

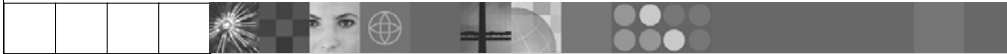


IBM Software Group

iSeries AD Strategy and WebSphere Development Studio Introduction

iSeries Application Development Team, IBM Toronto
Phil Coulthard: coulthar@ca.ibm.com

WebSphere. software



2003 | WDS V5R2 and WDSC 5.1

© 2003 IBM Corporation

This presentation reviews the tools and features of IBM WebSphere Development Studio Client for iSeries (Development Studio Client is the short form). It is a level setting presentation. Its purpose is two fold. First to describe the strategy and technology that forms the foundation of Development Studio Client. Second to overview the tools included in Development Studio Client.

Table of contents

□ □ **Strategy**

WebSphere Development Studio Client 5.1

Summary

This presentation covers first the strategy of application development tools and is then followed by a review of the tools included in WebSphere Development Studio Client Version 5.0 and 5.1

Strategy

- Vision
- Packaging
- Eclipse
- WebSphere Studio Family

The first section of this presentation covers the iSeries Application Development Strategy. The strategy covers four areas: new tools, new packaging, new technology and new user interface.

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 Application Development team has been working towards a strategy to re-energize application development on iSeries, returning the iSeries to its historical position as the world's most productive operating system for application development.

The IBM iSeries application development 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 application development 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.

iSeries Application Development Strategy (cont'd)

New Packaging

WebSphere Development Studio

New Technology

Eclipse, Web, XML, WebServices, etc

New Tools

WebSphere Development Studio Client


New User Interface

Web

Here we see IBM's strategy is to reach its goals of building community and building excitement through 4 areas: New Packaging, New Technology, New Tools and New User Interface. 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 tools 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 tools, 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 Software Group | WebSphere software 

www.ibm.com/software/awdtools/wds400

WebSphere Development Studio V5

Unlimited Licenses

RPG

COBOL

C/C++

**PDM, SEU
SDA, RLU**

V5R1 or V5R2
 5722-WDS
 No-cost Upgrade: 5905
 (V5R1) or 5913 (V5R2)

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

WebSphere Development Studio Client V5

+CODE
 +VisualAge RPG

iSeries Application Development Strategy | 2003 © 2003 IBM Corporation

There is now only one application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), 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).

If you are an existing customer who has a subscription, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio 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 you have RPG at V5R1 or higher, you must have Development Studio and hence are entitled to Development Studio Client.

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

Development Studio 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 Application Development tools from IBM. This ubiquity is especially important for business partners who build and sell software. These Business Partners are now free to build software using any of the technologies or tools in Development Studio, 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.



www.ibm.com/software/awdtools/wds400

WebSphere Development Studio Advanced V5

NEW!

C/C++

PDM, SEU
SDA, RLU

Unlimited Licenses

V5R1 or V5R2
5722-WDS
Cost Upgrade 5906
(V5R1) or 5914 (V5R2)

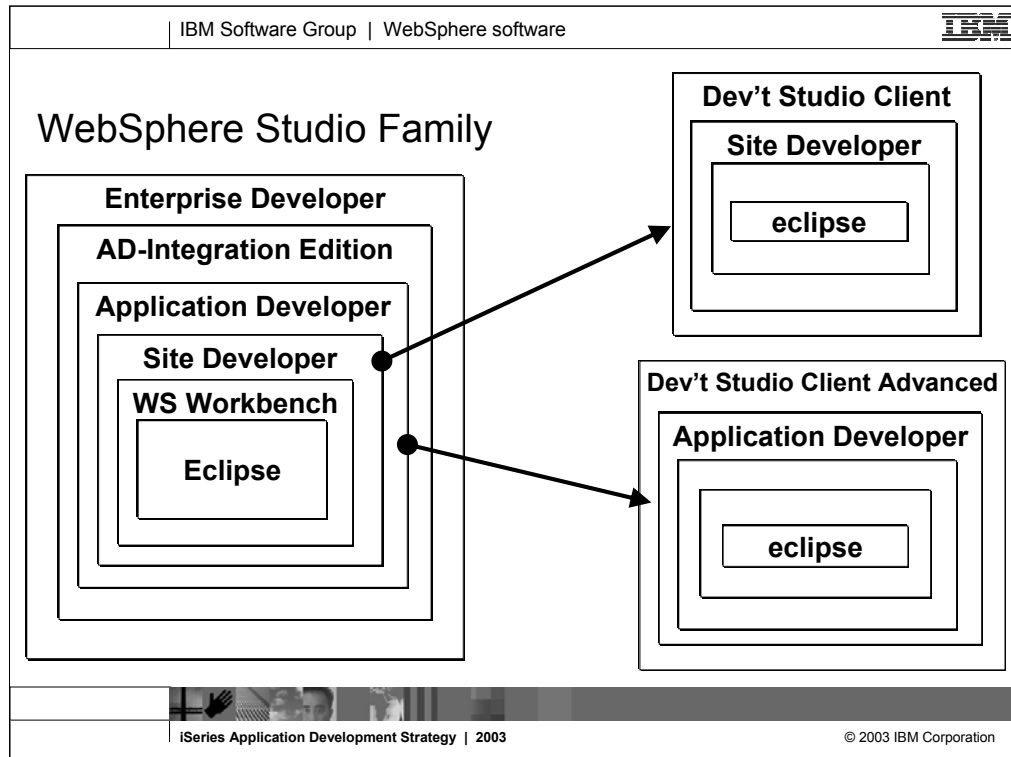
iSeries	iSeries	iSeries*	iSeries	Web Facing*	iSeries Projects
Java	Debug	Struts Web	Web Service		RSE
Trace	Profiling	DB	XML	App Server	EJB* J2EE*
					Test Cases*

WebSphere Development Studio Client Advanced V5

+CODE
+VisualAge RPG

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

The difference of Development Studio Client Advanced over Development Studio Client is that it has additional tools (blue boxes with asterisks) and some enhancements to existing iSeries tools (green boxes with asterisks) which will be discussed later in this presentation. The majority of the new tools are related to Enterprise Java Bean (EJB) development. You'll see later that Development Studio Client Advanced is based on WebSphere Application Developer (Application Developer) versus WebSphere Studio Site Developer (Site Developer).



Here you see that Development Studio Client is based on WebSphere Studio Site Developer, while Development Studio Client Advanced is based on WebSphere Studio Application Developer.

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.

Site Developer is IBM's entry level offering based on eclipse, and it is for building dynamic Web sites out of non-EJB Java. Application Developer extends Site Developer and adds support for EJBs. Application Developer-Integration Edition extends Application Developer and adds support for JCA Connectors and for Workflow. Enterprise Developer extends Application Developer-Integration Edition and adds support for S/390 and Enterprise Generation Language (EGL), the follow-on to VisualAge Generator.

WebSphere Development Studio Client 5.1

- **New release**, generally available Oct 17th, 2003
 - Focus on performance and quality
 - Based on new Eclipse 2.1 release
 - Supports WebSphere 5.0.2
 - New Web Site Designer support
 - Numerous enhancements to all tools

- Previous releases:
 - 5.0 in April 2003
 - 4.0 in July 2002
 - WebSphere Development Tools V5R1 in May 2001 (pre-Eclipse)



A new release of WDS is now available, only a few months after the 5.0 release. Since it was a short release cycle, it focused on performance, quality and high-demand customer requirements. Furthermore, it adopted the latest version of eclipse, and supports the latest release of WebSphere Application Server.

Eclipse – Java Toolset

eclipse.org

- A base **I**ntegrated **D**evelopment **E**nvironment (IDE)
 - Comes with built-in rich Java tools
 - Extensible via plug-ins
- Used as basis of products
 - Product = Eclipse + plug-ins
- Open source
 - Contributed by IBM, managed by consortium
- Popular!
 - 12 million user downloads
 - Dozens of product offerings powered by Eclipse
 - 600 open-source or freeware plugin projects
 - 40 companies in consortium
 - Hundreds of companies writing plug-ins

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 tools 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 integrated development environment (IDE) that has built-in support for teams and projects and a robust and revolutionary user interface framework. It also has tools built-in to 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 that MKS is an iSeries tool vendor. See www.eclipse.org for the latest list.

Eclipse – core development platform

- **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 plug-ins (for example, tools) that do something interesting.

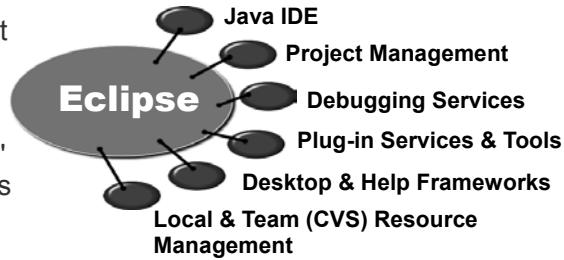
An Eclipse-based product is Eclipse plus a number of interesting plug-ins. IBM is building numerous such Eclipse-based products, including as we have seen Site Developer and Development Studio Client. Further, because Eclipse is free and business partners are free to include it in their products, there 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 application development tools, Eclipse is a fantastic opportunity. By writing plug-ins for Eclipse, those plug-ins 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 plug-ins for their tools. This will result in a rich offering of third party plug-ins for developers to choose from, all of which extend their core Development Studio Client development environment. One community, one core development platform, many IBM and 3rd party tools. This is community and excitement!

Eclipse Plug-in Development



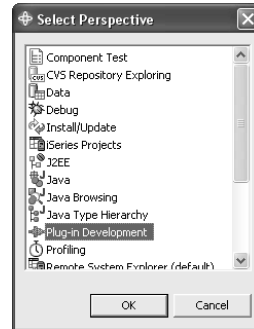
Products built with Eclipse inherit these capabilities plus 'plug-ins' built by others



Some of the common services Eclipse supplies include: the Java integrated development environment (IDE), project management, debug, plug-in frameworks, desktop and help frameworks, and resource managers.

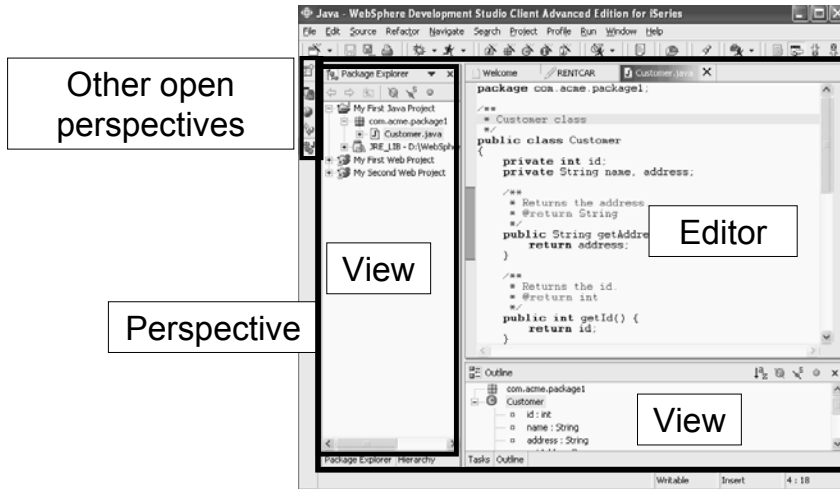
Eclipse User Interface – Select Perspective

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




The core features of the Eclipse user interface include perspectives which is a collection of views and tools. Perspectives allow role based development. For example, if you are a Java developer you would use the Java perspective which includes tools and views for Java development. You can also create your own perspectives. Naturally, the Eclipse user interface applies to all Eclipse-based products, like Development Studio Client.

