Tutorial: Maintaining an ILE RPG application: Introducing a new level of server tool integration for iSeries application development

About the tutorial

This tutorial through a series of exercises, checkpoints and practices, introduces you to a new level of tool integration for iSeries application development. The iSeries server development capabilities of the new integrated development environment (IDE) provides many compelling reasons for you to upgrade from Application Development Toolset (ADTS) or CoOperative Development Environment (CODE).

The IDE reduces the learning curve by providing a consistent interface for developing server applications and e-business applications. This allows you to progress to new levels of application development. The IDE delivers on the promise of tool integration with best-of-breed tools from IBM and several partners to support the end-to-end application development life cycle.

The IDE also includes a robust, easy-to-use development environment for creating, building and maintaining iSeries RPG, COBOL, C, C++ applications, and Web-enabled applications using the IBM WebFacing Tool.

In this tutorial you will follow the typical iSeries application development life cycle tasks using a payroll application written in RPG. You will work through a series of exercises that will teach you what the new tools are and how to use them.

By the end of this tutorial you will be able to realize significant productivity and usability gains in your iSeries server application development and be on your way to increasing your skills to make the transition into new programming models such as Java, Web, Web Services, and XML.

Completing this tutorial should take you about 2 hours.

Work on the exercises in sequence. The pictures in the exercises show similar tasks. Some of the names and icons may be different from the environment you will be working with when you complete the exercises.

Following each exercise it is recommended that you take the checkpoint quiz. Taking each quiz will help you to determine if you have mastered the exercise content and are ready to move on to the next exercise. The quiz answers are included at the end of this tutorial so you can check your answers.

Following each exercise checkpoint is a practice section. This section gives you the opportunity to apply what you have learned in the exercise. Often you will complete different but similar tasks to

what you have just learned in the exercise. The Development Studio Client for iSeries online help can assist you in completing the practice tasks.

The tutorial is available in PDF format. You can view the PDF version from the WebSphere Development Studio Client for iSeries Version 5.1 product CD or the product Web site (ibm.com/software/adwtools/iseries).

Tutorial business problem

You are an iSeries RPG application developer but you are familiar with basic Microsoft Windows operations such as working with the desktop and basic mouse operations such as opening folders and performing drag-and-drop operations. You are currently using the iSeries Application Development Tools (ADTS) products (PDM, SEU, SDA) or CoOperative Development Environment (CODE). ADTS has been the traditional method for developing and maintaining server-side iSeries applications. But now there is a new set of highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offer you a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications. The new tools also allow you to leverage the classic CODE tools for extremely rich editing and DDS design support. These new generation tools offer rich support for exploring the file system, compiling/building, editing, running, and debugging. They offer significant productivity and usability gains, support for disconnected and team development, and a common harness for the tight integration of IBM and partner-supplied tools for server development.

You know this is the place you and your company want to be especially as you both transition to delivering iSeries applications through the Web. Your first step in modernizing your iSeries applications is to move to the next generation of iSeries server application development tools. Here is what you and your company know about these next generation tools.

The first server application development tool is the Remote System Explorer which has its own tools and views. Remote System Explorer is similar to Programming Development Manager (PDM) in that it allows you to drill down to 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 you to explore iSeries jobs and commands, and the IFS file system. Further, you can also use it to explore the file system of remote Linux, UNIX and Windows systems.

The Remote System view is the primary drill-down view, similar to PDM. Double-click a member and you open the Remote Systems LPEX Editor, built right into the IDE and with rich editing functions. This editor goes well beyond SEU's function; it is a superset of SEU! and is getting close to all of the CODE Editor's functions. The syntax checker is ported down from SEU, the compilers are imbedded for verifying errors and the reference manuals are built-in and F1 cursor sensitive. You can see the program hierarchy in the Outline view. The content assist is a popular feature. There is explicit and rich iSeries specific support for not only edit, but verify, compile, run and debug of RPG, COBOL, C, C++, CL and DDS not just from the Remote Systems LPEX Editor but also from the IDE in your very own Remote System Explorer perspective; the place for views and tools specific for iSeries server application development.

Instead of the Remote Systems view which is a tree view, PDM users are use to a table, so there is an iSeries Table view that shows what the tree shows but in table format. You can sort the columns when you click on a column heading. The contents of the table are easily replaced when you click on a file in the Remote Systems view. There are right-click actions that are the same as PDM's and a command line at the bottom of the table just like PDM.

There is no built-in tool for display file and printer file development yet, but you can easily launch the CODE Designer from the Remote Systems view.

RPG program debug is easy from the Remote System Explorer interactive debug perspective. Through this perspective you can debug Java, OPM/ILE, RPG, COBOL, CL, ILE C and C++ programs.

With this information in hand you are ready to move on.

Before you begin

Before you begin, you must install the following:

On iSeries:

- OS/400 Version 5 Release 1 or later. The Remote System Explorer component will need additional PTFs. The PTF information is available at ibm.com/software/adwtools/iseries. OS/400 is needed since this tutorial includes programming objects located on the iSeries server and the applications contain backend code on the iSeries server.
- IBM WebSphere Development Studio for iSeries (5722-WDS)
- IBM WebSphere Development Studio for iSeries requires V5R2M0 OS/400 (5722-SS1)
- GUIDESWSS.savf and RSELABXX restored on iSeries
- iSeries system servers started with the command STRHOSTSVR *ALL and STRTCPSVR *DDM.

On the workstation for iSeries server application development:

- Intel Pentium II processor
- 256 MB RAM minimum
- Required hard drive space: 1650 MB
- Additional 700MB of temporary hard drive space is needed during product installation
- Windows: VGA graphics card (800 x 600, or higher, recommended, 256 colors)
- CD-ROM/DVD drive
- Mouse or pointing device

Software

- IBM WebSphere Development Studio Client for iSeries, Version 5 (5724-A81)
- eNetwork Personal Communications Version 5.5 or later
- TCP/IP access to an iSeries system.
- For Windows NT:
 - Windows NT Server or Workgroup, Version 4.0, Service Pack 6a or higher
 - Microsoft Internet Explorer 5.5 Service Pack 1 or higher
- For Windows 2000:
 - Windows 2000 Professional, Service Pack 2 or higher
 - o Microsoft Internet Explorer 5.5 Service Pack 1 or higher
- For Windows XP
 - Windows XP Professional

To use the library for this tutorial, you must restore it to your iSeries.

On your iSeries, create a save file: CRTSAVF <library_name>/RSELAB

On your workstation, open a Command Prompt window and go to the directory where you downloaded the save file and do the following:

ftp <iSeries_name> <user_name> <password> binary put rselab.sav <library_name>/RSELAB quit

On your iSeries, restore the RSELAB library: RSTLIB SAVLIB(RSELAB) DEV(*SAVF) SAVF(<library_name>/RSELAB)

Once you have restored the library, you can delete the save file library_name>/RSELAB.

Conventions that are used in this tutorial

This tutorial uses typographical conventions in the text to help you distinguish between the names of controls and text that you type. For example:

• Menu items are in boldface font:

Click Menu → Menu choice

• The names of fields, check boxes, and buttons are also in boldface font:

Type text in the Field field.

• Text that you type is in example font on a new line:

This is the text that you type.

Related information

This tutorial covers the most common tasks that you can accomplish with the Remote System Explorer. For more information about related tasks, see the following documents: WebSphere Development Studio Client for iSeries

- Maintaining and Developing iSeries applications online help
- Remote System Explorer presentation on the Library page (ibm.com/adwtools/iseries)
- Remote System Explorer demo on the Library page (ibm.com/adwtools/iseries)
- iSeries Server Application Development Quick Tour on the Library page (ibm.com/adwtools/iseries)

Exercise 1: Introducing to WebSphere Development Studio and Remote System Explorer

In this exercise you are introduced to the IBM WebSphere Development Studio for iSeries (Development Studio) product and its relationship to IBM WebSphere Development Studio Client for iSeries. You learn which product makes up the host components and which product makes up the workstation components. You recognize the server application development tools included with Development Studio Client for iSeries programmers. You then are introduced to Remote System Explorer the launching point for iSeries server application development tools.

At the end of the exercise, you should be able to:

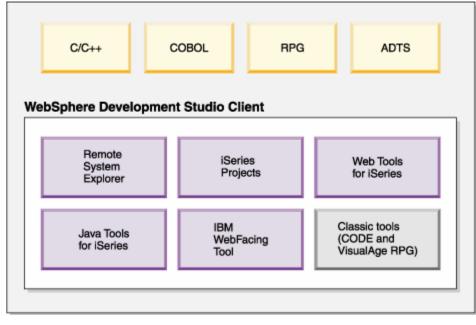
- Describe the Development Studio product
- Describe the Development Studio Client product
- Describe how Development Studio Client fits into the WebSphere Studio family of products
- Explain the tools available to iSeries programmers for iSeries application development

Let's get started by first looking at what makes up the Development Studio and Development Studio Client products.

Development Studio and Development Studio Client

With Development Studio Client, you can quickly develop and deploy traditional and e-business applications for your iSeries system. You receive unlimited licenses of this powerful suite of tools when you use Development Studio for your host development. The following diagram illustrates the interaction between host and client tools:

WebSphere Development Studio



©Copyright IBM Corporation 2003

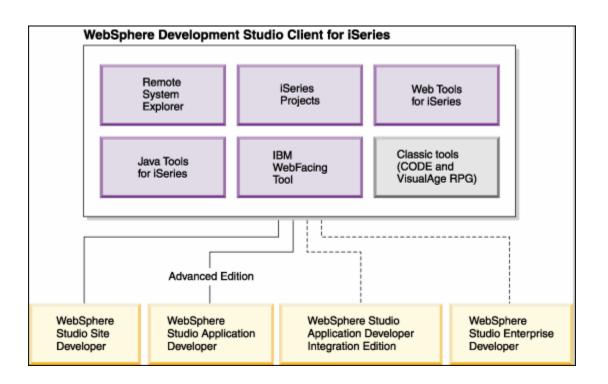
Course material may not be reproduced in whole or in part without the prior consent of IBM Development Studio Client is designed to help you:

- 1. Develop and maintain iSeries applications using the Remote System Explorer
- 2. Develop Web GUIs for iSeries classic applications using the IBM WebFacing Tool and other Web Tools
- 3. Develop client and server applications for iSeries using Java Tools
- 4. Work with other Integrated Site Developer Tools (XML, Web Services, SQL, Relational Databases)

Now, You know Development Studio Client makes up the workstation tools while Development Studio makes up the host tools. But there's more. WebSphere Studio is IBM's solution for application and Web development. Both versions of our product come with an additional base WebSphere Studio product.

- IBM WebSphere Development Studio Client for iSeries includes WebSphere Studio Site Developer. Site Developer includes support for Web services, XML development tools, and core support for Java and Web development tools.
- **Advanced** IBM WebSphere Development Studio Client Advanced Edition for iSeries includes WebSphere Studio Application Developer. This base product provides end-to-end support for the creation and maintenance of J2EE applications and Web services. It also provides extensive support for Enterprise Java Beans, and for Java Messaging services.

And if you happen to have other WebSphere products installed you can also install either edition of the product on top of WebSphere Studio Application Developer - Integration Edition or WebSphere Studio Enterprise Developer. The following diagram illustrates how this product fits into the WebSphere Studio family of products:



iSeries Server Application Development Tools

Now, you know what the two flavors are of Development Studio Client and why you would want to use each one. Next let's look at those next generation iSeries server application development tools mentioned at the beginning of this tutorial. Just what are they and what do they do.

Remote System Explorer

You can manage your development-cycle tasks in the Remote System Explorer. This is an enhanced and more flexible workstation version of Program Development Manager (PDM). You can create and manage development projects on your iSeries system from your Windows-based workstation with the Remote System Explorer and iSeries projects. With these tools, you can connect to an iSeries remote host, view iSeries libraries, files, and members. You can also launch the host compilers, the workstation editor, a program verifier and various debuggers all from the Remote System Explorer. This tool also supports other system types, such as UNIX^(R), Linux, and Windows.

LPEX Editor

Your program editing tasks are simplified with the Remote Systems LPEX Editor. This is a powerful language-sensitive editor that is easy to customize. Token highlighting of source makes the various program elements stand out. It has SEU-like specification prompts for RPG and DDS to help enter column-sensitive fields. Local syntax checking and semantic verification for your RPG, COBOL and DDS source makes sure it will compile cleanly on an iSeries system. If there are verification errors,

an Error List lets you locate and resolve problems quickly. On-line programming guides, language references, and context-sensitive help make finding the information you need just a keystroke away.

Shells and Commands in the Remote Command view

You can use the Remote Commands view to run and interact with commands and command shells on universal systems. A universal system includes Windows, Linux, and UNIX system types. Specifically, use the view to:

- Run commands in a command shell
- Display and interpret the output of a program
- Enter input to a program
- Display and manage different commands and shells from the same view. Multiple commands can be run in a single shell (one command at a time per shell), multiple shells may be run on a single system, and multiple systems may be running shells.

Whenever a command shell is launched or a command is run from the Remote System Explorer, the Remote Commands view displays the output and provides a way to work with that output.

Program Verifier

One of the most powerful and unique features of the Remote System Explorer is the Program Verifier. Before you compile your code on an iSeries system, you can ensure that there are no errors by invoking the Program Verifier. The verifier checks for semantic (compile) errors on your workstation so that you can guarantee a clean compile on the iSeries. Think of the host cycles you'll save. It is especially handy when you are writing code but you are disconnected from an iSeries system. You can do this because Remote System Explorer ported the parsing and checking code from the iSeries host compilers to the workstation. 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.

CODE Designer

Using an editor to create and maintain DDS source for your display and printer files can be a frustrating and difficult task. What would be great is a graphical design tool that lets you design your screens and reports visually and then generate the DDS source for you. Well, that's exactly what the CODE Designer does for you.

The CODE Designer interface was designed to help the novice DDS programmer create screens, reports and databases quickly and easily without worrying about the details of the DDS language, while at the same time letting the expert DDS programmer get access to all the features and power of the language. Let's now step through each part of the interface and update some DDS as well.

iSeries Debugger

With the integrated iSeries debugger you can debug your program running on the iSeries system from a graphical user interface on your workstation. You can also set breakpoints before running the debugger, by inserting breakpoints directly in your source while editing. The integrated iSeries debugger client user interface also enables you to control program execution. For example, you can run your program, set line, watch, and service entry point breakpoints, step through program instructions, examine variables, and examine the call stack. You can also debug multiple applications, which may be written in different languages, from a single debug window. Each session you debug is listed separately in the Debug view. In this tutorial you are going to learn some of the basic features and functions of each of these tools. We are confident that Development Studio Client will save you lots of time and effort in your day-today programming tasks. It will make you a more efficient and effective programmer. At the same time, it will save cycles on your iSeries and better yet it will get you ready for the next step and that is moving your iSeries applications to the Web.

So, let's get started! Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. WebSphere Development Studio for iSeries:
 - A. Includes all four host compilers and all traditional tools (ADTS)
 - B. Includes all four host compilers, all traditional tools (ADTS) and unlimited licenses of the workstation-based tools named Development Studio Client
 - C. Includes only the workstation-based tools named Development Studio Client
 - D. Includes only the four host compilers
- 2. WebSphere Development Studio Client for iSeries Version 5 includes:
 - A. WebSphere Studio Site Developer Version 5 for e-business development
 - B. Cooperative development environment (CODE)
 - C. VisualAge RPG
 - D. Java tools
 - E. Web tools
 - F. WebFacing tool
 - G. All of the above
- 3. WebSphere Development Studio Client for iSeries Advanced Version 5 includes:
 - A. WebSphere Studio Application Developer Version 5 for e-business development
 - B. Cooperative development environment (CODE)
 - C. VisualAge RPG
 - D. Java tools
 - E. Web Tools
 - F. WebFacing Tool
 - G. All of the above
- 4. WebSphere Studio Application Developer includes support for:
 - A. Creation and maintenance of J2EE applications
 - B. Creation and maintenance of Web services
 - C. Enterprise Java Beans
 - D. Java Messaging Services
 - E. All of the above
- 5. WebSphere Studio Site Developer includes support for:
 - A. Web services
 - B. XML development tools
 - C. Java tools
 - D. Web tools
 - E. All of the above
- 6. You can manage your development-cycle tasks in:

- A. The Remote System Explorer
- B. iSeries Projects
- C. The IBM WebFacing tool
- D. All of the above
- With the Remote System Explorer and iSeries Projects you can view iSeries libraries, files and members. You can also launch the host compilers, the workstation editor, and various debuggers. (T, F)
- 8. Your program editing tasks are simplified with the:
 - A. The Remote System Explorer
 - B. iSeries Projects
 - C. The IBM WebFacing tool
 - D. The LPEX Editor
 - E. All of the above
- The editor can access source files on your workstation or on your iSeries system directly. When a compilation results in errors, you can jump from the compiler messages to an editor containing the source. The editor opens with the cursor positioned at the offending source statements so that you can correct them. (T, F)
- 10. You can debug your program running on the iSeries system from your workstation using:
 - A. The Remote System Explorer
 - B. iSeries Projects
 - C. The IBM WebFacing tool
 - D. The LPEX Editor
 - E. The Integrated iSeries Debugger
 - F. All of the above
- 11. The graphical design tool that lets you design your screens and reports visually and then generates DDS source for you is:
 - A. The Remote System Explorer
 - B. CODE Designer
 - C. The IBM WebFacing tool
 - D. The LPEX Editor
 - E. The Integrated iSeries Debugger
- 12. Before you compile your code on an iSeries system, you can ensure that there are no errors by invoking the:
 - A. The Remote System Explorer
 - B. CODE Designer
 - C. The IBM WebFacing tool
 - D. The LPEX Editor
 - E. The Integrated iSeries Debugger
 - F. Program Verifier
- 13. You can use the Remote Commands view to:
 - A. Run commands in a command shell
 - B. Display and interpret the output of a program
 - C. Enter input to a program
 - D. Display and manage different commands and shells from the same view
 - E. All of the above

- 14. The integrated iSeries Debugger enables you to run your program, set line, watch, and service entry point breakpoints, step through program instructions, examine variables, and examine the call stack. (T, F)
- 15. If the Advanced Version of the product is not installed on your workstation then you will not see the word Advanced in the Start menu. (T, F)

What you just did

In this exercise you were introduced to Development Studio and Development Studio Client and how Development Studio Client fits in the WebSphere Studio family of products, the difference between Development Studio Client and Development Studio Client Advanced and what iSeries server application development tools the Development Studio Client workbench provides.

In the next exercise you will start Development Studio Client and open the Remote System Explorer.

Exercise 2: Starting Development Studio Client and Opening the Remote System Explorer

In this exercise you are introduced to workbench, the workspace, a perspective and specifically the Remote System Explorer perspective. You then learn how to start Development Studio Client and how to open the Remote System Explorer perspective.

At the end of the exercise, you should be able to:

- Explain workspace, workbench and perspectives
- Describe the Remote System Explorer perspective
- Start Development Studio Client
- Open the Remote System Explorer perspective

First let's start Development Studio Client.

Starting Development Studio

To start Development Studio Client:

- 1. Click Start on the task bar of your desktop
- 2. Select Start ->Programs-> IBM WebSphere Studio -> Development Studio Client Advanced Edition for iSeries

Note: If the **Advanced** Version of the product is not installed on your workstation then you will not see the word **Advanced** in the Start menu

A window may appear. Here you specify the name of the workspace where your projects and other resources such as folders, subfolders and files that you are developing in the workbench will reside.

WebSphere Studio	×
WebSphere Studio stores your work in a directory called a workspace. You can o you start WebSphere Studio. Specify the directory to use for this session:	change the workspace each time
C:\Documents and Settings\chng1me\My Documents\IBM\wdscrselabxx	Browse
Use this workspace as the default and do not show this dialog box again	
	OK Cancel

Figure 1: WebSphere Studio window for specifying workspace directory name

3. (Optional) Change the field in this window and use a unique directory name, for example, rselabXX (where XX is your unique number). If you use 01 use directory name rselab01 as shown in Figure 1. After a few moments of loading, the workbench appears.

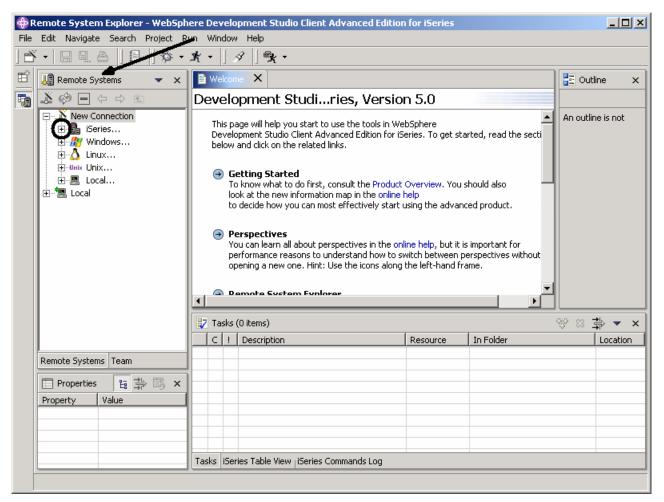


Figure 2: Workbench with Remote System Explorer

The workbench refers to the desktop development environment. The workbench aims to achieve seamless tool integration and controlled openness by providing a common paradigm for the creation, management, and navigation of workbench resources. Each workbench window contains one or more perspectives.

Open Remote System Explorer perspective

1. Check for the name of the perspective, the arrow in Figure 2 indicates where to look for the

perspective name.

A perspective defines the initial set and layout of views in the Workbench window. Within the window, each perspective shares the same set of editors. Each perspective provides a set of capabilities aimed at accomplishing a specific type of task or working with specific types of resources. For example, the Java perspective combines views that you would commonly use while editing Java source files, while the Debug perspective contains views that you would use while

debugging Java programs. Perspectives contain views and editors and control what appears in certain menus and tool bars.

If you see a different perspective, not the **Remote System Explorer** open in the workbench or no

perspective:

- 2. Select Window from the workbench Menu bar
- 3. Select Open Perspective

4. Select Remote System Explorer

You work in the Remote System Explorer perspective in the workbench. This perspective is for an iSeries programmer to display the connections that you have already configured, create a new connection, connect to and disconnect from the connections that you have defined, work with iSeries files, commands, jobs, and integrated file system files. This is the active perspective when you start Development Studio Client with a new workspace. If you had used the workspace before then, the workbench would come up with the perspective that you last opened. You will learn more about the Remote System Explorer perspective in the coming exercises as this is where you launch the iSeries programmer tools and views from the workbench.

Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. A workspace:
 - A. Aims to achieve seamless tool integration and controlled openness by providing a common paradigm for the creation, management, and navigation of workbench resources.
 - B. Defines the initial set and layout of views in the Workbench window.
 - C. Refers to the desktop development environment.
 - D. Specifies where your projects and other resources such as folders, subfolders and files that you are developing in the workbench will reside.
- 2. A workbench:
 - A. Aims to achieve seamless tool integration and controlled openness by providing a common paradigm for the creation, management, and navigation of workbench resources.
 - B. Defines the initial set and layout of views in the Workbench window.
 - C. Refers to the desktop development environment.
 - D. Specifies where your projects and other resources such as folders, subfolders and files that you are developing in the workbench will reside.
 - E. A and C
- 3. A perspective:
 - A. Aims to achieve seamless tool integration and controlled openness by providing a common paradigm for the creation, management, and navigation of workbench resources.
 - B. Defines the initial set and layout of views in the Workbench window.
 - C. Refers to the desktop development environment.

- D. Specifies where your projects and other resources such as folders, subfolders and files that you are developing in the workbench will reside.
- 4. Match the perspective with its correct definition.
 - A. Combines views that you would commonly use while editing Java source files
 - B. Contains views that you would use while debugging Java programs
 - C. Contains views that you would use while developing Web applications
 - D. Contains views that you would use while maintaining iSeries applications.
 - A. Java perspective
 - B. Web perspective
 - C. Remote System Explorer perspective
 - D. Debug perspective
- 5. In the Remote System Explorer perspective you can:
 - A. Display configured connections
 - B. Create a new connection
 - C. Connect and disconnect defined connections
 - D. Work with iSeries files, commands, jobs, IFS files
 - E. All of the above

Practice

Given your experience in opening the Remote Systems Explorer perspective, open the Web perspective. Explore the tools and views available to the Web developer. Next open the Java perspective. Explore the tools and views available to the Java developer. Now since you are supposedly in the Java perspective, open the Web perspective. Be careful not to open another Web perspective.

Tip: Look in the workbench left-frame for the Web perspective icon. Now close both the Java perspective and the Web perspective.

What you just did

In this exercise you were introduced to workbench, the workspace, a perspective and specifically the Remote System Explorer perspective. You then learned how to start Development Studio Client and how to open the Remote System Explorer perspective.

In the next exercise you will configure a connection to an iSeries system and then connect to your iSeries system.

Exercise 3: Configure a connection to an iSeries system and Connect to your iSeries

In this exercise you create a connection to an iSeries server and select objects using the Remote System Explorer perspective. You will learn the steps to create a connection. You will then learn how to find a library in your library list. Finally you open a member in the Remote Systems LPEX Editor. You also learn about several views such as the Remote Systems view, iSeries Table view, and the Outline view.

At the end of the exercise, you should be able to:

- Explain the Remote Systems view
- Configure a connection to an iSeries system
- Connect to an iSeries system
- Describe what you need to set up a connection
- Describe a profile
- Find an object in the Remote System Explorer perspective
- Explain subsystems
- View and access objects
- Explain the Outline view
- Explain iSeries Table View
- Explain the difference between the iSeries Table View and the Remote Systems view
- Show source physical file members in an iSeries Table view
- Lock and unlock the iSeries Table view
- Open a second source member
- View the outline of a file

Configuring a connection to an iSeries system

When you first open the Remote System Explorer, you are not connected to any system except your local hard drive on your workstation. To connect to a remote iSeries system, you need to define a connection. When you define a connection, you specify the name or IP address of the remote system and you give your connection a unique name that acts as a label in your workspace so that you can easily connect and disconnect. When you connect to the iSeries system, the workbench prompts you for your user ID and password on that host.

The first time you connect to an iSeries system, you need to define a profile. All connections, filters, and filter pools belong to profiles. We will describe filters in a later exercise. We will discuss profiles when you create your first connection.

Ok, let's get started. Remember you have already opened the Remote System Explorer perspective in the previous exercise.



Figure 3: Configure a connection

- 1. Switch to the Remote System Explorer perspective.
- In the Remote Systems view, New Connection is automatically expanded to show the various remote systems types you can connect to through the Remote System Explorer. Expand iSeries to configure a connection to an iSeries system

The New – Name personal profile page of the New Connection wizard appears:

New				
Name personal pro				_
Welcome to Remote you. Enter a profile r decide for each new profile.	name to unique	ely identify you	from your team m	embers. You will
Profile: weiss1				
	< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

Figure 4: New – Name personal profile page

On this page of the New connection wizard and since it is the first time you connect to an iSeries system, you need to define a profile. All connections, filters, and filter pools belong to profiles. Profiles offer a way to group connections, share connections, or keep them private, and they help you partition data if you have a lot of connections or filter pools. Your first profile will be for your local workstation. This is your personal profile. You use this profile as you want to keep your connection private. You do not want to share resources and information with other people.

3. Leave the Profile information as is. Click Next

The second page of the **New connection** wizard shows:

©Copyright IBM Corporation 2003

New					
Remote iSeries Sy Define connection in		ction			
Parent profile:	weiss1				•
	s400a				•
✓ Verify host name	:				
	< <u>B</u> ack	Next >	<u> </u>	ish	Cancel

Figure 5: The Remote iSeries Sysem Connection page

On this second page you specify the information for your connection. The cursor in the Remote iSeries System Connection page is positioned in the **Host Name** field.

4. Enter the name of your host system in the Host name field

The **Connection name** is automatically filled with the host name. Leave it this way. This name displays in your Remote Systems view and must be unique to the profile.

- 5. Leave the **Parent profile** as is. You don't need to change it. As the files will be kept private the parent profile defaults to the default profile shown on the previous page of the wizard. If you were sharing resources you would select a team profile that you would have created earlier.
- 6. Leave the Verify host name check box checked
- 7. Click **Finish** to define your system.

Connecting to your iSeries system

After you configure a connection to an iSeries system, you can easily connect and expand your new connection to reveal your subsystems. Subsystems are a function grouping of the various types of remote resources that can be explored in the remote system. There are four subsystems:

- iSeries Objects is a PDM-like group, allowing access to libraries, objects and members
- iSeries Commands allow you 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 allow you to see various jobs, subset by job attributes, and to perform a limited number of operations on those jobs.
- IFS Files allow you to explore folders and files in the Integrated File System of the remote iSeries system

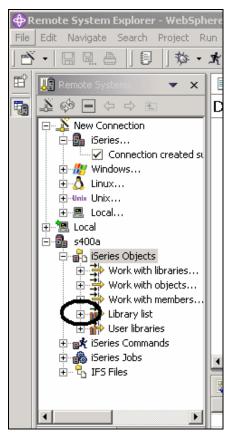


Figure 6: Remote Systems view with iSeries connection

To connect to an iSeries system:

- 1. In the Remote Systems view, your new connection is expanded to reveal your subsystems. 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.
- 2. Notice the first three entries under the iSeries Objects subsystem are named after the PDM options, because they have similar capabilities:
 - Work with libraries (similar to WRKLIBPDM)
 - Work with objects (similar to WRKOBJPDM)
 - Work with members (similar to WRKMBRPDM)

In addition there are entries for working with library lists and user libraries:

• Library list (to simulate PDMs STRPDM option 12 you can start with the pre-defined Library

list filter, that when expanded lists all libraries in your library list.)

User libraries

You also have more entries to work with under the connection itself and you can see from these entries that Remote System Explorer goes well beyond PDM! It allows you to explore iSeries jobs and commands and the IFS file system.

Now let's work with a library in our library list.

3. Click the plus sign beside Library list.

Now the connection will be activated and you will be prompted for a user ID and password.

