



IBM Toronto

Phil Coulthard and iSeries AD Team

iSeries Application Development Strategy

IBM eServer iSeries Application Development

2003 | WDS V5R2 and WDSC 5.0

© 2003 IBM Corporation

Title page

Agenda

- **Strategy**
 - Vision
 - Packaging
 - Eclipse
 - WebSphere Studio Family
- **WebSphere Development Studio Client 5.0**
 - RSE and iSeries Projects
 - WebFacing
 - Web Tools, Java Tools, Web Services Tools
 - XML, DB Tools

This is the agenda we'll follow in this presentation

iSeries Application Development Strategy



The iSeries Application Development team at IBM Toronto brings to you the RPG, COBOL, C, C++ compilers; the traditional toolset like PDM and SEU, and then new generation of workstation-based tools

Since the year 2000, The iSeries AD team has been working towards a strategy to re-energize AD on iSeries, returning the iSeries to its historical position as the world's most productive operating system for application development

The IBM iSeries AD team is committed to re-focusing the iSeries programming community around a single common set of technologies and tools that all developers have and all developers use. Much as it was in 1988 when all developers had and used RPG/COBOL, PDM and SEU.

This also extends to the business partner community that supplies tools: the goal is a technology and tool common base that business partners complement versus compete with. This is as opposed to the client/server years where there were numerous competing technologies and tools, none of which were compatible with each other.

Further, the IBM iSeries AD team is committed to re-energizing the iSeries programming community with very exciting and compelling technologies and tools, both from IBM and business partners. These technologies and tools in turn will entice developers to build compelling and competitive applications.

IBM Toronto

iSeries AD Strategy

- New Packaging**
WebSphere Development Studio
- New Technology**
Eclipse, Web, XML, WebServices, etc
- New Tools**
WebSphere Development Studio Client
- New User Interface**
Web

4 | 2003 | WDS V5R2 and WDSC V5.0 | © 2003 IBM Corporation

These points tell what IBM's strategy is to reach its goals of building community and building excitement.

The new packaging was actually done in May 2001, with the introduction of WebSphere Development Studio for iSeries, which as we will see offers a single product with all the host and client tools needed for all application development needs, from traditional to e-business

With the 4.0 release of the client tooling in June 2002, we introduced new Eclipse technology and totally new Eclipse-based tools. Further, for technology we also plan to keep up with technology both in the compilers and the tools, so iSeries programmers are very current. This was evident in the V5R2 release of the compilers in Sept 2002, and will be again in the V5R3 release of the compilers.

With the April 2003 new 5.0 release of the client tooling, we are in the second release of the eclipse-based technology and continue to improve the tools and introduce new technology.

The preferred user interface for iSeries applications is now a Web browser, versus a 5250 green screen or even a client/server GUI. There are exciting new and enhanced tools to make it easy to convert 5250 user interfaces into Web user interfaces, and to build new Web user interfaces on top of new or existing business logic

IBM Toronto | New Packaging

WebSphere Development Studio V5

Unlimited Licenses

iSeries	iSeries	iSeries	iSeries	Web Facing	iSeries Projects
Java	Debug	Struts Web	Web Service		RSE
Trace	Profiling	DB	XML	App Server	

RPG **COBOL**

C/C++ **PDM, SEU, SDA, RLU**

V5R1 or V5R2
5722-WDS
No-cost V4 Upgrade 5903
www.ibm.com/software/awdtools/wds400

+CODE
+VisualAge RPG

WebSphere Development Studio Client V5

5 | 2003 | WDS V5R2 and WDSC V5.0 | © 2003 IBM Corporation

There is now only one AD product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (=WDS), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and unlimited licenses of the workstation-based toolset named WebSphere Development Studio Client (formerly WebSphere Development Tools).

Existing AD customers who have subscription can upgrade to WDS free of charge. Without SS, there is an upgrade fee. New licenses of WDS are priced very competitive compared to the combined prices of all constituent products. As of V5R1, there is no way to purchase the compilers or tools individually. So if a customer has RPG at V5R1 or higher, they must have WDS and hence are entitled to WDSC.

For consultants who do not have an iSeries of their own, but still wish to have the client tools, WDSC is also made available as a passport advantage product so it can be purchased "off the shelf" from IBM Direct.

WDS has been a huge success, with over 80,000 licenses sold

Just as every development machine used to have PDM and SEU, every development machine will now have all the modern AD tools from IBM.

This ubiquity is especially important for business partners who build and sell software. These BPs are now free to build it using any of the technologies or tools in WDS, and can assume their customers will have the tools required to tailor everything from RPG to Java and Web user interfaces. This effectively raises the lowest common denominator to a level unparalleled by any other operating system.

IBM Toronto | New Packaging

WebSphere Development Studio Advanced V5

NEW!

Unlimited Licenses

RPG	COBOL
C/C++	PDM, SEU SDA, RLU

V5R1 or V5R2
5722-WDS
Cost Upgrade 5904
www.ibm.com/software/awdtools/wds400

iSeries	iSeries	iSeries*	iSeries	Web Facing*	iSeries Projects
Java	Debug	Struts Web	Web Service	App Server	RSE
Trace	Profiling	DB	XML	EJB* J2EE*	Test Cases*

+CODE
+VisualAge RPG

WebSphere Development Studio Client Advanced V5

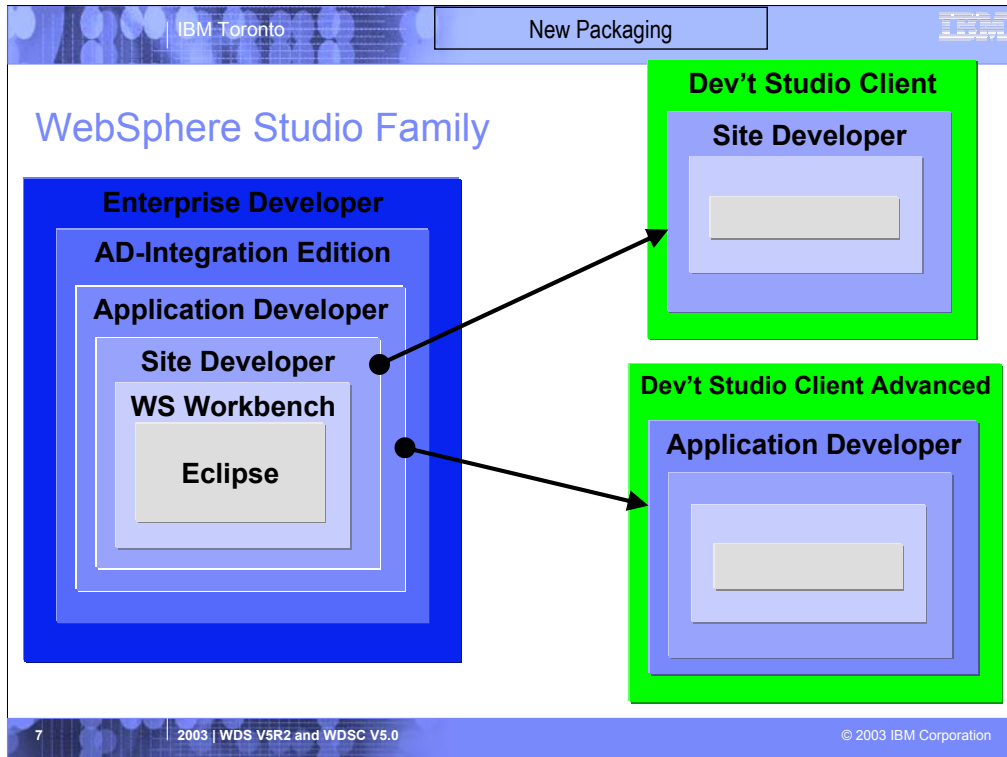
6 | 2003 | WDS V5R2 and WDS V5.0 | © 2003 IBM Corporation

The Advanced edition of WDSC, and WDS, is new as of April 25th 2003. WDS Advanced is currently the same as WDS, except the customer is entitled to unlimited licenses of WDSC Advanced versus just WDSC.

The difference of WDSC Advanced over WDSC standard is that it has additional tools (blue boxes with asterisk) and some enhanced to existing iSeries tools (green boxes with asterisks) which will be discussed as go on.

The majority of the new tools are related to EJB development.

You'll see later that Advanced is based on WSAD versus WSSD.



The point of this slide is to show that WDS*c* is based on WSSD, while WDS*C* Advanced is based on WSAD

The Workbench is based on the open-source eclipse technology about to be discussed. It is not for sale, but is the basis of all IBM WebSphere Studio products, and is available to business partners

WSSD is IBM's entry level offering based on eclipse, and it is for building dynamic web sites out of non-EJB Java

WSAD extends WSSD and adds support for EJBs

WSAD-IE extends WSAD and adds support for JCA Connectors and for Workflow

WSED extends WSAD-IE and adds support for s/390 and EGL, the follow-on to VAGen

IBM Toronto | New Technology

Eclipse

eclipse eclipse.org

- A base Integrated Development Environment
- Comes with built-in rich Java tooling
- Extensible via plug-ins
- Used as basis of products = Eclipse + plug-ins
- Open source
 - Contributed by IBM, managed by consortium
- Millions of user downloads
- 35 products offerings powered by Eclipse
- 25 companies in consortium
- 175 companies writing plug-ins

8 | 2003 | WDS V5R2 and WDSC V5.0 | © 2003 IBM Corporation

Eclipse was developed by IBM and donated to the open source community. That donation is estimated to be worth \$40 million.

Anyone can download eclipse for free, including the source code, from www.eclipse.org

Eclipse has generated extraordinary excitement in the development community and the tooling community. It is written in Java, and can be extended by tools that are also written in Java. These tools are known as plug-ins.

Out of the box, eclipse offers an IDE that has built-in support for teams and projects and a robust and revolutionary user interface framework. It also has tooling built-in for create eclipse plug-ins.

Further, there are extensive and very powerful tools built-in for developing Java applications with eclipse. So, if all you want is the world's best Java toolset, then all you need is eclipse. You can't beat the price!



IBM is not alone with eclipse. The open-source consortium that oversees contributions made to it include a number of large companies, and that list is growing.

Note MKS, an iSeries tool vendor, is on the list.

See www.eclipse.org for the latest list

Eclipse

- **Eclipse + plug-ins = product**

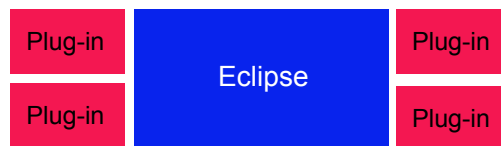
Although it is a great Java IDE out of the box

- **Eclipse is already the basis of many products, and more to come**

Both from IBM and Business Partners

- **Eclipse is a huge opportunity for ISVs**

For writing and selling plug-ins



So what is eclipse? With the exception of the built-in Java tools, eclipse itself is not that interesting. It only gets interesting when you add to it some plugins (eg tools) that do something interesting.

An Eclipse-based product is eclipse plus a number of interesting plugins. IBM is building numerous such eclipse-based products, including as we have seen WSSD and WDS. Further, because eclipse is free and business partners are free to include it in their products, there will be many other eclipse-based products from other companies too, including Rational and TogetherSoft.

For business partners or software developers who write and sell AD tools, eclipse is a fantastic opportunity. By writing plugins for eclipse, those plugins can be sold to any developer using any product based on eclipse, or even just the raw eclipse as downloaded from www.eclipse.org

This opportunity is not lost on iSeries tool vendors, who are all looking at offering eclipse plugins for their tools. This will result in a rich offering of third party plugins for developers to choose from, all of which extend their core WDS development environment. One community, one core development platform, many IBM and 3rd party tools. Community and excitement!

Eclipse Plug-in Development



Products built with Eclipse inherit these capabilities plus 'plug-ins' built by others



Shows some of the common services Eclipse supplies

Eclipse User Interface

■ User interface

Users work with **perspectives**

Collection of **views** and **editors**

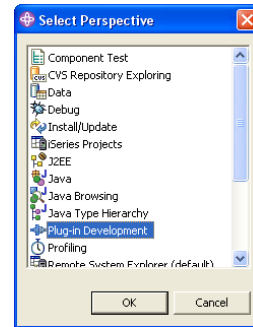
Tools for a particular task

Allows for role-based development

Many perspectives are pre-supplied

For specific tasks like Java, Web, XML, RPG/CBL

Users can create their own perspectives



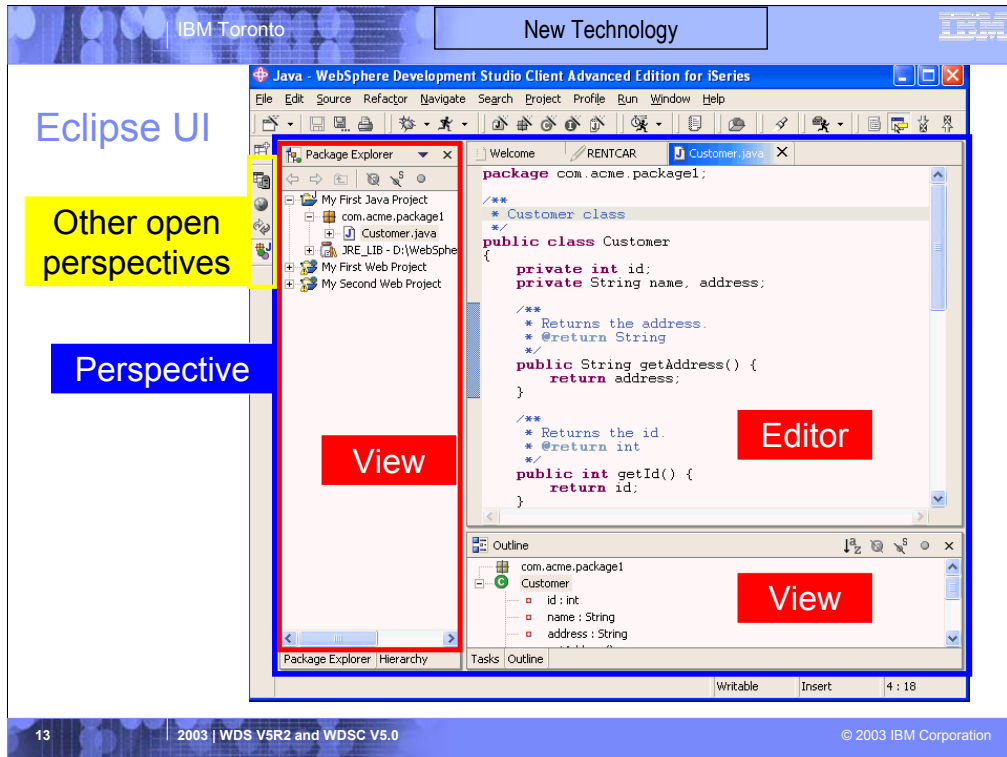
The user interface is very Windows-like

Build from a Java wrapper of OS widgets

Behaves and feels like any other native app

Eg: views can be re-sized and re-positioned via drag 'n drop

Describes the core features of the Eclipse user interface.
Naturally, applies to all eclipse-based products, like WDSC.



