IBM WebSphere Development Studio Client for iSeries

IBM

Using the Remote System Explorer (RSE) Version 6.0.1

IBM WebSphere Development Studio Client for iSeries

IBM

Using the Remote System Explorer (RSE) Version 6.0.1

Contents

Introduction	. v
Chapter 1. Module 1. Starting the product and the Remote System	4
Explorer. Exercise 1.1: Starting the product Exercise 1.2: Opening the Remote System Explorer	. 1
perspective	. 3
connection to an iSeries system and connecting to an iSeries	. 7
connecting to an iSeries Exercise 2.1: Configuring a connection to an iSeries	. /
0	.7 .10
Exercise 2.3: Viewing and accessing objects in the Remote System Explorer perspective Exercise 2.4: Opening a second source member and	. 14
multiple views	. 19 . 21
Chapter 3. Module 3. Editing source Exercise 3.1: Introducing the editor	
The second se	
	. 26 . 33
Exercise 3.4: Requesting undo and redo operations	34
	. 35
	. 40
Exercise 3.7: Indenting source	. 42
Exercise 3.8: Finding and replacing text	. 44
Exercise 3.9: Filtering lines by string	. 46
0 , 11	. 47
0 1	. 48
Exercise 3.12: Comparing file differences from the	50
Remote Systems view	. 50
Exercise 3.13: Comparing files in the CODE Editor (optional)	. 54
(optional)	. 56
Chapter 4. Module 4. Verifying and	
compiling source	63
Exercise 4.1: Verifying the source	62
Exercise 4.1: Verifying the source	. 03
Exercise 4.2: Complifing source remotely	. 07
iSeries Table view	71
Exercise 4.4: Running commands and programs .	
Chapter 5. Module 5. Debugging a	
program	79
Exercise 5.1: Introducing the Integrated iSeries	
Debugger	. 80 . 80

Chapter 6. Module 6: Exploring Remote System Explorer	105
Exercise 6.1: More about the Remote System	105
	105
Exercise 6.2: Creating a library filter	
Exercise 6.3: Creating an object filter.	100
Exercise 6.4: Creating a user action	113
Exercise 6.5: Running commands from the Remote	. 115
System Explorer	117
	. 117
Chapter 7. Module 7. Designing	
	100
screens.	123
Exercise 7.1: Opening a DDS member in the	101
	. 124
	. 125
	. 126
	. 127
Exercise 7.5: Creating groups from existing records	
Exercise 7.6: Creating new screens	. 130
	. 132
Exercise 7.8: Switching between multiple records	135
Exercise 7.9: Adding field error handling	. 137
	. 139
	. 141
Exercise 7.12: Verifying the source changes	. 143
Exercise 7.13: Switching between designing and	
editing the screen	. 145
Exercise 7.14: Compiling your source changes and	
closing the Designer	. 146
0 0	
Chapter 8. Module 8. Introducing the	
product and Remote System Explorer	
(optional)	1/10
	145
Introducing Development Studio and Development	
Studio Client	. 149
Introducing iSeries Application Development Tools	150
Chapter 9. Summary	155
Appondix Noticos	157
	157
	158
Trademarks	. 159
	iii

Exercise 5.3: Setting breakpoints85Exercise 5.4: Monitoring variables88Exercise 5.5: Stepping into a program.93Exercise 5.6: Listing call stack entries95Exercise 5.7: Setting breakpoints in PAYROLLG95Exercise 5.8: Removing a breakpoint in PAYROLLG97Exercise 5.9: Monitoring variables in PAYROLLG98Exercise 5.10: Adding a memory monitor99Exercise 5.11: Setting Watch breakpoints100Exercise 5.12: Closing the debug session102

Introduction

This tutorial teaches you how to maintain a payroll application written in ILE RPG using the Remote System Explorer.

You will learn how to start the product and open the Remote System Explorer perspective, use tools and views in this perspective to connect to an iSeriesTM system and edit, verify, compile and debug a payroll application.

Prerequisites

To complete this tutorial, you should be 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.

It will also help if you understand:

- ILE RPG
- DDS

The file(s) required for this tutorial are available for download at http://ibm.com/software/awdtools/wdt400/library.

Time required

To complete the modules of this tutorial, you will need approximately **3 hours**.