🕀 Enter Passw	vord	×
System type:	iSeries	
Host name:	5400A	
User ID: 🔇	wdsclabxx	
	Permanently change	user ID
Password:		
(ave password	
	<u>o</u> k	<u>C</u> ancel
		,

Figure 7: Enter password window

- 4. Enter your user ID and password.
- 5. Check the check box Permanently change user id

- 6. Check the check box Save password
- 7. Click OK.

Make sure that your user ID has been setup, so that your library has been added to the library list automatically. You can use the properties of iSeries Commands to set connection information such as adding a library to a library list.

Note: if you do not see RSELABxx in your library list you can right click on the Library list and use the Library List Entry window to add RSELABxx to your library list.

Back in the workbench in the Remote System Explorer perspective you will see the libraries in your job's library list.

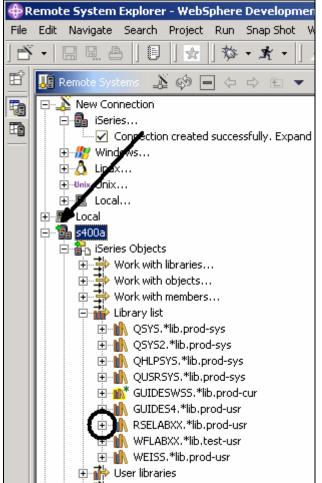


Figure 8: Remote Systems view with libraries in library list

Notice that the s400a node now has a small green arrow in the icon to indicate it is an active connection.

Also your **RSELABxx** library should have a small green star that indicates it is the current library. In Figure 8 the library **GUIDESWSS** is the current library, if you have sharp eyes you can spot the star in the icon.

For each library, you can right-click and select from a number of 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.

Viewing and accessing objects in the Remote System Explorer perspective

Now you are ready to view and access objects in your current library RSELABxx.

To view an object:

1. Expand library RSELABXX.

You will see all objects in this library appear in the Remote Systems view. For each object you can right-click and select from a number of actions also. The list of actions depends on the object selected and whether you selected one or multiple objects. For a source file the pop-up menu has an action to create a new member within the selected file, to refresh the contents of a file if it is expanded, to rename the file, copy the file and delete the file. These actions remotely run the appropriate iSeries command and you will see it logged in the Commands view.

- 2. Drill-down through the files in the Remote Systems view until you find **QDDSSRC** source file and then expand it.
- Drill-down through the files in QDDSSRC until you find QRPGLESRC source file and expand it as well.

⊕ R	emote System Explorer - WebSphere Developmen	t St
File	Edit Navigate Search Project Run Snap Shot W	indo
Ď	- 🛯 🖳 🗛 🛛 🗐 👷 🛛 🏘 • 🖈 • 🗍 🔗	9
Ē	🜆 Remote Systems 🏾 🏂 🏟 🖃 🗇 🖨 🔻	x
5	RSELABXX.*/lib.prod-usr	
Ð	CLC1.*pgm.clle	
	R CLR1.*pgm.clle	
	STRCODECPP,*pam.clp	
	⊡ ⊕ CPOEXER2.*file.pf-src	
	EVFEVENT.*file.pf-dta	
	⊕ MSTDSP.*file.dspf	
	⊕MSTDSP2.*file.dspf	
	🕀 🗐 PRJMST.*file.pf-dta	
	🕀 📳 QCBLLESRC.*file.pf-src	
	🕀 🗐 QCLSRC.*file.pf-src	
	QCMDSRC.*file.pf-src	
	DDDSSRC.*file.pf-src	
	EMPMST.pf	
	🐚 MSTDSP.dspf	
	MSTDSPV.dspf	
	🕼 MSTDSP2.dspf	
	PRJMST.pf	
	🕒 REFMST.pf	
	BSNMST.pf	
	QRPGLESRC.*file.pf-src	
	PAYROLL.rpgle	
		-1

Figure 9: Remote Systems view with expanded source files

Now you can see and access the members in these two source files. For each member you can right-click and select from a number of actions. The exact list of actions depends on whether the member is a data file or source file and whether you select one or multiple members. For a member, the popup menu actions are edit, rename, copy, move, delete, and compile.

🕀 R	emot	e System	Explore	er - We	bSphere	e Developr	nent St	udio Clier
File	Edit	Navigate	Search	Proje	ct Run	Snap Shot	Windo	w Help
ÌÈ	-		5] C]] 🖈	珍	• <u>*</u> •] \$	® ₹•
Ē		Remote Sys	tems	X Ø	•	¢ € .	• x	🖹 Weld
5					D.*pgm.			Deve
Ŧ					.G.*pgm.i ECPP.*pg			This
_					E.*file.pf			Dev
			÷-6 (POEXER	ι.*file.pf	src		belo
					2.*file.p			
					*file.pf-c			⊖
					T.*file.pf *© d0			
					*file.dspl 2.*file.dsp			
					*file.pf-d			
					RC.*file.			
			••• • • • ••	-)CLSRC,	*file.pf-s	rc		
					<u>c *filo pl</u>			
					C.*file.pf	-src		
				_	New			
				_	Go To		>	
				_	Refres	h		
				F	Expan			
			0	_	Collaps			
				👌 R: 🚽				
				-	Renam	e		
				Pr Pr	Сору			
			±	_	Paste			
				EFM	Move			
				SNM 🥈				
			÷-6 s	iour 🎽	Delete		_	
			EN N	IYCO	Show i	n Table		🍳 iSer
			1_	_	Cache	Filo Bosarip	lions	ID

Figure 10: Select Show in table for QDDSSRC file

Before you go ahead and work with these members, let's see the members in the iSeries Table view as well because that is more similar to the view you are used to from PDM. You use this view to display a list of items, for example, members or objects, in a table format similar to PDM. You can also perform actions against these items such as editing and compiling.

4. Right-click the QDDSSRC file and select Show in Table.

The iSeries Table view takes the selected object in the Remote Systems view as input, and displays the contents in the table. For source physical files, this step displays the members inside, their names, types, attributes, text descriptions, and status.

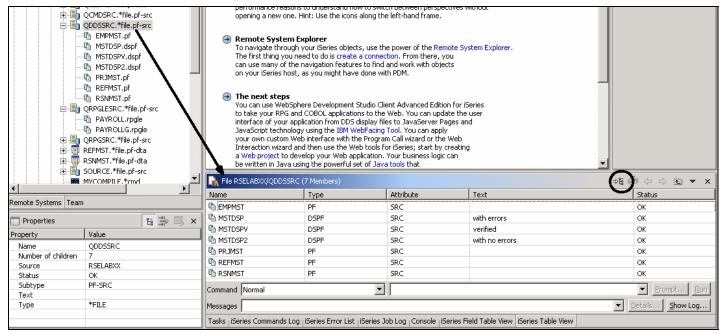


Figure 11: Table view with lock/unlock button

The top of the iSeries Table view contains a lock icon that controls the correlation between the Remote Systems view and the iSeries Table view.

If the lock is disabled (the default state) then whenever you click an object or library in the Remote Systems view, the associated contents of that item automatically populate the iSeries Table view.

If the lock is enabled then when you click on various items in the Remote Systems view, this view does not change the input to the iSeries Table view. The default is to have the lock disabled since you might want to explore various branches on the Remote Systems view while maintaining a specific list of items in the iSeries Table view.

To enable or disable the lock, you can click it once to change its state.

Tip: You click any of the column titles to sort by that attribute, for example, click **Type** to sort the items by type, which sorts them from A to Z. Click the same attribute again to reverse the order that the items are sorted, for example, click **Type** again to sort from Z to A.

©Copyright IBM Corporation 2003

You can unlock this iSeries Table view. It then switches to the selected object in the Remote Systems view, as you select objects in the Remote Systems view.

- 5. In the iSeries Table view toolbar make sure the lock/unlock button 🖄 is in the unlock position. Since it is in the unlock position by default it is now unlocked from the current QDDSSRC object. Meaning it is not locked to QDDSSRC.
- 6. In the Remote Systems view, select QRPGLESRC. The table shows the members in QRPGLESRC

📊 File CODELABXX/QRPGLESRC (2 Members) 👘 🖓 🖘 🕸 🕫 🗢 🔁 🔻 🗙						
Name	Туре	Attribute	Text		Status	
PAYROLL	RPGLE	SRC			ОК	
PAYROLLG	RPGLE	SRC			ОК	
Command Normal 💌 💽 Prompt						
Messages			-	Details	Show Log	
Tasks iSeries Table View iSeries Commands Log						

Figure 12: iSeries Table view for ORPGLESRC

Now you are ready to use the Remote Systems LPEX Editor to edit the member MSTDSP found

in QRPGLESRC.

7. From the Remote Systems view double-click member **MSTDSP** in the QDDSSRC source file.

Tip: You can do this in the Remote Systems view or in the iSeries Table view.

The Remote Systems LPEX Editor opens, 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 us the program hierarchy. There is explicit and rich iSeries support for verify, compile, run and debug of RPG, COBOL, C, C++, CL and DDS from the Remote Systems LPEX Editor.

Many of the editing features offered in the CODE Editor for RPG, COBOL, CL, and DDS source, such as syntax checking, automatic uppercasing, program verification, and so on, are now available in the Remote Systems LPEX Editor.

)] []] 🅸 • ★ •] 🖉 🎘 •] 🖓 🗣 💷 🕨 Δ 🕍 🛱	
) Welcome	MSTDSP.DSP	
Row 1	Column 1 Replace	
000100	.+A*.1+2+3+4+5+6+7+8 A*%/TS DD 20020613 153333 QUSER REL-V5.1 iSeries WDT	
000200	$\mathtt{A}*$ This is the DSPF use for the CODE Introduction 2 day course	
000300	A* 5763CL2 (C)Copyright 1998 A***********************************	
000500	A* THIS DISPLAY FILE PROVIDES MAINTENANCE FORMATS FOR ALL THE	
000600 000700	A* TIME REPORTING MASTER FILES - EMPLOYEE MASTER A* - PROJECT MASTER	
000800	A* - REASON CODE MASTER	
000900 001000	A*************************************	
001100	A*%%FD with errors	
001200 001300	A *%%EC λ	
001400	A DSPSIZ(24 80 *DS3) A REF(QGPL/QINVREC INVREC)	
001500 001600	A PRINT A INDARA	
001700	A CA03(03 'end of job')	
001800 001900	A CA04(04 'return to maintenance sele- A ction')	
002000	A Ction) A*	
002100 002200	A* THE SELECT FORMAT ALLOWS SELECTION OF THE TIME REPORTING A* MASTER FILE THE OPERATOR WANTS TO MAINTAIN	
002200	A* MASTER FILE THE OPERATOR WANTS TO MAINTAIN A R SELECT	
002400	A*XXTS DD 20020613 153333 QUSER REL-V5.1 iSeries WDT	
002500 002600	A* A BLINK	
002700	A 60 ALARM	
002800 002900	A 2 5'PRG01' A 2 30'Time Reporting System'	
003000	A 2 70DATE(*YY)	
003100 003200	A EDTCDE(Y) A 3 30'Maintenance Selection'	
003300	A 3 70TIME	
003400 003500	A 6 14 'Enter an X beside the application - you want to maintain'	
003600	A 9 28'Employee Master Maintenance'	
003700 003800	A PRJAPL 1A B 10 25 A 10 28'Project Master Maintenance'	
003900	A RSNAPL 1A B 11 25	
004000 004100	A 11 28'Reason Code Master Maintenance' A EMESS 50A 0 21 16	
004200	A 23 7'F3-End of Job'	
004300 004500	A EMPAPL 1 B 9 25INDTXT(01 '?') A*%%GP MAINPANEL 01	
004600	<u>A</u> *	
004700 004800	A* THE EMPSEL FORMAT ALLOWS SELECTION OF THE EMPLOYEE TO BE A* MAINTAINED AND THE TYPE OF MAINTENANCE TO BE PERFORMED	
004900	A R EMPSEL	
005000 005100	A*%%TS SD 19920629 165055 SMYTH REL-V2R2M0 5738-PW1 A*	
005200	A BLINK	
005300 005400	A 60 ALARM A 2 5'PRG01'	
005500	A 2 30 'Time Reporting System'	

Figure 13: Source Editor with DDS member

8. Double click the **MSTDSP** tab to maximize the Editor window, as shown by the arrow in Figure 13. Double click the **MSTDSP** tab again to return the window to its original size.

Opening a second source member

Next let's open a second member in the editor. Back in the Remote Systems view:

1. Double-click member **PAYROLL** in the QRPGLESRC source file.

This member will be loaded into the editor as well. Your Editor window will look something like Figure 14. Notice the two tabs in the editor pane. Click on each tab to switch from one edit session to another edit session.

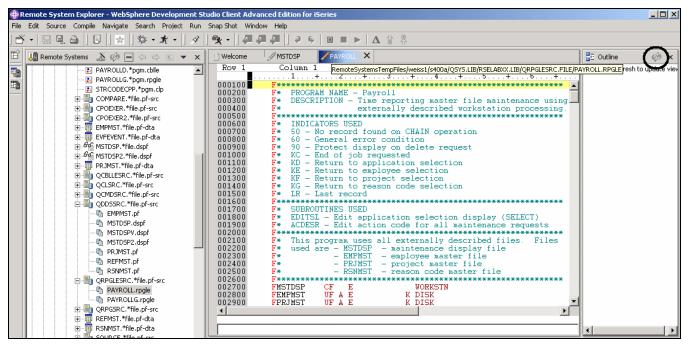


Figure 14: Editor with RPG source member and Outline view frame

Displaying an outline of a structured file

Next let's look at the Outline view. This view displays an outline of a structured file that is currently open in the editor area, and lists structural elements. The contents of the Outline view and toolbar are editor-specific.

To see an outline view of your RPG source:

1. Click Refresh on the Outline view toolbar, as shown by the circle in Figure 14

You can see the Outline view contains your source program in a tree view without the lines

containing logic.

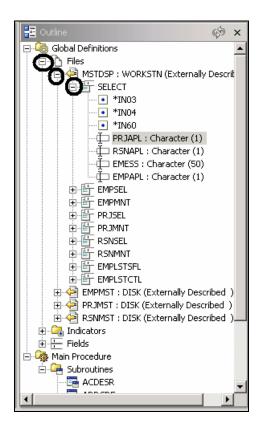


Figure 15: Outline view with file and record format expanded

Now you want to see more details of your source member.

- 2. Expand Files
- 3. Expand the **MSTDSP** workstation file
- 4. Expand the SELECT record format

Double click on any of the icons in the Outline view. This will position the source editor accordingly.

Now if you want to switch back to a different member loaded in the source editor just click on its tab and the Source Editor window with the correct member comes into focus.

	Remote System Explorer - WebSphere Development St	tudio Client Advanced Edition for iSeries
File	Edit Source Compile Navigate Search Project Run	Snap Shot Window Help
] 🖻	Š• 🔜 🖳 🕒 🛛 🗐 🕁 🛛 🅸 • 🖈 • 🗍 🖋] ♀, -] ┩┩┩ │ ३ → ■ ■ ▶ / Δ 盆 祭
Ē	🜆 Remote Systems 🍶 🔅 🖃 🗇 🗁 📧 🔻 🗙	
	PAYROLLD.*pgm.cblle	Row 1 Column 1 Replace
	PAYROLLO, pgintolie	11 + 2 + 3 + 4 + 5 + 6 +
E		000100 F ************************************
	STRCODECPP.*pgm.clp	000200 F* PROGRAM NAME - Payroll
	🕀 🗄 COMPARE.*file.pf-src	000300 F* DESCRIPTION - Time reporting master file maintenance using.
	🕀 🗐 CPOEXER.*file.pf-src	000400 F* externally described workstation processing.

Figure 16: Click the tab to give focus to the edit session

Now you want to switch back to the DDS source member.

©Copyright IBM Corporation 2003

Course material may not be reproduced in whole or in part without the prior consent of IBM

- 5. Click the **MSTDSP** tab to get the **MSTDSP** Editor window in focus.
- 6. Click the **PAYROLL** tab to get the **PAYROLL** Editor window in focus for the next exercise.

Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. When you first open Remote System Explorer, you are not connected to any system except your local workstation. To connect to a remote iSeries system, you need to:
 - A. Start Remote System Explorer communication server
 - B. Start a 5250 session
 - C. Define a connection. Specify the name or IP address of a remote system
 - D. Define a profile
- 2. The first time you connect to an iSeries system, you need to define a:
 - A. Profile
 - B. Filter
 - C. Filter pool
 - D. Connection
 - E. All of the above
- 3. Profiles:
 - A. Help partition data when you have a lot of connections or filter pools
 - B. Include all connections, filters, and filter pools
 - C. Group connections
 - D. Share connections
 - E. Keep connections private
 - F. All of the above
- 4. There are several type of profiles:
 - A. Team
 - B. Personal
 - C. Both
- 5. A team profile is used to share resources and information with other people. (T, F)
- 6. Subsystems include:
 - A. iSeries Objects
 - B. iSeries Jobs
 - C. IFS Files
 - D. iSeries Commands
 - E. All of the above
- 7. iSeries objects include:
 - A. Work with libraries
 - B. Work with objects
 - C. Work with members
 - D. Library list
 - E. All of the above
- 8. The iSeries Table view is used to:
 - A. Display a list of items, for example, members or objects in a table format similar to PDM
 - B. Perform actions against a list of items, such as editing and compiling

- C. Both
- 9. The lock icon controls the correlation between the Remote Systems view and the iSeries Table view. (T, F)
- 10. A disabled lock means when you click on an object or library in the Remote Systems view
 - A. The associated contents of that item automatically populate the iSeries Table view
 - B. Only the Remote Systems view is updated
 - C. Only the iSeries Table view is updated
 - D. None of the above
- 11. An enabled lock means when you click on an various items in the Remote Systems view
 - A. The associated contents of that item automatically populate the iSeries Table view
 - B. The iSeries Table view input does not change
 - C. Only the Remote Systems view is updated
 - D. Only the iSeries Table view is updated
 - E. None of the above
- 12. A disabled lock is the default state. (T, F)
- 13. You can maximize the Editor window by double clicking on its window heading. You can get back to its original size, by double-clicking on the heading again (T, F).
- 14. The outline view:
 - A. Displays an outline of a structured file currently open in the editor
 - B. Lists structural elements
 - C. Contents and toolbar are editor specific
 - D. All of the above

Practice

Given that you have access to your own iSeries systems, configure a new connection and connect to this iSeries system. Now rename the connection, move the connection up or change the attributes of the connection. Then use the iSeries objects subsystem to list the libraries in your library list. Use the iSeries Tables view to see the objects in your library. Subset objects in the iSeries Table view. Open the Remote Systems LPEX Editor from the iSeries Table view. Use the Development Studio Client for iSeries online help to assist you in these tasks.

What you just did

In this exercise you were introduced to profiles, subsystems, and the iSeries Tables view. You defined connection information for an iSeries server, selected the iSeries Objects subsystem and viewed the libraries in the library list. You then expanded the current library and looked at the iSeries Table view of QDDSSRC. You locked and unlocked the iSeries Table view. You then opened several source members with the Remote System LPEX Editor. You maximized the Editor window, opened an Outline view and switched from the Outline view to the member ready for edit.

In the next exercise you will continue to work with the source members you just opened and learn how to simplify your editing tasks with the Remote Systems LPEX Editor. Remember you are editing right from the Remote Systems view! And keep in mind you can use SEU like format line rules, SEU specification prompting and SEU style commands in the Remote Systems LPEX Editor.

Exercise 4: Editing ILE RPG

In this exercise you edit ILE RPG source and using some of the Remote Systems LPEX Editor's language features right from the Remote Systems view.

At the end of the exercise, you should be able to:

- Describe the features of the Remote Systems LPEX Editor
- Edit columns
- Invoke SEU commands in the prefix area
- Undo and redo edit changes
- Invoke RPG programming language specific help
- Prompt for language specific information
- Indent source
- Find and replace
- Filter source
- Find a string in multiple files with the search utility
- Compare two files with the compare utility
- Identify lines with syntax errors
- Correct syntax errors

Your program editing tasks are simplified with the Remote Systems LPEX Editor. The editor can access source files on your workstation or your iSeries system directly. When a compilation results in errors, you can jump from the compiler messages to an editor containing the source. The editor opens with the cursor positioned at the offending source statements so that you can correct them.

Here is a list of the some of the basic editor features that you would expect in a workstation editor:

- Cut, copy, and paste
- Block marking of lines, characters, or rectangles with copy, move, and delete operations
- Powerful find and replace function
- Unlimited undo and redo
- Automatic backup and recovery

In addition there are a few more functions that you may not have seen in a workstation editor:

- Token highlighting where different language constructs are highlighted using different colors and fonts to help identify them in a program
- SEU-like format-line rulers to show the purpose of each column for column-sensitive languages like RPG and DDS. These rulers can automatically update themselves to reflect the current specification.
- SEU-like specification prompting for RPG and DDS
- Sequence numbers, which allow SEU-style commands in the prefix area

- Intelligent tabbing between columns for column-sensitive languages
- Automatic uppercasing for languages that expect uppercase
- For column-sensitive languages there is a command that simplifies text insertions and deletions
- On-line language reference help

Now let's take a minute to try a few of these features.

Working with the RPG source member PAYROLL, which should be already open, try some of these features by following the instructions below. You might want to maximize the Editor window during this exercise.

Editing Columns

LPEX provides special support for insertion and deletion in column-sensitive languages. When column sensitive editing is selected, each column is considered as a separate entry space. For example, in an RPG source member, if you are inserting or deleting characters into a string that is in the Factor 2 entry, the Result field entry does not move. The default editor preference is that column sensitive editing is off. You can switch this support on by going to the workbench preferences window. The LPEX Editor has predefined settings, but also has an associated preference page containing settings that you can define. The name of the category is LPEX Editor and it appears in the tree view of the Preferences window.

To edit columns:

1. Select **Window > Preferences** from the workbench menu bar.

The Preferences window appears.

I Preferences	×
Parsers Print Save Tabs User Actions User Key Actions User Key Actions User Mouse Actions User Mouse Actions User Mouse Actions User Mouse Actions User Systems Profile Recourse Systems Priles Files	Restore Defaults Apply
Import Export	OK Cancel

Figure 17: Preferences window

In the tree view in the left pane:

- 2. Expand Remote Systems
- 3. Expand iSeries
- 4. Click LPEX Editor Parsers

The right pane allows you to set preferences for this feature.

5. Select the **Column sensitive editing** check box.

When selected, each column is considered as a separate entry space.

6. Select the **Highlight current line** check box.

This option highlights the line that the cursor is currently on. The option applies to all source files opened in the editor area.

- 7. Click **Background color**. The Color window opens.
- 8. Select light yellow from the color palette.
- 9. Click **OK**. The Preferences window shows.

©Copyright IBM Corporation 2003

Tip: Other interesting preference settings are located under LPEX Editor Parsers. When you expand this entry you see preference settings for the individual language environments.

10. Click **OK** again on the Preferences window.

Now let's see what this does.

🕀 R	emote !	Syster	n Explore	er - WebSj	phere D	evelopm	ient St	udio Clie	nt Advan	ced Edi	ition f	or iSeries	i	
File	Edit S	ource	Compile	Navigate	Search	Project	Run	Window	Help					
B	- 8		a E] 炒	• <i>\$</i> {] \$] R	-]@	6			▲ ∦ ┦	2 -	
Ē	/ РАУ	ROLL.R	PGLE X											
5	Ro	w 2 <u>7</u>		olumn (Inser								
e e	_		۴۴	llename	FŦIPEA	SERIE	5+LKI	en+All.	evice+	. Keyw	ords	*+++++	++++++++	+++++++++++++++++++++++++++++++++++++++
	002	700	FM9	STDSP	CF	Е		Į,	ORKSTN					
	002	800	FEN	(IPMST	UF A	Е		K I	DISK					
	002	900	FPF	RJMST	UF A	Е		KI	ISK					
	003	ñññ	FRS	SNMST	UF A	E		K I	ISK					
	003		D*			-								
	003		D*	Compi	le tim	e arra	av oo	ntaini	ng erro	m de	sori	ntions		
	003	300	D B			S	.,		50				PERRCD(1)

Figure 18: Editor in Insert mode with highlighted line 27

In the Editor window:

- 11. Move the cursor to line 27, column 10.
- 12. Make sure the editor is in **Insert** mode.

If the status area says 'Replace':

- 13. Press the Insert key.
- 14. Press the spacebar 3 times.

Notice that only the filename is shifted but none of the other columns to the right are affected.

🕀 R	emote Syste	m Explorer - Web9	phere D	evelopn	nent Sl	udio Clie	nt Advanced Edition for iSeries
File	Edit Source	Compile Navigate	Search	Project	Run	Window	Help
Ď	• 8 9	≙ 0 ☆	• <i>\$</i> { •] \$] R	•] @	⊈ I ■ ► Δ ₩ ₽
Ē	/*PAYROLL	RPGLE X					
5	Row 27	Column		Inser	-		
- <u>-</u>		FFileneme	++IPEA	SFRle	n+LK]	en+AID	evice+.Keywords++++++++++++++++++++++++++++++++++++
	002700	FMS DE	P CF	Е		W	ORKSTN
	002800	FEMPMOT	UF A	Ε		K D	ISK
	002900	FPRJMST	UF A	ιE		K D	ISK
	003000	FRSNMST	UF A	E		K D	ISK

Figure 19: Inserted blanks in file name, none of the other columns has been shifted

15. Press the backspace 3 times.

Once again the filename is affected but no other columns are touched.

Entering SEU Commands

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 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.

If you are an SEU expert you will appreciate the ability to use SEU commands:

- 1. Move the cursor into the gray sequence number area to the left of the edit area.
- 2. On any sequence number type dd
- 3. Go down a few lines and type ${\tt dd}\;$ again and press Enter.

Notice that the lines have been deleted.

- 4. Now type 15 in the sequence number area.
- 5. Make sure the cursor is within the sequence number area.
- 6. Press Enter.

Five new lines are inserted.

Requesting Undo and Redo operations

The editor records each set of changes you make to a file in the Editor window. The number of changes made since the last file save is displayed on the status line. If you want to undo a set of changes made to a file you use the Undo operation. You can also cancel the effects of an Undo operation by using the Redo operation.

Now you are going to undo some of the changes you just made to the file. Then you will cancel the changes by using the Redo operation. Finally you will reload the source so that it is back to its original form.

To undo and redo edit changes:

1. Select Edit > Undo from the workbench menu bar.

Notice that the 5 new lines disappear.

2. Do another undo action by pressing Ctrl+Z.

Notice that the deleted lines reappear.

3. Select **Edit > Redo** from the workbench menu bar

Notice that the lines are deleted again. At this point you will reload the source from the iSeries to make sure that it is back in its original form. To do this:

4. Select File > Close option on the workbench menu bar

A Save Resources window will appear asking you to save the latest changes.

🕀 Save Resource			×
PAYROLL.RPGLE has been	modified. Save chang	jes?	
	Yes		Cancel

Figure 20: Confirmation window

- 5. Click No.
- Go back to the workbench to the Remote System Explorer perspective and reload the PAYROLL member in the QRPGLESRC file.

Invoking Language-Sensitive Help

Inside the editor, there is cursor-sensitive language-reference help available. This help is invaluable if you cannot remember the order of fields in an RPG specification or the possible values for a variable field. This help is available from the LPEX Editor window.

To receive language sensitive help, press F1 in an Edit window. If the cursor is on an operation code, you receive help for that operation code; otherwise, you receive help for the current specification.

To access language sensitive help:

- 1. Position the cursor over the word MOVE in line **112** of the ILE RPG source.
- 2. Press F1.

Language-sensitive help for the MOVE operation code appears in a Help window.

🕀 Help - WebSpt	nere Developme	ent Studio Client Ad	vanced Edition fo	r iSeries					_ 🗆 ×
Search:	G	Advanced Search							_
Contents		6	Development St	udio Clier	nt for iSeries			(⇒ ⇒ [
	DUMP ((Program Dump)	Wohs	nha	(D)	Develop	nont St	udio I	T F
	ELSEIF		webs	pne	re(K)	Developi	nent st	uulo I	
		(End a Structured	RPG I	Dafa	nono	`			
	ENDSR	(End of Subroutine		vere	rence	5			
	🖹 EVAL (E	Evaluate expression							
	EVALR	(Evaluate expressi							
	EXCEPT	T (Calculation Time							
	EXFMT	(Write/Then Read	MOME	A	2				
	🖹 EXSR (I	Invoke Subroutine)	MOVE	<u>10100</u>	ej				
	EXTRCI	T (Extract Date/Tin							
	🖹 FEOD (I	Force End of Data)							
	FOR (F	ior)			6	1 .1 THEATTO		1 1 2	<u> </u>
	FORCE	(Force a Certain F	Free-Form S	Syntax		ed - use the EVALR			
	🖻 бото ((Go To)			such as %	DATE, %TIME, %T	IMESTAMP, %	CHAR, %U	CS2,
	🖹 IF (If)				or %GRA	PH)			
	🗎 IFxx (If	f)	1			· ·			
	🖹 IN (Ret	trieve a Data Area)							
	🗎 ITER (I	iterate)					Result		
	🖹 KFLD (D	Define Parts of a Ke	Code	Fa	ctor 1	Factor 2	Field	Indicate	ors
		Define a Composite	MOTE OF	T		a	m		
		(Leave a Do/For Gr	MOVE (P)	Data At	tributes	Source field	Target field	+ -	ZB
		5R (Leave a Subrou							
		IP (Look Up a Table	The MOVE of	neration	transfers ch	aracters from factor 2	to the result fiel	d Moving st	tarts
) (Move High to Hig	with the rightn						
		(Move High to Low	wiai ale righai	1051 01141	acter of fac	101 2.			
		(Move Low to High		-	.				
		(Move Low to Low				mestamp data , facto	or 1 must be blan	ik unless eith	er the
		OR (Begin a Monite	source or the t	target is a	a character	or numeric field.			
	MOVE (
		A (Move Array)	Otherwise, fac	ctor 1 co	ntains the d	ate or time format cor	npatible with the	character or	
		(Move Left)				arget of the operation			
	MULT (I					Time Data Type, and			into mat
		Nove Remainder)	can be used si	ce <u>Date 1</u>	иаа туре,	The Data Type, and	Timestamp Data	а туре.	
	NEXT (P								
		R (Set/Get Occurrer	If the source of	or target i	is a charact	er field, you may optic	onally indicate the	e separator fo	ollowing
		ROR (On Error)	the format in f	actor 1.	Only separa	itors that are valid for	that format are a	allowed.	
	OPEN (Open File for Proce							
		Or) (Otherwise Select)	If factor 2 is *	DATE ~	r IIIDATE :	and the result is a Date	e field factor 1 is	a not require	4 TF
		Vrite a Data Area)				nust be compatible wi			
		(Identify Parameter						DATE OF U	DAIL
4	E PARM (Tuendry Parametel	as specified by	y the DA	TEDIT key	word on the control s	pecífication.		
									-
	**	100	4						Þ

Figure 21: Help for MOVE and synchronize button

3. Click the Synchronize Navigation icon on the Help window toolbar

This will synchronize the topics in the left overview pane to the help topic you are viewing in the main help pane.