This shows the eclipse workbench, which has four open perspectives (see them lined up on the left). The currently active perspective is the one whose icon is indented, which in this case is the java perspective.

To open a new perspective, you use Window->Open Perspective giving you the Select Perspective dialog from the previous page.

This particular perspective has a tree view which drive the others view, and editor, and an outline view.

Eclipse Team Support



Team

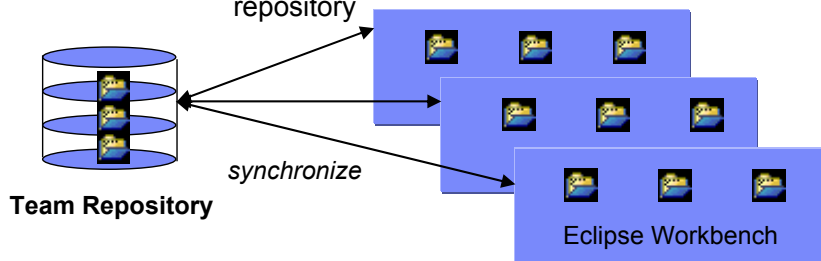
Users create and work with **projects**

Contain **folders** and **files**

Which are versioned as they change

Are “typed” and have type-unique tooling

Are **synchronized** between local workspace and team repository



Central to the eclipse IDE is support for projects. Projects are simply a grouping of folders and files

Plug-ins have types, such as Java or Web or XML. Tools that plug into eclipse can define their own new project types. Tools that plug into eclipse work against resources (folders or files) within projects. They can be scope to only projects of a particular type, if appropriate.

All projects, regardless of type, have common behavior and support. This includes the ability for a team to share a project by using an eclipse-supporting software change management (SCM) product such as CVS or Rational Clearcase. This SCM acts as a central repository for one or more projects. Each team member can easily keep their local copy of any project "in synch" with the central repository copy.

CVS is Concurrent Version System, which is a free open-source SCM. It runs on Linux, Unix and Windows. It comes on the Linux distribution CDs for iSeries Linux lpar.

All IBM SCM vendors for iSeries are enabling, or have enabled, their products to eclipse

Agenda

- Strategy
 - Vision
 - Packaging
 - Eclipse
 - WebSphere Studio Family
- WebSphere Development Studio Client 5.0
 - RSE and iSeries Projects
 - WebFacing
 - Web Tools, Java Tools, Web Services Tools
 - XML, DB Tools

We will now focus on a brief discussion of the tools in WDSC 5.0

WebSphere Development Studio Client

- *Tools for ebusiness*

- For **RPG and COBOL**

- Remote System Explorer and iSeries Projects perspectives

- For **WebFacing**

- WebFacing perspective

- For **Java**

- Java perspective and iSeries enhancements

- For **Web Services**

- Web Services perspective and iSeries enhancements

- For **Web**

- Web perspective and iSeries enhancements

- For **XML**

- XML perspective

- For **Database**

- Database perspective

This lists the primary tools in WDSC we will briefly discuss. It is not a complete list.

WebSphere Development Studio Client

RPG and COBOL Tools

This leads the primary tools in WDSC we will briefly discuss. It is not a complete list.

RPG and COBOL Tools

▪ Remote System Explorer (RSE)

The
new
PDM!

The
new
SEU
and
SDA!

- Perspective plus supporting views and editors
- Random-access drilldown to **remote files** on iSeries, Unix, Linux, Windows and local. Like Windows Explorer
- For iSeries, supports **all environments**:
 - QSYS, IFS and Linux LPAR
- Rich command shell for running **remote commands**
- Explicit and rich iSeries-specific support for **edit, compile, run, debug** of RPG, COBOL, C, C++, CL and DDS . . .
- No built-in tool for WYSIWYG DSPF/PRTF (yet), but can easily launch the CODE Designer from the Remote System view
- Framework** for ISVs/PBs to plug-in additional actions and remote-accessing tools. Lots of re-usable APIs for accessing remote resources

The first area of functionality is the Remote Systems Explorer, which has its own perspective and views

This first area of functionality is similar to PDM in that it allows the developer to drill down into the QSYS file system, or use filters to list specific objects within the QSYS file system

The Remote Systems Explorer goes well beyond PDM however! It also allows exploration of iSeries jobs and commands, and the IFS file system. Further, it can also be used to explore the file system of remote Linux, Unix and Windows systems. The Linux support works for any Linux, including Linux in an iSeries Logical Partition.

IBM Toronto | New Tools

WDSC 5.0 Enhancements

- **What's new in RSE for 5.0**
 - Library list manipulation support
 - More wizards, improved user actions and compile support
 - Drag and drop
 - PDM-like table view with command line
 - Editor enhancements:
 - Syntax checking for RPG, COBOL, DDS, CL
 - Program verifiers for RPG, COBOL, DDS
 - F1 help support for RPG, COBOL, DDS, CL
 - Outline view for RPG, COBOL + content assist for RPG
 - List fields support for F-spec, COPY-DDS
 - Performance enhancements
 - Rich command console for Unix, Windows, Linux and Windows
 - New common debug UI, and editor breakpoint support
 - Remote Text Search (ie, FNDSTRPDM interface)

19 | 2003 | WDS V5R2 and WDSC V5.0 | © 2003 IBM Corporation

Lists the primary enhancements in 5.0 versus 4.0: it is not a complete list.

Library List -> support for a new library list filter, and actions for manipulating the library list. Much like *LIBL support in WRKLIBPDM.

More Wizards -> new easy-to-use wizards to create libraries, files, members, data areas, data queues and message files

Drag and Drop -> Can now drag source members to another file, or another iSeries, or to any other system. Similarly for local, unix, windows and linux source files.

PDM-like Table -> The primary view in RSE is a tree view, but PDM users are used to a table, so there is now an iSeries Table view that shows what the tree shows,

but in tabular format. The columns are sortable by clicking their header, and contents easily replacable ... eg by double clicking on a file. There are

right-click actions that are the same as PDM's, and a command line at the bottom just like in PDM.

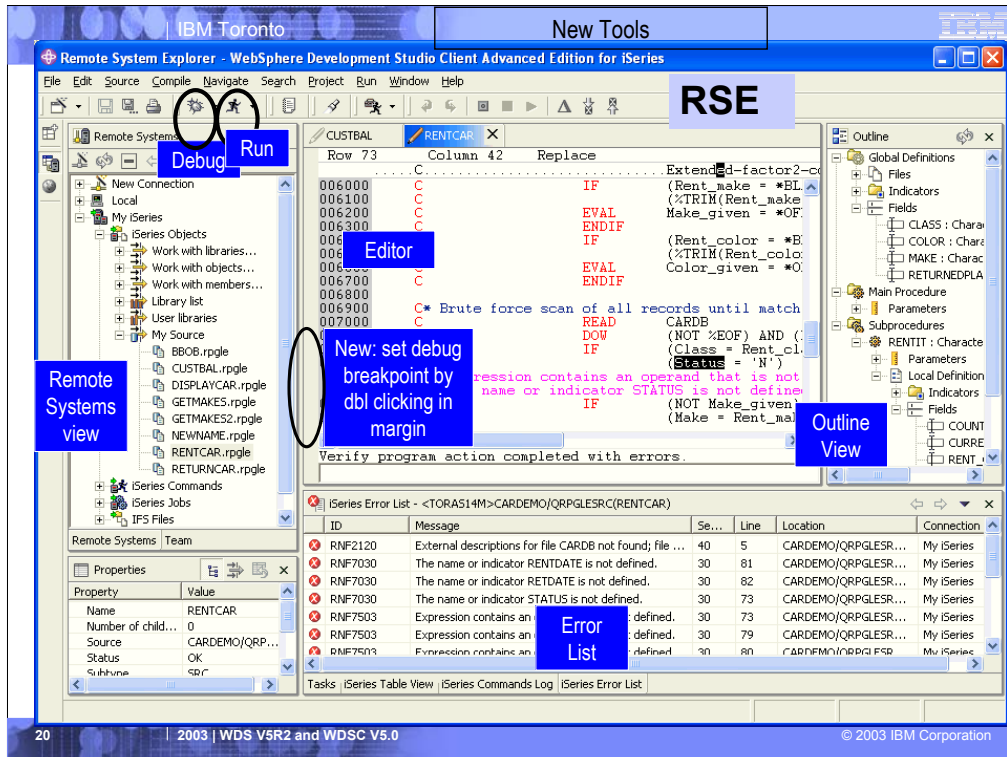
Editor Enhancements -> Now well beyond SEU's function, and getting close to all of CODE's. The syntax checker is ported down from SEU, the compilers are imbedded

for verifying all errors, and the reference manuals are built-in and F1 cursor sensitive. There is a new outline view showing the program hierarchy

at a glance, and something really cool called content assist: using Ctrl+Spacebar it offers lists of options for code-completion, sensitive to the cursor.

This is currently only for RPG. There is also a List fields option when the cursor is on an F-spec or COBOL Copy-DDS statement ... it shows a window

showing all the fields in the referenced file. There are also significant



This is the Remote System Explorer perspective.

See the debug and run icons in the toolbar. These are common in eclipse, and now apply to iSeries programs too.

The Remote System view is the primary drill-down view, similar to PDM. Double-clicking a member opens the built-in Lpex editor shown in the middle. Notice the outline view to the right. Clicking on an element in the outline view positions you in the editor. The error list below shows the results of doing a program-verify, or a compile. The former is launchable from the Source pulldown, the latter from the Compile pulldown.

Note the other tabs behind the error list view. There are numerous useful views in the RSE perspective.

RPG and COBOL Tools

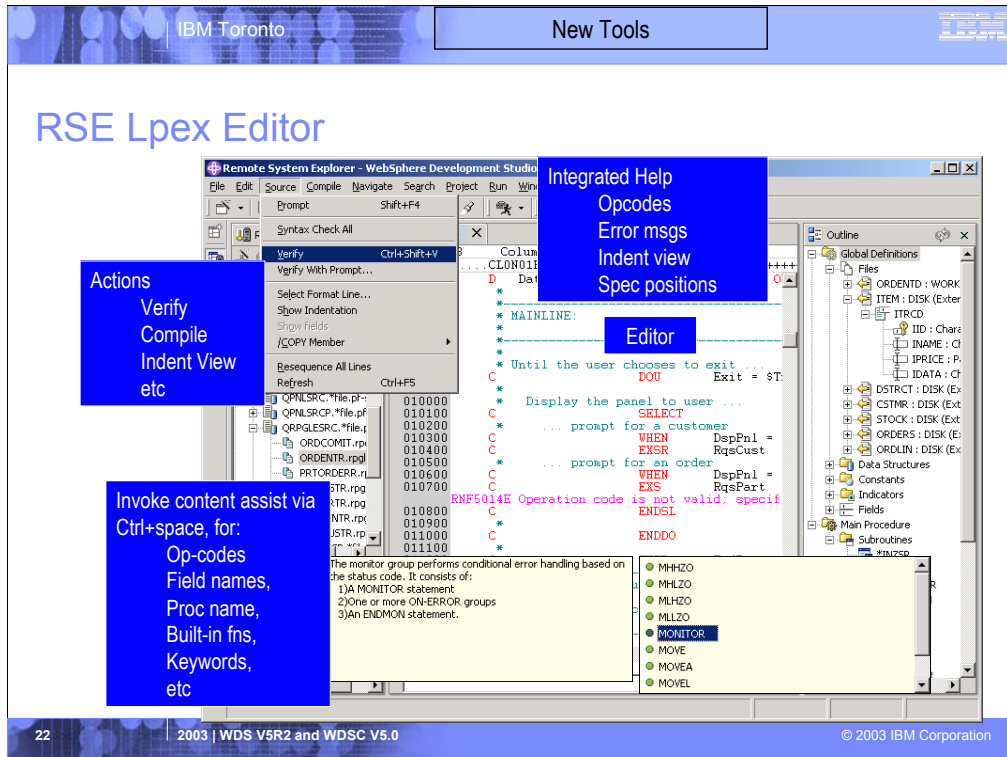
▪ Remote System Explorer Scenario

- Create a connection to your iSeries system
 - After created, configure its library list, env variables, etc
- Expand connection and subsystem
 - Decide if you want to work with objs, cmds, jobs or IFS
- Create Filter, such as MYLIB/*ABC OBJTYPE(*FILE)
 - For accessing particular objects or members
- Expand Filter
 - Drill down to object for manipulation
 - Use right click actions to change, renamed, delete, etc
 - Drill down to member for editing
 - Double click to open in editor
 - Edit, Syntax Check, Verify, Compile, Run, Debug

The first area of functionality is the Remote Systems Explorer, which has its own perspective and views

This first area of functionality is similar to PDM in that it allows the developer to drill down into the QSYS file system, or use filters to list specific objects within the QSYS file system

The Remote Systems Explorer goes well beyond PDM however! It also allows exploration of iSeries jobs and commands, and the IFS file system. Further, it can also be used to explore the file system of remote Linux, Unix and Windows systems. The Linux support works for any Linux, including Linux in an iSeries Logical Partition.

