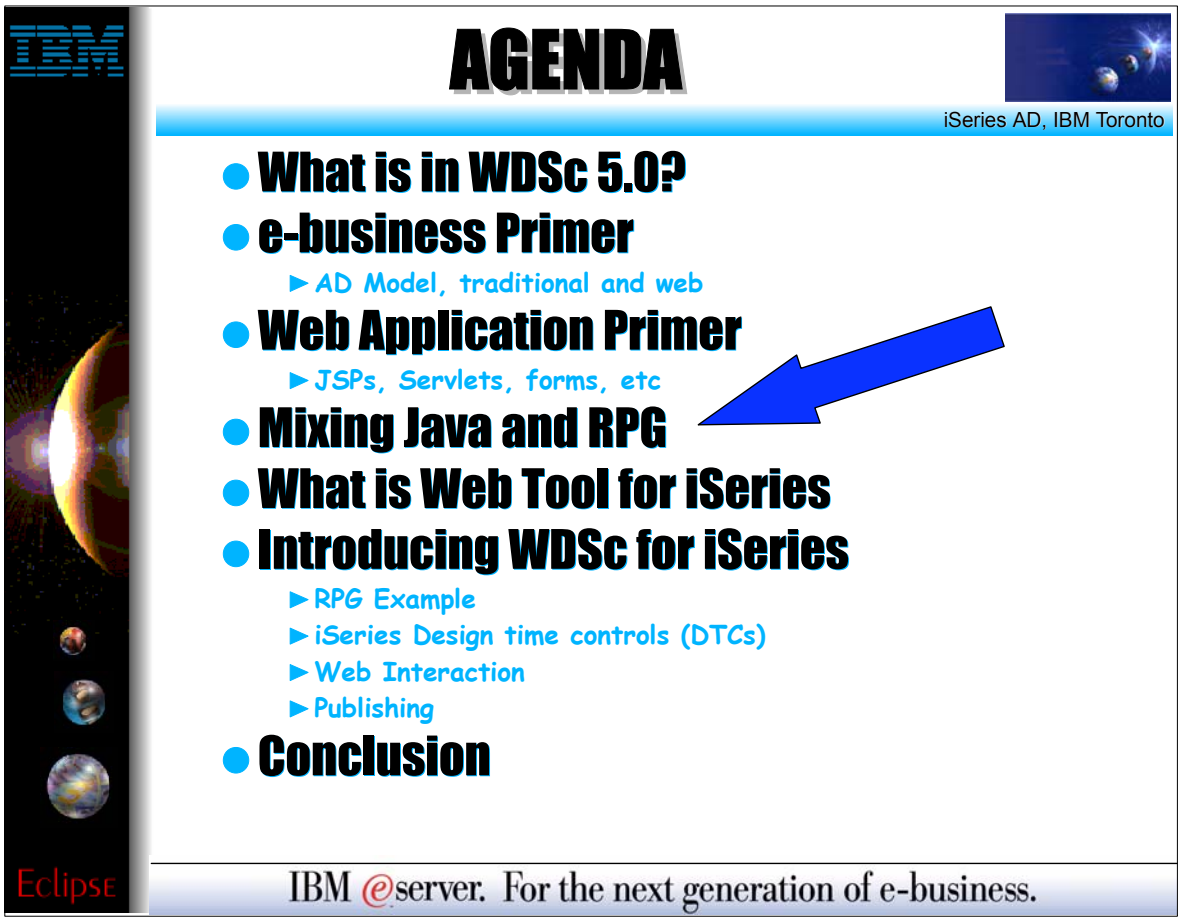
SUBMIT button

RESET button

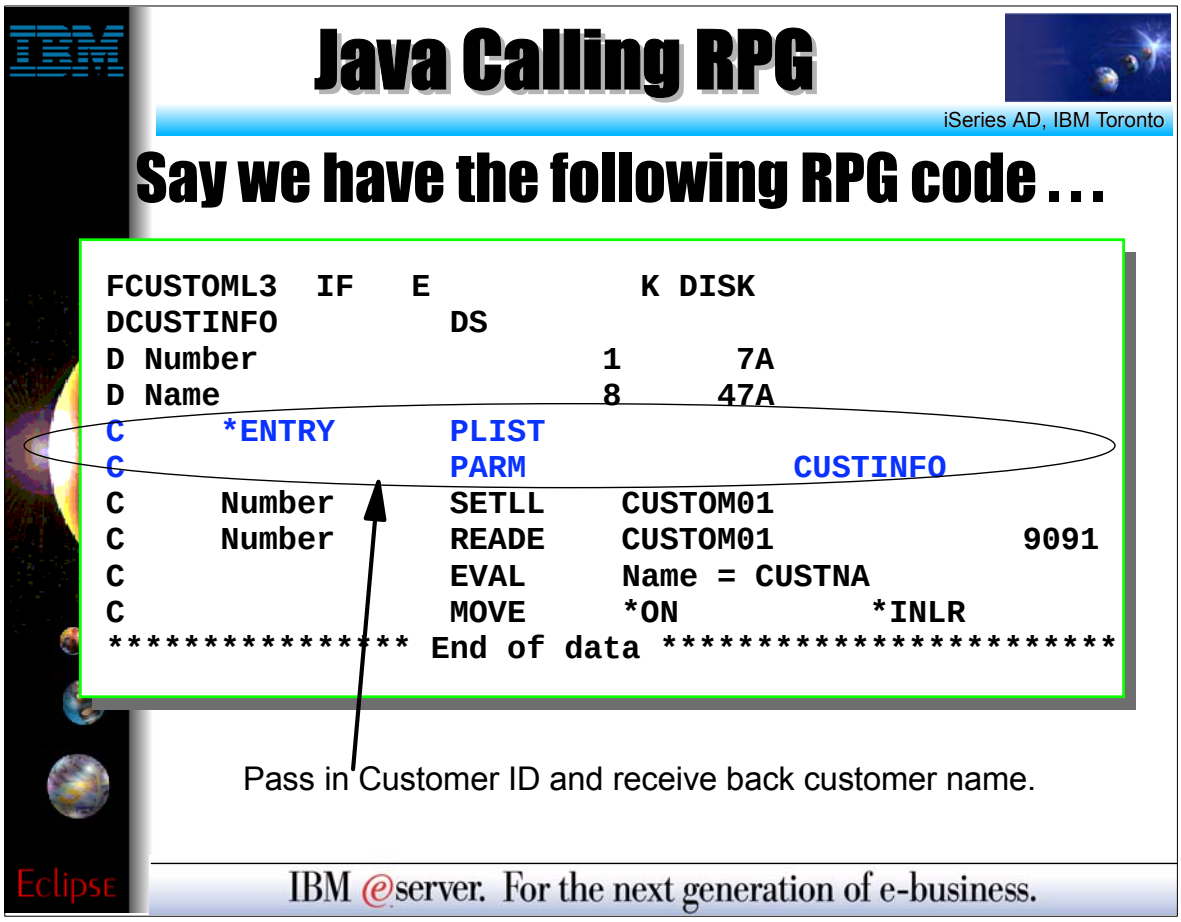
http://www.w3.org/TR/REC-html40/

Example of a FORM





- Now you understand how jsp's and servlets work. The next slides define the interface (signature) of an iSeries program to make it easy to call it from Java.



- Here, the RPG program to call expects one parameter, that is defined as a data structure. The data structure contains 2 fields: Number and Name

- This slide shows the definition of this interface in PCML.
- First on top is how the structure is defined
- It has a name "custinfo" in this case and 2 fields are defined inside the structure
- These are the fields Number and Name they have the same data definition as in the RPG program on the previous slide.
- Also the number field gets intialized with a value
- At the bottom of the slide the name of the program and where it is located on the iSeries is defined. Also defined is to pass the structure as a parameter

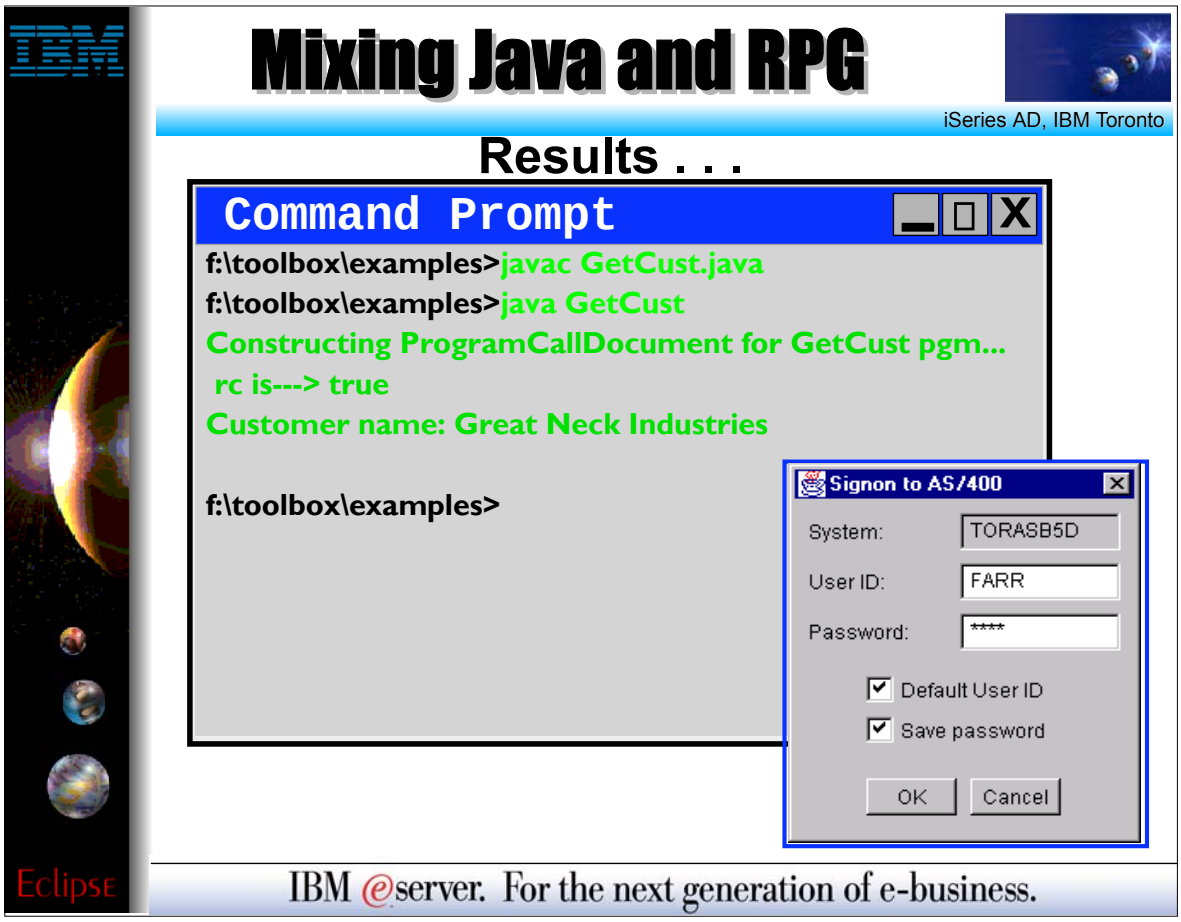
Call RPG from Java Servlet

File: GetCust.java
Class: GetCust

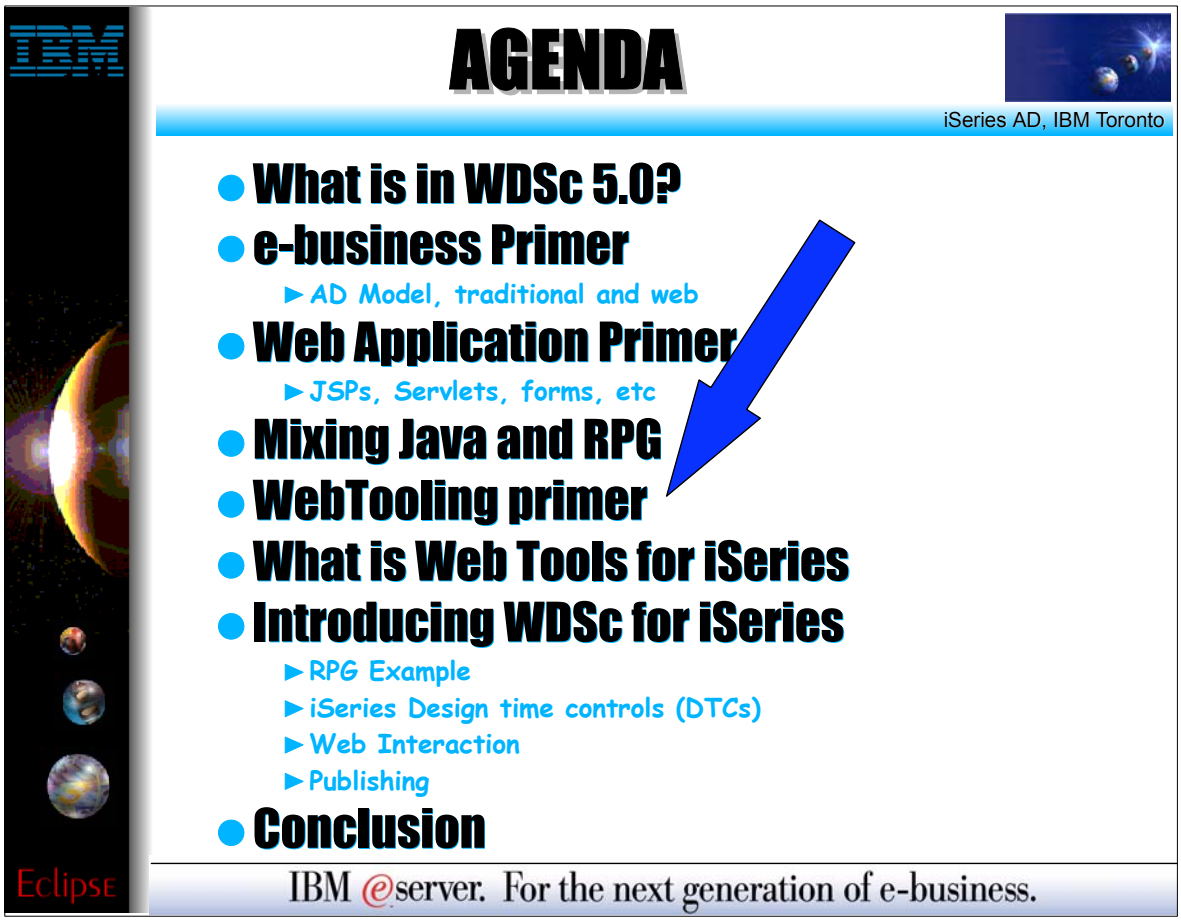
```
public static void main(String[] argv)
{
    AS400 as400System = new AS400();
    ProgramCallDocument pcml = null;
    String msgId, msgText;
    Object value = null;

    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
```

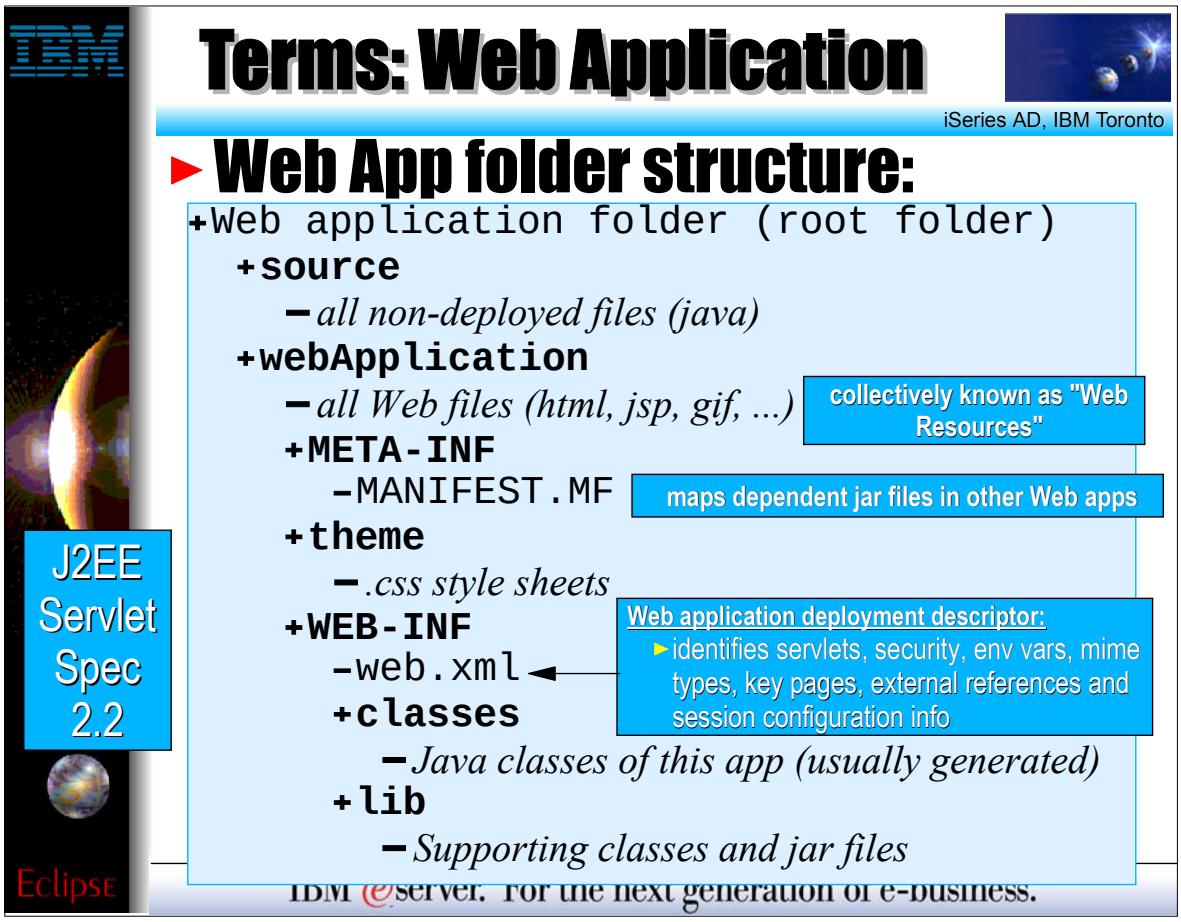
- Here is the sample code in Java that now allows the use of the PCML interface description of our RPG program.
- In Java the interface description in the PCML file is used and along with a call the program described in the PCML file.
- This code will call the RPG program, it will pass a customer number to the RPG program which will fetch the corresponding customer name from the database



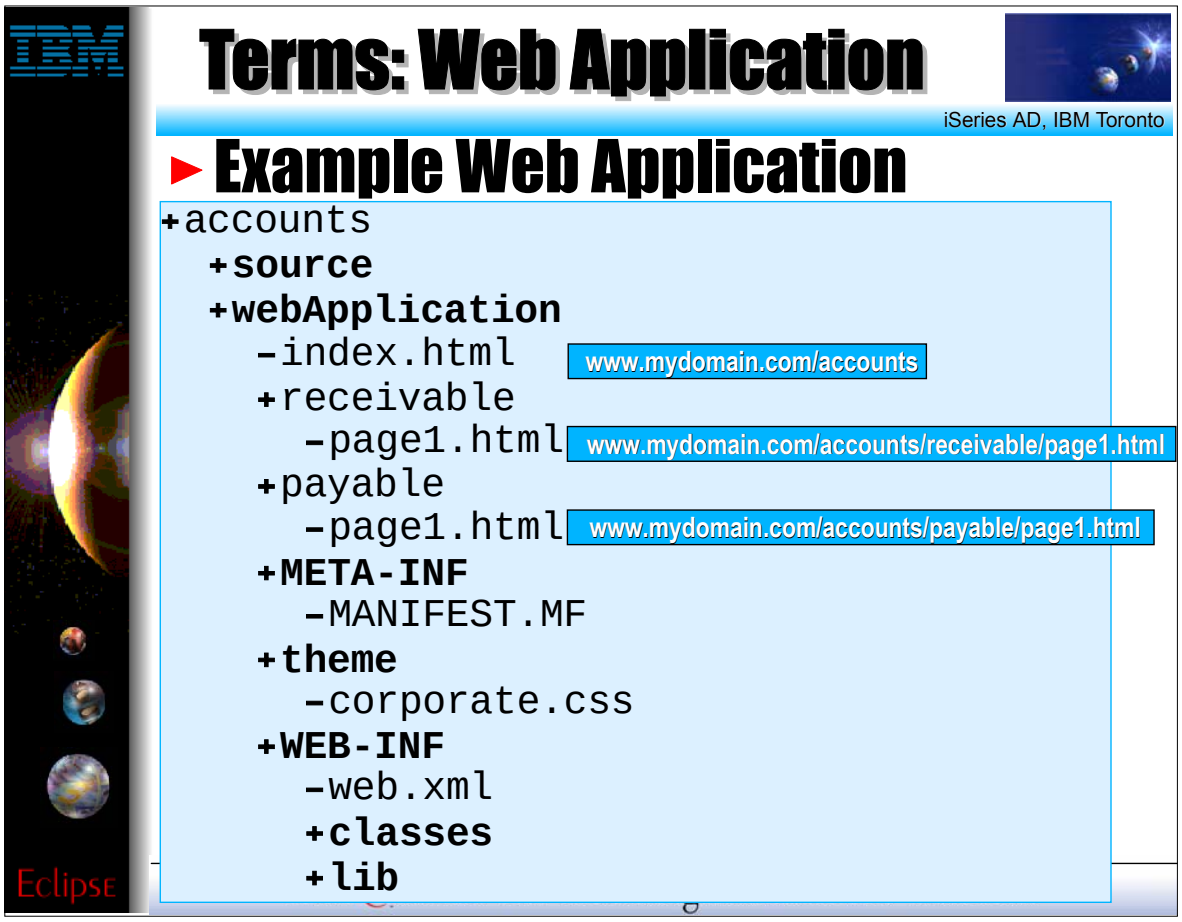
- Here is an example of running the example and getting a customer name for the default customer number back. Note that to access the iSeries program requires the user to logon to the iSeries with a normal authentication dialog.