- 4. Play around in the Help window to see what else is available.
- 5. Minimize the Help window.
- 6. Select **Help > Help contents** on the workbench menu bar.

🏶 Help - WebSphere Development Studio Cli	ent Advanced Edition for iSeries
Search: 60 Advanc	ed Search
Contents 🗎	
Development <u>Studio</u> <u>Client</u> for iSeries Series programming information Extending the <u>Remote</u> System Explorer	eclipse
🔋 WebSphere Application Server Network	Welcome to the Eclipse help system
 J2SE SDK 1.3.1 APIs Application developer information Tool developer information 	Click on one of the top-level sections in the Contents frame on the left to browse that section's contents. Each section loads a different navigation tree. To return to the main Table of Contents view, click the Home button (), above the Contents frame.
J2EE SDK 1.3.1 APIs	Searching

Figure 22: Main Help window

7. Select iSeries programming information from the topic list in the left pane

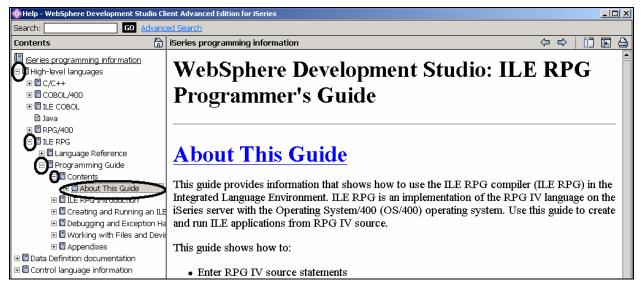


Figure 23: Finding the RPG IV manuals

- 8. Expand High level languages
- 9. Expand ILE RPG
- 10. Expand Programming Guide

Here you find the table of contents and can select your topic of interest. Having the latest version of the manuals at your fingertips makes it easier to find programming information.

Prompting Language Specifications

Instead of entering or changing code directly in the Editor window, you can use prompts. When you request a prompt for a specification line, a window appears where you can enter or change that line using entry fields.

To access prompts:

- 1. Return to the workbench. In the Editor window move your cursor to the D-spec on line 33.
- 2. Select **Source** from the workbench menu bar.
- 3. Select Prompt (or press Shift+F4).

The Editor window will reduce in size and allow you to see the iSeries Source Prompter window at the bottom of the workbench. The iSeries Source Prompter window shows the specification line broken down into its individual fields.

Studio Client Adv	vanced Edition for iSeries		>
n Snap Shot Wir	idow Help		
' 🛛 🗣 🗸 🗸	47 47 ■ ■ ▶ ▲ ▲ ▲		
) Welcome	PAYROLL X	🗄 Outline	⊚ x
Row 33	Column 9 Replace	📮 🧠 Global Definitio	ins
000100	DName++++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++++++++++++	🕀 🖸 Files	
000200	F* PROGRAM NAME - Payroll	😟 🞑 Indicators	
000300 000400	F* DESCRIPTION - Time reporting master file maintenance using F* externally described workstation processing.	English Main Procedure	e
000500	F***************	E Gabroutine	
000600	F* INDICATORS USED F* 50 - No record found on CHAIN operation		
000800	F* 60 - General error condition		
000900	F* 90 - Protect display on delete request F* KC - End of job requested		
001100	F* KC - End of job requested F* KD - Return to application selection		
001200	F* KE - Return to employee selection		
001300 001400	F* KF - Return to project selection F* KG - Return to reason code selection		
001500	F* LR - Last record		
001600 001700	F*************************************		
001800	F* EDITSL - Edit application selection display (SELECT)		
001900 002000	F* ACDESR - Edit action code for all maintenance requests F***********************************		
002100	F* This program uses all externally described files. Files		
002200 002300	F* used are - MSTDSP - maintenance display file F* - EMPMST - employee master file		
002400	F* - PRJMST - project master file		
002500 002600	F* - RSNMST - reason code master file F************************************		
002700	FMSTDSP CF E WORKSTN		
002800 002900	FEMPMST UF A E K DISK FPRJMST UF A E K DISK		
003000	FRSNMST UF A E K DISK		
003100 003200	D* D* Compile time array containing error descriptions.		
003300	D ERR S 50 DIM(10) CTDATA PERRCD(1)		
003400 003500	D EMESS S 50 E*		
•			
<u> </u>			
🖌 iSeries Source	Prompter	- \$	∢ ⊴¢ ف
Line type: D: Def	finition	F	
Line type. Jeaneral			
	Externally Data structure Definition Internal Decimal		Ē
Name	described type type From To/Length data type positions Keywords	TTL DEDCD (1)	
ERR	▼ S ▼ 50 ▼ DIM(10) CTD	ATA PERRCD(1)	
	++++++++ETDsFrom+++To/L+++IDc.Keywords++++++++++++++++++++++++++++++++++++	nts++++++++++	+
DERR	S 50 DIM(10) CTDATA PERRCD(1)		
•			

Figure 24: Prompt window

To display context sensitive help for any field in the iSeries Source Prompter window:

©Copyright IBM Corporation 2003 Course material may not be reproduced in whole or in part without the prior consent of IBM

- 1. Move the cursor to the **Keywords** field using the Tab key.
- 2. Press F1 to see help for this field.

The Help window with help for the D spec keywords will appear. If it doesn't appear automatically, you might have to bring it to the foreground by clicking on its icon on the Windows task bar.

You will see words in the help that appear in a different color than the regular text. These are help links, and they show that there is additional help available on that word or phrase.

- 3. Click on any **link** to see specific help for that item.
- 4. Minimize the Help window.

To get to the iSeries Source Prompter window from the maximized Editor window without bringing the size of the Editor window down, position the iSeries Source Prompter window on the left task bar as a Fast view.

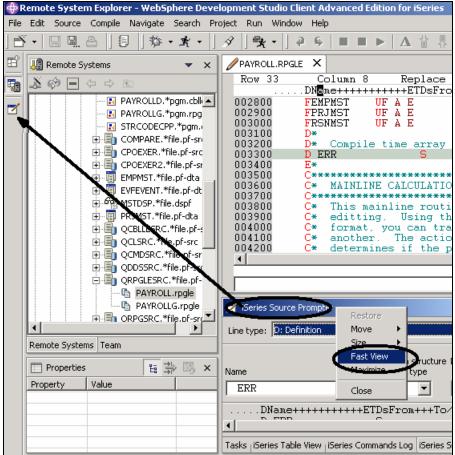


Figure 25: iSeries Source Prompter window moved to fast view on taskbar

- 5. Right-click on the iSeries Source Prompter window heading
- 6. Select Fast view from the pop-up menu

You will see the iSeries Source Prompter window icon on the task bar on the left hand side of the workbench

- 7. Maximize the Editor window
- 8. Select a line in the Editor window and press Shift+F4 to prompt for a line

The iSeries Source Prompter window will be placed on top of the Editor window

		ystem Ex							d Editio	n for i9	ieries									
		urce Com		-																
	•	e 4		於·1	ξ - ∐.	<i>\$</i>]	°}: - ∫	96		Δ										
Ē	🗹 iSerie	es Source P	rompter														-	\$ \$2)* _ ×	
	Line type	e: D: Defi	nition																	
0	Name ERR			Externall describe	d ty	tructure pe	Definition type		To/Leng	ith 50		l Decim e positio	ns Keywo		CTDAT	A PERR	CD(1)		<u> </u>	
		DName+- D ERR	++++++	++++ETD S		+++To	/L+++I 50				++++++ RRCD(1		+++++	+++Co	mments	+++++	+++++	+		
	 																			
	•														1				- •	
-									 											

Figure 26: Prompt window on top of Editor window

9. Click the **iSeries Source Prompter** icon on the task bar to minimize the Prompt window

Note: On the iSeries Source Prompter window toolbar you can use the three push buttons to:

Disable source prompt view, Disable syntax checking, and Change to insert mode

Indenting Source

When editing ILE RPG source, it can be difficult to determine the beginning and ending of constructs. The indent option allows you to view your source with constructs in an indented mode. By default, the indent option will split the screen horizontally and show the indented view in the

bottom pane.

To indent source:

1. Select **Source > Show Indentation** from the workbench menu bar

The indented view is displayed at the bottom of the workbench.

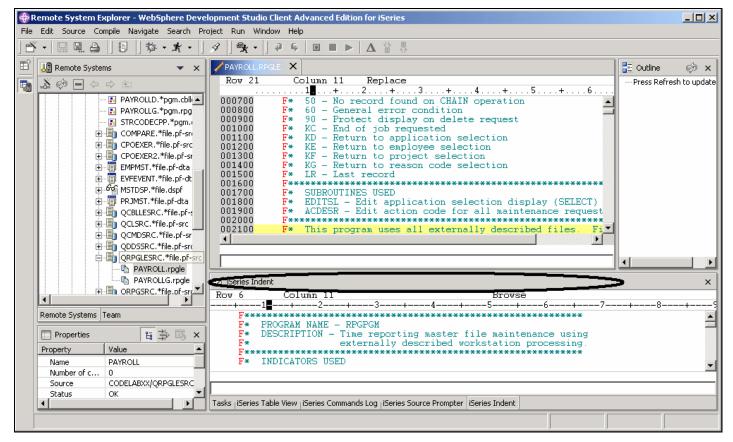


Figure 27: Indent view at bottom of workbench

You can display the indented view as a full view.

2. Double-click iSeries Indent window title bar of the indented view.

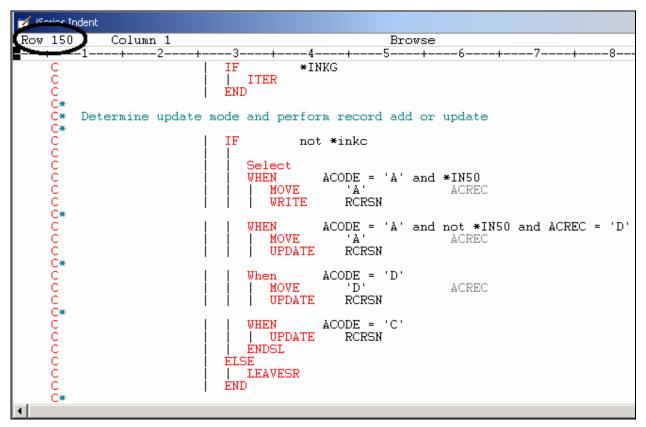


Figure 28: Maximized Indent view

3. Scroll down to line 150

In this area you see some nested conditions with indented lines, as you will notice this helps to recognize the beginning and ending of these conditions.

Tip: You can move this view as a Fast view to the left task bar as well, if you plan to use this view frequently.

Note: The INDENTED view is Browse mode only and cannot be edited.

4. Switch back to the Editor window with the **PAYROLL** program.

Hint: Double-click the iSeries Indent view title.

Finding and Replacing text

The LPEX Editor also has a powerful find and replace text feature. You use the Find and Replace window to search for an item. You can search for a word, a partial word, or a sequence of such. You can also enter a pattern you want to match, provided that the pattern follows the rules of regular expression. You can replace the found search item. If the entered text or pattern is found, the cursor moves to either the next or previous occurrence of the search item, according to your chosen search

direction, and replaces the found text according to your selections.

To find and replace text in the Payroll Editor window:

- 1. Press **Ctrl+Home** to go to the top of the file.
- 2. Select Edit > Find/Replace from the workbench menu bar or press Ctrl+F.

The Find /Replace window appears at the bottom of the Editor window.

🕀 Re	emote System Exp	plorer - WebSphere Development Studio Client Advanced Edition for iSeries	
File	Edit Source Com	ipile Navigate Search Project Run Snap-Shot Window Help	
B	- 🗌 🖳 🖨] 目] ☆ ☆ - ★ -] ダ st -] 砷 택 택] ֎ ֍ 回 ■ ▶ Δ 🖀 🛱	
e	📄 Welcome 🛛	PAYROLL X	
	Row 52	Column 26 Replace	
		.CLON01Factor1++++++#@pcode(E)+Factor2++++++Result++++++Len++D+HiLoEqComment	ts++++
	003600	C*************************************	
	003700 003800	C* MAINLINE CALCULATIONS	
	003900	C* This mainline routine controls the display file processing and	
	004000	C* editting. Using the function keys described on each display	
	004100	C* format, you can transfer from one maintenance application to	
	004200	C* another. The action code you select on the selection formats	
	004300	C* determines if the program will add a new record to the file or	
	004400 004500	C* update an existing record in the file.	
	004600	C* Housekeeping, clear display fields and reset indicators.	
		C*	
	004800	C EXSR MAIN	
	004900	C* If MAIN is done program ends	
	005000 005100	C eval *INLR = *on * MAIN SUBROUTINE	
	005100	C MAIN BEGSR	
	005300	C dou *inkc	
	005400	\overline{C} EVAL *IN60 = *OFF	
	005500	C EVAL *IN60 = *OFF C EVAL EMESS = *BLANK C EVAL EMPAPL = *BLANK	
	005600	C EVAL EMPAPL = *BLANK	
	005700 005800	C EVAL PRJAPL = *BLANK C EVAL RSNAPL = *BLANK	
	4	C LVAL ROMARL - *DLANK	
	<u>•</u>		
	Ein	<u>Next</u>	us <u>A</u> ll
	Replace	Keplace Reg	place all 📗
	📃 🗖 Case sensitiv	ve 🗖 Regular expression 🔽 👾 rap 🔲 Select found text 🔲 Res <u>t</u> rict search to selection	
	🗌 🗌 Restrict sear	rch to columns Start column 1 End column 80	
ļ			

Figure 29: Find/Replace window

At the bottom of this window, you will notice that you have some options to select from, for example, search only in certain columns, etc. You want to find the first occurrence of BEGSR.

3. In the Find field, enter BEGSR to find the start of a subroutine. Make sure the Replace field is

blank. You would use this field for text replacement.

You will notice that the Editor moves the active line, to line 52, which contains the first BEGSR phrase in the file.

4. Click Next (or Press Ctrl+F) to go to the next location of BEGSR in the file.

Filtering Lines by string

The LPEX Editor allows you to filter or subset your source so that you see only lines containing a given string. Filtering lines makes it quick and easy to find lines without having to scroll through your source.



Figure 30: Selected operation code BEGSR

To filter source by string:

- 1. Select the **BEGSR** operation code in the Edit window
- 2. Right-click in the Editor window and select Filter Selection from the pop-up menu

🕀 R	emo	te Syste	m Explore	er - WebS	phere D	evelop	ment S	tudio (Elient	Advanc	ed Editio:	n for is	5eries	
File	Edit	Source	Compile	Navigate	Search	Projec	t Run	Snap	Shot	Window	Help			
] 🖻	-		a E	J ↓	珍 •	\$ <u>`</u> -] \$	1	•]	a a	🚚 🛛 4	і Б	0	
Ē	١Ľ	Welcome	/P/	AYROLL	<									
5]	Row 52		Column		Repl				_				
			!	CLONO1E	actor	1++++	. +++<u>©</u>p	code	(E)+	Facto:	r2+++++	++Re	sult	++
\mathbf{Z}	∣⊎						_							
	⊕	00520	0 (C 1	IAIN 👘		EE	GSR						
	⊕	01100	0 0	C F	RSNTAG		BE	GSR						
	⊕	01830	0 0	С В	EMPTAG		BE	GSR						
	I Ā	02640	0 0	C H	PRJTAG		be	qsr						
	I Ā	03470	0 0	C H	DITSL		BE	ĞSR						
	I Ā	04040	0 0	C 2	CDESR		BE	GSR						
	ΙĂ	04310	0	C 1	DDCDE		BE	GSR						
	ΙĂ	04440	0	0 0	HGCDE		BE	GSR						
	ŦŒŒŒŒŒŒŒ	04610	0	BEGSE	DELC	DE;								
	₩	04760	0	BEGSE										

Figure 31: Editor window showing all lines with BEGSR phrase

- 3. Move the cursor down a couple of lines, to line 347
- 4. Click the plus beside the line to expand this section

🕀 R	emote System Ex	plorer - V	YebSphere Devel	opment Studio Cli	ent Advanced Edi	tion for iSeries	
File	Edit Source Con	npile Nav	igate Search Pro	iject Run Snap-Sh	ot Window Help		
] 🖄	- 🗌 🗒 🖨] 🛛]	☆ 🛛 🏷 • 🕺	•] 🖉] 💱 •		4 6 0 ■ ▶	
Ē	📄 Welcome	/ PAYRC	u X				
	Row 347	Colu	umn 1 Rep	blace			
5		CL0	N01Factor1++	+++++Opcode(E)+Factor2+++	++++Result+++++	+++Len++D+H
\mathbf{Z}	8						
	005200 005200 011000 018300 018300	ç	MAIN	BEGSR			
	윤 011000 윤 018300		RSNTAG EMPTAG	BEGSR BEGSR			
	026400	2	PRJTAG				
	028400	č	EDITSL	begsr BEGSR			
	034800	C*	EDITOL	DEGOR			
	034900	Č*	Housekeepin	g. clear dist	lav fields a	and reset indica	tors.
	035000	Č*		,			
	035100	С		MOVE	*BLANKS	EMESS	
	035200	С		EVAL	*IN60 = *0)FF	
	035300	C*					
	035400	C*				checks the appl	
	035500	C*		ields to ensu	re that only	v one applicatio	n has been
	035600	C* C*	selected.				
	035700 035800	C	EMPAPL	IFEO	'X'		
	035900	č	PRJAPL	ANDEO	'X'		
	036000	č	EMPAPL	OREQ	'X'		
	036100	C C	RSNAPL	ANDEO	'X'		
	036200	Č*	THE BUG I				
	036300	C*	*******	*****			
	036400	С		MOVE	'1'	*IN60	
	036500	С		MOVE	ERR(2)	EMESS	
	036600	00000		ELSE			
	036700	ç		MOVE	'0'	*IN60	
	036800	C C*		END			
	036900	U*					
	4						

Figure 32: Section expanded

Now you want to show the entire source again.

1. Right-click in the Editor window and select **Show all** from the pop-up menu or press **Ctrl+W**.

Your cursor is still positioned on the same line that you moved the cursor to, even though all lines are now showing.

Filter lines based on type

To help you navigate quickly through your ILE RPG source the Editor allows you to filter lines based on the line type. Imagine you want to see where all the subroutines are defined in your source:

• 🛛 🖳 🖉	10 10 1	弥・光・ 夕 雪	 □●●● □●● □ □
U Welcome	PAYROLL >		
Row 47	Column		+4+5+6+
004700	C* T		······································
004800	C	Prompt	MAIN
004900	C* If MA	Syntax Check Line	*INLR = *on
005100	* NAIN	Select Format Line	*INLR = *on
005200	C NA	Gut	
005300	C I	Copy	*inkc
005400	S I	Paste	*IN60 = *OFF EMESS = *BLANK
005600	000000		ENPAPL = *BLANK
005700	č	Find selection	PRJAPL = *BLANK
005800	C I	Select line	RSNAPL - *BLANK
005900	C* Urit	Select character	display. If end of job request
006100		Select rectangle	b display. If end of job request
006200	C.	Deselect	
006300	ç I	Copy selected text	not *IN60 SELECT
006400	č ×I	Overlay with selected text.	'1'
006600	č	Move selected text	
006700	ç	Delete selected text	
006800	ç		EDITSL
007000	C* C* C C *I C C C C C C C C C C	Effer colocilico	
007100	C* IF c	Filter view	By Date on (equal to 1), t
007200	C* prog	EXCLOSE PERCODER	Code
007300	C.	Show all	Comments
007400	Ŭ	5ave	Control
		5515	Procedures

Figure 33: Editor pop-up menu with Filter sub menu

- 1. Right click in the **Editor window**
- 2. Select Filter view > Subroutines.

All subroutines specifications are displayed allowing you to move quickly and easily to the area in your file where the desired subroutine is.

⊕ R	emote System	Explorer - W	ebSphere Deve	lopment Studio Clier	nt Advanced E
File	Edit Source (Compile Navig	ate Search Pr	oject Run Snap-Shol	: Window He
	- 8 8 4) [] [] ·	≿ 珍 • ★	•] \$] ? •	
Ē	🗋 Welcome	- PAYROLI			
5	Row 52	Colum		place +++++Opcode(E)	+Factor2+
7	⊕ ⊕ 005200 舟 010200	– C	MAIN	BEGSR endsr	
	H 011000 H 017600 H 018300	C C C	RSNTAG EMPTAG	BEGSR ENDsr BEGSR	
	H 016300 H 025700 H 026400	000	PRJTAG	ENDSR begsr	
	# 005200 C # 010200 C # 011000 C # 017600 C # 017600 C # 017600 C # 018300 C # 025700 C # 026400 C # 033500 C # 034700 C # 039800 C # 042700 C # 042700 C # 0440400 C # 044000 C # 044000 C # 044000 C # 047200 C # 047600 C # 047600 C # 047600 C		EDITSL	endsr BEGSR ENDSR	
	H 040400 H 042700 H 043100	000	ACDESR ADDCDE	BEGSR ENDSR BEGSR	
	H 044000 H 044400	C C C	CHGCDE	ENDSR BEGSR	
	H 045400 H 046100 H 047200	BE	GSR DELCDE; DSR;		
	⊕ 047600 ⊕ 047900		GSR INVCDE; DSR;		

Figure 34: Subroutine filter in action

- 3. Move your cursor to the line with the subroutine declaration CHGCDE. (line 444)
- 4. Click the plus sign beside the declaration to show all lines in this subroutine

Now you can work with the source inside this subroutine.

Searching multiple files

If you would like to search through the members in a source physical file or through the files in a local directory, you can use the Search tool. The Multi-File Search utility allows you to search for a particular string of text in many members on the host. This function can also be used on local files.

To search multiple files:

1. Select **Search** from the workbench menu bar.



The Search window appears.

2. In the Search string field type ENHRS

Search 🔀
🖻 iSeries Search 📴 File Search 😔 Help Search 💀 Java Search
Search string ENHRS
Connection: s400a
Target
Library RSELABxx Browse
File: QDDSSRC Browse
Member: *ALL Browse
Columns
All columns
C Between 1 and 80
C Between 1 and end of line
Customize Search Cancel

Figure 35: File Search window

The Connection entry field should contain your iSeries server name, otherwise enter it there.

In the Target area:

- 2. Enter RSELABXX in the Library field, where xx is your workstation number
- 3. Enter QDDSSRC in the File field, to search all members in this source physical file.
- 4. Enter *All in the member field
- 5. Click Search.

The Multi-File Search window lists all the lines in all the files that reference ENHRS.

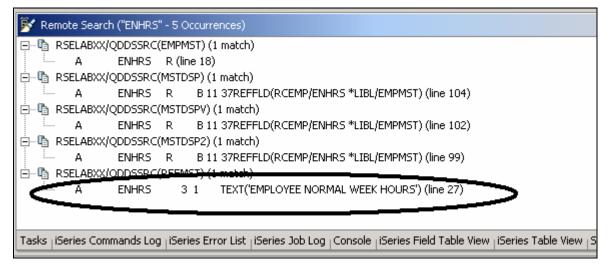


Figure 36: Search result

6. Double-click the last line in the list

🗋 Welcome	PAYROLL	REFMST	×		
Row 27	Column				
	A	Name++	++++RLen++7	DpB	Functions+
000013	A				COLHDG('DA
000014			RELATED FI		
000015	A	EMPNO	6	0	TEXT('EMPI
000016	A	ENAME	20		COLHDG('EN
000017	A A	ENAME	30		TEXT('ÉMPI COLHDG('EN
000019	Å	EMCAT	1		TEXT('EMPI
000020	Ä	LIIONI	-		COLHDG('EN
000021	A	EDEPT	5		TEXT('ÈMPI
000022	A				COLHDG('EN
000023	A	ELOCN	30		TEXT('ÉMPI
000024	A				COLHDG('EN
000025	A	EUSRI	8		TEXT('EMPI
000026	À	ENHRS	3	1	TEXT('EN
000028	A	EMHKS	J	1	COLHDG('NO
000029	Ä	EPHRC	5	1	TEXT('PROJ
000030	A				COLHDG('PF
000031	A	EPHRY	7	1	TEXT('PROJ
000032	A		_		COLHDG('PF
000033	A	EPHRP	7	1	TEXT('PROJ
000034	A A	EPNRC	5	1	COLHDG('PF TEXT('NON
000036	Å	EFARC	5	1	COLHDG('NO
000037	Å	EPNRY	7	1	TEXT ('NON
000038	A			_	COLHDG('NO
000039	A	EPNRP	7	1	TEXT('NON
000040	A		_		COLHDG('NC
000041	A	EHWRK	5	1	TEXT('EMPI
Remote Sear	ch ("ENHRS" - 5 C	ccurrences)			
	(QDDSSRC(EMPM				
	,	ne 18) (1			
	<pre></pre>				
A	ENHRS R		RCEMP/ENHRS *L	IBL/EMPMS	T) (line 104)
⊑… 崎 RSELABX>	<td>SPV) (1 match)</td> <td></td> <td></td> <td></td>	SPV) (1 match)			
A	ENHRS R	B 11 37REFFLD	RCEMP/ENHRS *L	IBL/EMPMS	T) (line 102)
🖹 🖓 🐚 RSELABX	(/QDDSSRC(MSTD	SP2) (1 match)			
Α	ENHRS R	B 11 37REFFLD	RCEMP/ENHRS *L	IBL/EMPMS	T) (line 99)
🗄 🖷 RSELABX>	(/QDDSSRC(REFM	IST) (1 match)			
Α			OYEE NORMAL W	EEK HOURS	i') (line 27).
					, , ,

The member REFMST is automatically loaded into the editor and the cursor is placed on the correct line. Wow !

Comparing file differences from the Remote Systems view

If your product undergoes many changes, you will find the Compare utility useful. It allows you to compare different versions of a program and find the differences. There are 2 ways in doing the compare: use the Compare utility in the workbench or use the Compare utility in the CODE tool. The compare in the CODE tool is more intuitive but requires you to start the CODE Editor outside of the workbench.

Using the compare utility in the workbench you can view the differences between two files by comparing them. You can compare different files, and you can compare versions in the Workbench with versions in the repository or with the local edit history. In some cases you can compare three files (when a common ancestor exists).

After a comparison is carried out, the Compare Editor opens in the editor area. In the compare Editor, you can browse through all the differences and copy highlighted differences between the compared resources. You can save changes to resources that are made in the comparison editor.

Using the compare utility in CODE you can also view the differences between two files by comparing them. You enter a name of a file to compare against the file in the CODE Editor view. You can type the name of a file, or you can select one from the list of files already open in the editor. If you type the name of the file that is not already open in the editor, it is loaded into the editor. If no file is specified, the current file is compared against a new, untitled file. The current file appears on the left side of the Compare view, and the specified file on the right. You use the Compare menu to view the next and previous mismatch and to select options such as ignore case, font, protect view and show mismatches only.

To compare files in the workbench:

- 1. Select Window > Preferences from the workbench menu bar
- 2. Expand LPEX Editor from the tree view in the Preferences window
- 3. Select Compare under LPEX Editor

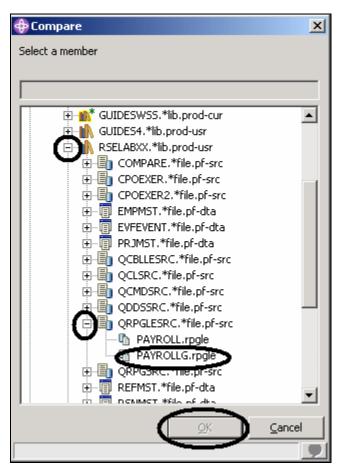
Preferences]
Agent Controller Build Order Component Test Data Debug External Tools Help Install/Update Internet	Compare Ignore case Ignore leading blanks Ignore trailing blanks Ignore all blanks Ignore comments		
- J2EE - Java - Logging - LPEX Editor - Appearance - Block - Compare - Comp			
Find Text Parser Associations Parsers Print Save Tabs			
User Actions User Commands User Key Actions User Key Actions User Profile C:\Documents and Sett ▼	1	<u>R</u> eset Restore Defaults	Apply

- 4. In the Compare window make sure that the Ignore blanks check boxes are selected
- 5. Click OK in the Preferences window

Back in the Edit window of the Payroll member:

	ate System Explo								on for iSe	ries		
Edi	it Source Compile	Navigate	Search	Project	Run	Snap Sh	iot '	Window Help				
<u> </u>	Undo	Ctrl+Z	Ð	1	ß	<u></u>	•		96	0		Δ
6	Redo	Ctrl+Y	Ś	•	⇔	È 🔻	х	📄 Welcome	/PA	YROLL	×	
A	" Cut	Ctrl+X	f-src					Row 1	Co	olumn .1		Rep. 2
Re	і Сору	Ctrl+C	-src					000100			. T *****	- · · ·
			pf-src					000200	F*		RAM NA	
_Ē	Paste	Ctrl+V	dta					000300	Ē*		RIPTIO	
-			- f-dta					000400	F*			
8	Delete	Delete	F					000500	F**	*****	*****	****
	Select All	Ctrl+A	fta					000600	F*	INDI	CATORS	USE
	Delect All	CUITA		_				000700	F*		No re	
	Find/Replace	Ctrl+F	P	E				000800	F*	60 -	Gener	
	Find Next	Shift+F4	src					000900	F*	90 -	Prote	
			f-src					001000	F*	KC -	End o	
	Find Previous	Ctrl+U	-src					001100	F*	KD -	Retur	
	Find Other		.pf-sr	c				001200	F* F*	KE –	Retur	
	Filter Selection		le					001300	г* F*	KF -	Retur Retur	
	Exclude Selection							001500	r* F*	TP _	-	n uu recc
			gle					001600	F**	*****	Fast .	****
	Show All	⊂trl+W	f-src					001700	F*		OUTINE	S IIS
	Add Bookmark		dta					001800	Ē*		5L – E	
			dta					001900	F*	ACDES	5R – E	dit
	Add Task		src					002000	F**	****	*****	****
	Colora.							002100	F*	This	progr	am u
	Select							002200	F*	used	are -	MST
	Selected		•					002300	F*		-	EMF
	Deselect	Alt+U						002400	F*		-	PRJ
								002500	E*		-	RSN
_	mark		معدا ب	_		_	~	002600	-		*****	
	Compare		MA C	iompare.		_		02700		TDSP	CF	, E
	Novetroke Recorde			<u> </u>						PMST		A E
				ext Mism	iatch	<u>Chrl+S</u>	iit+i	02900		JMST NMST		A E A E
	Content Assist	Ctrl+SPACE	P	revious N	/lismato	hCtrl+S	hift+P	03100	PRO D*	NEST	OF	A E
		THIR PATROLLE	D	efresh C	ompari	e Ctrl+S	hift+P		D.*			
		PAYROLLO	a.r	lear	a. np. ar i							
		🐚 PAYROLLO	יייוג	liear								
		IB DO IMCT -	6				_ [

6. Select Edit > Compare > Compare from the workbench menu bar



In the Compare window:

- 7. Expand your connection
- 8. Expand *LIBL
- 9. Expand RSELABxx
- 10. Expand QRPGLESRC
- 11. Select member PAYROLLG
- 12. Click **OK** on the Compare window

The editor now will show the differences of these two members PAYROLL and PAYROLLG You can move from mismatch to mismatch by going back to the Compare menu option under the Edit menu option or by using short cut keys CTRL+Shift+N.