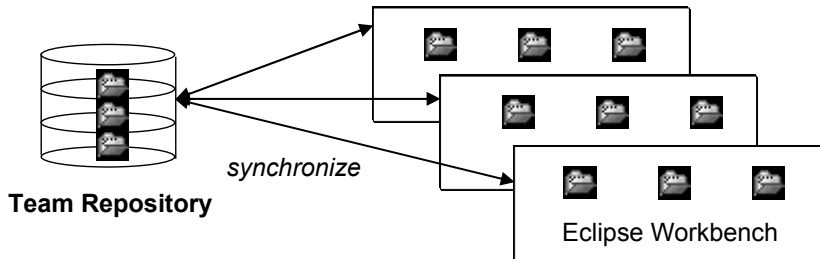
Eclipse User Interface – Open Perspective



The Eclipse workbench has four open perspectives. You can see them lined up on the left frame of the workbench. The current 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 that you saw previously. This particular perspective has a Package Explorer view which drives the Editor and the Outline view.

Eclipse Team Support

- Users create and work with **projects** 
 - Contain **folders** and **files**
 - Which are versioned as they change
 - Are “typed” and have type-unique tools
 - Are **synchronized** between local workspace and team repository



Central to the Eclipse integrated development environment 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 scoped to 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 Concurrent Version System (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 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.

Table of contents

Strategy

□ □  **WebSphere Development Studio Client Version 5.1**

Summary

Now we know about the new tools, package, technology, and user interface directions for iSeries application development. Let's see how that all comes together with Development Studio Client Version 5.0 / 5.1. As we will see Development Studio Client Version 5.1, is the next generation of workstation development tools to create iSeries e-business applications that leverage the iSeries and the power of WebSphere Application Server.

WebSphere Development Studio Client Version 5.0

- Remote System Explorer and iSeries Projects
- WebFacing Tool
- Web Tools, Java Tools, Web Services Tools
- XML, DB Tools



The second section of this presentation covers the tools in Development Studio Client. Development Studio Client Version 5.0 inherits and extends Site Developer's or Application Developer's (if you install Development Studio Client Advanced) powerful Java, Web, Web Services, XML, and database tools that speed the creation, testing, and deployment of sophisticated e-business applications. It also includes a robust, easy-to-use development environment including the Remote System Explorer (RSE) and iSeries Projects for creating, building and maintaining iSeries RPG, COBOL, C, C++ applications, and Web-enabled applications using the IBM WebFacing Tool. The Remote System Explorer (RSE) also makes it easy to access UNIX®, Windows, and Linux application resources.

Development Studio Client – Tools for e-business

- 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

You will most likely be working with the **Remote Systems Explorer (RSE)** perspective in the workbench for PDM-like access to OS/400® resources, and iSeries Projects for disciplined team development that leverages the excellent team capabilities of Eclipse. A perspective is a specific arrangement of views and tools in the workbench, depending on what role you have, you will use a different perspective. A Web developer who wants to quickly convert classic application's UI components like DDS, help, and message files into Web GUI components will use the **WebFacing perspective**. A Java developer who wants to develop Java applications or applets with Interfaces to iSeries classic applications will use the **Java perspective**. A Web developer who wants to use the Web tools iSeries Extensions to develop and maintain Web GUIs to iSeries applications will use the **Web perspective**. An XML developer who wants to work in the Visual XML Development Environment will use the **XML perspective** and the Database developer who wants access database tools will use the **Database perspective**.

Development Studio Client

RPG and COBOL Tools

ADTS has been the traditional method for developing and maintaining server-side iSeries applications. WebSphere Development Studio Client for iSeries, V5.0, includes new highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offer programmers a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications, lowering the learning curve for all.

These new generation tools include the Remote System Explorer for a PDM-like experience, and iSeries projects for team-based development (together with a Eclipse-compliant software change management repository). They offer rich support for exploring the file system, compiling/building, editing, running, and debugging. The iSeries Projects support enables effective team support leveraging any iSeries or LAN resident source repository which supports Eclipse.

Remote System Explorer (RSE)

▪ Perspective plus supporting views and editors

The
new
PDM!

The
new
SEU
and
SDA!

- 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(RSE), which has its own perspective and views. It is similar to PDM in that it allows you 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, you can use RSE 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.

The remote commands interface allows you to submit requests to the iSeries to invoke actions like compile, bind, or build objects on the host. The Remote Systems LPEX Editor is built right into the workbench, with rich editing functions and is iSeries aware! It is a superset of SEU! The syntax checker is ported down from SEU, the compilers embedded for verifying errors and the reference manuals are built-in and F1 cursor sensitive. The Outline view will show you the program hierarchy. There is explicit and rich iSeries support for verify, compile, run and debug of RPG, COBOL, C, C++, CL and DDS right from the Remote Systems LPEX Editor. Instead of SDA you can use a rich graphical user interface that makes designing or maintaining display file screens, printer file reports, and physical file databases easy and fun. This is CODE Designer and although it is not built-in to the workbench yet, you can easily launch it from the Remote System Explorer.

Business partners can easily leverage and extend RSE and iSeries Projects. There are also many enhancements to make it even easier to tightly integrate business partner tools with the IBM base development environment.

What's new in RSE for Version 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, Windows
- New common debug UI, and editor breakpoint support
- Remote Text Search (eg, FNDSTRPDM interface)

○○○ A partial list

Some of the primary enhancements in 5.0 are:

- Support for a new library list filter, and actions for manipulating the library list. This is much like *LIBL support in WRKLIBPDM.
- New easy-to-use wizards to create libraries, files, members, data areas, data queues and message files.
- Ability to drag source members to another file, or another iSeries, or to any other system such as local, UNIX, Windows and Linux source files.
- The primary view in Remote Systems Explorer is the Remote Systems tree view, but PDM users are used to a table view, so there is now an iSeries Table view that shows what the Remote Systems tree view shows, but in table format. You can sort the columns when you click on the column heading. The contents of the table are easily replaced for example when you double click 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.
- The Remote Systems LPEX Editor is 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 which offers lists of options for code-completion based on the location of the cursor in the source. You invoke Content assist easily in the editor through the Ctrl+spacebar keys.
- The List fields option is available when the cursor is on an F-spec or COBOL Copy-DDS statement. This option opens a Window that shows all the fields in the referenced file.
- There are also significant enhancements to the editor's open, save and runtime performance.
- A new command console is available for QShell, UNIX, Windows, Linux and local which are very rich in capability. You can run any command remotely, and see the feedback. The feedback is parsed, so when listing files, for example, you can double click to open the file. Or double click on a folder name to open a table view on the folder's contents.
- The common Eclipse-supplied Debug perspective to iSeries programs is integrated into the workbench, offering skills transfer. You can set breakpoints in the built-in Remote Systems LPEX editor, and then run to the breakpoint. This opens the Debug perspective. There is also new support for the common Eclipse run-configurations and debug-configurations.
- The Remote Text Search is a local GUI interface to FNDSTRPDM (find string PDM) that searches remote members for a given string. The results are shown in a local window, and you can double click to open the editor and position the cursor.

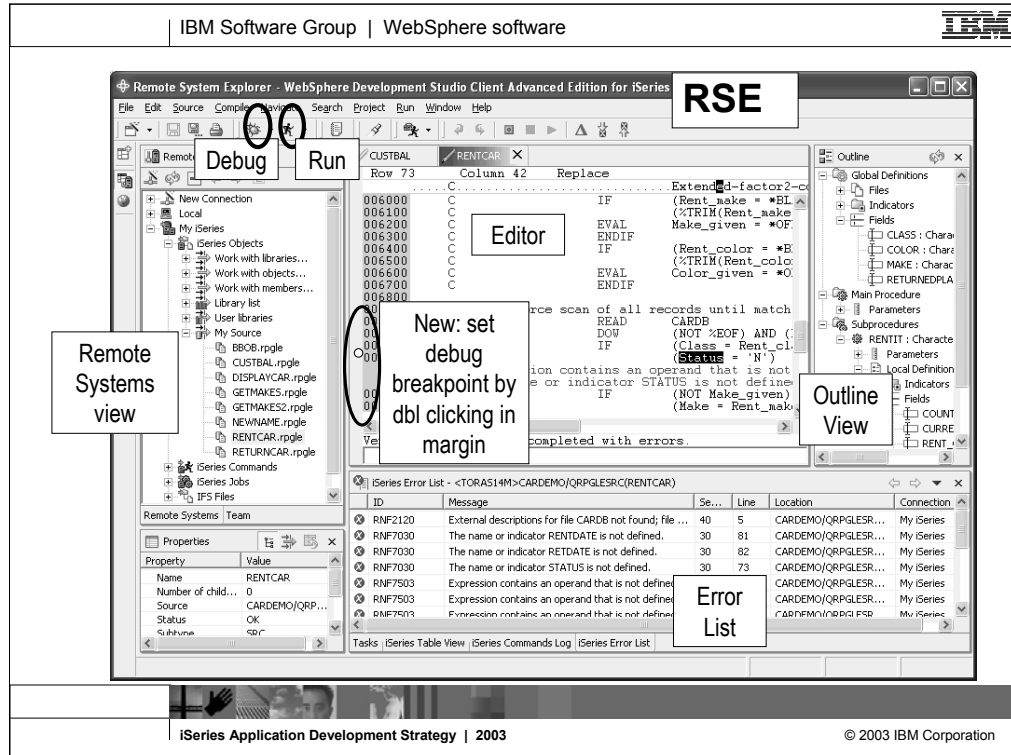
What's new in RSE for Version 5.1

- Improved caching support for Program Verifiers
- Enhancements to CL command prompting
- Multiple-selection for compiles
- Expandable *PGMs, *SRVPGMs (to see modules)
- Expandable modules (to see exported procedures)
- Editor wizard for creating RPG IV procedures
- Editor enhance'ts to RPG IV, COBOL outline view and code assist
- Editor template support for RPG IV free form, COBOL, C and C++
- Editor quick-open dialog (Ctrl+Shift+N)
- Remote Agent Controller support for iSeries
 - Enables publishing of Web projects and profiling of Java code
- Enhancements to Linux/Unix/Windows/QShell support
 - Compare two files, or two versions of same file
 - Replace file from local history
 - Drag 'n drop to editor, Drag 'n drop to populate filters
 - Content assist for commands, in console

○ ○ ○ A partial list

Some of the primary enhancements in the 5.1 release are listed here.

- Improved caching for the verifiers. This now behaves as it did in CODE – the cache information is used if available, unless the refresh-cache option is chosen. This can significantly improve verifier performance.
- CL command prompting is now SWT-based versus Swing-based, so it looks like the rest of the IDE.
- It is now possible to select multiple members to submit to compile. The order they are compiled is configurable by member type.
- Program objects are expandable to show modules.
- Modules objects are expandable to show exported procedures within the module.
- The CODE/400 SmartGuide for creating new procedures has been ported to the new editor, and enhanced
- The outline view for RPG IV shows more information now, and positions properly when items selected. There is totally new support for code assist for COBOL, and the RPG IV code assist is way better with support for parameters to keywords, functions, etc and case sensitivity.
- There is a new action from the local toolbar of the Remote Systems view to quickly open a member for edit or browse. It is also easily invoked via Ctrl+Shift+N. This can be faster than drilling down if you know what you want to edit.
- The Remote Agent Controller has been ported to iSeries, as well as the agents that come with it for Web publishing and Java profiling. See the documentation for more details. There are also actions in the RSE now for starting and stopping the RAC server on the iSeries.
- The non-iSeries support is also much enhanced this release:
- New actions to compare a file with another, or another version of itself.
- New action to replace a file with a previous version from the local history. Each edit of the file creates a new local history.
- Open files by dragging them to the editor.
- Add folders/files to filters by dragging them to a filter.
- Use Ctrl+Space in the command console to see a list of commands available (built-in plus on path).



This is the Remote System Explorer perspective. Here you can 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. You double-click a member to open the built-in LPEX editor shown in the middle. Notice the Outline view to the right. When you click on an element in the Outline view, the cursor is positioned to that element in the editor. The iSeries Error List below shows the results of a program-verify, or a compile. You launch program verify from the Source pulldown menu while Compile is launched from the Compile pulldown. There are other tabs behind the Error List view such as the Properties view, the iSeries Table View, and the iSeries Commands Log. As you can see there are numerous useful views in the Remote System Explorer perspective.