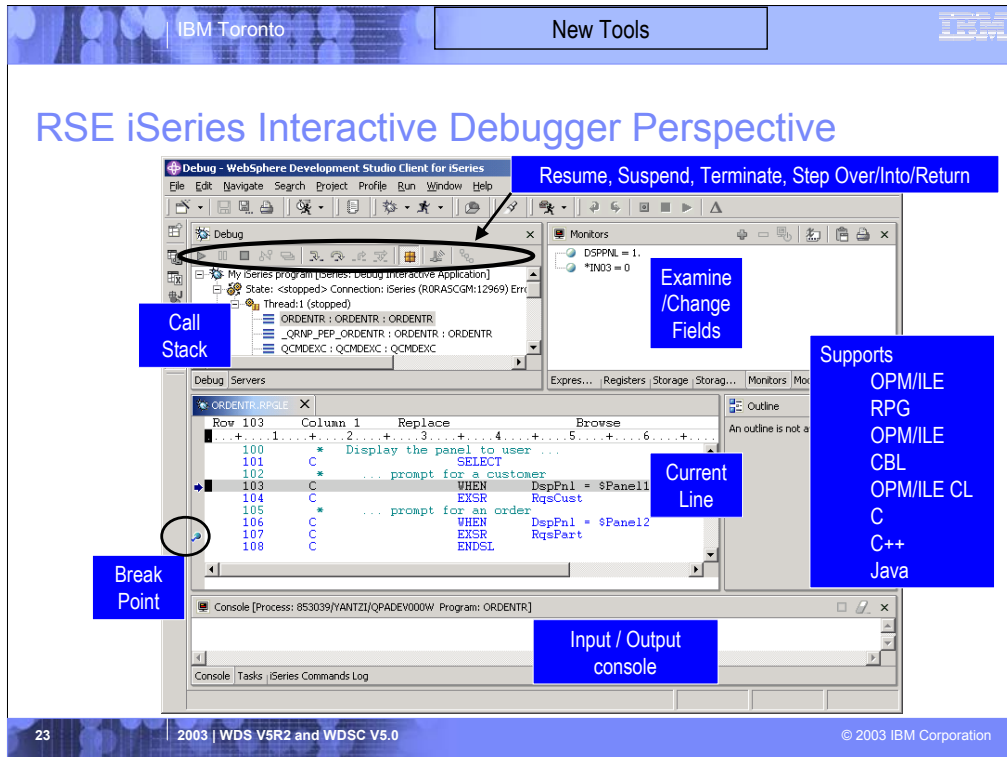


This picture zero's in on the most compelling aspect of the RSE ... the Editor with its rich support.

For example, the content-assist shown here is very well received by programmers. It is invoked via Ctrl+Spacebar.

The Source pulldown is where many of the editor's advanced functions are launched.

The RSE Editor (Lpex, written in Java) is a superset of SEU.



This picture shows the common Eclipse debug perspective, but which is being used to debug an RPG program.

The common debug user interface has been connected to the iSeries debug engine in 5.0, to offer a common and compelling debug story for Java, OPM/ILE RPG and COBOL and CL, and ILE C and C++.

There are numerous ways to launch debugger, including a popup menu action in the RSE and the Debug icon in the toolbar.

In the upper left is the call stack, much like option 11 in the OS/400's WRKACTJOB. It shows the calls that got you to the current position.

By double clicking an item in the stack its source (if available) is shown in the source pane in the middle.

The upper right is where all the various views are for working with data contents.

The middle is the debugger source view, which source executable (debuggable) lines in blue, others in green.

The current line of execution is highlighted, and breakpoints appear as a dot in the left margin.

RPG and COBOL Tools

■ iSeries Projects Perspective

- Structured support** for RPG/CBL development
 - Treats iSeries as a “push” and “build” target
 - Leverages Eclipse team support for source control
 - Extendible by iSeries Change Management vendors
 - Can plug-in own build “styles” and remote actions
 - Is extended by SoftLanding, MKS and Aldon!**
 - Is an effective disconnected environment
 - Use import, or RSE actions, to pull down
 - Verify while connected to populate cache
 - Create, edit and verify while disconnected
 - Verify uses caching support for F-specs
 - Push changes and build when connected again

The second area of functionality is iSeries Projects, which also has its own perspective and views.

Unlike the Remote Systems Explorer, an iSeries project fully leverages the Eclipse support for resources. This means an iSeries project contains folders and files that exist in the local file system, yet can be shared and synchronized among a team, if an eclipse-compliant repository is used. The options for this today include Rational's ClearCase or the free and open-source product CVS or Concurrent Versioning Systems, or the MKS Integrity. CVS is available on the distribution CDs for the Linux LPAR on iSeries. In the near future, the remaining iSeries change management vendors will also be available as repository options.

As you will see, an iSeries project allows RPG and COBOL developers to fully exploit the power Eclipse, while developing applications targeted to run on iSeries.

WDSC 5.0 Enhancements

■ What's new in iSeries Projects for 5.0

- Enhanced "build" support
 - Support for generating and running CL to build changed source
 - New view for monitoring build jobs
 - Compile-error feedback after build (clickable error list)
- Checking for synchronization errors (file changing locally and remotely)
- RSE actions to create project from library, or add to project
- Action to show project object in RSE
- ISV support and documentation
 - Pluggable build support
 - APIs for creating and populating projects
 - Support for ISV-specify persistable properties

Lists the primary enhancements for 5.0.

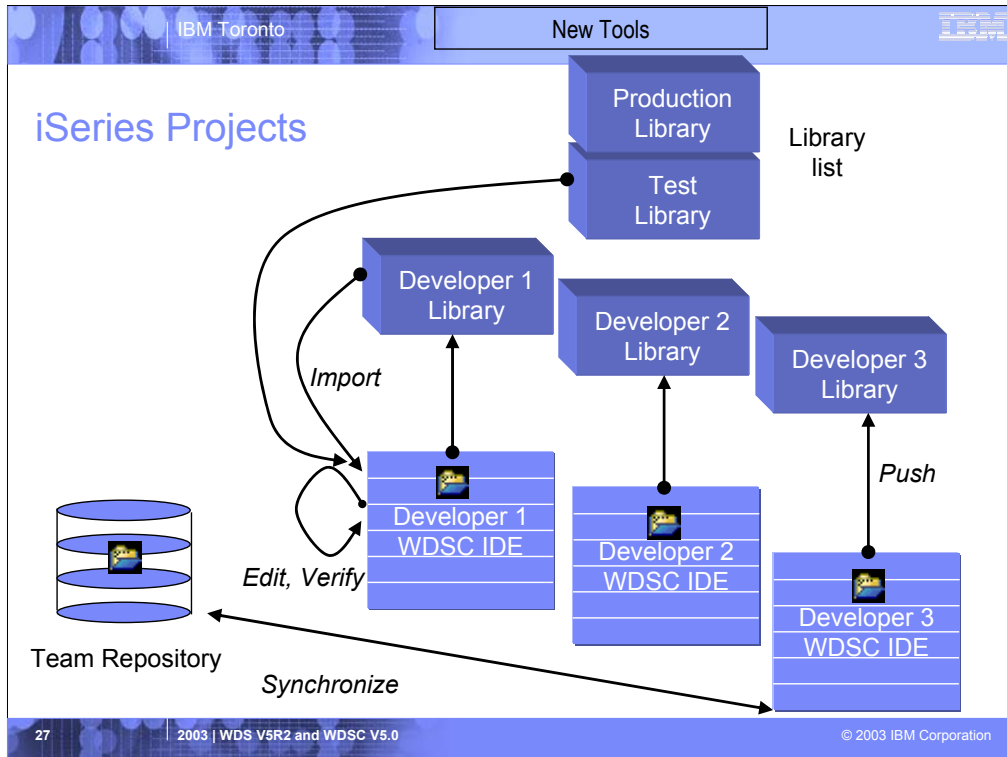
This is not a complete list

RPG and COBOL Tools

■ iSeries Projects Scenario

- Project created to do a fix or enhancement
 - Associated with library
 - Populated with source required for work
 - From RSE, or via Import action
- Editing work done (can be done offline)
 - Edit, syntax check, verify, F1
- Changes pushed to associated library
- Build performed
 - Using IBM-supplied build style, or 3rd party such as SoftLanding or MKS
- Run or Debug
- If working in team, synchronize with team repository

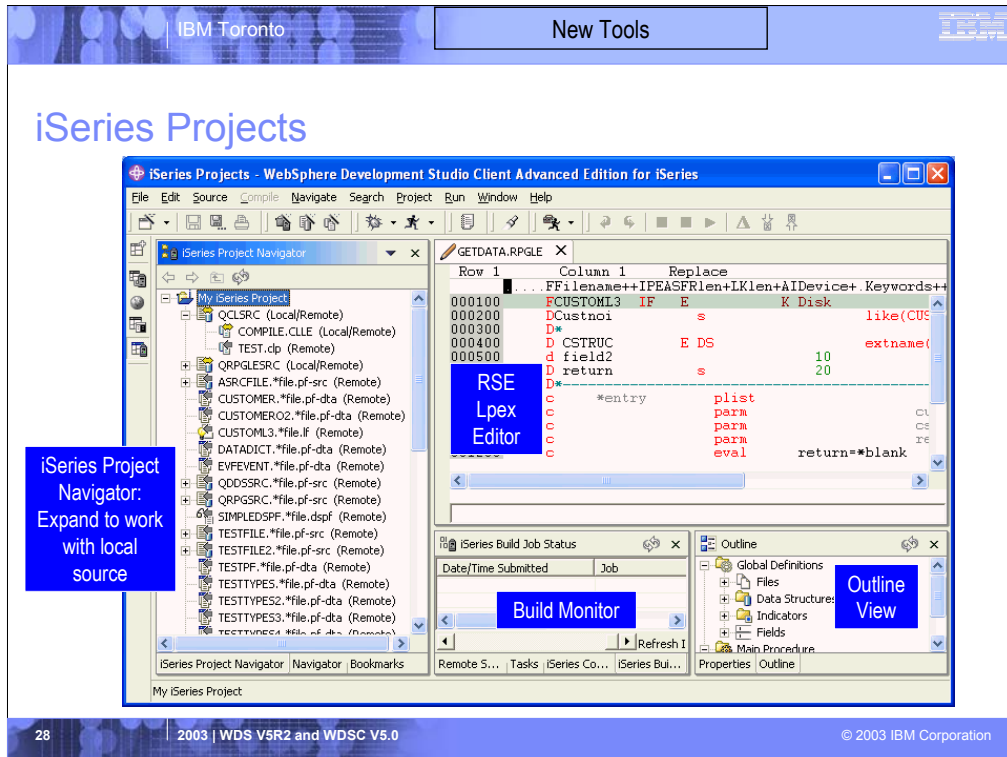
This shows a typical usage scenario for iSeries Projects.



Because iSeries projects are in fact Eclipse projects, they inherit the team capabilities of all eclipse projects. This offers developers the ability to share their work among a team, using the eclipse team support

One scenario where this is useful is the case where a small team is working on an enhancement to an existing application. If there are 3 developers, one would create an iSeries project, associate it with a team repository and synchronize it with that repository to make it available there. The other members would use the Add To Workspace action in the eclipse team perspective to copy it to their local workspace within their IDE. Each developer would change the properties of the project to specify their own library as the associated library for the project.

Then, each developer would import or create members, do their changes, push their changes to their associated library and do build to compile their changes there. Every so often, they would synchronize with the team repository to pick up the changes of their colleagues, then push those changes to their own library and build them there. This way, only when they decide it is time, they would test their individual changes together. When the work is done, the leader would use their iSeries change management User Interface to promote the changes from the leader's associated library to the change management system so that official testing could begin



iSeries Project Navigator: Expand to work with local source

RSE Editor

Build Monitor

Outline View

This is the iSeries Projects perspective.

The navigator view on the left is the primary view that drives the other views. It lists all the local files in the project, by default, but can also be used to show all the remote objects in the associated library as well, as is the case here. Anything only in the remote library is adorned with the text (Remote), anything only available locally but not yet created in the library shows (Local) while something in both places shows (Local/Remote). If a member is in both the library and the local project, its icon is yellow if they are in sync, and red if there are collisions.

The editor is the same rich editor we saw for the RSE.

When done editing and the project is pushed to the library and built, the build job is monitored in the job status window. When done, you can right click on a job and select to see its error list, which uses the same error list window as the RSE does.

RSE and iSeries Projects

- Make WebSphere Development Studio Client a follow-on to PDM, SEU, SDA, RLU and System Debugger
- Make WebSphere Development Studio Client immediately useful and usable by existing PDM and SEU developers
- Offer a compelling “first step” in a modernization roadmap
- Are the new integration point for iSeries tool vendors

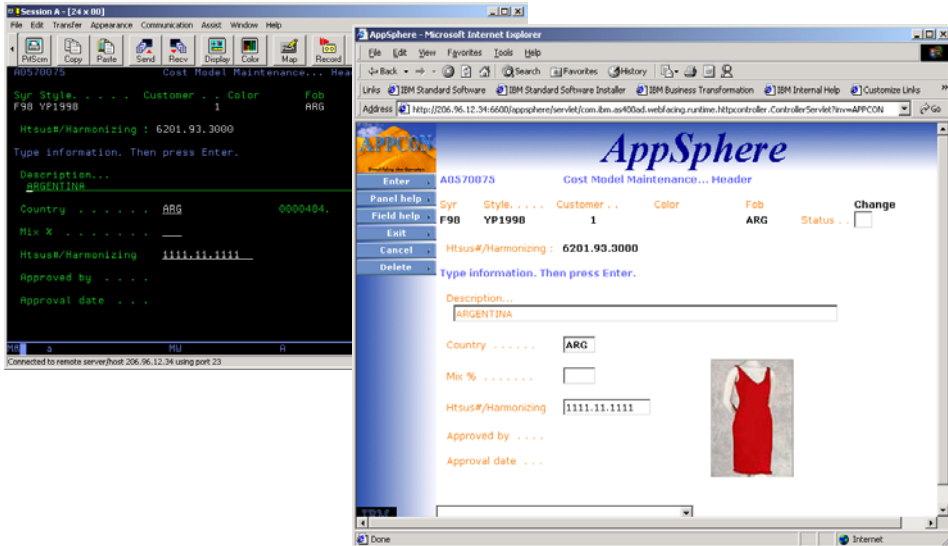
Drives home the point these tools are the follow-on to ADT.