The color of PAYROLL and PAYROLLG will be different so you have an indication where lines are different from each other.

©Copyright IBM Corporation 2003 part

a is	eries Projects	- WebSobere Deve	lonment Stud	io Client A	vanced Edition for iSe	ries
-	-	-	-		p Shot Window Help	
			· · · ·	1.5	· · ·	!:
] 🖻	- 🛛 🖾 (≜ ¶§16¥ M] 🖯 🛛 🖈	珍・:	१ •] ४] ९ २ •] 4 4 4 1 4 6 0 1
Ē	📄 Welcome	PAYROLL X				
TE-	Row 52	Column 1	Repla	ce		
					+ 4 +	.5+6+7
	004800	С		EXSR	MAIN	
	004900	⊂ ⊂ ⊂ ⊂ ⊂ Č* If MAIN	is done p			
	005000	С	-	eval	*INLR = *on	
	005100	* MAIN SU	BROUTINE			
	005200	<u>C</u> HAT		BEGSR		
		C		dou	*INKC	>
	005300	C C		dou	*inkc	
	005400	с с с с		EVAL	*1N6U = *OFF	
	005500	Ç		EVAL	EMESS = *BLA	
	005600	Ç		EVAL	EMPAPL = *BLA	
	005700	<u> </u>		EVAL	PRJAPL = *BLA	
	005800			EVAL	RSNAPL = *BLA	NK
	005900	C* C* Write	AL OFTROT		A	
	006100	C* write	the SELECI	Iormat	to display. If	end of job requested,
	006200	C*				
	006300	C.*		DOU	not * IN60	
	006400	č		EXFMT	SELECT	
	006500	C *INF	in l	IFEO	'1'	
	006600	C		LEAVESR	I	
	006700	C C C C C C C		ELSE		
	006800	č		EXSR	EDITSL	
	006900	č		END		
	•					
	One or w	ore mismatche	s found			
	5110 OI 1	CIC MIGMOUTO	s round.			

Next, end the compare session.

13. Select Edit > Compare > Clear from the workbench menu bar

Comparing file changes from the CODE Editor (optional)

The CODE tool provides a side by side view of the members being compared. If you prefer this kind of view follow the steps described next. You might want to skip this section if the Compare Tool you just used provides sufficient information for you.

Now you will open a couple of files and edit them and use the CODE compare utility. In the Remote systems view:

- 1. Right-click member **PAYROLL** in **QRPGLESRC**
- 2. Select **Open with > CODE Editor** from the pop-up menu

This opens the CODE Editor window with the PAYROLL member. It will come up in browse mode since it is locked by the LPEX Editor session.

In the CODE Editor, open the PAYROLLG member:

3. Select File > Open option from the CODE Editor menu bar

😽 Select file - Open for Edit			×
File name	<r0s400a>RSELABXX/QRPGLESRC(PAYROLLG)</r0s400a>	Source type	RPGLE
Contains sequence <u>n</u> umbers		Subset by source type	< All Files >
<r0s400a>RSELABXX/QRPGLE</r0s400a>	GF Contents of ' <r0s400a>RSELABXX/QRPGLESRC(*)'</r0s400a>		
QUSRSYS	PAYROLL BPGLE		A
SELABXX	PAYROLLG RPGLE		
COMPARE			
CPOEXER CPOEXER2			
QCBLLESRC			
QCLSRC	_		
QRPGLESRC			
QAPuanc			T
SOURCE	4		Þ
⊡- WEISS ⊡- WFLABXX			
	Befresh Sort Subset: Eind:		
OK Ogen Stop C	ancel Help		
<r0s400a>RSELABXX/QRF</r0s400a>	GLESRC(PAYROLLG)		

In the Open window:

- 4. Expand **R0s400A** connection (the Remote System Explorer connection)
- 5. Expand library RSELABxx
- 6. Select file **QRPGLESRC**
- 7. Select member PAYROLLG from the member list on the right side
- 8. Click **OK** on the Open window
- 9. From the Actions menu, select Compare. The Compare window appears.

Compare X
<r0s400a>RSELABXX/QRPGLESRC(PAYROLLG)</r0s400a>
Select or enter a file name
R0S400A>RSELABXX/QRPGLESRC(PAYROLL)
<r0s400a>RSELABXX/QRPGLESRC(PAYROLL)</r0s400a>
<u>Compare</u> Cancel Help

©Copyright IBM Corporation 2003

Course material may not be reproduced in whole or in part without the prior consent of IBM

All entries are preloaded

10. Click Compare.

The editor now has the PAYROLLG member loaded on the left and member PAYROLL loaded on the right. In between the two members are two long vertical blue lines with horizontal yellow and red bars highlighting the differences in these members.

CODE - <r< th=""><th>R05400A>R5ELABXX/QRPGLESRC(PAYROLLG): 2</th><th></th><th></th><th></th><th></th></r<>	R05400A>R5ELABXX/QRPGLESRC(PAYROLLG): 2				
	ew Actions Compare Options Windows Help				
han	▙		a		
<r05400a>I</r05400a>	RSELABXX/QRPGLESRC(PAYROLLG): 2				>RSELABXX/QRPGLESRC(PAYROLL): 2
Row 1	Column 1 Replace			Row 1	Column 1 Replace Browse
	1+2+3+4+5+6+7.				
000001 000002	F*************************************			000100 000200	F*************************************
000003	F* DESCRIPTION - Time reporting master file maintenance using			000300	F* DESCRIPTION - Time reporting master file ma
000004	F* externally described workstation processing.			000400	F* externally described workstat
000005	F*************************************		-	000500	F*************************************
000006	F* INDICATORS USED F* 50 - No record found on CHAIN operation			000600	F* INDICATORS USED F* 50 - No record found on CHAIN operation
000007 000008	F* 50 - No record found on CHAIN operation F* 60 - General error condition			000700 000800	F* 50 - No record found on CHAIN operation F* 60 - General error condition
000009	F* 90 - Protect display on delete request			000200	F* 90 - Protect display on delete request
000010	F* KC - End of job requested			001000	F* KC - End of job requested
000011	F* KD - Return to application selection			001100	F* KD - Return to application selection
000012 000013	F* KE - Return to employee selection F* KF - Return to project selection			001200 001300	F* KE - Return to employee selection F* KF - Return to project selection
000013	F* KG - Return to reason code selection			001300	F* KG - Return to reason code selection
000015	F* LR - Last record			001500	F* LR - Last record
000016	F*************************************			001600	F*************************************
000017	F* SUBROUTINES USED			001700	F* SUBROUTINES USED
000018 000019	F* EDITSL - Edit application selection display (SELECT) F* ACDESR - Edit action code for all maintenance requests			001800	 F* EDITSL - Edit application selection display F* ACDESR - Edit action code for all maintenar
000020	**************************************			002000	F*************************************
000021	F* This program uses all externally described files. Files			002100	F* This program uses all externally described
000022	F* used are - MSTDSP - maintenance display file			002200	F* used are - MSTDSP - maintenance display fi
000023	F* - EMPMST - employee master file F* - PRJMST - project master file			002300	F* - EMPMST - employee master file F* - PRIMST - project master file
000024 000025	F* - PRJMST - project master file F* - RSNMST - reason code master file			002500	F* - PRJMST - project master file F* - RSNMST - reason code master fil
000026	**************************************			002600	F*************************************
000027	FMSTDSP CF E WORKSTN			002700	FMSTDSP CF E WORKSTN
000028	FEMPMST UF A E K DISK			002800	FEMPMST UF A E K DISK
000029 000030	FPRJMST UFAE K DISK FRSNMST UFAE K DISK			002900 003000	FPRJMST UF A E K DISK FRSNMST UF A E K DISK
000031				003100	
000032	D* Compile time array containing error descriptions.			003200	D* Compile time array containing error descrip
000033	D ERR S 50 DIM(10) CTDATA PERRCD(1)			003300	D ERR S 50 DIM(10) (
000034 000035	D EMESS S 50			003400 003500	D EMESS S 50 F*
000036	E*			003600	L* C***********************************
000037	C* MAINLINE CALCULATIONS			003700	C* MAINLINE CALCULATIONS
000038	C ************************************			003800	C ************************************
000039	C* This mainline routine controls the display file processing			003900	C* This mainline routine controls the display
000040 000041	C* editting. Using the function keys described on each displa C* format, you can transfer from one maintenance application t			004000 004100	C* editting. Using the function keys describe C* format, you can transfer from one maintenar
000042	C* another. The action code you select on the selection forma			004200	C* another. The action code you select on the
000043	C* determines if the program will add a new record to the file			004300	C* determines if the program will add a new re
000044	C* update an existing record in the file.		-	004400	C* update an existing record in the file.
000045 000046	C*************************************			004500 004600	C*************************************
000040	C* Housekeeping, clear display fleids and reset indicators.			004000	C* Housekeeping, clear display fields and rese C*
000048	C EXSR MAIN			004800	C EXSR MAIN
000049	C* IF MAIN is done program ends			004900	C* IF MAIN is done program ends
000050	C eval *INLR = *on			005000	C eval *INLR = *on
000051 000052	* MAIN SUBROUTINE C Main Begsr			005100 005200	* MAIN SUBROUTINE C Main Begsr
000053	C dou *INKC			005300	C dou *inkc
000054	C EVAL *IN60 = *OFF			005400	C EVAL *IN60 = *OFF
000055	C EVAL EMESS = *BLANK			005500	C EVAL EMESS = *BLANK
•	· · · · · · · · · · · · · · · · · · ·			•	
IBM Live H	Parsing Editor			IBM Live	Parsing Editor
			_		
		_	_		

11. Use the vertical scroll bars to move within the files. As you scroll, you will see where the differences are in the members. RPG experts will notice that PAYROLL has some errors in it. We will fix these in a few moments.

12. From the **Compare** menu (which was inserted while performing this action), select **Exit**

Compare to go back to the original view.

13. Close the CODE Editor and continue working in the workbench.

Checking Syntax

One of the powerful features that the LPEX Editor shares with SEU is its ability to syntax check your source. Syntax checking can be done either when the cursor leaves each line of source or all at once on either the currently selected source or on the entire file. Now create a syntax error and watch for the prompt to correct it:

1. In the PAYROLL Editor window move the cursor to line 221 which contains 'EXSR

ACDESR'

You might already be on that line but if not, type the line number in the sequence number

column, or scroll down

- 2. Append an x to the **EXSR** op-code to make it **EXSRX**.
- 3. Move the cursor off of the line.

An error message appears to draw attention to the error.

🕀 i 9	õeries Proje	cts - WebS	phere Dev	velopme	nt Studi	o Client a	Advanc	ed Edition	for iSerie	5		
File	Edit Sourc	e Compile	Navigate	Search	Project	Run S	nap Shoi		•			
] 🖻	- [] []	. 🖨 🛛 🕯	66] 0] 📩	珍 -	<u>*</u> -] \$]	Ŝ t •]	a a	s 9	6 I I I
Ē	📄 Welcome	. /*	PAYROLL	×								
5	Row 2		Column		Replac		chang					
		CI	0N01Fac	tor1+-	+++++	Opcod <mark>e</mark>	(E)+F	actor24	-+++++]	Result	+++++++	+Len++D+HiLa
	020300	C				IF	÷	INKD				
	020400	C				leaves	r					
	020500					END						
	020600											
	020700	C*	Acces	s empi	loyee	master	to v	alidate	e actio	n code	reques	t
	020800											
	020900					IF	-	OT *INF	(C			
	021000		EMF	'NO	-	OHAIN	_	MPMST				50
	021100					EXSRx		CDESR				
			E Opera	tion o				; speci	ficati(on is	ignored	
	021200					ELSE						
	021300					LEAVES	R					
	021400	C C				END						

Figure 37: Editor window with syntax error

4. Move the **cursor** onto the pink error message.

5. Press **F1**.

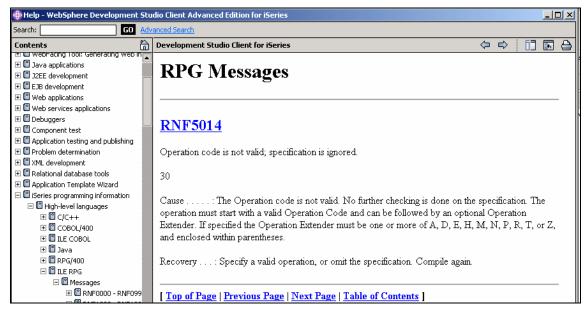


Figure 38: Second level help for syntax error

This opens a window with second level help for the error.

- 6. Minimize the Help window.
- 7. Change EXSRX to EXSR to correct the error.
- 8. Move the cursor off the line you just fixed.

The error message is automatically removed from the editor.

Tip: You can toggle automatic syntax checking by using the Windows > Preferences option on the workbench menu bar and **Remote Systems** then **iSeries** then **LPEX Editor Parsers** and then **ILE RPG** in the Preferences tree view.

Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. When column sensitive editing is selected:
 - A. Each column is considered a separate entry space
 - B. If you are inserting or deleting characters into a string that is in the Factor 2 entry, the result field entry does not move
 - C. The default editor preference is that column sensitive editing is off
 - D. All of the above
- 2. The LPEX Editor has predefined settings, but also has an associated preference page containing settings that you can define. (T, F)
- 3. LPEX Editor preferences are set in the:
 - A. Preferences window

- B. Editor window
- C. New wizard
- D. Remote Systems view
- 4. You can configure the LPEX Editor to adopt the keyboard and command personalities of many popular editors. (T, F)
- 5. If you want to undo a set of changes made to a file you must use the ______ operation. Name the operation.
 - A. Insert
 - B. Replace
 - C. Redo
 - D. Undo
- 6. You can also cancel the effects of an undo operation by using the _____ operation. Name the operation.
 - A. Insert
 - B. Replace
 - C. Redo
 - D. Undo
- 7. This help is invaluable if you cannot remember the order of fields in an RPG specification or the possible values for a variable field. Name the help. Choose the best answer.
 - A. Context sensitive help
 - B. Language sensitive help
 - C. Display help
 - D. Field help
- 8. To receive language sensitive help, press _____ key in an Edit window. Name the key:
 - A. F2
 - B. F3
 - C. F1
- D. F49. If the cursor is ______ an operation code, you receive help for that operation code; otherwise,
 - you receive help for the current specification.
 - A. Before
 - B. After
 - C. On
 - D. Off

10. Instead of entering or changing code directly in the Editor window, you can use _____.

- A. Prompts
- B. Filters
- C. SEU commands
- D. Format line
- E. All of the above
- 11. _____ views are hidden views, which can be quickly made visible. They work identical to normal views, the only difference being that when hidden they do not take up screen space on your workbench window.
 - A. Remote Systems
 - B. Navigator
 - C. Outline
 - D. Fast

- 12. The ______ option allows you to view your source with constructs in an indented mode. By default, the ______ option will split the screen horizontally and how show the indented view in the bottom pane.
 - A. Fast view
 - B. Outdent
 - C. Indent
 - D. Edit
- 13. The indented view is browse mode only and cannot be edited. (T, F)
- 14. You use the ______ window to search for an item. Choose the best answer.
 - A. Search
 - B. Find
 - C. Edit
 - D. Find and Replace
- 15. You can search for:
 - A. A word
 - B. A partial word
 - C. A sequence of words
 - D. A pattern if it follows the rules of regular expression
 - E. All of the above
- 16. The LPEX Editor allows you to _____ or subset your source so that you see only lines containing a given string.
 - A. Search
 - B. Find
 - C. Sort
 - D. Filter
- 17. To help you navigate quickly through your ILE RPG source the editor allows you to filter lines based on the _____. Choose the best answer.
 - A. String
 - B. Line type
 - C. Line number
 - D. All of the above
- 18. If you would like to search through the members in a source physical file or through the files in a local directory, you can use the _____ tool. Choose the best answer.
 - A. Compare
 - B. Search
 - C. Find
 - D. Edit
- 19. The ______ tool allows you to compare different versions of a program and find the differences. Choose the best answer.
 - A. Convert
 - B. Migrate
 - C. Compare
 - D. Search
- 20. There are two ways to compare files. They are Compare tool in the workbench and the Compare tool in the CODE Editor. (T, F)
- 21. Syntax checking can be done either when the cursor leaves each line of source or all at once on either the currently selected source or on the entire file. (T, F)

Practice

Given your experience in this exercise using the Remote Systems LPEX Editor, try these new tasks: Check out how token highlighting works in your ILE RPG source. Display a format line and use it to enter your source. Filter source by comments. Set auto-uppercase on. Use the Development Studio Client for iSeries online help to assist you in these tasks.

What you just did

In this exercise you opened the Remote Systems LPEX Editor and learned about these editor features: column sensitive editing, SEU commands, undo and redo operations, language sensitive help, prompts, indent source, find and replace option, filter lines by string, filter lines by line type, search files, compare files, and syntax checking.

In the next exercise you will verify your source to ensure you have a clean compile on the iSeries system. This approach saves you iSeries cycles! And you perform both verify and compile from the Remote Systems view!

Exercise 5: Verify and Compile RPG

In this exercise you verify and compile ILE RPG in the Remote Systems LPEX Editor. The program verifying and program compiling are the final steps in compiling a program or module. When errors are found by either, the iSeries Error List appears. The iSeries Error List is a powerful tool that manages errors found by verify and compile utilities in the Remote Systems LPEX Editor. You will become familiar with these tools, the various capabilities of the iSeries Error List and the RPG program that you have created.

At the end of this exercise, you should be able to:

- Describe the purpose of the Program Verifier
- Invoke the Program Verifier
- Use the iSeries Error List to insert messages and to check that errors were fixed
- Use the iSeries Table view in Remote System Explorer to submit iSeries commands
- Use the Compile utility to select compile options and compile ILE RPG source
- Explain the generated ILE RPG program called PAYROLL

Invoking the Program Verifier

Now you get to play with one of the most powerful and unique features of the Remote System Explorer – the Program Verifier. Before you compile your code on an iSeries, you can make certain that there are no errors by invoking the Program Verifier. The verifier checks for semantic (compile) errors on your workstation so that you can guarantee a clean compile on the iSeries. Think of the host cycles you'll save. It is especially handy when you are writing code but you are disconnected from an iSeries. You can do this because Remote System Explorer ported the parsing and checking code from the iSeries system compilers to the workstation. The iSeries 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.

To invoke the verifier:

1. Select Source > Verify from the workbench menu bar

After a moment the verifier will display an iSeries Error List underneath the Editor window.

%]	📲 iSeries Error List - CODELABXX/EVFEVENT(PAVROLL) 🗢 🗢 🤝										
	ID	Message	Seve	Line	Location	Con 🔺					
	RNS9308	Compilation stopped. Severity 30 errors found in program.	50	0	CODELABXX/QRPGLESRC(PA	s40i					
	RNF5178	ENDSR operation is missing for subroutine EDITSL; ENDSR statement is assu	30	0	CODELABXX/QRPGLESRC(PA	s40i					
	RNF7030	The name or indicator EMES is not defined.	30	365	CODELABXX/QRPGLESRC(PA	s40i					
	RNF7030	The name or indicator RSNTAX is not defined.	30	95	CODELABXX/QRPGLESRC(PA	s40i					
	RNF7018	The operand RSNTAX of EXSR is not a subroutine name.	30	95	CODELABXX/QRPGLESRC(PA	s40i					
	RNF5184	Factor 2 must be '0' or '1' when the Result is an indicator or an INDDS data st	30	364	CODELABXX/QRPGLESRC(PA	s40					
	RNF7515	Move operands ERR and EMES have types that are not compatible.	30	365	CODELABXX/QRPGLESRC(PA	s40					
i	RNF7031	The name or indicator *IN03 is not referenced.	00	1	CODELABXX/QRPGLESRC(PA	s40i					
i	RNF7031	The name or indicator *INO4 is not referenced.	00	1	CODELABXX/QRPGLESRC(PA	s401 🚽					
•				1							

Figure 39: Verifier error list

The error list will show you:

- 1. The error message itself
- 2. The severity
- 3. The line number
- 4. The source location
- 5. The connection name

Fixing errors

Next you fix the errors in your source.

To fix an error in your source go the error list:

1. Double-click on the error **RNF7030**

You are automatically brought back into the Editor window to the line where the error occurred. The error on line 365 is a typo.

The variable EMES is really EMESS, fix that, one good way of finding the correct name is the content assist.

ment Studio Clie	ent Adv	anced Edition for is	ieries				
ct Run Snap-Sh	ot Win	idow Help					
] I] 🍕 🕇		a e a e					
📋 Welcome	PA'	YROLL ×				3	Outline 😥
Row 365			place				lobal Definitions
		N01Factor1+++			+++++++ <mark>Re</mark> sult++++++++	Len++D+	🗄 🖓 Files
036500	С		MOVE	ERR(2)	EMES	A	indicators
RNF	7515:	Mov Field (char:50)			I ↓ EDEPT		A
036600	7030:	The			ELOCN		edure
036700	č				Ш́шемсат		utines
036800	č				EMESS		- denies
036900	C*						
037000	C*	The					
037100	C*	app			<u></u> EMPNO		
037200 037300	C*	1			I I ENAME		
037400	C C	PRJAPL	ANDEQ	1 1			-
037500	č	RSNAPL	ANDEQ	1 I	L		
000000			NOUT		T11 / O		

Figure 40: Content assist

 Select the misspelled variable and press CRTL+spacebar If it starts correctly the selection presented should contain the correct name Another way is to use the Outline view and see what variables are declared already

ient Studio Clier	nt Advar	nced Edition for iSeri	es				_ 🗆 🗙
Run Shap Sho	t Windo	w Help					
A % •] 🚑 🧔	4 4 1					
📄 Welcome	/ PA	roll ×				🚰 Outline	ي چ کې
Row 365		olumn 51 Rej N01Factor1++++	place	R) . R			
	C CLU	NUIFactor1++++				i Files	
036500		Move operands	MOVE	ERR(2)		🙈 🖳 Indicators	
	F7030:	The name or i				Fields	
036600	C C	THE HAME OF I	ELSE	neo is not u	er meu .	ACODE : Character (1)	
036700	č		MOVE	'0'	*IN60	ACREC : Character (1)	
036800	č		END	-		EDEPT : Character (5)	
036900	C*						
037000	C*	The following			sures that	ELOCN : Character (30)	
037100	C*	application h	as been sel	lected.			- II
037200	C*					EMESS : Character (50)	>
037300	ç	EMPAPL	IFEQ			La contrata character (1)	
037400	Č,	PRJAPL	ANDEQ			EMPNO : Packed Decimal (6,0)	
037500	000000	RSNAPL	ANDEQ MOVE	111	*IN60	ENAME : Character (30)	
037800	č –		MOVE	ERR(3)	EMESS	ENHRS : Packed Decimal (3,1)	
037800	- č		END	ERR(3)	LIESS	EPHRC : Packed Decimal (5,1)	
037900	Č*		LHD				
038000	Č*	The following	code checl	ks each appl	ication sel	EPHRP : Packed Decimal (7,1)	
038100	Č*	ensure that i				EPHRY : Packed Decimal (7,1)	
038200	C*					EPNRC : Packed Decimal (5,1)	
038300	C	EMPAPL	IFNE			EPNRP : Packed Decimal (7,1)	
038400	С	EMPAPL	ANDNE	'Χ'		EPNRY : Packed Decimal (7,1)	
038500	ç		MOVE	<u>'1'</u>	*IN60	ERR : Character (50) DIM(10)	
038600	ç		MOVE	ERR(1)	EMESS	EUSRI : Character (8)	
038700	0000	DDIIDI	END				
038800		PRJAPL	IFNE	'X'		OPCODE : Character (3)	
038900	ç	PRJAPL	ANDNE	· X ·	*TN60		

Figure 41: Outline view with expanded fields

The next error is a RNF7030 as well

3. Double click on it

Fix it in the editor

- 4. **RSNTAX** should really be **RSNTAG**. Make the appropriate change.
- 5. Go to the next error **RNF5184**

Actually the next error is **RNF7018** but it is related to the first one and can be ignored, it shows the same line number, which an indication that both errors are related.

- 6. Double click on **RNF5184**, certainly somebody didn't know his RPG and tried to assign a value 2 to an indicator
- 7. Fix this by replacing the 2 with a 1

Error **RNF7515** is related to the first error you fixed, same line number, you can ignore it.

The only serious remaining error is **RFN5178E**. This error is caused by a missing ENDSR.

Note: The verifier could not determine where the ENDSR is missing so the line number reported is 1, for this reason you can't just double click on the error message, you need to investigate where the missing statement belongs.

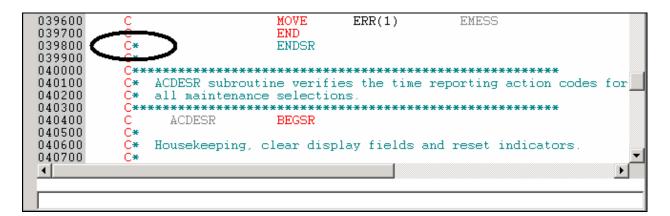


Figure 42: ENDSR commented out by mistake

- 8. Move to line 398 in the editor
- 9. Remove the * from the C spec

Tip: you can use the Tab key to quickly move to the appropriate columns

All the non-informational errors are now fixed.

You can filter out different severities by using the filter menu

	ID	Message	Seve	L ↓	· f	Information	Show Sever	rity 🕨 📘
	RNS9308	Compilation stopped. Severity 30 errors found in program.	50	ί,		Warning	Messages	+
	RNF5178	ENDSR operation is missing for subroutine EDITSL; ENDSR statement is assu	30	(4	wanning	GLESRC(PA	s400a
	RNF7030	The name or indicator EMES is not defined.	30	K *	. 🥸	Error	GLESRC(PA	s400a
	RNF7030	The name or indicator RSNTAX is not defined.	30	9.4		Severe	GLESRC(PA	s400a
	RNF7018	The operand RSNTAX of EXSR is not a subroutine name.	30	٤.		Terminating	GLESRC(PA	s400a
	RNF5184	Factor 2 must be '0' or '1' when the Result is an indicator or an INDDS data st	30	3	_	- concerning	GLESRC(PA	s400a
	RNF7515	Move operands ERR and EMES have types that are not compatible.	30	365	5	CODELABXX/QRF	PGLESRC(PA	s400a
i	RNF7031	The name or indicator *IN03 is not referenced.	00	1		CODELABXX/QRF	PGLESRC(PA	s400a
i	RNF7031	The name or indicator *IN04 is not referenced.	00	1		CODELABXX/QRF	PGLESRC(PA	s400a
i	RNF7031	The name or indicator *IN05 is not referenced.	00	1		CODELABXX/QRF	PGLESRC(PA	s400a

Figure 43: Filter for severity levels

- 10. Click the arrow in the window bar as shown in Figure 43
- 11. Deselect the severities you don't want to see in the list (Information for example)

Saving a Source Member

Now before you loose any of your changes, it's a good idea to save them. Make sure the PAYROLL

member is selected. You then verify the source again to make sure that all the errors are fixed. You

can save the member using one of these ways:

- 1. The File option on the workbench menu bar
- 2. The workbench toolbar (click on \square)
- 3. Press Ctrl+S.

Changes are uploaded to the iSeries.

4. Verify your source again.

ID	Message	Severity	Line	Location	Connection
RNF7031	The name or indicator ACDEND is not referenced.	0	414	CODELAB/QRPGLESRC(PAYROLL)	s400a
RNF7066	Record-Format EMPLSTCTL not used for input or output.	0	27	CODELAB/QRPGLESRC(PAYROLL)	s400a
RNF7066	Record-Format EMPLSTSFL not used for input or output.	0	27	CODELAB/QRPGLESRC(PAYROLL)	s400a
RNF7089	RPG provides Separate-Indicator area for file MSTDSP.	0	27	CODELAB/QRPGLESRC(PAYROLL)	s400a
RNF7031	The name or indicator OPCODE is not referenced.	0	2	CODELAB/QRPGLESRC(PAYROLL)	s400a
RNF7031	The name or indicator POS_TO is not referenced.	0	2	CODELAB/QRPGLESRC(PAYROLL)	s400a
	· · · ·			· · ·	-

Figure 44: Error list, only informational message are left

Everything should be ok. You are ready to compile the program.