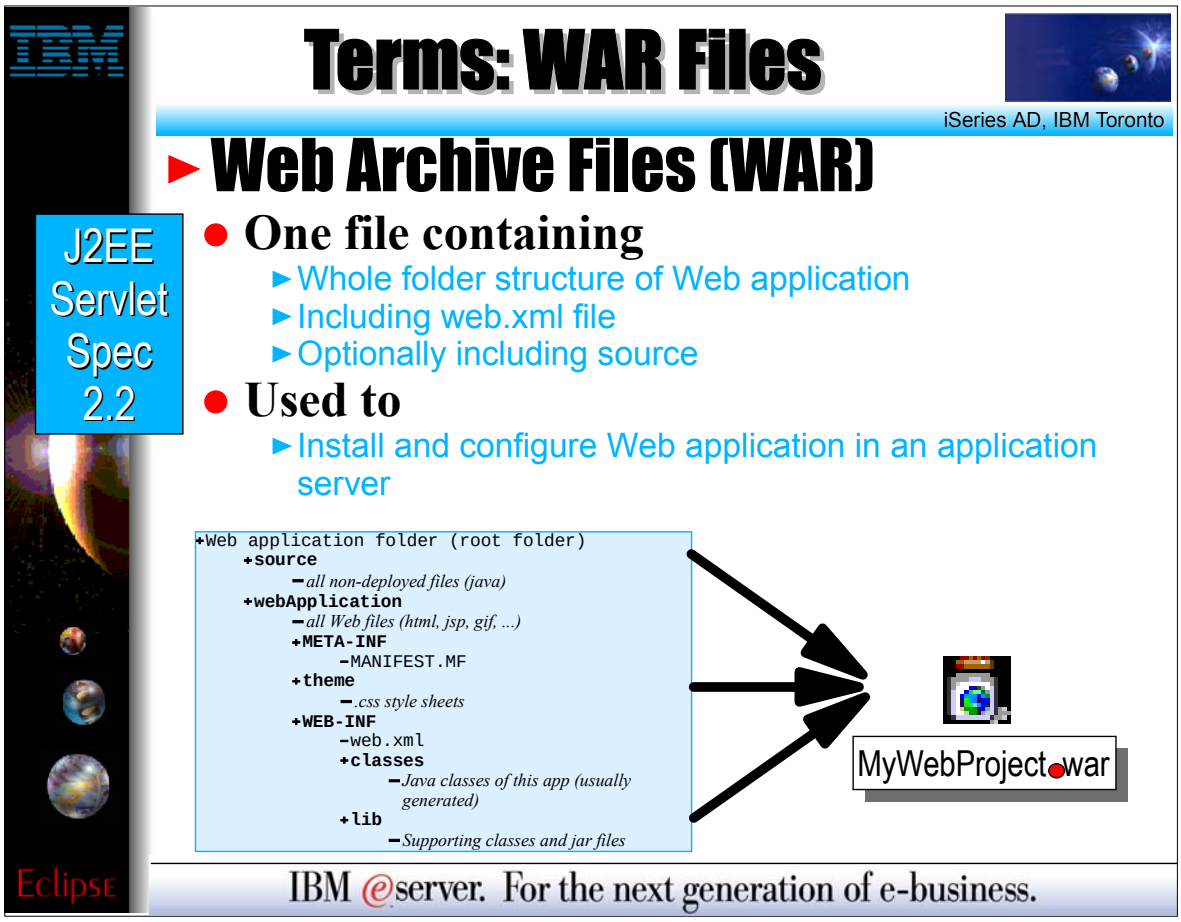
- Next is Web applications, since we now understand how to define the interface of an iSeries program in XML and how to use the XML definition in Java



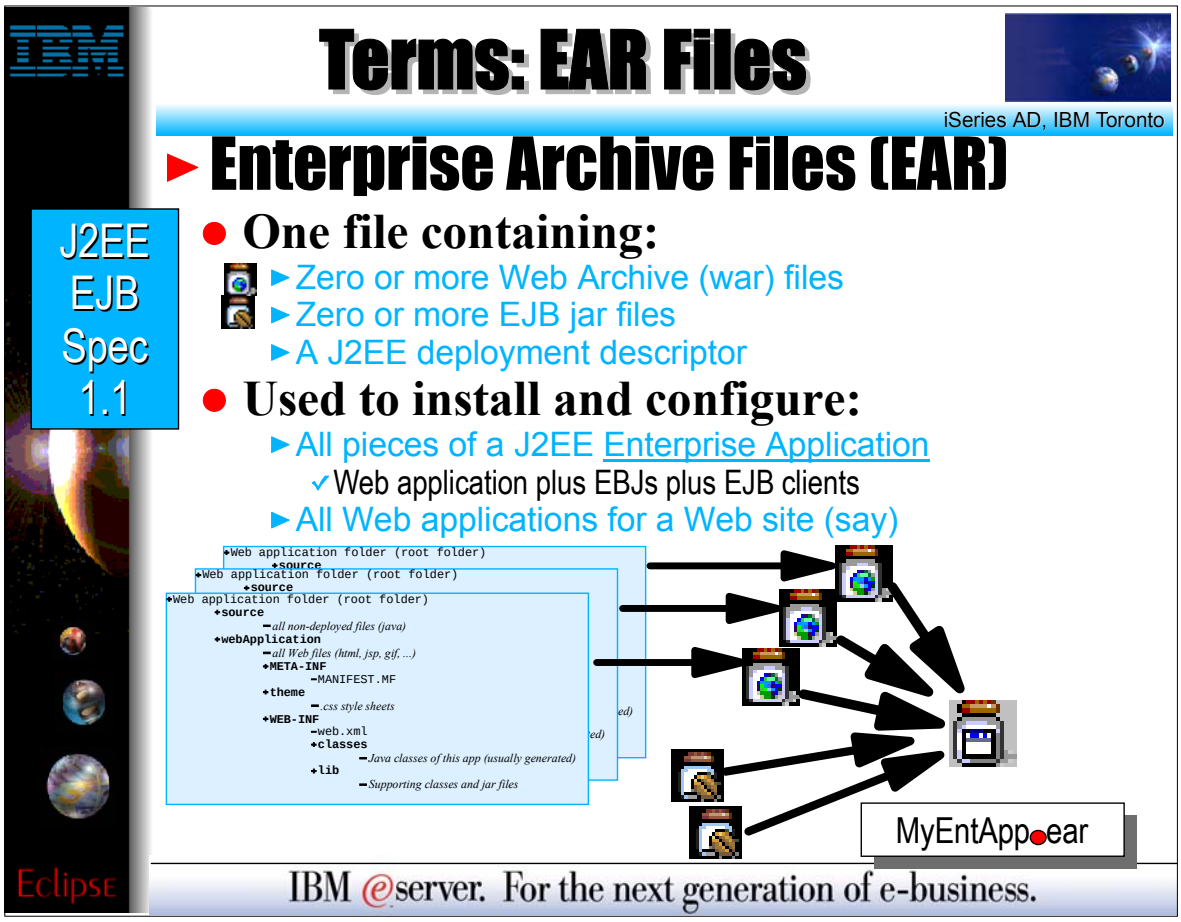
- Here is the Web Application folder structure.
- The Web application will run in an application server.
- The application server expects the application to conform to certain standards in the way its files are stored.
- The slide shows the standard for the J2EE Spec 2.2
- WAS 5.0 implements this standard and the Web tools in WebSphere Studio will generate and work with Web applications conforming to this standard.



- Shows an example of the Web application folder structure and where in this folder structure the files of a Web application have to be stored

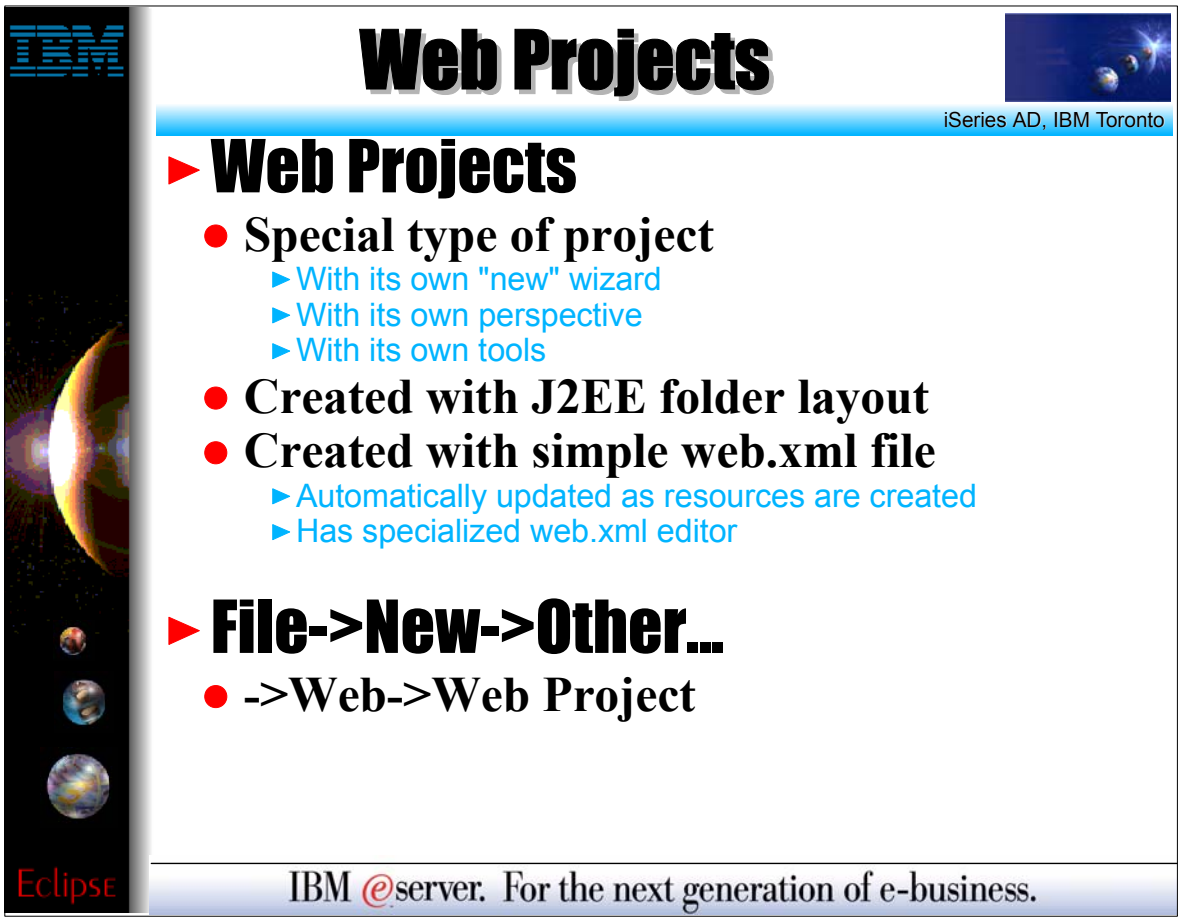


- A Web Archive (WAR) file, is a file that can store a full web application in a single file similar to a ZIP file, storing the web application makes it easy to move it around and deploy it to different servers.
- The application server has the capability to take the WAR file and create a web application inside the application server from it.

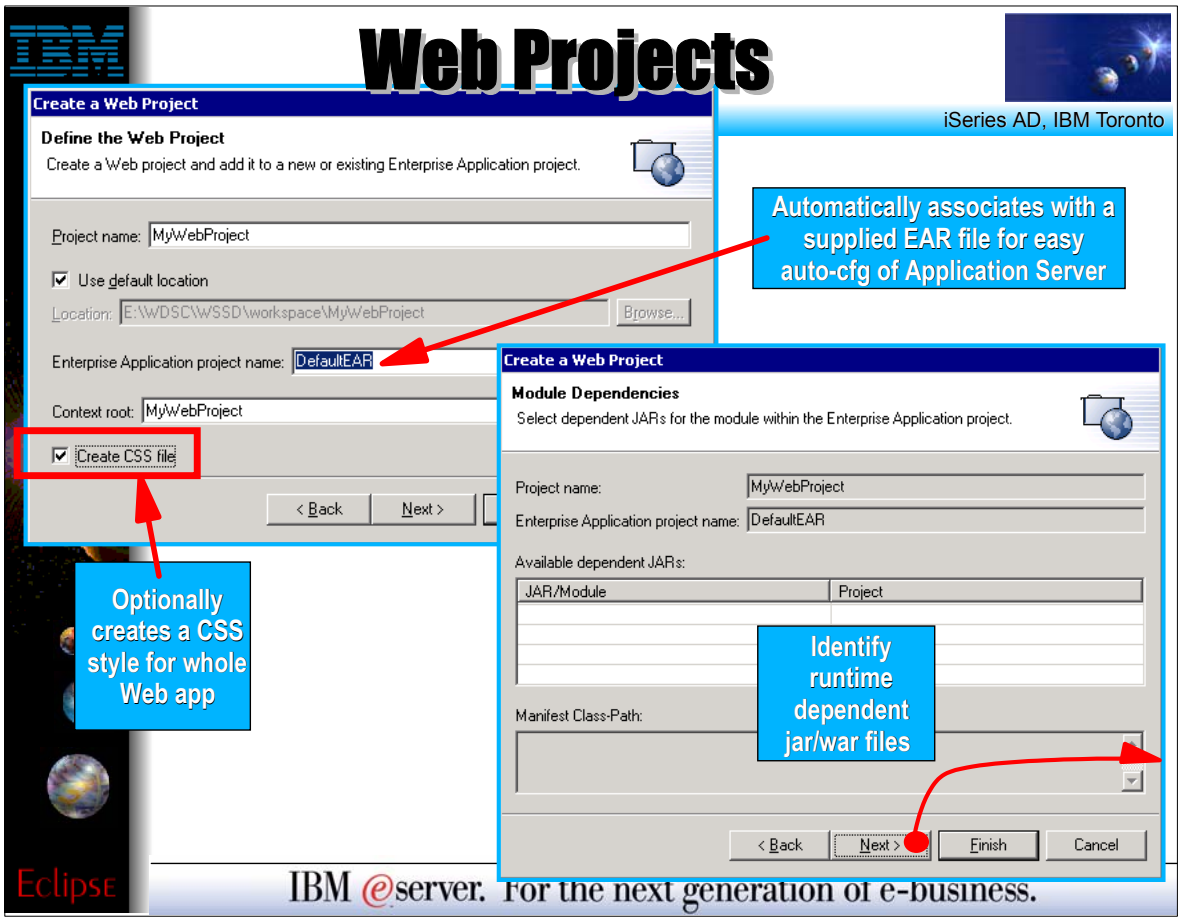


- An EAR file takes the concept of a WAR file one step further and stores multiple WAR files (Web applications) in one EAR file
- EAR files can also directly be installed in WebSphere Application Server 5.0

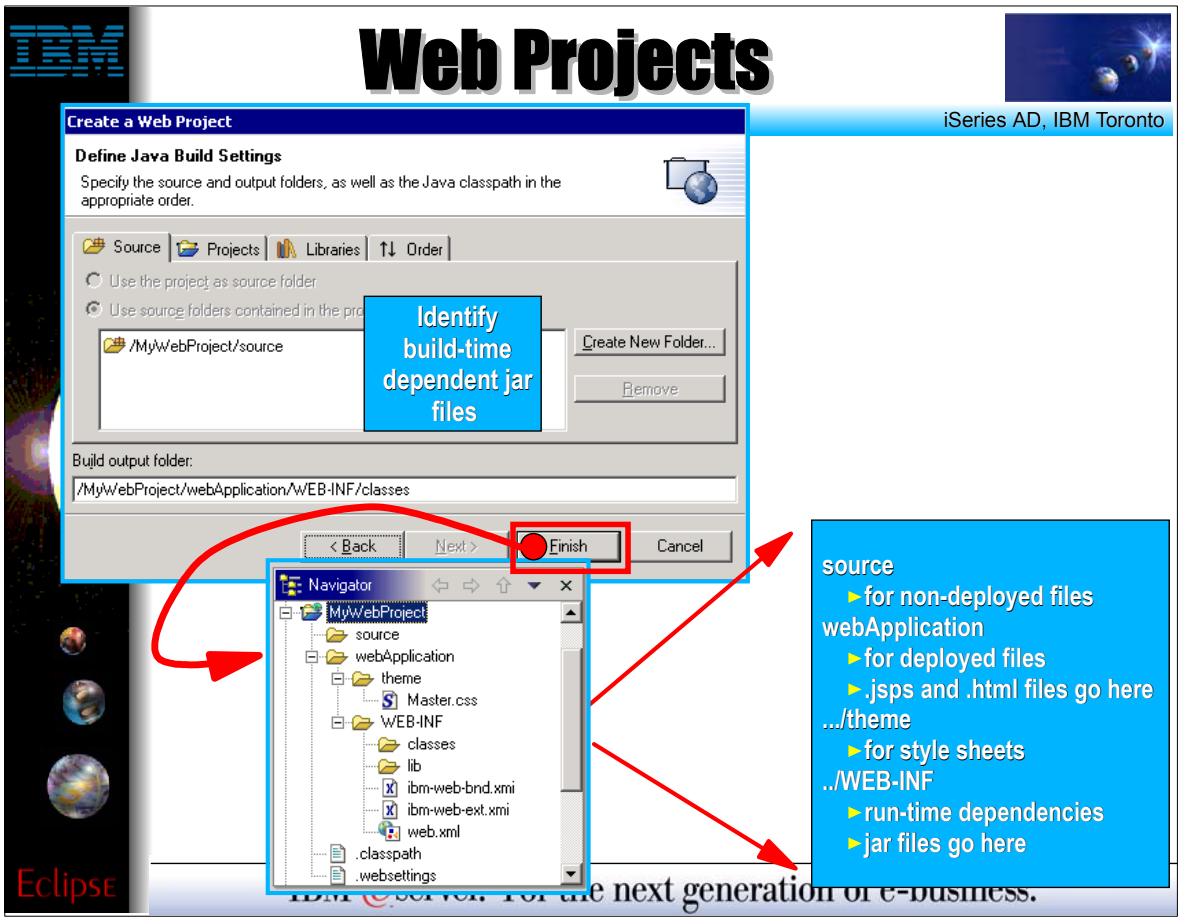
- Now that we know what it takes to build a Web application let's look at what the WDSc tooling can do to make the creation of Web applications easier
- Not to worry if this all sounds too complicated, the WDSc tools help by creating most of the pieces of a Web application.