Learning objectives

This tutorial is divided into eight modules, each with its own learning objectives. You can choose to skip modules 7 and 8 and you can complete modules 5, 6, 7 and 8 in any order after module 4. Each module contains several exercises that must be completed in order for the tutorial to work properly.

Chapter 1, "Module 1. Starting the product and the Remote System Explorer," on page 1 teaches you about the workbench, a perspective, and specifically the Remote System Explorer perspective. In this module, you will:

- Start the product
- Specify the workspace for your project resources
- Open the iSeries default perspective

Chapter 2, "Module 2. Configuring a connection to an iSeries system and connecting to an iSeries," on page 7 teaches you how to create a connection to an iSeries server and select objects using the Remote System Explorer perspective. In this module, you will:

- Create a connection to an iSeries system
- Connect to an iSeries system
- Add a library to your library list
- · View libraries in your job's library list from the Remote Systems view
- Find a source physical file in you library

- · View members in a source physical file using the iSeries Table view
- Customize the columns in the iSeries Table view
- · Open a member for edit from the iSeries Table view or the Remote Systems view
- Maximize the editor window
- Open another member for edit
- · Open multiple views of a source member
- · Switch from one edit session to another edit session
- · Display a structural outline of items defined in a source member

Chapter 3, "Module 3. Editing source," on page 25 teaches you how to use the Remote Systems LPEX Editor to edit source. In this module, you will:

- Change the default settings of the LPEX Editor Parsers
- Change the color settings and font used by the Editor
- · Change the default behavior of the Enter key
- Use SEU commands to edit source
- Undo and redo source changes
- View language sensitive help for the MOVE operation code
- View a list of all help contents
- Limit the search of help to specific documents
- Search the help
- Request a prompt for a specification line
- · Display context sensitive help for any field in the iSeries Source Prompter
- View the beginning and ending of constructs in your source
- · Use the Find and Replace window to search for an item in your source
- Filter or subset your source
- · Filter lines based on line type
- Search through members in a source physical file
- · Compare different versions of a program and identify the differences
- Syntax check source by line
- View help on syntax errors

Chapter 4, "Module 4. Verifying and compiling source," on page 63 teaches you how to verify and compile source in the Remote Systems LPEX Editor. In this module, you will:

- · Check for semantic errors on your workstation
- Start the Program Verifier tool
- Use the iSeries Error list to locate each error in the source
- Use content-assist to fix an error
- Save your source
- Re-verify source
- Change compile preferences
- Invoke the compile command
- · Change the current library using the Command field in the iSeries Table view
- Start an interactive connection
- Invoke the payroll program

Chapter 5, "Module 5. Debugging a program," on page 79 teaches you how to debug your payroll program from the workstation. In this module, you will:

- Invoke the debugger from the Launch Configurations window
- · Add a breakpoint
- Add a conditional breakpoint
- · Edit a breakpoint
- Monitor a variable through the Monitors view
- Step into your payroll program
- Show a Listing view
- List the call stack entries in the Debug view
- View all breakpoints
- Remove a breakpoint
- Monitor storage
- Set a Watch breakpoint
- Close the Debugger

Chapter 6, "Module 6: Exploring Remote System Explorer," on page 105 teaches you how easy it is to define filters, perform actions and define your own actions. In this module, you will:

- · Know the features of Remote System Explorer
- · Create a filter to show specific iSeries libraries
- Change the filter to add more iSeries libraries
- Create a filter to show all the source files in a library
- · Access members to edit from your filter
- Create a user action that copies a source file with data to a new source file in the same library
- Specify user action parameters
- Specify a restriction on a user action
- Try the user action
- Run an OS/400[®] command from the iSeries Table view

Chapter 7, "Module 7. Designing screens," on page 123 teaches you how to use CODE Designer to modify a display file. In this module, you will:

- Open a DDS member for edit with CODE Designer
- · Show file-level keywords and record-level keywords
- · View the details of records, record-level keywords and field-level keywords
- View the design of the payroll application main menu
- Create a group from an existing record format
- Create a new group and add a subfile record and a subfile control record
- Add columns to the subfile record
- · Add fields to the subfile control record
- Copy existing fields
- Set indicators to handle field errors
- View and update record and field properties
- · View keywords and the properties of a keyword
- Insert a keyword
- View help for a keyword