Invoking a remote compile

The remote compile capability is part of the Remote System Explorer. It gives you a workstation interface to submit requests to iSeries to compile, bind, or build objects on the host. It allows for easy access to all the compile options available for all the supported CRTxxx commands.

If you used the local program verifier, then your host compiles should be successful -- no wasted iSeries cycles. However, if there are errors, the host compiler will send the error information back to the workstation and they will be loaded into the iSeries Error List window, which behaves just as it did when you did a program verify.

The default for compiling programs is to submit the compile to the batch job queue. Here in this exercise you can run the compile interactive.

To change the preferences to compile in interactive:

1. Select the **Windows > Preferences** from the workbench menu bar

🕀 Preferences			
🕂 Plug-In Development	•	Command Execution	
Profiling and Logging AD RAD Remote Systems		Command Execution Preferences	
		Preferences for compiles and use	r action variables
- Cache		Object library:	*SRCLIB
Command Executio	>	Replace object	
Command Subsyste	<	Compile in batch	
COBOL		Preferences for batch compiles, o	commands, and user action variables
DDS ILE RPG		Job description library:	*LIBL
RPG/400		Job description:	*USRPRF
- Objects		Job description:	
🖻 Program Verifiers		SBMJOB additional parameters:	
DDS PRTF			
. ILE COBOL			

Figure 45: Change Compile command default

- 2. Expand Remote Systems
- 3. Expand iSeries
- 4. Select Command Execution
- 5. De-select the Compile in batch check box
- 6. Click **OK** to get back to the Remote System Explorer perspective

Starting the compile

Starting a compile is like using any other command in the Remote System Explorer. The pop-up menu for source members includes compile submenus. The compile submenus contain several compile commands for various member types. A compile name is an identifier for a compile command to run on a remote system. For example, the compile label CRTBNDRPG tells the iSeries system to create a bound RPG program. Compile names are only associated with source member types: members (*MBR) and modules (*MODULE). For this reason, you will only see the options to compile when you right-click these types of elements. There are two ways to compile: prompted and non-prompted. Both compile actions invoke a submenu with a list of compile names. In the Remote Systems view that shows your expanded library RSELABxx:

🕀 R	emot	e Syste	m Expl	ore	r - Web	Sphe	re D	evelo	pm	ent	Stu	lio C	lient	Adv	anc	ed E	ditio	n for	iSeri	ies	
File	Edit	Source	Comp	ile	Navigati	e Se	arch	Proj	ect	Ru	in V	/indo	w H	łelp							
) È	-		æ	0	1	: - ,3	k -		4	6	<u>}</u> t •	1	a I	6			►	Δ	1L M	圆	
Ē		Remote S	vstems		1		-	7.		×								<u> </u>			
					_		-	<u> </u>	_	~											
	<u>}</u>	\$ ⁹ 📃	\$ ¢		-					_	ode	(E)	+Fa	cto	r2+	+++4	+++-	+Res	ult	+++	++++Le
Ċ.				_	.C1.*pgr						Е		ER	R(1	0)			EME	SS		
_				_	.R1.*pgr			_													
2			. –		YCOMPIL AYROLLD		·				sr										
					AYROLLU AYROLLG																
					RCODE																
					OMPARE						[
					1PMST.*			-			SR								~ ~		
					FEVENT			а			E E		'1 FR	R(4	5			*IN EME			
				- ·	5TDSP.*						ίŜR -				<u></u>				.00		
			÷[PR	UMST.*I	file.pf	-dta													-	
				_	CBLLESR		•	rc										elow ing			array array
				-	CLSRC.*							uen.				Jeg I		ing	01	CHE	array
				-	CMDSRC		•				pti										
				_	DDSSRC								EQU FO					°₽			
					RPGLESE			src					AIN				HIN	-E			
					PAYRO			- 60	То				•		"I						
				_	PHIRC										TEI	Į FI	LE				
				_	EFMST.*					With			•		EXI						
				- · · ·	5NMST.*	•		Bre	owse	e Wil	th		•		TEI						
					DURCE.*			Т Da	n am						EXI	IST ETEI					
					/COMPIL		od 📃			IC I					CLI		, 				
					RCODET			o ا	ΡY					tie	ona	l a	nd/	or ,	warı	nin	g messa
			H II O	ODEI	LABADM	.*lib.p	ro 🔹	🖡 Mo	ve												-
			P 🕪 🛛	ODEI	LABXX.*	lib.pro	od 🧕	🗴 De	lete					E	_	_	_	_		_	
		÷	Prill 🚺 🖸	ODEI	LAB00.*	lib.pro								-							
		Ė			LAB01.*	•	bd	Ve	rify												Severity
				-	.C1.*pgr			Ve	rify	(Pro	mpt)			:d.							0
					.R1.*pgr				انصم					hro	ut <u>nu</u>	ŀ					n
								_	<u> </u>	<u> </u>	rompt)	•		B ₂ (IRTBI	NDRE	•G		5	
				_	AYROLLD AYROLLG			Us	er A	caoi	15				Eòr (RTR	PGM	OD			
				_	RCODE			۵d	d To	n iSe	ries P	rojec	t		10						
					DMPARE						able (·			- 	Nork	With	Comp	oile Co	omma	nds
					OEXER.								-	1							
		I I	- 1			1		Pro	oper	ties				Ser	ries F	rror l	list				

Figure 46: Select member to compile

- 1. Right-click the **PAYROLL** member in QRPGLESRC
- 2. Select Compile (Prompt) > CRTBNDRPG option from the pop-up menu

Compiling Your Source

You will now use the prompt for the CRTBNDRPG command to specify your compile parameters.

All entry fields pertaining to names are already filled in with the correct information.

📸 Create Bound RPG Program	(CRTBNDRPG)	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp		
Program: Library:	PAYROLL	Name Name
Source file: Library:	QRPGLESRC - RSELABXX -	Name, QRPGLESRC Name
Source member:	PAYROLL 💌	Name
Generation severity level:	10	0-20
Text 'description':	*SRCMBRTXT 💽	Character value
Default activation group:	⊙ *YES O *NO	
Adv Compiler options:	/anced Parameters	Add
	*EVENTF	Remove Move up
Debugging views:	*ALL	Move down
Advanced	ОК	Cancel Help ?

Figure 47: Prompt for CRTBNDRPG

- Change the **Debugging views** parameter to***ALL**.
 If you want check out the other parameters available, click **Advanced**
- 2. Click **OK** when you are done.

A window box will show that the compile started.

The progress bar on the workbench (bottom right corner will indicate that the compile runs) then the error list will be shown, with no errors, just informational messages. Make sure you have Information Messages selected as a type of message to show. You may have unchecked this selection in a previous exercise.

If you are not sure that the compile was successful:

🗋 Welcome	PAYROLL X	🗄 Outline
Row 101	Column 1 Replace	🖃 🎑 Global Defini
	CLON01Factor1++++++Opcode(E)+Factor2++++++Result++++++Len++D	Files
007400	C ENDdo	
007500	C*	🕀 🖳 Indicato
007600	C* The application selection fields from the SELECT format are	🗄 는 Fields
007700	C* tested and the program will branch to the section specific	🖃 🍓 Main Proced
007800	C* that application.	🗄 🛱 Subrout
007900	C* If EMPAPL (employee maintenance) equals X, the program	
008000	C* branches to label EMPTAG.	
008100	C* If PRJAPL (project maintenance) equals X, the program	
008200	C* branches to label PRJTAG.	
008300	C* If the prior two tests were not successful, you have chosen	
008400	C* reason code maintenance. The program will continue with the	
008500		
008600		
008700	C SELECT	
008800	C when EMPAPL = 'X' C EXSR EMPTAG	
008900	C EXSR EMPTAG	
009000		
009100	C when PRJAPL = 'X' C EXSR PRJTAG C*	
009200	C EXSR PRJTAG	
009300	C*	
009400	C when RSNAPL = 'X' C EXSR RSNTAX	
009500	C EXSR RSNTAX	
009600	C ENDSL	
009700	C* if KC end program	
009800	C if *INKC	
009900	C leave	
010000	C endif	
010100	C enddo 🧾	
010200	C endsr 🗾	
•	b b	
🔹 iSeries Comm	ands Log	
**		
toras06m s400a		
the CENI VI, par	ameter. Compile again.	
one demote par	annasan asmpira againn	
CRIBNDRPG PG	M(RSELABXX/PAYROLLG) SRCFILE(RSELABXX/QRPGLESRC) SRCMBR(PAYROLLG) OPTION(*EVENTF) DBGVIEW(*ALL)
	DLLG placed in library RSELABXX. 00 highest severity. Created on 07/17/03 at 13:41:11.	
	Program PAYROLLG was successfully created in library RSELABXX. The highest message severity that resulted	was 00. The progra
	nd time are 07/17/03 and 13:41:11.	
]		
Command Norm		Prompt.
Tasks iSeries Con	nmands Log (Series Error List iSeries Job Log Console iSeries Field Table View iSeries Table View Search Re	emote Search
isones con		

Figure 48: iSeries Commands Log

3. Click the **iSeries Commands Log** tab at the bottom of the workbench.

This log shows a list of all commands run on the remote system and the messages that have been returned for each command.

Submitting iSeries Commands in the iSeries Table view

You can use the iSeries Table view inside the Remote System Explorer to submit commands to the iSeries. You can run commands in the Commands field beneath the iSeries Table view, and view messages in the Messages field. After you populate the table, you can enter a command and select either Prompt to specify parameters and then Run, or just select Run (which prompts and runs the command as normal). When you run a command, the Messages drop-down field is populated with the messages from the command. When you select a message, the Details button is enabled. When you press this button, the message and its help is displayed.

Also note that you can use the iSeries Table view or the Remote Systems view to run commands and programs. In the iSeries Table view, you can see the properties of all items at the same time; they are displayed as rows across the table. In the Remote Systems view, you have greater ease of navigation; you can work from your Library list in the iSeries Objects subsystem, and you can see the contents of many items before selecting the one you want to run.

	000	Diric .	vennea	OK .			
h MSTDSP2	DSPF	SRC	with no errors	ОК			
🐚 PRJMST	PF	SRC		ОК			
🐚 REFMST	PF	SRC		ОК			
🐚 RSNMST	PF	SR <u>C</u>		ОК			
Command Normal		chgcurlib rselab:	\sim	Prompt Run			
Messages Command has	completed			Details Show Log			
Tasks iSeries Commands Log iSeries Error List iSeries Job Log Console iSeries Field Table View Search Remote Search							

To change your library list:

Figure 49: iSeries Table view with command entry

- 1. Select the **iSeries Table View** tab from the views at the bottom of the workbench
- 2. Enter in the Command field: CHGCURLIB RSELABXX
- 3. Click Run.

If you haven't used the iSeries Table view to show iSeries objects in this view you will get this error message because the table view is not linked to an active connection.

⊕ E¥FM1025E		×
No iSeries connection exists for the view. view's menu.	Specify a connection us	ing the
	OK	<u>D</u> etails>>

Figure 50: Error message when using iSeries Table view without active connection

If you get this message, click **OK** and go to the Remote Systems view

	Right-click										
R	emote Syste	m Explor	er - WebS	phere De	evelopm	ient St	udio (
File	Edit Source	Compile	Navigate	Search	Project	Run	Winde				
B	- 🛛 🗒	8](]] 🏷	• <u>*</u> •] \$] R	-]				
Ê	🌆 Remote S	ystems		•	∎* -	×					
	è 🕸 🗖	$\Leftrightarrow \Rightarrow$	Ê.				le(E				
Ĉ.			ILC1.*pgm			▲ E					
_			ELR1.*pgm MYCOMPILE		_						
		. –	PAYROLLD.			.sı	r				
			ATROLLD.								
			STRCODEC								
		÷	COMPARE.*	*file.pf-sro	:		_				
			EMPMST.*fi			ISH TE	x				
			EVFEVENT.*		э	E					
		MSTDSP.*file.dspf ISR									
		E ■ PRJMST.*file.pf-dta ■ OCBLLESRC.*file.pf-src									
	to de:										
			QCMDSRC.'		-		tion				
		÷ E	ZODOORC."	filo of cro	_		NOT				
			QRPGLESP	- wear - e		dT F	FOTE				
				Go To							
			👌 PAYR(_	GOTO			<u> </u>				
			QRPGSRC	🔊 Refres	sh		6. N				
			REFMST.*	Expan	d		ľ				
			SOURCE.	Collap	se		P.				
				T -			P.				
				🔁 Renan	ne		f				
	÷	Hink Cod	ELABADM	Copy							
			ELABXX.*	Paste			F				
		- 🚺 COD		⊷ ∯ Move							
	Ē		LLADOI.								
				💥 Delete	:						
			ELR1.*ng MYCOMPL	Show	in Table						
						_					

4. Right-click QRPGLESRC

Figure 51: Select Table view to connect it to iSeries

- 5. Select the Show in Table option from the pop-up menu The table view is now populated with the member in the selected source file.
- 6. Run the CHGCURLIB again

The command will run on the iSeries and after completion you will see the completion message on the bottom of the iSeries Table view.

Name	Туре	Attribute	Text
PAYROLL	RPGLE	SRC	
DAYROLLG	RPGLE	SRC	
ommand Normal			
essages Current libra	- ary changed to RSELABXX.	$ \longrightarrow $	
		Series Job Log _I Console _I is	

Figure 52: Completion message for command

Tip: You can also connect to other than iSeries systems with the Remote System Explorer and

Launch commands for these systems as well, for example, your local system, or LINUX.

Running Commands and Programs

You can run programs and commands from the Remote Systems view or the iSeries Table view in three ways:

- 1. In the Remote System Explorer communications server job
- 2. In a batch job
- 3. In an interactive job

Using the first option lets you run the program in the same job as the communications server. 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 command ends. Batch jobs work as you would expect and do not require any initial setup. Interactive programs require a 5250 emulator, so you need to first run a STRRSESVR <connectionName> command to associate the emulator with a particular connection in the Remote System Explorer communications server.

Starting an interactive connection

To start an interactive connection:

- 1. Start a 5250-emulation session.
- 2. Sign on to the iSeries with your User ID and password

Note: Instead of the **Enter** key, you may have to use the **Ctrl** key in your 5250-emulation session.

The Edit View Communication Actions Window Help MAIN OS/400 Main Menu Select one of the following: System: TORASO7 1. User tasks Office tasks 2. Office tasks General system tasks 4. Files, libraries, and folders Programming 6. Communications Define or change the system 7. Define or change the system Problem handling 9. Display a menu Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant	🕫 🕽 Session A - [24 x 80]		- 🗆 ×
MAIN OS/400 Main Menu System: TORASO7 Select one of the following: 1. User tasks 2. Office tasks 3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command SIRRSESVR NAME (s400a) RMTLOCNAME (*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant			
System: TORASOTA Select one of the following: 1. User tasks 2. Office tasks 3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command TS=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant			
Select one of the following: 1. User tasks 2. Office tasks 3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command => <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant	MAIN OS/400 Main M	Menu	
<pre>1. User tasks 2. Office tasks 3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command => <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>		System: TORAS	S07M
<pre>2. Office tasks 2. Office tasks 3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection er command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>	Select one of the following:		
<pre>2. Office tasks 2. Office tasks 3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection er command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>			
<pre>3. General system tasks 4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command ==> STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>			
<pre>4. Files, libraries, and folders 5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</u></pre>			
<pre>5. Programming 6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>	-		
<pre>6. Communications 7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command => STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>			
<pre>7. Define or change the system 8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection er command => STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>			
<pre>8. Problem handling 9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>			
<pre>9. Display a menu 10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>			
<pre>10. Information Assistant options 11. Client Access/400 tasks 90. Sign off Selection or command ==> STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>	-		
<pre>11. Client Access/400 tasks 90. Sign off Selection or command ==> STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE) F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant</pre>	• -		
90. Sign off Selection or command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant			
Selection er command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant	11. Client Access/400 tasks		
Selection er command ==> <u>STRRSESVR NAME(s400a) RMTLOCNAME(*RESOLVE)</u> F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant	00 01 <i>(</i> (
F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant	90. Sign off		
F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant	Selection on the selection of the select		
F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant		VE)	
	STRASESTA NAME (S4008) AMTEOCNAME (*RESOL		
	E3=Exit E4=Prompt E9=Retrieve E12=Capo	cel E13=Information Assistant	
F23=Set initial menu	F23=Set initial menu		
	120 Oct Inference		
MA a 20/01	MA a	26	1/007
Image: Connected to remote server/host toras07m using port 23			

Figure 53: Start Remote System Explorer interactive connection

3. In the emulator, enter the command

STRRSESVR <connectionName> RMTLOCNAME(*RESOLVE)

The connectionName parameter is the name of your connection defined in the Remote Systems view. This associates the interactive job with the Remote System Explorer communications server. The ***RESOLVE** keyword will get the IP address of your workstation and with this information the Remote System Explorer server will communicate with the Remote System Explorer daemon that runs on your workstation.

Note: The connection name is case sensitive and must already be defined in the Remote System Explorer.

You should see a screen as shown in Figure 54.

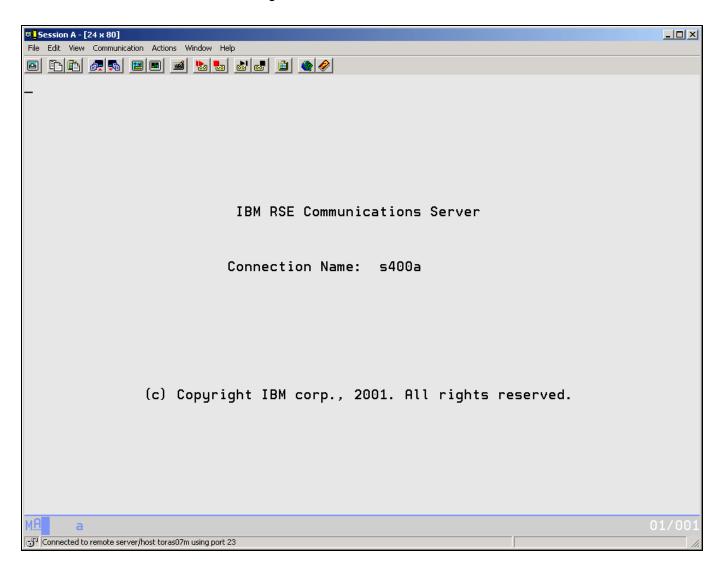


Figure 54: Remote System Explorer connection screen

Note: This screen will stay this way. Don't wait for it to finish. This session is the interactive session for interactive commands started from the Remote System Explorer.

Running the PAYROLL Program

Now you are ready to run the **PAYROLL** program that you just compiled.

Return to the workbench, and in the Remote Systems view:

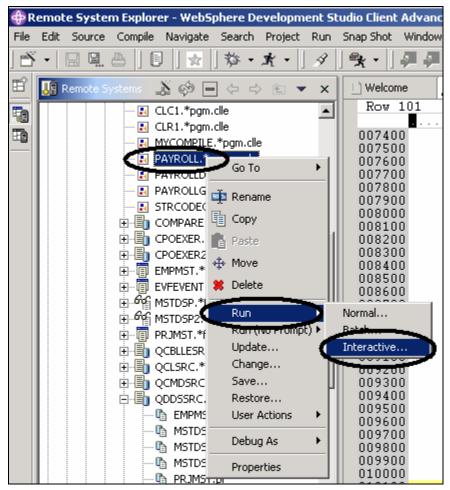


Figure 55: Run program in interactive job

- 1. Locate the **PAYROLL** program that you created
- 2. Right-click the PAYROLL program in the Remote Systems view
- 3. Select **Run As > Interactive** from the pop-up menu
- 4. Switch to your 5250-emulation session.

You will see the Start Menu of the payroll program.

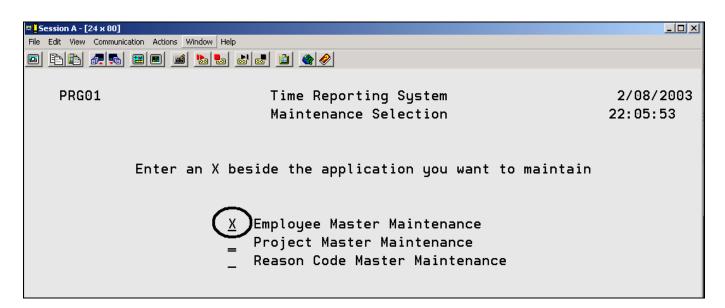


Figure 56: Payroll Start Menu

- 5. Place an 'x' beside Employee Master Maintenance.
- 6. Press Enter.
- 7. Type **123** for the Employee Number.
- 8. Type **A** for the Action Code to add employee **123**.
- 9. Press Enter.
- 10. Type any information you like about the employee.
- 11. Press Enter.
- 12. Play in the application as much as you like.
- 13. Press F3 to end the PAYROLL program.

Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. The _____ tool checks for semantic (compile) errors on your workstation so that you can guarantee a clean compile on the iSeries.
 - A. Compile
 - B. LPEX Editor
 - C. Program Generator
 - D. Program Verifier
- 2. The ______ view identifies each error with an icon that identifies the severity level of the error, the ID of the error, the message, the severity, the line in the source member that caused the error, the location of the source member that produced the error, and the connection name.

- A. Remote Systems
- B. Outline
- C. Navigator
- D. iSeries Error List
- 3. You can sort the entries in the iSeries Error List view by:
 - A. ID
 - B. Message
 - C. Severity
 - D. Line
 - E. Location
 - F. Connection
 - G. All of the above
- 4. If you used the local program verifier, then your host compiles should be successful; no wasted iSeries cycles. (T, F)
- 5. A compile command can be run on an iSeries server from the Remote System Explorer, and you can retrieve error feedback from the compile. (T, F)
- 6. Each profile in the Remote System Explorer has a set of source member types, and each source type has a set of compile names associated with them. (T, F)
- 7. There are several ways to compile. They are:
 - A. Prompted
 - B. Non-prompted
 - C. Interactive
 - D. Batch
 - E. Remote System Explorer communications server
 - F. All of the above
- 8. The iSeries Table view displays the same information as the ______ view, with the ability to sort items and perform numerous actions.
 - A. Remote Systems
 - B. Outline
 - C. Navigator
 - D. iSeries Error List
- 9. From the iSeries Table view you can:
 - A. List and sort libraries, objects, and members
 - B. Copy, rename, delete, edit, compile and run any item in the view from the pop-up menu
 - C. Transfer files from one system to another
 - D. All of the above
- 10. You can run programs and commands from the Remote Systems view or the iSeries Table view in:
 - A. A Remote System Explorer communications server job
 - B. A batch job
 - C. An interactive job
 - D. All of the above
- 11. The ______ view identifies each error with an icon that identifies the severity level of the error, the ID of the error, the message, the severity, the line in the source member that caused the error, the location of the source member that produced the error, and the connection name.
 - A. Remote Systems
 - B. Outline

- C. Navigator
- D. iSeries Error List
- 12. Interactive programs require a 5250 emulator, so you need to first run a STRRSESVR <connectionName> command to associate the emulator with a particular connection in the Remote System Explorer Communications server. (T, F)

Practice

Given your experience in this exercise using the Program Verifier and Compile command, and that you have your own RPG source on your own iSeries system try these new tasks: Check out Compile (Prompt) Work with Compile commands. Assuming you receive errors in your source (add some errors into your source if you don't have any) when you verify your source, choose insert all error messages into the editor from the Error list. Use the Development Studio Client for iSeries online help to assist you in these tasks.

What you just did

In this exercise you verified your source by invoking the Program Verifier from the Remote Systems view. You then reviewed the Error List, fixed some errors and saved the source. Next you ran a remote compile from the Remote Systems view. You used the iSeries Table view to enter a command. Finally you started an interactive connection to the iSeries and ran your program all from the Remote Systems view.

In the next exercise you will use the CODE Designer tool to visually work with DDS source to design a screen for the RPG payroll program.

Exercise 6: Design Screens and Reports

In this exercise you will become familiar with the various aspects of the CODE Designer while modifying a display file to add a screen.

At the end of the exercise, you should be able to:

- Describe the features of the CODE Designer
- Start the CODE Designer
- Open, modify and save a display file
- Add a group
- Add a record
- Add, delete and modify various DDS fields
- Verify DDS
- Compile DDS

Using an editor to create and maintain DDS source for your display and printer files can be a frustrating and difficult task. What would be great is a graphical design tool that let's you design your screens and reports visually and then generate the DDS source for you. Well, that's exactly what the CODE Designer does for you.

The CODE Designer interface was designed to help the novice DDS programmer create screens, reports and databases quickly and easily without worrying about the details of the DDS language, while at the same time letting the expert DDS programmer get access to all the features and power of the language. Let's now step through each part of the interface and update some DDS as well.

In the workbench, in the Remote System Explorer perspective use the connection that you used in the exercise before.

Opening a DDS member in the Remote Systems view

To open a DDS file member in the Remote Systems view:

- 1. Expand the Library List filter or if it is still expanded use it as is
- 2. Expand the **QDDSSRC** file in library **RSELABxx**
- 3. Right-click MSTDSP member
- 4. Select **Open with > CODE Designer**

The member **MSTDSP** will be downloaded to the workstation and loaded into CODE Designer.

Viewing the DDS Tree

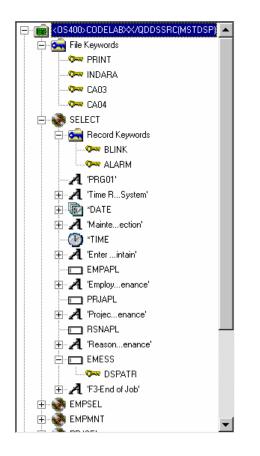
What you are looking at now is basically an explorer view of the DDS. The DDS Tree view on the left-hand side of the Designer displays the DDS source in its file, record, field, and keyword hierarchy. It is a familiar and intuitive way to see the overall structure of the DDS source and to navigate through it quickly. Don't worry if you're not a DDS expert, we'll explain everything you need to know.

The DDS Tree is located on the left-hand side of the CODE Designer window. This view presents the loaded DDS source as a tree structure, showing the hierarchy of files, records, fields, help specifications, keys, and keywords in each selected DDS object. The DDS Tree shows groups of records, which represent the screens or reports you are designing, as peers of the file in the tree hierarchy.

In this view, you can create groups, and copy or move keys, keywords, fields, and records. If any DDS object contains an error, the icon representing it displays a red *X*.

To open selection in the DDS tree to see file-level keywords and the record SELECT:

- 1. Click the plus sign beside the folder <Servername>RSELABxx/QDDSSRC(MSTDSP).
- 2. Click the plus sign beside the folder File Keywords.
- 3. Click the plus sign beside the record **SELECT**.
- 4. Click the plus sign beside the folder **Record Keywords**.
- 5. Click the plus sign beside the field **EMESS**.



The DDS Tree now shows you a summary of the file-level keywords and of the record SELECT.

Selecting the DDS object

In the upper right-hand side of the Designer is the Workbook with several different tabbed pages. The Workbook is the area of the CODE Designer where you design display files or printer files. You can view this notebook on the top right-hand side of the CODE Designer window. The top page is called Details and it provides a detailed view of the DDS objects selected by the DDS Tree. You can view this page in either details mode or list mode.

In *details mode*, columns display information about each DDS object. You can use this mode if you want to know more about each DDS object (record, help specification, field, keyword, or keys).

In *list mode*, the DDS objects appear as columns of named icons. You can use this mode if you want to see more DDS objects within the page.

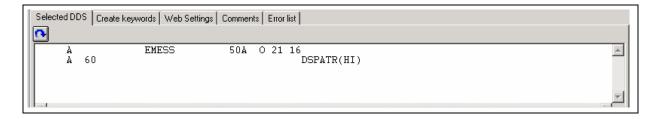
In the bottom right-hand side of the Designer is the Utility notebook. It lets you view specific elements of the DDS source such as errors and selected objects. The notebook contains several views: Selected DDS page, Error List page, Create keywords page, and Comments page. The Selected DDS page in the notebook shows the actual DDS source for the currently selected item.

To work with the DDS record **SELECT**:

 In the DDS Tree click the record SELECT. The Details page lists all the fields in the record SELECT and summarizes some of their properties. The Selected DDS page shows the DDS for the SELECT record.

Field	Position	Length	Туре	Shift	Usage	Sample
A 'PRG01'	2, 5	5	Text constant			PRG01
A 'Time RSystem'	2, 30	21	Text constant			Time Reporting System
🚾 *DATE	2, 70	6	Date constant			YY/MM/DD
A 'Mainteection'	3, 30	21	Text constant			Maintenance Selection
🕐 *TIME	3, 70	6	Time consta			HH:MM:SS
A 'Enterintain'	6, 14	54	Text constant			Enter an X beside the application you
EMPAPL	9, 25	1	Alphanumeric	A - Alphanumeric	Both	В
A 'Employenance'	9, 28	27	Text constant			Employee Master Maintenance
📼 PRJAPL	10, 25	1	Alphanumeric	A - Alphanumeric	Both	В
A 'Projecenance'	10, 28	26	Text constant			Project Master Maintenance
E RSNAPL	11, 25	1	Alphanumeric	A - Alphanumeric	Both	В
A 'Reasonenance'	11, 28	30	Text constant			Reason Code Master Maintenance
EMESS	21,16	50	Alphanumeric	A - Alphanumeric	Output	000000000000000000000000000000000000000
A 'F3-End of Job'	23, 7	13	Text constant			F3-End of Job
•						Þ

- In the DDS Tree click Record keywords immediately below SELECT. The Details page shows the current record-level keywords. The Selected DDS page still shows the DDS for the SELECT record.
- 3. In the DDS Tree click the field **EMESS**. The Details page shows its field-level keywords. The Selected DDS page now shows the DDS for the **EMESS** field.