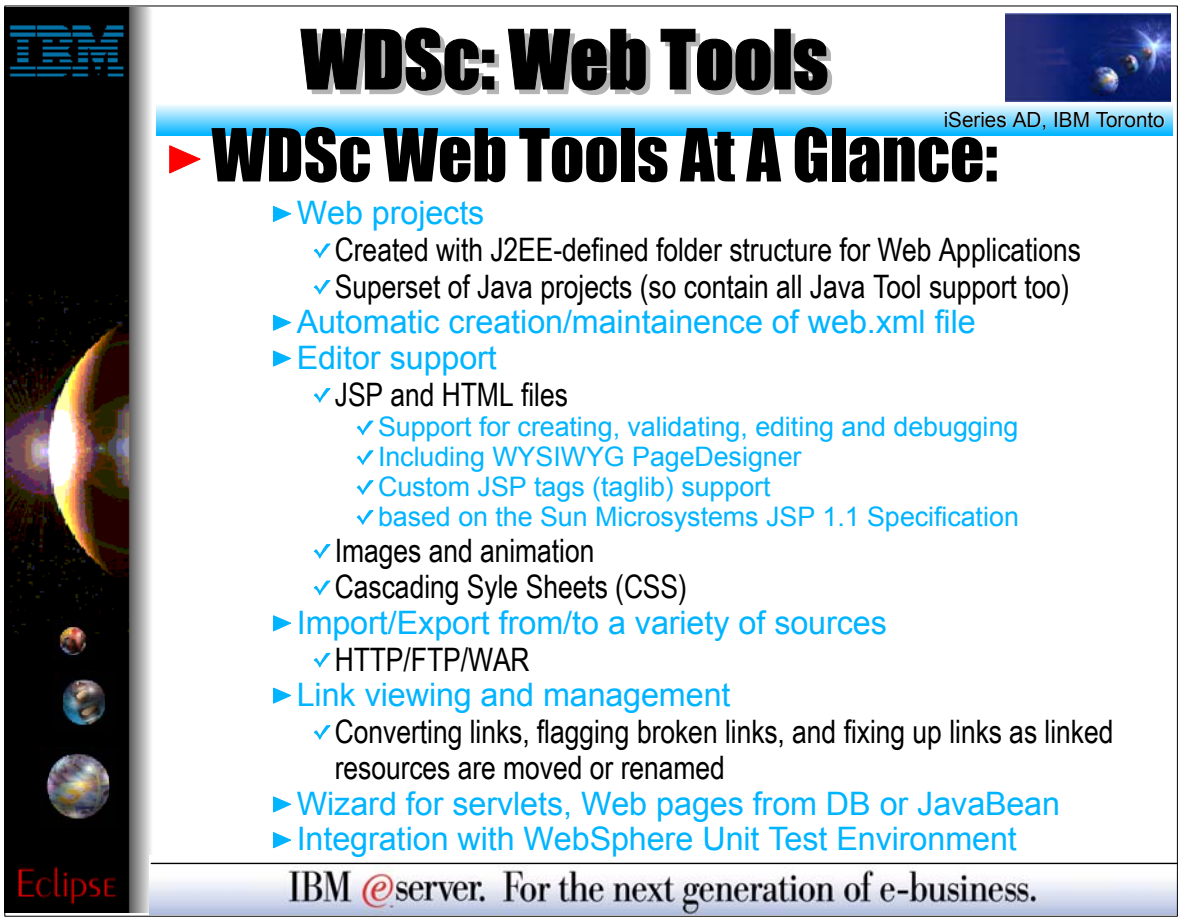


- In WebSphere Studio, the Web developer has a full suite of tools to support him
- When creating a Web project, it is created with the correct folder structure to comply to the J2EE standard
- Also a there are specific iSeries tools provided to help with creating the interface description of programs or service programs on the iSeries.
- The tools also allow to reference iSeries data base fields and to use these in the web


Web Projects
iSeries AD, IBM Toronto
Create a Web Project
Define the Web Project
Create a Web project and add it to a new or existing Enterprise Application project.
Project name: MyWebProject
Use default location
Location: E:\WDSC\WSSD\workspace\MyWebProject
Browse...
Enterprise Application project name: DefaultEAR
Context root: MyWebProject
Create CSS file
< Back
Next >
Automatically associates with a supplied EAR file for easy auto-cfg of Application Server
Optionally creates a CSS style for whole Web app
Create a Web Project
Module Dependencies
Select dependent JARs for the module within the Enterprise Application project.
Project name: MyWebProject
Enterprise Application project name: DefaultEAR
Available dependent JARs:
JAR/Module
Project
Manifest Class-Path:
Identify runtime dependent jar/war files
< Back
Next >
Finish
Cancel
Eclipse
IBM @server. For the next generation of e-business.

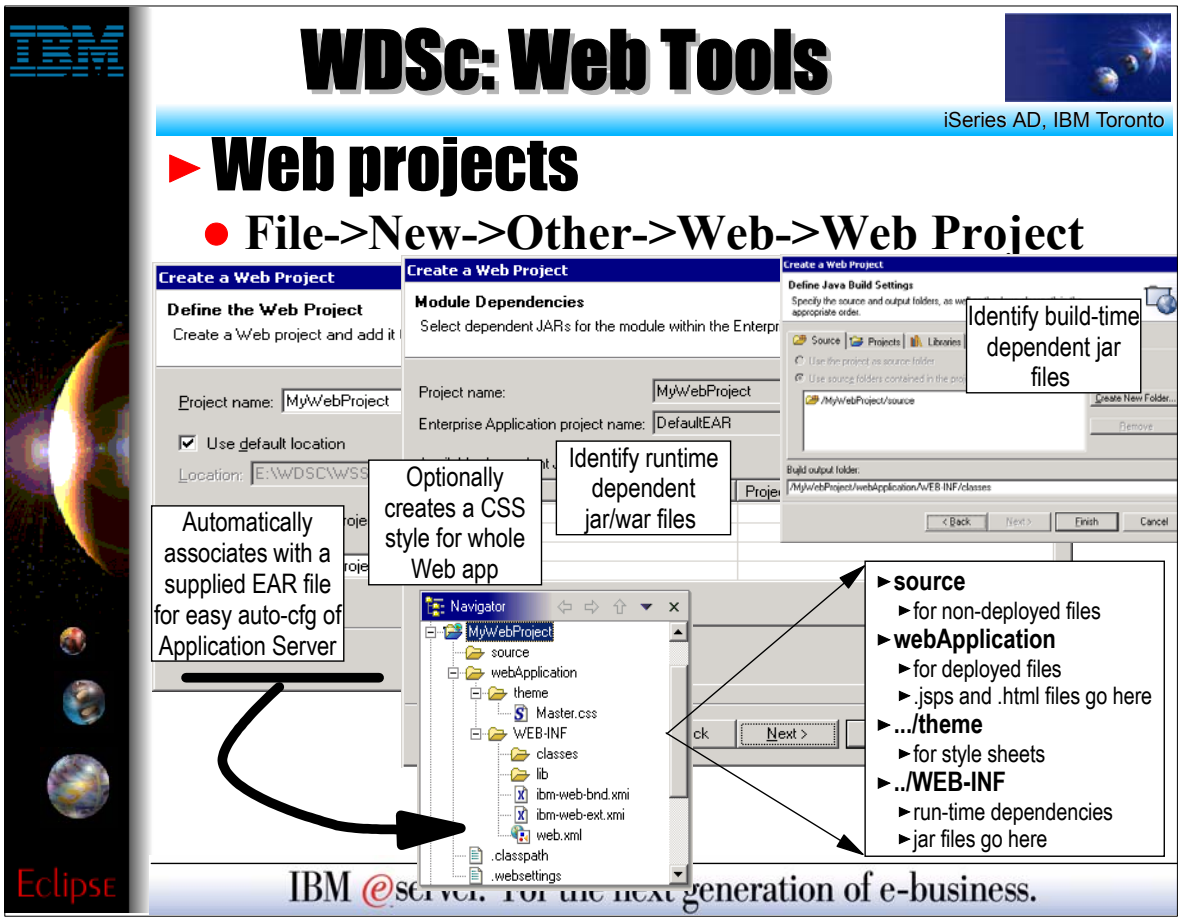
Web Projects



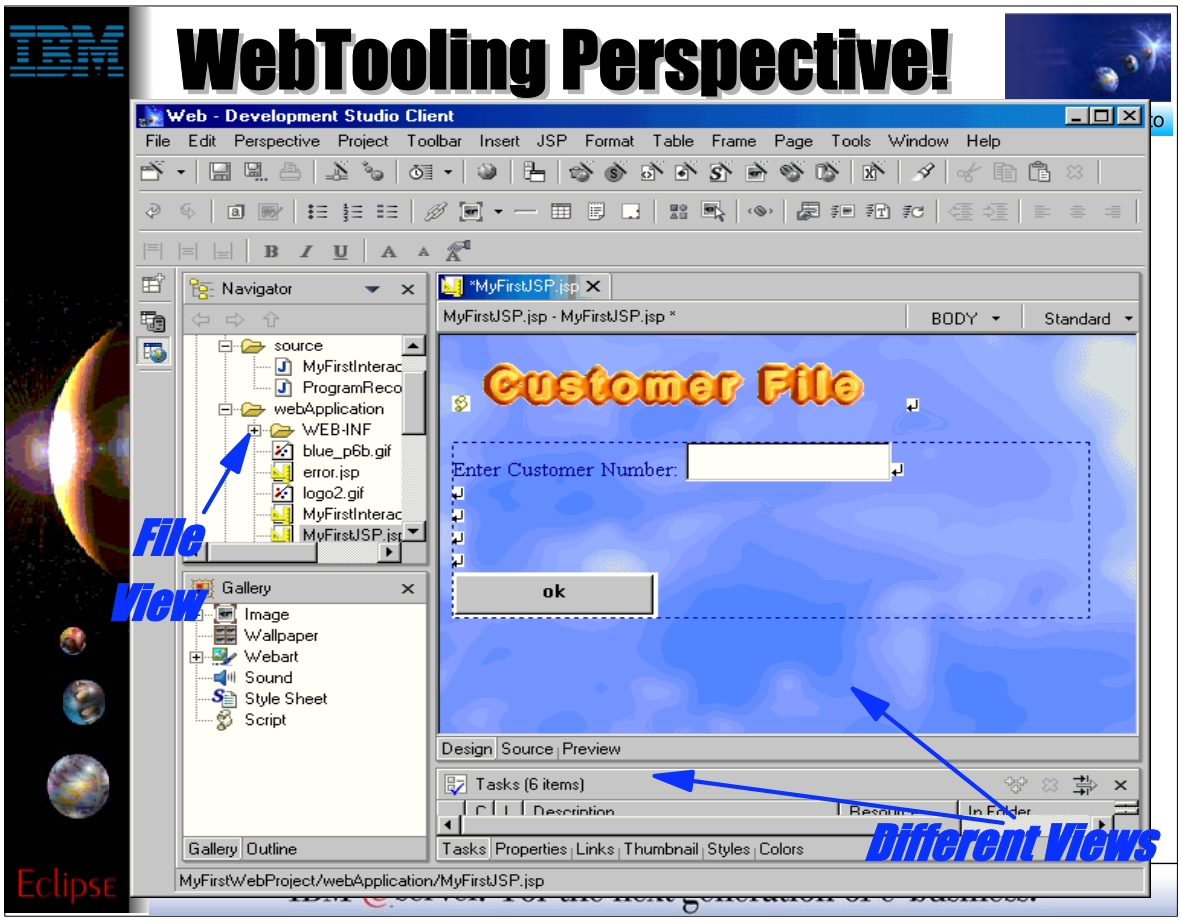

WDSc: Web Tools
iSeries AD, IBM Toronto
WDSc Web Tools At A Glance:
Web projects
Created with J2EE-defined folder structure for Web Applications
Superset of Java projects (so contain all Java Tool support too)
Automatic creation/maintainence of web.xml file
Editor support
JSP and HTML files
Support for 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 Syle Sheets (CSS)
Import/Export from/to a variety of sources
HTTP/FTP/WAR
Link viewing and management
Converting links, flagging broken links, and fixing up links as linked resources are moved or renamed
Wizard for servlets, Web pages from DB or JavaBean
Integration with WebSphere Unit Test Environment
Eclipse
IBM @server. For the next generation of e-business.

Web tooling Components

- This slide shows the different components that support the creation of Web Applications on an iSeries server.
- The workbench integrates all the different tools
- Page designer allows WYSIWYG user interface composition for both html or jsp definitions
- Wizards allow you to get started fast on your Web project
- Specific iSeries tools help you to exploit the content of your iSeries server



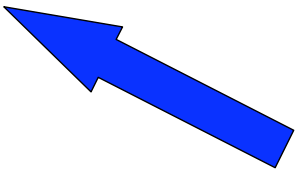
- Here is a sample of the Web project wizard
- It will guide the user through specifying the information necessary to create a Web project, based on this information a Web project infrastructure will be created.
- This base Web project can then be filled with content
- It can be tested directly in the Studio test environment or can be deployed in EAR files to remote servers.

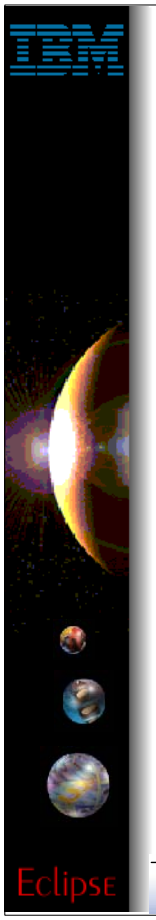


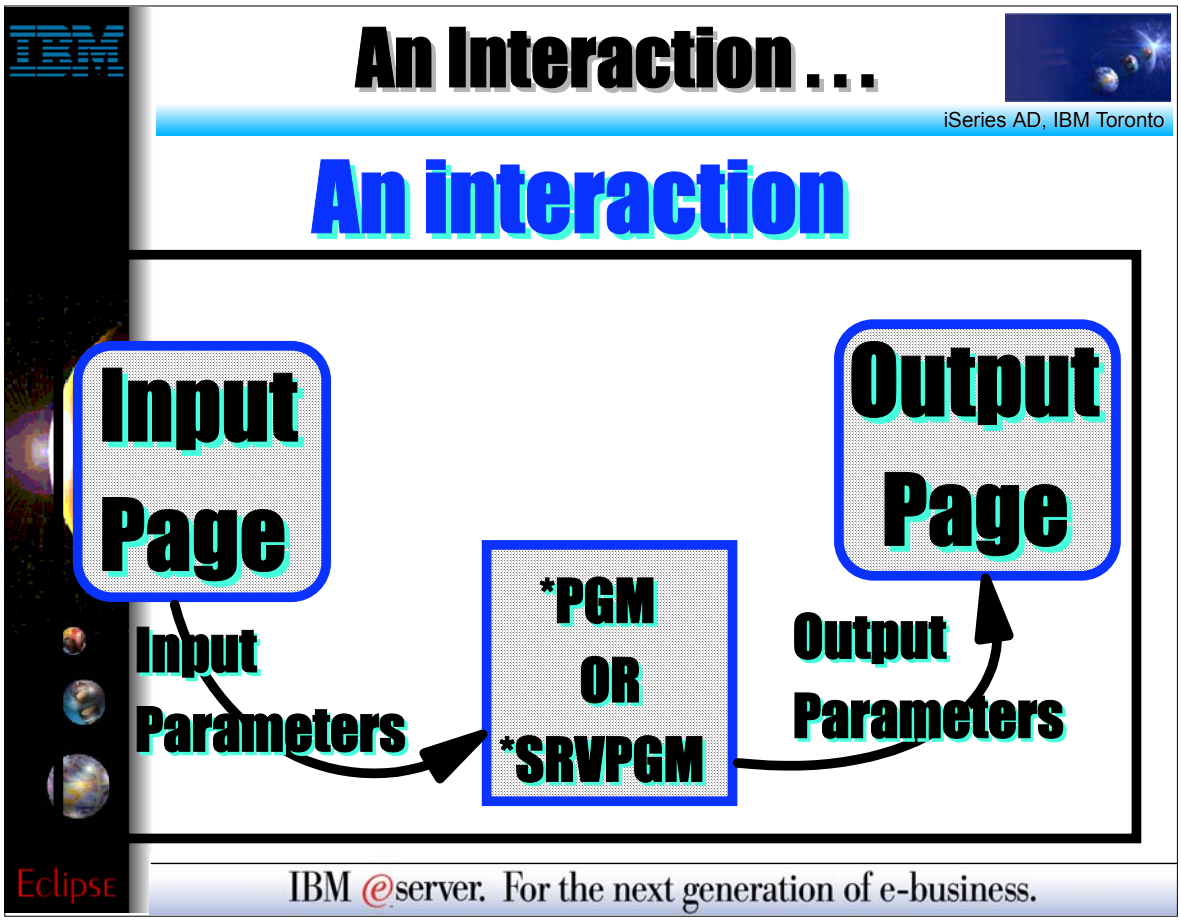
- Here is a sample of the Web Perspective which contains the tools a Web developer needs. In this example, the Page Designer tool , file view, and a view of multi media files shows.
- Perspective's can be tailored to individual needs.

AGENDA

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



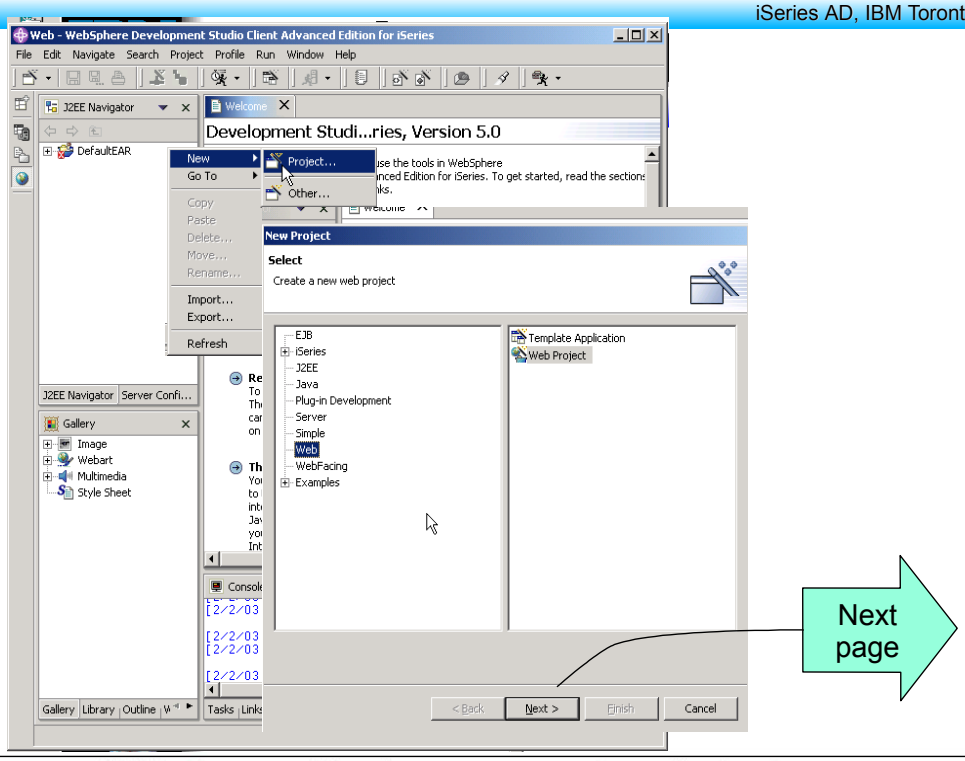




- An Interaction is defined as an Input Page which allows data input, this data is submitted to a server and processing of the request is done on the server. The server will then respond with serving a result page. The content of the result page is based on the data on the input page
- The combination of the Input Page and the Output page with the Server in the middle handling the dynamic requests is called an Interaction.

