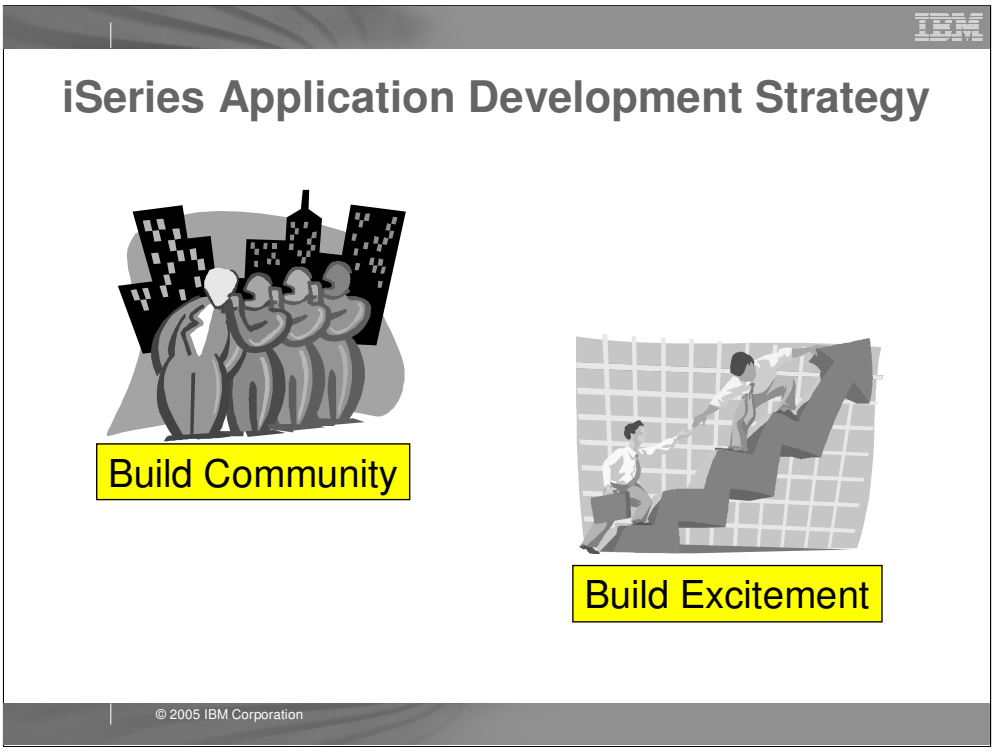

WebSphere Development Studio (WDS)
WebSphere Development Studio Client (WDSc)
© 2005 IBM Corporation

Table of Contents

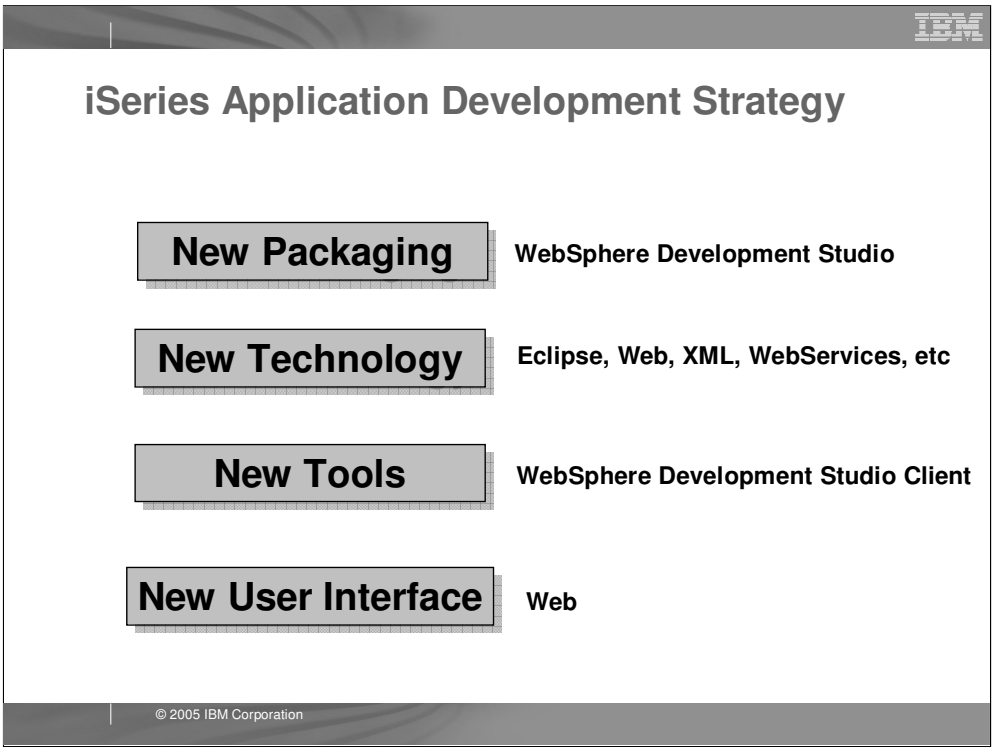


The iSeries Application Development team at IBM Toronto brings to you the RPG, COBOL, C, C++ compilers; the traditional toolset like PDM and SEU, and then new generation of workstation-based tools.

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.**

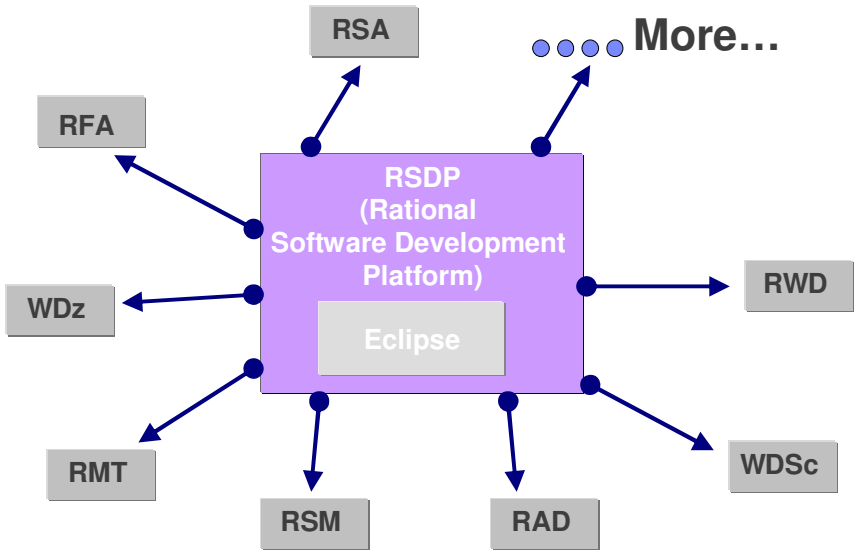


Here we see IBM's strategy is to reach its goals of building community and building excitement through 4 areas: New Packaging, New Technology, New Tools and New User Interface. The new packaging was actually done in **May 2001,** with the introduction of WebSphere Development Studio for iSeries, which 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.

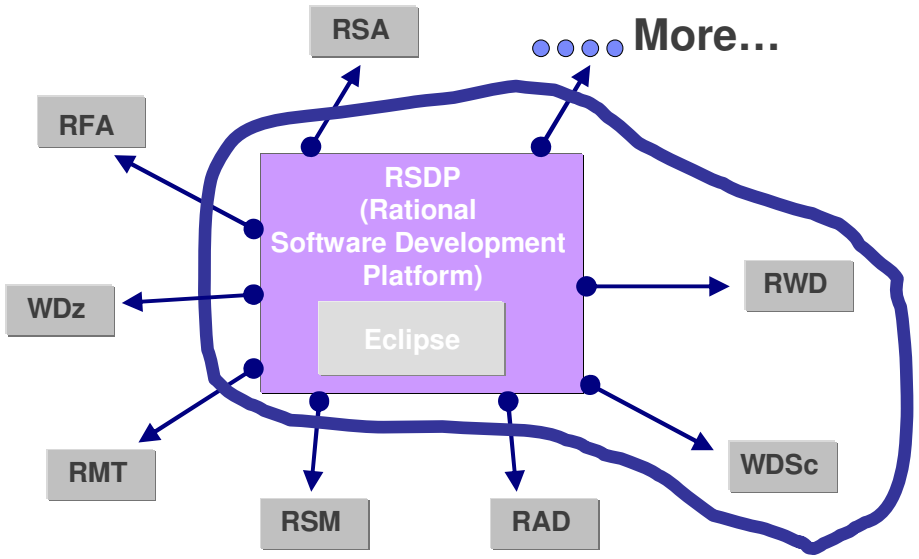
With the 4.0 release of the client tools in June 2002, we introduced new Eclipse technology and totally new Eclipse-based tools. Further, for technology we also plan to keep up with technology both in the compilers and the tools, so iSeries programmers are very current. This was evident in the V5R2 release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

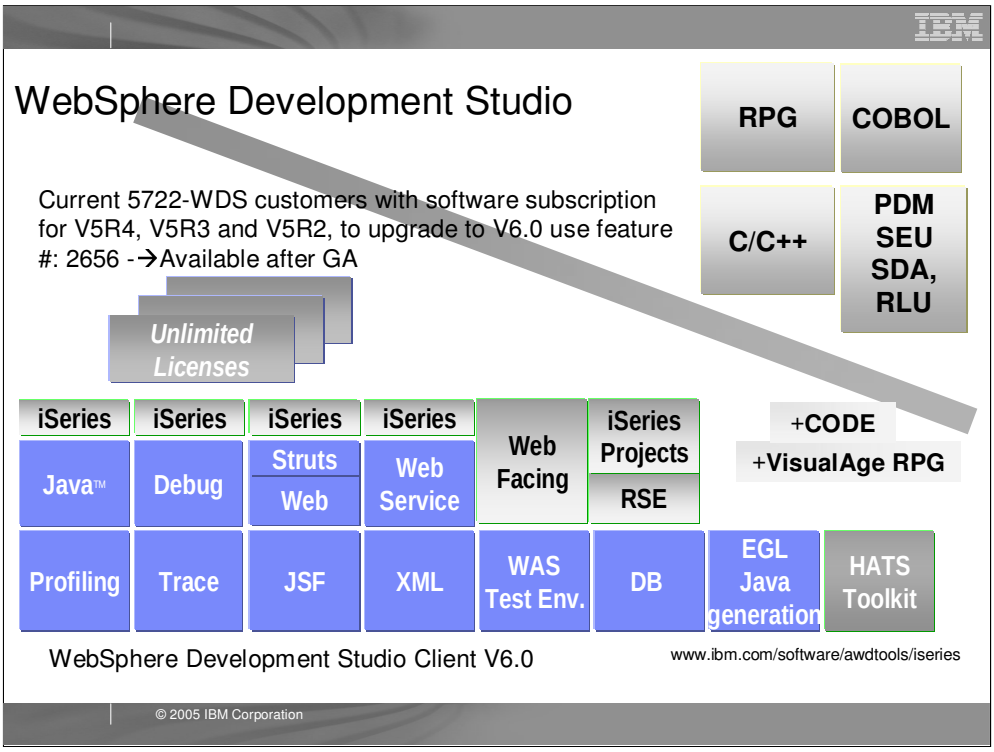
The preferred user interface for iSeries applications is now a Web browser, versus a 5250 green screen and **in addition to a desk top GUI**. There are exciting new and enhanced tools to make it easy to convert **5250 user interfaces** into Web user interfaces, and to build new Web user interfaces on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

Rational Family Tools … Componentization



WebSphere Development Studio Client



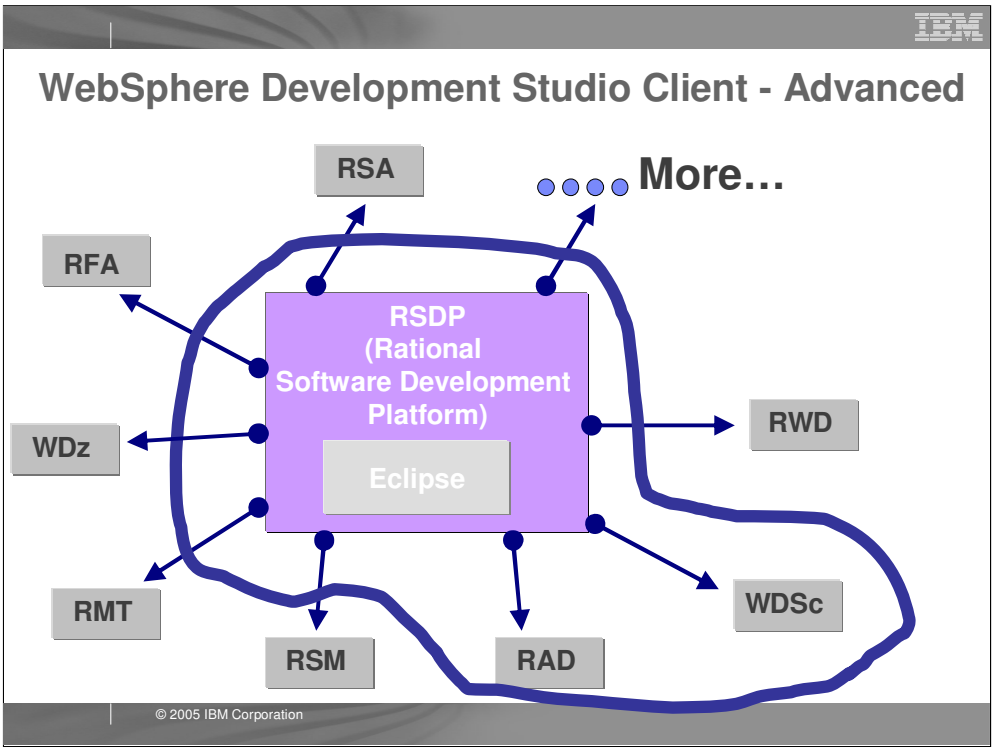


There is one family of application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and **unlimited licenses** of the workstation-based 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.



RSM – Software Model

RMT – Manual Test

RFT – Functional Tester

RSA – Rational Software Architect

WDZ – WSED)

WebSphere Development Studio Client Advanced Edition V6.0

Workstation License
order through Passport Advantage
http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home

+CODE
+VisualAge RPG

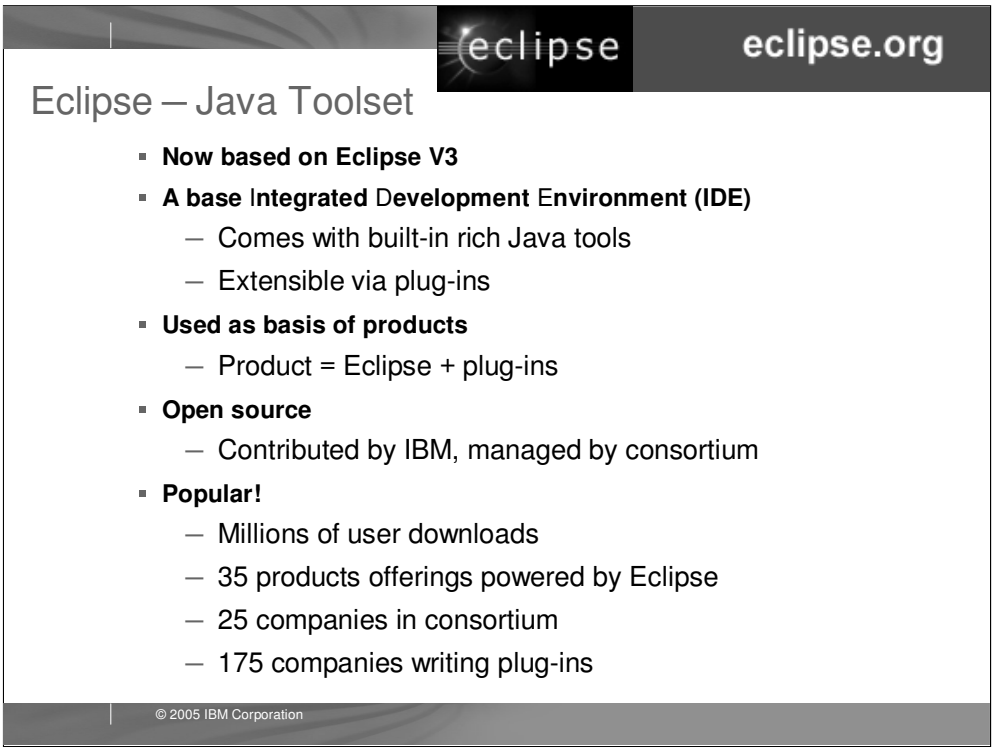
iSeries Java	iSeries Debug	iSeries * Struts Web	iSeries Web Service	Web Facing *	iSeries Projects / RSE		
Profiling	Trace	JSF	XML	WAS Test Env.	DB	EGL Java generation	HATS Toolkit
5724–D46			EJB * / J2EE *	Portal * Toolkit	Test * Cases	EGL * COBOL generation	

www.ibm.com/software/awdtools/iseries

General Comments

- **WDSc 6.0 is built with**
  - Eclipse 3.0
- **CODE/400 and VARPG shipped on separate CDs**
  - Optionally they can be installed
- **WebFacing does not depend on CODE/400**
- **Product co-exist with WDSc 5.x.x**

Table of contents

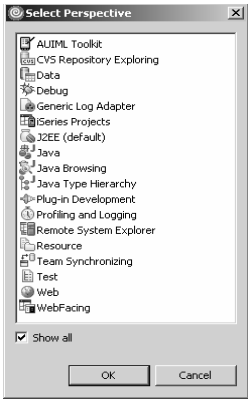


Eclipse was developed by IBM and **donated to the open source community**. That donation is estimated to be worth **$40 million**. Anyone can download Eclipse for free, including the source code, from **www.eclipse.org**. Eclipse has generated **extraordinary excitement** in the development community 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!

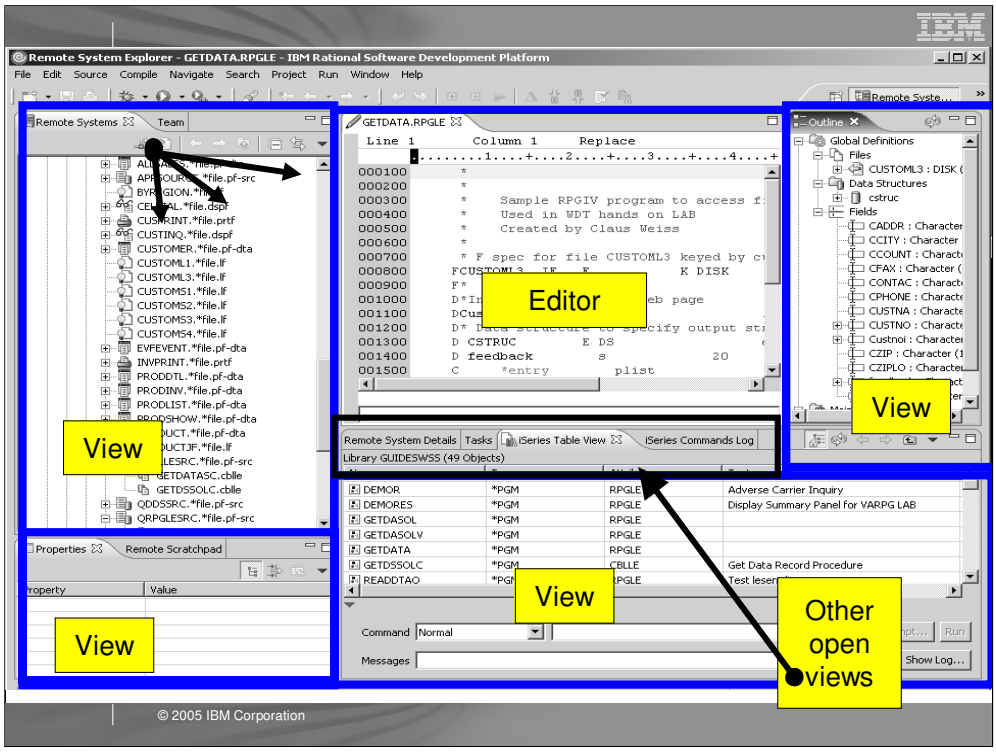
IBM is not alone with Eclipse. The open-source consortium that oversees contributions made to it include a number of large companies, and that list is **growing**. Note that **MKS** is an iSeries tool vendor. See **www.eclipse.org** for the latest list.

Eclipse Perspectives

- **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
  - –Views can be re-sized and re-positioned through drag ‘n drop



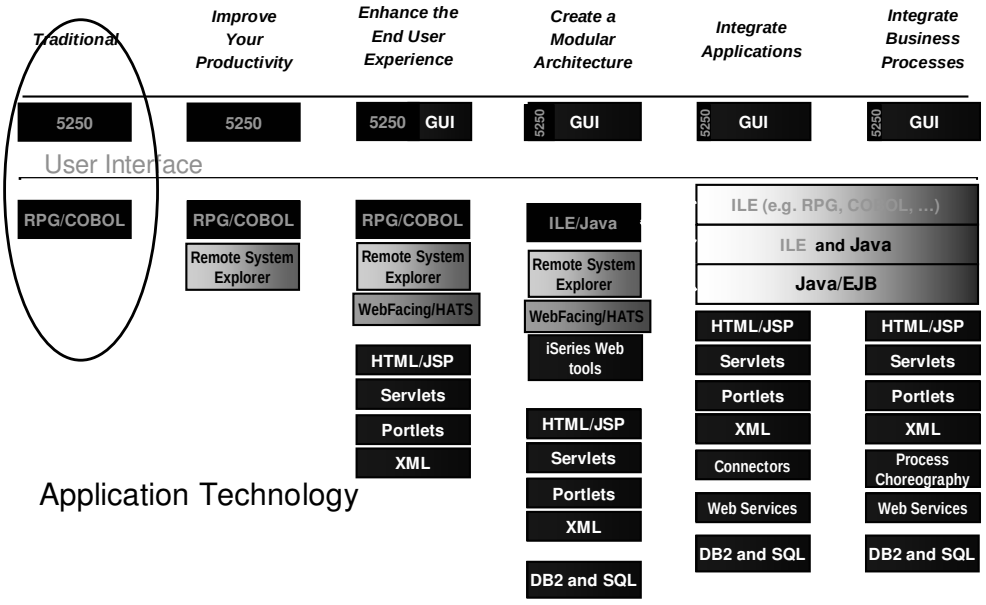
The **core features** of the Eclipse user interface include perspectives which is a collection of views and tools. Perspectives allow role based development. For example, if you are a Java developer you would use the Java perspective which includes tools and views for Java development. You can also **create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

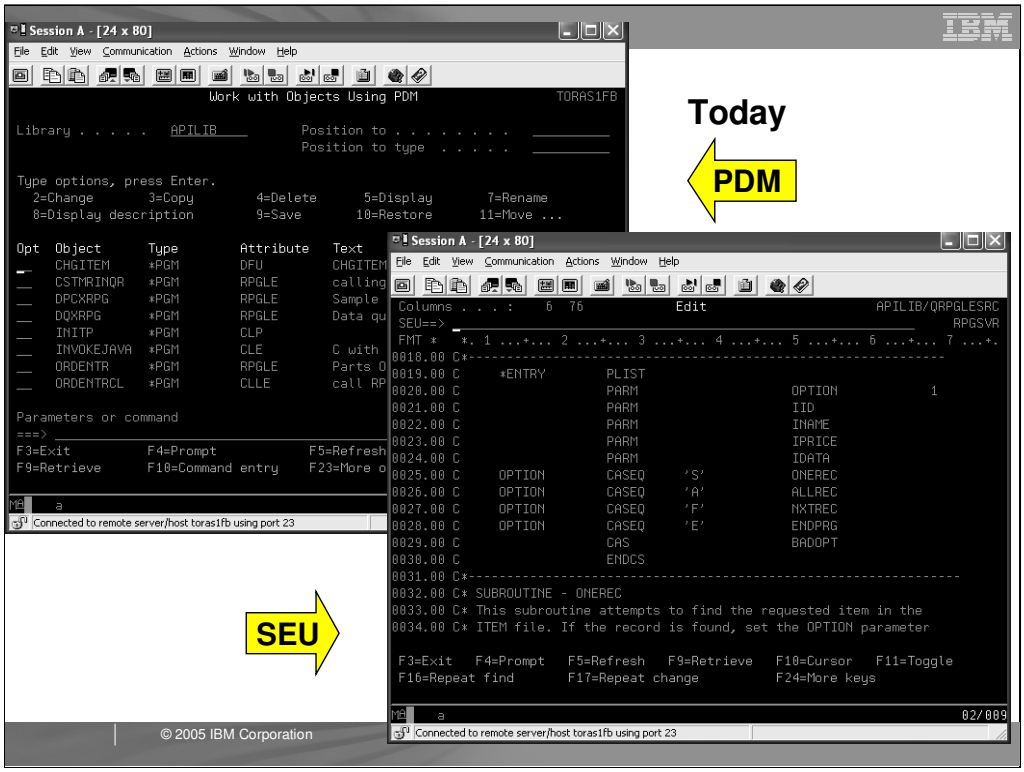

Remote System Explorer - GETDATA.RPGLE - IBM Rational Software Development Platform
File Edit Source Compile Navigate Search Project Run Window Help
Remote Systems
Team
GETDATA.RPGLE
Outline
Editor
View
View
View
View
Other open views
Remote System Details
Tasks
iSeries Table View
iSeries Commands Log
Properties
Remote Scratchpad
Command
Normal
Messages
© 2005 IBM Corporation

AGENDA

- Strategy & Packaging
- Eclipse & WebSphere Studio family
- **WDSc AD Developer RoadMap**
- V5.1.2 Enhancements
- V6.0 Enhancements
- Future Enhancements
- Conclusion