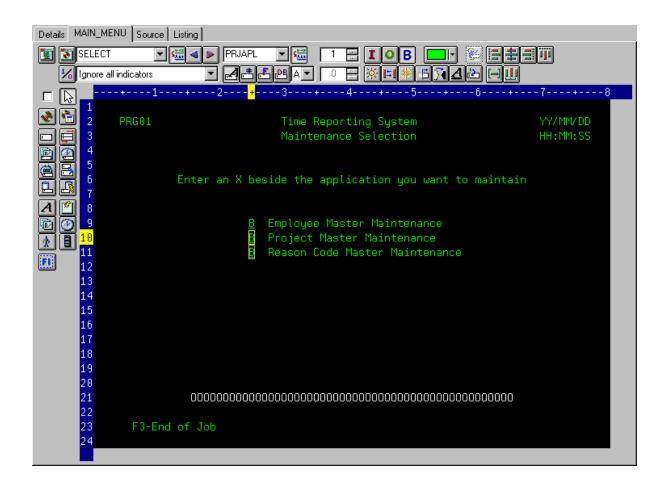
Even this relatively small and simple DDS source member demonstrates how much easier it is to use the Designer to navigate through your DDS source. The syntax is being interpreted in intuitive graphical ways making it an ideal tool for learning DDS. But to get orders of magnitude improvement in your productivity what you really need is to work with your screens and reports in a WYSIWYG fashion, completely oblivious to the DDS required to make things appear the way they do. You need the Design Page.

Designing your DDS Screen

You will spend most of your time creating, updating, and designing your DDS screens and reports in the Design page. The Design pages allow you to design your screens or reports visually using an intuitive graphical user interface. The Design page shows the DDS source as it would appear on either a screen (for display files) or a printed page (for printer files). It allows you to design your application's screens or reports by laying out records and fields in a graphical user interface.

On the Design page, you can easily create, edit, resize, and move DDS objects graphically. Create new records, fields, and constants directly on the Design page by using the palette push buttons to the left of the Design area or from the pop-up menus. The toolbar above the Design area allows quick access to many of the editing features as well as information about the currently selected object.

Click the tab **MAIN_MENU** in the workbook.



In order to understand where MAIN_MENU came from, we need to describe the concept of a group. A group is simply a collection of one or more DDS records that make up a single screen or report. It is the set of records that is written by the application to the display or printer device at one time. Grouping records together allows you to work on one record while still seeing the related records in the background. The Workbook has a Design page tab for each group defined for quick access to each group of records.

Creating Groups from Existing Records

If you are working with existing DDS, you will want to create groups that will correspond to how the records are being used. In this example you will create a group for the next screen, where the user selects which employee in the payroll database to maintain.

To create a group:

©Copyright IBM Corporation 2003

 Scroll to the bottom of the DDS Tree and click the plus sign beside the MAIN_MENU group. The SELECT record appears as the only record in this group.



- 2. Right-click on the **MAIN_MENU** group.
- 3. From the pop-up menu, select **Insert group**. A Group Properties notebook appears and a blank Design page for the group **SCREEN1** also appears.

The Properties notebook lets you view and update the properties of the currently selected DDS object. You can open this notebook from any view, pop-up menu, or menu bar of the CODE Designer. The properties notebook is modeless. When you change an object's properties, the selected object changes immediately.

- 4. On the Group Properties notebook, click the record **EMPSEL** in the **Available** list box and click the button. For simplicity this is the only record you will add for now. The Design page now shows us what the record **EMPSEL** looks like.
- 5. Name the group by typing over SCREEN1 with EMPL_SELCT

🚞 Group prop	erties		? ×
Basics Style	s		
Group	EMPL_SEI	LCT	
A	vailable	Selected	
SELECT EMPMNT PRJSEL PRJMNT RSNSEL RSNMNT		EMPSEL	
	Þ		

6. Click the **X** in the top right corner to close the Group Properties notebook.

Well, it appears that this is one of those unusable applications where you have to know the employee number ahead of time instead of being able to browse what is in the database. What we really need is a subfile. But aren't those difficult to code, you ask? Not with CODE Designer.

Creating New Screens

You can create a new record for a screen on the Design page creating a subfile record:

- 1. Right-click the new EMPL_SELCT group in the DDS Tree.
- 2. From the pop-up menu, select **Insert group**. A Group Properties notebook appears and a blank Design page for the group **SCREEN1** also appears.
- 3. Rename the group to EMPL_LIST and close the Group Properties notebook.
- 4. You can create things on the Design page by selecting the appropriate tool from the palette on the left-hand side and then click on the Design page where you want it to be created. Right now, most things are disabled in the palette because there is no record in which to create fields. The only two tools available are <u>Create standard record</u> and <u>Create subfile record</u>. If you leave the mouse over a button for a second or two, flyover help will appear describing the indicated button.



5. Click the **Create subfile record** button and then click in the dark gray area. A subfile and a subfile control record pair are created.

Adding fields to the subfile record

Next you will add fields to the subfile record using the field creation and Design page toolbar:

Now click the Create named field button and then click somewhere on row 8. Six fields appear in a vertical column. This is because the subfile you created, currently specified a SFLPAGE (visible list size) of six.

2. Click the top field and **hold** the mouse button down and **move** it to **row 8, column 5.** Note the current row and column appear just above the field as you move it.



🔊 EMPLSTSFL 💽 🖼 🕢 🗩	OPCODE 🔽 🛛	🚟 🛛 З		
Ignore at Spaticators	<mark>∕∄∄</mark> ₿		l 🗗 🕅 Z	

- Move the mouse over to the **right edge** of the field. It turns into a double-headed arrow.
 Hold down the mouse button and **move** it to the **left**. The size of the field will be reduced. The current size will appear just above the field. When the size is **3**, **let go** of the mouse.
- 4. The toolbar at the top of the Design page is a very convenient place to monitor and manipulate the currently selected field. Rename the record from RECORD1 to EMPLSTSFL and the field from

84
lettel

FIELD1 to OPCODE by typing over the text in the drop-down list box. Click the

button and select pink to change the color of the field. Click the 💷 button to change the usage of the field to input.

5. Position the cursor at row 8, column 9. Note that the bottom right of the Designer frame shows the current cursor position If you can't see the field with the cursor position on your screen, click the Maximize button in the top right corner. You can use the cursor keys or the

6. If you are creating a long field with an exact length, the SDA syntax can be easier. Type:

+0(30)

and then press the **back arrow (not Backspace!)** to select the text you created. Notice from the Selected DDS page that you have created a text constant containing '+O(30)'.

- 7. Click **Convert string to field** button on the toolbar or press **F11** to convert the SDA syntax into a character output field of length 30.
- 8. Rename the new field to ENAME using the toolbar. This will show the name of the employee.
- 9. Position the cursor to 8, 41.
- 10. Now you will add a field for the employee's salary. Now wouldn't it be better if we could just tell the Designer what we wanted the number to look like and then have Designer generate all the cryptic EDTCDEs to make it happen? Type

\$666,666.66

and then press the back arrow.

- 11. Press **F11** to convert this field into an output numeric field with comma delimiters, two decimal positions, a currency symbol and no sign. Look at the Selected DDS page to see what was generated for you. Impressive!
- 12. Rename the field to SALARY and change its color to yellow, using the toolbar.
- 13. Our subfile seems compacted to the left. It would be better to space it out evenly. Just select

ШA

the field and click Space horizontally on the far right side of the toolbar. The other alignment buttons will align fields, left, right, center and top.

14. Just below the palette there are three spin buttons. The top one, Subfile size, specifies the total number of entries in the list that will be filled in by the application. The second one, Subfile page size, is how many entries appear on the screen. Set the Subfile size to 300 (by over typing) and the Subfile page size to 9. The Design page is updated accordingly.

Switching between Multiple Records

Now let's fix up the Subfile control record. The group you created contains 2 records. You can verify this by dropping down the record drop-down list box in the toolbar:



1. Change the current record by selecting RECORD1CTL from the drop-down list box or click or press **Alt+End**. The fields in the subfile still appear so that column heading can be

lined up, but they appear at half-intensity so that they can be distinguished from the fields of the current record.

2. Rename the record to EMPLSTCTL using the toolbar.

Let's provide a 'position to' entry in the subfile control header.

3. Position the cursor at **4**, **9** and type:

Position to:

4. Now you need an employee name field. You could create a named field with the right

characteristics like you did in the subfile, or you could create a source reference using the Ξ

button in the palette or you could reference the original database field using one of the 🕒 🔒

buttons. But there is an even simpler way. Use copy and paste! In the DDS Tree expand (click on the plus sign) the **EMPMNT** record.

- 5. Click the **ENAME** field and press **Ctrl+C**. (The pop-up menu or Edit pull down would give the Copy menu item as well).
- 6. Position the cursor to 4, 23 and press Ctrl+V. Now that was easy!
- 7. Rename the file to POS_TO.

Adding field error handling

Let's put in some error handling for the 'position to employee name' field. If the employee name is not found in the database, the program will turn on indicator 60. An indicator is a two digit variable (01, 02, 03, ... 98, 99) to signal whether or not certain events have occurred during processing. Using indicators enables program events to control screen displays. These indicators are set 'on' or 'off' in one part of the program while their status is referenced in another part of the program. By setting the indicators 'on' or 'off', you can associate fields with indicators to control whether or not a field will appear on the screen.

The screen should turn the field red, reverse image and position the cursor to it. Now wouldn't it be better if we could work something higher level and easier to remember than some arbitrary number from 1 to 99.

To set indicators:

- 1. Click on the button on the Design page toolbar (or press **F7**). The **Named indicator sets** window appears.
- 2. In the Settings name field, type: Not Found

3. Click Create.

Named in	dicato	or sets							×
				Indica	tors				
	10 11 12 13 14 15 16 17	20 21 22 23 24 25 26 27 27 20 27 20 27 20 20 20 20 21 22 23 23 24 25 25 25 26 27 27 27 27 27 28 29 29 20 20 20 20 20 20 20 20	30 - 31 - 32 - 33 - 34 - 36 - 37 - 37 -	40 41 42 43 44 45 46 47	50 51 52 53 54 55 56 57	60 61 62 63 64 65 66 67	70 71 72 73 74 75 76 77	80 81 82 83 84 85 86 87	90 91 92 93 94 95 96 97
08 18 28 38 48 58 68 78 88 98 09 19 29 39 49 59 68 79 88 98 Checked indicators are:									
Ignore all	Settings name: Not found Create Ignore all indicators All indicators are off Delete						me		
				<u>0</u> K.	App	ly 🗌	Cancel	Не	elp

4. Click the check box next to 60 and click OK. The Not found indicator set is now in effect. The Design area is shown as if indicator 60 was on and all other indicators were off. The Design page toolbar shows the current indicator set in the drop-down list box on the bottom left.

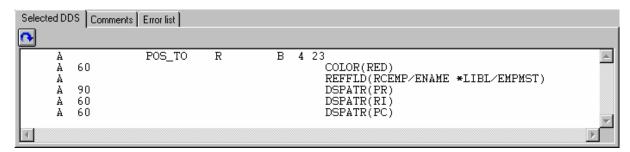
🔳 💽 E	MPLSTCTL	- 🧱
1⁄0 N	lot for d	
	Select na	amed indicator sets

- 5. Now select the **POS_TO** field.
- On the toolbar, select the color red and the display attributes reverse image and position cursor.

(The set of toolbar buttons representing the current display attributes is found just below the color button). The toolbar should look as follows:

🛐 💽 EMPLSTCTL	💌 🗺 < 🕨 (POS_1	io 🔽 🚟 [30 🚊	IOB		e seen
1/2 Not found	╶╶╶┛	i B r [0	× 🖬 🏶 🛛	6 MA	2 - 11

7. Examine the DDS generated in the Selected DDS page.



Notice that all the new keywords were created with a condition of 60. (The DSPATR(PR) was pasted in with the field originally).

8. Now let's try it out! From the **Select named indicator sets** drop-down list box, select **All indicators are off**.

💽 E	MPLSTCTL	▼ ⁴⁴⁰ / ₁₀₀ ≪ 1	POS_TO	- 🦾 🛛	30 🚍 🔳 (ob 🗖 🖉
1/0	All indicators are	off	▾◪▰ֿ	PB A 💌 🛛	0 🚍 🔆 🛛	╕╫╔╝╗┓╦
	+	1+2-	+3- <mark>-</mark>	+4-	+5	+6+
R	<mark>4</mark> P	osition to:	BBBBBBBBBB	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	

Wow! This really works!

9. From the **Select named indicator sets** drop-down list box, select **Not found**. The field appears red and reverse imaged.

Accessing field properties

Second to the direct manipulation and the toolbar on the Design page, the easiest and quickest ways of getting access to the properties of a field, record, or entire file is a Properties notebook.

The Properties notebook lets you view and update the properties of the currently selected DDS object. You can open this notebook from any view, pop-up menu, or menu bar of the CODE Designer.

The Properties notebook is modeless. When you change an object's properties, the selected object changes immediately.

You can get to a Properties notebook from the Selected menu, by pressing F4, or double-clicking on anything in the DDS Tree or the Details page or Design page.

To open the Properties notebook:

- In the DDS Tree, click the record SELECT and press F4 to see the Record properties. As you select different items, the Properties notebook will continuously update itself to show you the properties of the selected item.
- 2. Now click the ***DATE** field in the **SELECT** record. (You may have to move the Properties notebook out of the way.) This field has a different set of pages describing its properties.
- 3. Let's say we had to change the year from 2 to 4 digits. Click the **Length of year** check box.
- 4. Select **4 digits** from the drop-down list box. Notice how the sample is updated on the Properties notebook.

orting System	YYYY/MM/DD
nce Selection	HH:MM:SS
Field properties	? ×
Basics Attributes Color Editing Ind	dicators
Sample Source of date	4 digits
Position Size <u>Row Column</u>	
27X132 1 1	
Description	

5. To test the Design page, click on the **MAIN_MENU** tab in the workbook and look at the upper right corner of the screen. The date now has 4 digits.

6. Now click on the EMPAPL field in the SELECT record. On the Field Properties notebook click on the Basics tab. On this page you can change the field's name, usage, length, type, screen position. The other pages give you quick access to other properties for this field.

🚞 Field properties	? ×
Basics Attributes Color In	dicators Styles
EMPAPL Usage S IOBHPM	vpe

Adding New Keywords

"OK", you might think, "I can see where CODE Designer helps me manage the visual aspects of my displays and reports. But I have real work to do. I need access to the full power of DDS. I need to access keywords."

To add keywords:

- 1. Click **EMPAPL** field in the DDS Tree.
- 2. Press F5 or select Insert keywords from the pop-up menu.

You are taken to the Details page for the **EMPAPL** field and the **Create keywords** tab is selected in the Utilities notebook. This page shows you the subset of keywords that are allowed for the selected file, record or field and it takes into account the field's type, usage, shift and what record it is in. It is very powerful to know exactly what your options are. This information cannot be quickly ascertained from the Reference manual.

Insert Subset	: All				_
🕶 ALIAS	🗫 CHECK(ME)	🗫 CHECK(RL)	🚞 COLOR>>>	🖙 ERRMSG>>>	🗠 OVRATR
🕶 BLANKS	🗫 CHECK(MF)	🗫 CHECK(RZ)	🚧 СОМР	🗫 ERRMSGID>>>	🚧 OVRDTA
🗪 CHANGE	🗫 CHECK(M10)	🗫 CHECK(VN)	⊶ DFT	🗫 FLDCSRPRG	🚧 PUTRETAIN
🗪 CHECK(AB)	🗫 CHECK(M11)	🗫 CHECK(VNE)	⊶ DFTVAL	🗫 IGCALTTYP	🚧 RANGE
🛰 CHECK(ER)	🕶 CHECK(M10F)	🚧 CHGINPDFT	📰 DSPATR>>>	🗪 INDTXT>>>	🚧 VALUES
🕶 CHECK(FE)	🗪 CHECK(M11F)	🗪 CHKMSGID	🚾 DUP	🗪 MSGID>>>	🗫 WRDWRAP
🗠 CHECK(LC)	Section CHECK(RB)	⊶ CHRID	🗪 ENTFLDATR	🚧 NOCCSID	🕶 UNKNOWN>>>

- 3. With the **EMPAPL** Properties notebook at the **Basics** page, click the th button to change the field to numeric type. Notice that the list of keywords in the **Create keywords** page has changed.
- 4. Click the button to change the field back to alphanumeric. Notice that the list of keywords in the **Create keywords** page has changed.
- 5. Click the ALIAS keyword and press F1.

The DDS Reference help for the ALIAS keyword appears. It is important to mention that the CODE Designer has lots of on-line help. Press F1 anywhere you want to see help for an item, icon or notebook. You will get help relevant to what you are currently trying to do. From the Help menu you can get quick access to the DDS Language Reference as well as several other useful sources of information.

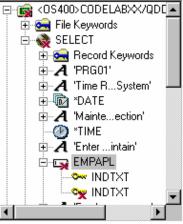
- 6. Minimize the Help window
- 7. Double-click the **INDTXT** keyword. (You may have to scroll to the right to find it). The keyword is created with default values which can be changed when you want.
- 8. Double-click on the **INDTXT** keyword again. The keyword is created with the same default values creating a conflict. You will use this error in the next section.

Verifying Your Source

You have just added a new record and some new fields to our DDS source. Everything that the CODE Designer adds to your DDS source is certain to have the correct syntax. Now you need to make sure that there are no semantic errors. You just introduced one in the last section by creating two INDTXT keywords describing the same indicator.

To verify your source:

- 1. From the **Tools** menu, select **Verify file** (or **click** on the **I** button on the main toolbar). The DDS source is checked using the same verifier that the CODE Editor or LPEX Editor uses. A message appears on the status line at the bottom of the Designer stating that the verify process completed with errors.
- In the DDS Tree, there is a trail of red x's leading to the problem. The file icon has a red x, as does the SELECT record, the EMPAPL field and finally the second INDTXT keyword.
- 3. Click the **MAIN_MENU** tab in the workbook. The **EMPAPL** field is highlighted in red.
- Click the Listing tab in the workbook. This page shows you the listing generated by the most recent program verify. A warning message is buried somewhere in the listing but it's not easy to find.



- If there are problems, they will show up in the Error list page in the Utility notebook. It behaves exactly like the Error list in the CODE Editor or LPEX Editor. Click the Error list tab.
- 6. Double-click the warning **DDS7861** in the Error list.

(Press **F1** to see detailed help on the message). The **Source** page appears and the cursor is placed exactly where the error is in the source. The **Source** page is a tokenized read-only view of the current state of the DDS source. Read-only? Wouldn't it be great if you could just clear the error right here. There are some things that are just plain faster in the editor and many others that are faster in the visual environment. It would be great to switch between the two modes at the push of a button. Well, let's just do that.

Details MAIN_MENU EN	IPL_SELCT EMPL_LIS	T So	urce	Li	sting		
	·						
AAN01N02N03.					Functions++++++++++++++++++++++++++++++++Co		
A					you want to maintain' 🗾 🔺		
A	EMPAPL	1	В	- 9	25INDTXT(01 'Default')		
A					INDTXT(01 'Default')		
DDS7861W Text already assigned to indicator.							
A				- 9	28'Employee Master Maintenance'		
A	PRJAPL	18	В	10	25		
A					28'Project Master Maintenance'		
A	RSNAPL	18	В	11			
A				11	28'Reason Code Master Maintenance'		
•							
Selected DDS Comments	Error list						
COS400>CODELABXX/QDDSSRC(MSTDSP)							
DDS7861W Text already assigned to indicator.							

Switching between Designing and Editing your screen

To switch between the Design mode and the Edit mode:

- 1. Click the button or select **File-> Edit DDS Source** from the Editor menu. You now have access to the full power of the editor.
- 2. Explore the Edit and View pull downs.
- 3. Press Ctrl-F to bring up the Find/Replace window
- 4. Type in INDTXT and click **Find**.

Editor - Find and Replace	×
Find INDTXT	•
Replace with destroy	
Replace this occurrence	☑ Case sensitive
C Replace, then find <u>n</u> ext	Pattern match
O Replace <u>all occurrences</u>	₩rap
<u>Find</u> <u>R</u> eplace <u>O</u> ptions	Cancel Help

- 5. Press **Ctrl-N** to find the next occurrence.
- 6. Delete the second INDTXT line.

	Autor View Insert Undo Cut Cut Copy Paste Delete Duplicatg Select all Deselect all Find up Find next Find up Find up Find selection Block Locate Set guick mark	Selected Id Ctrl+Z Ctrl+X Ctrl+C Ctrl+V Ctrl+Del Ctrl+Del Ctrl+A	Replace	n D +TD 1 1A 1A Alt+L Alt+B	pBI B B	10 11 11 11 11	PosFunctions************************************
943 9449 9459 Error li	0 A	Ctrl+F4 INTAINED R EP	Move Delete Copy C(MSTDSP)	Alt+M Alt+D Alt+C			DN OF THE EMPLOYEE TO BE ENANCE TO BE PERFORMED
The ve	DDS7861W Te		signed to indicator. DSSRC(MSTDSP)' (complete	ed w	ith er	rors.

Compiling Your Source

Now you will compile the source on the iSeries just as you did in the Remote Systems LPEX Editor:

- 1. From the **File** menu, select **Save** to save your source to the iSeries.
- 2. From the **Tools** menu select **Compile** and then select **No prompt** (or **click** the **button** on the main toolbar).
- 3. A message will tell you when the compile is complete. Click **OK** in the Message window.

If you run the PAYROLL program, you will see the 4 digit year change you made to the opening screen of the program.

Closing CODE Designer

From the File menu, select Exit.

Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. There is a graphical design tool that let's you design your screens and reports visually and then generate the DDS source for you. This tool is named:
 - A. Page Designer
 - B. CODE Designer
 - C. Screen Design Aid
 - D. None of the above
- 2. The DDS tree view presents:
 - A. The loaded DDS source as a tree structure
 - B. A hierarchy of files, records, field, help specifications, keys and keywords in each selected DDS object
 - C. Groups of records, which represent the screens or reports you are designing, as peers of the file in the tree hierarchy
 - D. All of the above
- 3. Match the item with its correct description.
 - A. Workbook
 - B. Details page
 - C. Utility notebook
 - D. Design page
 - E. Properties notebook
 - F. Group
 - G. Indicator
 - A. View specific elements of the DDS source such as errors and selected objects
 - B. Lists the contents of the currently selected item in the DDS tree
 - C. Contains several different tabbed pages
 - D. For visually designing screens and reports
 - E. A collection of one or more DDS records that make up a single screen or report
 - F. Access properties of a field, record, or entire file
 - G. Two digit variable to signal whether or not certain events have occurred during processing
- 4. The Utility notebook contains:
 - A. Selected DDS page
 - B. Error List page
 - C. Create keywords page
 - D. Comments page
 - E. All of the above
- 5. There are two modes in the Details page: Details and List. The Details mode shows more about each DDS object while the List mode shows more DDS objects in a page. (T, F)
- 6. On the Design page, you can easily:
 - A. Create DDS objects
 - B. Edit DDS objects
 - C. Resize DDS objects
 - D. Move DDS objects
 - E. Create records
 - F. Create fields
 - G. Create constants
 - H. All of the above

- 7. Grouping records together allows you to work on one record while still seeing the related records in the background. (T, F)
- 8. The Properties notebook is modeless. When you change an object's properties, the selected object changes immediately. (T, F)
- 9. The DDS source is checked using the same verifier that the LPEX Editor or CODE Editor uses. (T, F)
- 10. You can switch between design mode and edit mode in CODE Designer. (T, F)
- 11. You can compile sources on the iSeries just as you in the LPEX Editor. (T, F)

Practice

Given your experience in this exercise using the CODE Designer from the Remote Systems view, try creating a printer report. You can use the physical file specification REFMST in member QDDSSRC in library RSELABxx. Take time to explore the fields and information for this physical file. You may want to refer back to this information as you create your report. When you are familiar with the file REFMST, you can create your printer file. Use the Development Studio Client for iSeries online help to assist you in this task.

What you just did

In this exercise you: You used CODE Designer to modify a display file to add a screen. You learned how to add groups, records and fields. You saved your DDS and then verified it for compile errors. When you had no compile errors you then compiled your DDS.

In the next exercise you will learn how to debug and test an application running on an iSeries system.

Exercise 7: Debug an ILE RPG Program

The goal of this exercise is to review the Debugger features. You then start the debugger, set breakpoints, monitored variables, run and step into a program, view the call stack in the Debug view, remove a breakpoint, add a storage monitor, and set watch breakpoints and all from the Debug perspective.

At the end of the exercise, you should be able to:

- Describe some of the features of the Debugger
- Start a Debug session
- Add and delete line breakpoints
- Display a variable
- Change a variable
- View the call stack
- Run, Step, Step into, Step Over a program
- Switch between Source and Listing view
- Add a storage monitor
- Add a watch breakpoint

Debugging iSeries programs from the Remote Systems view

The debugger is a source-level debugger that allows you to debug and test an application running on an iSeries system from your workstation. It provides a functionally rich interactive graphical interface that allows you to:

- View source code or compiler listings, while the program is running on the iSeries system.
- Set, change, delete, enable and disable line breakpoints in the application program. You can easily manage all your breakpoints using the Breakpoints list.
- Set watch breakpoints to make the program stop whenever a specified variable changes.
- View the call stack of your program in the Stacks page. As you debug, the call stack gets updated dynamically. You can view the source of any debug program by clicking on its call stack entry.
- Step through your code one line at a time.
- Step into or step over program calls and ILE procedure calls.
- Display a variable and its value in the Monitors page. The value can easily be changed to see the effect on the program's execution.
- Locate procedure calls in a large program quickly and easily using the Programs page.

- Debug multithreaded applications, maintaining separate Stacks for each thread with the ability to enable and disable any individual thread.
- Load source from the workstation instead of the iSeries useful if you don't want the source code on a production machine.
- Debug client/server and distributed applications.

The Debugger supports Java as well as RPG/400 and ILE RPG, COBOL and ILE COBOL, C, C++ and CL.

In the following exercise you will be given the opportunity to learn about some of the basic features of the Debugger. For the purpose of this exercise you will debug an ILE RPG/400 program. Don't worry if you don't know RPG.

Starting the Distributed Debugger

You will be working with the ILE RPG program **PAYROLLG**.

Note: PAYROLLG is the same RPG program as PAYROLL but without compile errors. You are using it instead of PAYROLL in this exercise, to accommodate anyone who decided to skip right to this exercise without completing the editor exercise.

You can start the Debugger in several ways: direct from the Remote Systems view, or from the Launch Configurations window. Starting direct from the preview doesn't allow you to specify parameters to be passed to the program. The Launch Configurations window allows you to modify how the program is invoked and includes specifying parameters. To make the exercise interesting you will use CL program CLR1 to call PAYROLLG and you will pass one parameter to CLR1. This means you will use the Launch Configurations window.

Use the Remote System Explorer to start the Launch Configurations:

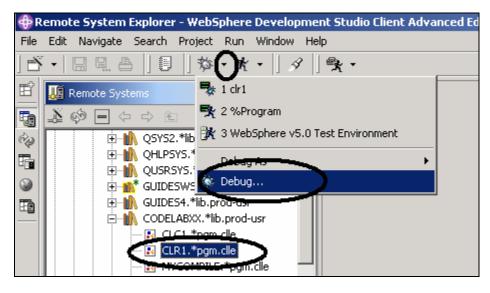


Figure 57: Start the Launch Configurations window

In the Remote Systems view:

- 1. Expand the Library list filter, if it isn't expanded already
- 2. Expand library RSELABxx, if it isn't expanded already
- 3. Select program CRL1 in library RSELABxx

On the workbench menu:

- 4. Click the menu drop-down arrow beside the **DEBUG** icon as shown in Figure 57
- 5. Select **Debug** from the drop-down menu

The Launch Configurations window appears:

©Copyright IBM Corporation 2003

BLaunch Configurations
Create, manage, and run launch configurations
Select a type of configuration to create, and press the 'New' button
aunch Configurations:
Attach to a Running Proc Debug a Compiled Applic Series Bobag Batch Apr Series: Debug Interactiv
 iSeries: Debug Multi-Thre Java Application Java Application in iSerie Java Bean / Applet Junit
Remote Java Application Run-time Workbench Server WebSphere v5.0 Tes Stored Procedure Debug
WebSphere Application S WebSphere v5 Applicatic

Ne <u>w</u> Delete
TMDSPC *ETLE PE-SPC

Figure 58: Launch Configurations window

In the tree view:

- 1. Select iSeries: Debug Interactive Application
- 2. Click New.

The Debug Interactive Application pane appears on the right side of the window:

Haunch Configurations	×
Create, manage, and run laun	ch configurations
Launch Configurations:	Name CLR1
Attach to a Running Proc Debug a Compiled Applic. Debug a Compiled Applic. Series: Debug Batch App Welspress Mcusingnew Milabxx CLR1 My iSeries program Cusingnew New_configuration Wilabxx1 Cusingnew New_configuration Wilabxx2 Cusingnew New_configuration Wilabxx2 Claus Series: Debug Multi-Thre Wilabxx2 Claus Java Application Java Application Java Application Junit Remote Java Application Claus Junit Series: Debug Multi-Thre Wilabxx2 Claus Stored Procedure Debug WebSphere v5.0 Tes Stored Procedure Debug WebSphere v4 Application WebSphere v5 Applicatic	What To Debug How To Start Advanced S Common Connection: 4003 Vew Program or Service Program Library: RSELAB01 V. Browse CLR1 V. Browse Service Program: V. Browse Step into Step into Update production file:
New Delete	Apply Reyert
	Deb <u>ug</u> Close

Figure 59: Debug Interactive Application

- 3. Enter the program name CLR1 in the Name field
- 4. Select the **Update production files** check box

©Copyright IBM Corporation 2003

5. Click the How To Start tab

On the How To Start page:

Name: CLR1
What To Debug How To Start Advanced 🕉 Common
what to bebug them to start Advanced As Common 1
Command to start application:
CALL PGM(RSELABxx/CLR1)
-
Prompt

Figure 60: How to Start page

6. Click Prompt.

The Prompt window opens.

👹 Call Program (CAI	LL) 🗡
<u>File Edit View H</u>	<u>f</u> elb
Program:	CLR1 Name
Library:	RSELABxx 🔄 Name
Parameters:	
	Add
	Remove
	Move up
	Move down
	OK Cancel Help ?

Figure 61: Prompt window

- 7. Enter 'xx', xx being your team number in the Parameters field
- 8. Click Add.

The parameter value will appear in the lower list box.