- Check there are no semantic errors in the DDS source
- View help for an error
- · Launch the editor in read mode from the error list
- Launch the editor in write mode to fix the error
- Find a keyword in the source
- Save source changes
- Compile your source changes
- Close the Designer

Chapter 8, "Module 8. Introducing the product and Remote System Explorer (optional)," on page 149 teaches you about IBM[®] WebSphere[®] Development Studio Client for iSeries and describes its relationship to IBM WebSphere Development Studio for iSeries. In this module, you will:

- Know the goals of the product
- Know the editions of the product
- Identify the host tools and the client tools
- List and describe the iSeries application development tools

When you are ready, begin Chapter 1, "Module 1. Starting the product and the Remote System Explorer," on page 1.

Chapter 1. Module 1. Starting the product and the Remote System Explorer

This module teaches you about the workbench, the workspace, a perspective and specifically the Remote System Explorer perspective.

In this module, you will:

- · Start the product
- Set the default workspace
- Access unique tools and views targeted towards iSeries application development tasks

Requirement: Before beginning this module, you should have the prerequisite knowledge outlined in "Introduction" on page v.

Exercises

The exercises within this module must be completed in order:

- "Exercise 1.1: Starting the product"
- "Exercise 1.2: Opening the Remote System Explorer perspective" on page 3

Time required

This module will take approximately 10 minutes to complete.

Exercise 1.1: Starting the product

If you want to know more about the product before you get started you can read Chapter 8, "Module 8. Introducing the product and Remote System Explorer (optional)," on page 149.

First you must start the product. Follow these steps to start the product:

- 1. Click **Start** on the task bar of your desktop.
- 2. Select Programs > IBM Rational[®] > IBM WebSphere Development Studio Client for iSeries V6.0 > WebSphere Development Studio Client for iSeries

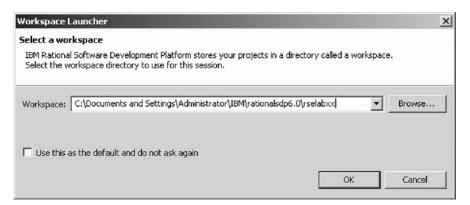
🔚 IBM WebSphere Development Studio Client for iSeries V6.0 🔸 🍪 🛛 WebSphere Development Studio Client for iSeries 🔪

If you are working with the Advanced Edition of the product you will see the words Advanced Edition in the product name.

If you are working with the WDSC Lite version you will see the following start menu option.

🔚 IBM WebSphere Development Studio Client Lite technology preview V6.0.1 🔹 🔍 WDSC Lite

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



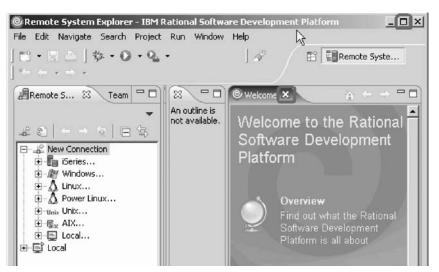
- 4. (Optional) Change the field in this dialog and use a unique directory name, for example, RSELABxx (where xx is a unique number).
- 5. Click **OK** to open the workbench.



6. Click the icon in the far right of the Welcome page to go to the Workbench.



7. Click the maximize button to maximize the workbench.



8. Click the X in the Welcome tab to close the Welcome view.

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 views and an editor.

You have started the product and now you are ready to begin "Exercise 1.2: Opening the Remote System Explorer perspective."

Exercise 1.2: Opening the Remote System Explorer perspective

Before you begin, you must complete "Exercise 1.1: Starting the product" on page 1.

In these steps you will open the Remote System Explorer perspective.

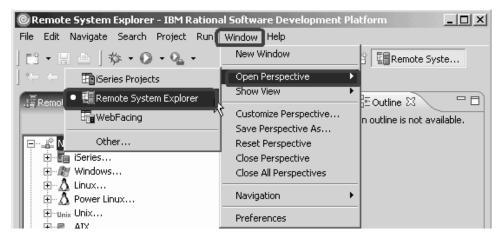
1. Check for the name of the perspective.



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 a program. 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:

 Click Window > Open Perspective > Remote System Explorer from the workbench menu.



The Remote System Explorer perspective opens.

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 perspective will be active when you start the product 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 use the views from the workbench.