Create a Web project

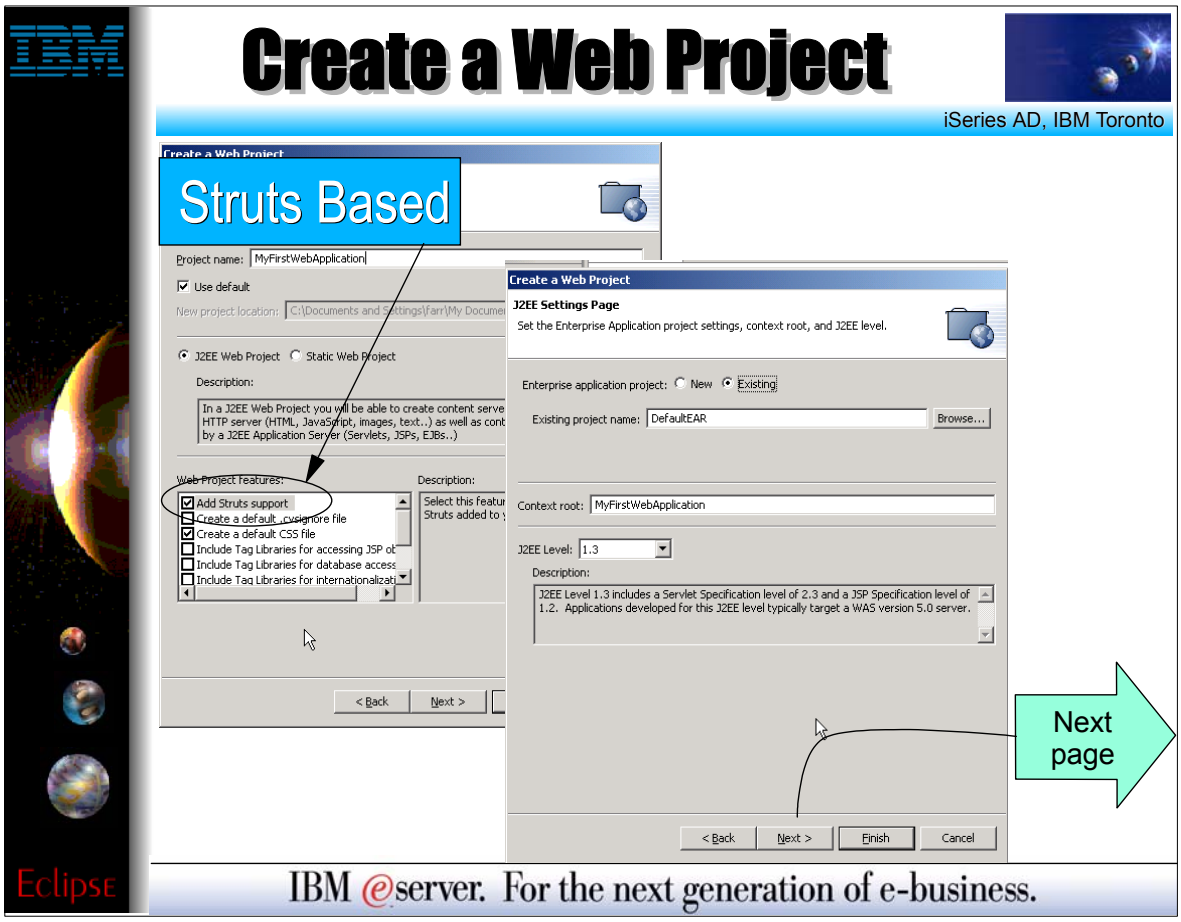
iSeries AD, IBM Toronto

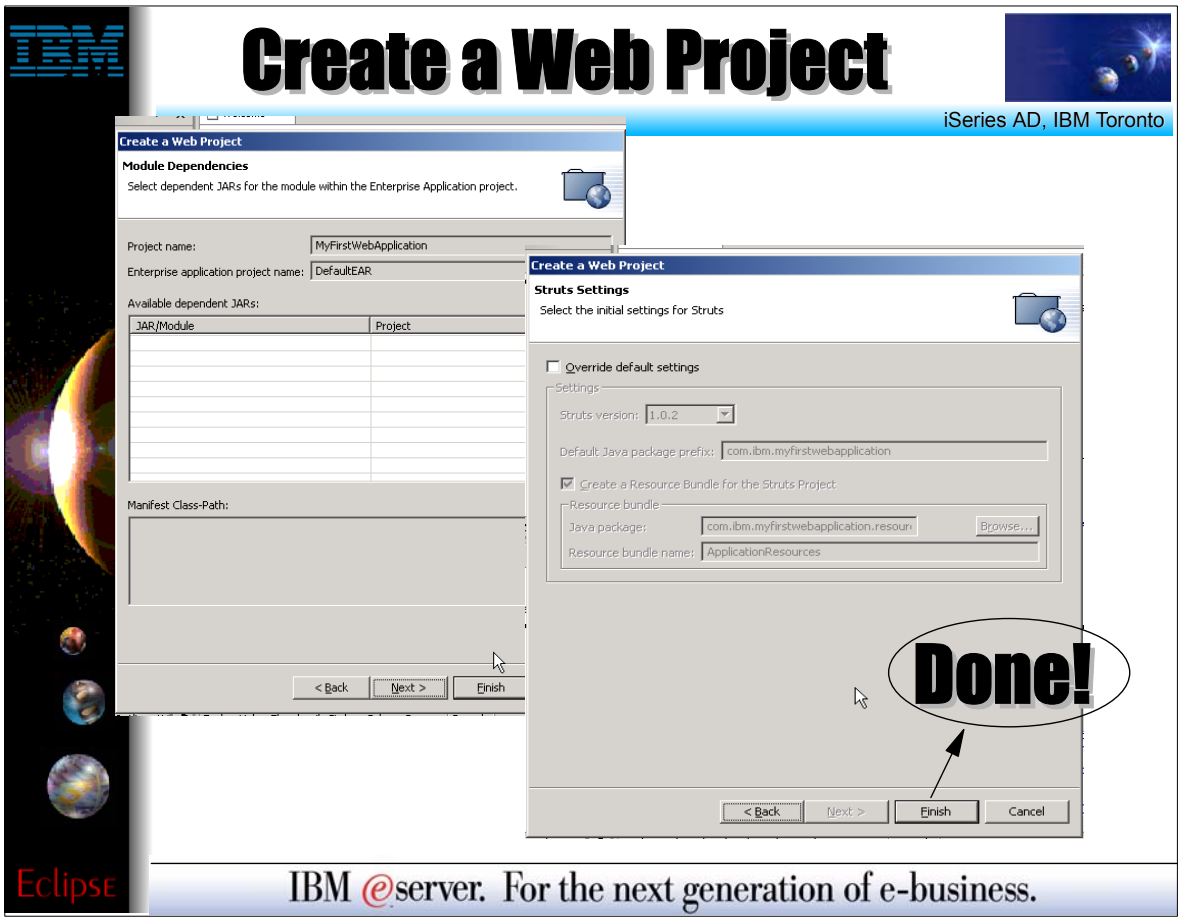


Eclipse

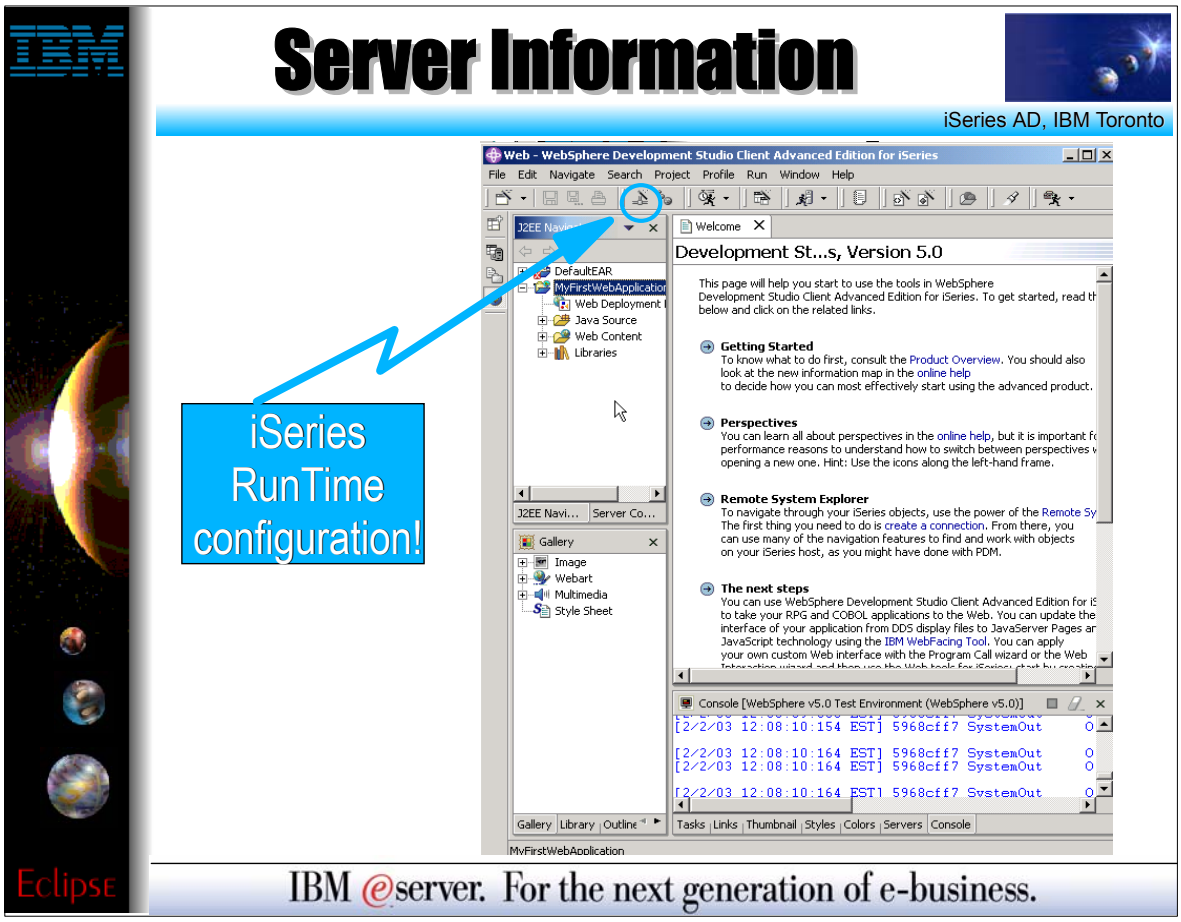
IBM @server. For the next generation of e-business.

Create a Web Project

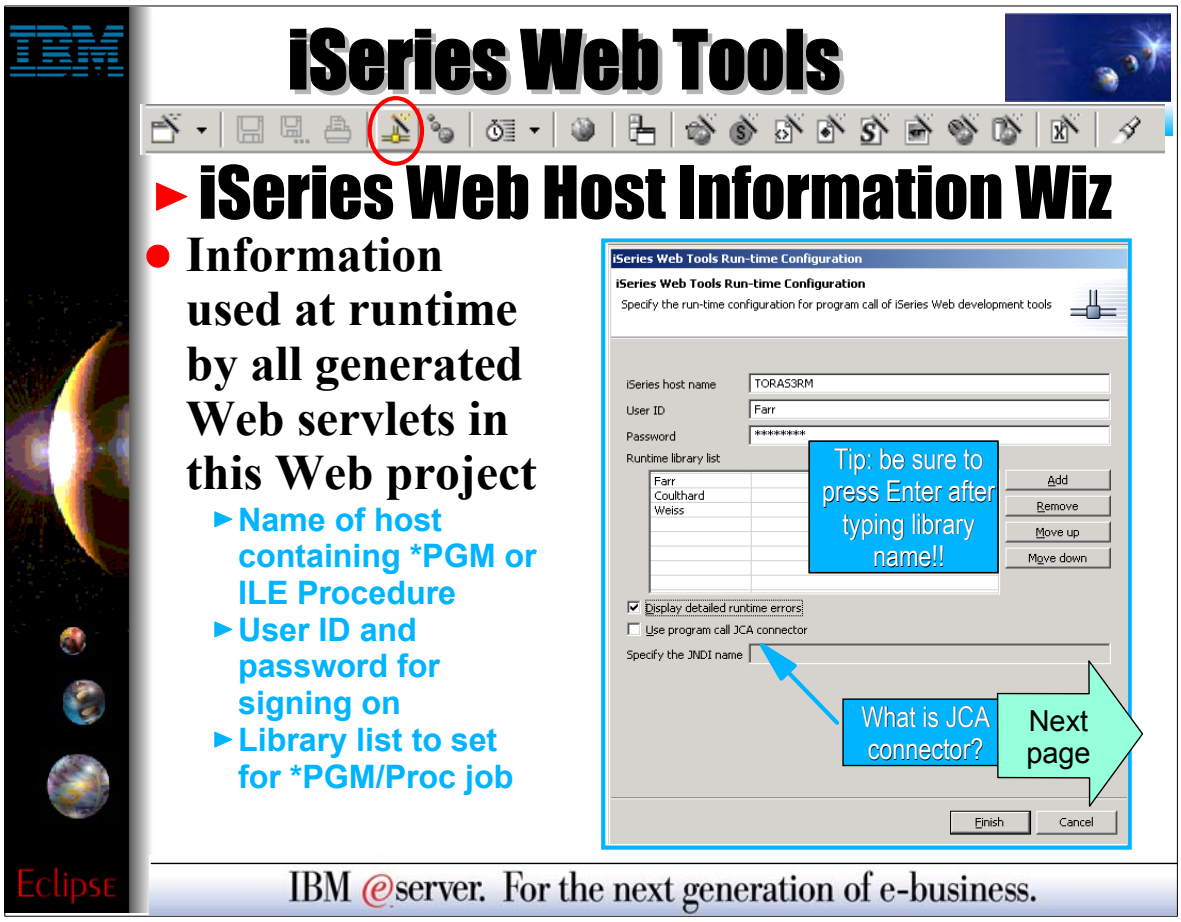




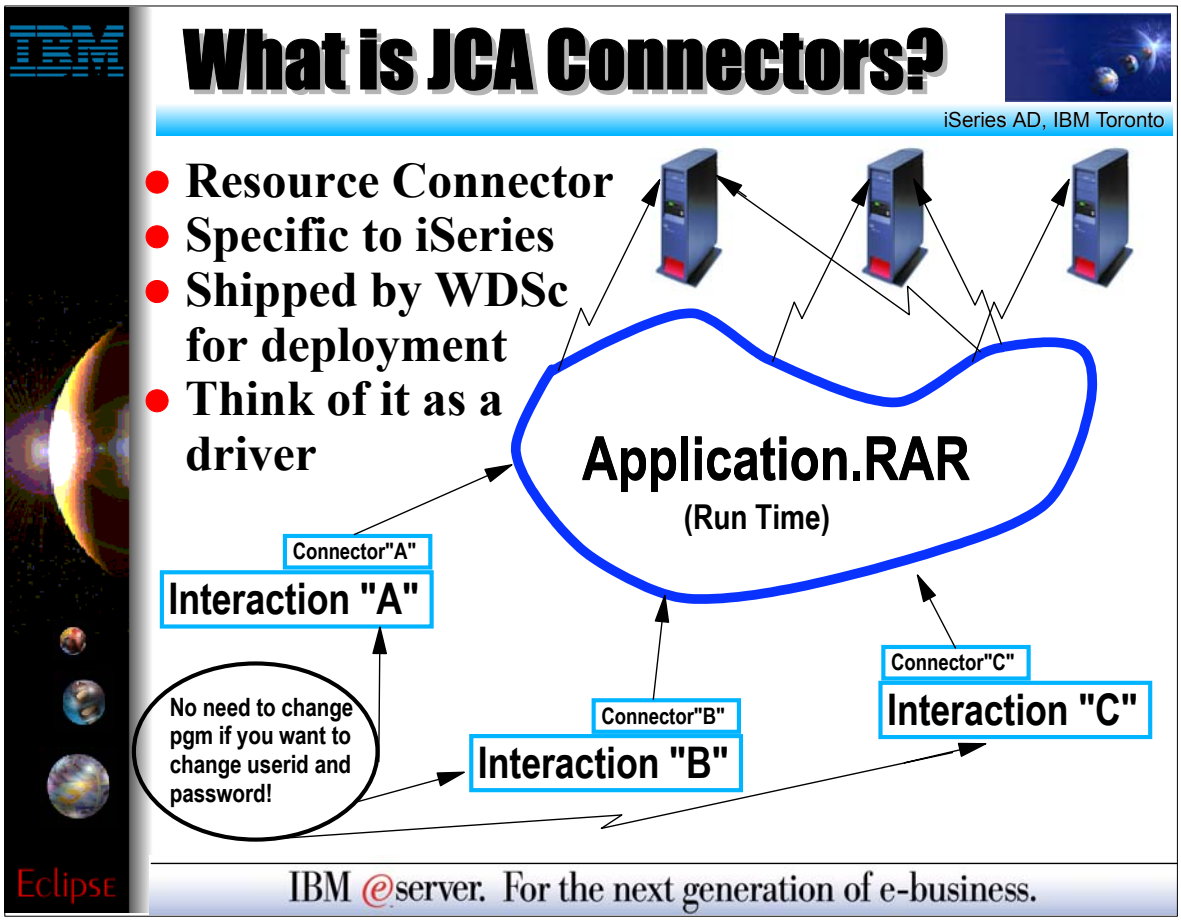
- You can create a Web application based on a Struts infrastructure



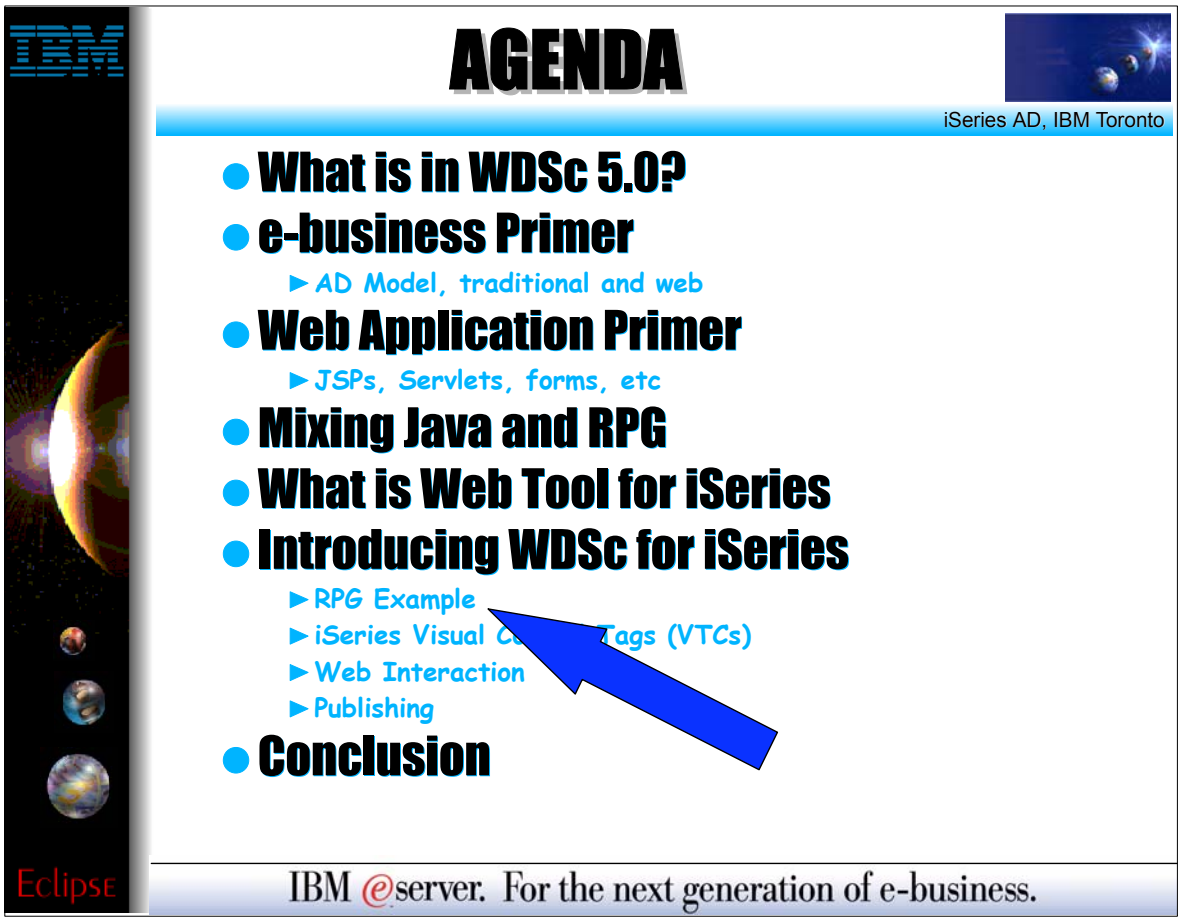
- Here we specify what server our Program or Server Program resides on
- Which Userid / Password we want to use to start a job on our server
- Also this dialog allows us to add specific libraries to the job environment if this is needed for the Interaction



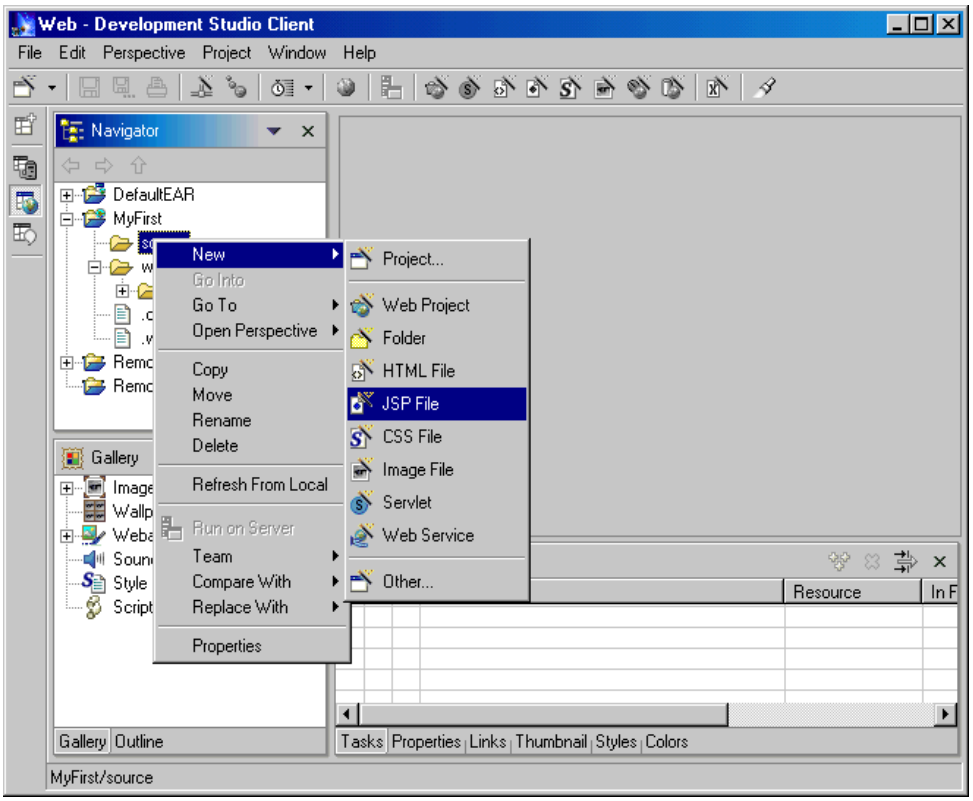
- The iSeries Web Tools Run-time Configuration wizard captures information such as server name, userid, password and library list used by Web applications