9. Click OK.

The complete start command for the program appears in the window:

©Copyright IBM Corporation 2003

Launch Configurations:		
	Name: CLR1	
Attach to a Running Proc		
Debug a Compiled Applic	What To Debug How To Start Advanced 🕸 Common	
🖃 👼 iSeries: Debug Batch App	what to bebug them to state Advanced Age Common	
getrecord		
Mcusingnew	Command to start application:	
wflabxx	CALL PGM(RSELABXX/CLR1) PARM('XX')	Л
📄 🎭 iSeries: Debug Interactiv		
My iSeries program		
iSeries: Debug Job		
cusingnew		
New_configuration		
wflabxx1		
🖻 🎲 iSeries: Debug Multi-Thre		
wflabxx2		-1
🖻 📲 Java Application		
≣J Claus		
Java Application in iSerie	Prompt	
🗐 🗐 🦼 Java Bean / Applet		
Junit		
Remote Java Application		
Run-time Workbench		
Server		
WebSphere v5.0 Tes		
Stored Procedure Debug		
WebSphere Application S		
WebSphere v4 Application 5		
WebSphere v5 Applicatic		
Ne <u>w</u> Dele <u>t</u> e	Apply Revert	
	\sim	

Figure 62: Ready to debug

10. Click Debug

If an error message like this appears:



Figure 63: Error message if interactive session doesn't exist

The Remote System Explorer server has been shut down in the meantime. Go to your 5250 emulator and restart the Remote System Explorer server following the instructions in the message.

Cancel the message in the Debug perspective.

Now the Debug perspective is loaded in the workbench:

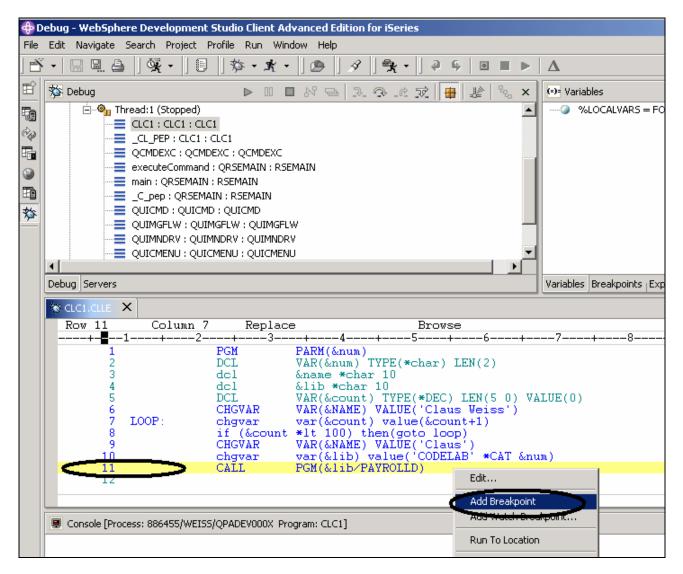


Figure 64: Debug perspective, adding breakpoint

Now that the program is active on the iSeries and stopped at the first executable statement, the debugger displays the source.

Setting Breakpoints

You can only set breakpoints at executable lines. All executables lines are displayed in blue.

The easiest way to set a breakpoint is to right-click on the line in the **Source** pane: 1. Right-click anywhere on **line 11** 2. Select Add breakpoint from the pop-up menu.

To add a conditional breakpoint in the loop when it loops the 99th time:

🔶 D	ebu	ıg - WebSph	here Dev	velopment :	Studio Client A	vanced Edition for iSeries				
File	Edi	it Navigate	Search	Project P	Profile Run Wi	low Help				
] 🖄	•		- ÿ	ŧ • 🗊	♪ か ま] @ ୬ % • ₽ ⊊ I		• <u>A</u>		
Ē	鶖	Debug			▶ 00	u	°c, >	× 💌 Vari	ables	
Bi ⊗ Ei ⊗ Ei 🕉	• Del		CLC1 _CL_F QCME execu main : _C_pe QUICI QUIM QUIM QUIM	teCommand QRSEMAIN ep : QRSEMA MD : QUICME GFLW : QUIM NDRV : QUIM	CLC1 (EXC : QCMDEXC : QRSEMAIN : RS : RSEMAIN IN : RSEMAIN	V,		Variables	Breakpoints	= FOR RPG, (
	*	CLC1.CLLE	×						\sim	
	-	Row 8	0	olumn 24	4 Replac					
	-		-1	+2		-+4+5+	6+	7	-+8	+
		1 2 3 4 5 6 7	LOOF) <u>; </u>	PGM DCL dcl DCL CHGVAR chgvar	PARM(#) VAR(#) TYPE(*char) LEN(&name *char 10 &lib *char 10 VAR(&count) TYPE(*DEC) LEN VAR(&NAME) VALUE('Claus We var(&count) value(&count+1 *lt 100) then(goto loop)	(5 0) siss')	VALUE(0)		
	~	9	_		LI (&COUNT	*It IUU) then(goto loop) VAR(@NAME) VALUE(Claus)				
	~	10 11 12			chgvar CALL	var(&lib) value('CODELAB' FGM(&lib/PAYROLLD)	*CAT &	num)		

Figure 65: Adding a break point and selecting the breakpoint view

- 3. Right-click on line 8
- 4. Select Add breakpoint from the pop-up menu.
- 5. Select the **Breakpoints** tab in the upper right pane of the workbench as shown in Figure 65 In the **Breakpoints** view:

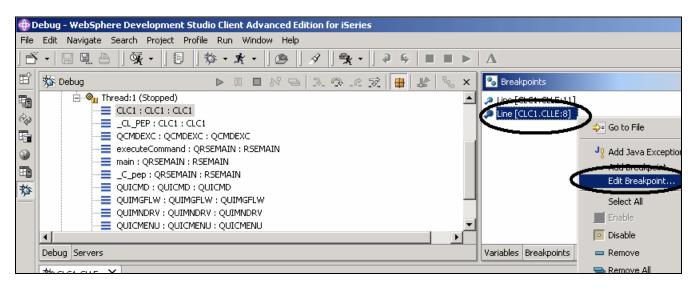


Figure 66: Edit breakpoint in Breakpoint view

- 6. Right-click line 8 breakpoint
- 7. Click **Next** in the **Edit a line breakpoint** window

You only want to stop in the loop when it executes for the 99th time or more. You can do that by setting the **From** field of the Frequency group to 99.

Edit a Line Breakpoi	nt			
Optional paramete Make the breakpoint		n the followin	g parameters	兹
T <u>h</u> read: Every	•			
Frequency Fron 99 To: Infinity Every: 1				
Expression:				
	< <u>B</u> ack	Next >	<u>F</u> inish	Cancel

Figure 67: Specify condition for breaking

- 8. Enter 99 in the From entry field
- 9. Click Finish.

Monitoring Variables

You can monitor variables in the Monitors view. Now you will monitor the variable &COUNT.

In the Source pane:

©Copyright IBM Corporation 2003

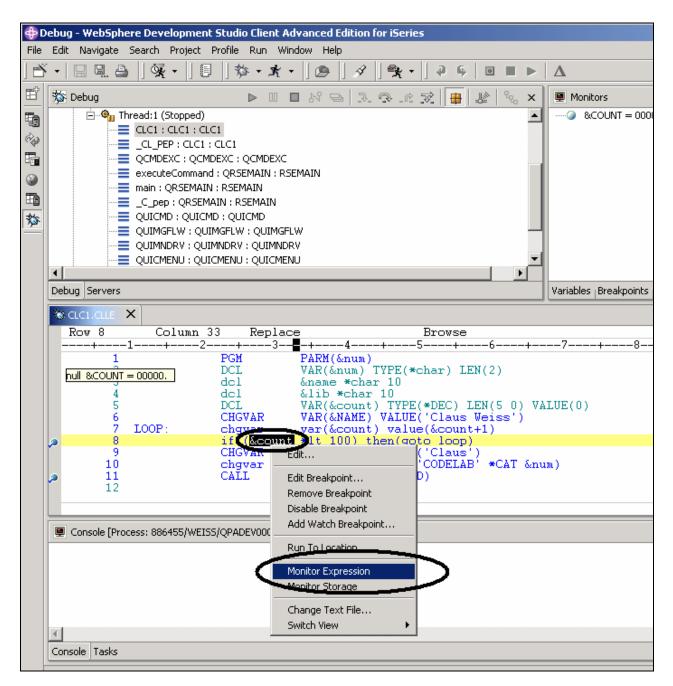


Figure 68: Select variable to be monitored

- 1. Double-click on the variable &COUNT
- 2. Right click &COUNT

3. Select Monitor Expression from the pop-up menu.

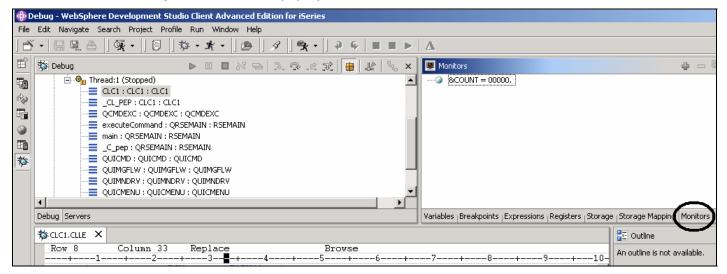


Figure 69: Monitor view with variable & COUNT

The variable appears in the Monitors pane. Its current value is zero.

Tip: If you quickly want to see the value of a variable without adding it to the Monitor, leaving the

mouse pointer on a variable for a second or so will display its value in a pop-up window.

Now that some breakpoints are set, you can start to run the application:

⊕D	ebug	- WebS	pher	e Deve	lopme	ent Stu	dio Cli	ent Aq	lvance	ed Editio	n for i	Serie	25			
File	Edit	Naviga	te Se	earch	Project	t Profi	le Rur	n Win	idow	Help						
ÌÈ	-	8	æ	9	•]	0]	静,	×.] 🛽	▶] ⊅	2	₹ •		p (6		$\blacktriangleright $
Ē	🔯 C	ebug						00 0	- 14		1.9	_¢	r		⁹ 20	x

Figure 70: Run icon on Debug toolbar

4. Click **Run** I from the Debug toolbar.

The program starts running and stops at the breakpoint at line 8. (Be patient, the Debugger has to stop 98 times but because of the condition continues to run until the 99th time.) Notice in the Monitors view, that &COUNT now has the value 99.

5. Click Run again.

The program stops at the breakpoint at line 8 again and &COUNT now has the value 100.

6. Click **Run** once more so that the program runs to the breakpoint at line 11.

Stepping into a Program

The Debugger allows you to step over 🔄 a program call or step into it 🗷 . When you step over a program call, the called program runs and the Debugger stops at the next executable statement in the calling program. You are going to step into (Step into) the Payroll program.

⊕ D	ebug	- WebS	iphe	re Deve	elopmen	t Studio	o Clier	nt Adv	ance/	d Edition	for iSe	ries	
File	Edit	Naviga	te	Search	Project	Profile	Run	Snap	Shot	Window	Help		
B	•		₿	j 🕵	•] []] 🛛	2	월 -	×	•] @] \$	* , -	4
Ê	🏂 (Debug							24	(7	9r	• .	° ₂₀ ×

Figure 71: Debug toolbar with Step Debug icon

1. Click *Step into* on the Debug toolbar. The source of PAYROLLG is displayed.

Depending on the option you used to compile the program (*SRCDBG or *LSTDBG for RPG, or *SOURCE, *LIST, or *ALL for ILE RPG), this window displays either the Source or Listing View.

🕀 De	bug - WebSphere Development !	Studio Client Advanced Edition for iSeries
File	Edit Navigate Search Project P	rofile Run Window Help
È	• 🔛 🗳 🗍 🕵 • 🗍 🕖	弥・★・ @ メ 喚・ ♀ ♀ 回 ■ ►
I L	🏂 Debug	× % \$\$ \$ \$ 10 € \$ \$ \$ \$ \$ \$
- - -	CLR1 [iSeries: Debug Interactive	
- (* E	PAYROLLG : PAYRO ODND DED DAYR	
 	CLR1 : CL	
	QCMDEXC : QCMD executeCommand main : QRSEMAIN	: QRSEMAIN : RSEMAIN
		IN : RSEMAIN
	QUIMGFLW : QUIM QUIMNDRV : QUIM	IGFLW : QUIMGFLW INDRV : QUIMNDRV
	The PAYROLLG RPGLE X	
	Row 7 Column 2	
		+3+4+5+6+- **********************
	2 F* PROC	GRAM NAME - RPGPGM CRIPTION - Time reporting master file mainten externally described workstation p
		ICATORS USED
	Edit	No record found on CHAIN operation General error condition Protect display on delete request
	Add Breakpoint Add Watch Breakpoint	End of job requested Return to application selection Return to employee selection
	Run To Location	Return to project selection Return to reason code selection
	Monitor Expression Monitor Storage	Last record
	C Change Text File	Show *SOURCE
	Console Tasks	

Figure 72: RPG source view

- 2. Right-click anywhere in the **Source view**
- 3. Select **Switch view > Show *LISTING** from the pop-up menu.

Row 47	Column 1 Re	place		Br	owse						
1-	+2+	-3+	4	-+5	+	6-	+	7+	8	+9-	+1
47	46 C*										
48	47=ISELECT										
49	48=I			A	1	1	PRJAPL				
50	49=I			A	2	2	RSNAPL				
51	50=I			A	3	3	EMPAPL				
52	51=IEMPSEL										
53	52=I			S	1	6	OEMPNO				EMPLO
54	53=I			A	7	- 7	ACODE				
55	54=IEMPMNT										
56	55=I			A	1	30	ENAME				EMPLO
57	56=I			A	31	31	EMCAT				EMPLO
58	57=I			A	32	36	EDEPT				EMPLO
59	58=I			A	37	66	ELOCN				EMPLO
60	59=I			A	67	74	EUSRI				EMPLO
.1											

Figure 73: Expanded Input specs for exxteranl described files

4. Page down in the source and take a look at the expanded file descriptions.

You don't have any /Copy member in our PAYROLL program but these would also be shown in a Listing view. Switch back to the Source view.

5. Select Switch view > Show *Source

Listing Call Stack Entries

The Debug view in the upper left pane, lists all call stack entries. It contains a tree view for each thread. The thread can be expanded to show every program, module, procedure and method that is on the stack at the current execution point. If you double click on a stack entry you will display the corresponding source if it is available. Otherwise the message No Debug data available appears in the Source view.

In the Debug view, expand the stack entry of Thread1 if it is not expanded already, then click the plus sign in front of it.

de Da	ebug - WebSphere Development Studio Client Advanced Edition for iSeries
-	Edit Navigate Search Project Profile Run Window Help
🖄	- L L A
11 11 12 12	Image: State: State
	PAYROLLG : PAYROLLG : PAYROLLG
	Debug Servers
	Row 1 Column 1 Replace Browse +12345

Figure 74: Call Stack in Debug view

It allows you to work with and switch between different programs and/or ILE modules.

Setting Breakpoints in PAYROLLG

Now you add some breakpoints in PAYROLLG.

To add breakpoints:

- 1. Right-click line 57
- 2. Select Add Breakpoint from the pop-up menu.

A blue breakpoint icon is added to the prefix area of this line to indicate that a breakpoint is set.

- 3. Repeat the above step for line **58**.
- 4. Repeat the above step for line **87**

To view all breakpoints, select the **Breakpoints** tab from the top left pane. This view shows all breakpoints currently set in your programs.

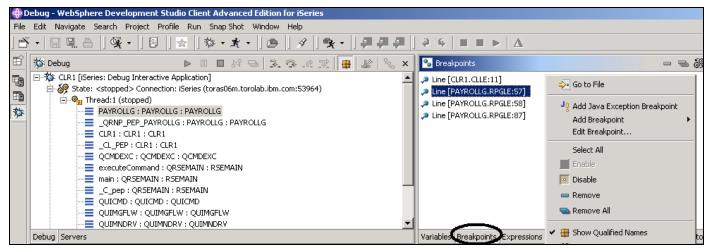


Figure 75: Breakpoints view

This is a convenient place to work with breakpoints. You can delete, disable/enable, or edit a breakpoint. These tasks are available from the pop-up menu when you right mouse click in the view area. Double click any entry to show the source where the breakpoint is set.

Removing a Breakpoint

It is also easy to remove breakpoints from the **Source** pane:

- 1. Right click line 58
- 2. Select Remove Breakpoint

The blue icon is removed from the prefix area indicating that no breakpoint is set on that line.

©Copyright IBM Corporation 2003

<u>:</u>] Welcome	S PAYROLLG.RPGLE	×							
	Row 58	Column 6	Replace		Bi	rov	/se			
	+=:	1+2	+3	+4	+5		-+6	+	'	78-
	54	С		EVAL	*IN60	=	*OFF			
	55	С		EVAL	EMESS	=	*BLANK			
	56	С		EVAL	EMPAPL		*BLANK			
,,	57	С		EVAL	PRJAPL		*BLANK			
۵.	58	C		EVAT.	RSNAPT.	=	*BLANK			
	59	C*		Edit						
	60		the SELEC				It end	ot	JOP	requested,
	61	C*		Edit Broo						
	62 63	C* C	0	Remove B	Breakpoint		•			
	64	č		Disable D	calipoint					
	65	C *IN	xc 👘		ch Breakpoint					
	66	Č Č		Haa Waa	an broatpoint					
	67	č		Run To Lo	ocation					
	•			Monitor E	xpression					
				Monitor S	itorage					

Figure 76: Remove breakpoint

Now you are ready to run the PAYROLLG program:

- 3. Click **Run b** from the Debug toolbar.
- 4. The program starts running and runs to the breakpoint at line **57**.
- 5. Click Run again.

The program waits for input from the 5250-emulation session.

- 6. Type an x beside the Project Master Maintenance option
- 7. Press Enter in the emulation session.

The program runs to the breakpoint at line 87.

Monitoring Variables in PAYROLLG

Now lets monitor variables and change them in PAYROLLG.

In the Source pane:

- 1. Double-click the variable EMPAPL on line ${\bf 88}$
- 2. Right click the variable
- 3. Select Monitor Expression

:) Welcome	The PAYROLLG.RP	GLE X				
	Row 88	Column 5	2 Replace		Brows	;e	
	+	-12-			-+5	6+-	7
	81	C* If	PRJAPL (pro	ject maint	enance) equa	ds X, the r	program
	82		anches to la				
	83		the prior t				
	84	C* re	ason code ma	intenance.	The progra	un will cont	inue w
	85		xt executabl	e operatio)n .		
	86	C*					
۲	87	C		SELECT			
	88	C		when	EMPAPE '	X'	
	89	ç		EXSR	Edit		
	90	C*			-		
	91	ç		when	🕴 🛛 Add Breakp	oint	
	92 93	Č.		EXSR	Add Watch	Breakpoint	
	93	C*		when		broarpointern	
	24			when	Run T <u>o Loc</u>	ation.	
	•						
					Monitor Exp	pression	
_							

Figure 77: Add variable to monitor

4. Click the Monitors tab in the upper right pane

The variable appears in the **Monitors** view. Its value is blank because you did not select the

Employee Master Maintenance option.

5. In the same way add the variables PRJAPL on line 91 and RSCDE on line 113 to the monitor. Variable PRJAPL equals 'x' because you did select the Project Master Maintenance option. In the Monitors view:

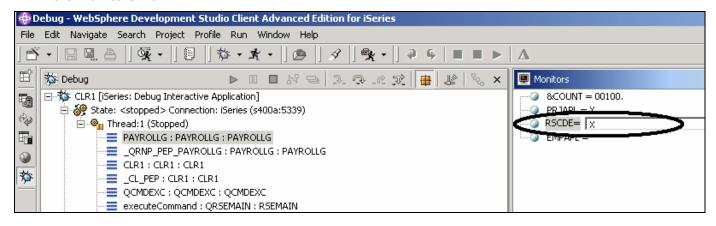


Figure 78: Change variable content

- Double-click the variable RSCDE.
 The value changes into an entry field.
- 7. Type in the new value x for the variable.
- 8. Click another variable to leave the field.

The variable is successfully changed.

Adding a Storage Monitor

Adding a storage monitor for a variable allows you to view the storage starting with the address where the variable is located. You can choose between two display formats: hexadecimal and character or character only.

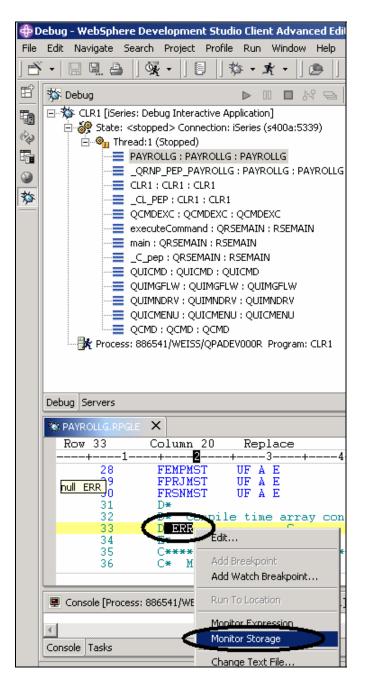


Figure 79: Adding a Storage Monitor

In the Source window:

- 1. Double click the variable ERR in line 33
- 2. Right click and select Storage Monitor from the pop-up menu

-	ebug - WebSphere Development Studio Client Advanced Edition for iSeries Edit Navigate Search Project Profile Run Window Help							<u> </u>	
ÌÈ	·· 〗〗▲│嗓· 目│弥·★· ♪ // ♪ ♪+ ● ==	Δ							
Ē	🐝 Debug 🕒 🕒 🕹 🔍 🖷 🔊 🗞	< 🚺 St	orage					🔒 🗙	
1	Science (Interactive Application)	ERR :	ERR : Thread 1						
È	Erein State: <stopped> Connection: iSeries (s400a:5339)</stopped>		0 - 3	4 - 7	8 – B	C - F	Text		
	PAYROLLG : PAYROLLG : PAYROLLG	0	C47F4040	40404040	40C1C4C4	40D9C5D8	D" ADD REQ		
3		0	E4C5E2E3	C5C440C2	E4E340D9	C5C3D6D9	UESTED BUT RECOR		
		O	C440C1D3	D9C5C1C4	E840C5E7	C9E2E3E2	D ALREADY EXISTS		
棼	CL_PEP : CLR1 : CLR1	O	40C9D540	C6C9D3C5	40404040	4040E6C1	IN FILE WA		
		0	D9D5C9D5	C7406040		D9C440E6	RNING - RECORD W		
	executeCommand : QRSEMAIN : RSEMAIN main : QRSEMAIN : RSEMAIN	0	C1E240D7	D9C5E5C9			AS DEFUIOUSLY DE		
		0	D3C5E3C5	C4404040	Monitor E>	pression	C		
		D	C8C1D5C7	C540D9C5	📼 Remove S	torage Monitor	QUESTED		
		0	C2E4E340	D9C5C3D6	Reset Tab	le to Monitored A	RD DOES		
	CUIMGELW : QUIMGELW : QUIMGELW QUIMNDRV : QUIMNDRV QUIMNDRV : QUIMNDRV	D	D5D6E340	C5E7C9E2	Content Style		T		
		0	404040C3	C8C1D5C7	Content Style	E REQUES			
		0	E3C5C440	C2E4E340	🚔 Copy View	to Clipboard	RECORD I		
	Process: 886541/WEISS/QPADEV000R Program: CLR1	0	E240C4C5	D3C5E3C5			D		
			404040C4	C5D3C5E3	🚔 Print View		E REQUES		
		D	E3C5C440	C2E4E340	D9C5C3D6	D9C440C4	TED BUT RECORD D	العر	

A new page is added to the Storage view. The tab shows the name of the variable.

Figure 80: Change storage monitor

- 3. Use the scroll bar on the right of the Storage view to scroll down. You can see the content of the memory at this moment
- 4. Right-click in the view area
- 5. Select **Reset Table to Monitored Address** to get back to the starting address.
- 6. Right click the view area
- 7. Select Remove Storage Monitor to remove the storage monitor.

Setting Watch Breakpoints

A watch breakpoint provides a notification to a user when a variable changes. It will suspend the execution of the program until an action is taken.

In the Source view:

- 1. Go to line 116.
- 2. Double-click variable *IN60 to highlight it.
- 3. Select Add Watch Breakpoint.

The **Watch Breakpoint** window appears. The **Expression** field is pre-filled with the highlighted variable *IN60.

Add a Watch Breakpoint		
Required information	xecution when data changes at a specific address	夵
Address or expression: Number of bytes to watch:	*IN60 0	V
	< Back Next > Einish C	ancel

Figure 81: Add Watch Breakpoint

4. Click Finish.

The watch breakpoint is now set.

5. Click **Run** on the Debug toolbar.

The application waits for input from the 5250-emulation session.

In the 5250-emulation session:

- 6. Type 123 for **Project Code** and D (for delete) in the **Action Code** field.
- 7. Press Enter.

A message is displayed indicating that the variable ***IN60** has changed.

🕀 Debugger Message	×
Variable *IN60 has changed at statement PAYROLLG in program module QPADEV000R/WEISS /886541, which is running in job	
	OK

Figure 82: Watch breakpoint message

8. Click OK

The program stops at line **465**. This line is located immediately after the statement which caused the variable *IN60 to change.

Closing the Debug Session

1. Click **Run** on the Debug toolbar.

The application waits for input from the 5250-emulation session.

- 2. Switch to the 5250 emulation session.
- 3. Press F3 to end the job.

A message **Debug session terminated** comes up.

4. Click OK

Close the Debug perspective by performing the following steps:



Figure 83: Close debug perspective

- 5. Right click the Debug icon is on the left hand frame in the workbench
- 6. Select Close from the pop-up menu

©Copyright IBM Corporation 2003

Complete the checkpoint below to determine if you are ready to move on to the next exercise.

Checkpoint

- 1. You can start the debugger:
 - A. From the Remote Systems view
 - B. Launch Configurations window
 - C. Both
- 2. You can only set breakpoints at executable lines. (T, F)
- 3. The easiest way to set a breakpoint is to:
 - A. Right-click on the line
 - B. Right-click after the line
 - C. Right-click before the line
 - D. All of the above
- 4. You can change variables and indicators in the:
 - A. From the Remote Systems view
 - B. Debug view
 - C. Monitors view
 - D. Storage monitor
 - E. All of the above
- 5. The debugger allows you to:
 - A. Step over a program call
 - B. Step into a program call
 - C. Both
- 6. The Debug view lists all call stack entries. It contains a tree view for each thread. (T, F)
- 7. You can perform which actions on breakpoints:
 - A. Delete
 - B. Add
 - C. Disable
 - D. Enable
 - E. Edit
 - F. All of the above
- 8. Adding a storage monitor for a variable allows you to view the storage starting with the address where the variable is located. (T, F)
- 9. The Storage monitor supports these display formats:
 - A. Hexadecimal and character
 - B. Character only
 - C. Decimal
 - D. All of the above
 - E. A and B
- 10. A ______ breakpoint provides a notification to a user when a variable changes. It will suspend the execution of the program until an action is taken. You can start the debugger:
 - A. Watch
 - B. Support

- C. Java Exception
- D. Type

Practice

Given your experience in working with the debugger features, in your own source, try setting, changing, deleting, enabling, disabling line breakpoints, setting watch breakpoints, displaying and changing variables and viewing the call stack as you debug. Use the Development Studio Client for iSeries help to assist you in these tasks.

What you just did

In this exercise you learned about the Debugger features. You then started the debugger, set breakpoints, monitored variables, ran and stepped into a program, viewed the call stack in the Debug view, removed a breakpoint, added a storage monitor, and set watch breakpoints and all from the Debug perspective.

In the next exercise you will learn how to create filters and actions to manage your iSeries objects all from the Remote Systems Explorer. In short, you'll see how Remote System Explorer can organize and integrate your work and make that work easier.

Exercise 8: Exploring Remote System Explorer

In this exercise you will use the Remote System Explorer perspective to work with the iSeries objects that you used in the previous exercise. You will also see how easy it is to define filters, perform actions and define your own actions. In short, you'll see how Remote System Explorer can organize and integrate your work and make that work easier.

At the end of the exercise, you should be able to:

- Describe Remote System Explorer
- Describe Remote System Explorer, filters, user actions and running commands
- Open Remote System Explorer
- Create filters (library, object)
- Change the library filter
- Create a user-defined action
- View properties
- Run commands from iSeries Table view

Introducing the Remote System Explorer

Most of the function of the CODE Project Organizer has been replaced by WebSphere Studio function with the exception of accessing ADM parts.

The Remote System Explorer is replacing PDM (Program Development Manager) on the workstation. It currently doesn't have all the function of PDM but will overtime be a full replacement for PDM.

Remote System Explorer allows you to:

- Simplify your work by giving you quick access to lists of iSeries libraries, objects, members, IFS files, UNIX files, and local files.
- Use the context-sensitive pop-up menus on these lists to perform actions such as start the LPEX Editor, CODE Designer, or Distributed Debugger or other common iSeries actions.
- Use the **Work with User Actions** option to create and manage your own user-defined actions and have them appear in the pop-up menus.
- Use the command support to increase your productivity by allowing you to enter and repeat iSeries or local commands without switching to an emulator session.

Creating a library filter

In the Remote System Explorer perspective, you now want to work with specific iSeries objects.

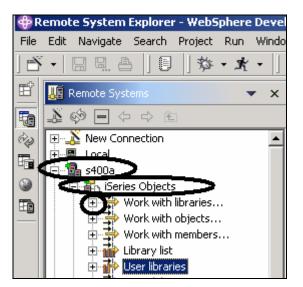


Figure 84: Expand Work with Libraries

In the previous exercises you have worked with the Library list filter. Now you will create your own library filter. Library filters list a set of libraries from your iSeries system in the Remote Systems view. But first let's understand what filters are all about.

As a user, you want a flexible user interface that allows for integration between systems. As a developer, you often need to create libraries, source files, and source members. The Remote Systems view exposes subsystems, filters, and items specified by each filter. This view lets you organize your filtered information in an easy-to-understand tree view. You can create filters for collections of libraries, objects, and source members.

The Remote System Explorer provides iSeries native file system (QSYS) support. The iSeries native file system lets you query what objects are on your iSeries host and perform actions against those objects. Filters allow you to easily organize elements within your system. Use the filter function to list iSeries native file system objects (such as libraries, objects, or members).