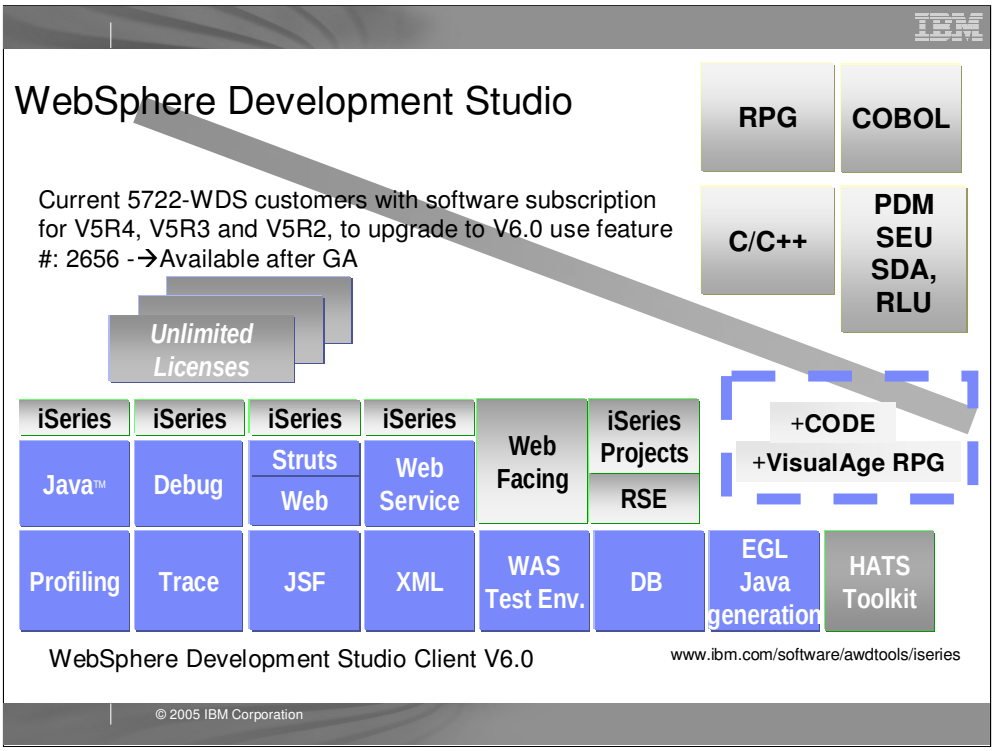
iSeries Developer Roadmap





Today…

Here, you can briefly look at a green screen display for PDM and for SEU. We are sure you are already familiar with these development methodologies. But keep your imagination open and receptive to the use of a “Better tool”…



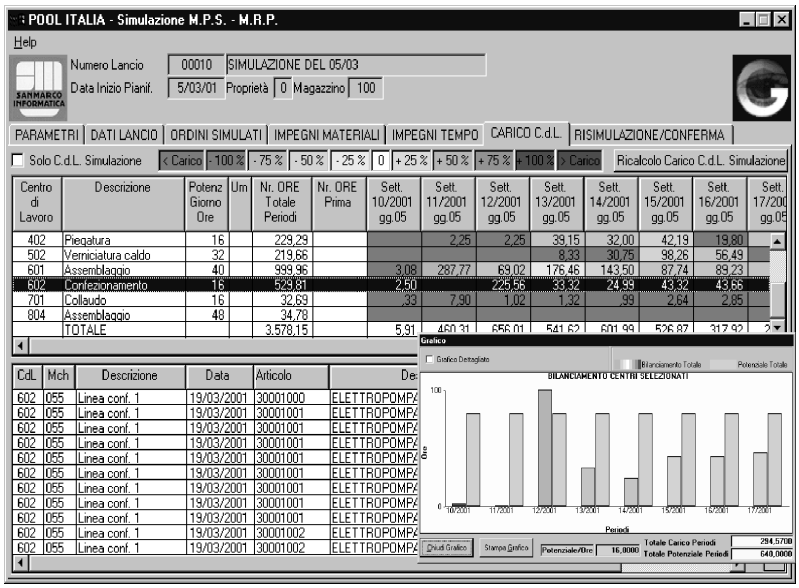
There is one family of application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and **unlimited licenses** of the workstation-based 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.

San Marco (Italy) ERP sample - VARPG



What's New in VARPG for V5.1.2

- Java generation
  - **support of JDK1.4**
  - **Applet support now also when VARPG runtime is not locally installed**
- Invocation of Java methods from Windows runtime
- Allow hide/re-show cursor for window part
- Media part: ShowPlyBar attribute added
  - **to hide control tool bar**
- *Component part: Added DlgPrompt, SelFolder, FolderName attributes
  - **These attributes will bring out a standard Windows folder-selection dialog and allow the user to choose a folder**
- Added memo datatype to ODBC support

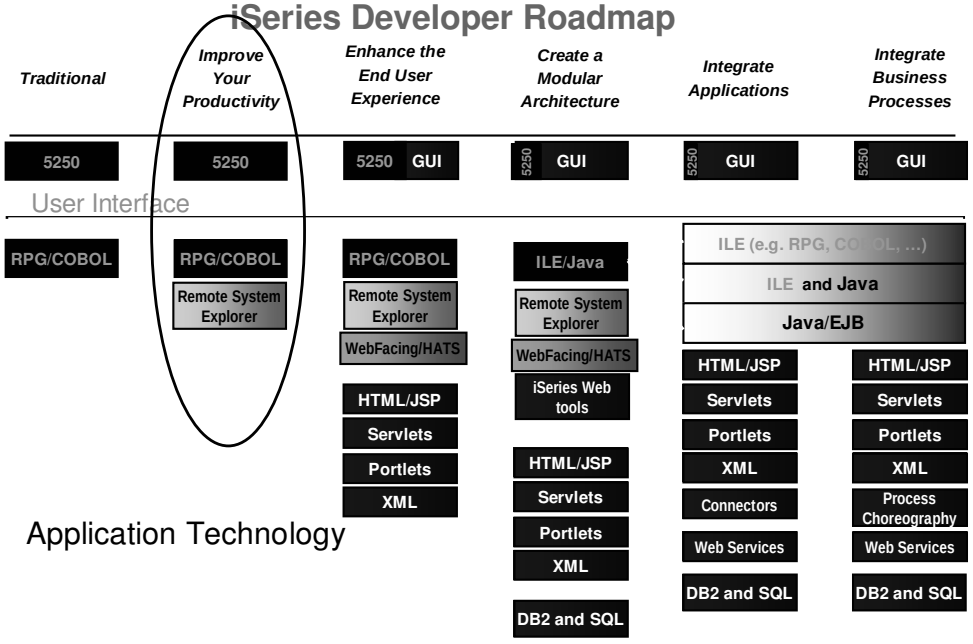
New in VARPG for version 6.0

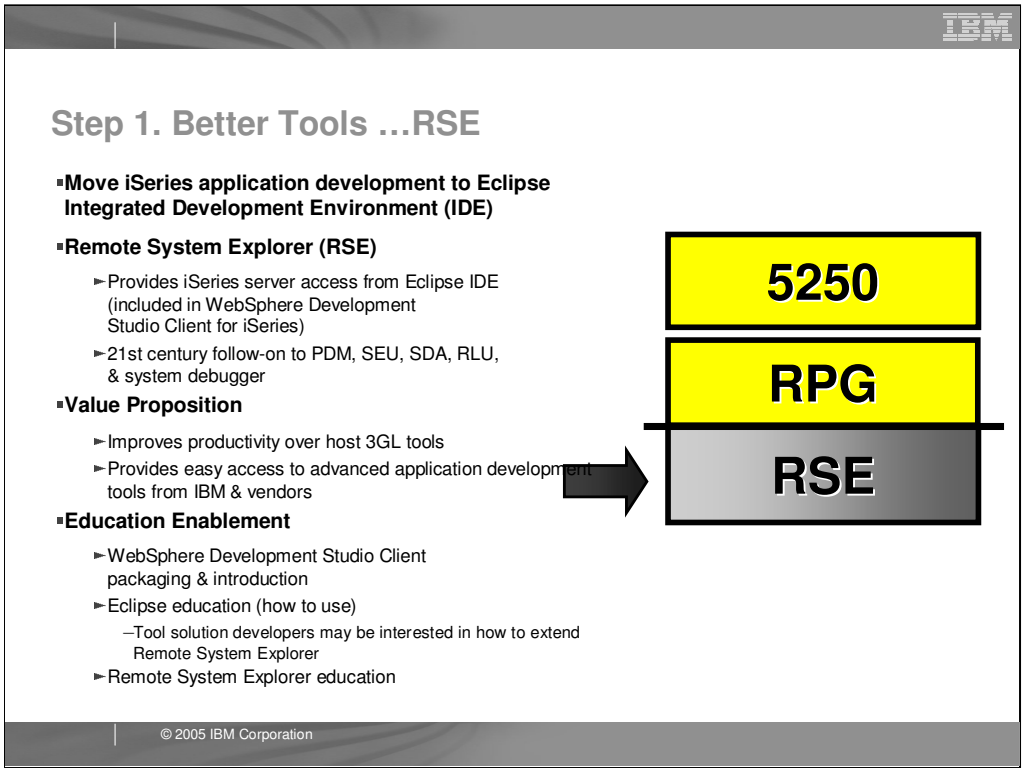
- **Language level has been updated to V5R3. The following function is now available in Version 6.0**
  - New built in Function %SUBARR
  - Direct conversion of date/time/timestamp to numeric using %DEC
  - Second parameter for %TRIM, %TRIMR and %TRIML indicating character to trim
  - New prototype option … OPTIONS(*TRIM) to pass trimmed parameters
  - Relaxing the rule for using result data structure for I/O to externally-described files and record formats

What about CODE/400?

- 99% of the CODE/400 Functionality is in Remote System Explorer… We will cover this in few minutes!
- Trying to totally move CODE function into eclipse and WDSc

iSeries Developer Roadmap



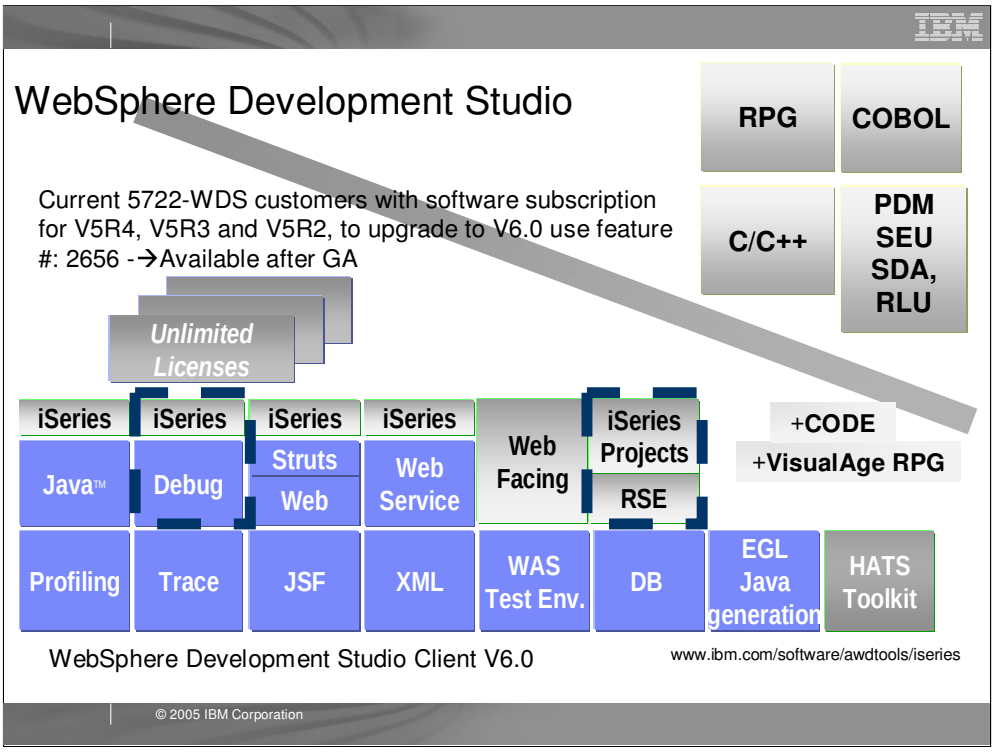


Step 1. Better Tools

The first step in the J2EE roadmap does not involve any change to the applications in use within your organization. Rather, it allows you to simply replace your traditional development tools with more exciting and modern tools to support the same code base.

RSE is the follow-on to PDM, SEU, SDA, RLU, and the system debugger. It offers highly productive follow-on's to these "primitive by today's standards" tools. And these new RSE tools are all highly integrated with each other and with the Integrated Development Environment of Eclipse. RSE is also the 21st century Product Data Management (PDM) tool in the sense that it is the point of integration for all iSeries tools vendors, many of whom have already released plug-ins to complement the functionality provided by IBM with additional capabilities.

RSE comes with WebSphere Development Studio Client. By learning and using RSE, PDM programmers will enjoy greater productivity than with host-based 3GL tools. Programmers gain transferable skills that later apply to other Eclipse-based tools in WebSphere Development Studio Client. Learning RSE also opens opportunities to access the next generation of third-party tools that are built on top of Eclipse. Further, RSE works not only with OS/400® files, commands, and jobs, but also with IFS files and Qshell commands, and with Linux files and commands that reside in their own logical partition



There is one family of 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).

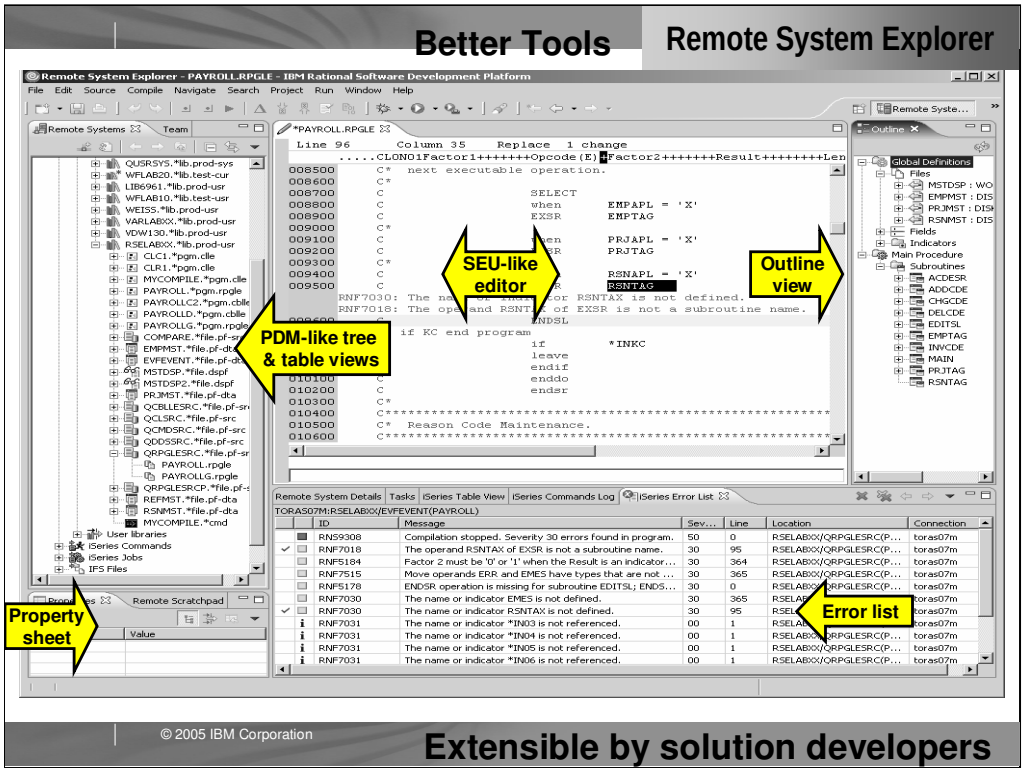
If you are an existing customer who has a **subscription**, you can upgrade to Development Studio free of charge. Without a Software Subscription, there is an upgrade fee. New licenses of Development Studio are priced very **competitive compared to the combined prices of all constituent products**. As 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.

Better Tools

Remote System Explorer and Debug V5.1.2

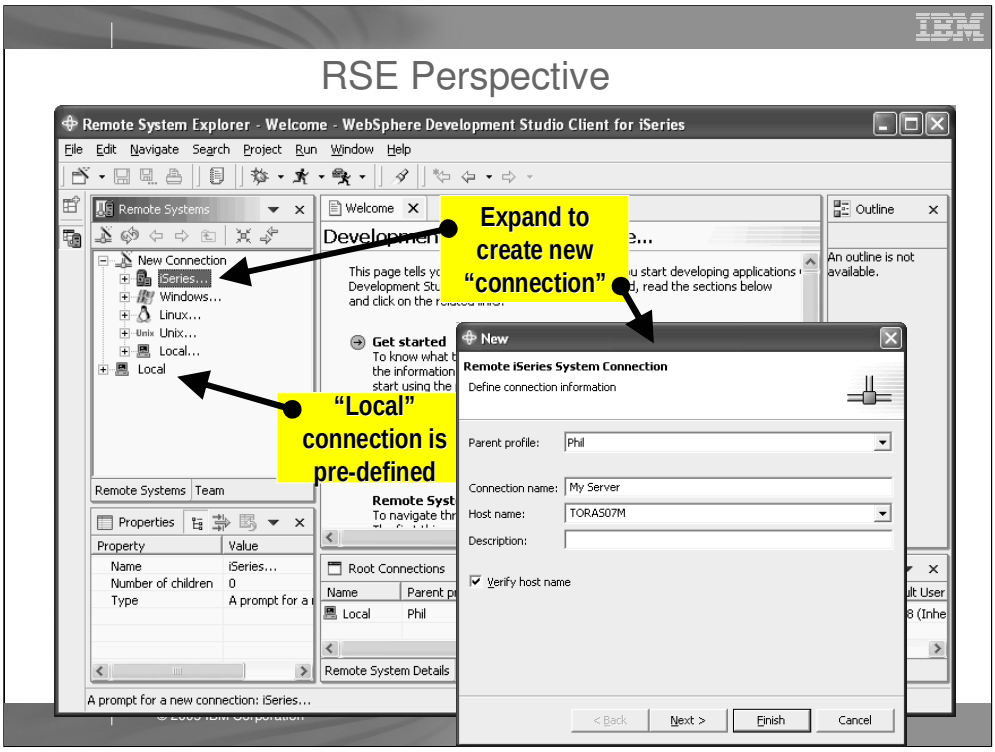


Better Tools – RSE

Examine this screen capture of Remote System Explorer. The collection of views and editors in RSE are called the **Remote System Explorer "perspective."** Developers can open multiple perspectives and flip between them via icons in the bar on the left.

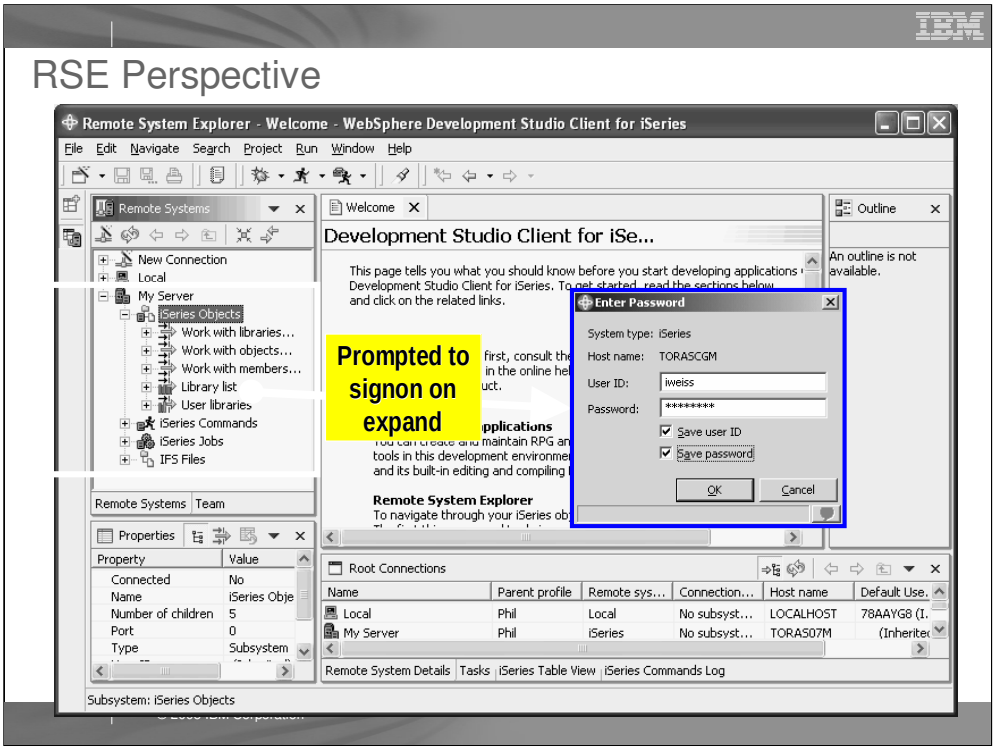
The primary view in the RSE perspective is the Remote Systems perspective, where you **create connections to remote iSeries, UNIX, Windows, or Linux servers.** This screen shows a connection to an iSeries server whose OS/400 objects are expanded so you can work with them, as well as with members… similar to PDM. Look more carefully at the "My iSeries" tree—you can also work with **commands, jobs, and IFS files**. As you select objects in the Remote Systems view, the property sheet (lower left) shows information about the selected object (which here is RENTCAR). Some of the object's property sheet information is **directly editable—right there, for your convenience.**

Many **right-click actions** are provided for all object types and members—including source members. The option to open the member in the RSE editor is shown on this screen capture. Beyond what IBM supplies, **user-defined actions are possible**. The editor is rich in function, far exceeding SEU while retaining features such as **entering "D" in the prefix area** to delete a line. The editor supports **syntax-checking and cursor-sensitive F1** language help. It also



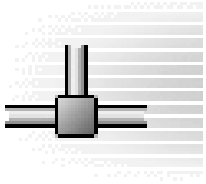
When you first open the Remote System Explorer, you are not connected to any system except your local hard drive on our workstation. To connect to a remote iSeries host, you need to define a connection. In the Remote Systems view you use the New Connection wizard to configure a connection to your iSeries host. You can also use the Remote Systems view to explore the file system of remote Linux, UNIX and Windows systems. The Local connection is pre-defined.

RSE Perspective



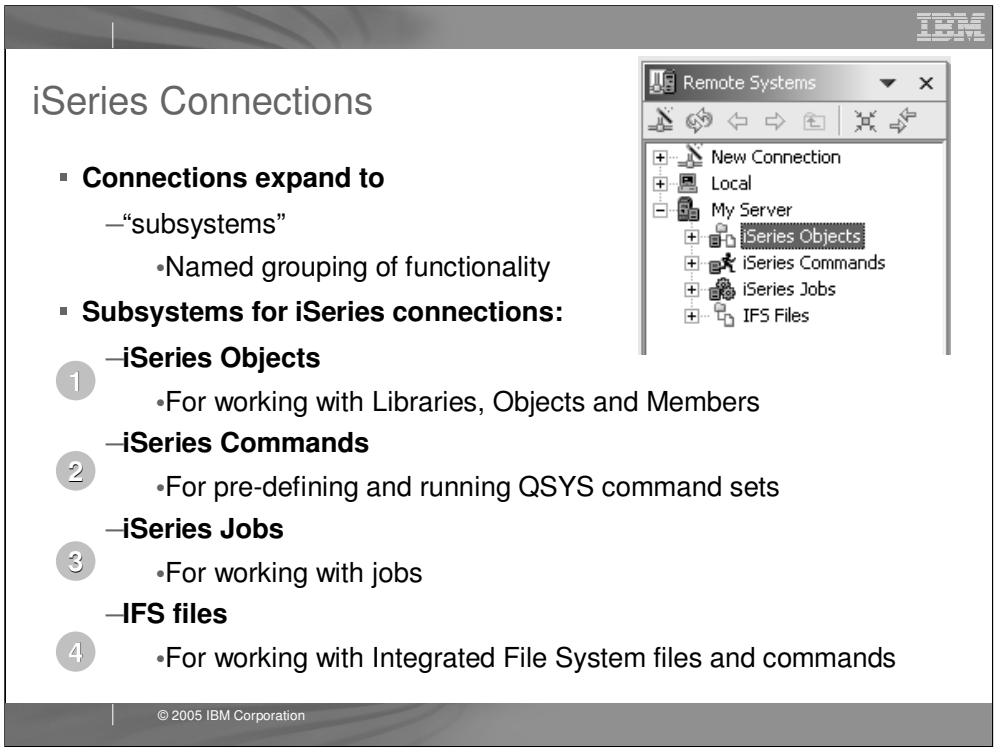
After creating an iSeries connection, and then expanding it to the point where stuff from the iSeries is to be shown, you are asking to signon to the iSeries. You can optionally choose to remember the user ID and password, such that you won't be prompted again for them.

What is an RSE “Connection”?



- **Represents a remote system**
  - Given an arbitrary name
  - Multiple connections to one system permitted
- **Contains environment info**
  - Such as library list and environment variables
  - Specified in properties dialogs of connection
- **Used in many Development Studio Client for iSeries tools**
  - RSE, iSeries Projects, Java Tools, Web Tools, WebFacing

A very central concept to all of Development Studio Client for iSeries is that of connections. A connection defines information needed to access a remote system. Each connection is given an arbitrary name by you, and so multiple connections to the same system are permitted. Each connection also captures information that is applied when connecting to that remote system, such as the initial library list for iSeries connections. All iSeries tools within Development Studio Client for iSeries use connections to access a remote iSeries system. Connections are created and managed in the Remote Systems Explorer.