WebSphere Development Studio Client

WebFacing
Tools

We'll now talk about WebFacing briefly.

WebFacing



This slide shows a sample "before and after" of a green screen that has been WebFaced. These screens are courtesy of APPCON, an IBM iSeries business partner. See more at www.appcon4.com.

As you can see, the role of WebFacing is turn green screens into Web pages...

WebFacing Tools

▪ WebFacing

- Given DSPF DDS input, **generates a Web application** that is still driven by the unchanged iSeries business logic.
- DDS and UIM-Help become JSPs, JavaBeans and a controller Servlet. All generated for you.
- Advanced-only capability
 - Option to use a **struts**-based framework for gen'd code
 - Option to generate **portal**-tolerant app via iPortal
 - Option to use **custom tags** in generated JSPs for easier customization

There are many options available now for transforming green screens to Web pages, but WebFacing is unique among them. WebFacing does this conversion at development, taking as input the display file DDS source, and generating JavaServer Pages for the output (ie, Web pages)

The JSPs generated by WebFacing are then deployed to a Web application server, such as WAS, and the application runs as is, but in a Web browser. To enable this to work, with no changes to the underlying application, there is a runtime intercept in OS/400, which detects when an application is running in WebFacing mode. When this happens, the screen data from the application is passed directly to the WebFacing runtime, instead of being used to generate a 5250 datastream.

So WebFacing absolutely is not a screen scraper. Indeed, it can't be because there is no 5250 datastream to scrape when running in WebFacing mode.

WebFacing is the result of business partners pleading for a conversion option that allows infinite finessing of the generated output, and is cost effective. WebFacing enables this because its output is editable at development time, and the runtime is a free part of OS/400 in V4R5 or higher.

WDSC 5.0 Enhancements

■ What's new in WebFacing for 5.0

- Automatic configuration of UTF-8 in WAS 5.0
- Web page compression when using WAS 5.0 (performance)
- Support for fn key buttons within windows (or anywhere)
- Now generate one xml file versus two java beans, per record
 - War file is much smaller, faster to create and deploy
 - Less memory used at runtime
- Many performance, scalability, usability and functional enh'ts
- Support for Struts in advanced only
- Support for working with, viewing and printing spool files, in advanced only (requires the struts support)
- Support for custom tags in advanced only (mainly for ISVs)

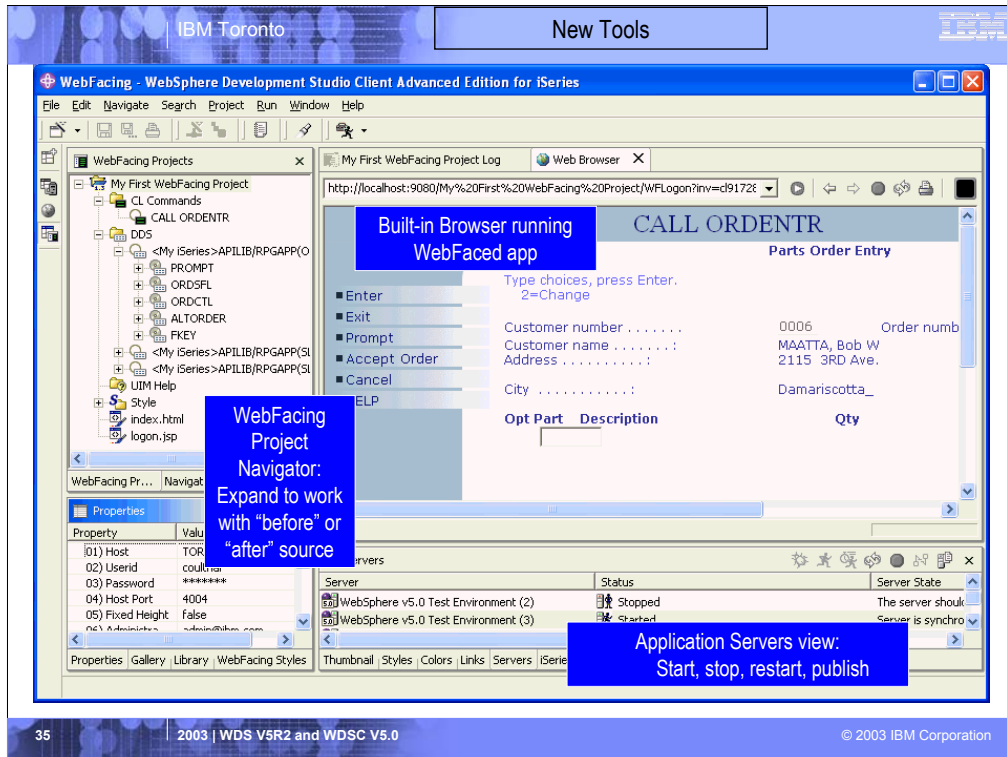
This is a partial list of the many enhancements for WebFacing in V5.0

WebFacing

■ WebFacing Scenario

- Create new WebFacing project
 - Specify members to convert, look-and-feel style, how to call/invoke the application
- Convert the project
 - Creates Web stuff from the DDS and UIM source
- Run it!
 - Select "Run on Server" to run in built-in WebSphere
- Refine it
 - Using Web Settings in CODE Designer (launched from WF)
 - Using project's Properties dialog
- Repeat
- Export as EAR file, import into WAS on iSeries
 - Or use built-in server-tools support to publish to WAS on i

Shows a typical usage scenario for WebFacing



This is the WebFacing perspective.

There is the project navigator on the left, which is the primary view driving the other views. It allows access to the DDS selected for conversion, and the output of the conversion. Using properties and actions, many facets of the conversion can be affected in the navigator. Also, the conversion itself is launched from the popup menu of the project (or an individual DDS member).

Notice the built-in web browser when running the converted app using the built-in copy of WebSphere, which is managed in the Servers view at the bottom.

WebSphere Development Studio Client

Java
Tools

We will now discuss briefly the Java tools in WDSC

Java Tools

■ Java Development Tools

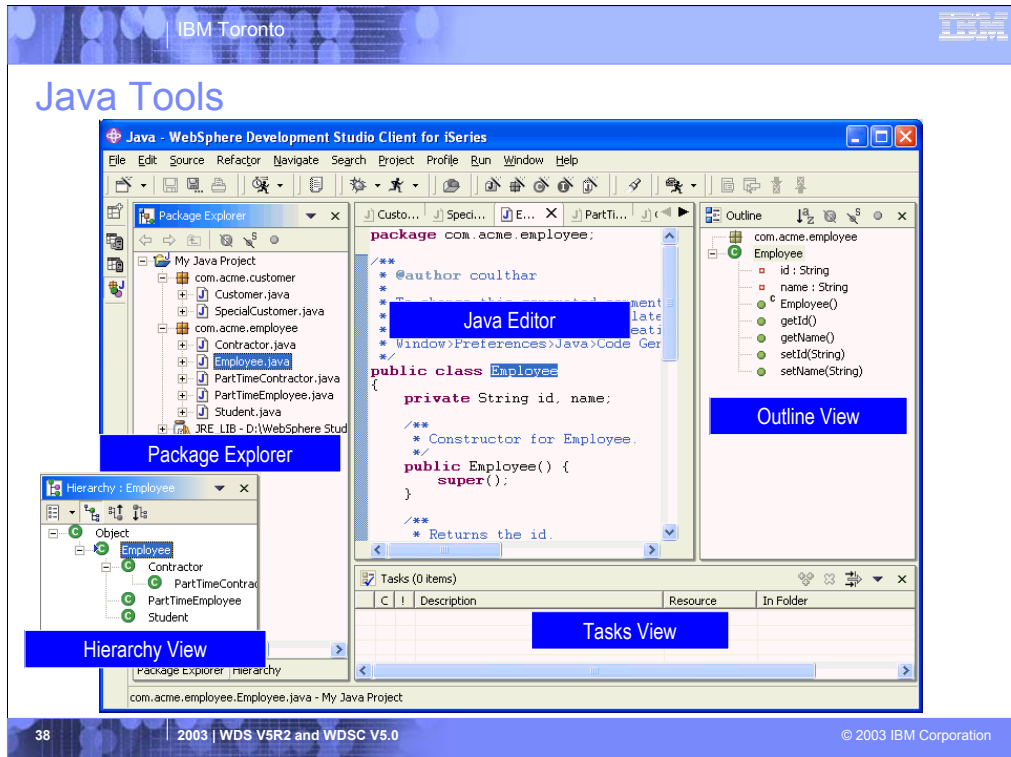
- Java Projects and Perspective
 - Optimized for Java development
 - Specialized views and editor
 - Package View and Hierarchy View
 - Java Editor
 - Support includes content assist and refactoring
 - Incremental compile on save
 - New this release: Visual Editor for Swing GUI
 - Outline View
 - Many tools launchable from here
 - A number of other views
 - Choice of Java level per project
 - IBM Java Runtime Environment 1.3 shipped as default
 - Also can set classpath per project

The Java tools in WebSphere Development Client come from the base Eclipse product. Java tools offer their own project type and perspective, optimized for ultra-productive Java development

There are many Java-unique views and actions available for you as you work on Java, within the Java perspective. These include:

- A special view to see all your Java packages
- A special view to see the class inheritance hierarchy
- An outline view that complements the editor and shows a bird's eye view of your Java class. From here many wizards are launchable
- A Java editor that is super optimized for Java. Its cool function includes the ultra-popular content assist from VisualAge, and new functionality that shows the JavaDoc for a method or class when the cursor hovers over a reference to them. There is also an F3 hot key that when used while a Java class name is selected, opens a new editor on that class
- There is also hot new refactoring support for rename. This finds and updates all the references to the old name! A huge time-saver.
- Of course, the ultra-popular incremental compile capability from VisualAge is still there too. There is no compile step, you simply save your changes as you edit, and the compile happens automatically. You can also force it via a popup menu on the project.

The 1.3 level of Java is supported by default, but as new releases of Java come out you do not need to wait for IBM to step up to them. The Java level is specifiable as a property so you can download the latest version and use the new Java tools to work with it!



This is the java perspective. It has a Package Explorer view that is the primary view driving the rest. The editor in the middle is awesome and full of amazing rich capabilities.

The command outline and task views are there as well.

The Hierarchy view shows the class hierarchy, either top down or bottom up, relatively the selected class in the Package Explorer.

Java Tools iSeries Extensions

▪ iSeries Java Tools

-Remote support

- Export to iSeries, Import from iSeries (IFS)
- Remote transform, run, debug

-Programming support

- Wizard to generate Java wrapper of *PGM or procedure
- iSeries Toolbox for Java built-in
- Java beans for iSeries access
- Documentation and samples

In addition to all the awesome base tools for Java, WebSphere Development Studio Client offers unique additional tools specifically for the iSeries Java developer. These are the new Eclipse-based versions of the tools that were previously known as Enterprise Toolkit for AS/400 (ET/400) in VisualAge for Java. These tools include:

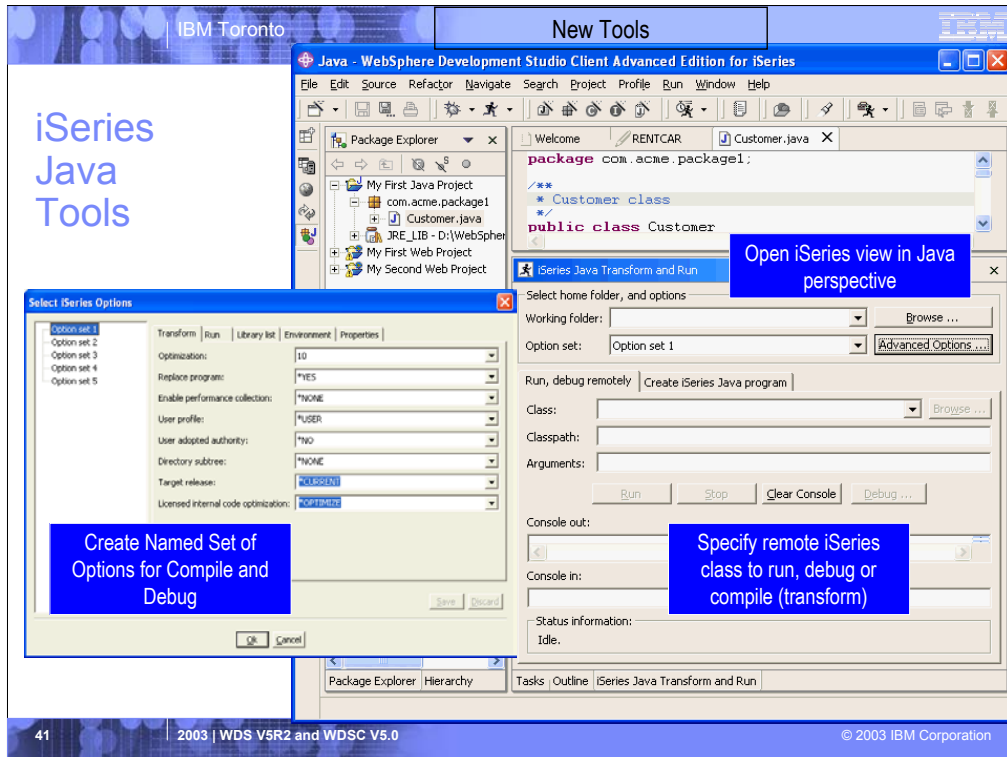
- Wizards to import Java source, classes and JAR files from the iSeries Integrated File System, into an Eclipse Java project.
- Wizards to export selected Java classes, packages and whole projects to an iSeries Integrated File System, either as source or compiled classes, and either as individual files or as a JAR file.
- The above import and export wizards are not only for iSeries IFS!! They can just as easily be used to import and export to and from any remote Unix, Windows or Linux system.
- A dedicated view for remotely compiling, running and debugging Java on iSeries, from within the WDS IDE
- A Program Call wizard to generate a Java class to call existing RPG or COBOL business logic
- A plugin that contains the full AS/400 Toolbox for Java, plus integrated help and samples
- Additional Java Beans to make it easy to write Java GUIs using your Display File skills, or to access iSeries data and file systems objects from Java