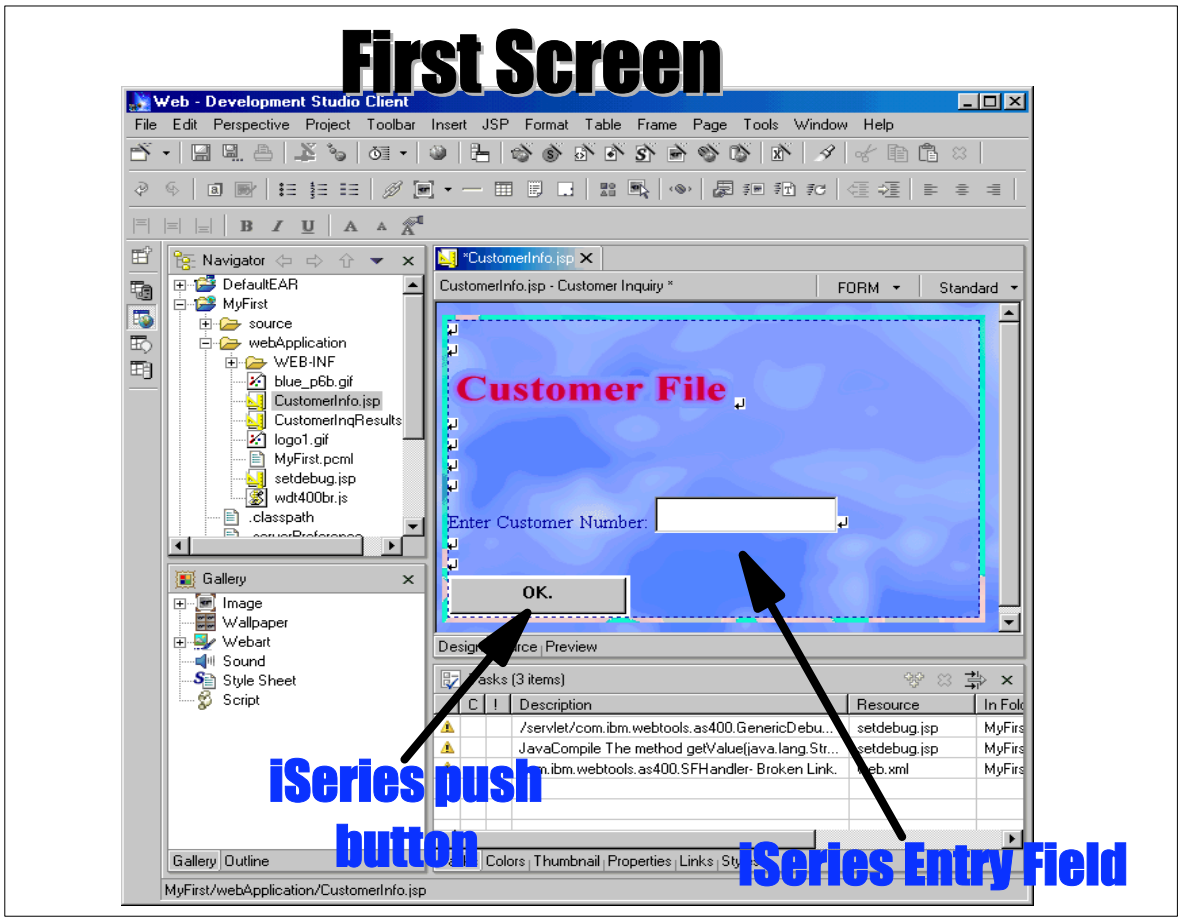
- JCA Program Call connector provides a standard interface to call any *PGM, based on the J2EE Connector Architecture


AGENDA
iSeries AD, IBM Toronto
What is in WDSc 5.0?
e-business Primer
AD Model, traditional and web
Web Application Primer
JSPs, Servlets, forms, etc
Mixing Java and RPG
What is Web Tool for iSeries
Introducing WDSc for iSeries
RPG Example
iSeries Visual
Tags (VTCs)
Web Interaction
Publishing
Conclusion
Eclipse
IBM @server. For the next generation of e-business.

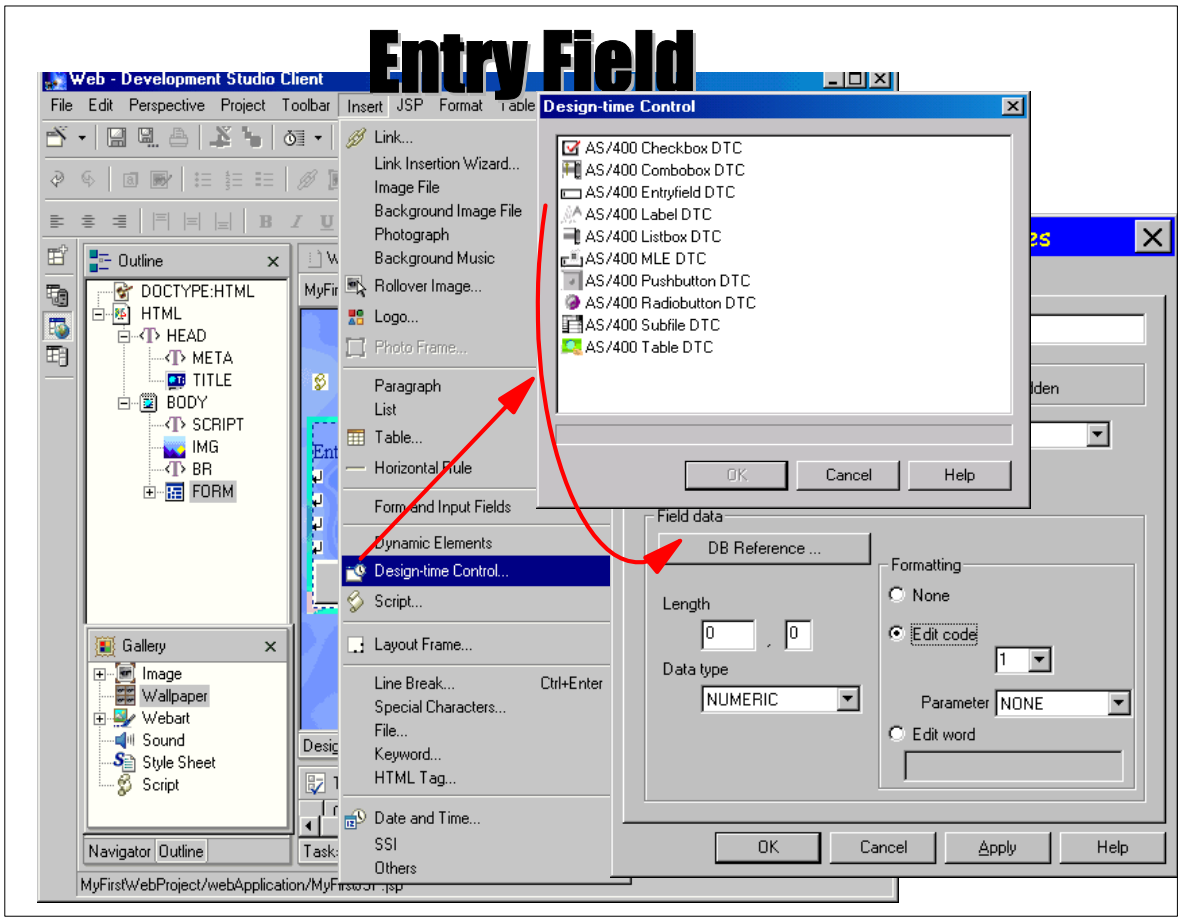
Your first JSP …



- First we want to build a user interface for a simple input page
- We select to create a new jsp file. This will automatically invoke the Page Designer

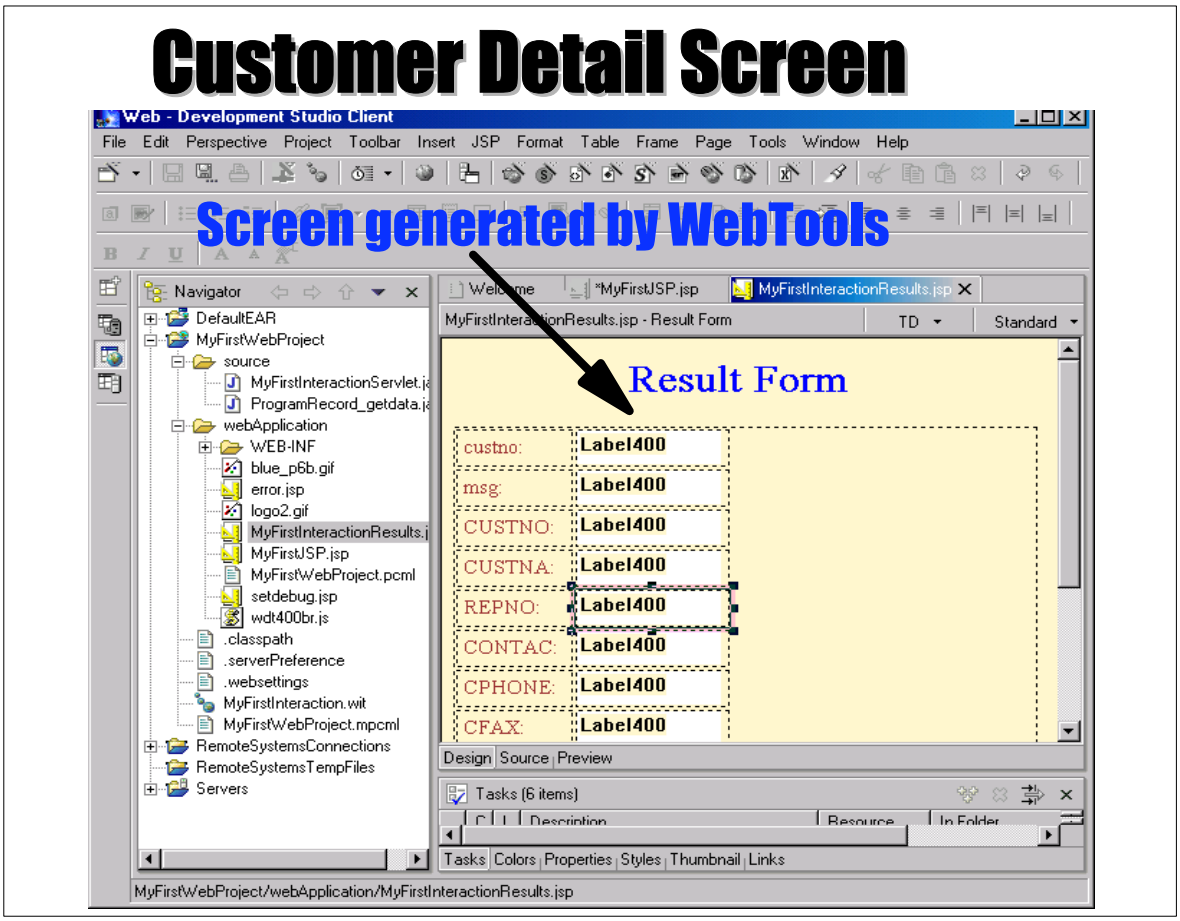


- This slide shows the Page Designer tool, which can be used to enhance the user interface.
- Putting a nice background to the page and changing some of the fonts will increase the look of the Web page.
- We also add a form tag to our page and an iSeries entry field and an iSeries push button.
- These iSeries controls are specific controls that inherit some of the capabilities of 5250 parts that normal html controls don't provide.
- For example the capability of referencing iSeries data base fields and the capability of specifying edit codes.



- Here is a list of all the iSeries special controls provided in the WDSC tools. Notice that even a subfile has been added to the list.

Customer Detail Screen



- We just looked at the input screen for our little application.
- This slide shows the Output or result screen.
- It uses reference fields from the iSeries database.
- The application will send data from the entry field on the input page to an RPG program, which will extract data from the data base and pass this data back
- The resulting data will be displayed on this page

RPG IV Program

NOTE Parameters to be passed

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

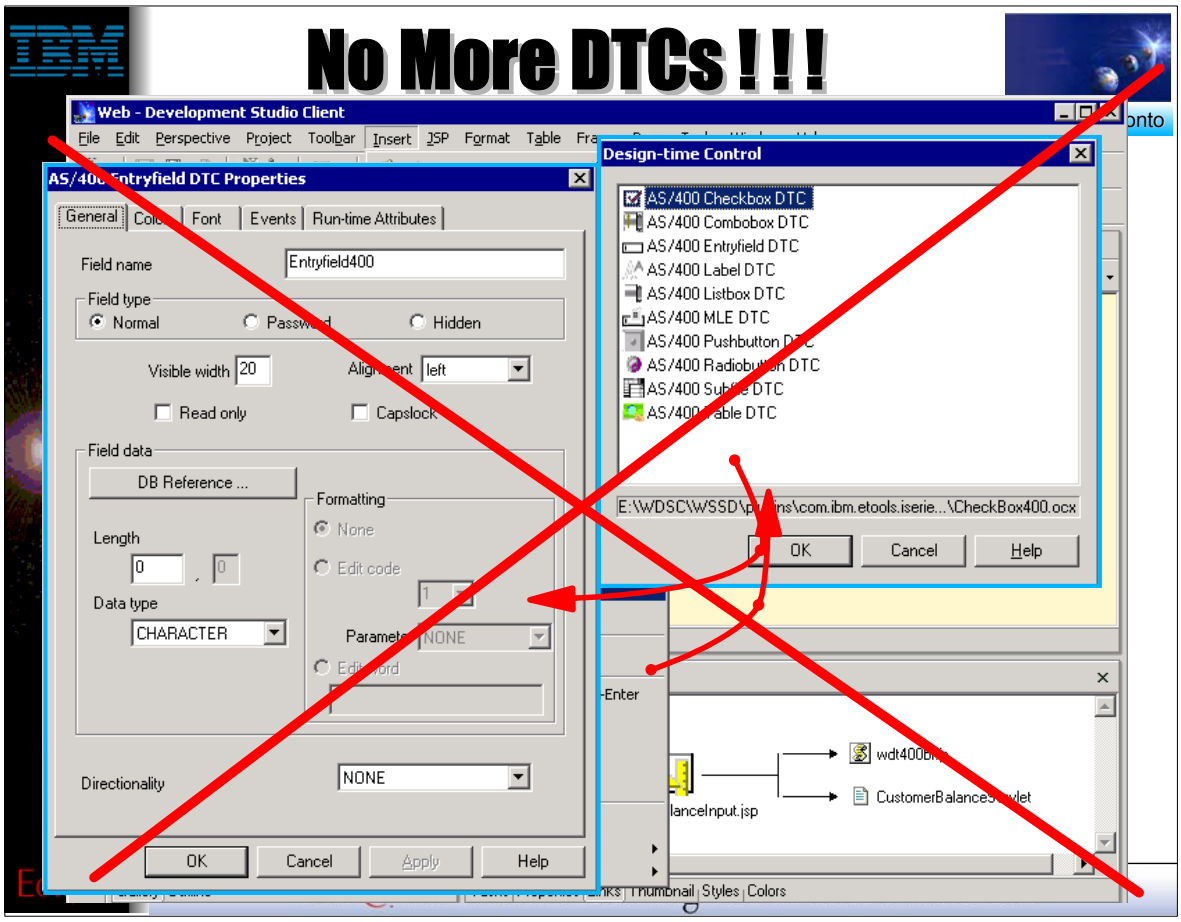
- This slide shows the RPG program that gets invoked
- A customer number is used to specify which customer data should be displayed. The first parameter custnoi will contain the customer number.
- The data structure CSTRUC will contain the full customer data passed back to the Servlet running in the Application server which will invoke the result jsp to show the data
- The parameter return, contains information whether the access to the data base was successful or not

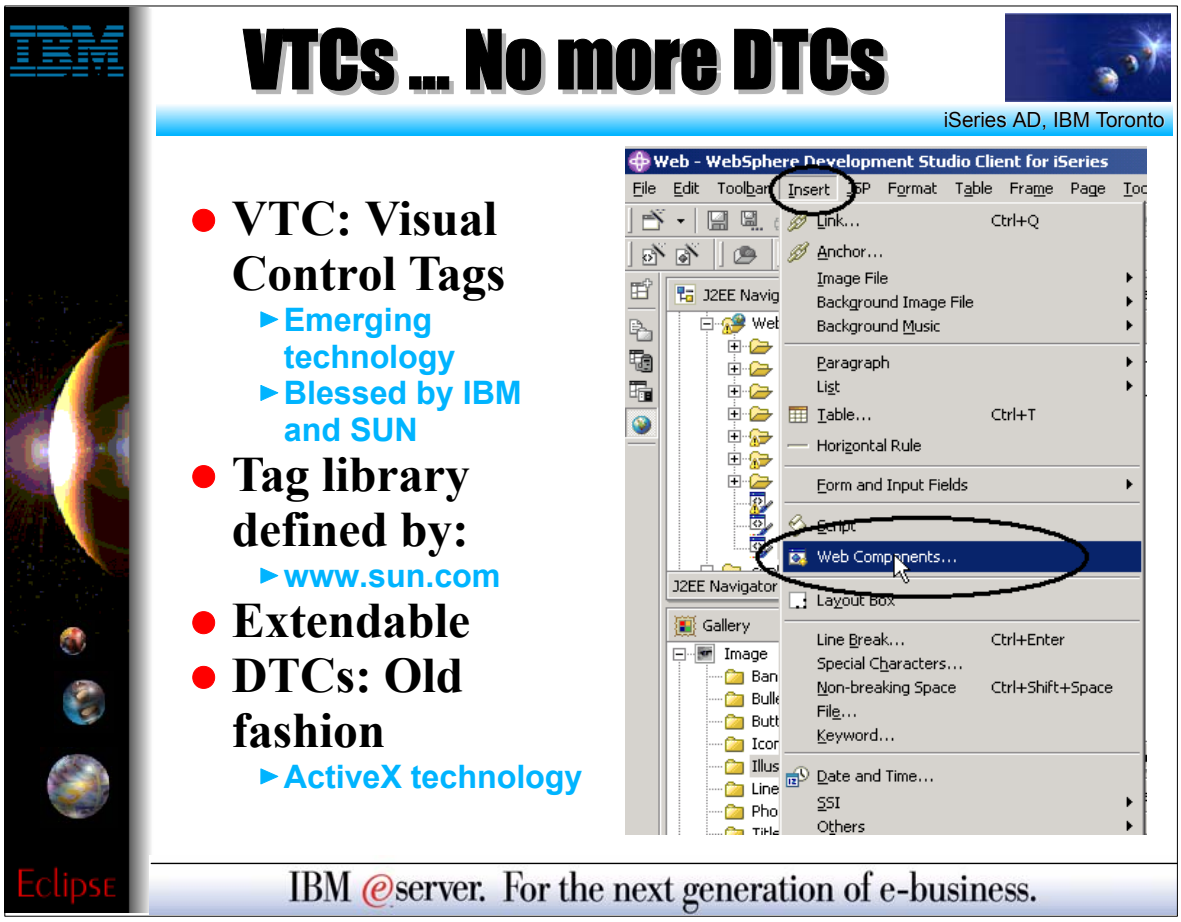
Files Used

```
A* Logical file description
A          R CUSTOM01                  PFILE(CUSTOMER)
A          K CUSTNO
```

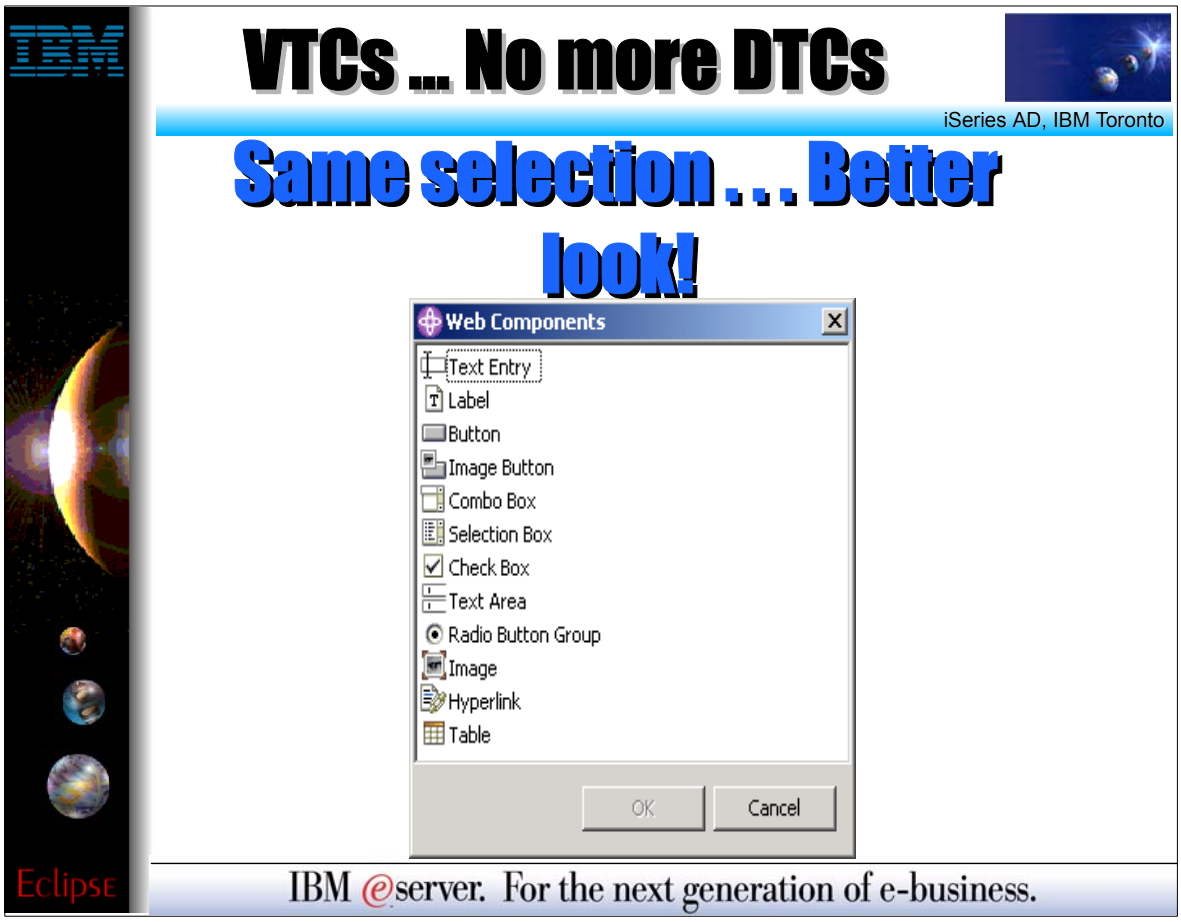
```
A* Physical file description - CUSTOMER FILE
A      R CUSTOM01
A        CUSTNO         7            COLHDG('Customer number')
A        CUSTNA        40            COLHDG('Company name')
A        REPNO          5            COLHDG('Rep identifier')
A        CONTAC        30            COLHDG('Name')
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
```

- Here is a description of the files used in this sample application.