Remote System Explorer Scenario

- Create a connection to your iSeries system
 - After created, configure its library list, environment variables, etc
- Expand connection and subsystem
 - Decide if you want to work with objects, commands, 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

When you first open the Remote System Explorer, you are not connected to any system except your local hard drive on our workstation. To connect to a remote iSeries host, you need to define a connection. Once the connection is configured you are ready to connect to your iSeries host and expand a subsystem. Subsystems are a functional grouping of the various types of remote resources that can be explored in the remote system. The most common subsystem you will use is the iSeries Objects which is the PDM-like grouping, allowing access to libraries, objects and members. Next you can work with a default or predefined filter such as the library list filter to access a library in your library list or you can create your own filter if you want to access specific objects or members. Using the filter you can drill down the list of objects and use right click actions on an object. For example you can drill down to a member that you want to edit, right-click on that file which opens the Remote Systems LPEX Editor. You can then perform a number of actions on that file right from the Editor, which include syntax checking, verifying, compiling, running and debugging.



Remote Systems LPEX Editor

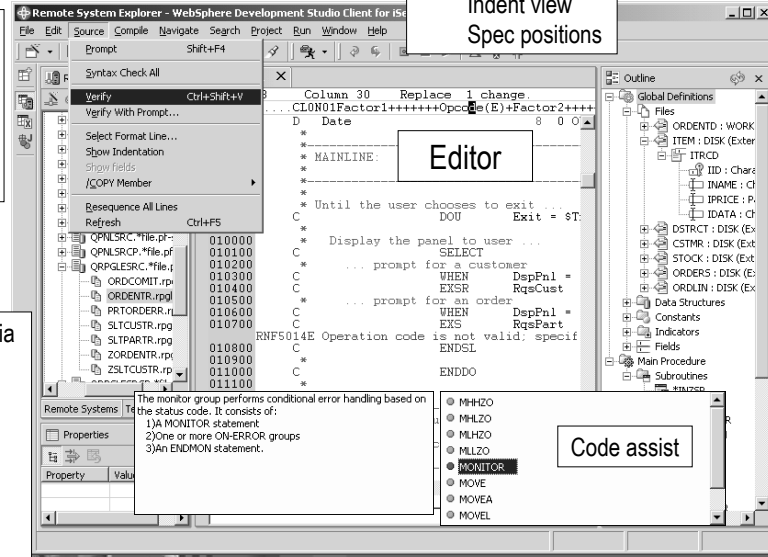
Actions

- Verify
- Compile
- Indent
- View
- etc

Invoke code assist via
Ctrl+space, for:

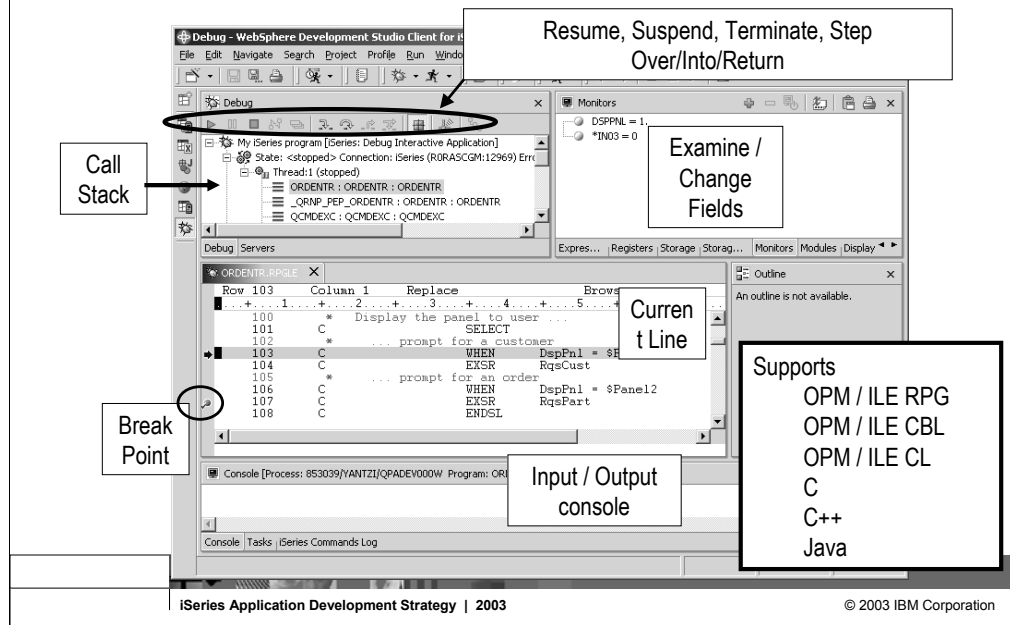
- Op-codes
- Field names,
- Proc name,
- Built-in fns,
- Keywords,
- etc

- Integrated Help
- Opcodes
- Error msgs
- Indent view
- Spec positions



Here we see the most compelling aspect of the Remote System Explorer, the Editor with its rich support. For example, the content-assist shown here is very well received by programmers. It is invoked through Ctrl+Spacebar. With the cursor on a specific code element you can request content assistance on Op-codes, field names, procedure names, built-in functions, keywords and so on. The source pull down menu is where many of the editor's advanced functions are launched. Here we see the Verify, Syntax check and Prompt options for example. The Remote Systems LPEX Editor written in Java is a superset of SEU. The syntax checker is ported down from SEU, the compilers embedded for verifying errors and the reference manuals are built-in and F1 cursor sensitive.

RSE iSeries Integrated Debug Perspective



Here we see the common Eclipse Debug perspective, which is being used to debug an RPG program. The common debug user interface has been connected to the iSeries debug engine in Version 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 the debugger, including a pop-up menu action in the Remote System Explorer and the Debug icon in the toolbar.

In the upper left pane is the call stack, much like option 11 in the OS/400's WRKACTJOB. It shows the calls that reflect your current program execution. When you double click 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, with source executable (debug) lines in blue, others in green. The current line of execution is highlighted, and breakpoints appear as a dot in the left margin. The bottom shows the console where text written to the console through the DSPLY opcode in RPG, for example, is shown. If the program prompts for standard input (like C and C++ and Java support), you enter input in the console view.

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 amongst 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 Concurrent Versioning Systems (CVS), or the MKS Integrity. CVS is available on the distribution CDs for 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 of Eclipse, while developing applications targeted to run on iSeries.

What's new in iSeries Projects for Version 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 lib, 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

The iSeries Projects support for the 5.0 release included significant enhancements:

- New extension points for business partners to leverage
- Enhancements for build support, including monitoring of the build steps and error feedback
- The text description is now captured for iSeries files and members imported into an iSeries project
- Based on last modified timestamps, iSeries projects visually identify when local and remote copies of the same member are in conflict, and offer resolution support when the local member is pushed to the remote server
- Business partners can easily leverage and extend iSeries Projects. There are also many enhancements to make it even easier to tightly integrate their tools with the IBM base development environment.

What's new in iSeries Projects for Version 5.1

- More intuitive perspective layout
- New right-click actions:
 - “Replace from Associated Library”
 - “Remote Actions->Select Compile Command”
 - “Remote Actions->Work with Compile Commands”
- Can now add tasks and bookmarks to files
 - Such that double clicking on task/bookmark opens the file
- Can “link editor”
 - Selecting an editor selects the member in the navigator view
- Can link project to local file system
 - Use an existing folder as-is, without having to import
 - Same as with all eclipse projects in 2.1

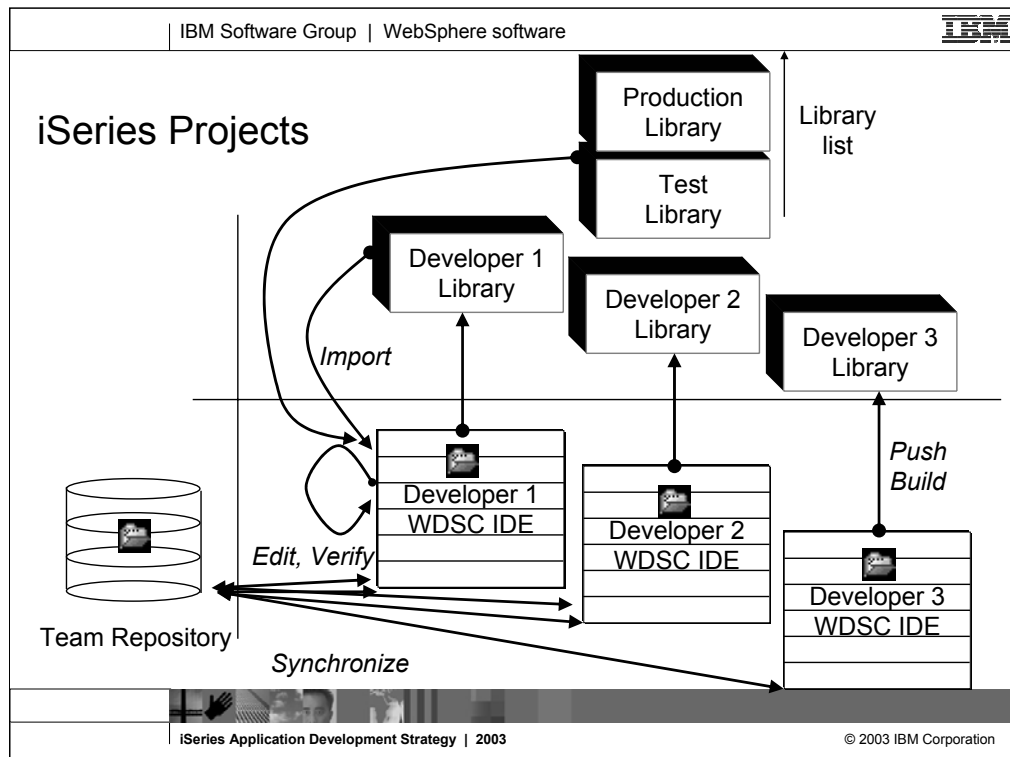
The iSeries Projects support for the 5.1 release included more enhancements still.

iSeries Projects Scenario

- **Project created to do a fix or enhancement**
 - Associated with library
 - Populated with source required for work
 - From RSE, or through 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 or Aldon
- **Run or Debug**
- **If working in team, synchronize with team repository**

A project is the top level of organization of your resources in the workbench. A project contains files and folders. Projects are used for building, version management, sharing, and organizing resources. A project can contain session and persistent properties, settings for environmental variables, and references to other projects.

The iSeries Projects perspective focuses on disconnected iSeries development on Windows platforms. Projects are local containers for local source, and the intent is that you edit these resources locally. Once you are finished updating files and members locally, you can push these changes to an iSeries host and submit a build of your project. You can then run and debug your program and of course synchronize your development work with a team repository.

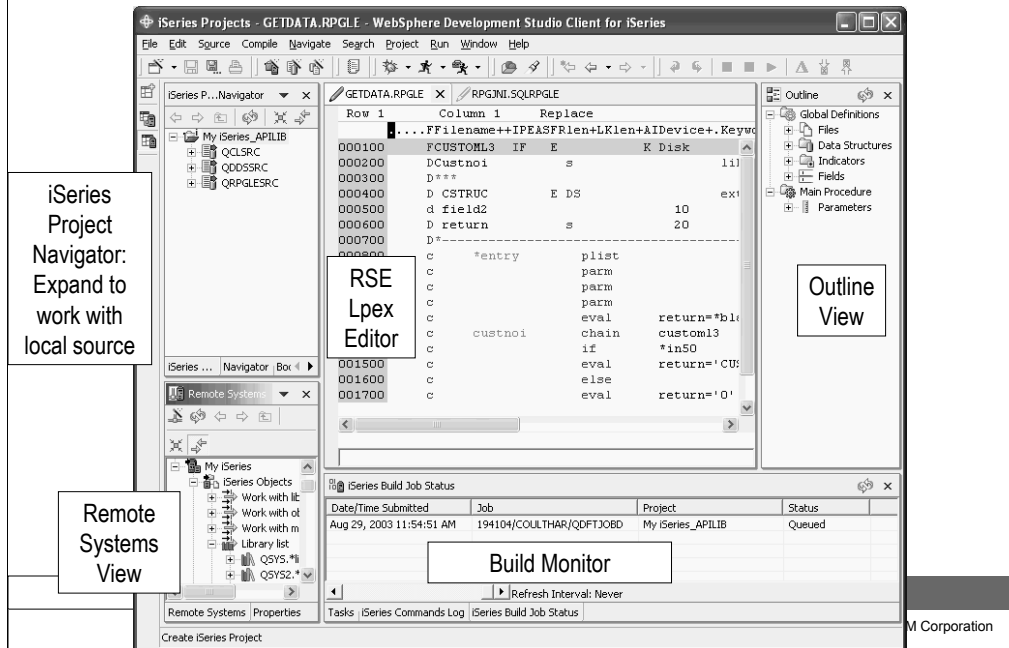


Because iSeries projects are in fact Eclipse projects, they inherit the team capabilities of all Eclipse projects. This offers developers the ability to share their work within a team, using the Eclipse team support.