Java Tools

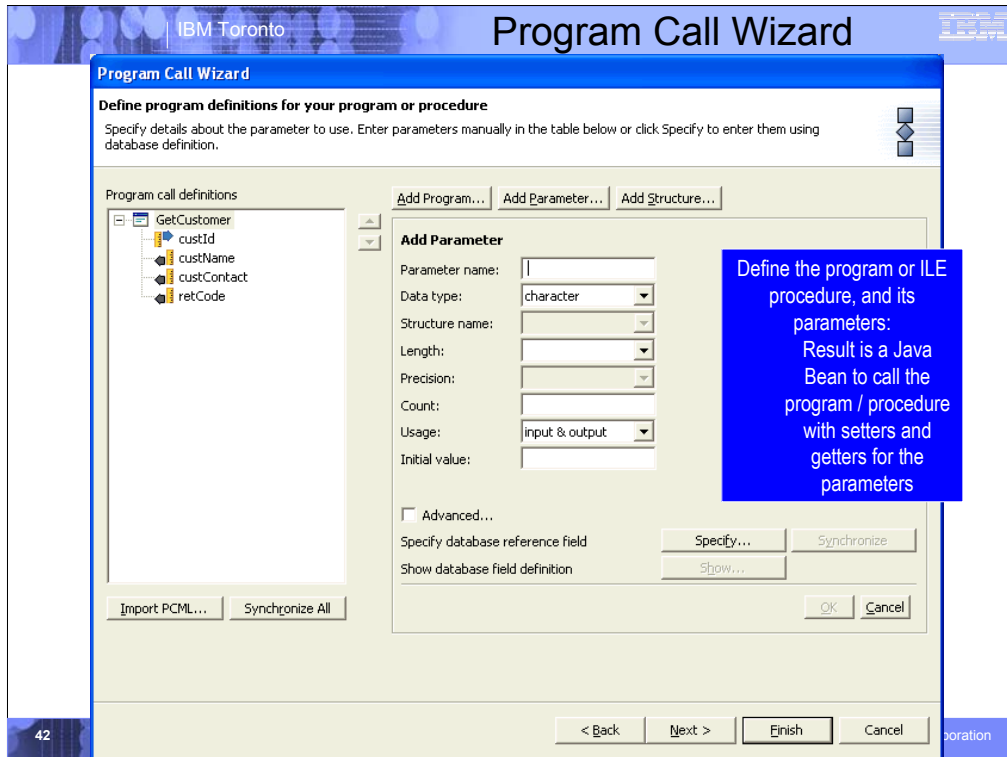
■ What's new in iSeries Java Tools for 5.0

- Eclipse 2.0 Java tools
- New Java Visual Editor for creating Swing UIs
- Launch configurations for running and debugging iSeries Java applications
 - Specify startup cmd+parms, and options
 - Once created, simply click to run/debug
- Updated to latest (V5R2) toolbox
- Option in Program Call wizard to generate to use new JCA-compliant connector. Advanced only

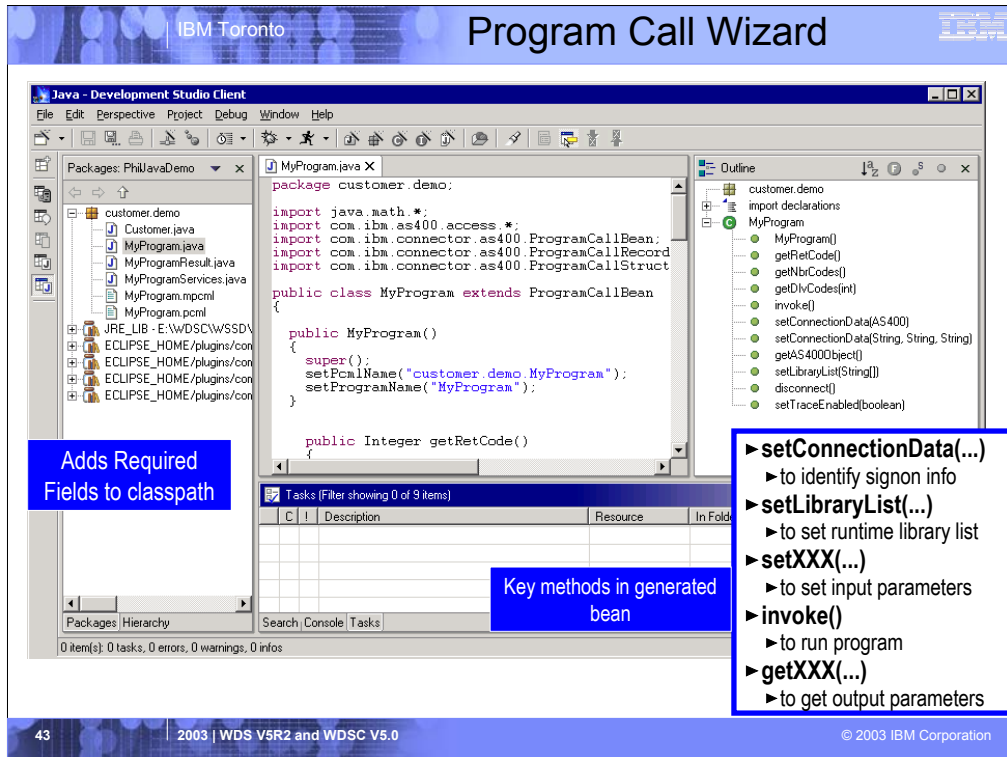
These are the enhancements to the Java tools in 5.0



This shows the Java perspective, but with the additional iSeries-specify view for remotely transforming, running and debugging Java on iSeries.



This is the wizard that, given the details of a *PGM or ILE procedure, will generate the Java bean to call it.



Here we see the result of the Program Call wizard.

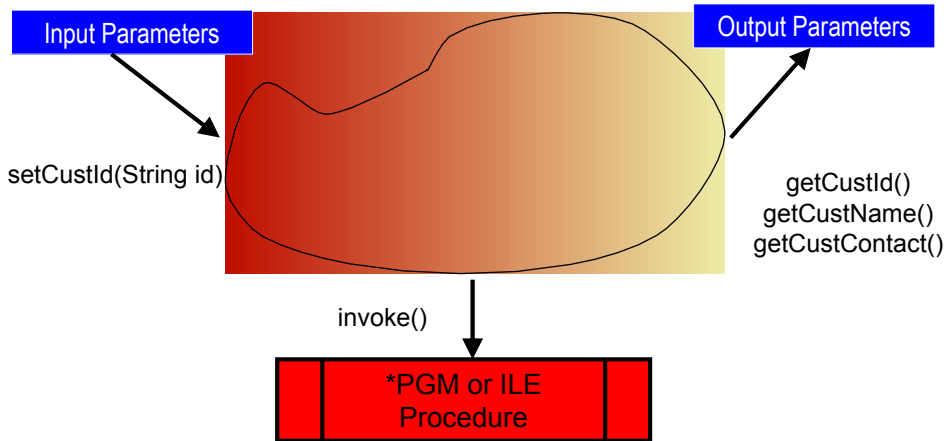
The selected Java package in the upper right now contains all the generated Java classes, one of which is shown in the editor in the middle view.

To use the generated Java bean, you write Java code to instantiate it and call the important methods in it, which are listed in the information box in the lower left. An example of Java code to use the bean is shown in the lower middle information box.

The pattern for using the generated bean is this:

1. Instantiate the bean using the New operator
2. Call the setLibraryList method with an array of library names to set up the library list
3. Call the setXXX methods to set the input parameter data. There will be one such method for each input parameter
4. Call the invoke method to call the program or procedure
5. Test the boolean result of invoke for true or false to determine if the invoke was successful
6. Call the getXXX methods to read the output parameter data that the program updated. There will be one such method for each output parameter

Program Call Wizard



This iSeries-unique function includes the program call wizard, that given a program or ILE procedure, will wrap it in a JavaBean. The JavaBean will contain one setXXX method for each input parameter, an invoke() method to actually call the program or procedure, and one getXXX method for each output parameter.

WebSphere Development Studio Client

Web
Tools

We will now discuss the web tools in wdsc.

Web Tools

■ Web Tools inherited from WSSD

- Web Projects and Perspective
 - J2EE-defined folder structure for Web Apps
 - Superset of Java projects (inherits Java Tools)
 - Automatic creation/maintenance of web.xml file
- Editor support
 - Create/Edit/Validate/Debug JSP and HTML files
 - Images and animation
 - Cascading Style Sheets (CSS)
- Import/Export from/to a variety of sources
 - HTTP/FTP/WAR/EAR
- Link viewing and management
 - Converting links, flagging broken links, and fixing up links as linked resources are moved or renamed

The Web tools in WebSphere Development Client replace the classic WebSphere Studio product. Web tools offer their own project type and perspective, optimized for ultra-productive Web development.

When a Web project is created, a folder structure that conforms to the J2EE standard is automatically created for you.

Web applications almost always contain Java code. As a result, Web projects are also Java projects, so all the great Java tools are also available for Web projects. You can switch back and forth between the Web perspective and the Java perspective as you work on different aspects of the project.

An important file for a Web application is the web.xml file that describes the servlets within the application. Using WSDc however, you need never worry about this file! It is created and maintained for you! However, should you wish to edit it there is a fantastic web.xml editor available for that.

There are also fantastic editors for creating JSP and HTML files, as well as tools for creating images and even logos. All Web sites today use cascading style sheets from the HTML 4.0 specification, and using these is easy with the editor support in the Web tools of WSDc.

To get started with your development, there are many ways to import existing Web applications or artifacts, and at the end of your work, there are many ways to export your Web applications to the file system for publishing and deployment

There is also fantastic support for visualizing all the interdependencies among the files in your Web application, and quickly seeing unresolved dependencies

Web Tools

▪ **WebSphere Studio Web Wizards**

- **Java Bean Web Pages Wizard**

Input: Java Bean

Output:

Input JSP page prompting for input properties

Output JSP page displaying output properties

Servlet that ties them together

- **Database Web Pages Wizard**

Input: SQL Query Statement (sub-wizard)

Output:

Input JSP page prompting for input parameters

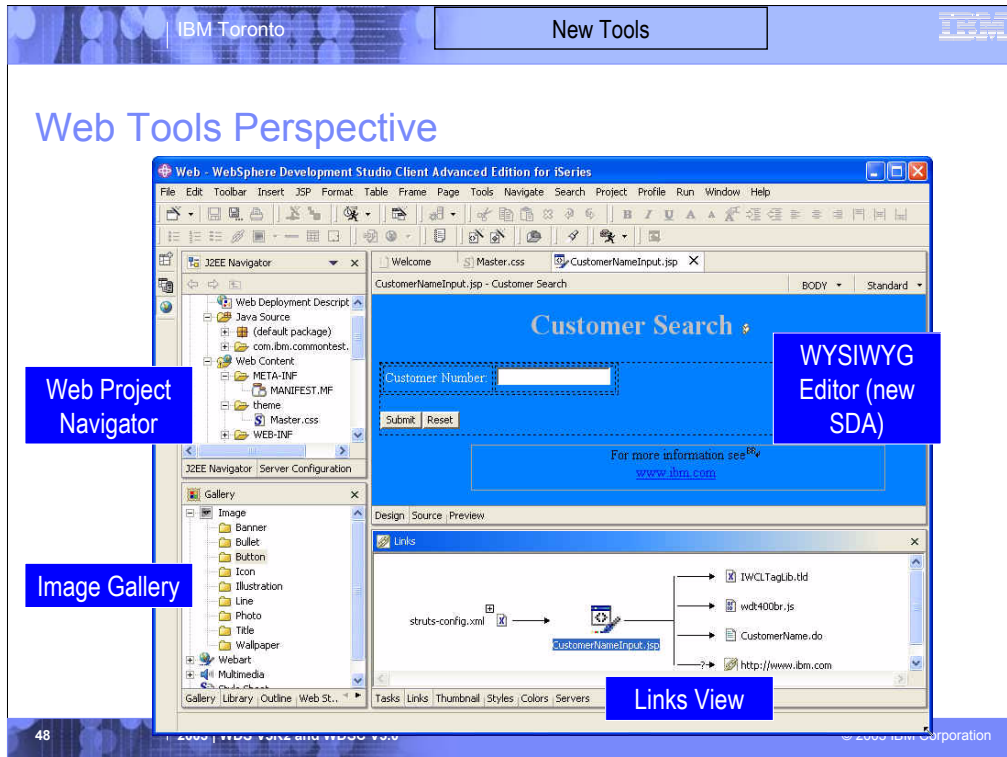
Output JSP page displaying output data

Servlet that ties them together

In addition to the many editors and views, the Web tools include a number of useful wizards to kick-start your Web development with minimal skills

The most exciting such wizard is the Java Bean Web Pages wizard. This wizard takes an existing Java Bean and generates an input Web page to prompt for the input properties of the bean, an output Web page to display the output properties of the bean, and a servlet that calls the selected methods of the Java bean when submit is pressed on the input page

Another exciting wizard is the Database Web Pages wizard. This wizard takes as input an SQL statement, which you can also generate via a sub-wizard, and generates an input Web page that prompts for the bound variables in the statement, and an output Web page that shows the result of the statement. It also generates a servlet that is called from the input page and runs the given SQL statement with the given input parameters



This shows the result of the Java Bean Web Pages wizard, and also highlights the Web perspective

The Web project in the upper right has been updated to include all the generated files.

The generated input page has been opened in the Page Designer editor. You can use this editor to finesse the generated pages, or any Web page. It supports a WYSIWYG mode (like SDA) and a source mode (like SEU) which is shown here. It also supports a preview mode that uses a built-in copy of Internet Explorer to show the page. You can toggle between the three modes using the buttons at the bottom of the editor.

While using the editor, there is a plethora of built-in tools to help with your Web development. These are available in the menu bar, the tool bar and the right-click popup menu.

In the bottom left of the Web Perspective is the Gallery view. This is a gallery of pre-supplied graphics, sound and other artifacts that you can easily drag and drop into your Web page to spice it up

In the bottom right of the Web Perspective is the Links view. This shows the relationships between the selected file, and other files in the project.

Web Tools: Run On Server

- **Run On Server**

- Now this is VERY COOL!

- When ready to test your Web app

- Right click on initial html or jsp file

- or whole project, which implies the index.html file

- Select "Run on Project"

- 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!!