No More DTCs ! ! !
Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table
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
AS/400 Table DTC
E:\WDSC\WSSD\plugins\com.ibm.etools.iserie...\CheckBox400.ocx
OK
Cancel
Help
Entryfield DTC Properties
General
Font
Events
Run-time Attributes
Field name
Entryfield400
Field type
Normal
Hidden
Visible width
20
left
Read only
Capslock
Field data
DB Reference ...
Formatting
None
Edit code
Length
Data type
CHARACTER
NONE
Directionality
NONE
OK
Cancel
Apply
Help
Thumbnail
Styles
Colors



- VCTs replace the ActiveX Design-Time Controls (DTCs). VCTs can be used and rendered in the HTML Designer.

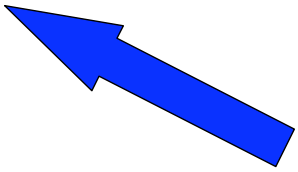


- VCTs provide the controls such as textField, label, dropDown, pushButton, and so on.

AGENDA

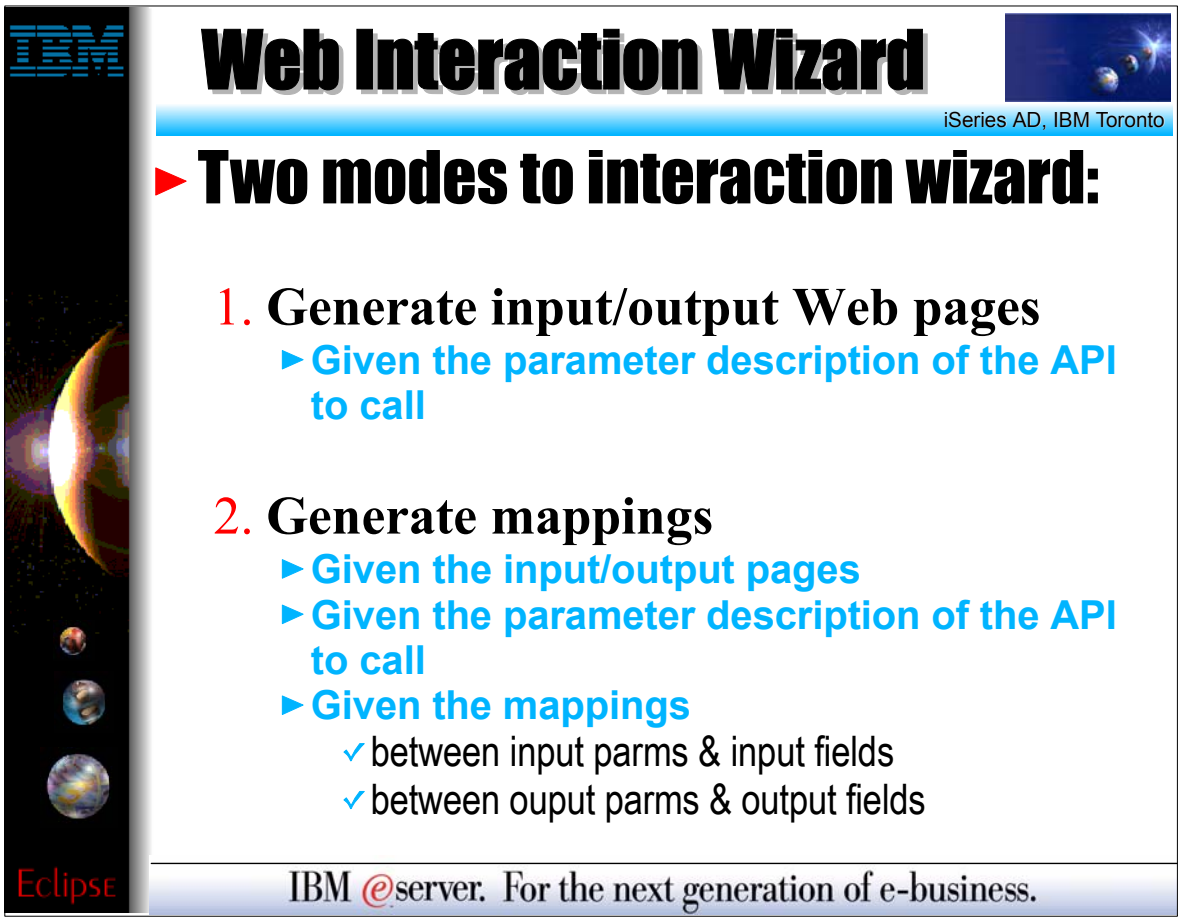
iSeries AD, IBM Toronto

- What is in WDSc 5.0?
- e-business Primer
  - AD Model, traditional and web
- Web Application Primer
  - JSPs, Servlets, forms, etc
- Mixing Java and RPG
- What is Web Tool for iSeries
- Introducing WDSc for iSeries
  - RPG Example
  - iSeries Visual Control Tags (VTCs)
  - Web Interaction
  - Publishing
- Conclusion



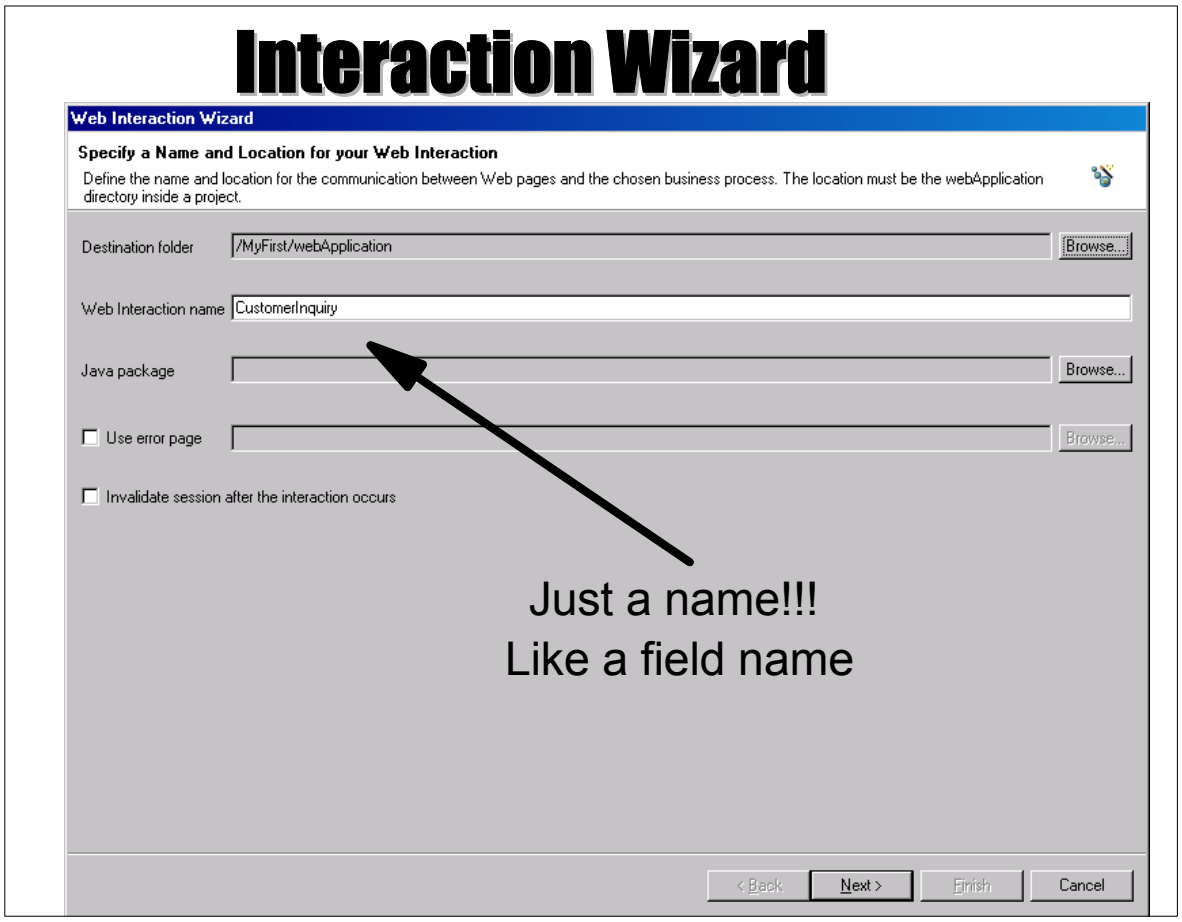
Eclipse

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



- This slide shows that we have a choice of having the Interaction Wizard create our User Interface or design it ourselves.
- In this example we decide to design the page ourselves. use Page Designer to add a background and heading.
- We add a form tag to the Input page. The form contains an entry field for data entry and a submit push button to send a request with the data in the entry field to a server.

Interaction Wizard



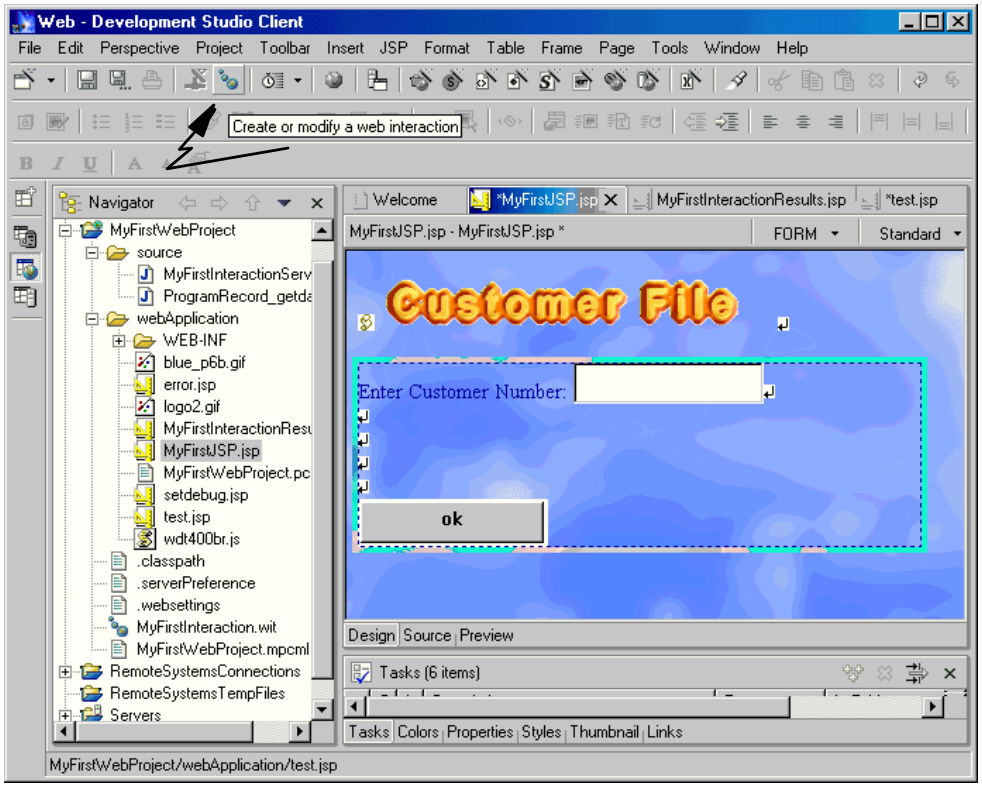
- After finishing our Input page, we are ready to start the Interaction wizard
- This slide shows that we need to specify the Interaction name and go on to the next page.

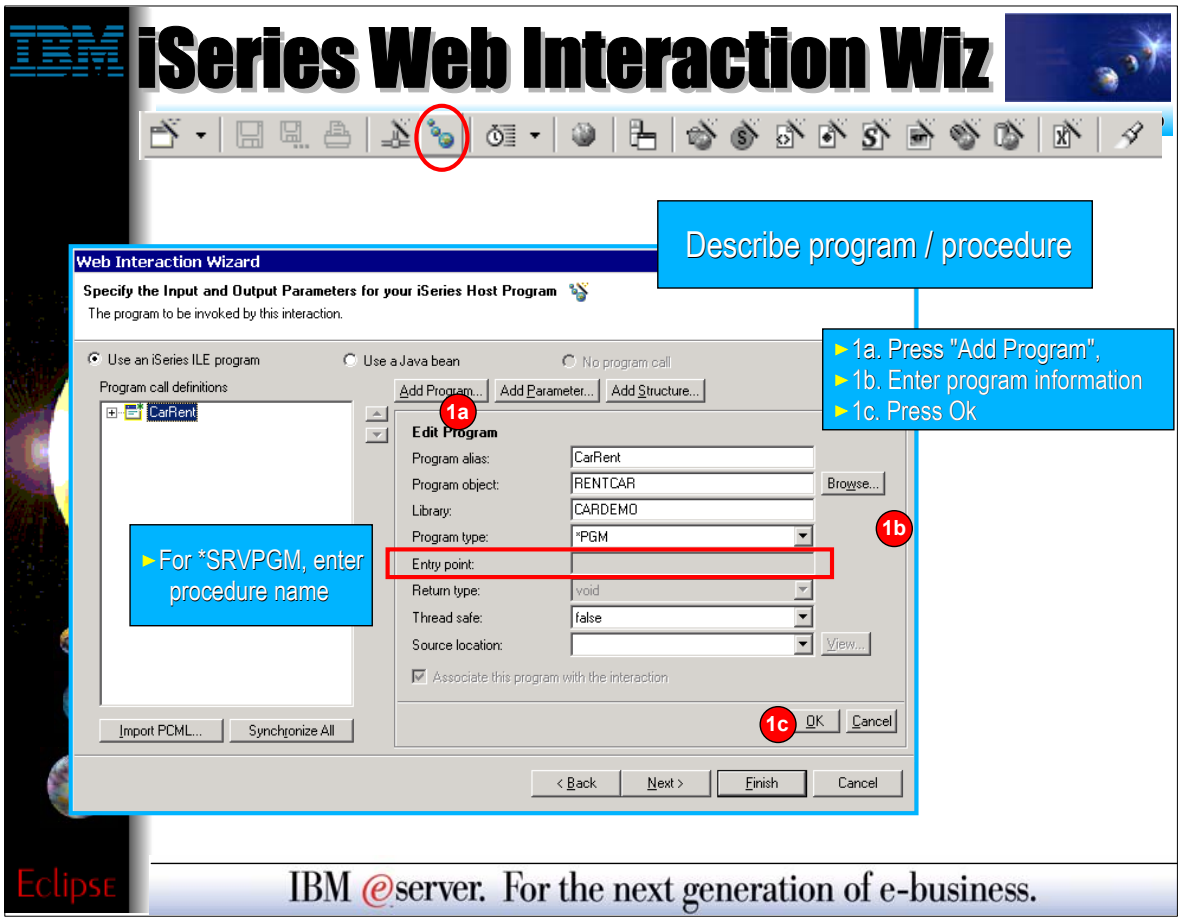
WDSc: iSeries Web Tools

iSeries AD, IBM Toronto

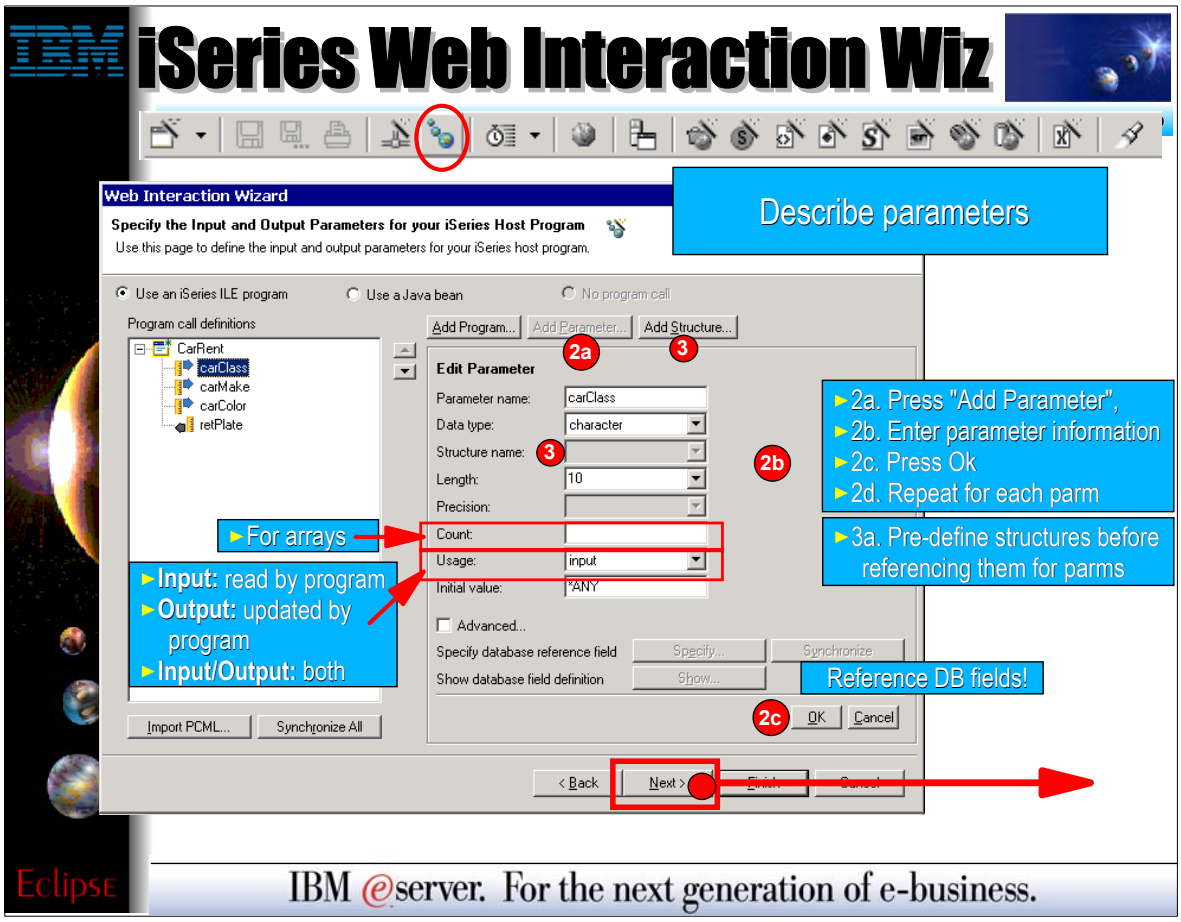
iSeries Web Interaction Wizard

Or ... Create your own!