Once connections are defined they can be expanded within the Remote Systems Explorer. On expansion, the user sees subsystems, which are merely a functional grouping of the various types of remote resources that can be explored in the remote system

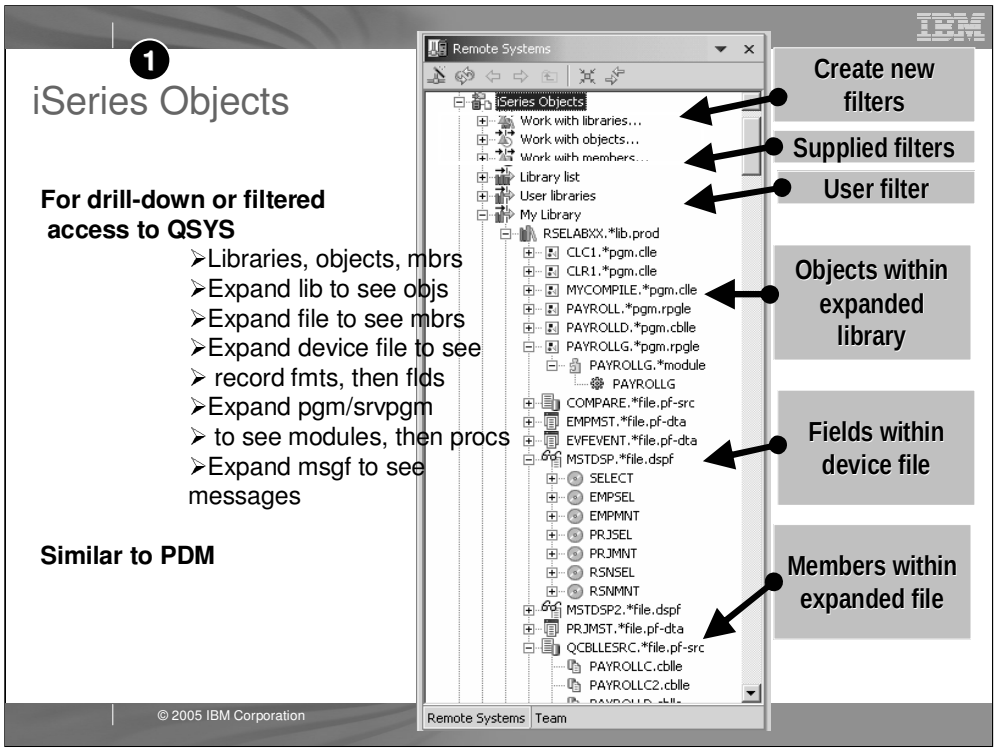
For iSeries connections, there are four subsystems:

•iSeries Objects is the PDM-like grouping, allowing access to libraries, objects and members

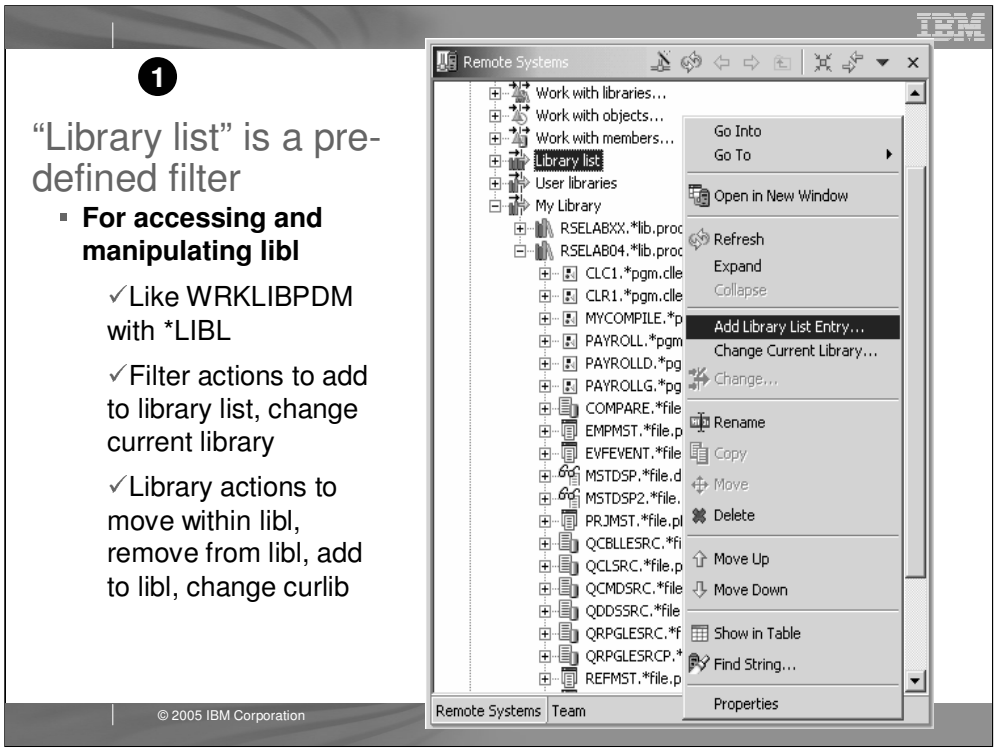
•iSeries Commands allows developers to predefine command sets each of which contain one or more often used commands. When run, all commands in a command set are sent to the remote system and executed, and the results are logged in the Commands view.

•iSeries Jobs allows developers to see various jobs, subsettable by job attributes, and to perform a limited number of operations on those jobs

•IFS Files allows developers to explore folders and files in the Integrated File System of the remote iSeries system

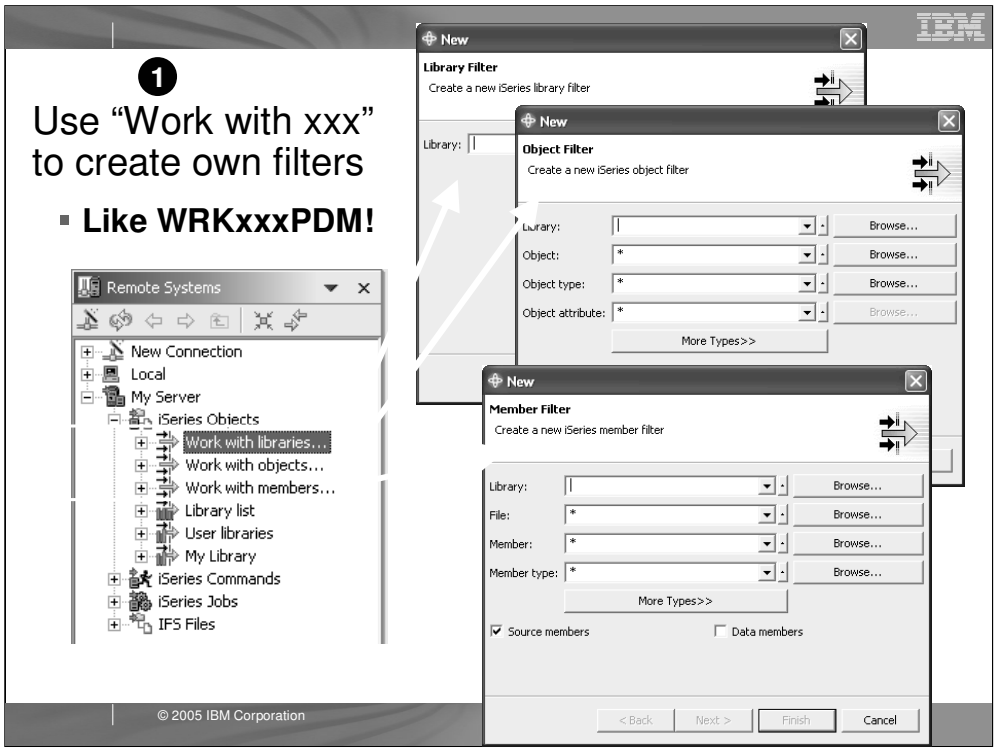


The iSeries Objects subsystem is the subsystem you will use most often! It is very similar to PDM, in that it allows you to access objects in the QSYS file system, and perform actions on those objects.



When the pre-defined library list filter is expanded, and the connection is successful, you will see the libraries on your library list. For each library, you can right-click and select from a number of useful actions. There is an action to create a new source file within the selected library, to refresh the contents of the library if it is expanded, to rename the library, copy the library or delete the library. These last three actions remotely run the appropriate iSeries command and you will see it logged in the Command Log view.

If you expand a library, you will see all the objects in that library...

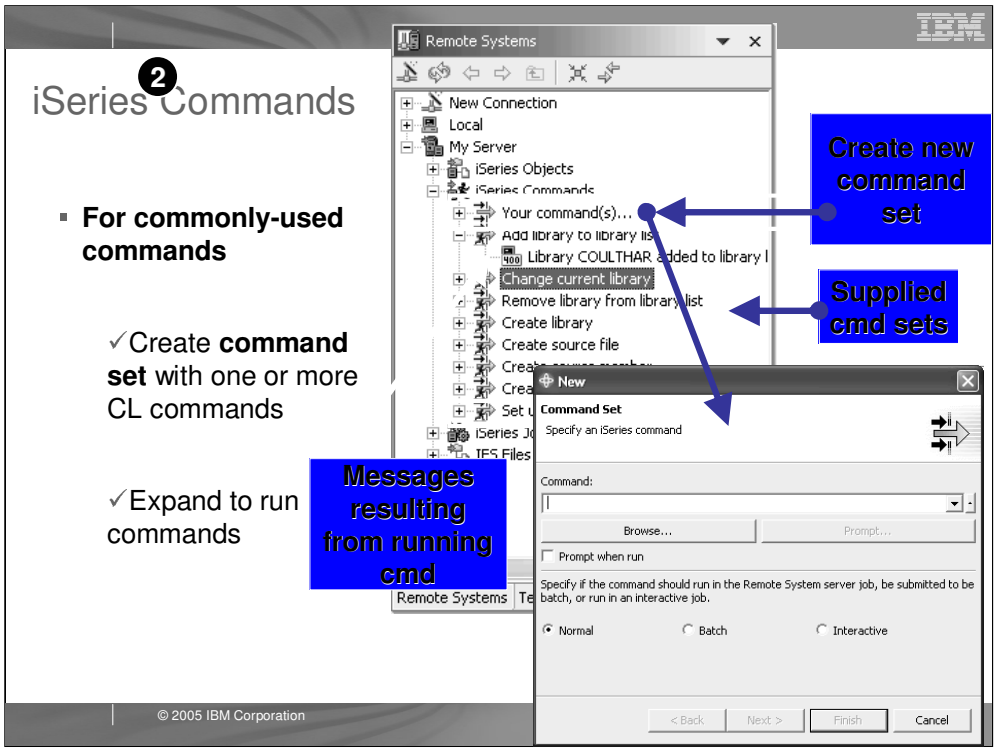


The three child items at the top of the list are for creating filters, much like in PDM:

-Work with libraries... prompts you for a simple or generic library name, and lists all matching libraries. It is similar to WRKLIBPDM.

-Work with objects... prompts you for a simple or generic library name and simple or generic object name, as well one or more object type and attribute pairs. It lists all matching objects in all matching libraries. it is similar to WRKOBJPDM.

-Work with members... prompts you for a simple or generic library name, simple or generic file name, and simple or generic member name, as well as one or more member types which can also be generic. It lists all matching members in all matching files in all matching libraries. It is similar to WRKMBRPDM. Unlike PDM, the filters you create are permanently remembered and displayed in this list for easy re-use. We will have more to say about filters. To simulate STRPDM's option 12, you can start with the pre-defined Library list filter, that when expanded lists all libraries in your library list. With any filter, once it is expanded you can subsequently expand a library to see all objects in the library, and expand files to see all members in the file. When you expand your first filter, such as the pre-defined Library List filter, you are prompted for your password and then connected to the remote iSeries. Then, the results of resolving the filter are shown...



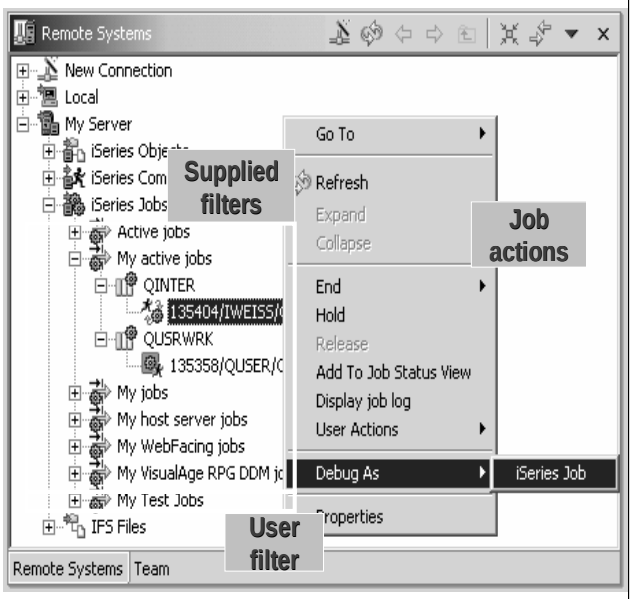
The iSeries Commands subsystem is merely a place for creating often-used commands. The commands are created in a Command Set, which is simply a named list of commands. By default you are only prompted for a single command, but after creating the command set you can use the Change action to add more commands.

Once a command set is created it appears in the tree. There are a number of command sets pre-supplied by IBM. When a command set is expanded, the commands in it are run and any messages are shown as children of the command set. The commands are also logged in the command log view.

This is only one way to run commands in the RSE. There is also a Command Log where you can enter commands just like a command line on the iSeries. You will see that you can also create user-defined actions that appear in the menu for selected objects and members, just like PDM user defined options.

iSeries Jobs 3

- **For accessing jobs**
  - ✓Like WRKxxxJOB
  - ✓Create filters to see jobs with specific attributes (right click)
  - ✓Use predefined filters to see your own jobs

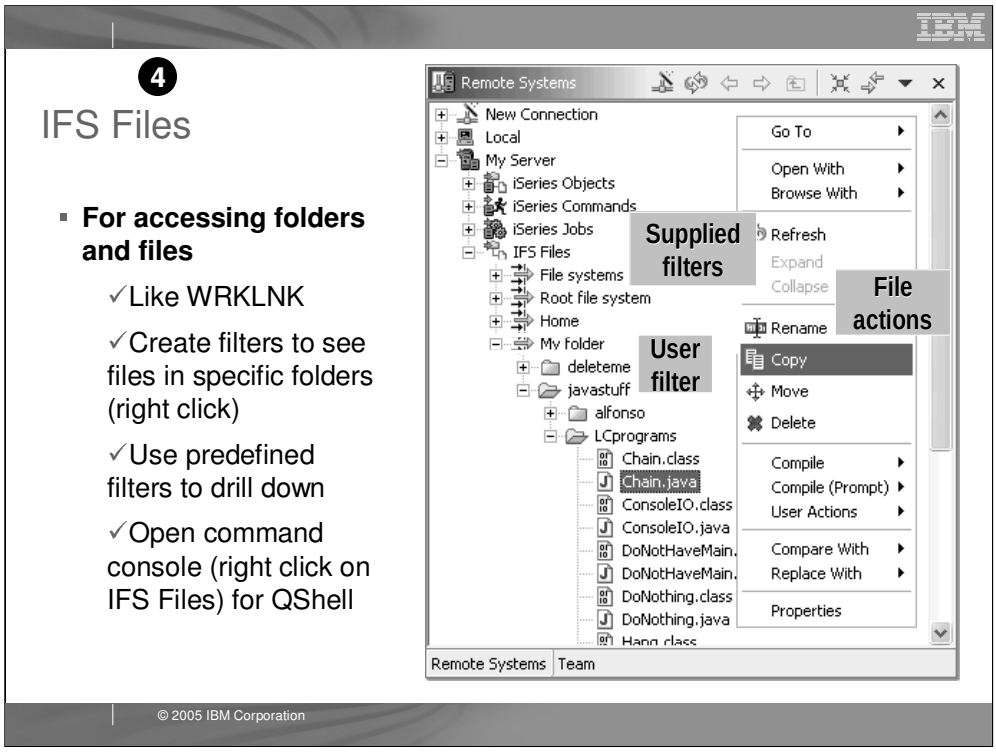


The iSeries Jobs subsystem is for working with jobs.

As with all subsystems, you can create filters (by right clicking on iSeries Jobs) to see just the jobs you want to see. There are three pre-defined filters for you to see all active jobs, only your active jobs and all your jobs.

The job's icon indicates if it is active, done or queued.

Right clicking on a job allows you to work with the job.

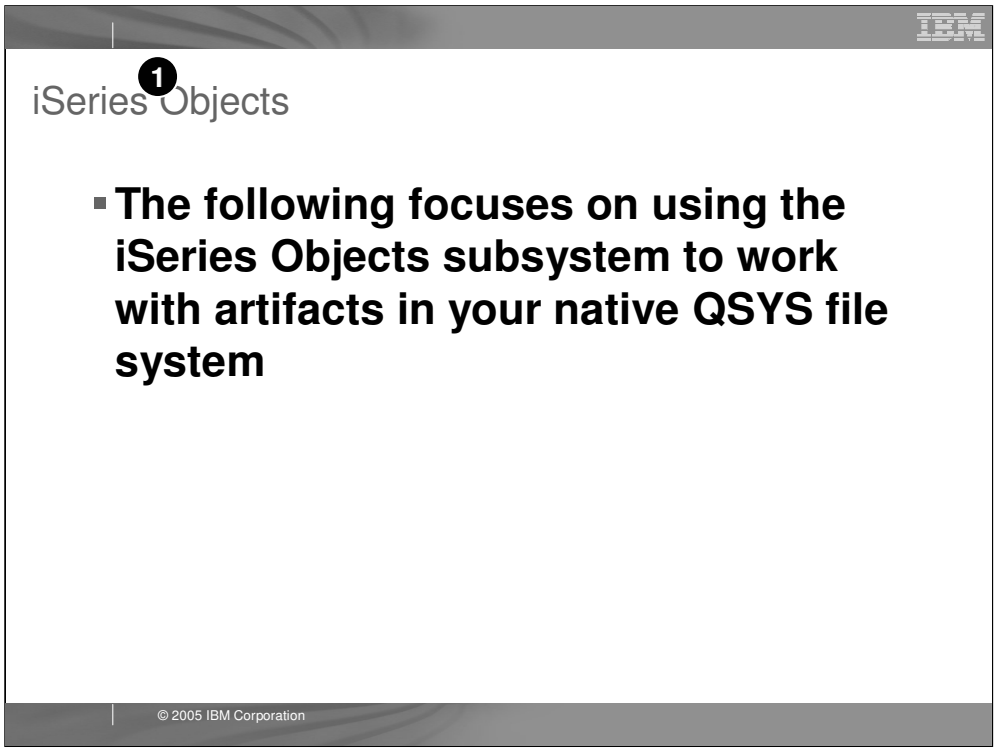


The IFS Files subsystem allows you easy access to IFS folders and files.

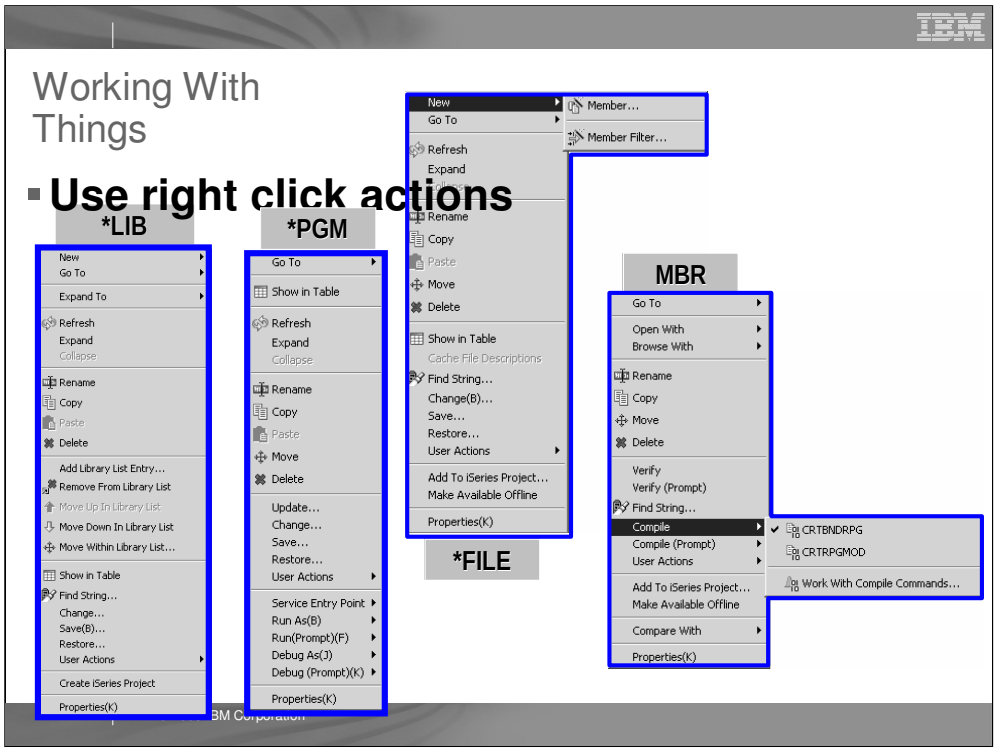
Like all subsystems, you can define your own filters and there are some pre-defined. In this case, the filters allow access to files within a particular folder.

Right-clicking offers access to a very rich set of actions you can perform against the selected folders and files. You can even create your own user actions, as can for the iSeries Objects and iSeries Jobs subsystems.

To run QShell commands, right click on the IFS Files subsystem object and select the Launch Shell action. This opens a command console that is also very rich in function.

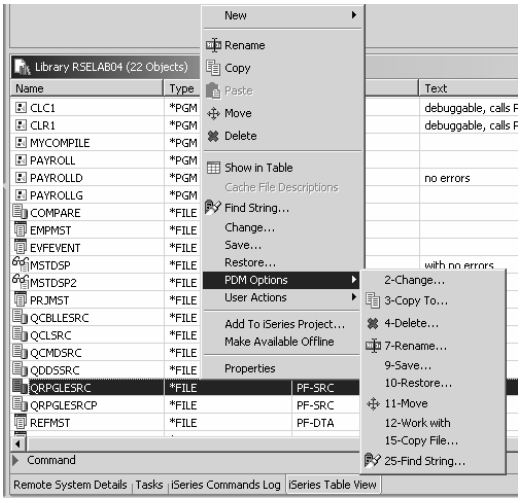


Now we drill down on the iSeries Objects subsystem, which is by far the most heavily used.



These are some of the right-click popup menus, based on object type. This shows how rich the functionality is. Indeed, it is a superset of PDM's functionality.

iSeries Table View

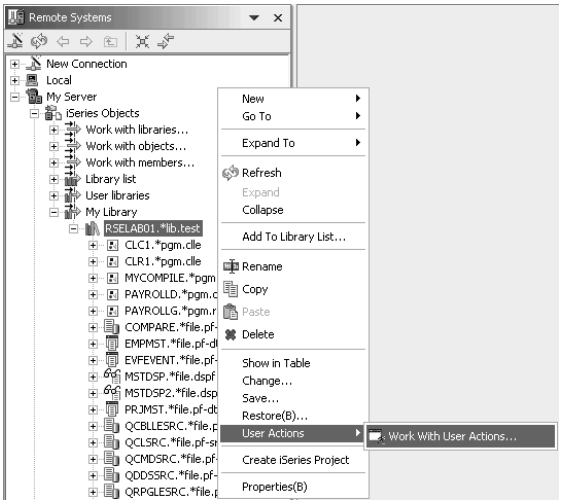


✓**Special PDM Options menu for PDM users**

While there is nothing new in this menu, there is a PDM Options menu in the right click menu from the table view, which shows all the actions by the numbers their corresponding option was in PDM, to help with the learning curve of PDM users.

RSE: User Actions

- User-Defined Actions (like PDM!)
  - Right-click on iSeries Objects or any iSeries object member
  - **Work With User Actions**
    - Create, delete or change user-defined actions
    - Scope them by type and attribute

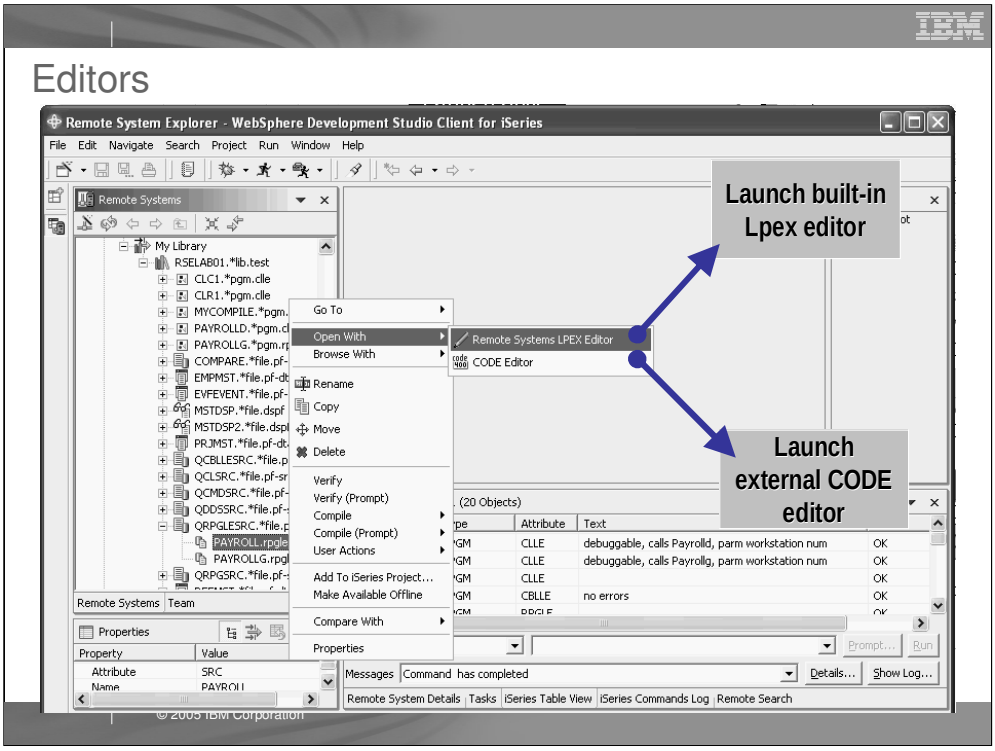


While IBM supplies a number of useful actions for remote iSeries objects, it is not possible to supply them all. Like PDM, you can easily define your own actions. To create your own actions, use the Work With actions in the popup menu for iSeries Objects. These user-defined actions will appear in the popup menus for remote resources. To avoid seeing all actions in all popup menus, you scope each action to a one or more object or member types. You first define named collections of object or member types, then you create your actions and scope them to one of these named collections of types. You actions will then only appear for object or members that match one of the types in the collection. If you are a CODE user, you can use File->Import to import existing actions from CODE Project Organizer.