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 a 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 Projects Perspective



The screenshot displays the iSeries Projects Perspective in the WebSphere Development Studio Client for iSeries. The interface is divided into several panes:

- iSeries Project Navigator:** Located on the left, it shows a tree view of the project structure, including local sources and remote systems.
- Remote Systems View:** A sub-pane within the navigator showing details for remote systems.
- RSE Lpex Editor:** The central editor window displaying a table of data with columns for Row, Column, and Replace.
- Outline View:** A pane on the right showing the structure of the current file, including Global Definitions, Files, Data Structures, Indicators, Fields, Main Procedure, and Parameters.
- Build Monitor:** A window at the bottom showing the status of build jobs, including Date/Time Submitted, Job, Project, and Status.

Row	Column	Replace
000100	FCUSTOML3	IF E K Disk
000200	DCustno1	s
000300	D***	
000400	D CSTRUC	E DS
000500	d field2	10
000600	D return	s 20
000700	D-----	
000800	*entry	plist
000900	c	parm
001000	c	parm
001100	c	parm
001200	c	parm
001300	c	eval return='bl
001400	c custno1	chain customl3
001500	c	if *in50
001600	c	eval return='CU
001700	c	else
001800	c	eval return='D'

Date/Time Submitted	Job	Project	Status
Aug 29, 2003 11:54:51 AM	194104/COULTHAR/QDFTJOB	My iSeries_APILIB	Queued

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 labeled 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, it's icon is yellow if they are in synch, and red if there are collisions.

The LPEX editor is the same rich editor we saw for the Remote Systems Explorer. When editing is complete and the project is pushed to the library and built, the build job is monitored in the job status window. When the build job is finished, you select the job and right click to see it's error list, which uses the same iSeries Error List window as the Remote System Explorer.

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

These new-generation tools include the Remote System Explorer for a PDM-like experience, and iSeries projects for team-based development (together with an Eclipse-compliant software change management repository). They offer rich support for exploring the file system, compiling/building, editing, running, and debugging programs. The iSeries Projects support enables effective team support leveraging any iSeries or LAN resident source repository which supports Eclipse.

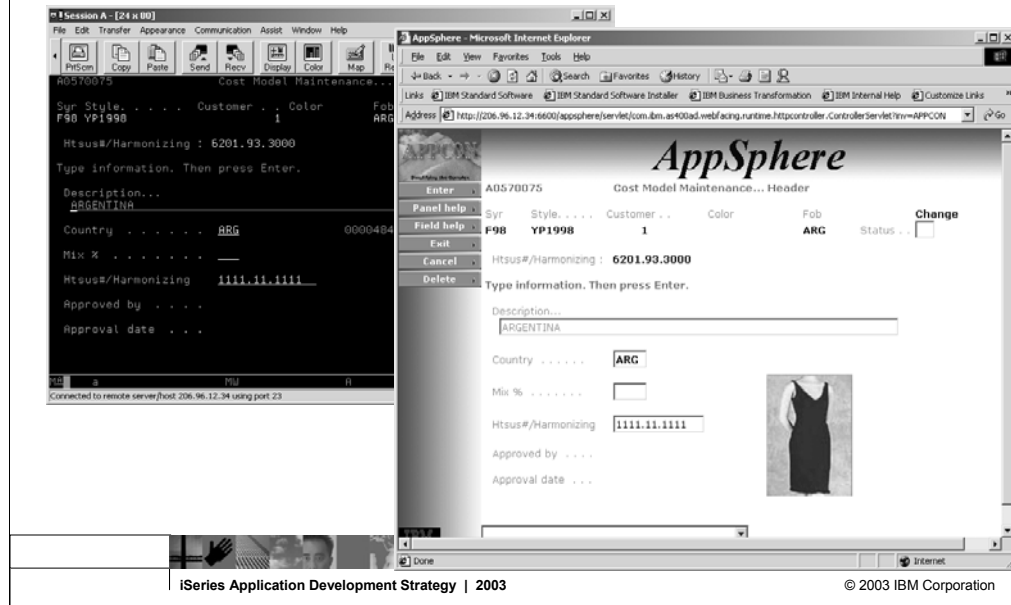
The new tools also allow you to leverage the classic CODE tools for rich editing and DDS design support. The combination of the Remote System Explorer, iSeries Projects, and the easily accessible CODE classic tools offers a superset of function over the classic, and feature-frozen, ADTS tools. It offers significant productivity and usability gains, support for disconnected and team development, and a common harness for the tight integration of IBM and ISV-supplied tools for server development. Using these new generation tools also implicitly increases a programmer's skills and makes the transition into new programming models easier, such as IBM WebFacing Tool, Web, Web Services, Java, and XML.

Development Studio Client

WebFacing Tools

One of the most exciting components of WebSphere Development Studio Client for iSeries is the IBM WebFacing Tool. The IBM WebFacing Tool creates a Web interface to an existing 5250 application in a development-time conversion process. The conversion process typically requires little or no change to the existing iSeries programs. The resulting application continues to support a green screen and will also support the Web interface. The tool generates standard JavaServer Pages and XML definitions, based on the display file DDS, and includes a servlet-based runtime. The JSPs and runtime run in any application server that supports these Java Web components.

WebFacing



Here you see a sample "before and after" of a green-screen that has been WebFaced. These screens are courtesy of APPCON, an IBM iSeries business partner. You can see more samples at www.appcon4.com.

As you can see, the role of WebFacing is to quickly convert classic application's UI components like DDS, help, and message files into Web GUI components.

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 generated code
 - Option to generate **portal**-tolerant application through 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 (for example, Web pages).

The JSPs generated by WebFacing are then deployed to a Web application server, such as WebSphere Application Server, 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.

What's new in WebFacing for Version 5.0

- Automatic configuration of UTF-8 in WebSphere AS 5.0
- Web page compression when using WebSphere AS 5.0 (performance)
- Support for function 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)

In the 5.0 release, IBM introduced the performance and scalability enhancements to allow the IBM WebFacing Tool to support extremely large and complex applications with high traffic flows. IBM is also providing a Web interface to the most requested system support — WRKSPLF. IBM is also making it easier to leverage and extend Web applications created with the IBM WebFacing Tool. IBM has provided an option during the code generation phase to output Struts-compliant Web application infrastructure that can be easily enhanced. The Struts-compliant runtime can now leverage the Struts tools to provide more customization capability. Also included in the IBM WebFacing Tool are:

- Conversion-time extension points to allow developers to replace the default HTML
- Option to use IBM and user-defined custom tags to enhance customization
- Ability to override command-key actions to invoke user-defined functions
- Support for displaying and printing iSeries spool files from inside your Web-enabled applications generated using the IBM WebFacing Tool

What's new in WebFacing for Version 5.1

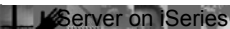
- Support for WebSphere Application Server 5.0.2
- Preference to compress/not-compress xml files.
- Better error-handling of userId/password prompt, including support to prompt for expired password.
- New properties affecting conversion / runtime:
 - Show function-key buttons within windowed records
 - Compress web pages before sending to browser
 - Mapping a key to field-exit behavior
- Support for dynamic function key fields
 - Web setting to identify output fields that hold function key descriptions
- Support for new IBM Site Designer
 - Create web sites easily, and contribute WebFaced apps to them
- Less resources used for subfiles, aiding scalability (large # users)
- Support for Struts 1.1 in advanced edition.

In the new 5.1 release, IBM continues the tradition of many enhancements to this strategic tool.

WebFacing Scenario

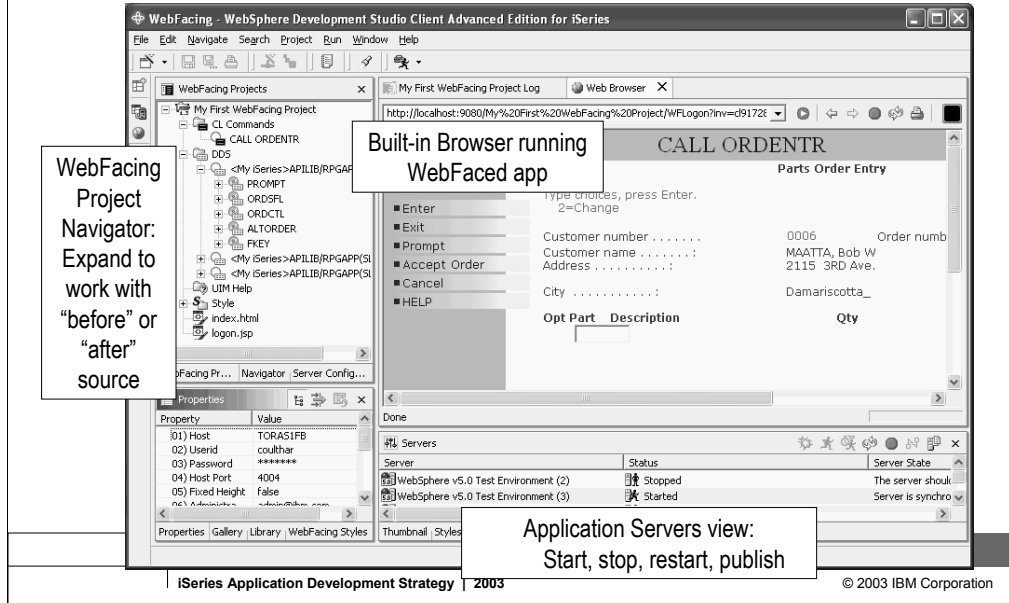
- **Create new WebFacing project**
 - Specify members to convert, look-and-feel style, how to call/invoke the application
- **Convert the project**
 - Creates Web stuff from the DDS and UIM source
- **Run it!**
 - Select “Run on Server” to run in built-in WebSphere
- **Refine it**
 - Using Web Settings in CODE Designer (launched from WebFacing perspective) and using project’s Properties dialog
- **Repeat**
- **Publish**

Export as EAR file, import into WebSphere Application Server on iSeries Or use built-in server-tools support to publish to WebSphere Application

 Server on iSeries

With the WebFacing Tool, you can quickly convert your DDS display file source members so that the user interface of your iSeries programs can run in a browser. You first create a WebFacing project to hold your converted files. When you convert your DDS display files, JSPs and Java beans are generated for you that substitute for the DDS code and make Web access possible. In the WebFacing Project wizard, you can select one or more DDS source members to convert, and select a Web look and feel from one of several predefined styles, or you can design your own Web style for use with your applications. The tool creates three Java beans and one JSP per record format; the Java beans hold the data for the record format, or control its appearance or other characteristics, and the JSP handles displaying the Web version of the screen, prompting for data, and handling input errors. The wizard generates an application home page to launch the Web-enabled version of your program. You run your Web-enabled program using the built-in WebSphere Application Server. You can then use the refine the Web input and output pages using Web Settings in CODE Designer. Then you re-run your Web enabled program. Finally you can export the project as an Enterprise Archive file to run on a production WebSphere Application Server on iSeries or use built-in tools to publish the Web program to WebSphere Application Server on iSeries.