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 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 tools and views that you would commonly use while editing Java source files
 - b. Contains tools and views that you would use while debugging programs
 - **c.** Contains tools and views that you would use while developing Web applications
 - d. Contains tools and 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

More practice

Given your experience in opening the Remote System Explorer perspective, now open the Web perspective. See the list of tools and views available to the Web developer. Next open the Java perspective. See the list of 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 top-right frame for the Web perspective icon. Now close both the Java perspective and the Web perspective.

Module recap

You have completed Chapter 1, "Module 1. Starting the product and the Remote System Explorer," on page 1. You have learned how to:

- Start the product
- Set the default workspace
- Access unique tools and views targeted towards iSeries application development tasks

Now that you have started the product and have opened the Remote System Explorer perspective, you can move on to getting connected to an iSeries system. Continue to Chapter 2, "Module 2. Configuring a connection to an iSeries system and connecting to an iSeries," on page 7.

Chapter 2. Module 2. Configuring a connection to an iSeries system and connecting to an iSeries

This module teaches you how to create a connection to an iSeries server, find a library in your library list, select objects from a library and finally 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.

In this module, you will:

- · Create a connection to an iSeries system
- · Connect to an iSeries system
- Add a library to your library list
- · View libraries in your job's library list from the Remote Systems view
- Find a source physical file in your library
- · View members in a source physical file using the iSeries Table view
- Customize the columns in the iSeries Table view
- Open a member for edit from the iSeries Table view or the Remote Systems view
- Maximize the editor window
- Open another member for edit
- · Switch from one edit session to another edit session
- Open multiple views of the same source member
- Display a structural outline of items defined in a source member

Exercises

The exercises within this module must be completed in order:

- "Exercise 2.1: Configuring a connection to an iSeries system"
- "Exercise 2.2: Connecting to an iSeries system" on page 10
- "Exercise 2.3: Viewing and accessing objects in the Remote System Explorer perspective" on page 14
- "Exercise 2.4: Opening a second source member and multiple views" on page 19
- "Exercise 2.5: Displaying an outline of a source member" on page 21

Time required

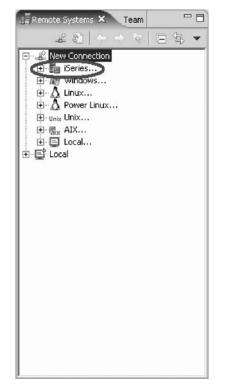
This module will take approximately 10 minutes to complete.

Exercise 2.1: 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 specify a profile. All connections, filters, and filter pools belong to profiles. Filters are described in a later exercise. Profiles are discussed when you create your first connection.

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

1. 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.



Click the plus sign + beside **iSeries** to configure a connection to an iSeries system.

The Name personal profile page opens.

New	And the second	
ame personal profile		
Uniquely name user profile		=0=
Welcome to Remote Systems, Conni you, Enter a profile name to uniquel decide for each new connection whe profile,	y identify you from your tea	m members. You will
weiter Uscherkh7hn		
Profile: Steckh7tp		
Profile: hsteckh7tp		
Profile: Steckh7tp		
Profile: [ˈnsteckh7tp		
Profile: Steckh7tp		

2. Click Next to accept the default value.

The profile defaults to the name of the workstation. Your profile will be different from the one shown here.

The Remote iSeries System Connection page opens.

New					x
Remote iSeries S Define connection	a share a shere	ction			
Parent profile:	msteckh7tp				•
Connection name:	s400a				
Host name: Description:	54008				
🔽 Verify host nan	le				
			G.		
	< Back	Next >	Finist	2	Cancel

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

- **3**. In the **Host name** field, type the IP address or the name of your host system. 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.
- 4. Leave the **Parent profile** default value. You don't need to change it.
- 5. Leave the **Verify host name** check box selected.

6. Click Finish to define your system.

You have configured a connection and now you are ready to begin "Exercise 2.2: Connecting to an iSeries system."

Exercise 2.2: Connecting to an iSeries system

Before you begin, you must complete "Exercise 2.1: Configuring a connection to an iSeries system" on page 7.