You can also create user actions for Jobs and for IFS (or local/windows/unix/linux) folders and files.

Editors

Editors

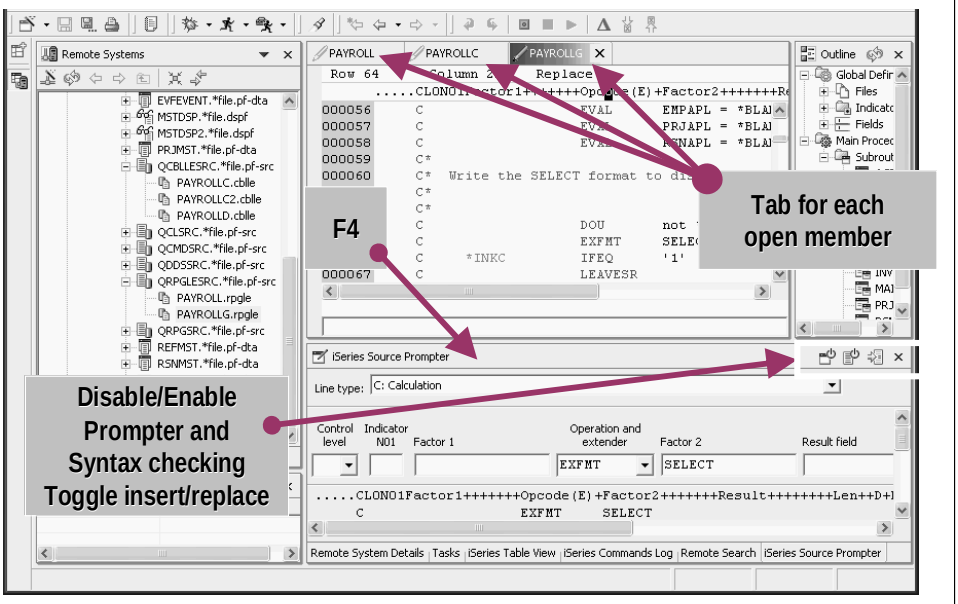


For a source member, there are two options for editing:

1. Remote Systems LPEX Editor. This is the new editor, written all in Java, that is built-in to the IDE. It is a re-write of the original CODE Editor, but as you will see has a subset of the functionality in CODE at this point.
2. CODE Editor. This is the classic full-functioned CODE editor, which is offered as an alternative until the Lpex editor catches up to the functionality of the CODE editor. This launches the CODE Editor in a separate window.

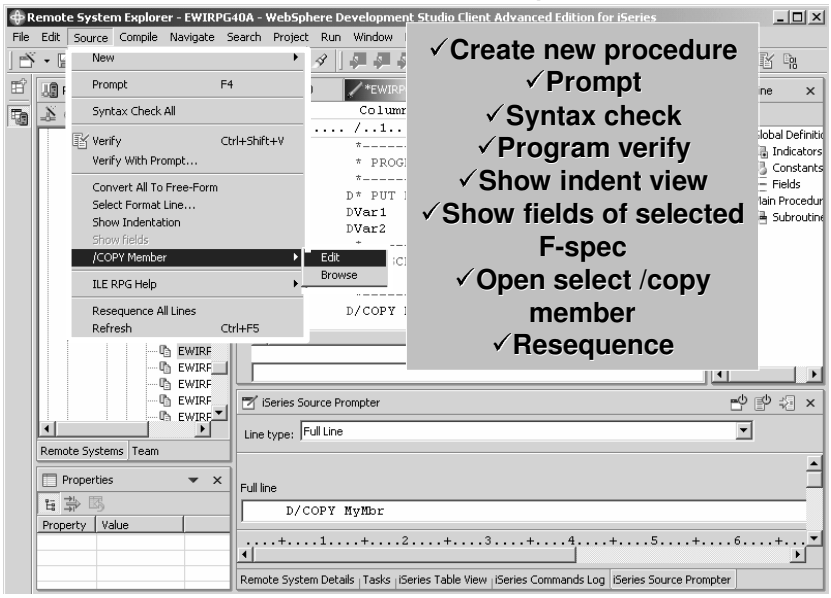
We will cover the Remote Systems LPEX editor next in more detail...

LPEX (“JLPEX”) built-in IDE editor

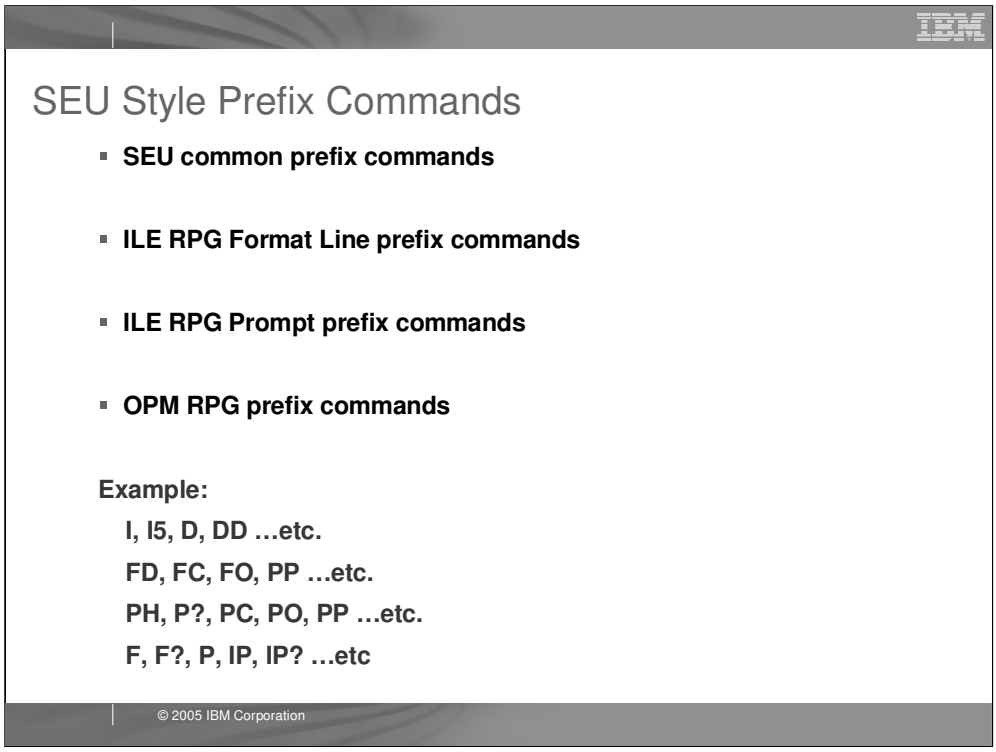


The Remote Systems LPEX editor is built-in, so it shows up in a pane within the IDE. You can open multiple members for editing, and each will be shown in the editor area with a tab that when selected brings that member to the foreground. You can double click on a tab to expand that member's edit window to full size. When a tab shows an asterisk in it, that indicates there are pending changes that should be saved. For RPG (both III and IV) you will notice there is color highlighting and familiar F4 support to prompt for the current line. The prompter sits in a view that doesn't overlap the editor. When done filling in the prompt, you can press one of two buttons to replace the current line or insert a new line.

LPEX Editor Functions: Source pulldown



This is the Source menu when the editor is open and in focus. It has a number of actions that apply to the entire source member, as opposed to what is selected.



You can configure the LPEX editor to adopt the keyboard and command personalities of many popular editors. Most editor profiles differ only in the keys and commands used to perform various editor tasks. Some base editor profiles, listed below, also add a prefix information and command area at the start of each line:

ispf

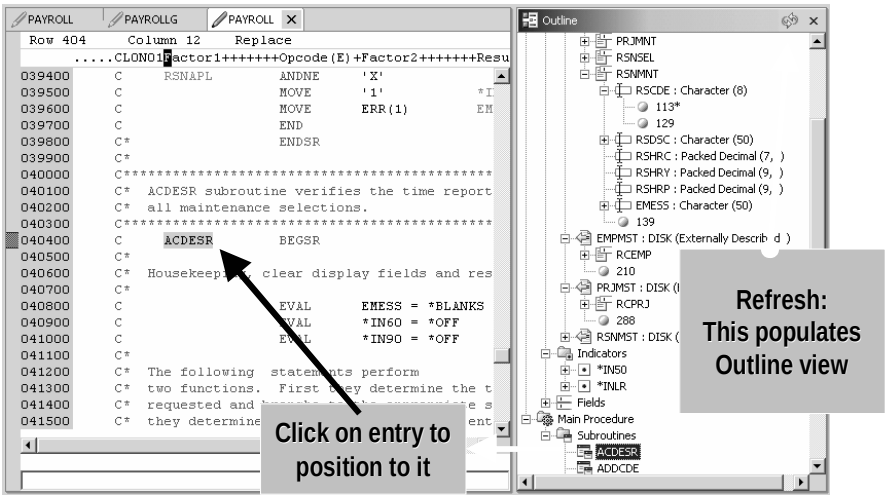
seu

xedit.

The editor recognizes prefix commands used by these editor profiles. Depending on which profile you are using, you can enter SEU, XEDIT, or ISPF commands when the prefix area is active.

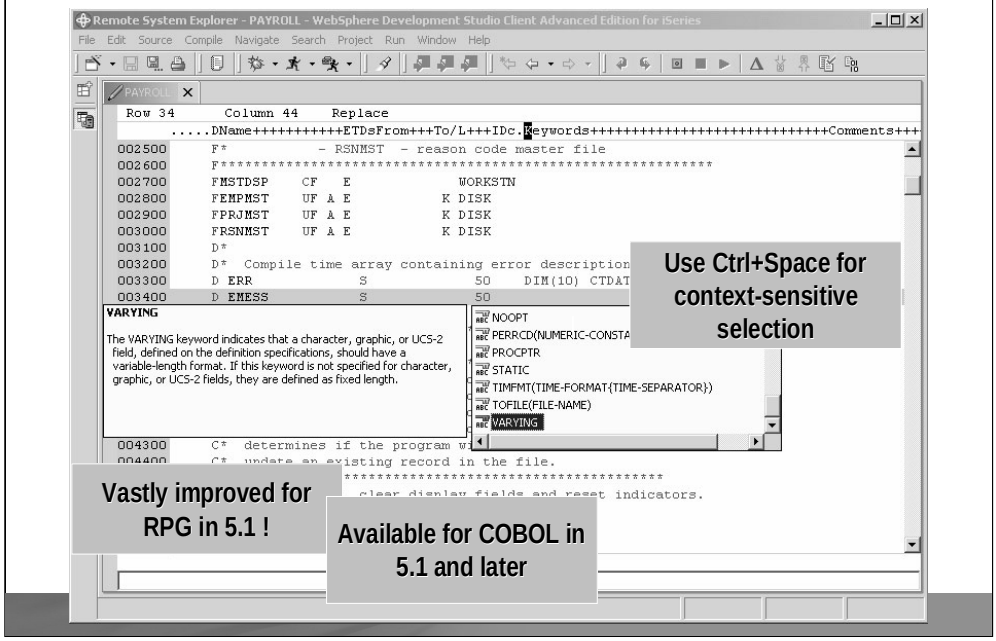
By default, the SEU profile is the active profile.

LPEX Editor Views – Outline view



The Outline view helps you visualize the member you are editing by displaying all the program structures and functions in a clear-cut view.

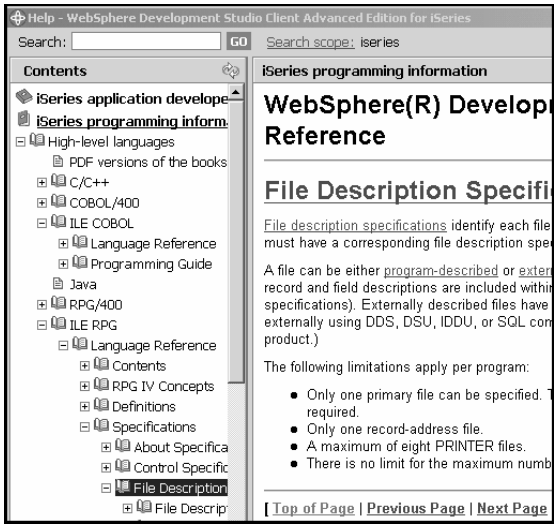
Content Assist



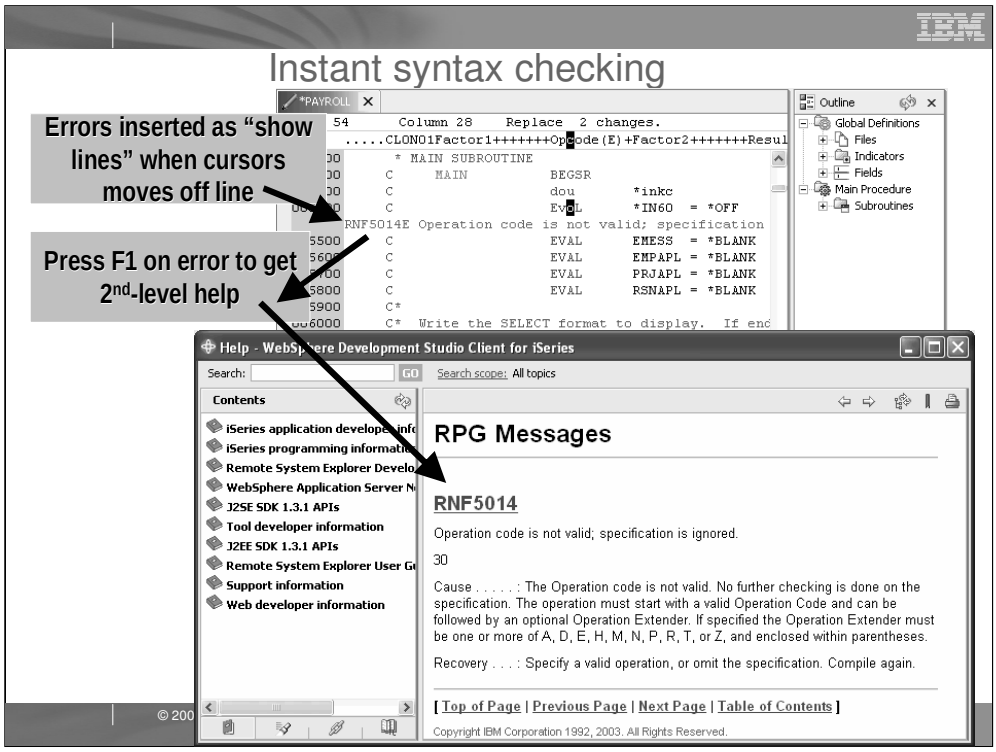
The Content Assist tool offers not only auto-complete functionality, by giving you a list of possible functions, objects or keywords to use, but also offers documentation on each of these to help you decide.

Help – at your Fingertips

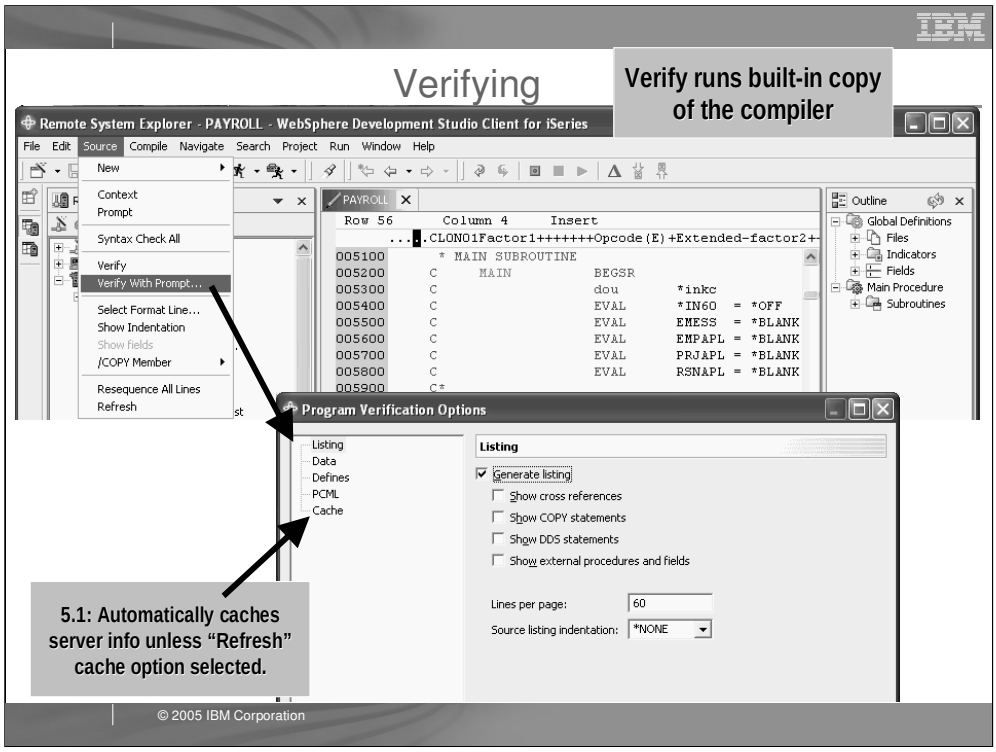
- **RPG, Cobol, DDS online references**
- **Programming Guides including CL**
- **F1 Help**
  - –Sensitive to where cursor is in the editor
- **Message Help**
- **and much more**



Now you can have a ILE RPG reference at your finger tips. Simply press F1

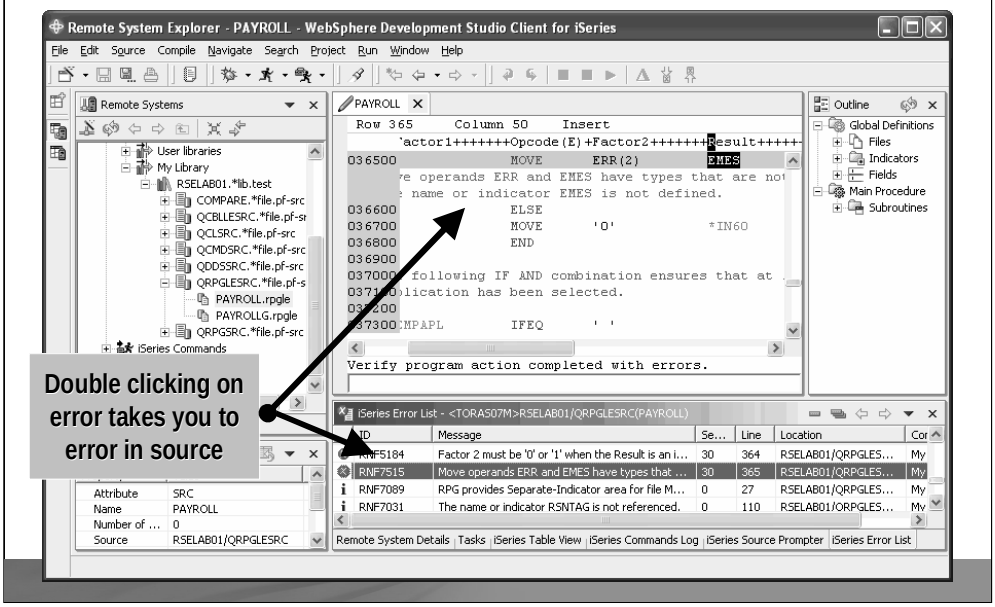


The Remote Systems LPEX Editor gives you instant feedback for your code, at the location where a syntax error occurs. This happens as you cursor off a changed line.



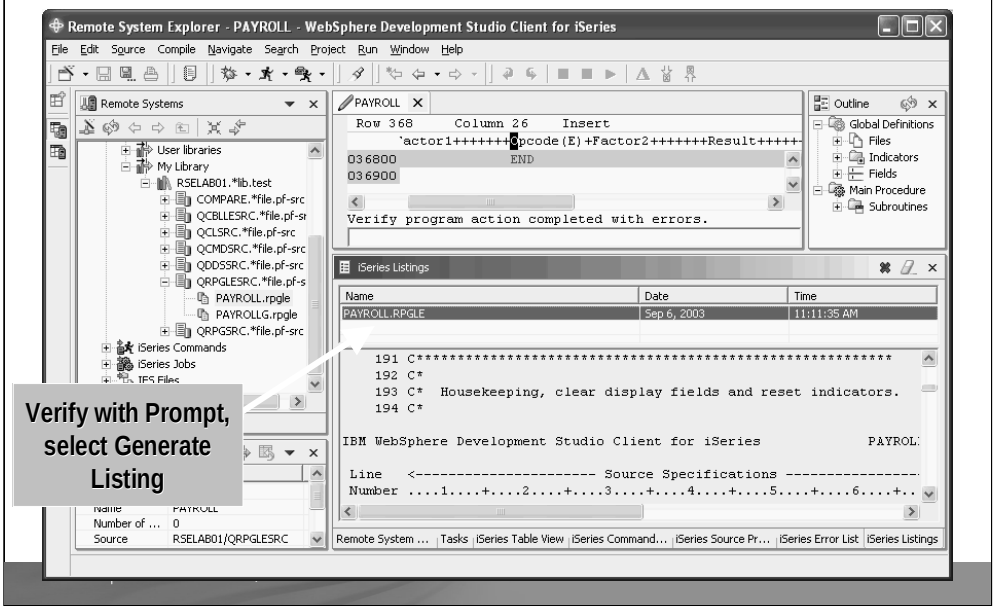
Use the Source pulldown to verify your work. The verifier is a built-in copy of the compiler, for the purpose of catching all errors, not just syntax. The verify has options, which effectively are the subset of the compiler options that affect error checking.

Verify – Error List

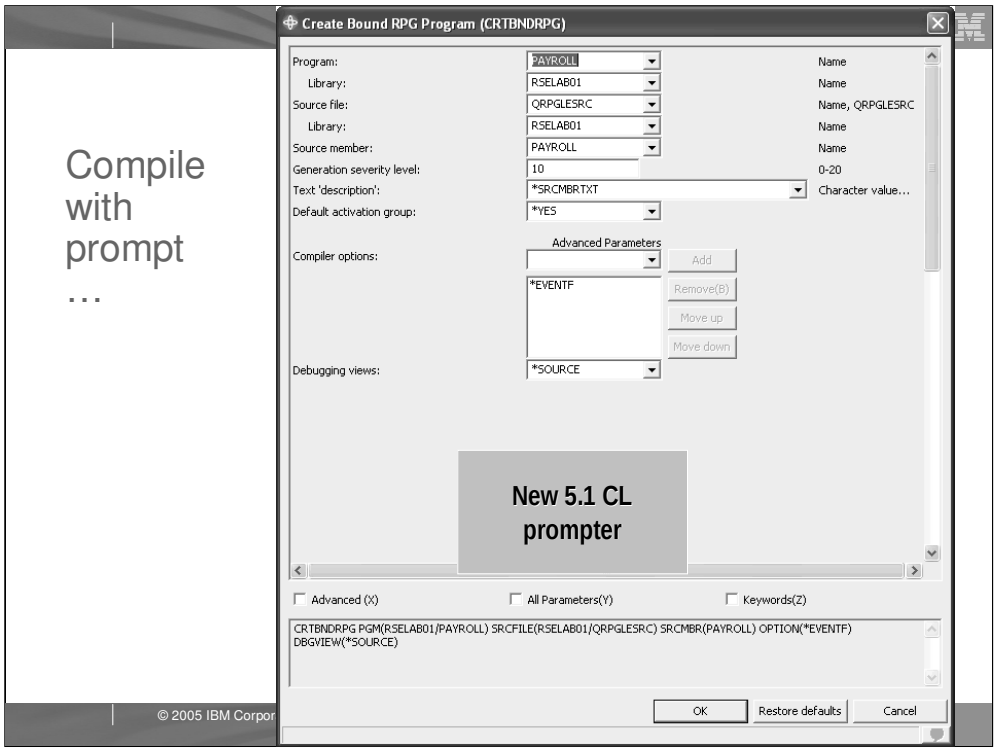


After performing a verify, the Error List window lists the errors that are found and their severity, inserts the error messages directly into the source and helps you to navigate between the errors.

Listing View



Here is the Listing view from the Program Verifier, if that option was chosen.



When you select to prompt the compile command, the command prompt is converted to a GUI and displayed. For the 5.1 release, this GUI was re-written from Java's Swing to Eclipse's SWT so it more consistent with the rest of the dialogs in the IDE.