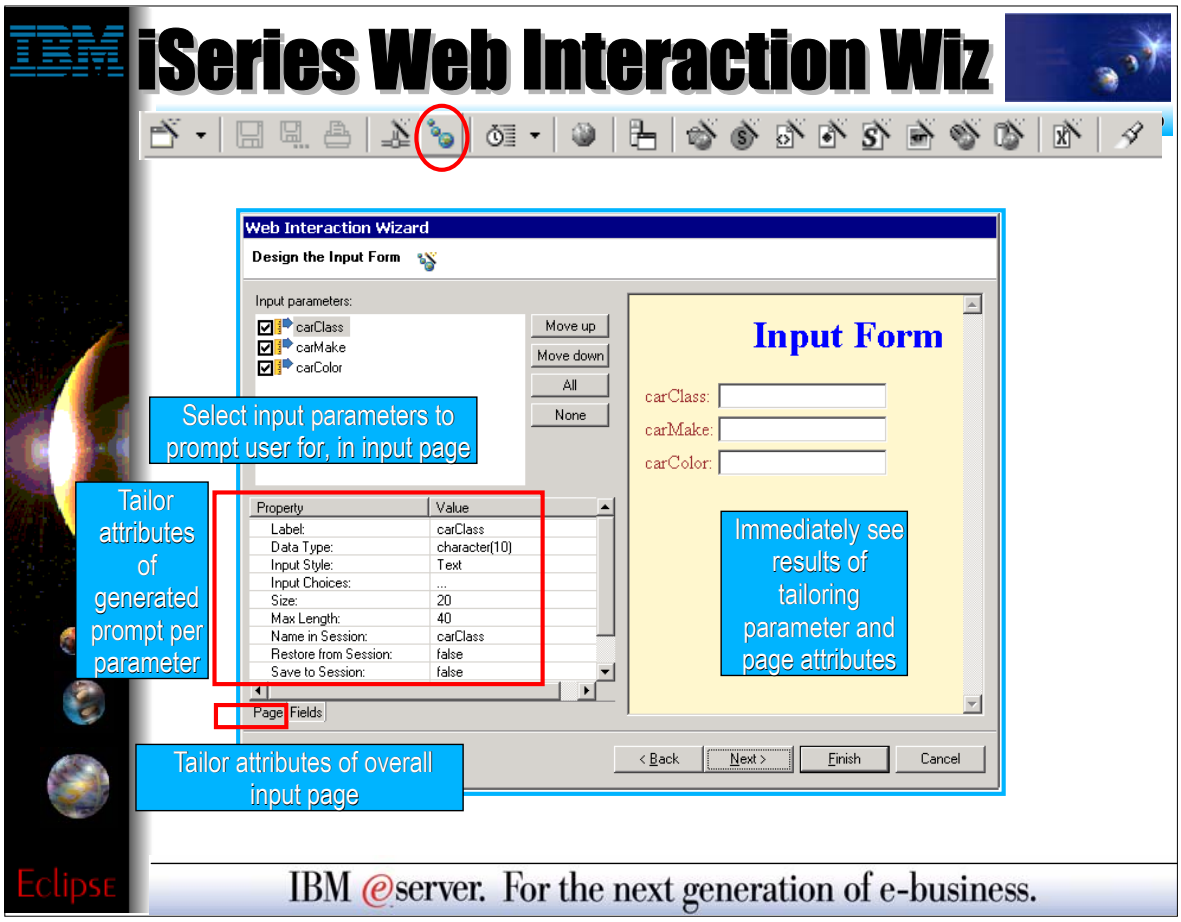




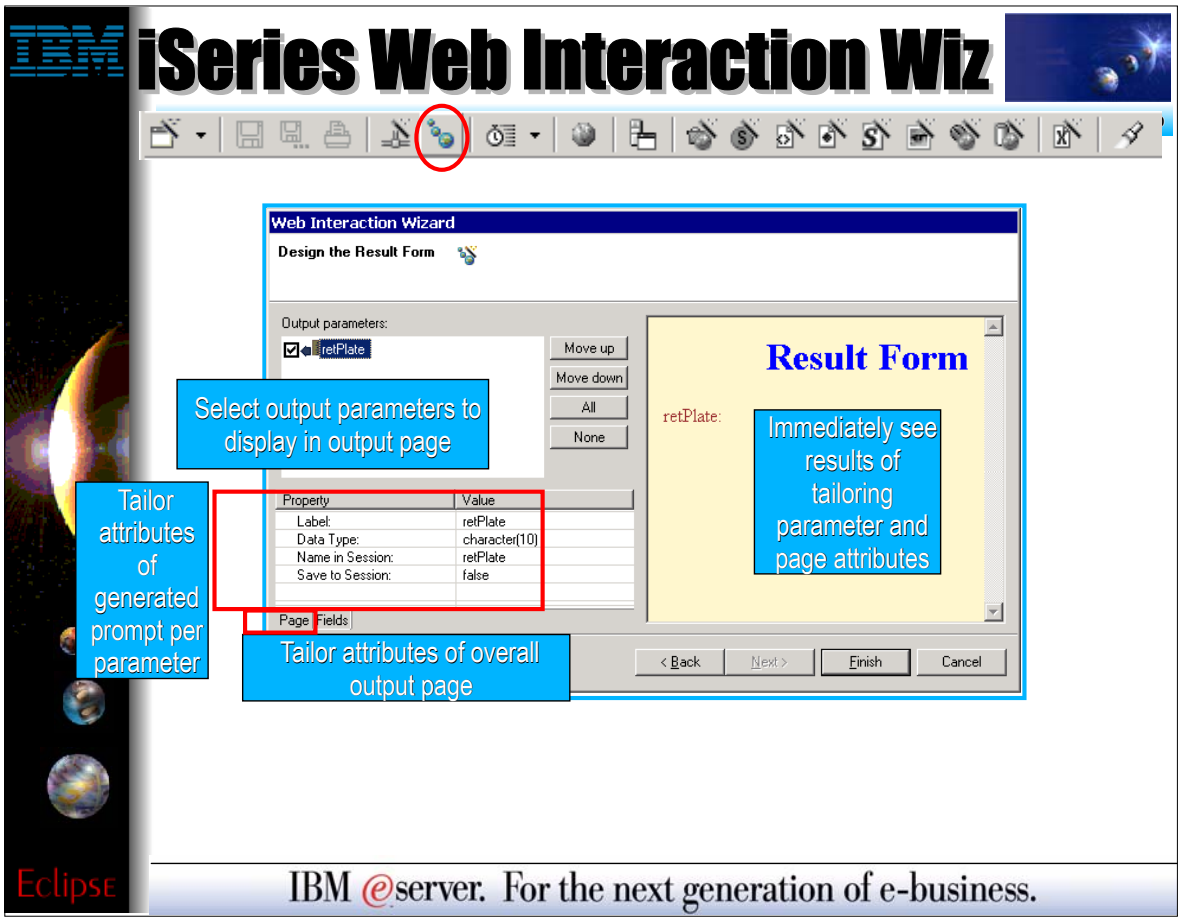
- Shows the next page of the wizard. On this page we need to specify the interface to our program on the iSeries
- The wizard will generate PCML out of this information
- We need to specify the program to call, with information which library the program resides in
- We could also specify to call a procedure in a Service program.



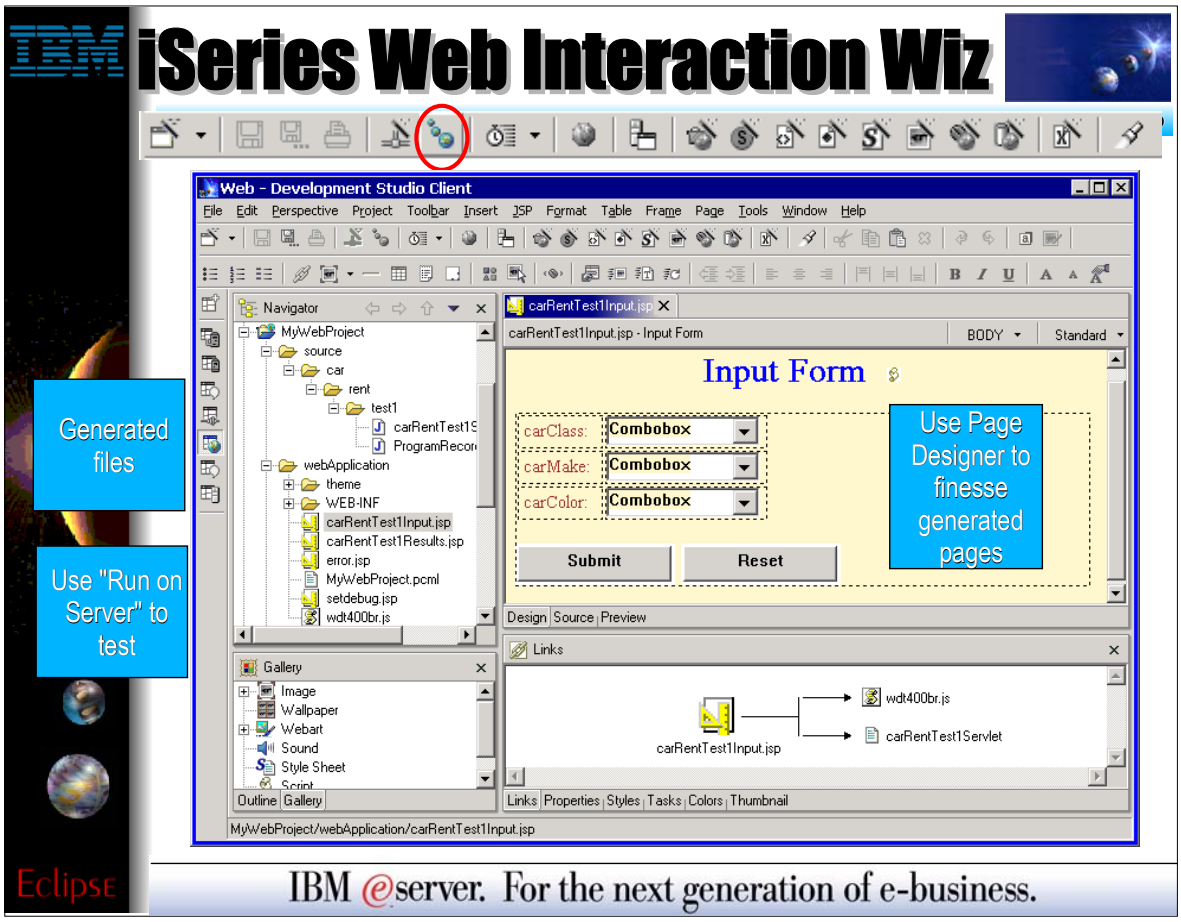
- Then we need to specify the parameters expected by the program
- If the parameters are data structures we can define these here as well
- All parameters can be defined by using reference fields



- This page allows you to specify which entry field on the Input page contains data for which parameter
- Entry fields and parameters with the same name are linked together



- Now the result page has to be defined.
- Automatically all output parameters are shown in the result page, if the page is generated by the wizard
- To deselect fields from being included on the result page, just uncheck them in the list on the left side of the dialog

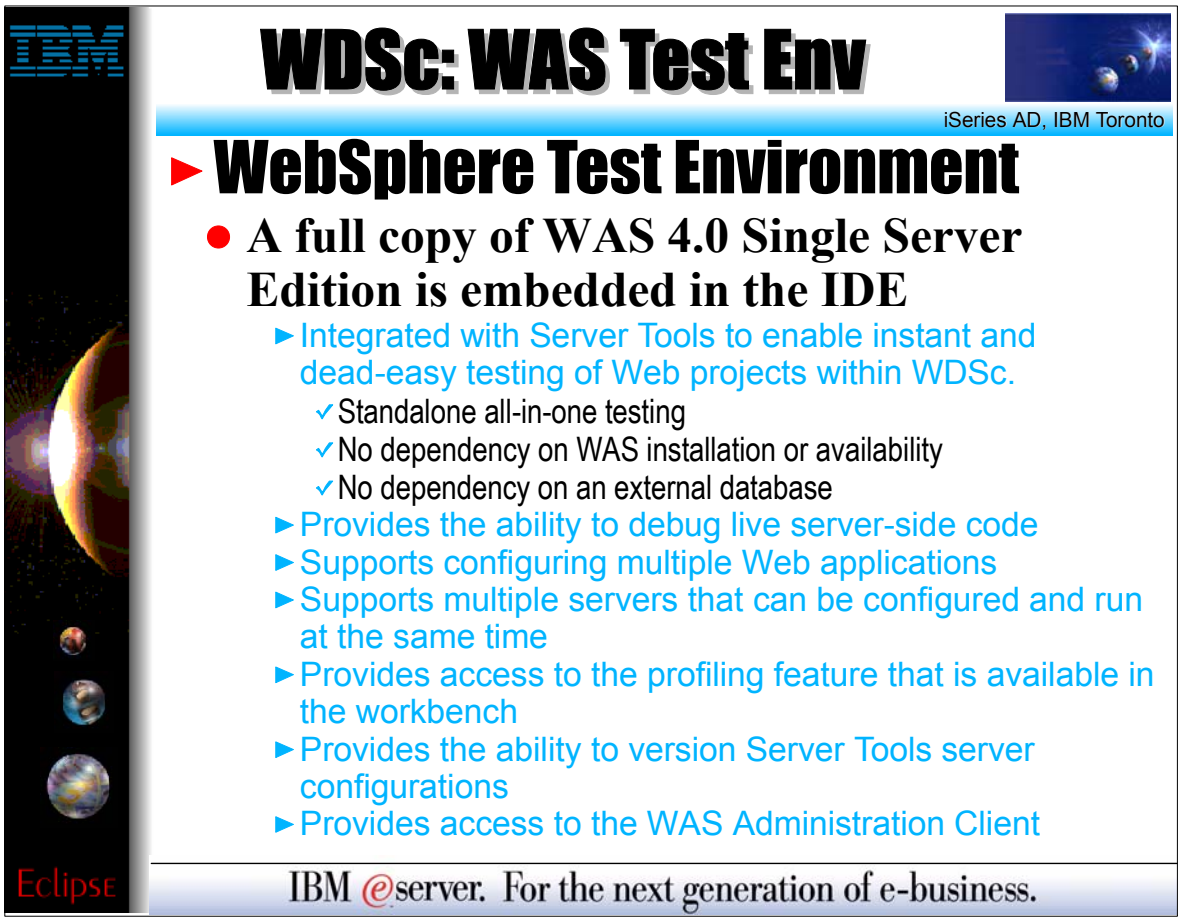

iSeries Web Interaction Wiz
Web - Development Studio Client
File Edit Perspective Project Toolbar Insert JSP Format Table Frame Page Tools Window Help
Navigator
MyWebProject
source
car
rent
test1
webApplication
theme
WEB-INF
carRentTest1Input.jsp
carRentTest1Results.jsp
error.jsp
MyWebProject.pcml
setdebug.jsp
wdt400br.js
Gallery
Image
Wallpaper
Webart
Sound
Style Sheet
Outline Gallery
carRentTest1Input.jsp
carRentTest1Input.jsp - Input Form
BODY
Standard
Input Form
carClass: Combobox
carMake: Combobox
carColor: Combobox
Submit
Reset
Design Source Preview
Links
wdt400br.js
carRentTest1Input.jsp
carRentTest1Servlet
Links Properties Styles Tasks Colors Thumbnail
MyWebProject/webApplication/carRentTest1Input.jsp
Generated files
Use "Run on Server" to test
Use Page Designer to finesse generated pages
Eclipse
IBM @server. For the next generation of e-business.

You are done! What's Next?

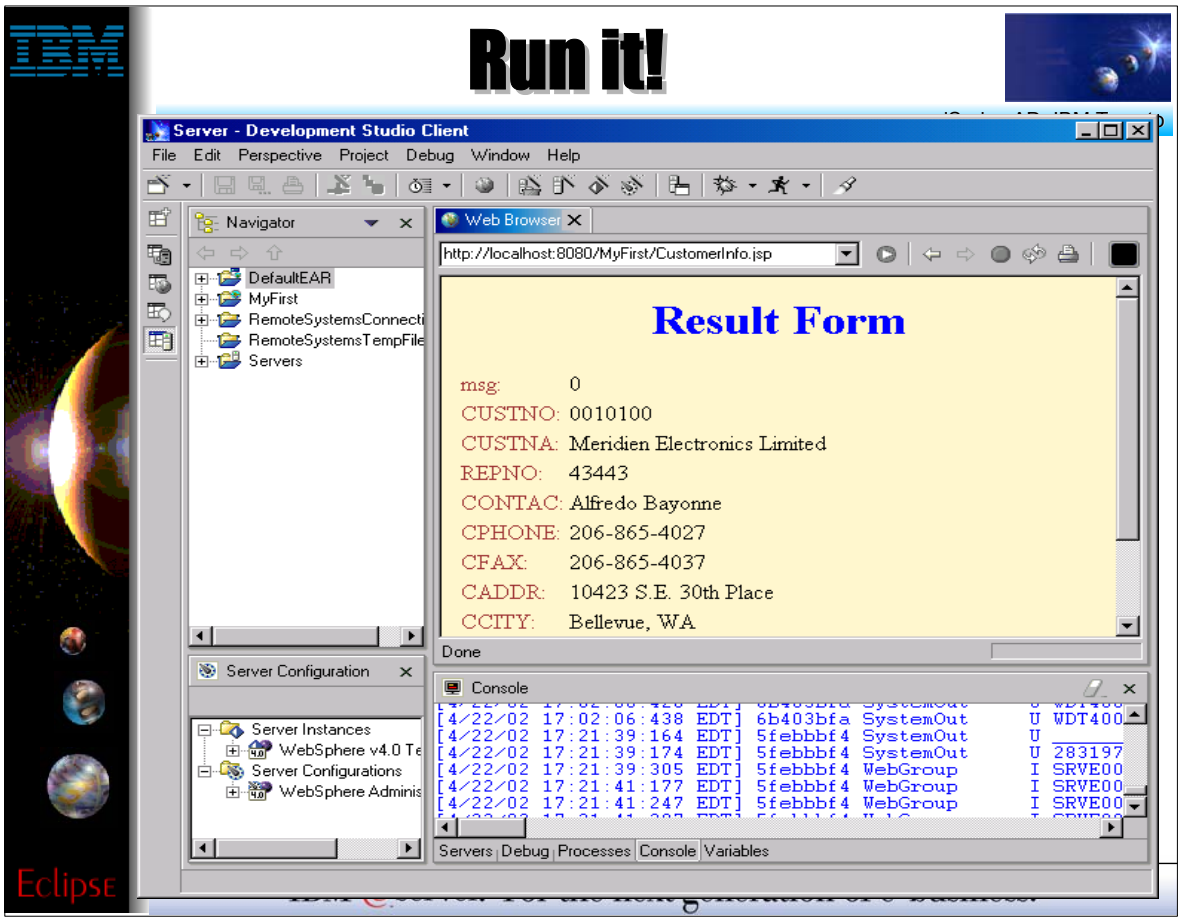
- **What did you do so far?**
  - You used the designer to create the input page
  - The output page was generated for you
  - Created your Files on the iSeries
  - Created your RPG Program
  - Created an 'iSeries Web Interaction' Interaction to link your program parameters to the input page and output page
- **Next?**
  - Run locally to test your application Or

- This slide shows what we just did and what gets generated:
- The output (result) page jsp
- The PCML to describe your iSeries program
- A servlet
- The Form tag on the input page gets updated with the information which servlet to invoke when the submit push button is pressed
- We only have to write the RPG program.