An initial library list filter also displays by default. It shows the initial libraries defined in your user profile on the iSeries host. You can manipulate this list to add or move libraries within the list.

The Remote Systems Explorer supports the Native File System (QSYS) and allows you to query the objects that are on your iSeries host. Using filters, you can easily organize how you wish to view the objects within your system. When you access the list by expanding the filter, you will see the Native File System objects (such as libraries or files) and can perform actions against these remote objects.

If you have been using the Remote Systems Explorer for some time, your workspace may contain too many filters to easily navigate. In this case, filters can be partitioned into pools, where one pool contains filters for many different elements. For example, one filter pool may contain filters for your accounts receivable program while another contains filters for your payroll program.

The Remote Systems Explorer supports the Native File System (QSYS) and allows you to query the objects that are on your iSeries system. Using filters, you can easily organize how you wish to view the objects within your system. When you access the list by expanding the filter, you will see the Native File System objects (such as libraries or files) and can perform actions against these remote objects.

First you will need to specify the library you want to work with directly:

- 1. In the Remote Systems view expand the connection that connects to your iSeries system, by clicking on the plus sign beside it.
- 2. Expand iSeries Objects

To create a new library filter:

 Expand Work with libraries. (You can also right-click iSeries Objects and select New > Library Filter).

Expanding Work with libraries corresponds to the WRKLIBPDM command, plus creates and expands the filter in the Remote Systems view. You see this window show up:

New				
Library Filter Create a new iSeries library	, filter			
Libraty: RSE*				▼ • <u>Br</u> owse
<	Back	<u>N</u> ext >	Einish	Cancel

Figure 85: Specify a filter string

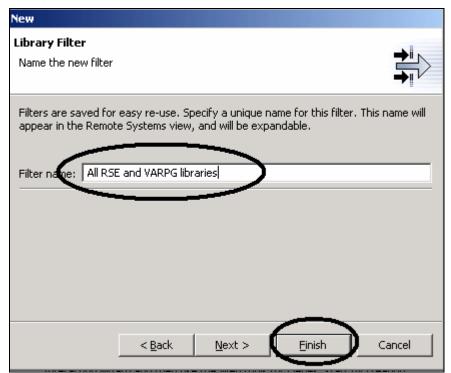
You are going to create a filter to specify the libraries you want to work with, so they will show in iSeries Objects. You want to create a filter that shows all libraries on the iSeries with the name **RSExxxxxx** and **VARxxxxxx**, xxx being any character.

Note: You may need to select different libraries that appear on your system if libraries with the above names do not exist.

Specify the first filter string that selects the libraries starting with Remote System Explorer:

- 1. Type RSE* into the Library field, using the * wild card character.
- 2. Click Next

The following window will show:



Specify a name for this filter:

- Type All RSE and VARPG libraries into the Filter name field.
 You give your filters a name because the Remote System Explorer saves them for future use, in opposition to PDM, which does not save filters.
- 4. Click Finish

Back in the **Remote Systems** view under **iSeries Objects** you will see the new filter expanded, listing all RSE* libraries. Now you need to add the VARPG libraries:

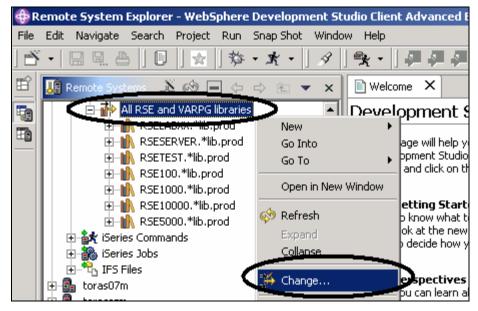


Figure 86: Select the Change option

 Right click the filter All RSE and VARPG libraries and select Change. The Change Library Filter window appears:

🕀 Change Library Filter	×
Filter name: Parent filter pool:	All RSE and VARPG libraries Weiss1 Filter Pool
Filter strings:	New filter string: Librate: VAR* Create Revert
1	

Figure 87: Add second filter string

- 6. Click **New filter string** in the list box
- 7. Enter VAR* in the Library field
- 8. Click Create

The VAR* filter string is added to the list box.

9. Click OK.

You are now back in the **Remote Systems** view. You will see the list expanded to include your filter.

🕀 R	emot	e Systen:	n Explore	r - WebS	phere	Developi	ment St
File	Edit	Navigate	Search	Project	Run	Snap Shot	: Windo
	-	명 및 (9 6	1	一参	• <u>\$</u> { •] \$
Ē		Remote Sy	stems 🔡	š 🗇 E	•	¢ 🗈	• ×
		i i 👬	All RSE an	d VARPG l	ibrarie:	s	
Ē		+	🔥 RSELA	BXX.*lib.p	orod		
		±		RVER.*lib	•		
		+		ST.*lib.pr			
	🗄 🔟 RSE100.*lib.prod						
		+		100.*lib.pr			
		+		1000.*lib.p			
		+		100.*lib.pr			
		.		ABXX.*lib.j			
		+		\BOO.*lib. \DO1_*‼⊨			
		±		4801.*lib. 4803.*lib.			
		+		4803.*110.1 4811.*11b.1			
				812.*lib.			
				,612, 110, (B13,*lib,)			
				B14.*lib.			
				B15.*lib.			

Figure 88: Expanded filter

Now you can work with the libraries directly and can drill down to the object you want to work with directly.

Now create an object filter. Object filters list a set of objects from your iSeries host in the Remote Systems view.

Creating an object filter

1. In the **Remote Systems** view, expand your connection and then expand **iSeries Objects**.

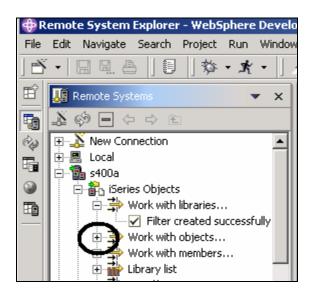


Figure 89: Create object filter

2. Expand Work with objects. You can also right-click iSeries Objects and select New > Object filter.

Note: Expanding Work with objects corresponds to the WRKOBJPDM command.

This will show the **New Object Filter** window

Now create a filter to show all your source files in your RSELABxx library.

©Copyright IBM Corporation 2003

New					
Object Filter Create a new iSe	eries object fil	ter			
Library:	RSELABXX	\mathbf{E}		••	Browse
Object:	*			••	Br <u>o</u> wse
Object type:	*				Bro <u>w</u> se
Object attribute:	*			•	Browse
		<u>M</u> ore T	ypes>>		
	4 D -	. [Rinsel a		
	< <u>B</u> a	CK.	<u>N</u> ext >	Einish	Cancel

Figure 90: Specify filter string

- 3. Enter RSELABXX in the Library field
- 4. Click Browse beside the Object type field

💠 Select Object Type		×
Select an object type:		
*CMD *CNNL *COSD *CSI *CSPMAP *CSPTBL *CTLD *CRG *CRQD *DEVD *DOC *DTAARA *DTAQ *EDTD *EDTD *EXITRG *FILE *FILE *FILE *FNTTBL *FNTSRC		
*FORMDF *FTR *GSS *IMGCLG		•
	QK	Cancel
		9

Figure 91: Select Object Type

- 5. Select the *File object type from the list
- 6. Click OK.

Back in the Object filter window:

New			
Object Filter Create a new iSe	ries object filter		
Library:	RSELABXX	.	Browse
Object:	*	▼·	Br <u>o</u> wse
Object type:	*FILE	••	Bro <u>w</u> se
Object attribute:	*		Browse
	More Types>>		
	< <u>B</u> ack <u>N</u> ext >	Einish	Cancel

Figure 92: Browse for Object Attribute

7. Click Browse beside the Object Attribute field

💠 Select Object Attribute		×
Select an object attribute: BSCF38 CMNF38 DDMF DFU		
DFUEXC DFUNOTEXC DKTF DSPF DSPF38 ICFF		
LF LF38 MXDF38 PF-DTA PF-SRC		
PF38-SRC PRTF PRTF38 SAVF TAPF		
	<u>OK</u> <u>C</u> an	cel

Figure 93: Select filter Object Attribute

- 8. Select **PF-SRC** from the Object Attribute list
- 9. Click OK
- 10. On the Object Filter window click Next.

New				
Object Filter Name the new filter				
Filters are saved for appear in the Remo				r. This name will
Filter name: My so				
	< <u>B</u> ack	<u>N</u> ext >	Einish	Cancel

Figure 94: Specify filter name

- 11. Specify the Filter name: My source files
- 12. Click Finish.

Note: You give your filters a name because the Remote System Explorer saves them for future use, in opposition to PDM, which does not save filters.

The new object filter now displays in the Remote Systems view under iSeries Objects:

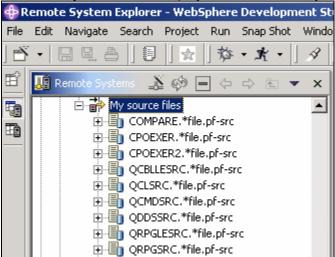


Figure 95: Object filter shows all source files in library

©Copyright IBM Corporation 2003

Now you know how to create filters and tailor your development environment. Filters can also be specified for non iSeries servers and your local system.

Tip: If you have been using the Remote Systems Explorer for some time, your workspace may contain too many filters to easily navigate. In this case, filters can be partitioned into pools, where one pool contains filters for many different elements. For example, one filter pool may contain filters for your accounts receivable program while another contains filters for your payroll program.

Now you can work with the objects you have in your Remote Systems view like you worked in PDM with libraries, objects, or members.

Assuming you want to edit the member **Payroll**, in QRPGLESRC you:

- 1. Expand **QRPGLESRC**
- 2. Right-click member **Payroll**
- 3. Select Open with > Remote Systems LPEX Editor

This will download the source member and open the editor with this member

After you have edited the member you could save it and then compile it from the Remote Systems

view by using the pop-up menu options on this member.

You can also create your own actions in addition to the default actions.

Creating a user action

In PDM you can create user actions in addition to the pre-supplied system actions. In Remote System Explorer you can do the same. So what are actions? Actions are host commands that you define on the Work With User Actions window, and will run against iSeries libraries, objects, jobs, and members. They can also be defined for folders and files in any remote UNIX, Windows, Linux, Local, or IFS system.

To create a user action:

1. Expand your iSeries connection and expand iSeries Object.

🕀 R	emote System	Explore	r - WebSphere	Developm	ent St	udio Client Ad	vanc	ed Editio
File	Edit Navigate	Search	Project Run	Snap Shot	Windo	w Help		
		4 D	☆ ☆	• <i>j</i> { •	A] 😤 🔻 🛛 💭	4	5
Ē	🜆 Remote Sys	tems 🛓	i 🏟 🗖 🗇	+ E -	r x	🖹 Welcome	×	
5	🖻 🖓 🔒 s400a					Developr	ner	nt Stuc
E	🖃 👬 iSeri							
	In the second se							elp you sta :udio Client
	Filter created successfully. Expand thi → → Work with objects							on the rela
			reated successf	ully. Expand	thi			
			members	aily i Expand		🕘 Getti	ng Sl	tarted
		Library list				Το κπα	ow wh	nat to do fi
		-	*lib.prod-sys					new inform ow you car
	. <u>.</u>	🍂 QSYS2	.*lib.prod-sys					011 you cu
	.		YS.*lib.prod-sy:			🕒 Pers	octi	V05
	•	-	5YS.*lib.prod-sy					rn all abou
	. <u>+</u>		SWSS.*lib.prod					e reasons
			54.*lib.prod-usr BXX.*lib.prod-us			openir	ngar	new one. H
		New			1			
		Go To	n	· · ·				i ystem Ex e through y
	∎ ⊕ 		-					ing you nei
		Expa	nd To	+				ny of the r
	E 👬	🚸 Refre	ach			on you	ur ise	ries host, a
	∎ ⊕ 酔	· ·						
	E E 🕂	Expa Colla				🕘 The n You ca		steps e WebSphe
			hae					ir RPG and
		💵 Rena	ime					f your appl
		🛱 Сору	,					technology ustom Web
		-				İntera	action	wizard and
	. <u>.</u>	🚊 Paste						ect to deve n Java usir
	🗉 🕀 🔁	🙁 Delet	e				_	
	│ ┊ ┊ 南⊶ ぶ ≫	Add I	Library List Entry	·	١Ť	Library RS	ELAB:	
	Remote System	🧩 Remo	ove From Library	List	F	Name		Туре
			e Up In Library Li	st	H			*PGM *PGM
	Properties	 Move	e Down In Librar	/ List	×	CLR1	-	*PGM *PGM
	Property					PAYROLL	-	*PGM
	Name	 ↔ Move 	e Within Library I	.ist		PATROLL		*PGM
	Number of ch	Show	/ in Table		\square	PAYROLLG		*PGM
	Source Status	Chan	ige		\vdash	STRCODEC	PP	*PGM
	Subtype		(B)		\vdash			1.91.1
	Text	Bosh		-	لمرا	Command Nor	mal	
	Туре	User	Actions	×	🞝 w	ork With User Ad	tions:	\sim
					_			

©Copyright IBM Corporation 2003 Course material may not be reproduced in whole or in part without the prior consent of IBM

Figure 96: Work with user actions

- 2. Expand Library list filter
- 3. Right-click **RSELABxx**
- 4. Select User Actions > Work with User Actions

You will see the Work with User Actions window:

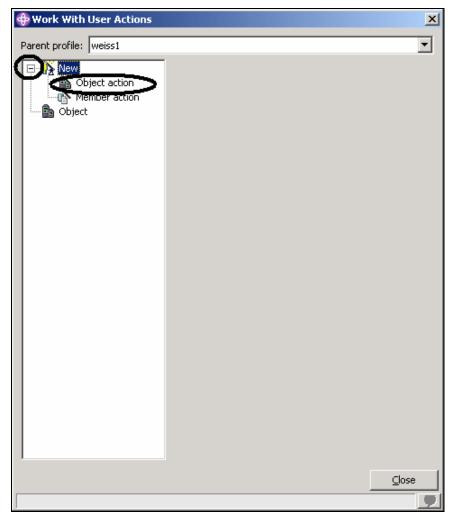


Figure 97: Work with User Actions window

- 5. Expand New in the list, if it is not expanded already
- 6. Click Object action

The Work with User Actions window appears

🕀 Work With User Actions		X
Parent profile: weiss1		•
New Object action Member action Object	Action name: Comment: Command:	Copy source file
	Insert variable	Browse Prompt
	Single selection	only \square Invoke once r which this action will appear
	Defined Types ALL CMD FILE FILE_DATA FILE_DSPF FILE_MBRS FILE_PRTF	Selected Types Add>> ALL < <remove< td=""> Edit</remove<>
	ALL: *	Create Revert
		9

Figure 98: Create an Object action

Now you want to create a user action that copies a source file with data to a new source file called QJUNKSRC in the same library.

- 7. Enter a name for the user action in the Action field: Copy source file
- 8. Enter a comment in the Comment field
- 9. Key in the command to execute CRTDUPOBJ
- 10. Click Prompt to get the command prompter for this command

Create Duplicate Object	t (CRTDUPOBJ)		×
From object:		 Name, generic* 	
From library:		ŀ− Name	
Object type:		▼ Add	
		Remove	
		Move up	
		Move down	
To library:	*FROMLIB	• Name	
New object:		 Name 	
Duplicate data:		D	
	Ok	Cancel	Help ?

Figure 99: Prompt for CRTDUPOBJ command

This is the command you will be running:

CRTDUPOBJ OBJ(&N) FROMLIB(&L) OBJTYPE(&T) NEWOBJ(QJUNKSRC) DATA(*YES)

- 11. Enter &N in the **From Object** field, to indicate to use the name of the selected object in the Remote Systems view
- 12. Enter &L in the From Library field, to pick up the library name form the selected object
- 13. Enter &T in the **Object Type** field, to pick up the object type from the selected object
- 14. Enter $\ensuremath{\texttt{QJUNKSRC}}$ in the New object field

Now you get the additional Duplicate Data parameter:

- 15. Select the View option on the menu bar of this window
- 16. Select the All parameters option from the pull down menu

Now the **Duplicate Data** parameter is also shown on the prompt window.

- 17. Select the Yes radio button for the Duplicate Data parameter
- 18. Click **OK**

You will be back at the Work with User Actions window

Refresh the Remote Systems view after the action has been run:

19. Select the Refresh after check box

Tip: Press **Insert variable** to bring up a list of valid replacement variables with the explanation of what they do.

🌐 Work With User Actions			×
Parent profile: weiss1			•
Parent profile: weiss1	Action name: Comment: Command: CRTDUPOBJ OBJ(&J NEWOBJ(QJUNKSR)	Copy source file Copy source file w Normal command N) FROMLIB(&L) OBJ C) DATA(*YES)	_
	Insert variable	only I i In<u>v</u>oke once	
	Defined Types	S	elected Types
	FILE_SRC: *FILE:PF	Apply	Rever <u>t</u>

Figure 100: Limit user action to source files

This user action is only valid for Source physical files. You need to specify this restriction so this user action will only show in pop-up menus when you right click on a source physical file. To specify this restriction:

- 20. Locate the **Defined Types** box
- 21. Select FILE_SRC in the list

22. Click **Add** beside the list

FILE_SRC is now one of the selected types. Actually since you only selected this one it is the only one.

- 23. Select the **Refresh** check box to refresh the Remote System view after the user action has been performed
- 24. Click Close.

Now, only when you right-click on a source file, will this user action appear on the pop-up menu selected. For any other object type it will not appear.

Back in the workbench and the Remote System Explorer perspective, give it a try.

	-	-	-		-			udio (Clien	t A	dvanced Edition for i	Series
File	Edit Navigate S	earch Proj	ect Run	Wind	wot	Hel	P					
	• 🛛 🖾 🐣		羚 • 考	•	5	3 []	R	•				
Ē	🌆 Remote System			× ×][
5	≱ 🕫 = ⇔ ⇔ 🖻											
Ċ.												
		1	-11									
	🕞 👬 My											
۲		COMPLINE										
Ŧð	.	с										
		QCMDSRC.*	_ '.									
										. 1		
	ORPGLESRC *file.pf-sr ORPGLESRC *file.pf-sr ORPGLESRC *file.pf-sr ORPGLESRC *file.pf-src ORPGLESRC *file.pf-src ORPGLESRC *file.pf-src				Nev					<u>}</u>		
					Go To					<u> </u>		
					🧒 Refresh							
	Geries Jobs Active jobs Geries Your active jobs Geries Your jobs Geries Geri			Expand Collapse Rename Copy Paste								
							-1					
	Hon	ne		🚸 Move								
	iSeries (Objecto		🗴 Delete						ŀ		
										_þ	9	
		rk with librari	ies		Show in Table							
				Change					ŀ			
	Remote Systems 1	ream		-	Sav -					Ē	ODELABXX type *FILE o	deleted.
		Γ		~	Res Use	r Acti	ions	_	-		🖉 Copy source file	
	Properties	Uslus	법 🖆 🗄		-	_		Dent	_		Copy source file	
	Property Name		c	-				Proje		C	🛃 Work With User Acti	ons
	Name QRPGLESRC Number of ch 0 Source CODELAB Status OK			Make Available Offline			-T	a abjects with supported abject to an a				
				Prop	pertie	es(B)				e objects with supported object types vel messages to determine if any object		
											,,	
	Subtype	PF-SRC										
	Text	*515			-14					_		1
	Туре	*FILE		_	30	Tasks	s iSei	ries Ta	able Vi	iew	iSeries Commands Log	iSeries Source
I	Jser defined actions	for the selec	ted resour	rces								

Figure 101: User action shows in pop-up menu

- 25. Locate your filter My Source files
- 26. Expand the **filter**, if it is not already expanded
- 27. Right click on the QRPGLESRC file
- 28. Select User Actions > Copy source file

The file gets duplicated and the list gets refreshed. Your new source file will show in the list.

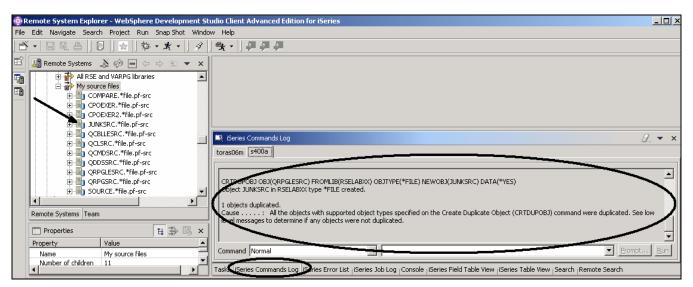


Figure 102: CRTDUPOBJ command in user action ran successful

You can check the messages of the CL commands you are running in the Remote System Explorer

job by looking at the Commands log in the right hand side bottom pane in the workbench.

To delete the source file QJUNKSRC that you just created:

- 29. Right-click **QJUNKSRC**
- 30. Select **Delete** from the pop-up menu.

Running commands from the Remote System Explorer

The Command entry is part of the Remote Systems view.

- 1. Check if you have an **iSeries Table View** tab in the bottom right view pane where your command log appeared in the previous task.
- 2. If you have it click on it
- 3. If you don't have it
 - a) In the Remote Systems view, right-click on your source file filter
 - b) Select **Show in table** from the pop-up menu

Now in the iSeries Table view you can run commands on the iSeries server that the table is connected to. You can run commands in the Commands field beneath the iSeries Table view, and view messages in the Messages field. After you populate the table, you can enter a command and select either Prompt to specify parameters and then Run, or just select Run (which prompts and runs the command as normal). When you run a command, the Messages drop-down field is populated with the messages from the command. When you select a message, the Details button is enabled. When you press this button, the message and its help is displayed.

Name	Туре	Attribute	Text	Status	Last modified	Created	Size
QCBLLESRC	*FILE	PF-SRC		OK	February 14, 20	June 5, 2001 10	307
QCLSRC	*FILE	PF-SRC		OK	February 14, 20	May 10, 2001 4:	942
QCMDSRC	*FILE	PF-SRC		OK	February 14, 20	May 10, 2001 4:	245
QDDSSRC	*FILE	PF-SRC		OK	February 14, 20	May 10, 2001 4:	208
QRPGLESRC	*FILE	PF-SRC		OK	February 14, 20	June 5, 2001 10	180
QRPGSRC	*FILE	PF-SRC		OK	February 14, 20	May 10, 2001 4:	655
SOURCE	*FILE	PF-SRC		OK	February 14, 20	May 10, 2001 4:	634
Command Norma	al		ADDLIBLE	>		•	Pro
Messages Objec	t QJUNKSRC in R	SELABxx type *FILE	deleted.			▼ <u>D</u> etai	s
Tasks JiSeries Co	mmands Lon LiSe	ries Source Promote	er TiSeries Error List TiSeries Jo	b Log Series Table	View		

Figure 103: Table view with command entry

In the **command** field:

4. Key in an iSeries command for example ?ADDLIBLE

The question mark is there to display a prompt screen

Tip: Instead of specifying a question mark you could use the Prompt push button

5. Click Run

The Command Prompt window for the ADDLIBLE command appears:

👹 Add Library List Entry (AD	DDLIBLE)	×
<u>File E</u> dit <u>V</u> iew <u>H</u> elp		
Library:	RSELABOO	
Library list position:		
List position:	*FIRST	
Reference library:	Name	
	OK Cancel Helt	0 ?

Figure 104: Command prompt window In the Prompt window:

6. Key in RSELAB00. That will add this library to the library list of your Remote System Explorer job on the iSeries server.

The messages field will confirm the successful completion of this command.

To get to the command history, similar to F9 on the green-screen, press the down arrow on the **Command** field.

Name	Туре	Attribute	Text	Status	Last modified	Created	Size
COMPARE	*FILE	PF-SRC		ОК	July 17, 2003 11	May 10, 2001 4:	49152
QCBLLESRC	*FILE	PF-SRC		OK	July 17, 2003 11	June 5, 2001 10:	311296
QCLSRC	*FILE	PF-SRC		OK	July 17, 2003 11	May 10, 2001 4:	114688
QCMDSRC	*FILE	PF-SRC		OK	July 17, 2003 11	May 10, 2001 4:	28672
0 QDDSSRC	*FILE	PF-SRC		OK	July 17, 2003 11	May 10, 2001 4:	204800
QRPGLESRC	*FILE	PF-SRC		OK	July 17, 2003 11	June 5, 2001 10:	180224
QRPGSRC	*FILE	PF-SRC		ОК	July 17, 2003 11	May 10, 2001 4:	139264
ommand Normal						(Prompt R
1essages Library			7ADDLIBLE 7rmvlible rselab00 DLTF FILE(RSELAB) DLTL BL(RSELAB)	XX/JUNKSRC)			s <u>S</u> how Log

Figure 105: Command history

You could also use the **iSeries Commands** subsystem in the Remote Systems view underneath the **iSeries Objects** subsystem and run predefined commands or define your own commands.

Complete the checkpoint below to determine if you have mastered the content in this exercise.

Checkpoint

- 1. The Remote System Explorer is a replacement for:
 - A. PDM
 - B. SDA
 - C. CGU
 - D. DFU
- 2. A filter:
 - A. Queries objects on your iSeries system
 - B. Easily organizes how you wish to view objects within your system
 - C. Shows the native file system objects (QSYS)
 - D. Allows you to perform actions against these remote objects
 - E. All of the above
- 3. Expanding Work with libraries corresponds to the:
 - A. WKRLIBPDM
 - B. WKROBJPDM
 - C. WRKMBRPDM

- D. All of the above
- 4. You give your filters a name because the Remote System Explorer saves them for future use. This is different from PDM that does not save filters. (T, F)
- 5. You can create:
 - A. Library filters
 - B. Library list filters
 - C. Object filters
 - D. Member filters
 - E. Job filters
 - F. All of the above
- 6. In PDM you can create user actions in addition to the pre-supplied system actions. In Remote System Explorer you can do the same. (T, F)
- 7. You can run commands on the iSeries server that the iSeries Table view is connected to. (T, F)

Practice

Given your own libraries on your iSeries system, create a member filter and a job filter. Then move libraries up, down and within your library list. Finally create a filter pool. Use the Development Studio Client for iSeries online help to assist you in these tasks.

What you just did

In this exercise you used the Remote System Explorer perspective to work with the iSeries objects that you used in the previous exercises. You also learned how easy it was to perform actions and define your own actions. In short, you saw how Remote System Explorer can organize and integrate your work and make that work easier.

Maintaining an iSeries application: Introduction to Remote Systems Explorer Tutorial Summary

See how easy it is to be more productive building new or traditional iSeries applications. The Remote System Explorer gives you PDM-like access to OS/400 resources. The PDM like table view meant you could right-click on objects to perform actions that are the same as PDM's and use a command line at the bottom of the table just like PDM. There were also user-defined actions that supported all PDM substitution variables and of course the editing, verifying, compiling, running and debugging tasks which were tightly integrated with the Remote System Explorer.

You have been introduced to a highly integrated and productive environment, offering a consistent experience for all development work from RPG to Java. This new generation of tools offered rich support for exploring the file system, compiling, building, editing, running and debugging. The new tools allowed you to leverage the classic tools for extremely rich editing and DDS design support. The Remote System Explorer tools offered a superset of functions over the classic, and feature frozen ADTS tools. It offered significant productivity and usability gains, support for disconnected and team development and a common harness for the tight integration of IBM and partner-supplied tools for server application development. Using these new generation tools also implicitly increases your skills and will make the transition into new programming models easier, such as the IBM WebFacing Tool, Web, Web Services, Java and XML.

Step aside ADTS the traditional method for developing and maintaining server-side iSeries applications. Here comes Development Studio Client which includes new highly integrated and highly extendible tools for iSeries RPG, COBOL, C, C++, CL and DDS development. These new tools offered you a development experience that is consistent with the experience for developing Java, Web, Web Services, and XML applications, lowering the learning curve for all and making your transition to these e-business application programming models that much easier.

1	2	3	4	5	6	7	8
1. B	1. D	1. C	1. D	1. D	1. B	1. C	1. A
2. G	2. E	2. A	2. T	2. D	2. D	2. T	2. E
3. G	3. B	3. F	3. A	3. G	3. AC,	3. A	3. A
4. E	4. AA,	4. C	4. T	4. T	BB,	4. C	4. T
5. E	BD,	5. T	5. D	5. T	CA,	5. C	5. F
6. A	CB,	6. E	6. C	6. T	DD,	6. T	6. T
7. T	DC	7. E	7. B	7. F	EF,	7. F	7. T
8. A	5. E	8. C	8. C	8. A	FE,	8. T	
9. T		9. T	9. C	9. D	GG	9. E	
10. E		10. C	10. A	10. D	4. E	10. A	
11. B		11. B	11. D	11. D	5. T		
12. F		12. T	12. C	12. T	6. H		
13. E		13. T	13. T		7. T		
14. T		14. D	14. D		8. T		
15. T			15. E		9. T		
			16. D		10. T		
			17. D		11. T		
			18. B				
			19. C				
			20. T				
			21. T				

Exercise Checkpoint Answers

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only the IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of the document does not give you any license to these patents. You can send license inquiries, in writing, to:

Director of Licensing Intellectual Property & Licensing International Business Machines Corporation North Castle Drive, MD – NC119 Armonk, New York 10504-1785 U.S.A

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND,K EITHER EXPRESS OR IMPIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner service as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact: IBM Canada Ltd. Laboratory Information Development B3/KB7/8200/MKM 8200 Warden Avenue Markham, Ontario Canada L6G 1C7

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Programming Interface Information

This publication is intended to help you to create and manage applications and user interfaces on the workstation, in a client/server environment. It contains examples of data and reports that are used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses that are used by an actual business enterprise is entirely coincidental. This publication documents General-Use Programming Interface and Associated Information provided by IBM WebSphere Development Studio Client for iSeries.

Trademarks and Service Marks

The following terms are trademarks or registered trademarks of the International Business Machines Corporation in the United States or other countries or both:

Application System/400 AS/400 AS/400e DB2/400 e-business IBM iSeries Integrated Language Environment OS/400 RPG/400 VisualAge WebSphere

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Active X, Microsoft, Windows, and Windows NT and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, or other countries, or both. Other company, product, or service names may be trademarks or service marks of others.