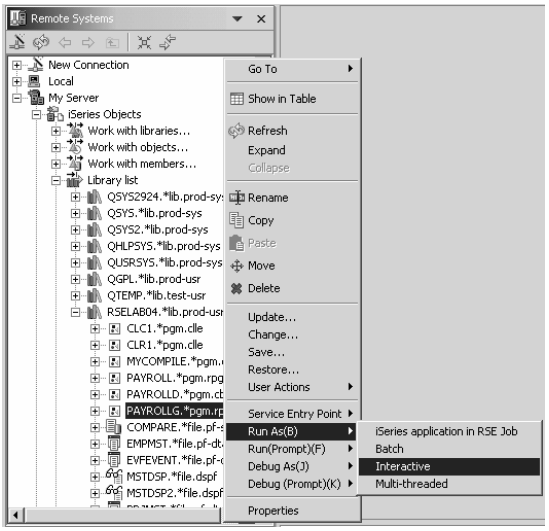
Run: from popup menus

Run from Popup menu of *PGM

- ✓ **iSeries appl. in RSE job**
- ✓ **Batch**
- ✓ **Interactive**
- ✓ **Multi-threaded**

Run As

Run(Prompt)



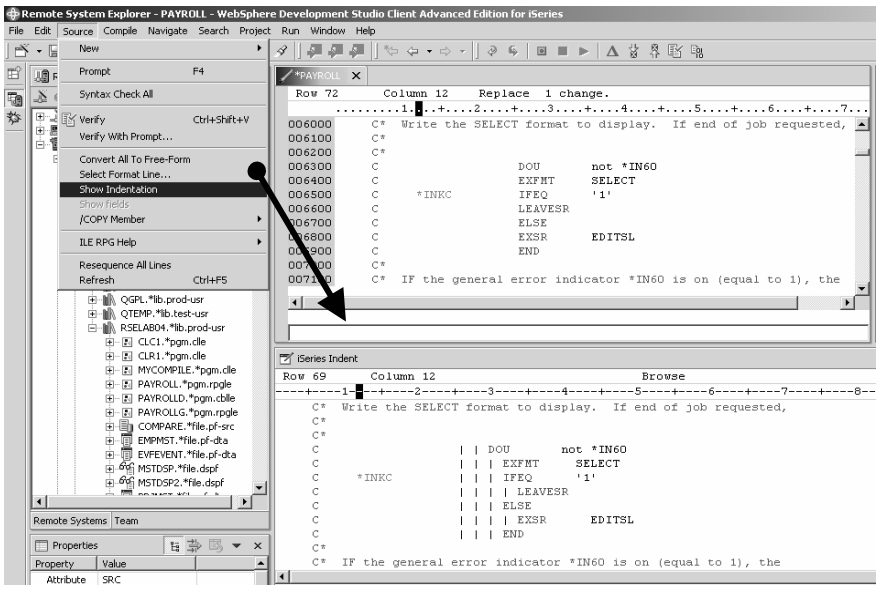
You can run programs from the Remote Systems view or the iSeries Table view in four ways:

- In the RSE communications server job
- In a batch job
- In an interactive job
- In a server job

In the first case, running the program in the same job as the Remote System Explorer communications server job,will tie up that job until your program ends. With batch and interactive jobs, you cannot monitor the status as easily, however, you do not tie up your communications server and you are notified when the program ends. Batch jobs work as you would expect, your application will be submitted to the default batch subsystem.

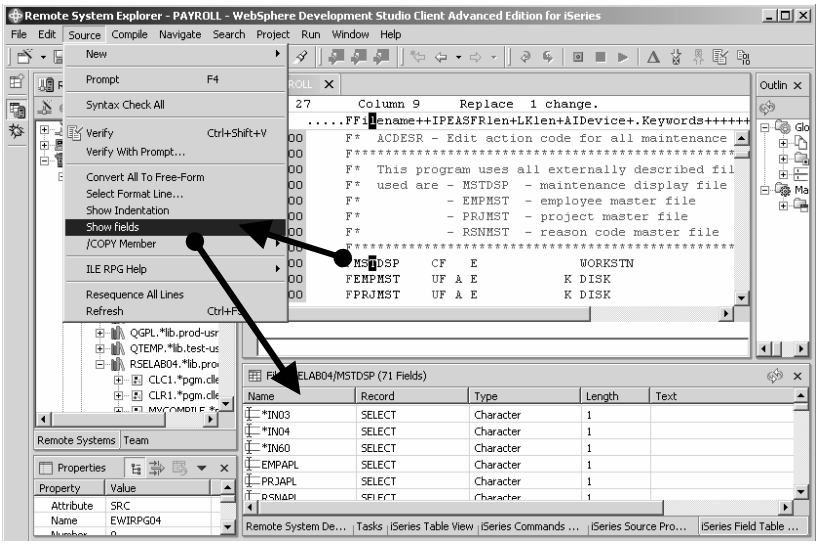
Note: A multi-threaded debug session creates a new server job and this way keeps the RSE communications server job free for other tasks.

Indent View



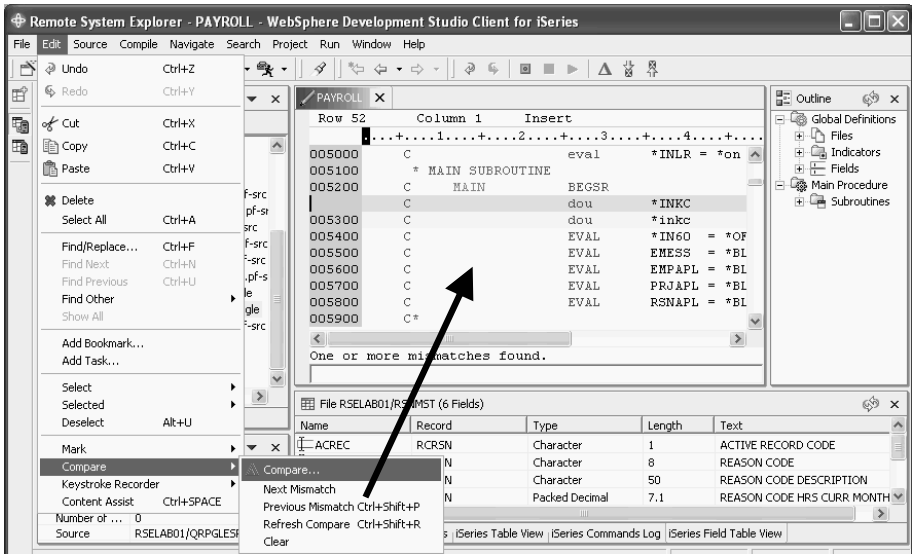
Here you see the results of the indent option: a readonly view showing the code indented, much as it is in a compiler listing, but with color.

Show Fields



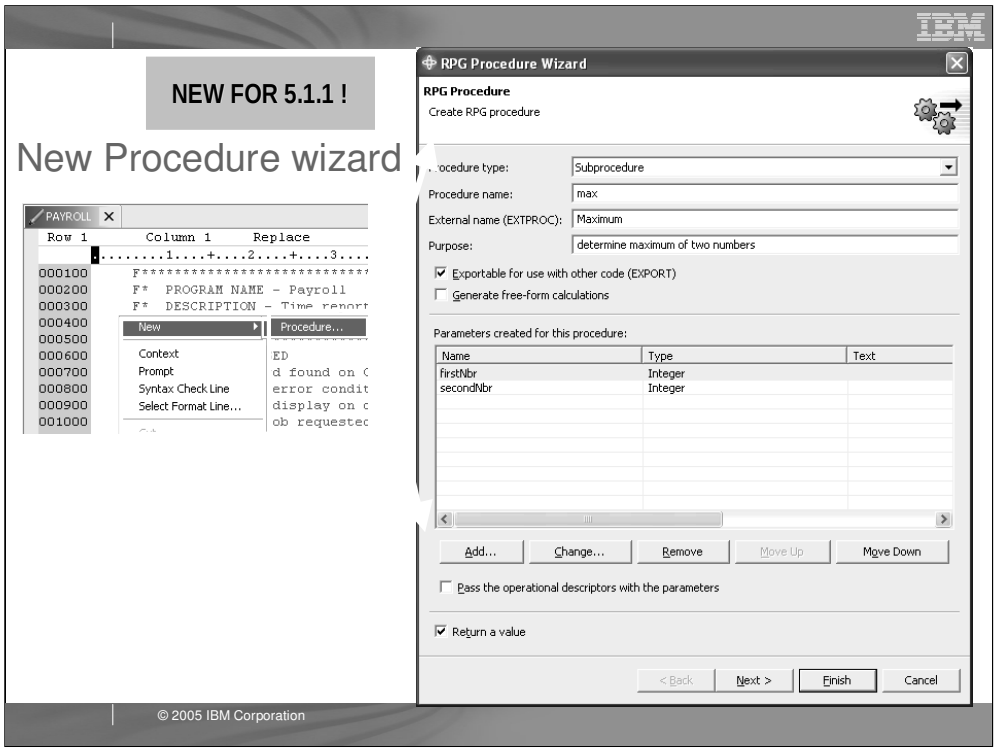
Here you see the results of the Show Fields tool. It retrieves the fields for the file under the cursor, and shows the fields in a table view.

Integrated Compare



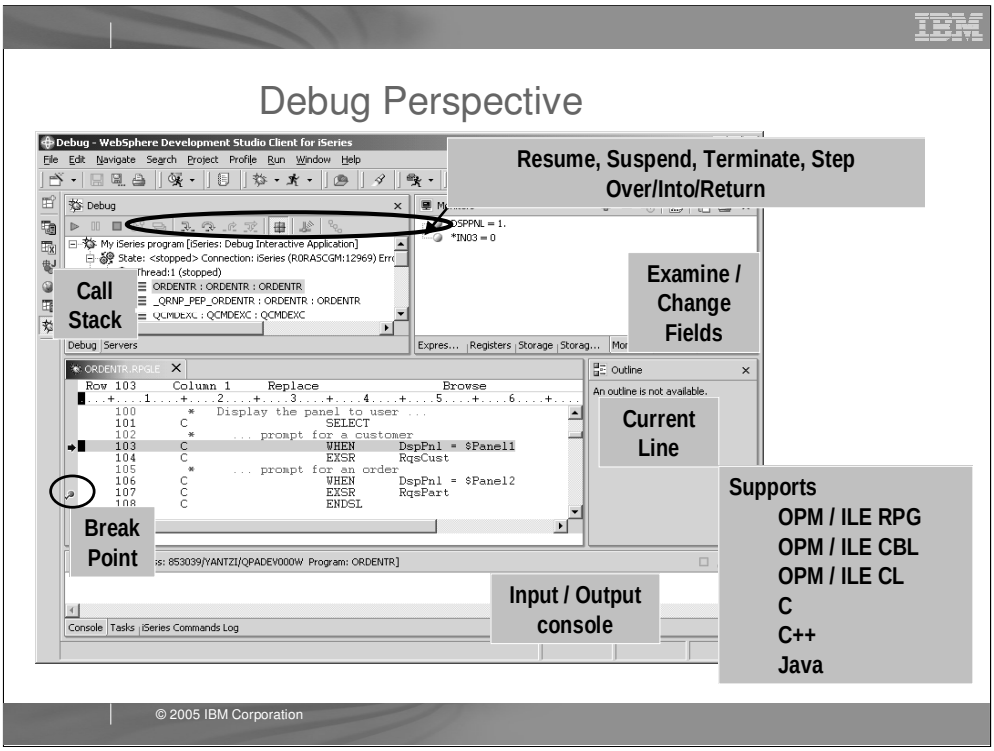
The compare tool under the Edit menu is used to compare the currently active member in the editor with another member you select, and the differences are shown within the edit window, as we see here.

Alternatively, there is another compare utility not editor based, in the popup menu for members.



In 5.1, the Procedure SmartGuide from CODE/400 was ported to the new editor. This will prompt for a procedures signature information (name, parameters and return type) and generate the procedure prototype and the body. It is up to you to subsequently add logic to that body.

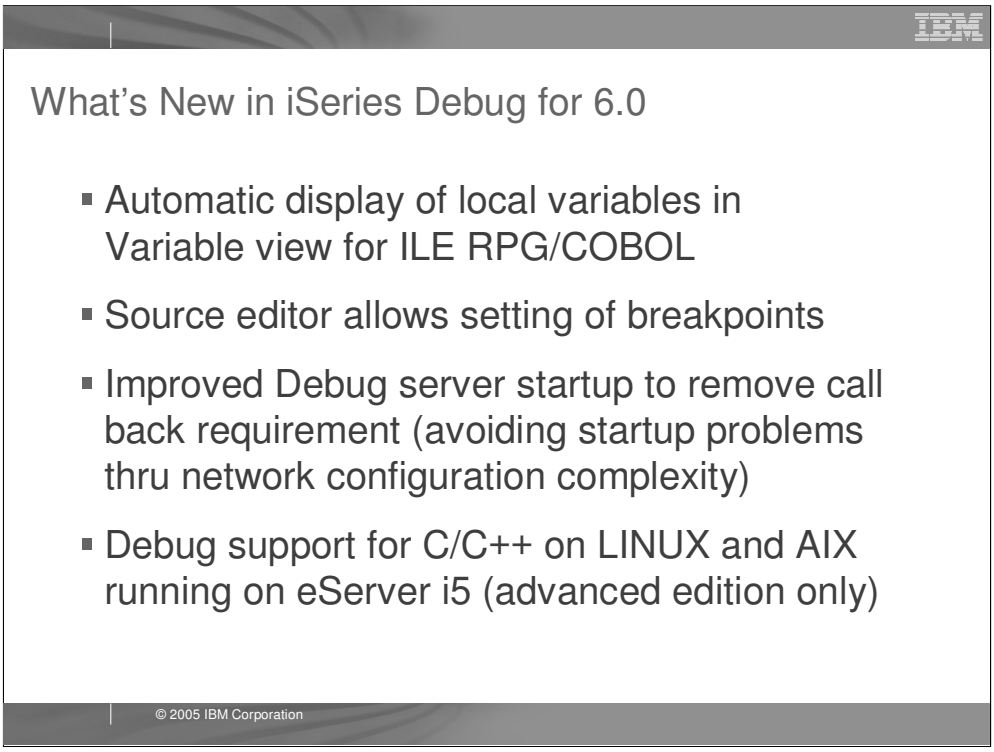
The prototype is generated at the top, after the last D-spec, while the body is placed at the bottom of the current member.



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**, 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.



Enhancements to the IBM WebFacing Tool in Version 5.0 include:

- User edit code support
- Programmatic invocation
- Web session management for coexistence with other Web applications
- Web browser management for back button and closing
- iSeries load balancing and WebSphere Application Server cluster support
- Reply to DSPPGMMSG, application error
- Many performance, scalability, usability and function enhancements

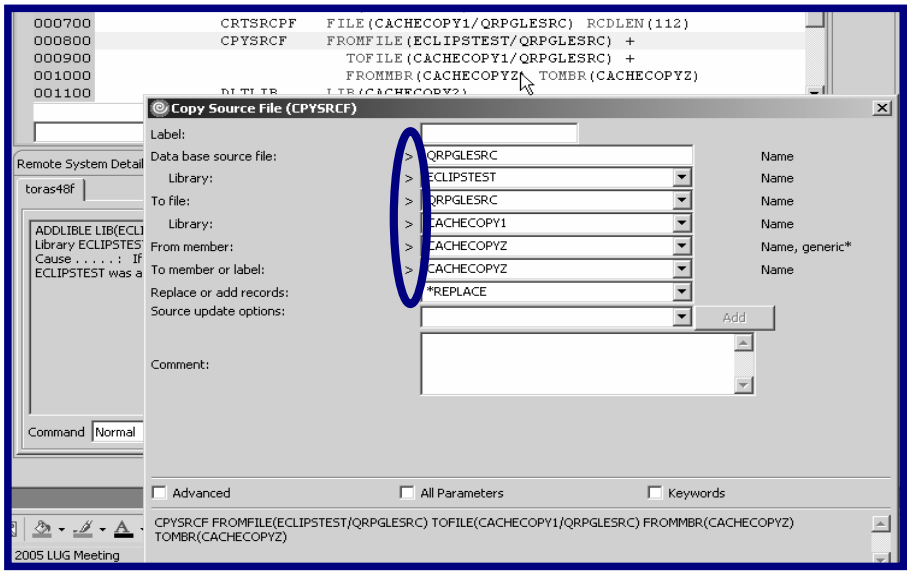
Better Tools

Remote System Explorer V6.0

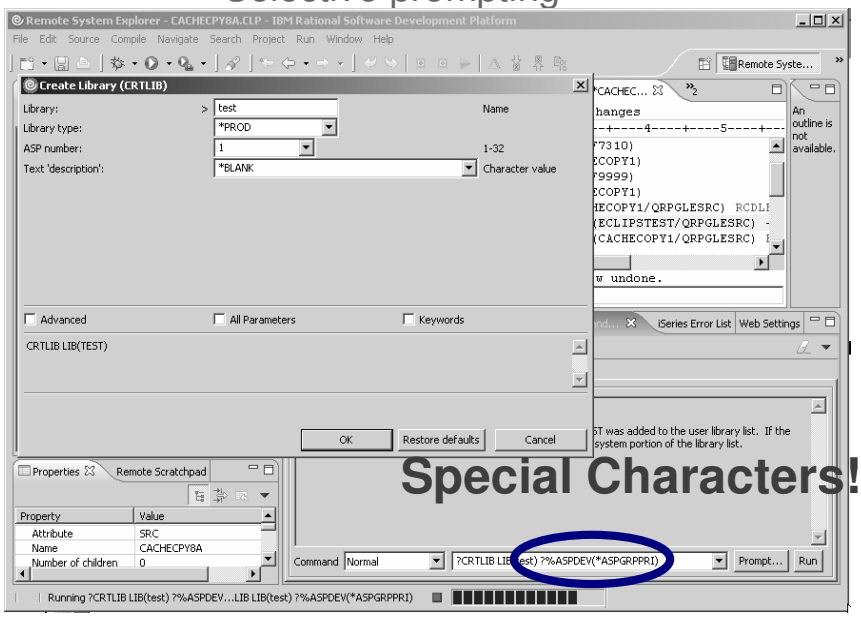
RSE Enhancements in version 6.0

- **CL Changes**
  - **Indicator for used parameters**
  - **Selective prompting:**
    - **Support of all prompt modifiers (ie ?*)**
  - **Support labels and comments (when editing CL source).**
  - **Additional error checking for built-in commands (ie %SST)**
  - **A refresh cache button**
  - **Context of non-edit and edit mode will be permitted**
- Error List enhancements
- No power user/admin authority on windows to access PV and SC

Indicator for specified parameters



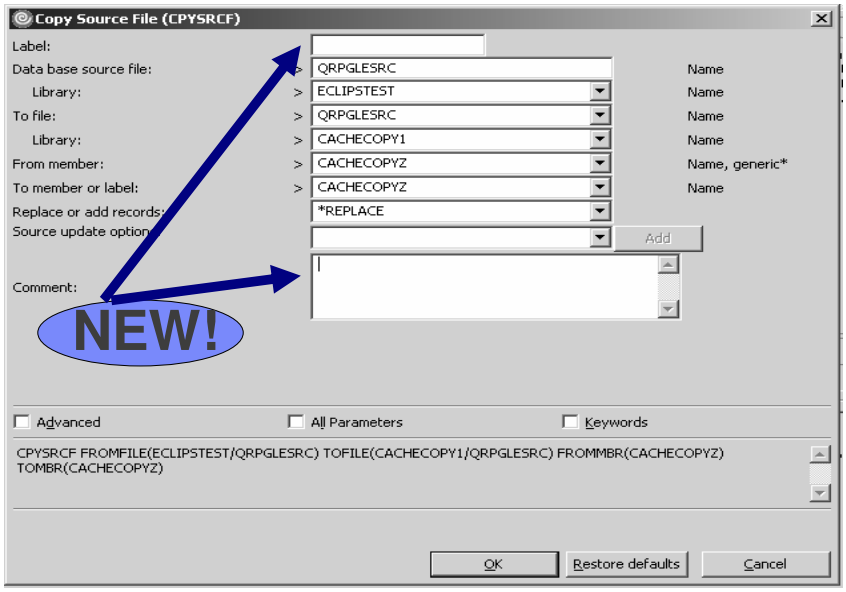
Selective prompting



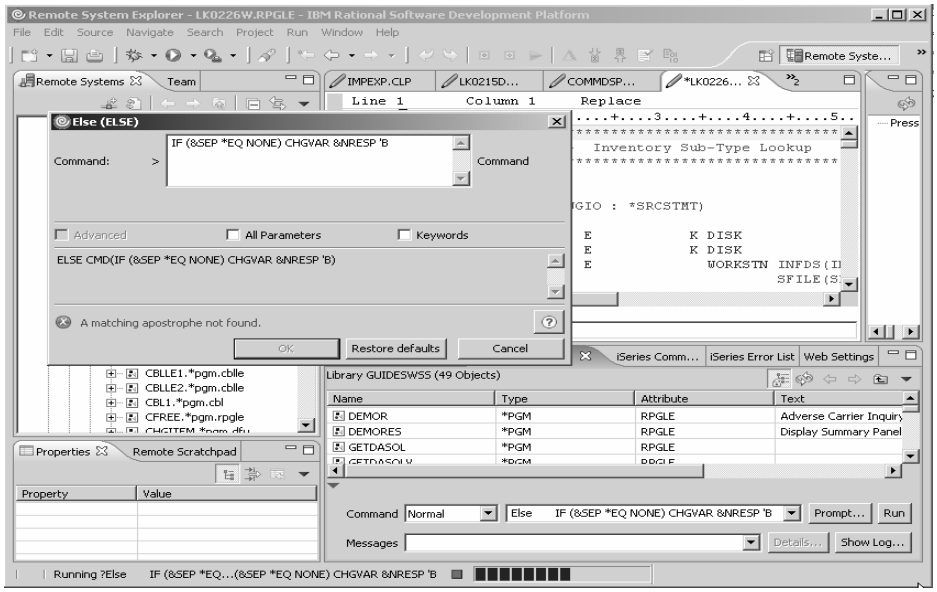
Selective prompting

Selective Prompting Character	Description
??	The parameter is displayed and input-capable.
?*	The parameter is displayed but is not input-capable. Any user-specified value is passed to the command processing program.
?<	The parameter is displayed and is input-capable, but the command default is sent to the CPP unless the value displayed on the parameter is changed.
?/	Reserved for IBM use.
?–	The parameter is not displayed. The specified value (or default) is passed to the CPP. Not allowed in prompt override programs.
?&	The parameter is not displayed until F9=All parameters is pressed. Once displayed, it is input-capable. The command default is sent to the CPP unless the value displayed on the parameter is changed.
?%	The parameter is not displayed until F9=All parameters is pressed. Once displayed, it is not input-capable. The command default is sent to the CPP.