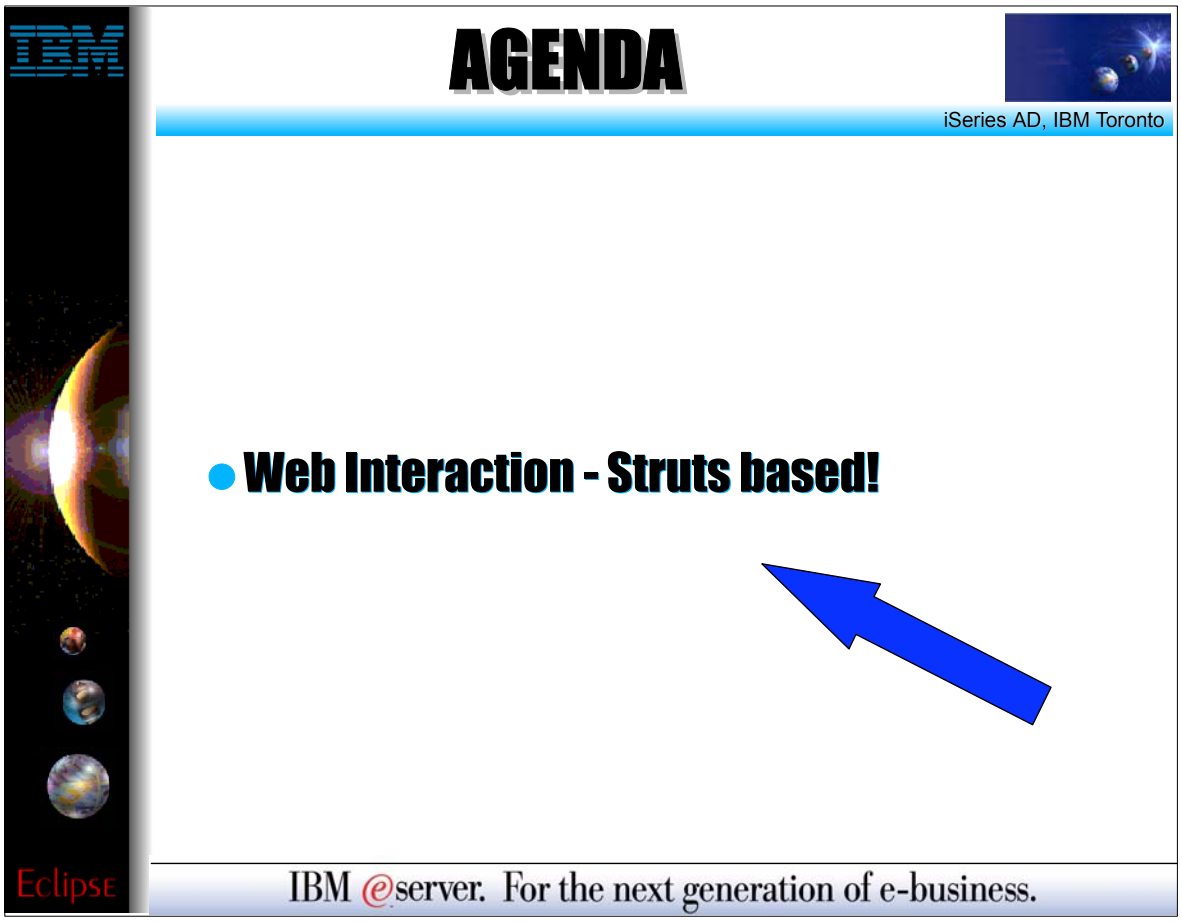
- To test this, we can select our .jsp input page in our Web project in the Web perspective in the workbench
- Right mouse click on it and select Run on Server from the popup window
- The built-in WAS test environment will start
- The built-in browser will show the input page
- Key in a valid customer number
- Press the Submit push button
- The customer number will be sent to the RPG program
- If the number was valid the customer data stored in our data base will be displayed



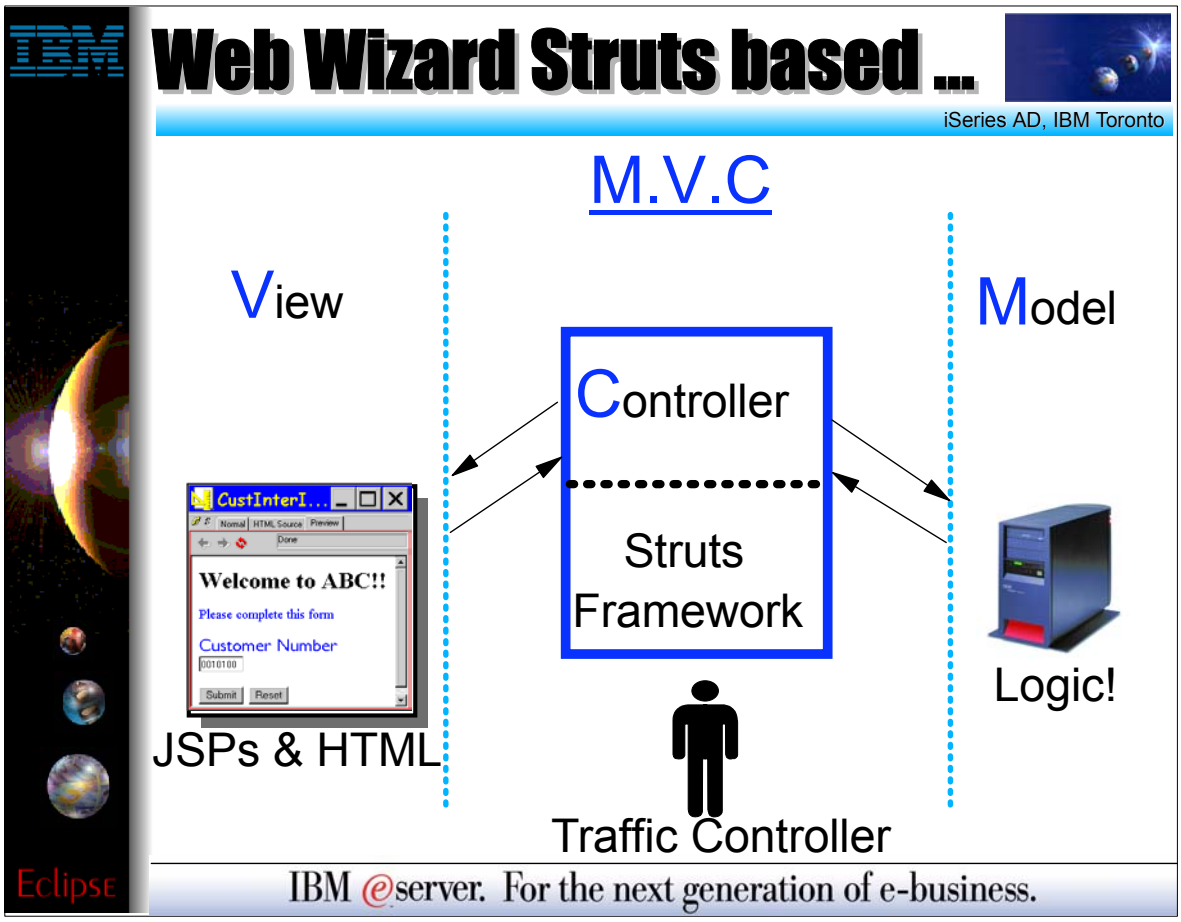
- The WebSphere test environment provides a fully functional test bed for Web applications without having to setup WAS on a remote server.



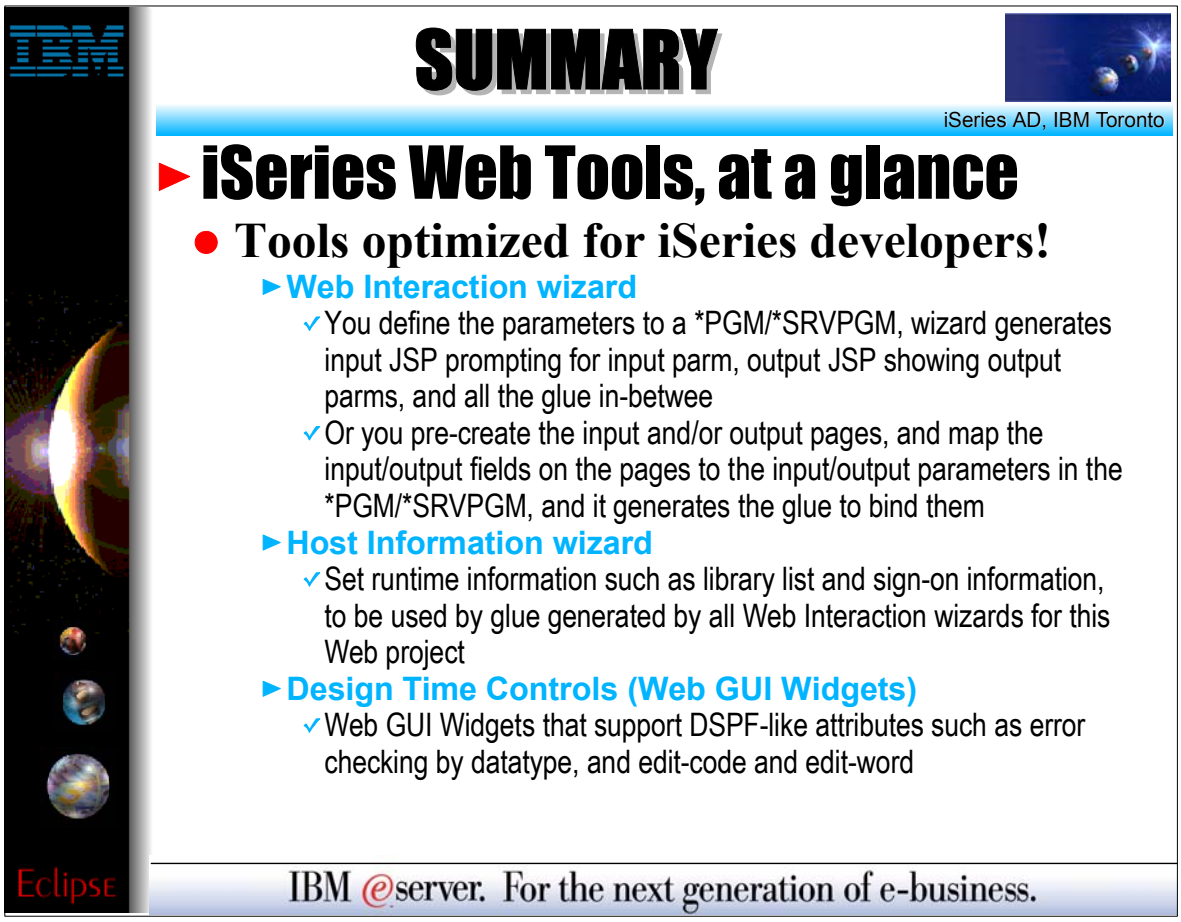
- Here we are, the Result page is displayed



- Web Interaction wizard now generates a Web application based on a Struts runtime

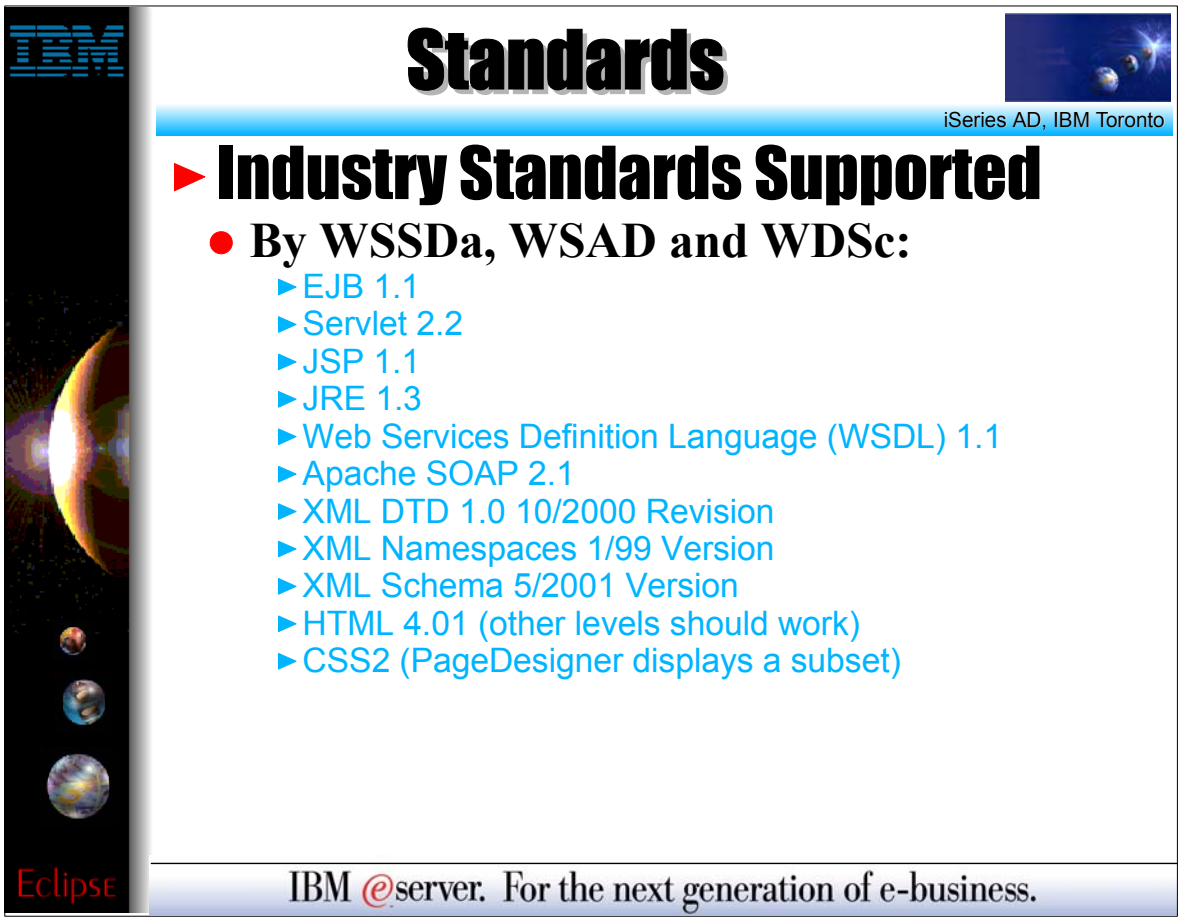


- Struts-based Model View Controller that allows you to map out and visually construct Web-based applications

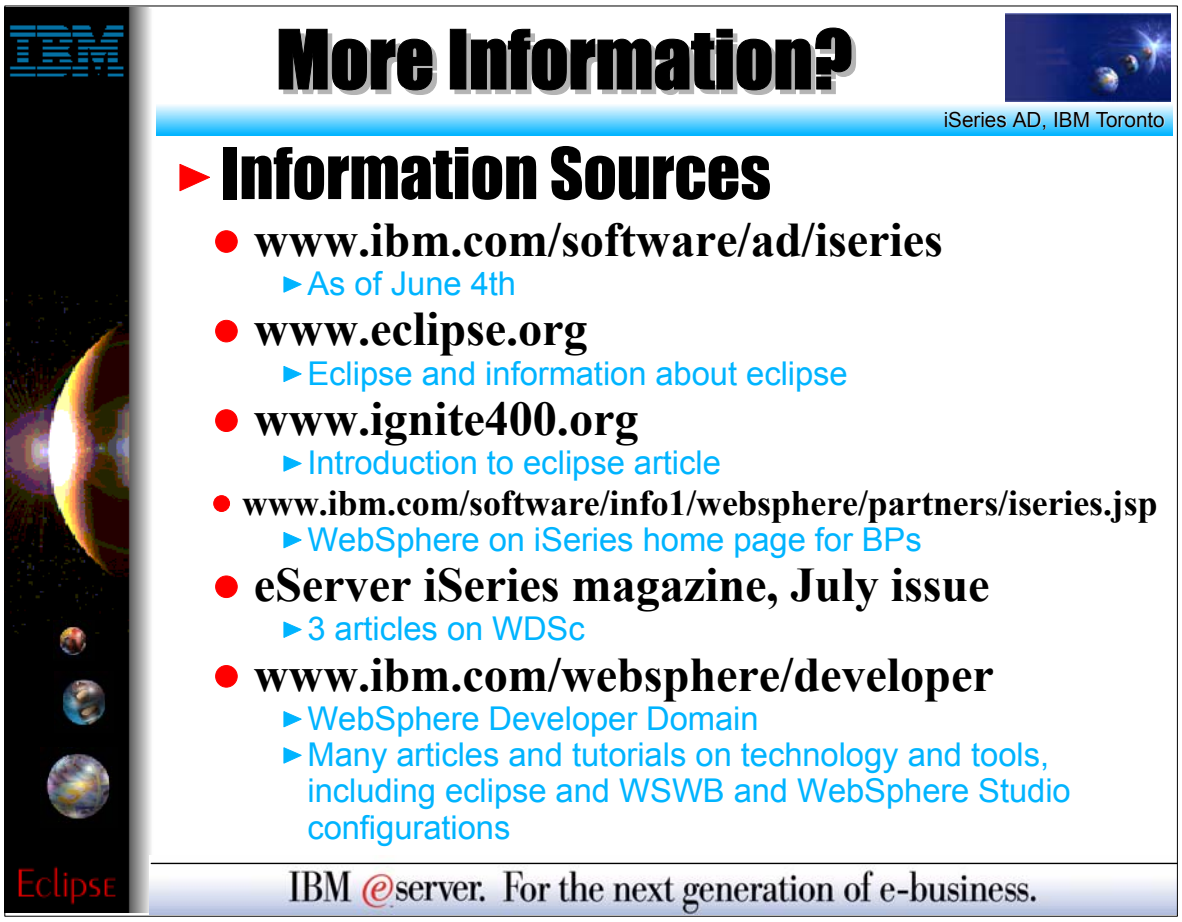


- Wizards make it easy to 'wrapper' existing iSeries programs as Java Beans and/or Web Services. This lets the iSeries developer access these programs from a Java application, a Web application, or include them as the base for a Web Service.
- Extensions to the Struts builder make it easy to visually construct Web applications based on the open-source Struts standard that accesses iSeries programs
- Visual Custom Tags bring some of the power of DDS to Web pages


Standards
iSeries AD, IBM Toronto
Industry Standards
Eclipse
IBM @server. For the next generation of e-business.



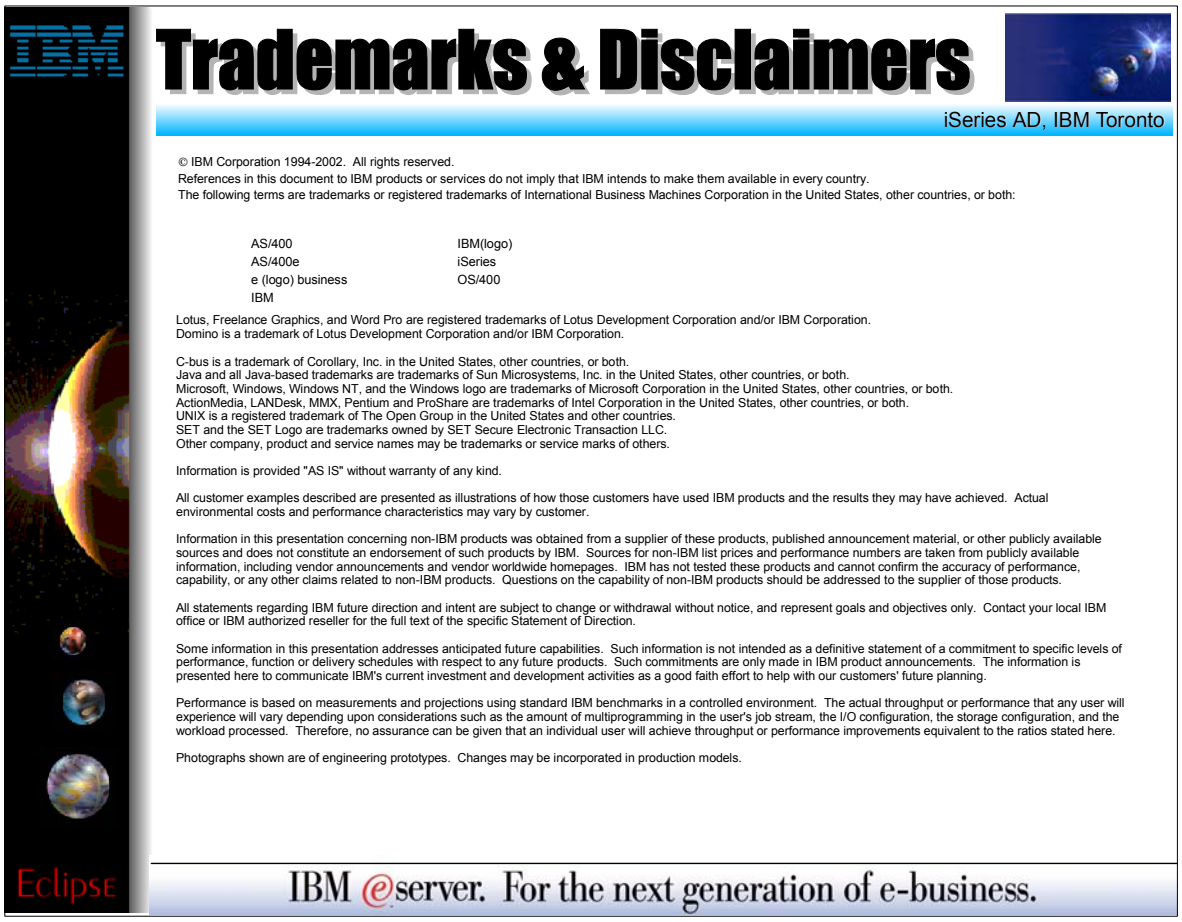
- WDSC is keeping up with industry standards, most of which IBM wrote or contributed heavily to



More Information?

Information Sources

- **www.ibm.com/software/ad/iseries**
  - As of June 4th
- **www.eclipse.org**
  - Eclipse and information about eclipse
- **www.ignite400.org**
  - Introduction to eclipse article
- **www.ibm.com/software/info1/websphere/partners/iseries.jsp**
  - WebSphere on iSeries home page for BPs
- **eServer iSeries magazine, July issue**
  - 3 articles on WDSc
- **www.ibm.com/websphere/developer**
  - WebSphere Developer Domain
  - Many articles and tutorials on technology and tools, including eclipse and WSWB and WebSphere Studio configurations


Trademarks & Disclaimers
iSeries AD, IBM Toronto
© IBM Corporation 1994-2002. All rights reserved.
References in this document to IBM products or services do not imply that IBM intends to make them available in every country.
The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:
AS/400
AS/400e
e (logo) business
IBM
IBM(logo)
iSeries
OS/400
Lotus, Freelance Graphics, and Word Pro are registered trademarks of Lotus Development Corporation and/or IBM Corporation.
Domino is a trademark of Lotus Development Corporation and/or IBM Corporation.
C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.
Other company, product and service names may be trademarks or service marks of others.
Information is provided "AS IS" without warranty of any kind.
All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.
All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.
Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.
Photographs shown are of engineering prototypes. Changes may be incorporated in production models.
Eclipse
IBM @server. For the next generation of e-business.