- Tip: you can set breakpoints in your Java code!

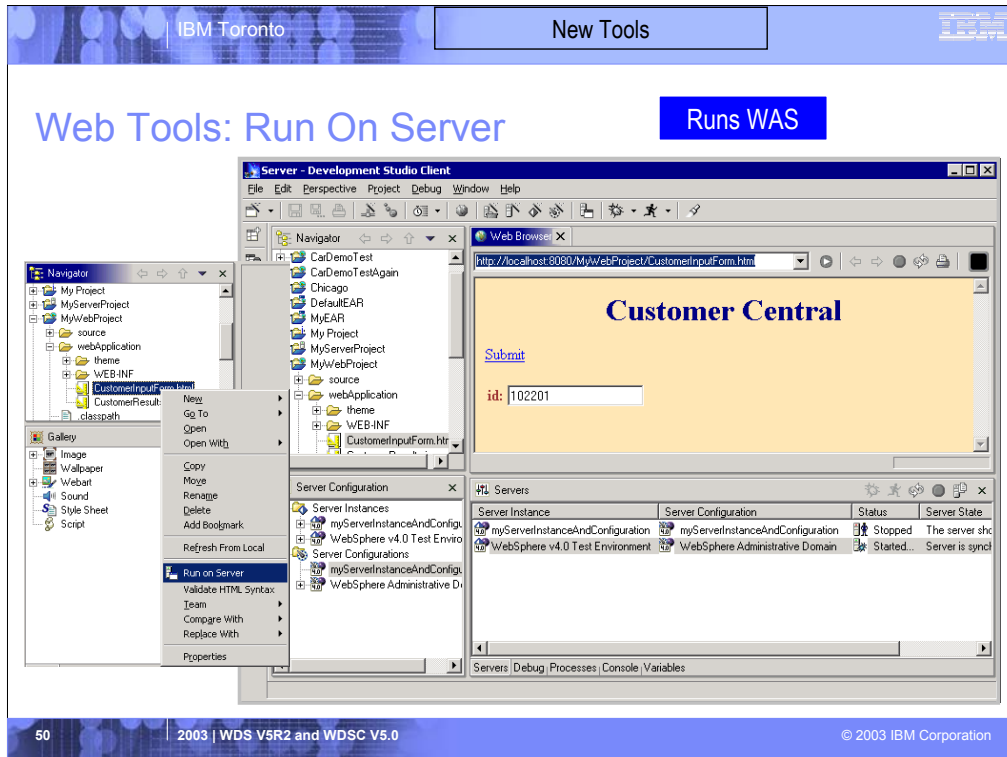
Having generated or hand-coded an exciting Web application, you will be impatient to see it running. Web applications that use servlets and JavaServer Pages (JSPs) cannot be "run" by simple preview in a browser, but rather require an application server to execute the servlets and JSPs.

Luckily and handily, an application server is built right into WDS so you can immediately run and see your beautiful work without wait. Of course, this built-in application server is in fact a full-fledged copy of WebSphere Application Server. In fact, there are multiple copies; WAS 4.0, WAS 5.0 Express and WAS 5.0

To see your Web application (that is, project) running, simply select the project, or a Web page file, and then right click and select "Run on Server".

This will start the built-in copy of WAS, and switch you to the Server view so you can manage the life cycle of it. Once WAS is started, your application will run and you will see it in a built-in copy of Internet Explorer!

If you desire, you can first set breakpoints, using the editor, within Java or JSP source. When a breakpoint is hit, you are presented with the Debug perspective

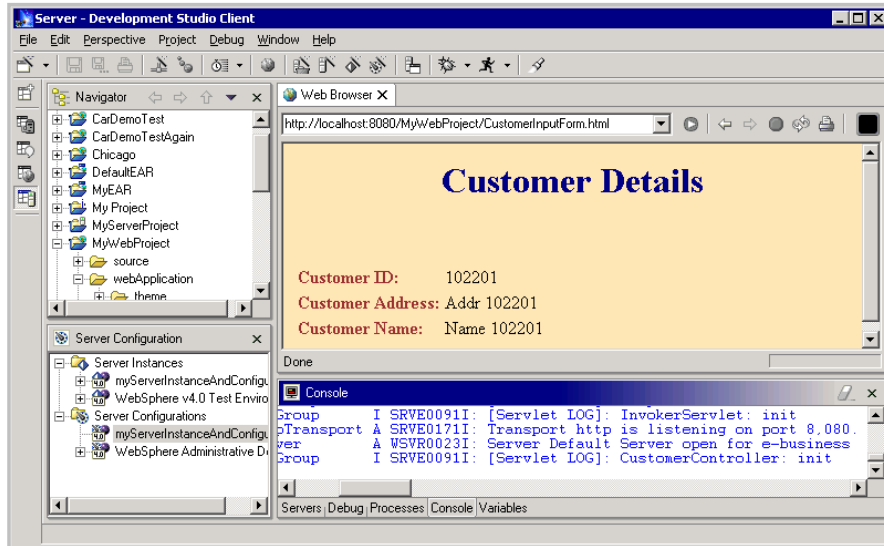


Here we see how easy it is to run your application.

Just right click, select "Run on Server", and presto, your application is running in the browser!

Tip: double click on the browser title bar to make it full screen, or press Ctrl+N to make it a separate window.

Web Tools: Run On Server



Here we see the result of pressing Submit on the input page of the previous slide.

Notice the output which indicates the servlet ran successfully. We also see many trace statements in the console window at the bottom, to help see what is going on as execution happens

WebSphere Test Environment

■ WebSphere Test Environment (WTE)

- A full copy of WebSphere Application Server is embedded in the IDE
 - Instant/easy testing of Web projects within Development Studio Client
 - Standalone all-in-one testing
 - No dependency on WAS installation or availability
 - Easily debug prior to publish/deploy
- Supports configuring multiple Web apps
- Access to WAS Administration Client

■ Server Tools

- Supports start, stop, restart, publish actions for WTE, local or remote WAS or local TomCat

When using "Run on Server" you are by default running the full fledged copy of WAS 4.0 that is built into WDSc

The purpose of this is to enable extremely easy testing of your Web application prior to publishing or deploying to your test or production server

The Server perspective that "Run on Server" switches you to is in fact quite rich in its functionality, allowing starting, stopping and publishing of the server via the right click popup menu. You can even bring up the full WAS Administration console directly from this perspective. Very handy!

In addition to managing the built-in copy of WAS, this server perspective can be configured to manage any local copy of WAS or TomCat, or any remote copy of WAS including WAS on iSeries

You can associate any Web project with any such local or remote WAS, and it will automatically be published and configured there for you when you "Run on Server" or explicitly "publish"!

There is simply nothing more productive for Web Development than the Server perspective. You cannot imagine living without it after using it.

Be aware though, that WAS requires at least 512MB of memory, so you will need this much to use this awesome functionality effectively.

WDSC 5.0 iSeries-Optimizations

■ iSeries Web Tools

-iSeries Web Components ... widgets for the Web

- DSPF-like capabilities like edit masks, validity checks
- Attributes inheritable from DB2/400 fields
- Rich table support, including page-at-a-time subfiling

-Web Interaction Wizard

- Creates Struts action from RPG/CBL logic (pgm/procedure)
- Given *PGM/*SRVPGM, generates I/O pages and glue
- Given *PGM/*SRVPGM and I/O pages, generates glue to bind page UI fields to *PGM/*SRVPGM parameters
- Given JavaBean, generates I/O pages and glues
- Given JavaBean and I/O pages, generates glue to bind page UI fields to JavaBean parameters

All the Web tools described so far are exactly as inherited from WSSD, and not unique to iSeries.

In addition to these, there are additional tools in WDS*c* that are in fact designed specifically for iSeries programmers, to maximize productivity and leverage their skills.

These iSeries-unique tools include:

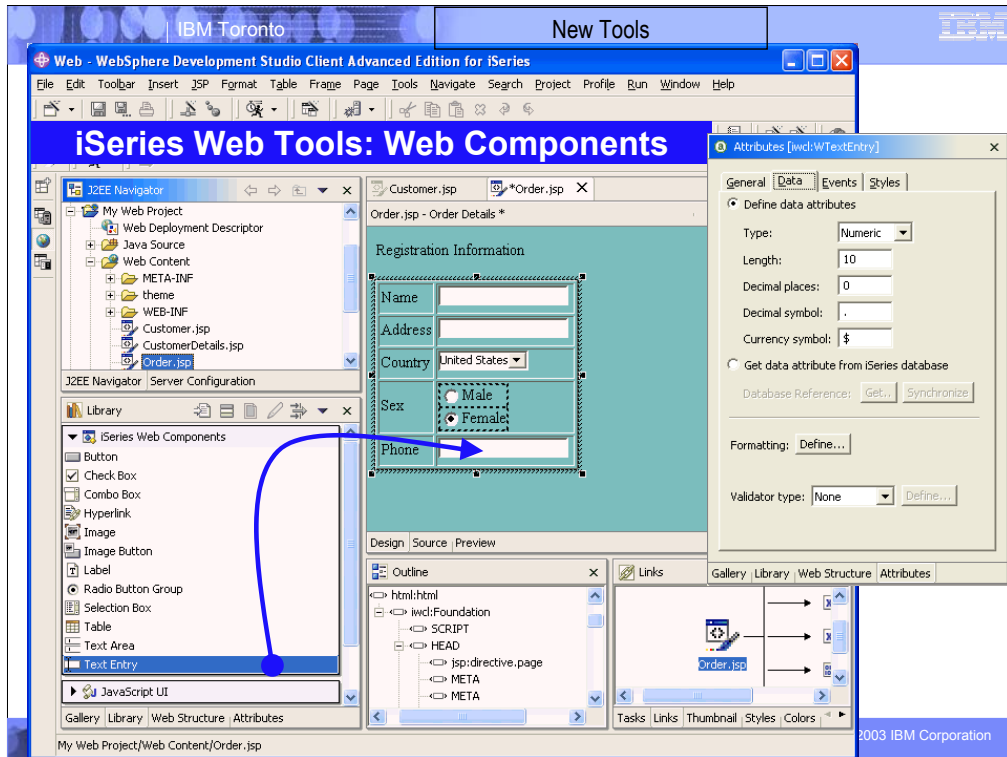
1. iSeries Web Components. These are Web Page widgets, that are available in the Page Designer tool. They can be inserted into a Web page, and generate HTML and JavaScript on your behalf, based on properties you specify. These properties are display file like, to leverage your skills, such as edit code.
2. An iSeries Web Interaction wizard. This wizard takes as input a program or ILE procedure, and generates as output an input Web page to prompt a user for the input parameters, and an output Web page to prompt the user for the output parameters. It also generates a Java Bean that calls the program or procedure when the user presses Submit on the input page. Finally, it generates a servlet that uses the Java Bean and ties all the pieces together.
3. An iSeries Host Information wizard. This wizard complements the Web Interaction wizard. It must be used to specify, per Web project, runtime information that all wizard-generated interactions use. This information includes signon information and library list setup.

WDSC 5.0 Enhancements

■ **What's new in iSeries Web Tools for 5.0**

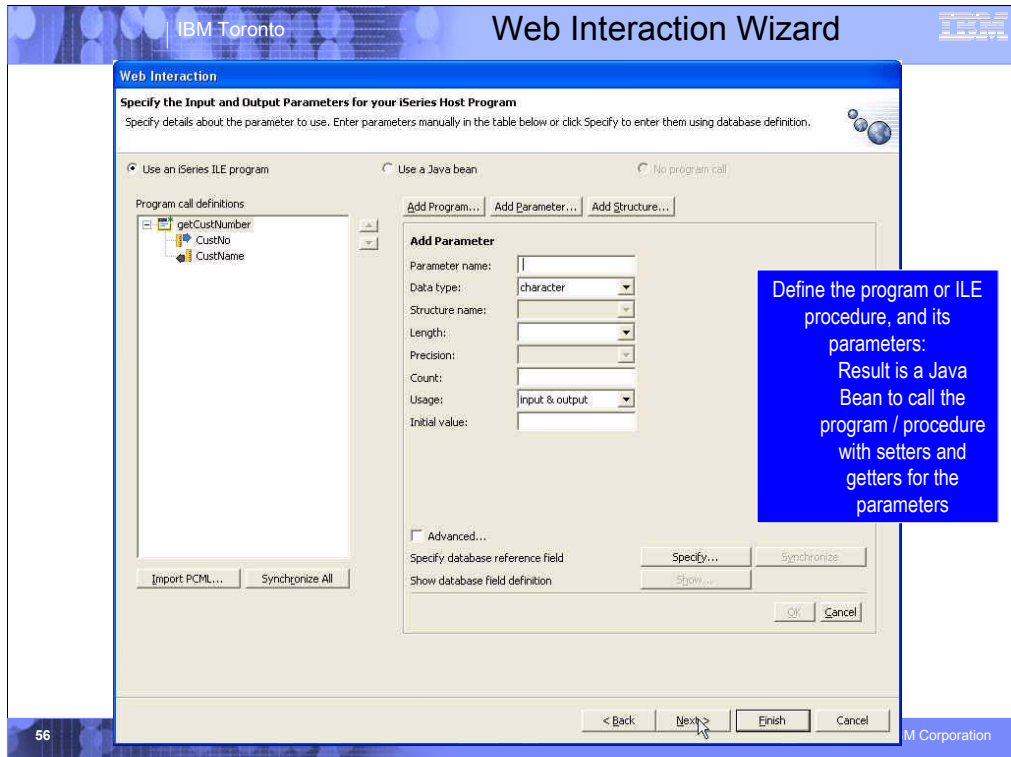
- Re-written Page Designer for WYSIWYG Web design
- V4.0 Design Time Controls replaced with Web Components
 - Uses JSP 1.2 custom tag technology
 - Based on Web widgets from Tivoli
- Supports for Struts in Interaction Wizard
 - Launched from new Web Diagram editor
- Many miscellaneous enhancements
- Advanced support for JCA connector

A partial list of the enhancements in V5.0 of the Web Tools.