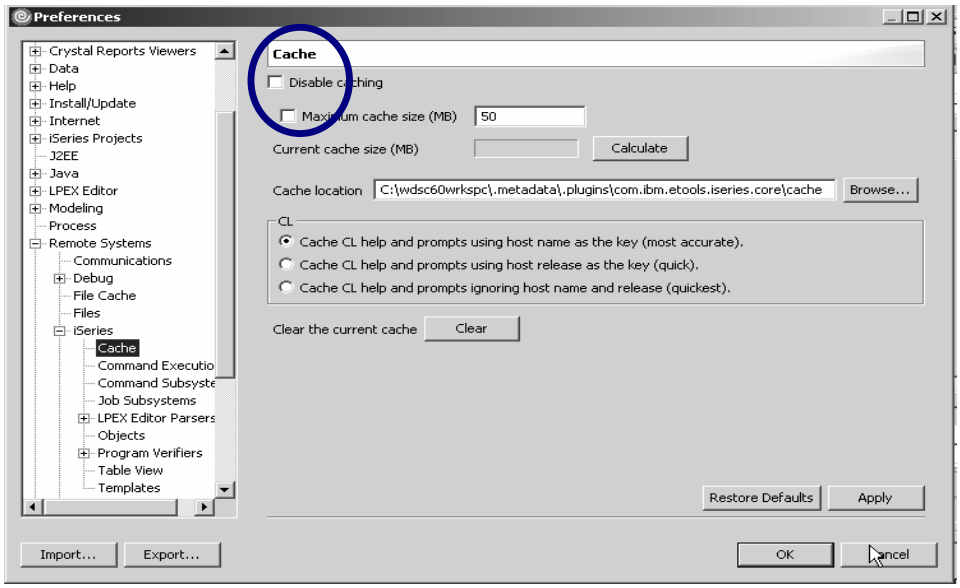
Support Labels and Comments



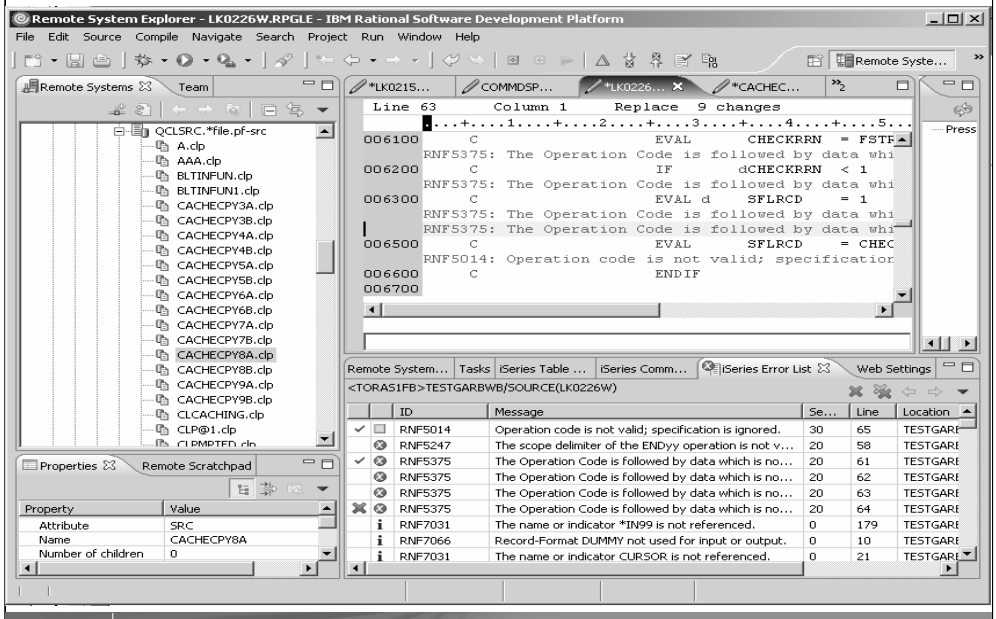
Error Checking for CL built-ins functions



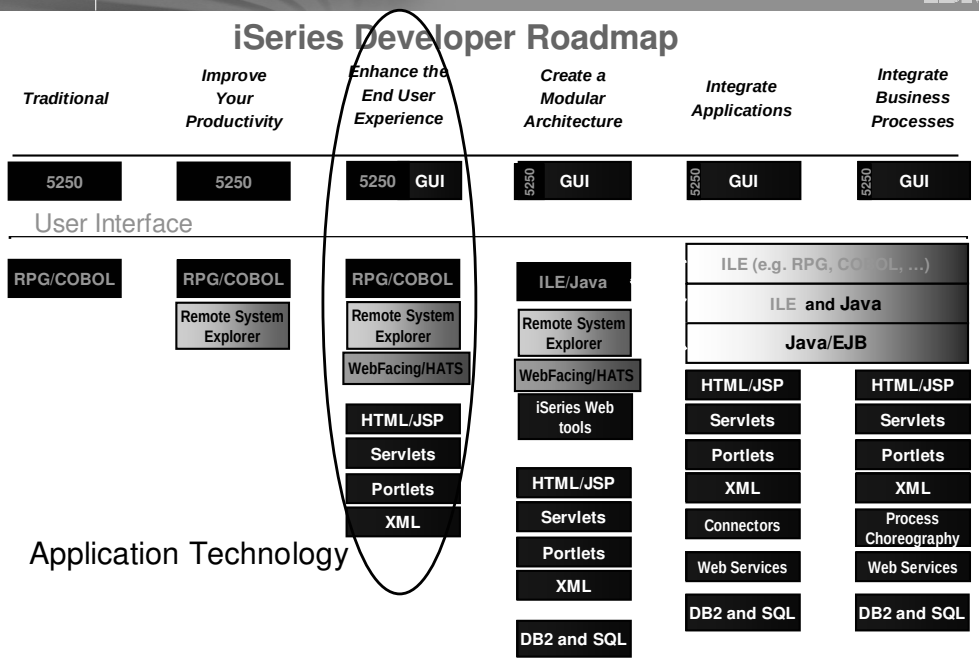
Refresh Cache

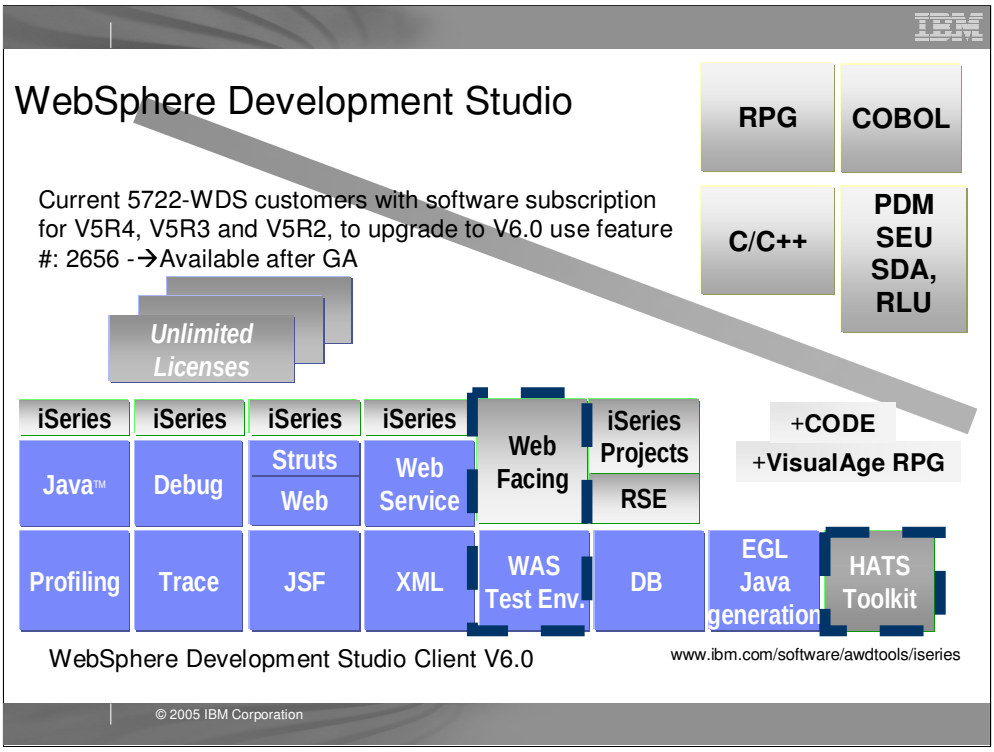


Error List improvements



iSeries Developer Roadmap



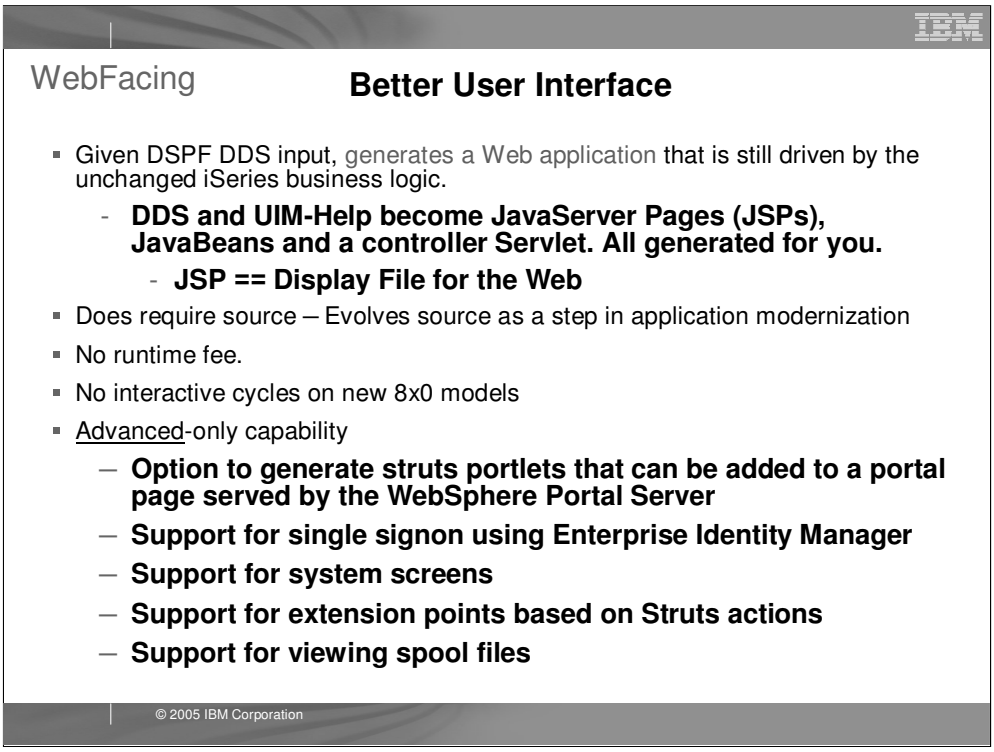


There is one family of application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and **unlimited licenses** of the workstation-based 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.



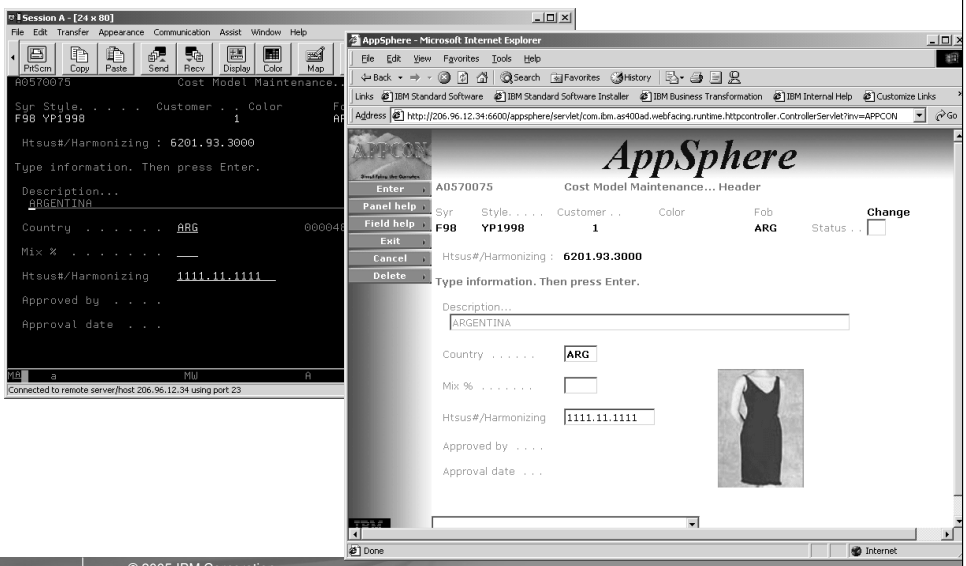
There are many options available now for transforming green-screens to Web pages, but WebFacing is unique among them. WebFacing does this **conversion at development,** taking as input the display file DDS source, and generating JavaServer Pages for the output (for example, Web pages).

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.

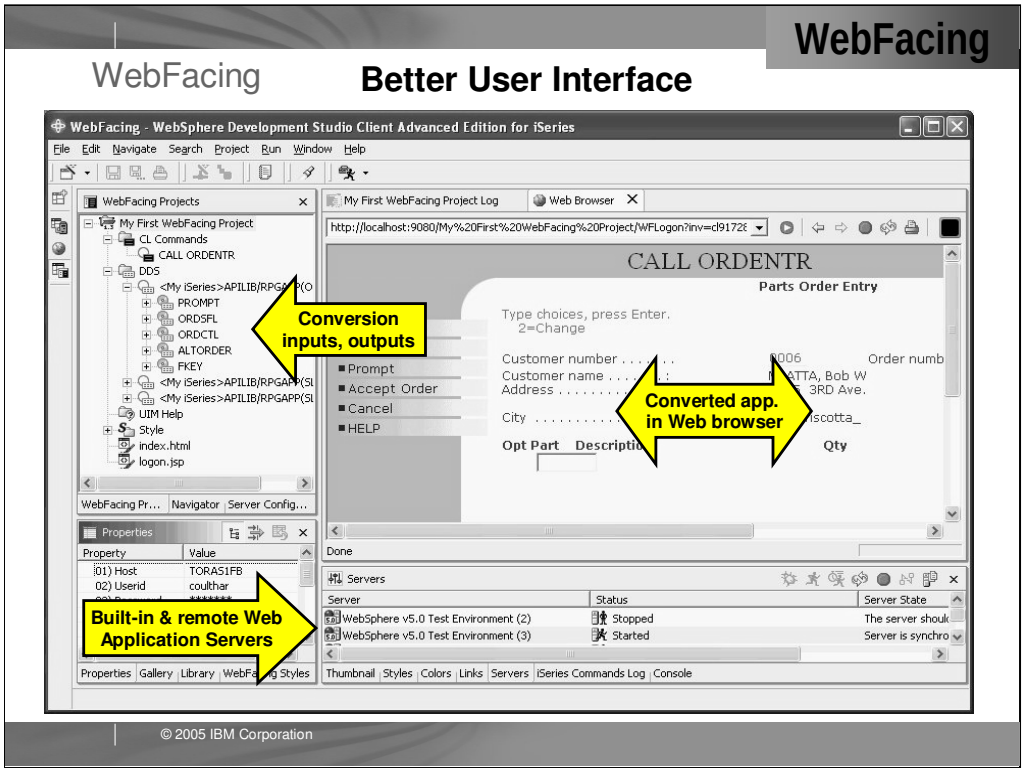
WebFacing

Better User Interface



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.



Better User Interface – IBM WebFacing Tool for iSeries

This is a screen capture of the IBM WebFacing Tool development environment provided by WebSphere Development Studio Client. The entire collection of views and editors is called the WebFacing "perspective." Developers can open multiple perspectives and flip between them using the icons in the bar on the left. Is **this beginning to look familiar? It should because all the development tools are built to have the same look and feel.**

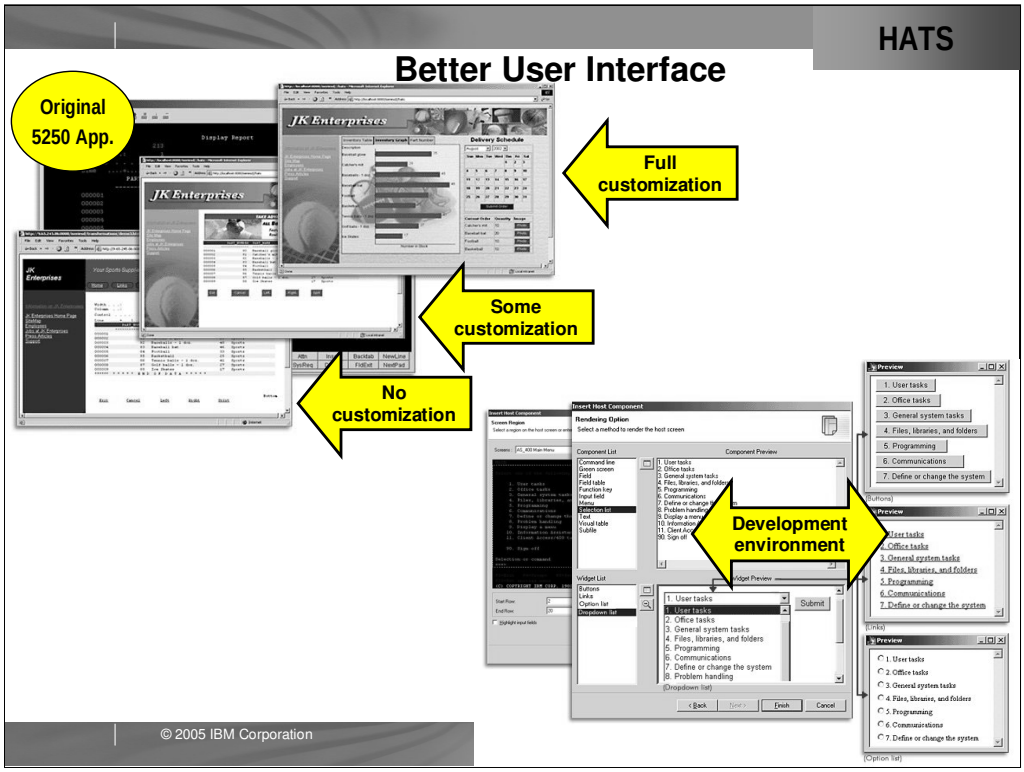
Developers use a wizard to create a project. They specify the DDS and the User Interface Member (UIM) Help Panel members to be converted. The resulting project allows editing access to the original DDS and to the generated files. It is easy to test the converted output in the IBM WebFacing Tool for iSeries by **right-clicking the project and selecting "Run on Server."** This executes the built-in, preconfigured copy of WebSphere Application Server (V4.0, V5.0, or Express V5.0).

The view at the bottom right shows the built-in copies of WebSphere and allows developers to easily **start, stop, and publish to them**. The same view can be used to **manage remote instances of WebSphere Application Server**. When running an application within the tool, the application's UI is shown in the built-in browser, as is seen in the large, right window shown on this screen capture.

HATS Toolkit

Better User Interface

- Generates a Web Application that converts 5250 datastream into a Web user interface with JavaServer Pages
- While default transformation is quite good, there is also extensive customization-capability
  - **Involves tools to identify the screen to be customized, and the type of transformation to apply to selected screen area**
- Does not require the source. Works with 5250 datastream
- Does require interactive cycle to deploy
- Does require runtime license to deploy
  - **Unless you use default transformation that comes with Client Access for the Web**



Better User Interface – HATS

The set of four progressive screen captures (on left) illustrate the result of using HATS:

- The original green screen
- The same 5250 screen as converted to a Web page by HATS—no customization
- The same UI with some customization
- The same user interface with significant customization

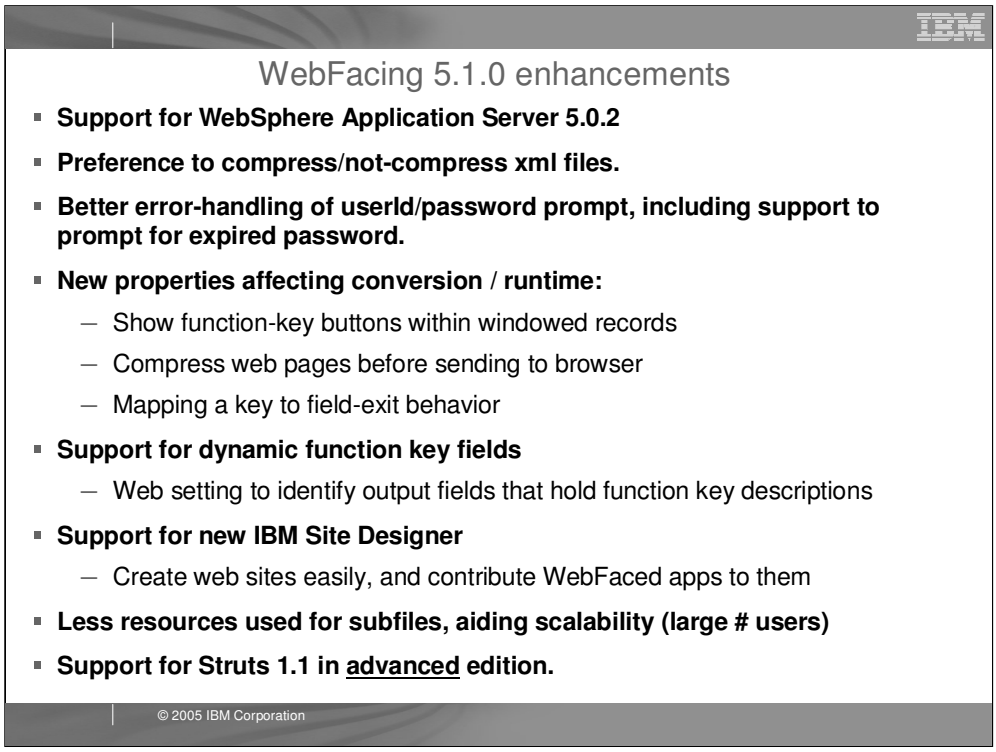
You can see by looking at the diversity among these screens, a great deal of built-in customization capability is provided in HATS, including the ability to turn subfiles into graphs and to insert calendar date-pickers.

The three screen captures on the right show the Eclipse-based development environment for HATS—used for the customization of the generated UIs. Alterations require no HTML skills and are accomplished via easy-to-use wizards and dialogs. As with the IBM WebFacing Tool for iSeries, HATS leverages the built-in test environment for WebSphere to simplify the process of seeing the results.

Better Tools

WebFacing

V5.1.2



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

What's new in WebFacing for Version 5.1.2 Standard Edition

DDS INVITE

- **Support for applications that use the DDS keyword INVITE and CL commands, such as SNDRCVF or RCVF WAIT(*NO), to perform asynchronous input/output of data to the screen**

DDS DSPATR(pfield)

- **Support for applications that use program-to-system fields for setting the display attributes of protected fields**

Job / System attributes

- **Performance improvement for applications that require job attributes information for resolving certain DDS keywords**

Project Migration

- **Project migration facility for migrating projects from the previous release to the new release**

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

What's new in WebFacing for Version 5.1.2 Advanced Edition

Portal

- **Access DDS based RPG, COBOL, or CL applications from portal pages by generating portlets.**
  - The generated portlets can be run locally from the IDE provided by WDSC for testing and can be deployed to IBM's WebSphere Portal Server for running in production mode.

Authentication

- **Use single signon to allow users who have been authenticated by the WebSphere server to traverse WebFacing web or portlet applications without having to re-login**
  - Uses Enterprise Identity Manager (EIM) (V5R2 and up)

System Screens

- **Support applications that display system screens during the application flow**

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

Better Tools

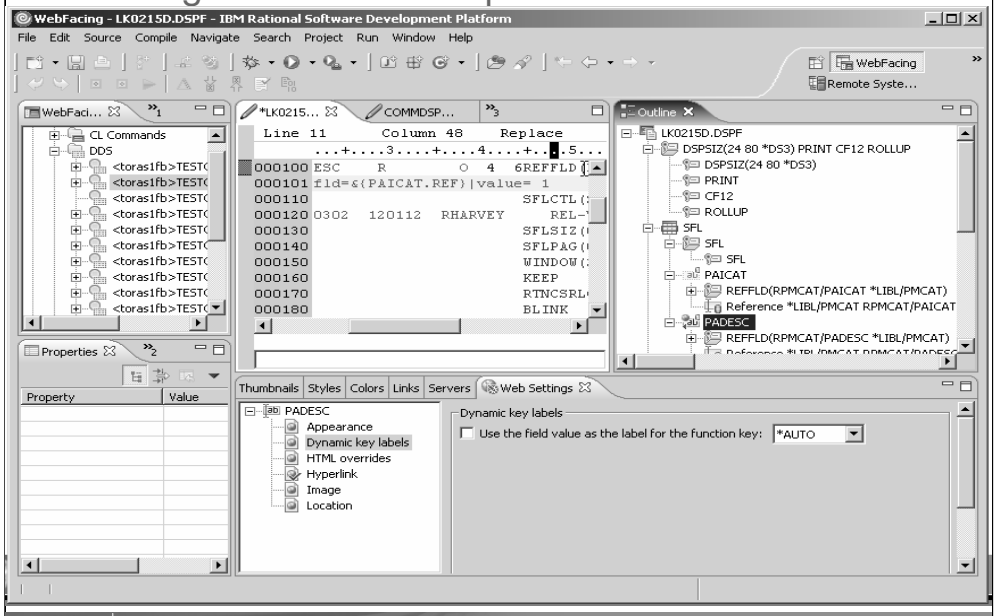
WebFacing

V6.0

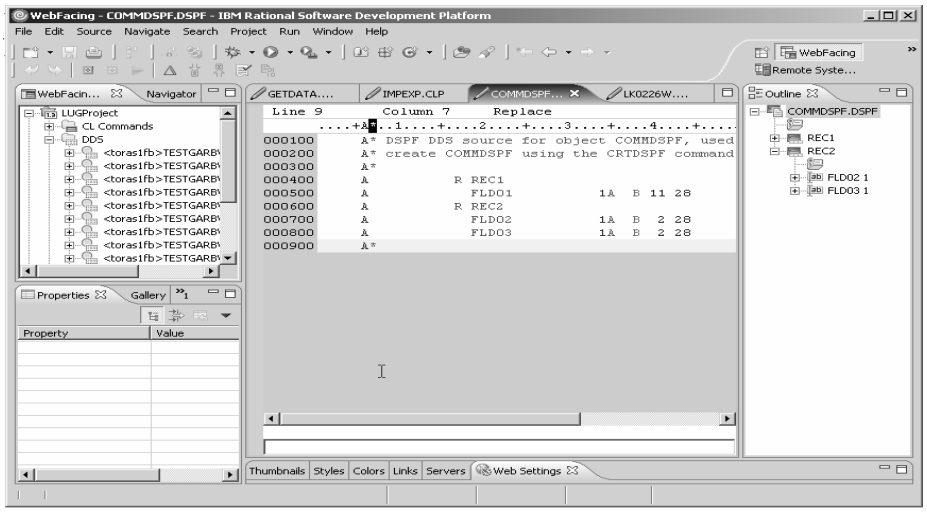
Version 6.0 WebFacing

- **CODE/400 components integrated into eclipse**
  - **CODE components are not required to run WF V6**
  - **CODE communications used by WF replaced *entirely* by RSE communications**
  - **WF conversion is now entirely based on a new DDS engine running in Eclipse**
    - **CODE Designer install is not required to use WebFacing**
    - **Positions WF for future DDS customization tooling for web or other client**
  - **V6 adds New WF WebSettings view in Eclipse.**
    - **On any perspective you can add the websettings view by the following action:** Open View > Other > WebFacing > Web Settings
- CODE Designer install is not required to create WebSettings