WebFacing Perspective



WebFacing Project Navigator:
Expand to work with "before" or "after" source

Built-in Browser running WebFaced app

Application Servers view:
Start, stop, restart, publish

Property	Value
01) Host	TORAS1FB
02) Userid	coulthar
03) Password	*****
04) Host Port	4004
05) Fixed Height	False

Server	Status	Server State
WebSphere v5.0 Test Environment (2)	Stopped	The server should be started.
WebSphere v5.0 Test Environment (3)	Started	Server is synchronizing.

ISeries Application Development Strategy | 2003

© 2003 IBM Corporation

Here you see the WebFacing perspective. There is the WebFacing Projects view 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 WebFacing Projects view. Also, the conversion itself is launched from the pop-up menu of the project (or an individual DDS member).

Notice the built-in Web browser when running the converted application using the built-in copy of WebSphere Application Server, which is managed in the Servers view at the bottom.

Development Studio Client

Java
Tools

Java development tools and iSeries Java development tools give you the ability to develop Java applications and write, compile, test, debug, and edit programs written in the Java programming language for Java applications development. Java development tools include an integrated environment that supports the complete cycle of Java program development.

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

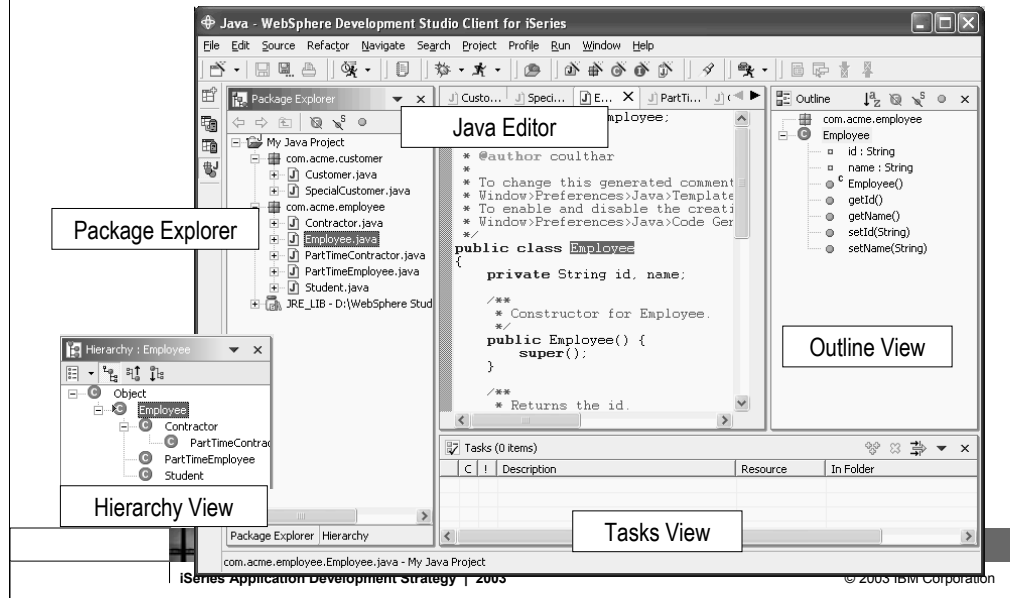
The Java tools in Development Studio Client come from the Site Developer or Application Developer if you have Development Studio Client Advanced Edition. Java tools offer their own project type and perspective, optimized for ultra-productive Java development.

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 high level view of your Java class. From here many wizards can be launched.
- A Java editor that is super optimized for Java. One key function includes the ultra-popular content assist from VisualAge, and another new function shows the JavaDoc for a method or class when the cursor hovers over a reference to them. There is also an F3 short cut key that when used while a Java class name is selected, opens a new editor on that class.
- There is also 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 just save your changes as you edit, and the compile happens automatically. You can also force a compile through a pop-up 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 specified as a property so you can download the latest version and use the new Java tools to work with it!

Java Perspective



Here you see the Java perspective. It has a Package Explorer view that is the primary view driving the other views. 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, using the selected class in the Package Explorer.

Java Tools iSeries Extensions

- **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, 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 Development Studio Client IDE
- A Program Call wizard to generate a Java class to call existing RPG or COBOL business logic
- A plug-in 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

What's new in iSeries Java Tools for Version 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 command+parameters, 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



Some of the 5.0 enhancements included:

- Visual editor with two-way synchronization between source and GUI
- JavaScript debugger, library, and code assist
- New JSP samples
- Launch Configuration support

New 5.0 iSeries extensions of Java development tools include:

- J2EE Connector Architecture and PCML 4.0 support
- The ability to externalize strings for creating multi-lingual applications
- iSeries-specific Launch Configuration support, so that you can remotely execute and debug Java programs on an iSeries host

What's new in iSeries Java Tools for Version 5.1

- Eclipse 2.1 Java tools
- Enhanced Java Visual Editor for creating Swing UIs
 - Support for events
- Enhancements to Program Call wizard
 - Right click actions on program/parameters page for ease of use



The 5.1 release was a big release in terms of inherited eclipse 2.1 support for Java, but a small release for the iSeries extensions to the Java tools.



iSeries Java views

The screenshot displays the IBM WebSphere Development Studio Client Advanced Edition for iSeries interface. The main editor shows a Java source file with the following code:

```

package com.acme.package1;

/**
 * Customer class
 */
public class Customer
    
```

Annotations indicate: "Open iSeries view in Java perspective" pointing to the code editor.

The "Select iSeries Options" dialog is open, showing configuration for "Option set 1":

- Transform: TO
- Replace program: *YES
- Enable performance collection: *NONE
- User profile: *USER
- User adopted authority: *NO
- Directory subtree: *NONE

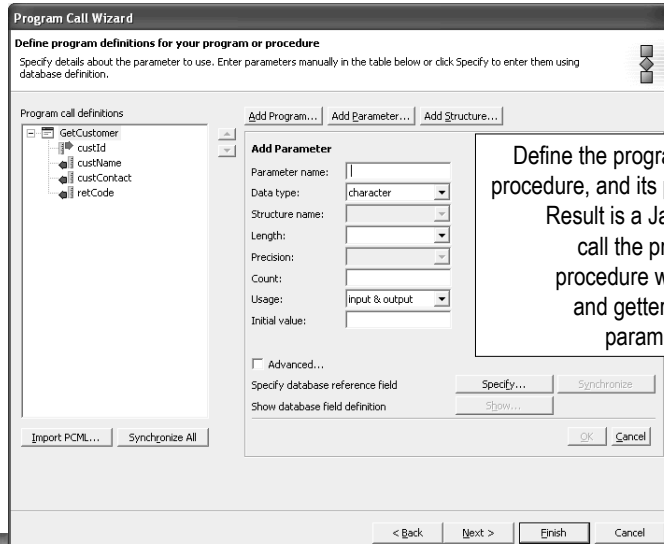
Annotation: "Create Named Set of Options for Compile and Debug" points to this dialog.

The "iSeries Java Transform and Run" dialog is also open, showing fields for "Working folder", "Option set", "Class", "Classpath", and "Arguments". A "Run, debug remotely" button is visible. Annotation: "Specify remote iSeries class to run, debug or compile (transform)" points to this dialog.

At the bottom of the screenshot, the text "iSeries Application Development Strategy | 2003" and "© 2003 IBM Corporation" is visible.

Here you see the Java perspective, but with the additional iSeries-specific view for remotely transforming, running and debugging Java on iSeries.

Program Call Wizard



Program Call Wizard

Define program definitions for your program or procedure

Specify details about the parameter to use. Enter parameters manually in the table below or click Specify to enter them using database definition.

Program call definitions

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

Add Parameter

Parameter name: |

Data type: character

Structure name:

Length:

Precision:

Count:

Usage: input & output

Initial value:

Advanced...

Specify database reference field Specify... Synchronize

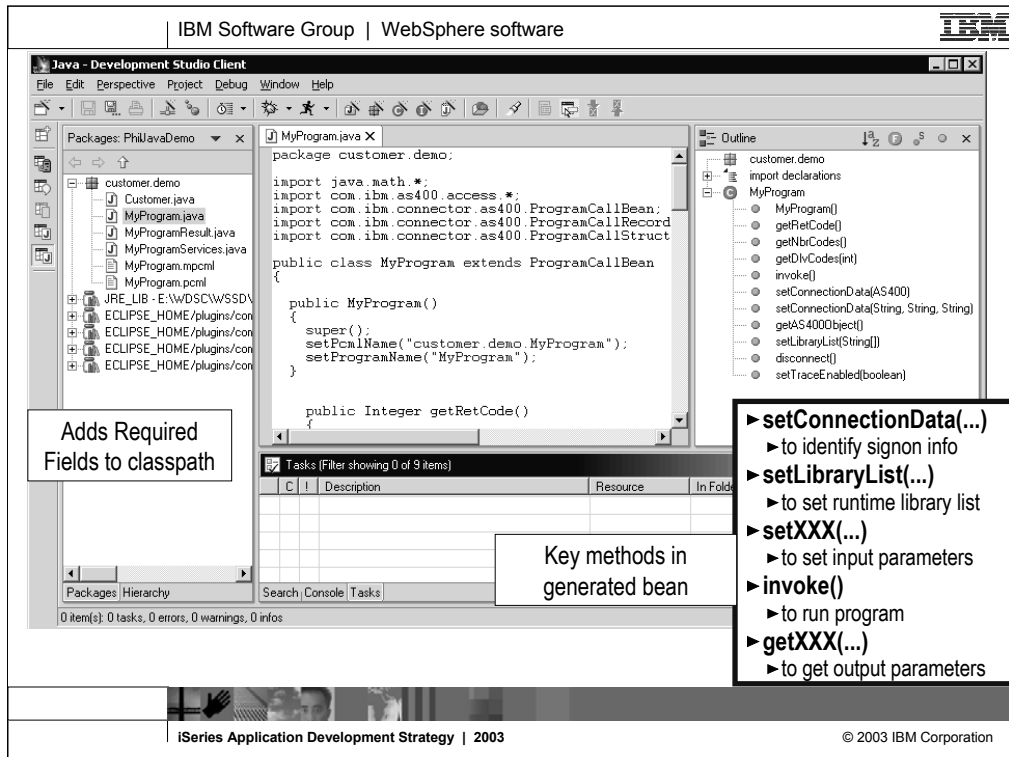
Show database field definition Show...

OK Cancel

< Back Next > Finish Cancel

Define the program or ILE procedure, and its parameters:
Result is a Java Bean to call the program / procedure with setters and getters for the parameters

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

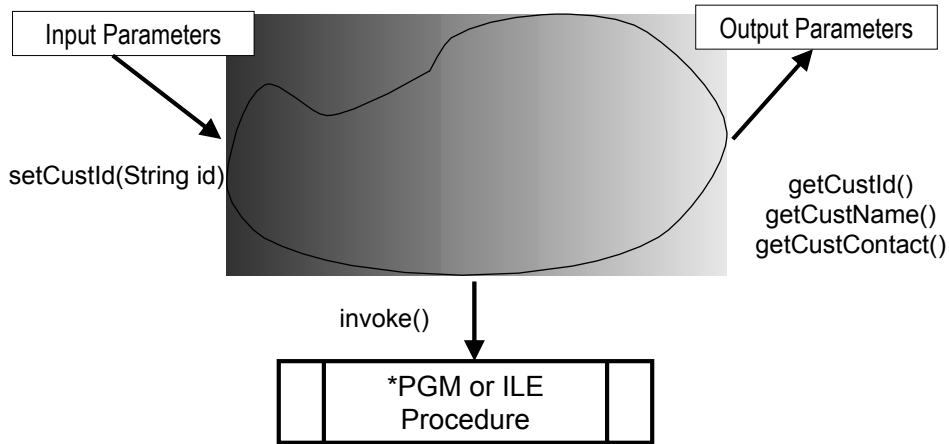


Here you see the result of the Program Call wizard. The selected Java package in the upper right now contains all the generated Java classes, one of which is shown in the editor in the middle view. To use the generated Java bean, you write Java code to instantiate it and call the important methods in 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 – how it works



This iSeries-unique function includes the Program Call wizard, that given a program or ILE procedure, will wrap it in a Java Bean. The Java Bean 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.