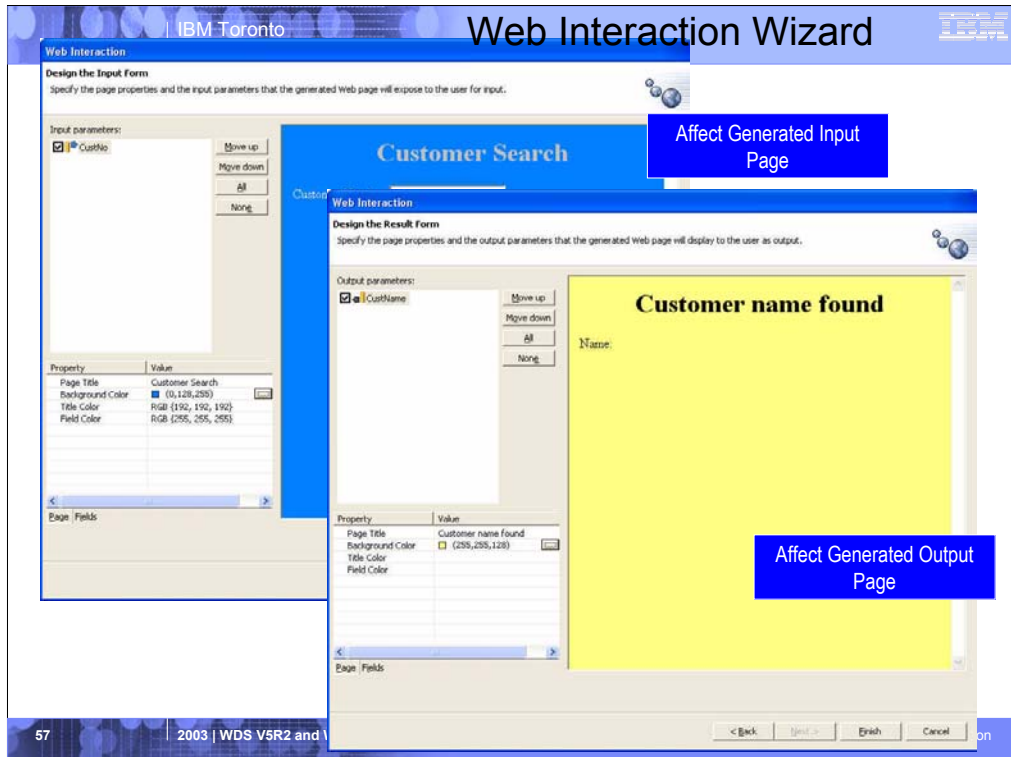
This shows the Web perspective. The upper left is the navigator that is the primary view that drives the others. It allows exploration of all Web projects. To the right is the Page Designer WYSIWYG editor, and to the lower left is the iSeries web components palette. Parts are dragged and dropped to the page, and then the Attributes view (behind the Library view but shown standalone) is used to configure the part.

There is also the usual outline view, an a Links view that shows all the relationships between the selected file and the other files in the project.

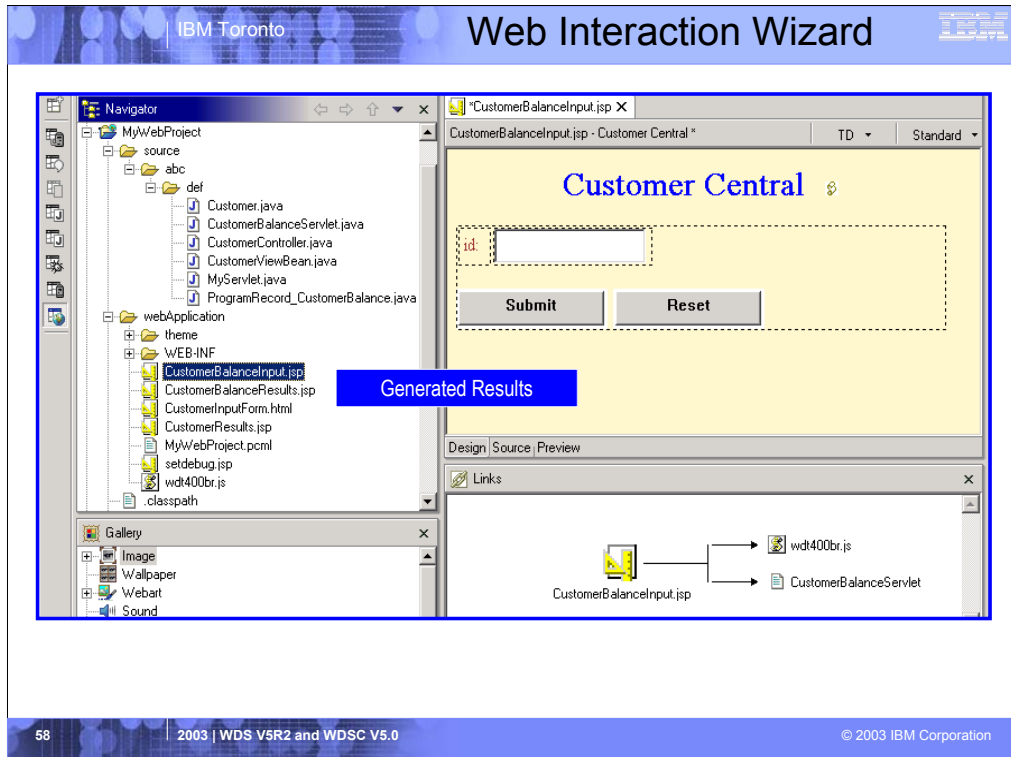


This is the Web Interaction wizard. This page of it is the same as the iSeries Java Program Call wizard, prompting for the program's name and parameter information.

But the next pages are unique...



Here we are prompted for the information to generate the input page prompting for the input parameters, and the output page showing the output parameters. Much of the generated pages can be affected by the property sheet in the lower left. Of course, once generated, they can be edited in the Page Designer.

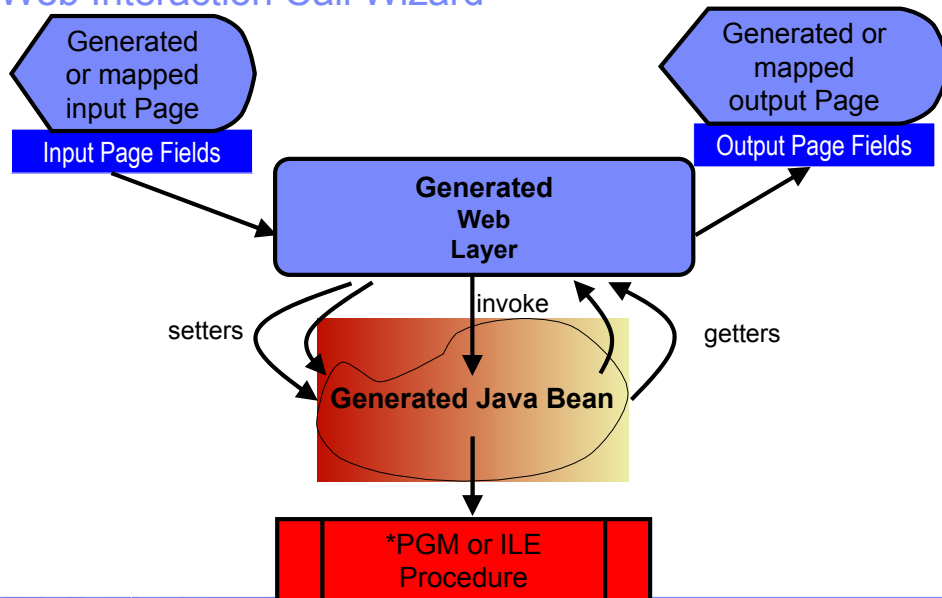


This shows the results of the Web Interaction wizard, in the Web perspective.

The Web Project in the upper right contains all the files that were generated by the wizard, and the generated input Web page is shown in the Page Designer editor.

At this point, everything we talked about so far applies now. For example, to run the generated interaction, simply right click on the input Web page file and select "Run on Server"!

Web Interaction Call Wizard



On top of the base Web tools, WDS adds Web tools unique to iSeries programmers, similar to the old iSeries Extension that were in WebSphere Studio in WDT.

This iSeries-unique function includes a wizard, that given a program or ILE procedure, will wrap it in a JavaBean and will generate an input and output Web page, and a servlet to tie it all together.

The input Web page will prompt the user for the input parameter, and the output page will display the output parameters. When Submit is pressed on the first page, the generated servlet is called, which reads the user-entered data, and then passes it to the Java bean, which in turn calls the program or procedure. The output parameters are then fed to the output page.

A single input and output page pair is called a Web Interaction.

WebSphere Development Studio Client

Web Services Tools

Now we'll discuss Web Services tooling.

Web Services Tools

▪ Consume/Construct Web services:

Discover

- Browse UDDI registry to locate existing services
- Generate Java Bean Proxy for existing services

Create/Transform

- Create new Web services from Java Beans, databases
- Including Java Beans output from iSeries ProgramCall wizard

Deploy

- Deploy Web service to WebSphere or Tomcat for testing

Test

- Built-in test client allows for immediate testing of local/remote Web services

Publish

- Publish Web services to UDDI registry

WebSphere Development Studio Client inherits a very rich set of Web Services tools from WSSD:

1. Built-in access to the IBM test or official UDDI (Universal Description and Discovery Interface) registry, which you can browse to select Web Services and build Java bean proxies from them. These Java bean proxies contain a method to call the service, passing in the required parameters and returning the results.
2. Tools to generate a Web Service from a Java Bean or database SQL statement. By using a Java bean that is generated by the iSeries program call wizard, you can easily expose RPG or COBOL logic as a Web Service!
3. Tools to deploy a Web Service to WAS (including the built-in copy) or TomCat to test it.
4. To test a Web Service you need Java code to call it, and preferably a Web page to prompt for the input parameters and display the output. All this can be generated for you! Awesome!
5. Once you have generated a Web Service, deployed it WAS or TomCat and tested it there, you can optionally publish it to the IBM test or official UDDI registry for all the world to use!!

Web Services Tools Enhancements

■ What's new in WebServices Tools for 5.0

- Support for **UDDI V2** and discovery through Web Service Inspection Language (**WSIL**)
- Support for Simple Object Access Protocol (**SOAP**), a standard for transporting data between business applications
- **Samples** enhanced with pop-up actions

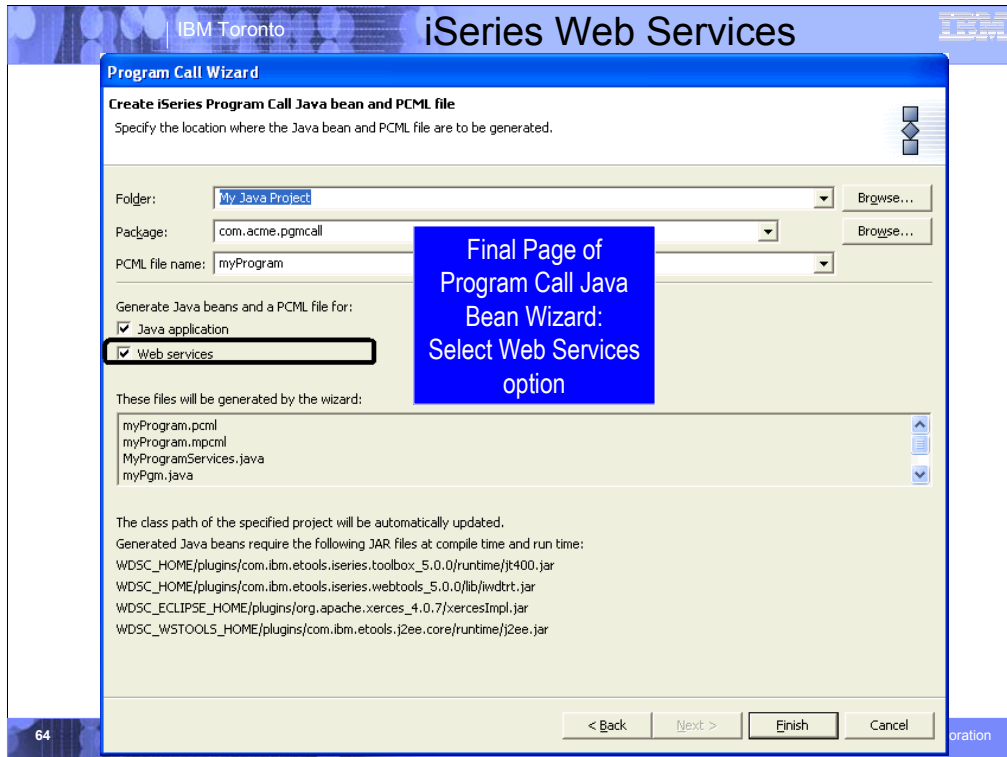
Shows what is new in the Web Services tooling.
These are all inherited from WSSD.

Web Services Tools

▪ iSeries Web Services Tools

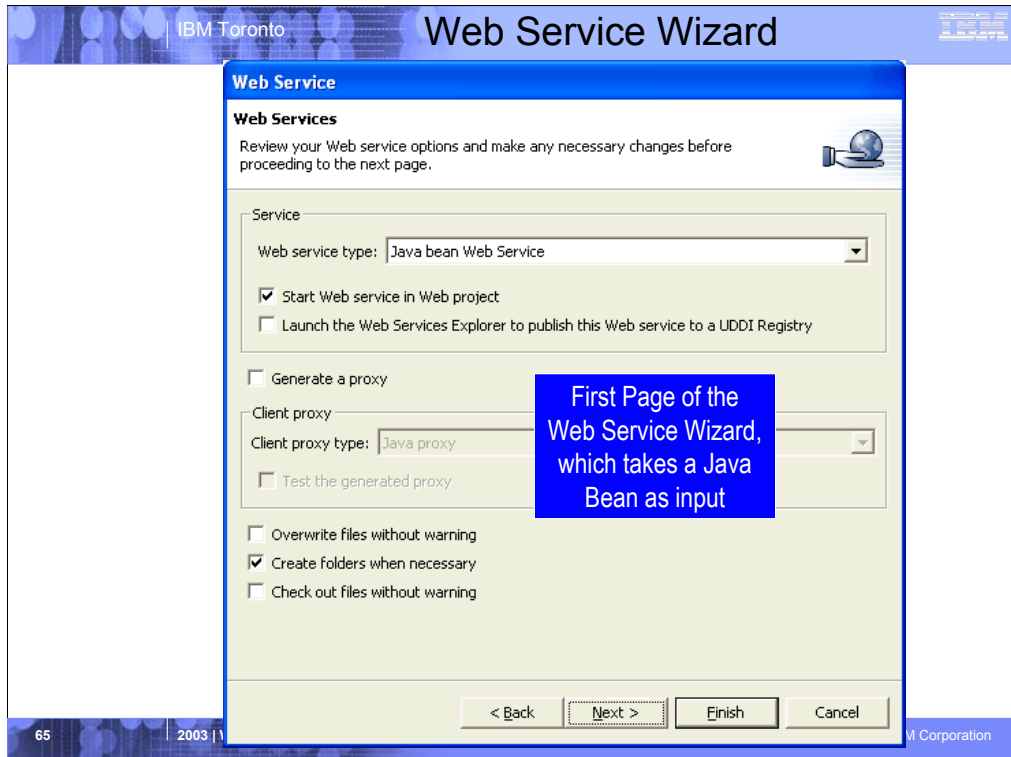
- The Program Call wizard in the iSeries Java Tools has an option to generate a **Java Bean wrapper** of a *PGM or ILE Procedure that is designed for use in the Web Services Java Bean wizard.
- The resulting Java Bean is fed into the Web Services Java Bean wizard, resulting in a **Web Service** that interface to RPG or COBOL business logic.