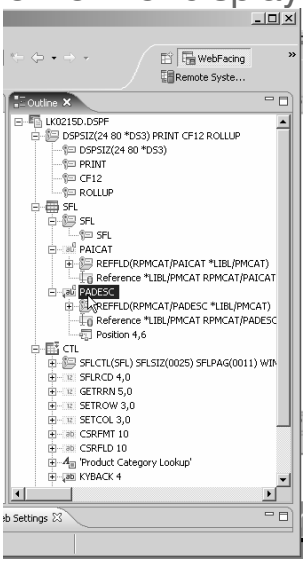
Integrated WF development environment



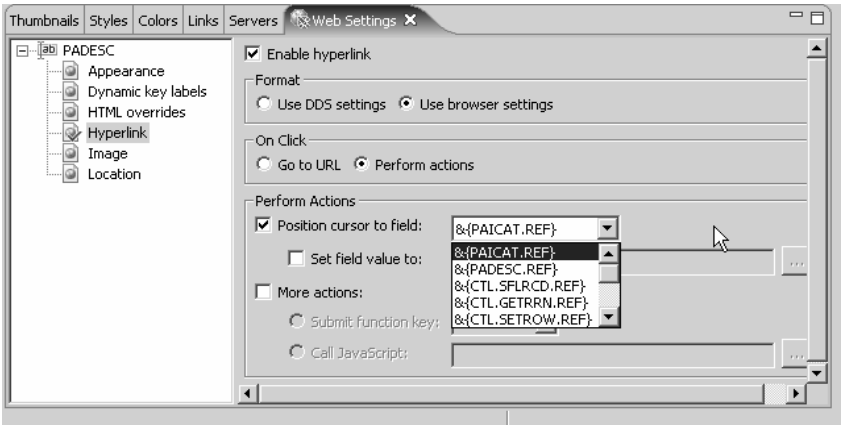
Better DDS ***edit*** integration



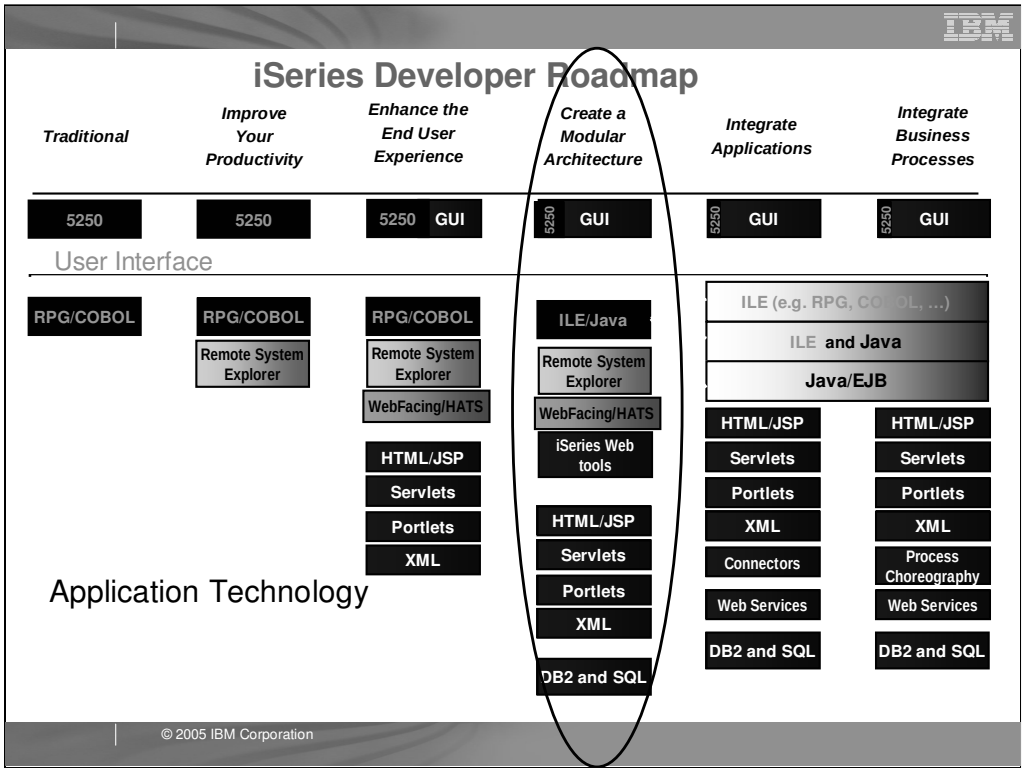
Outline view for display files

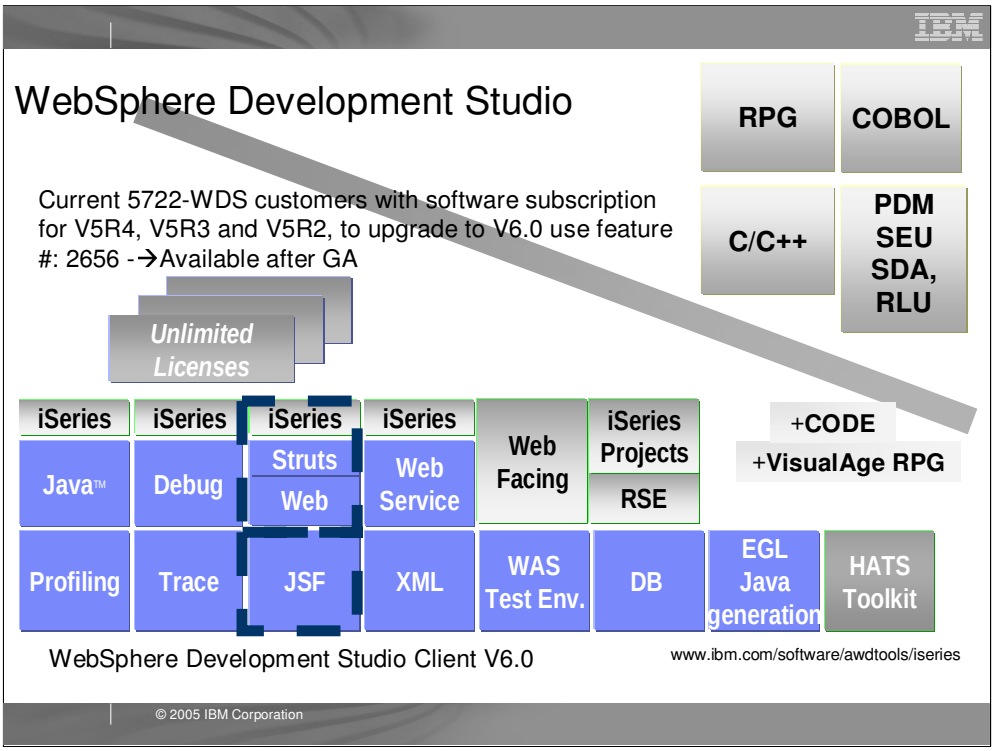


Web setting view in eclipse



Real-time source update!



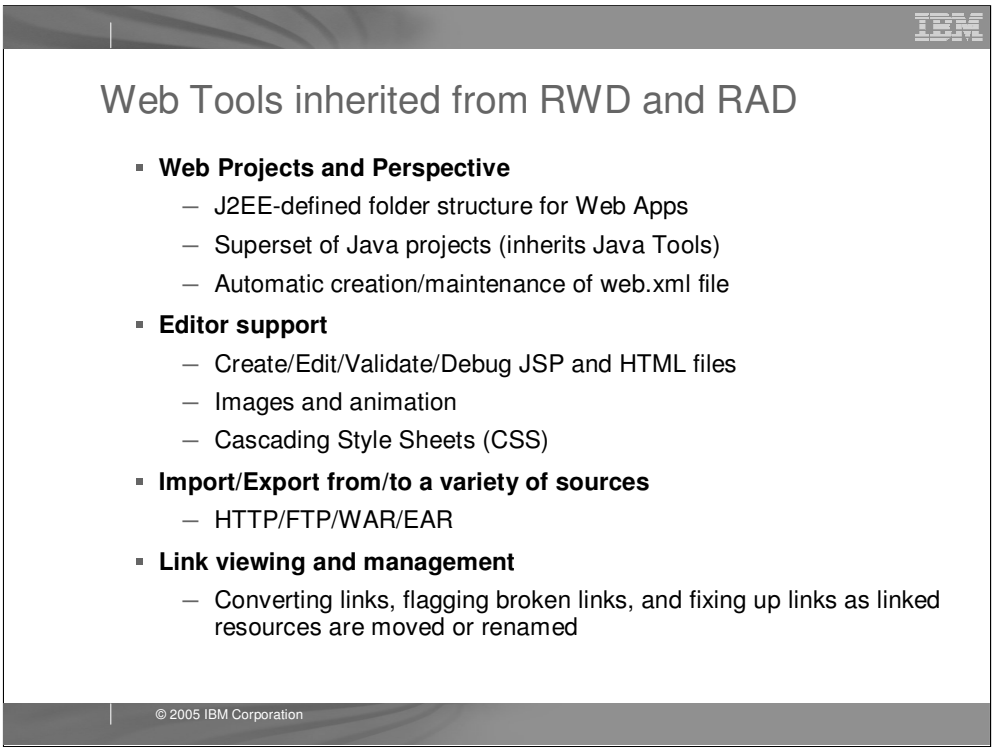


There is one family of application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and **unlimited licenses** of the workstation-based 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.



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.

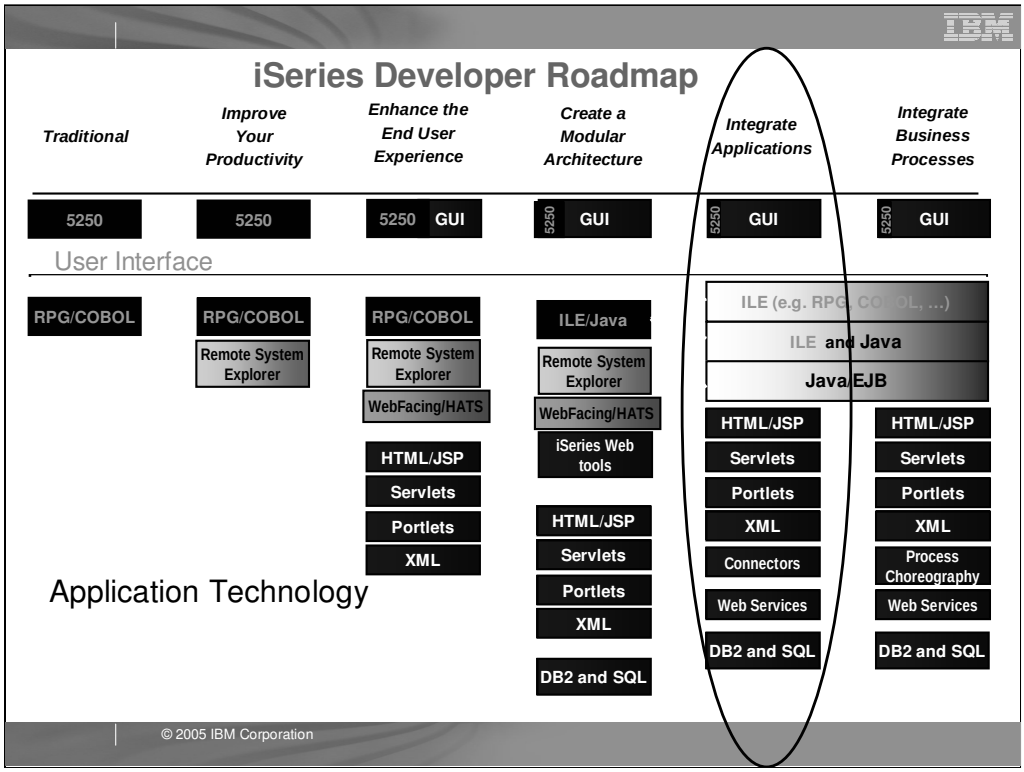
An important file for a Web application is the **web.xml file that describes the servlets within the application**. Using Development Studio Client however, you need never worry about this file! **It is created and maintained for you!** However, should you wish to **edit it** there is a fantastic web.xml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

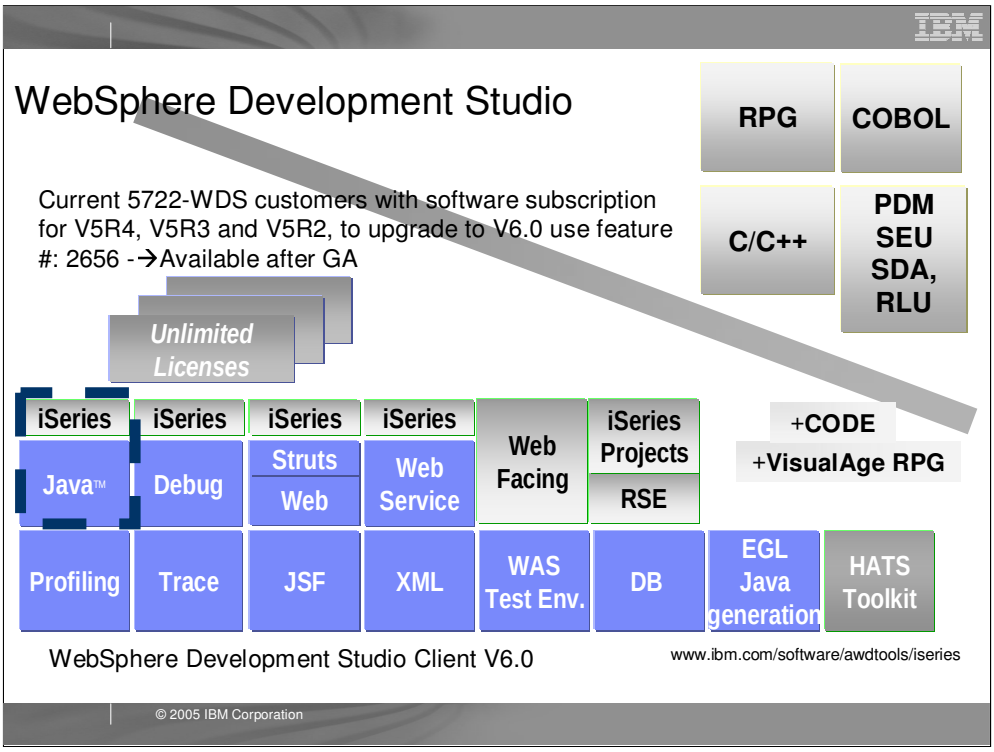
There are also fantastic editors for creating JSP and HTML files, as well as tools for creating images and even logos. All Web sites today use **cascading style sheets from the HTML 4.0 specification**, and using these is easy with the **editor support** in the Web Tools of Development Studio Client.

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.**

- **iSeries Web Tools, at a glance**
  - **Tools optimized for iSeries developers!**
    - **Host Information wizard**
      - ✓Set runtime information such as library list and sign-on information, to be used by glue generated by all Web Interaction wizards for this Web project
    - **Web Interaction wizard**
      - ✓You define the parameters to a *PGM/*SRVPGM, wizard generates input JSP prompting for input parm, output JSP showing output parms, and all the glue in-between
      - ✓Or you pre-create the input and/or output pages, and map the input/output fields on the pages to the input/output parameters in the *PGM/*SRVPGM, and it generates the glue to bind them
    - **Web Components palette**
      - ✓Web GUI Widgets that support DSPF-like attributes such as error checking by datatype, and edit-code and edit-word



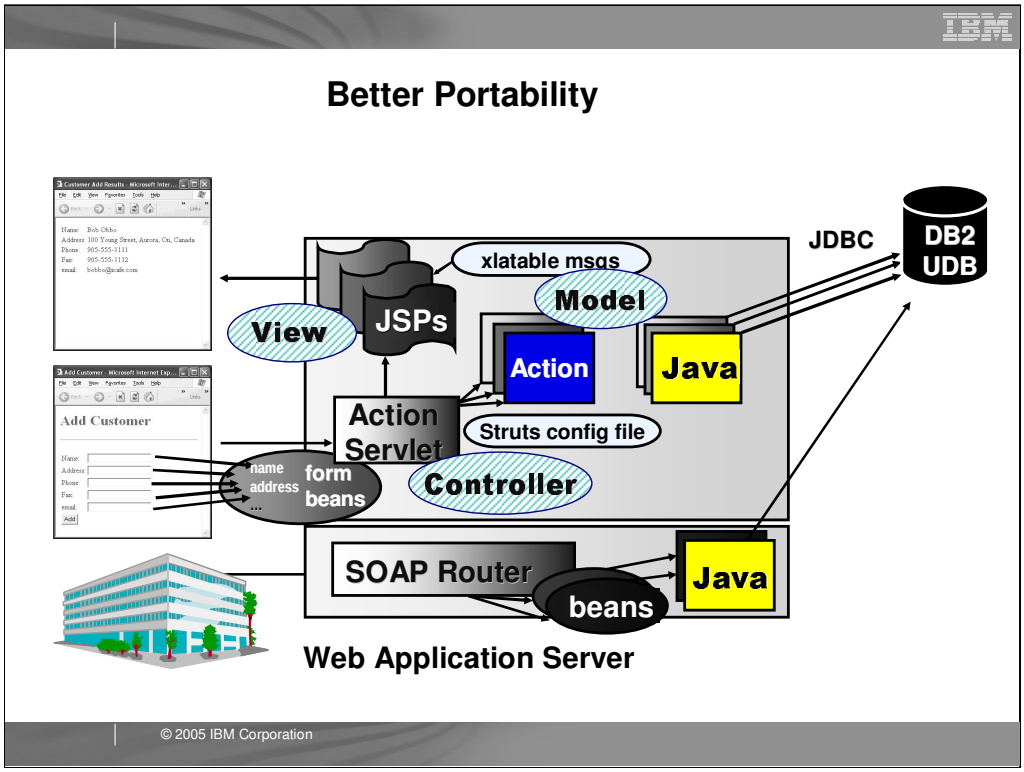


There is one family of application development product sold by IBM, for iSeries, as of V4R5. This is WebSphere Development Studio (Development Studio), which includes all four host compilers, all traditional tools (ADTS = PDM+SEU+SDA+RLU+DFU+AFP+CGU), and **unlimited licenses** of the workstation-based 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.



Better Portability

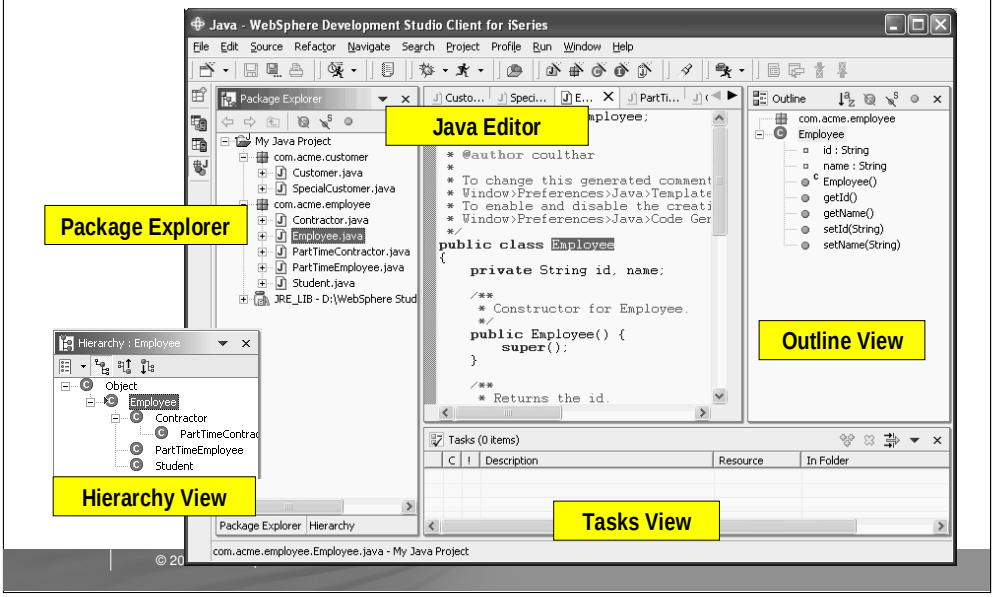
In this architectural diagram, you will notice that Java is used for the business logic. Just to reiterate, there is no RPG or COBOL code in this architecture… that would prevent the application from being portable… which now, is a primary goal.

The Java business logic runs in the second tier (shown in red text), but it is also possible to run it on the third tier (also shown in red text) by using technology such as Remote Method Invocation (RMI). In this case, each Action essentially calls one or more Java class to perform the needed business logic.

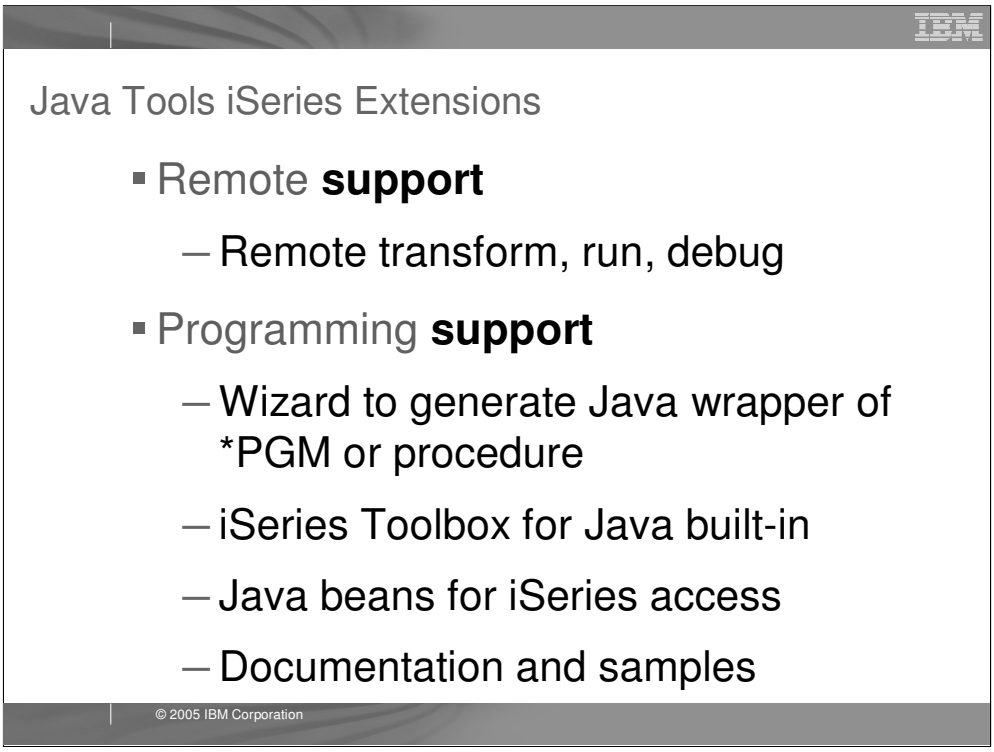
Please note that it is not recommended that the business logic be actually placed into the Action class. You can see in this chart that, in the second tier, the Java code stands alone from the action code.

Also, be aware that the Java code will probably use JDBC to access the data.

Java Perspective **Better Portability**



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.



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:

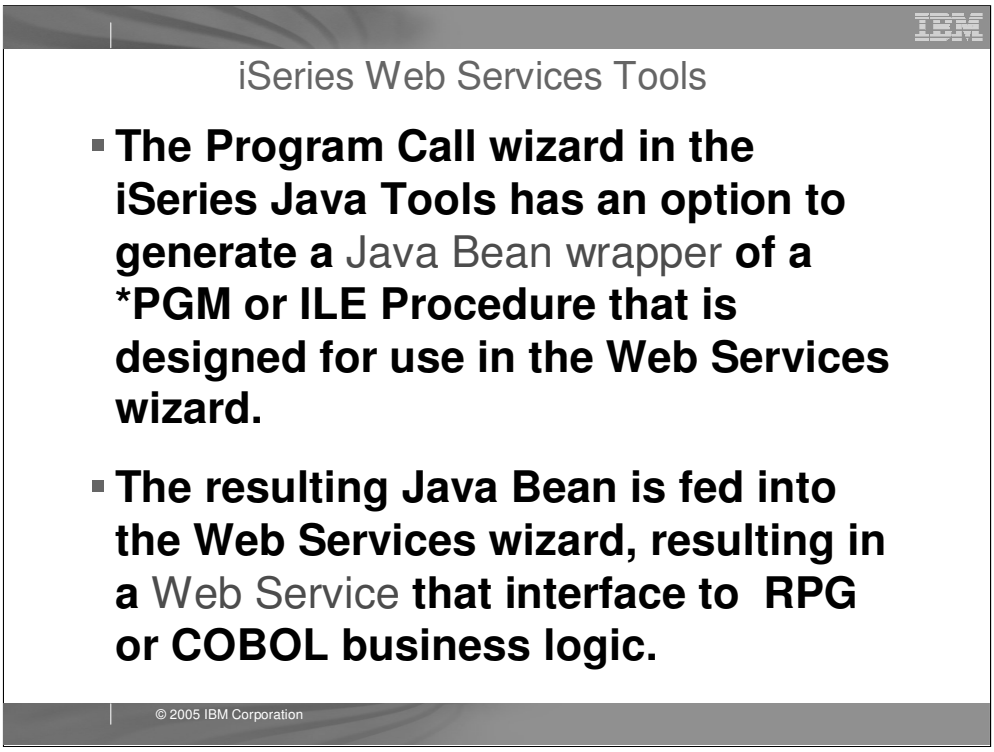
•A dedicated view for remotely transforming and running Java on iSeries, from within the Development Studio Client IDE

•A Launch Configuration wizard allows you to define run and debug configurations, such that you can run your Java applications remotely on iSeries as if running or debugging it locally in the IDE.