Development Studio Client



Web
Tools

Web development tools give you the ability to create new e-business applications that use a Web-based front end to communicate with the business logic in an ILE and non-ILE language program residing on an iSeries host. You can create your Web input and output pages with the Page Designer, or generate input and output pages from the Web Interaction wizard templates. You can also add iSeries Web components to your pages, for example, Web equivalents of iSeries command keys, input fields that accept only particular types of data, or subfile and other iSeries output fields.

Web development tools give you open standards, greater flexibility, and the ability to tie in existing applications with Web Services wizards to help you write applications that are portable across platforms.

Web Tools inherited from Site Developer

- **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 Development Studio 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 Development Studio Client 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 Development Studio Client.

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 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

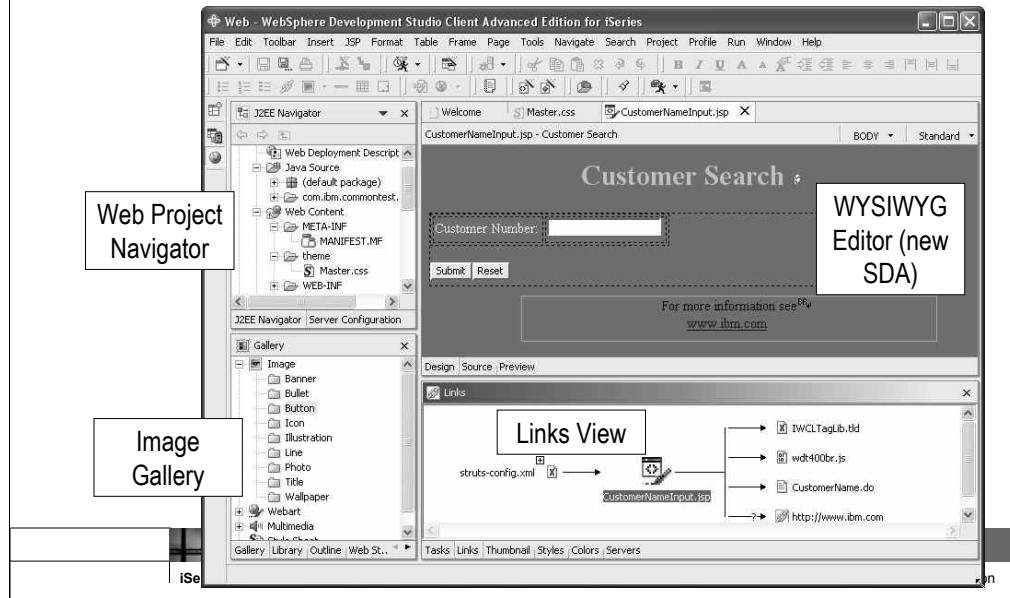


In addition to the many editors and views, the Web tools include a number of useful wizards to kick-start your Web development with minimal skills.

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 through 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.

Result Java Bean Web Page Wizard



Here you see 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 pop-up 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

- Now this is VERY COOL!
- When ready to test your Web application
 - 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 WebSphere Application Server
 - Starts built-in copy of WebSphere Application Server
 - 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.

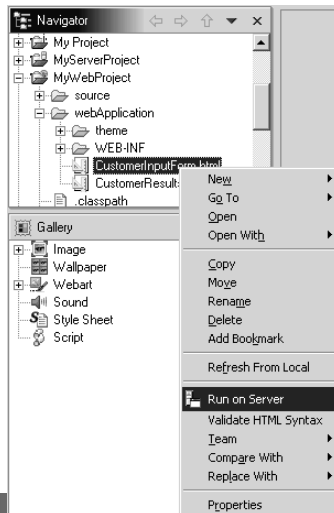
Luckily and handily, an application server is built right into Development Studio Client so you can immediately run and see your beautiful work without wait. Of course, this built-in application server is in fact a full-fledged copy of WebSphere Application Server. In fact, there are multiple copies; WebSphere Application Server 4.0, WebSphere Application Server 5.0 Express and WebSphere Application Server 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 WebSphere Application Server, and switch you to the Server view so you can manage the life cycle of it. Once the server 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.

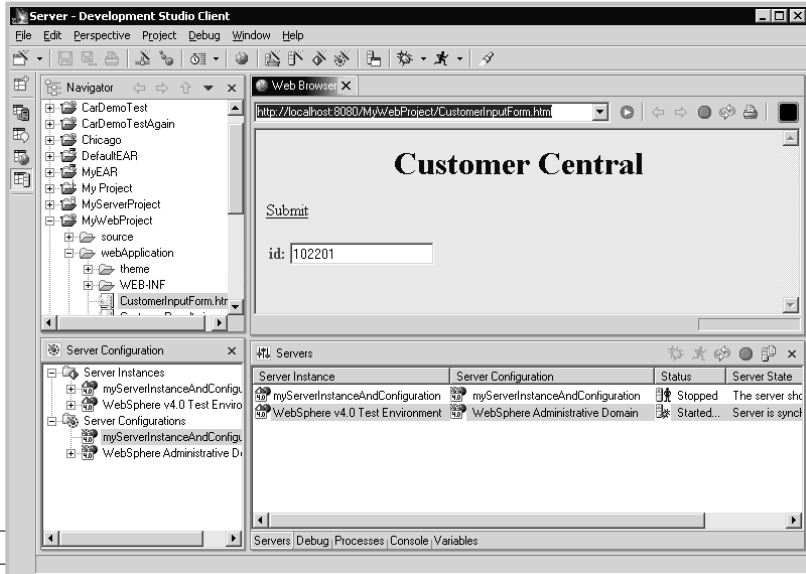
Testing Your Web Application



See how easy it is to run your Web application. Just right click, select "Run on Server", and...

Testing Your Web Application

Runs WebSphere
Application Server

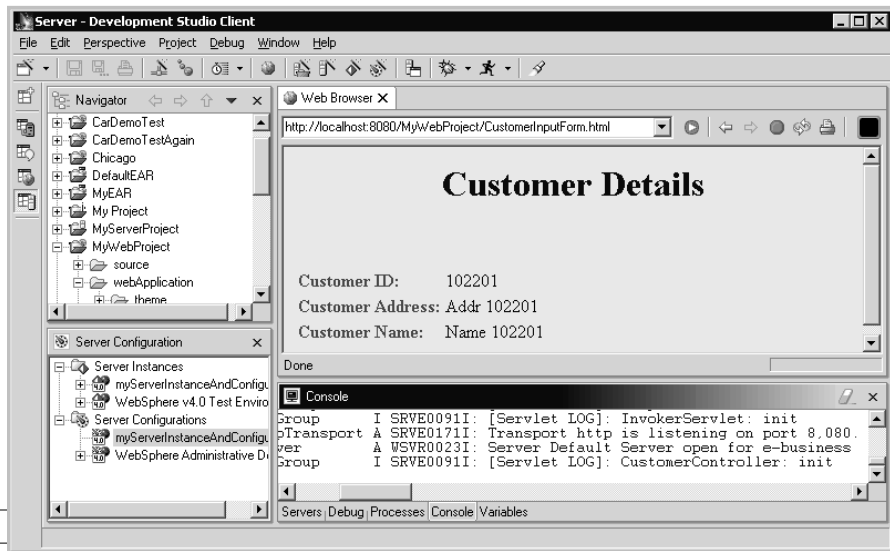


IBM Series Application Development Strategy | 2003

© 2003 IBM Corporation

... presto, your application is running in the browser! You can double click on the browser title bar to make it full screen, or press Ctrl+N to make it a separate window.

Web Application Results Shown



Here is the result of pressing Submit on the input page. 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 (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 WebSphere Application Server installation or availability
 - Easily debug prior to publish/deploy
- Supports configuring multiple Web apps
- Access to WebSphere Application Server Administration Client
- Server Tools
 - Supports start, stop, restart, publish actions for WTE, local or remote WebSphere Application Server or local TomCat

When using "Run on Server" you are by default running the full fledged copy of WebSphere Application Server that is built into Development Studio Client. 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 function, allowing starting, stopping and publishing of the server through the right click pop-up menu actions. You can even bring up the full WebSphere Application Server Administration console directly from this perspective. Very handy!

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

You can associate any Web project with any such local or remote WebSphere Application Server, 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 WebSphere Application Server requires at least 512MB of memory, so you will need this much to use this awesome functionality effectively.

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 WebSphere Studio Site Developer or Application Developer if you are using Development Studio Client Advanced, and not unique to iSeries.

In addition to these, there are additional tools in Development Studio Client 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 sign-on information and library list setup.

What's new in iSeries Web Tools for Version 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

Some enhancements in Version 5.0 of the Web Tools:

- Updates to Page Designer to support Web components and a Links builder. Web components replace ActiveX Design-time Controls, can be rendered in the HTML Designer, and provide test field, label, drop-down, and push-button controls.
- Support for the Struts run-time environment through the Web Interaction wizard.
- iSeries Web Tools Run-time Configuration wizard that captures hostname, user ID, password, and library list information.
- Support for static Web pages and XHTML.
- Visual Custom Tags (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 forth.

What's new in iSeries Web Tools for Version 5.1

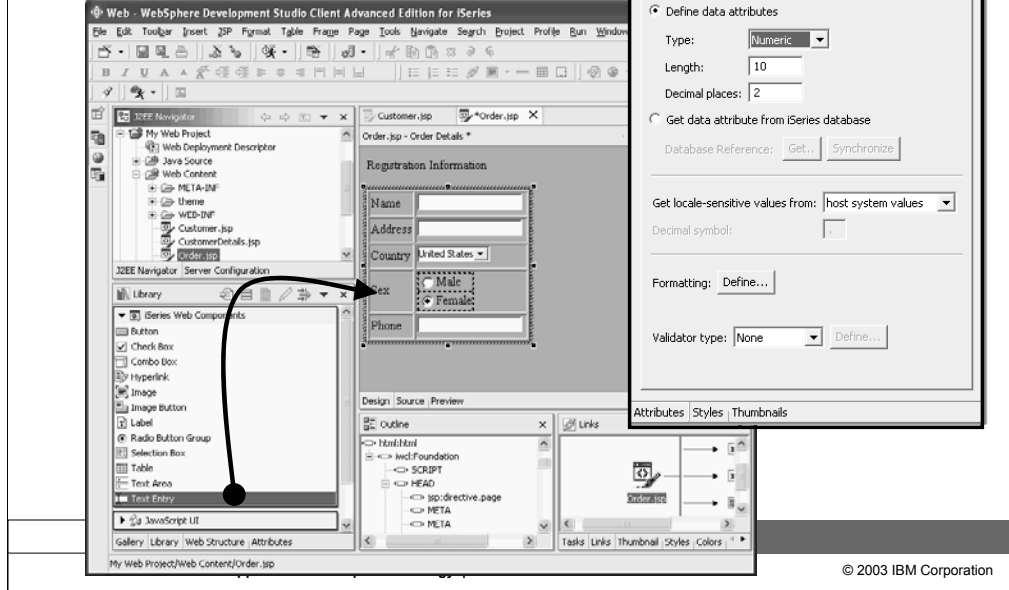
- Support for Struts 1.1 and WAS 5.0.2
- Support for internationalization:
 - Web pages generated with UTF-8 encoding in Web Interaction Wizard
 - Support in iSeries Web Components for translatable strings coming from a resource bundle
 - Support for placing translatable strings in resource bundle in generated Web pages in Web Interaction wizard
 - Support in iSeries Web Components for retrieving editcode/editword dollar sign, thousands-separator and decimal separator from iSeries host or Web page locale.
- Support for importing PCML generated by compilers
- Ease of use enhancements to Web Interaction Wizard



Some of the enhancements in Version 5.1 of the Web Tools

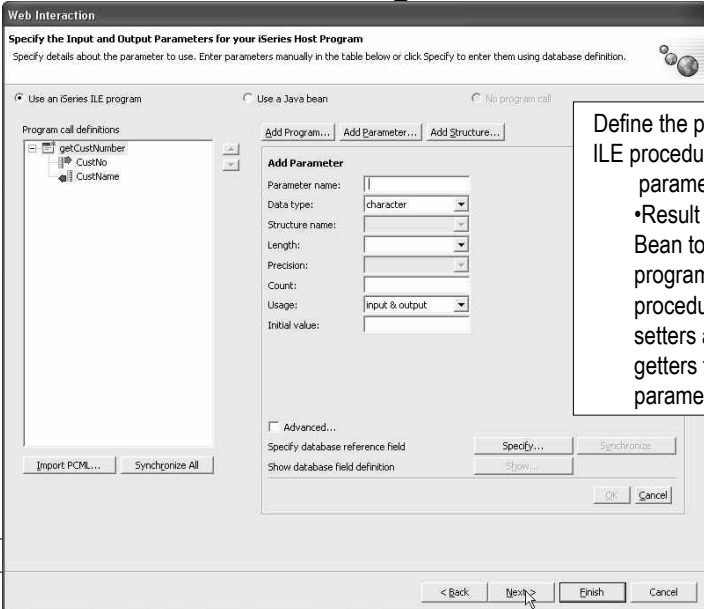
iSeries Web Tools: Web Components

Web Perspective



This is the Web perspective. The upper left is the Web Projects view that is the primary view that drives the other views. 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 and a Links view that shows all the relationships between the selected file and the other files in the project.