Describes what is added to the inherited Web Services tooling, uniquely for iSeries.



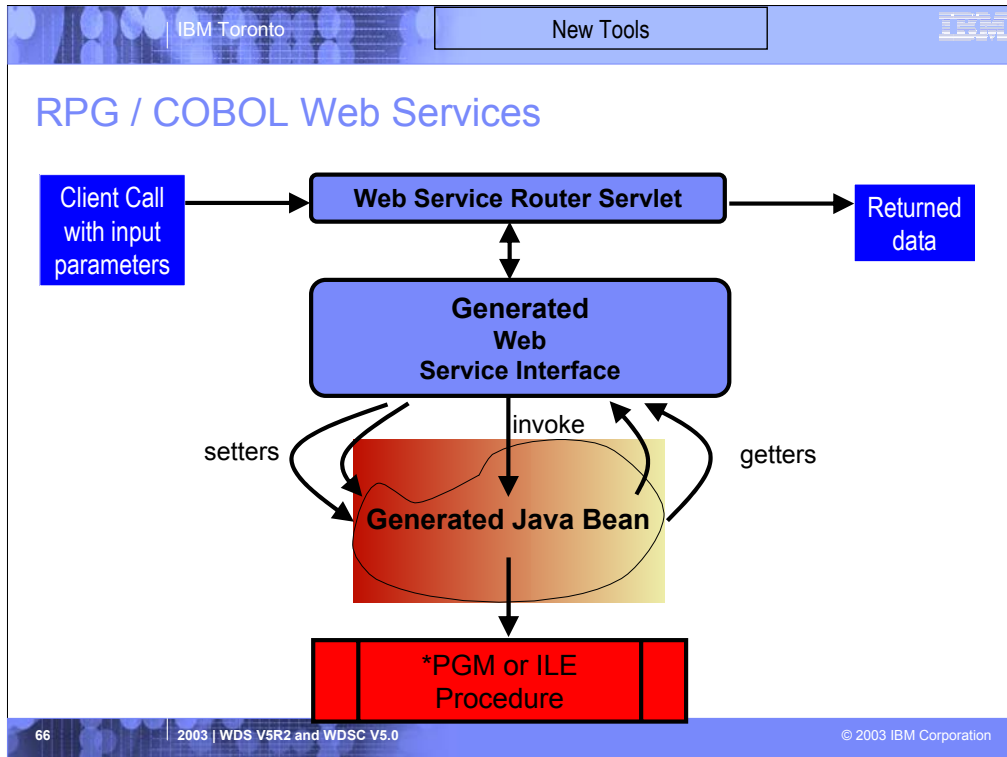
This shows the final page of the iSeries Java Program Call wizard, with the option to generate the Java Bean for the purpose of using it in a Web Service.

The next step after this is the use that Java Bean as the input to the Web Services wizard.



This shows the first page of the Web Service wizard, which creates a Web service from a Java Bean.

This is the wizard into which you would feed the Java Bean generated from the iSeries Program Call wizard



On top of the base Web Services tools, WDSi adds iSeries-unique support.

Since the Web Services tools take a JavaBean as input, the Java tools Program Call wizard can be used to generate a JavaBean wrapper to an existing program or ILE procedure. By selecting the Web Services option in the wizard, the JavaBean will be generated ready to be consumed by the Web Services wizard. This makes it easy to turn an existing non-interactive program or procedure into a Web Service.

The result is a Web Service that has a Java client API that takes the same input parameters, as the program or procedure, with the data types automatically mapped between Java and RPG/COBOL.

The output parameters are turned into a Web Services response, either as a Java data type or as XML data. Both options are supported.

WebSphere Development Studio Client

XML
Tools

We now introduce the XML tools in WDSC

XML Tools

- **XML Editor**
For creating and viewing XML files
- **DTD Editor**
For creating/viewing Document Type Definitions
- **XML Schema Editor**
For creating, viewing and editing XML Schemas
- **XML to XML Mapping Editor**
To map one+ source XML files to a target XML file
- **XSL Trace Editor**
To visually step through an XSL transform
- **XML and SQL Query Wizard**
To create an XML file from an SQL query
- **RDB to XML Mapping Editor**
To map one+ relational tables to a target XML file

This is the full list of XML tools available in WDS. As you can see, it is extremely rich.

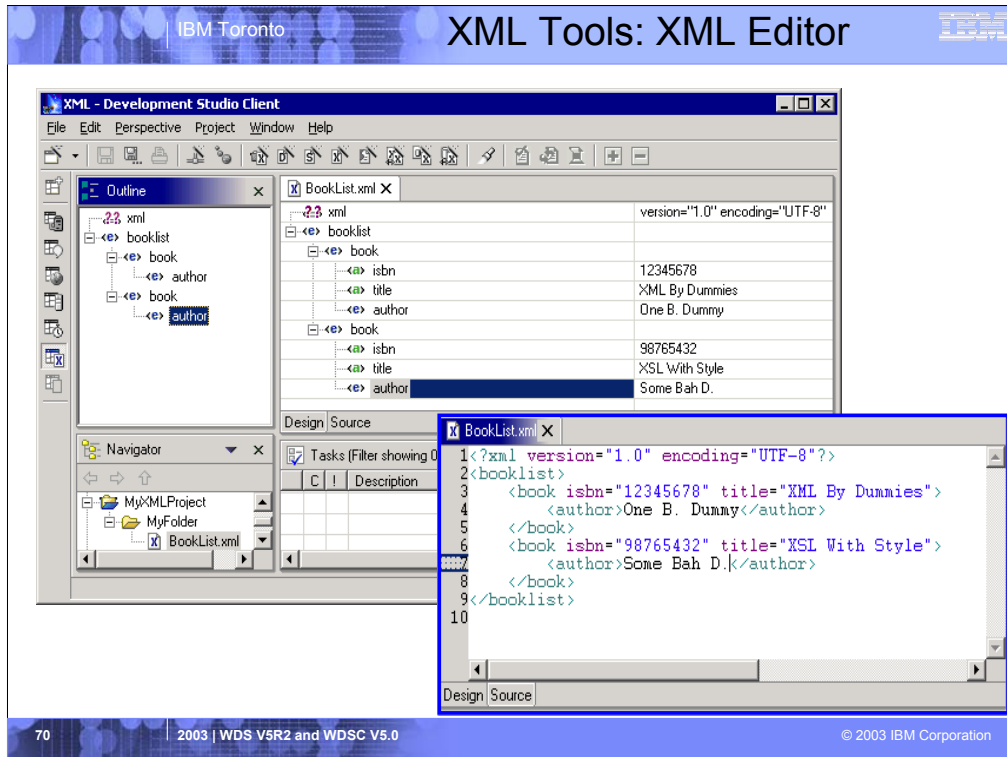
1. An editor for creating and visualizing XML files. This editor includes a wizard to generate a Java bean to parse and generate the XML.
2. An editor for creating Document Type Definitions, with minimal DTD skills. DTDs can be deduced from sample XML files. This editor includes a wizard to generate a Java bean to parse and generate any XML conforming to the DTD.
3. An editor for creating XML Schemas, which are replacing the older DTDs, with minimal XML Schema skills. Schemas can be deduced from sample XML files or DTDs. This editor includes a wizard to generate a Java bean to parse and generate any XML conforming to the schema.
4. A mapper tool that takes two XML DTDs or Schemas and allows you to map the tags and attributes from one to the other. The result is an XSLT file, that when run in an XSLT engine will map an XML file to an output XML file. XSLT is a standard language for defining XML mappings, and there are many engines that can "run" XSLT. Such an engine is supplied in WDS and WAS.
5. An XSLT trace editor. This will "run" and XSLT and map one XML file to another. You can single-step through the transformation and see the output being generated
6. A wizard that will generate an XML file from an SQL query. This can be a static one-time only operation, or you can generate a Java bean that will do this at runtime.
7. A wizard that maps database tables to target XML tags and attributes. This then generates code that turns queries into XML and turns XML into database updates.

XML Tools Enhancements

▪ What's new in XML Tools for 5.0

- Generate Java **Beans from XML** documents and vice-versa
- Use the Xalan processor to create HTML and XML documents with the **XSL Stylesheet**
- Create and execute an XPath with the **XPath wizard**
- Generate **document access definition scripts** for use with IBM DB2
- Create XML documents from **DB2 data** and vice-versa
- Validation for use with an **iSeries host**

The are the 5.0 enhancements to the XML tools.



Here we see the XML editor. The outline of the XML file is shown on the left. The right is the XML editor that allows editing of the XML data, either in design mode (shown) or source mode (shown at bottom).

WebSphere Development Studio Client

Database
Tools

We'll now briefly discuss DB tools

Database Tools

- **Relational Schema Center**

- Provides views geared for DBAs to
 - Create Databases
 - Create Tables/Views/Indexes/Keys
 - Generate DDL
- Online and offline support for working with databases
 - Metadata generated as XML

- **SQL Query Builder**

- Visually construct SQL statements
 - Insert, update, delete, select supported
 - Metadata generated as XML

- **SQL/XML mapping**

WebSphere Development Studio Client inherits a modest (stay tuned!) set of Database tools from WSSD:

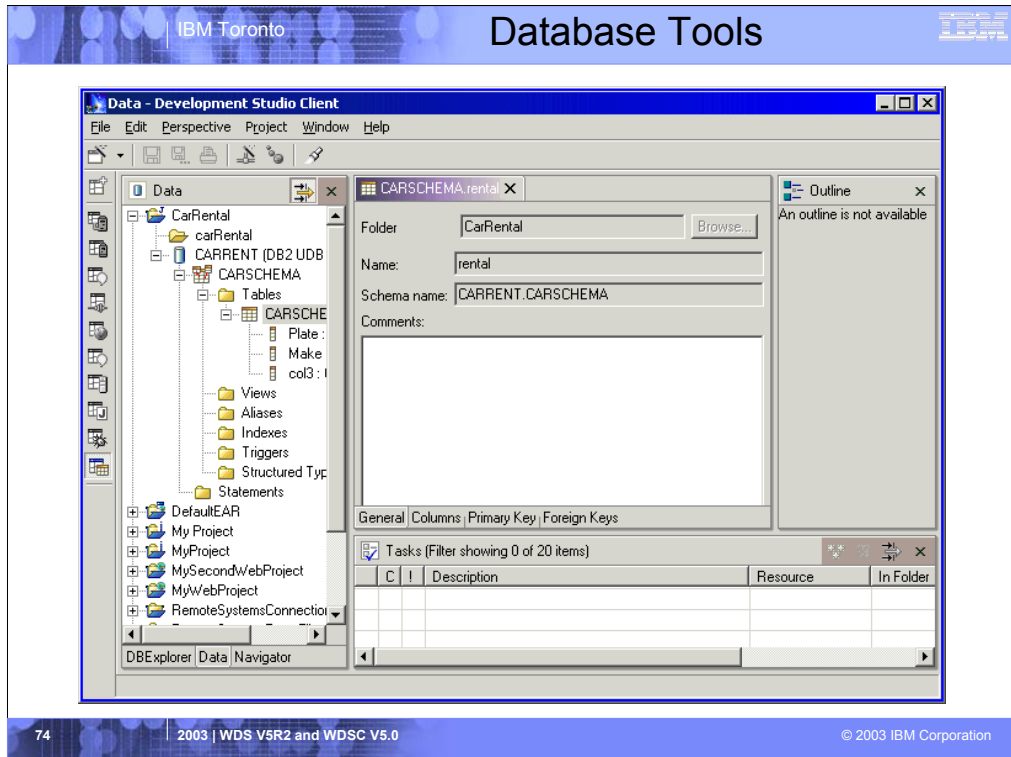
1. A perspective for working with database definitions. You can create/import database definitions, including tables, views, indices and keys. You can continue to work on the definition until complete, then generate SQL Data Definition Language (DDL) statements to create a database conforming to the definition. This allows you to work on the database definition offline, then connect and upload it to the server.
2. A wizard for building SQL Data Manipulation Language (DML) statements or query statements, with minimal SQL skills
3. A SQL to XML mapping wizard, already described as part of the XML tools, but also available in the data perspective

Database Tools Enhancements

■ What's new in DB Tools for 5.0

- A meta-data model that enables tools needing relational database information to share connections and function independently
- A visual interface for creating databases, tables, and views
- Support / better-support for DB2, DB2 for iSeries and for 390, Oracle, Oracle9i, Sybase, My1SQL, SQL server, and Cloudspace
- Validation for use with an iSeries host

The are the 5.0 enhancements to the DB tools.



For Database development, there is a dedicated perspective containing views and editors optimized for the task of Database design.

These tools include the Relational Schema Center shown here, which makes it possible to work with a database design even not connected to a remote server which the database will ultimately reside.

WebSphere Development Studio Client

Summary



✓ Community



✓ Excitement