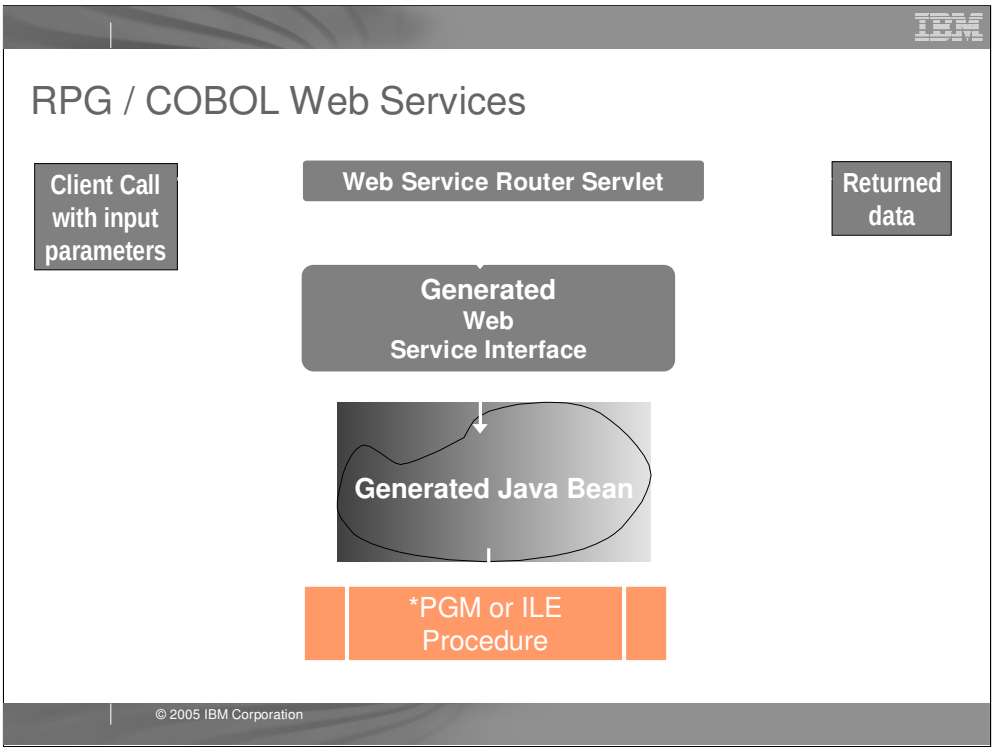
•A Program Call wizard to generate a Java class to call existing RPG or COBOL business logic

•A plug-in that contains the full IBM 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



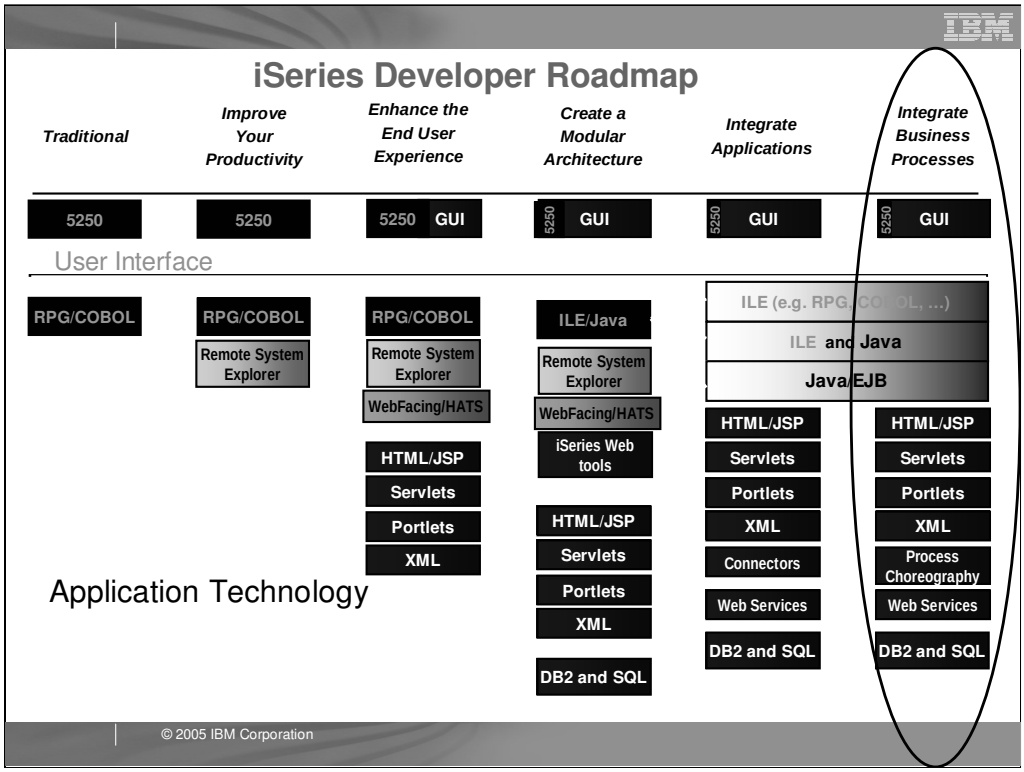
The Web Service wizard can create a Web Service, using a Java Bean generated by the iSeries Program Call wizard, to call one or more server programs or service program procedures on the iSeries.



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.

JavaTools Enhancements version 6.0

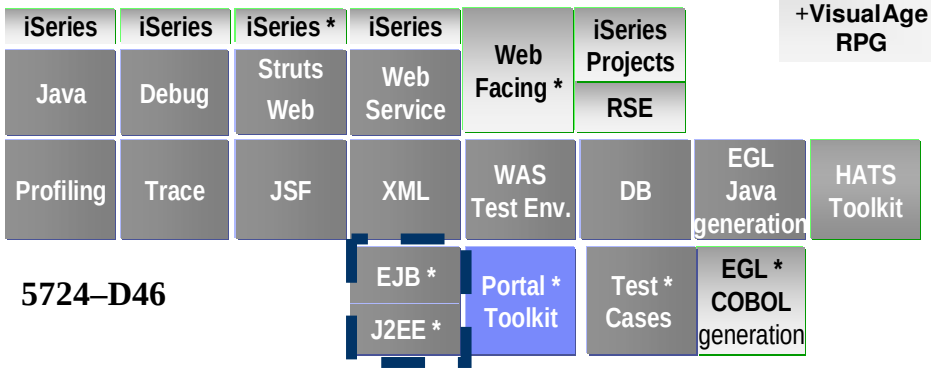
- Toolbox, JFormatted, DFU, and ObjectList beans are now available in the palette of the Visual Editor for Java. User can now create Java classes to make use of these visual components or Java beans inside the Visual Editor for Java.
- Enhancements to the Program Call wizard:
  - **Easier to launch the wizard**
  - **Ability to reuse configuration file for multiple program calls**
  - **Better entry field validations**
  - **Better user interface for importing PCML from source**
- Ability to browse for procedures of a service program



WebSphere Development Studio Client Advanced Edition V6.0

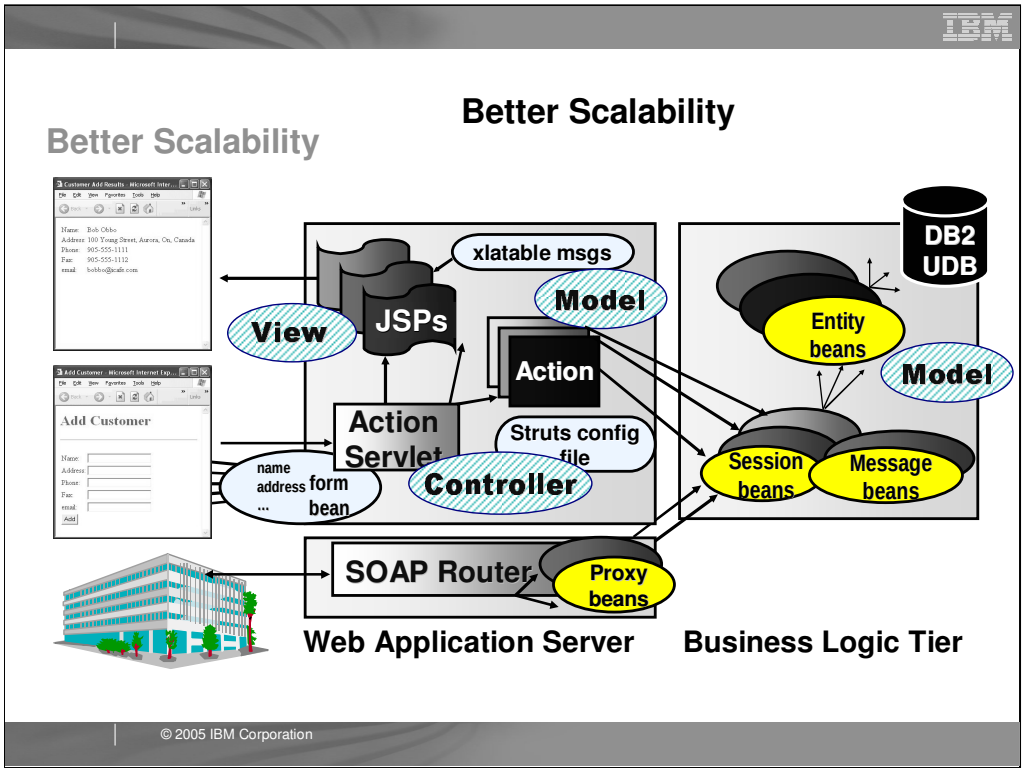
Workstation License
order through Passport Advantage
http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home

+CODE
+VisualAge RPG



5724–D46

www.ibm.com/software/awdtools/iseries



Better Scalability

Once again, you are looking at an architectural diagram… an optimal architecture that has been enhanced to use EJBs and MDBs in the third tier (shown in red text).

The important idea to take away from this diagram is that the Entity Beans are persisted to the database, and then session beans are used to "front end" the entity for a particular application. The message-driven beans are used if the Java Messaging Service is also being utilized for asynchronous message support.

The session bean proxies are called from the Action classes in the second tier. Similarly, a session bean proxy can be exposed as a Web service.

Table of contents

COBOL V5R3 and beyond

New Features in ILE COBOL V5R3

Compatibility & Portability with zSeries COBOL

- **Large VALUE clause support**
- **XML PARSE statement**
- **Alternate record key support**
- **DBCS data name support**
- **New CONSTANT data type**
- **Packed decimal, zoned decimal & numeric-edited item extended from 31 to 63 digits**
- **Seven new ANSI 85 intrinsic function**
- **Support of program status structure**
- **New PROCESS statement options**

ISO2000 for Constant

COBOL V5R4

Version 5 Release 4

- **XML GENERATE:**
- **Null-terminated Nonnumeric Literal:**
- **TRIM, TRIML and TRIMR Intrinsic Functions:**
- **DISPLAY-OF Intrinsic Function:**
- **NATIONAL-OF Intrinsic Function:**
- ***NOCOMPRESSDBG/*COMPRESSDBG Compiler Options:**

XML GENERATE:

XML support has been enhanced. A new statement, XML GENERATE, converts the content of COBOL data records to XML format.

Null-Terminated Nonnumeric literal

Null-terminated Nonnumeric Literal:

Nonnumeric literals can be null-terminated. This can be used anywhere a nonnumeric literal can be specified with the exception that null-terminated literals are not supported in "ALL literal" figurative constants. "Z" is used to indicate that a nonnumeric literal is null-terminated.

Example:

```
01 name1       pic X(10) value Z"TESTCASE1".
...
IF name1 = Z"TESTCASE1"
  DISPLAY "CORRECT".
```

TRIM, TRIML and TRIMR Intrinsic Functions

TRIM

The TRIM function returns the given string with any leading and trailing blanks removed, or the giving string with any leading and trailing specified characters removed. The type of the function is alphanumeric, DBCS or national depending on the class of its argument.

Example:

```
FUNCTION TRIM("xxxABxCxxx", "x") // returns 'ABxC'
```

TRIM, TRIML and TRIMR Intrinsic Functions

TRIML

The TRIML function returns the given string with any leading blanks removed, or the giving string with any leading specified characters removed.
The type of the function is alphanumeric, DBCS or national depending on the class of its argument.

Example:

```
MOVE "xyz" TO tc.
FUNCTION TRIML("xxyyzzyyzzABCxyzyxzxy", tc)
// returns 'ABCxyzyxzxy'
```

TRIM, TRIML and TRIMR Intrinsic Functions

TRIMR

The TRIMR function returns the given string with any trailing blanks removed, or the giving string with any trailing specified characters removed.
The type of the function is alphanumeric, DBCS or national depending on the class of its argument.

Example:

FUNCTION TRIMR(">>>>ABC<<<<<", "<>") // returns '>>>>ABC'

Other enhancements . . .

DISPLAY-OF Intrinsic Function:

The DISPLAY-OF function returns an alphanumeric character string consisting of the content of argument-1 converted to a specific code page representation. The code page representaion is determined by the CCSID compiler option which defaults to the job CCSID.

NATIONAL-OF Intrinsic Function:

The NATIONAL-OF function returns a national character string consisting of the UCS-2 representation of the characters in argument-1.

***NOCOMPRESSDBG/*COMPRESSDBG Compiler Options:**

CRTBNDCBL / CRTCBLMOD options *NOCOMPRESSDBG/*COMPRESSDBG enable the user to specify on the COBOL create commands whether listing view compression should be performed by the compiler when DBGVIEW option *LIST or *ALL is specified.

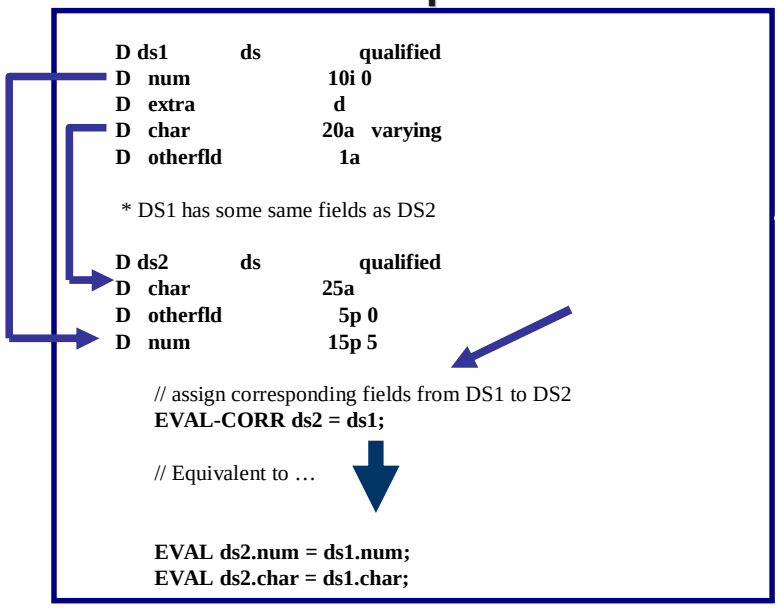
Table of contents

EVAL Corresponding

EVAL-CORR

An “EVAL-CORR" can be coded as opcode to move subfields in data structures with the same field names.

Example



EVAL-Corresponding

- While EVAL does not set the %NULLIND value, EVAL-CORR will (Only if ALWNULL(*USRCTL) is specified.
- Listing would contain a table of what will happen with every subfield

EXAMPLE

E V A L - C O R R S u m m a r y

EVAL-CORR	summary 1	14
	CHAR	Assigned; target and source are compatible
*RNFxxxx	OTHERFLD	Not same data type in source and target
	NUM	Assigned; target and source are compatible
*RNFxxxx	EXTRA	In source only

XML Support overview …

- **XML-INTO variable %XML(xml_document)**
  - Reads from an XML document into a field, data structure or array
- **XML-INTO %HANDLER(prototype : communication_area) %XML(xml_document)**
  - Reads from the XML document where there are zero or more repeated XML elements all the same.
  - Repeatedly builds up an array of information about the XML elements and passes it to the handling procedure indicated by the prototype, until all the XML elements have been handled. The procedure also gets passed the communication_area variable as a parameter.
- **XML-SAX %HANDLER(prototype : communication_area) %XML(xml_document)**
  - Reads from the XML document. For every XML *event (start-element, character-data, attribute-value etc),* it calls the handling procedure indicated by the prototype passing it information about the XML event. The handling procedure also gets passed the communication_area variable as a parameter.
  - First parameter of %XML can be a string actually containing XML data, for example '<a>Hello', or it can be the name of an XML document, for example 'mydata.xml'.
  - Second optional operand for %XML specifies options to control the parsing of the XML document. (It doesn't have to be a literal.)

  For example, %XML(xmldata : 'doc=file')

  indicates that the first operand of %XML is the name of a file. Other options for XML-INTO control how to handle extra data or missing data, in the XML document. The "path" option indicates where to find the data in the XML (the necessary data may not be in the very first XML element).

XML Support

EXAMPLE: XML-INTO variable %XML(xml_document)

```
D*************************************************
*
D info          DS
D   name                  10A
D   val                   5I 0  DIM(2)
D*************************************
/free

   XML-INTO info %XML('myfile.xml' : 'doc=file');
   // info now has the value
   //    name = 'Jill'
   //    val(1) = 10
   //    val(2) = -5
```

MyFile.XML

```
<info>
  <name>George</name>
  <val>10</val>
  <val>-5</val>
</info>
```

Target Field

Source of XML

Options

EXAMPLE: XML-INTO with Handling procedure …

XML Support

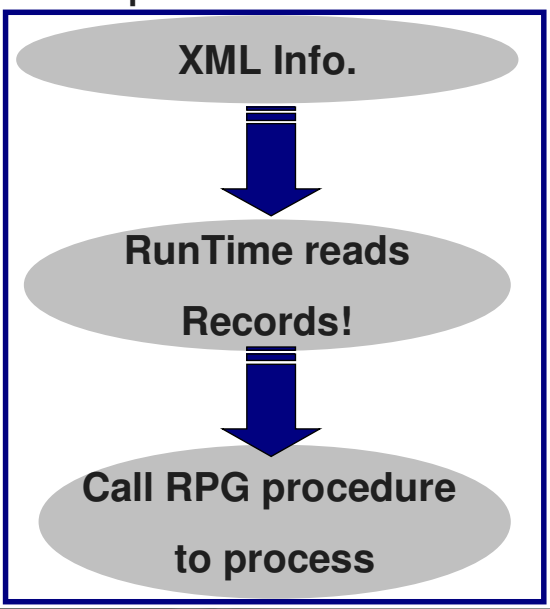
```
// DDS for "MYFILE"
// A        R PARTREC
// A          ID          10P 0
// A          QTY         10P 0
// A          COST         7P 2

// XML data in "partData.xml"
// <?xml version="1.0" standalone="yes" ?>
//  <parts>
//
<part><qty>100</qty><id>13</id><cost>12.03</cost></par
t>
//
<part><qty>9</qty><id>14</id><cost>3.50</cost></part>
//  ...
//
<part><qty>0</qty><id>254</id><cost>1.98</cost></part>
//  </records>
```

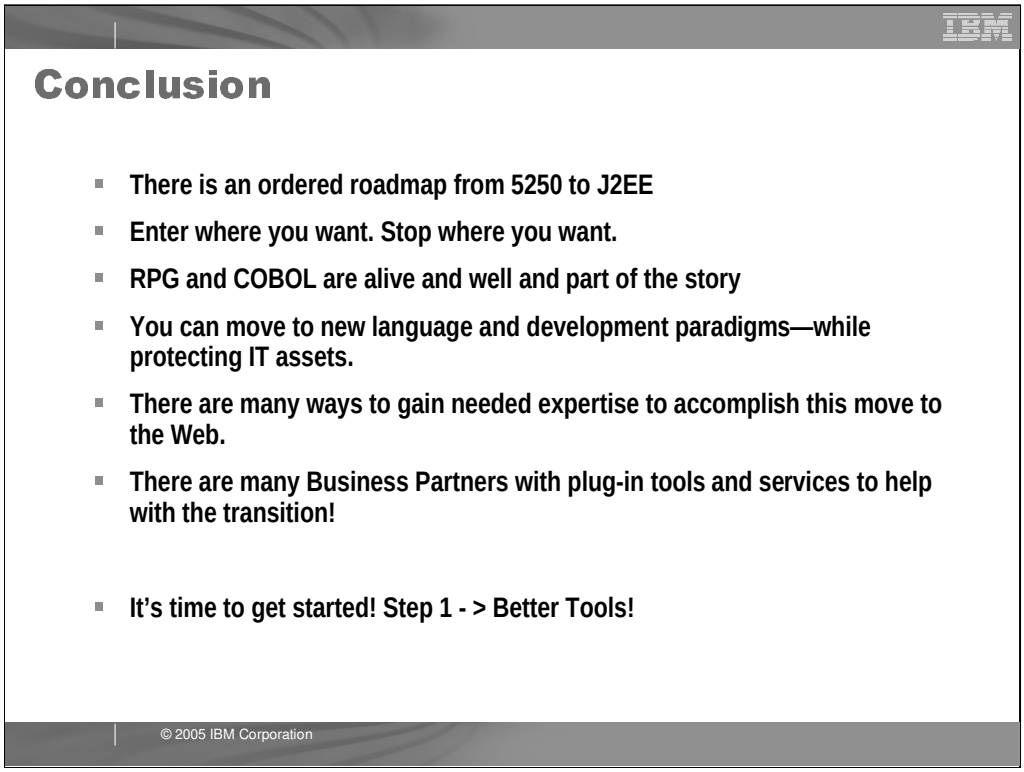
```
Fmyfile    if   e             disk
D options          S             100A
D partHandler      PR             10I 0
D   numErrs                       5P 0
D   parts                               LIKEREC(partrec)
                                        DIM(10)
D   numRecs                       10U 0 VALUE
D numErrors        S              5P 0 INZ(0)
:
:
:
/free
   // Initiating the parsing
   options = 'doc=file path=parts/part';
   xml-into %HANDLER(partHandler : numErrors)
            %XML('partData.xml' : options);
   // Communication-area variable "numErrors" holds
   // the number of file-output errors discovered by
   // the handling procedure.
/end-free
```

EXAMPLE: XML-INTO with Handling procedure …


```
// The procedure to receive the data from up to 10 XML elements at a time.  The first call to the
// this procedure would be passed the following data in the "parts" parameter:
//   parts(1) .id = 13   .qty = 100  .cost = 12.03
//   parts(2) .id = 254  .qty = 0    .cost = 1.98
// If there were more than 10 "part" child elements in the XML file, this procedure would be called more
// than once.
P partHandler    B
D                PI          10I 0
D   numErrs                  5P 0
D   parts                           LIKEREC(rec) DIM(10)
D   numRecs                  10U 0 VALUE
D i              S           10I 0
 /free
  for i = 1 to numRecs;
    write(e) partRec parts(i);
    if %error();
      // "numErrs" is the communication-area
      // parm, passed as the second operand of
      // %HANDLER on the XML-INTO operation
      numErrs += 1;
    endif;
  endfor;
  return 0; // continue parsing
 /end-free
P                E
```

EXAMPLE: The process





Conclusion

As you have gone through this course, we suspect it is possible you have run into many terms and concepts that are foreign to you—or are at best only fleetingly understood by you at present. If we tried to cover everything in detail that was summarily mentioned, this course would take weeks to complete. Instead, this course is intended to leave you with the following messages:

There is an ordered course in which to move your green screen-based application set to a much more progressive Web-based application set that supports your company's needs in today's always changing, ever demanding business environment.

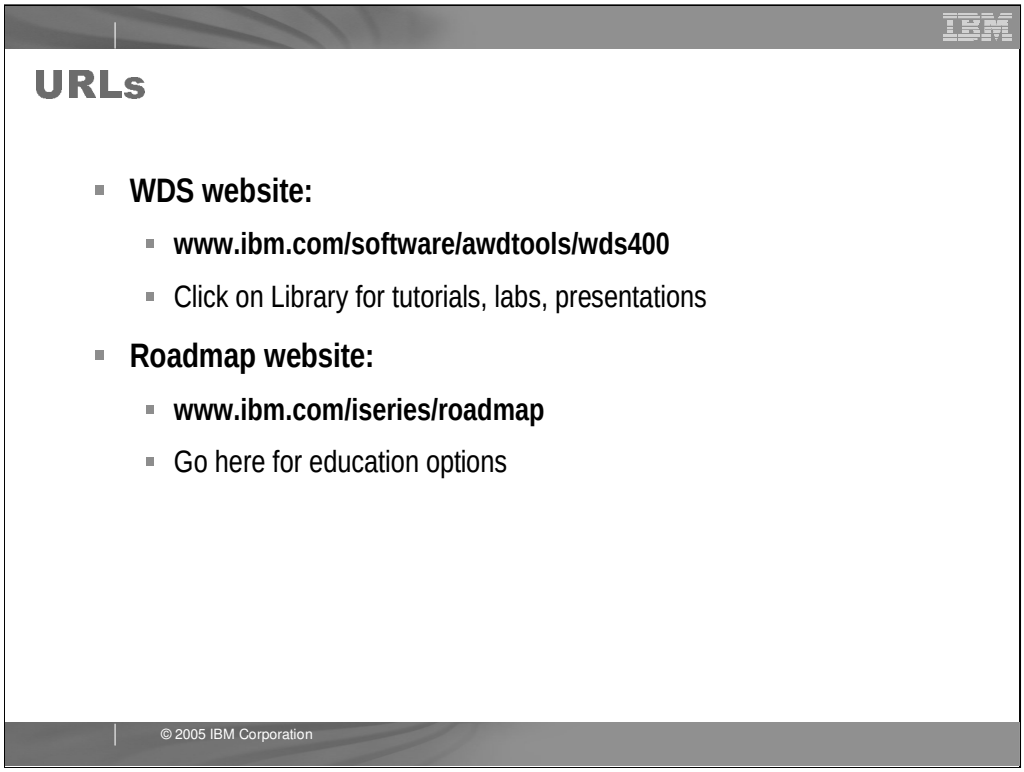
This roadmap has been designed to allow you to move, ultimately, to new programming languages and new development paradigms in a manner that both achieves your company's goals in a timely manner and protect the valuable IT assets in which you already have invested (line-of-business applications and many man years of experience).

There is a great deal to learn… and an inordinate number of methods to gain that learning (on-line training courses, classroom training, Web sites, Redbooks and redpieces, white papers, supportlines, and much more).

WDSC-L Dist list

To subscribe, unsubscribe, or change list options, visit:
http://lists.midrange.com/mailman/listinfo/wdsci-l

To post a message email→
WDSCI-L@midrange.com



Conclusion

As you have gone through this course, we suspect it is possible you have run into many terms and concepts that are foreign to you—or are at best only fleetingly understood by you at present. If we tried to cover everything in detail that was summarily mentioned, this course would take weeks to complete. Instead, this course is intended to leave you with the following messages:

There is an ordered course in which to move your green screen-based application set to a much more progressive Web-based application set that supports your company's needs in today's always changing, ever demanding business environment.

This roadmap has been designed to allow you to move, ultimately, to new programming languages and new development paradigms in a manner that both achieves your company's goals in a timely manner and protect the valuable IT assets in which you already have invested (line-of-business applications and many man years of experience).

There is a great deal to learn… and an inordinate number of methods to gain that learning (on-line training courses, classroom training, Web sites, Redbooks and redpieces, white papers, supportlines, and much more).

Trademarks & Disclaimers

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.


WebSphere Development Studio (WDS)
WebSphere Development Studio Client (WDSc)
© 2005 IBM Corporation