Web Interaction Wizard – Program Definition



Web Interaction

Specify the Input and Output Parameters for your iSeries Host Program

Specify details about the parameter to use. Enter parameters manually in the table below or click Specify to enter them using database definition.

Use an iSeries ILE program Use a Java bean No program call

Program call definitions

- getCustNumber
 - CustNo
 - CustName

Import PCML... Synchronize All

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

Add Parameter

Parameter name:

Data type: character

Structure name:

Length:

Precision:

Count:

Usage: Input & output

Initial value:

Advanced...

Specify database reference field Specify... Synchronize

Show database field definition Show...

OK Cancel

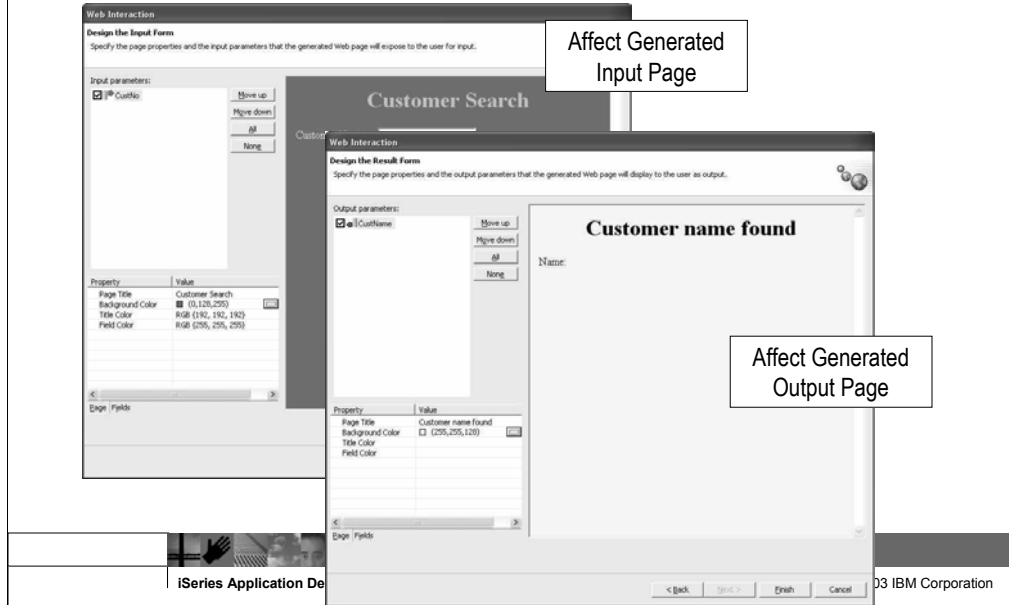
< Back Next > Finish Cancel © 2003 IBM Corporation

Define the program or ILE procedure, and its parameters:

- Result is a Java Bean to call the program / procedure with setters and getters for the parameters

This is the Web Interaction wizard. This page 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.

Web Interaction Wizard – Page Definitions



Design the Input Form
Specify the page properties and the input parameters that the generated Web page will expose to the user for input.

Input parameters:
 CustNo

Property Value
Page Title Customer Search
Background Color RGB (0,128,255)
Title Color RGB (192,192,192)
Field Color RGB (255,255,255)

Design the Result Form
Specify the page properties and the output parameters that the generated Web page will display to the user as output.

Output parameters:
 CustName

Property Value
Page Title Customer name found
Background Color RGB (255,255,128)
Title Color
Field Color

Affect Generated Input Page

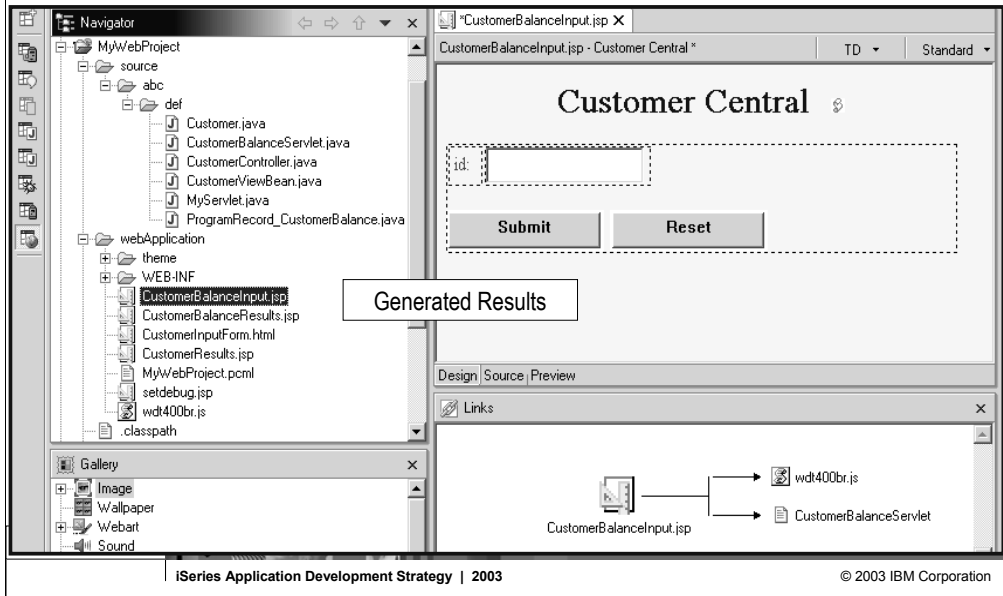
Affect Generated Output Page

ISeries Application De

03 IBM Corporation

This page requests 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.

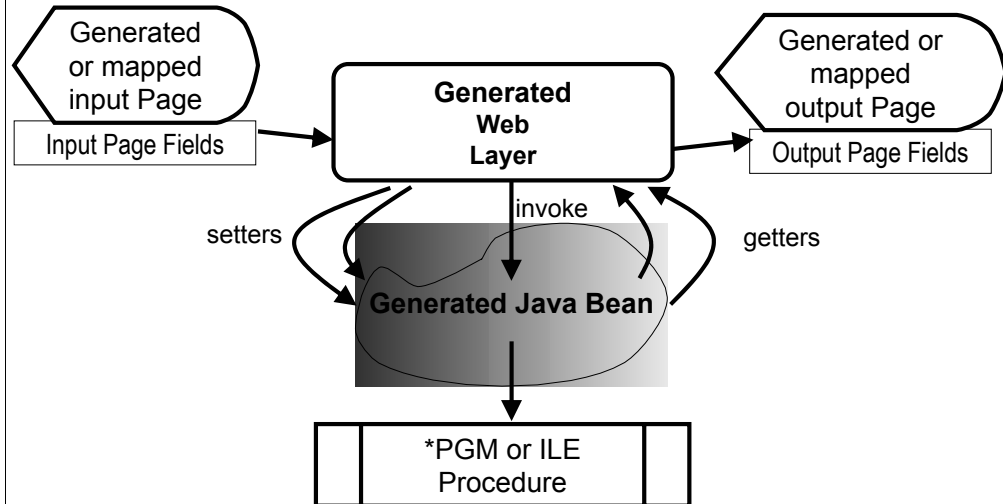
Web Interaction Wizard - Results



The screenshot displays the IBM WebSphere IDE interface. On the left, the Navigator pane shows a project named 'MyWebProject' with a directory structure including 'source' and 'webApplication'. The 'source' directory contains files like 'Customer.java', 'CustomerBalanceServlet.java', 'CustomerController.java', 'CustomerViewBean.java', 'MyServlet.java', and 'ProgramRecord_CustomerBalance.java'. The 'webApplication' directory contains 'theme', 'WEB-INF', and 'CustomerBalanceInput.jsp'. A callout box labeled 'Generated Results' points to the 'CustomerBalanceInput.jsp' file in the 'WEB-INF' directory. The main editor shows a preview of the 'Customer Central' web page, which includes an input field labeled 'id:' and two buttons, 'Submit' and 'Reset'. Below the preview, there are tabs for 'Design', 'Source', and 'Preview', and a 'Links' pane showing dependencies between 'CustomerBalanceInput.jsp' and 'wdt400br.js' and 'CustomerBalanceServlet'. The footer of the IDE shows 'I Series Application Development Strategy | 2003' and '© 2003 IBM Corporation'.

The results of the Web Interaction wizard show 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 – How it works



On top of the base Web tools, Development Studio Client adds Web tools unique to iSeries programmers, similar to the old iSeries Extensions that were in WebSphere Studio in WebSphere Development Tools.

This iSeries-unique function includes a wizard, that given a program or ILE procedure, will wrap it in a Java Bean 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.

Development Studio Client

Web Services Tools

Web Services are self-contained, modular applications that can be described, published, located, and invoked over the World Wide Web, employing 'just-in-time' integration of services. The architecture involves a relationship between Service providers, brokers, and requesters. More specifically, Service requesters submit a request to a Service broker, which finds the right service from a Service provider. Web Services tackle the problem of inefficient distributed computing, and Web Services development tools allow you to develop this service.

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 Program Call 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

Development Studio Client inherits a very rich set of Web Services tools from Site Developer or Application Developer (for Development Studio Client Advanced)

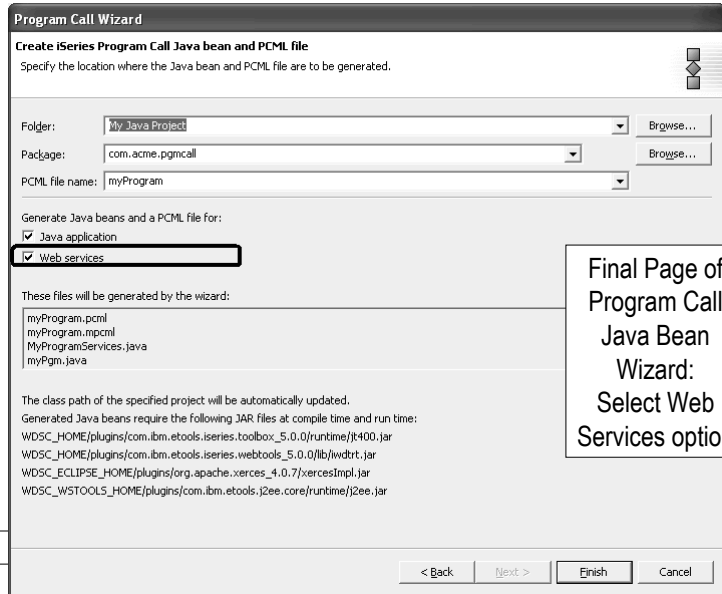
- Built-in access to the IBM test or official UDDI (Universal Description and Discovery Interface) registry, which you can browse to select Web Services and build Java Bean proxies from them. These Java Bean proxies contain a method to call the service, passing in the required parameters and returning the results.
- 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!
- Tools to deploy a Web Service to WebSphere Application Server (including the built-in copy) or TomCat to test it.
- 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!
- Once you have generated a Web Service, deployed it to WebSphere Application Server 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!!

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.

The Web Service wizard can create a Web Service, using a JavaBean generated by the iSeries Program Call wizard, to call one or more server programs or service program procedures on the iSeries.

Generate Java Bean for Web Services



Program Call Wizard

Create iSeries Program Call Java bean and PCML file

Specify the location where the Java bean and PCML file are to be generated.

Folder: Browse...

Package: Browse...

PCML file name:

Generate Java beans and a PCML file for:

Java application

Web services

These files will be generated by the wizard:

- myProgram.pcm1
- myProgram.mpcm1
- MyProgramServices.java
- myPgm.java

The class path of the specified project will be automatically updated.
Generated Java beans require the following JAR files at compile time and run time:

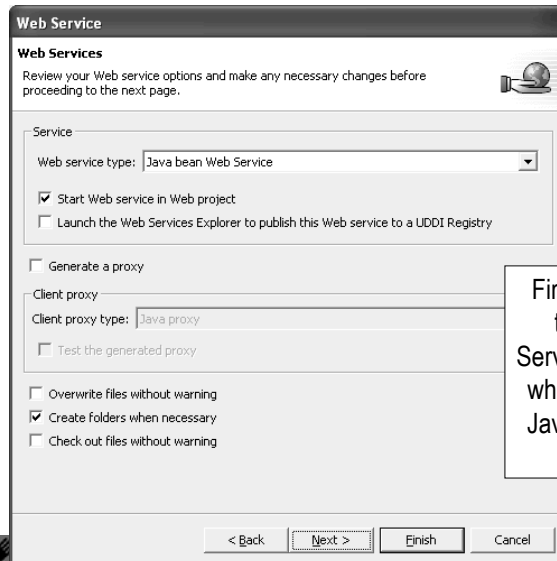
- WDCS_HOME/plugins/com.ibm.etools.iseries.toolbox_5.0.0/runtime/jk400.jar
- WDCS_HOME/plugins/com.ibm.etools.iseries.webtools_5.0.0/lib/wdbrt.jar
- WDCS_ECLIPSE_HOME/plugins/org.apache.xerces_4.0.7/xercesImpl.jar
- WDCS_WSTOOLS_HOME/plugins/com.ibm.etools.i2ee.core/runtime/i2ee.jar

< Back Next > Finish Cancel

© IBM Corporation

This is 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.

Web Service Wizard



Web Service

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

Service

Web service type: Java bean Web Service

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

Generate a proxy

Client proxy

Client proxy type: Java proxy

Test the generated proxy

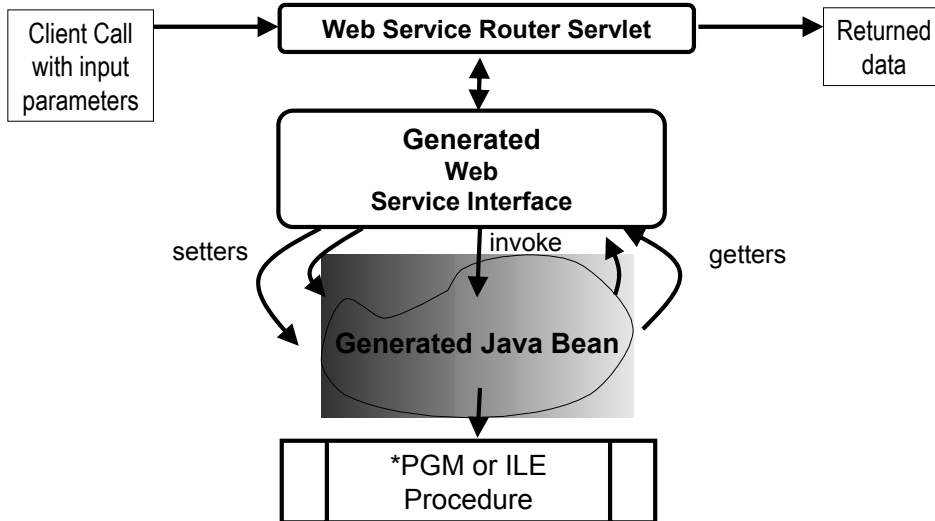
Overwrite files without warning
 Create folders when necessary
 Check out files without warning

< Back Next > Finish Cancel

First Page of
the Web
Service Wizard,
which takes a
Java Bean as
input

This is 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.

RPG / COBOL Web Services



On top of the base Web Services tools, Development Studio Client adds iSeries-unique support. Since the Web Services tools take a Java Bean as input, the Java tools Program Call wizard can be used to generate a Java Bean wrapper to an existing program or ILE procedure. By selecting the Web Services option in the wizard, the Java Bean 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.

Development Studio Client



XML
Tools

XML development tools support any XML-based development. The environment includes tools for building DTDs, XML Schemas, XML, XSLT, for mappings between XML and different back-end files such as an SQL query or relational table, and for debugging XSL code. These tools are inherited from Site Developer or Application Developer (for Development Studio Client Advanced).

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



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

- An editor for creating and visualizing XML files. This editor includes a wizard to generate a Java Bean to parse and generate the XML.
- An editor for creating Document Type Definitions, with minimal Document Type Definition (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.
- An editor for creating XML Schemas, which are replacing the older DTD's, with minimal XML Schema skills. Schemas can be deduced from sample XML files or DTD's. This editor includes a wizard to generate a Java Bean to parse and generate any XML conforming to the schema.
- 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 Development Studio Client and WebSphere Application Server.

XML Tools (continued)

- **XSL Trace Editor**

To visually step through an XSL transform

- **XML and SQL Query Wizard**

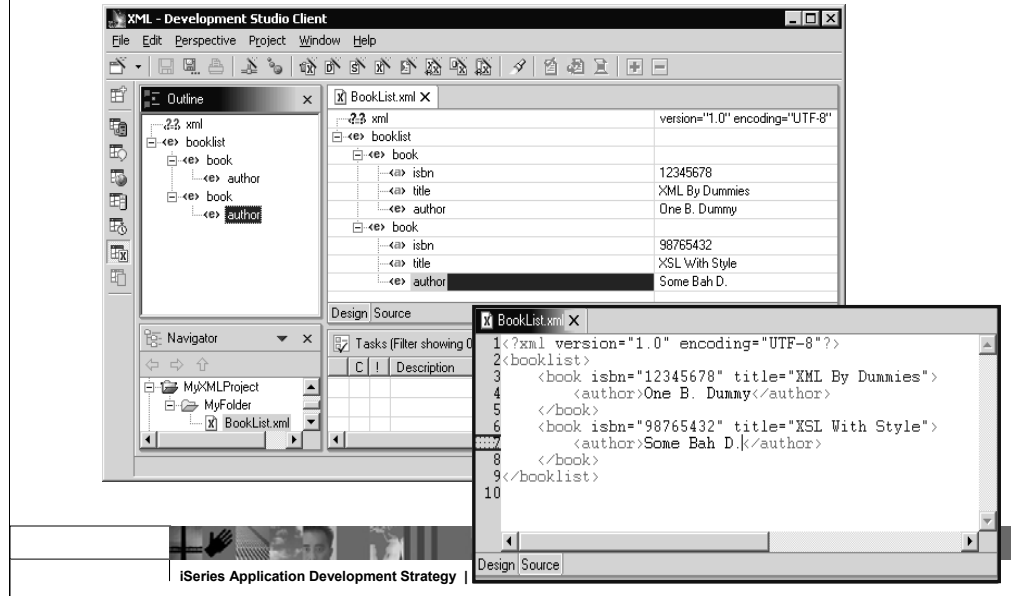
To create an XML file from an SQL query

- **RDB to XML Mapping Editor**

To map one+ relational tables to a target XML file

- 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.
- 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.
- 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 Editor



This is 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).

Development Studio Client

Database Tools

Database development tools (inherited from Site Developer or Application Developer (for Development Studio Client Advanced)) support any local or remote database that has a Java Database Connectivity (JDBC) driver. Database development tools contain the following features:

- Relational Schema Center- views to enable database, table, view, index, and key creation
- Online and offline support- metadata generated as XML
- SQL Query Builder- visually constructed SQL statements that perform SQL and XML mapping

Inherited 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**

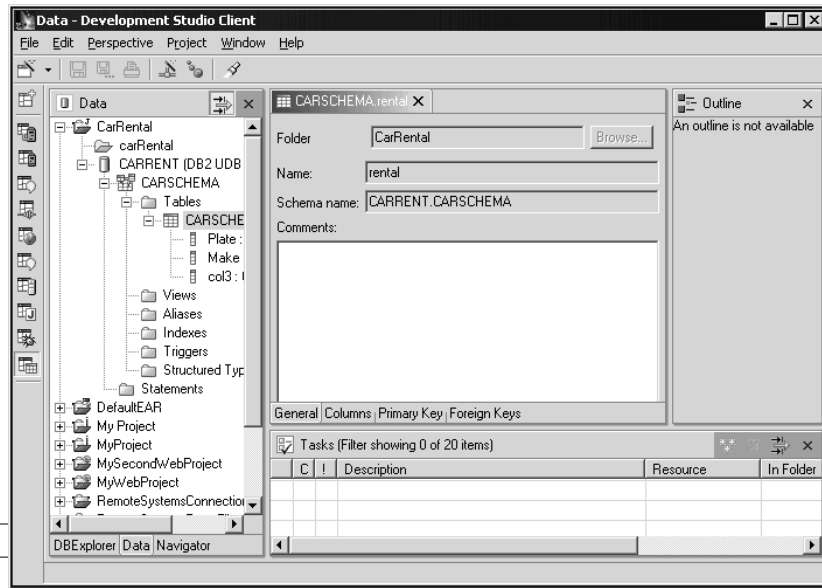
- Visually construct SQL statements
 - Insert, update, delete, select supported
 - Metadata generated as XML

- **SQL/XML mapping**

WebSphere Development Studio Client inherits a modest (stay tuned!) set of Database tools from Site Developer or Application Developer (for Development Studio Client Advanced):

- 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.
- A wizard for building SQL Data Manipulation Language (DML) statements or query statements, with minimal SQL skills.
- An SQL to XML mapping wizard, already described as part of the XML tools, but also available in the Data perspective

Database Tools



For Database development, there is a dedicated Data 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 if you are not connected to a remote server which is where the database will ultimately reside.

Table of contents

Strategy

WebSphere Development Studio Client 5.1

□□  **Summary**



Now you know all about the tools in Development Studio Client Version 5.0!

Summary



✓ Community



✓ Excitement



Today, iSeries programmers uniquely have a ubiquitous and comprehensive set of tools for all their application development needs now and into the future. Can you imagine developing for any other platform?



Trademarks & Disclaimers

© IBM Corporation 1994-2003. 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	IBM (logo)
AS/400e	iSeries
e (logo) business	OS/400
IBM	

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.



Disclaimer

- **Acknowledgement:**

This presentation is a collaborative effort of the IBM Toronto iSeries Application Development presentation team, including work done by:

Phil Coulthard, George Farr, Claus Weiss, Don Yantzi, David Slater, Alison Butteril, Linda Cole

- **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.

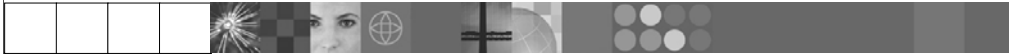




IBM Software Group

WebSphere Development Studio Client for iSeries Phil Coulthard

WebSphere. software



2003 | WDS V5R2 and WDSC 5.1

© 2003 IBM Corporation