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

iSeries Objects

A PDM-like group, allowing access to libraries, objects and members.

iSeries Commands

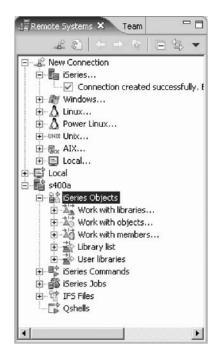
Contains predefined commands and allows you to create command sets each of which contain one or more often used commands. When run, all commands in a command set are sent to the remote system and executed, and the results are logged in the iSeries Commands log.

iSeries Jobs

Allow you to see various jobs, subset by job attributes, and to perform a 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.



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)

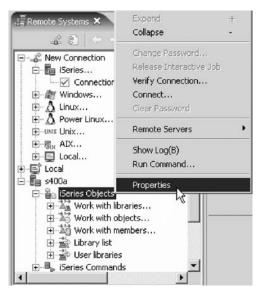
In addition there are entries for working with library lists and user libraries:

- Library list (to simulate PDMs WRKLIBPDM you can start with the pre-defined Library list filter, that when expanded lists all libraries in your library list.)
- User libraries (allows you to work with all user libraries you can access on that iSeries server.)

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 your library list and add the library that you'll be using in this tutorial:

a. Right-click iSeries Objects and click Properties on the pop-up menu.



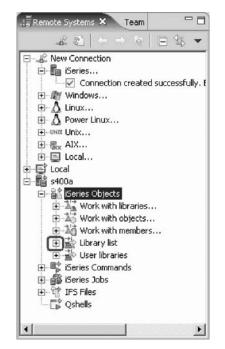
- b. Select Initial Library List on the left pane.
- c. Type RSELABxx where XX is a unique number in the **Library** field and click **Add**.

Add(B)
Add(B)
Remove
fove Up
ove Down
Apply

d. Click OK.

This will add the library RSELABxx to your library list every time you use this connection.

- Note: You can also change your library list using the pop-up menu items Add Library List Entry or Change Current Library on the Library list folder in the iSeries Objects subsystem. These changes are only valid until you disconnect.
- 3. Expand the Library list folder.



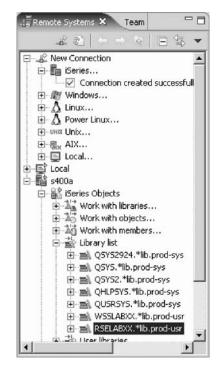
Now the connection will be activated and you will be prompted for a user ID and password.

System type:	iSeries
Host name:	TORAS07M
User ID:	WLABxx
Password:	****
	Save user ID
	✓ Save password

- 4. Enter your user ID and password.
- 5. Select the Save user ID check box.
- 6. Select the Save password check box.
- 7. Click OK.

As you know, you can use the properties of any of the subsystems to set connection information such as adding a library to a library list.

Back in the workbench in the Remote Systems view you will see the libraries in your job's library list.



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

For each library, you can right-click and select from a number of actions. For example, there is an action to create a new source file within the selected library. Common actions like delete, move, copy, etc. are valid for all kinds of objects.

You have connected to an iSeries system and now you are ready to begin "Exercise 2.3: Viewing and accessing objects in the Remote System Explorer perspective."

Exercise 2.3: Viewing and accessing objects in the Remote System Explorer perspective

Before you begin, you must complete "Exercise 2.2: Connecting to an iSeries system" on page 10.

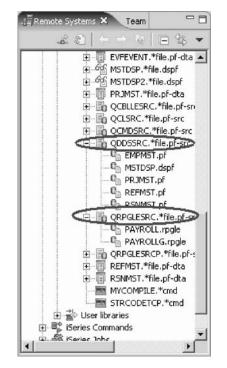
Now you are ready to view and access objects in your library RSELABxx.

To view and access 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. The list of actions depends on the object selected and whether you selected one or multiple objects. For example, for a source file the pop-up menu has an action to create a new member within the selected file.

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



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 RPG source member, the pop-up menu actions include:

- open with
- browse with
- verify
- compile

Before you go ahead and work with these members, let's see the members in the iSeries Table view as well because that is 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 then click **Show in Table** on the pop-up menu.

🗒 Remote System	s 🗙 Team	- 0)	
- # @		日 毎 👻	
÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷÷	PRJMST.*file	*file.pf-sro .pf-src ile.pf-src	
	EMPMST.		
	C MSTDSP.		
	PRJMST. REFMST. REFMST. QRPGLESRC. PAYROLL	pf 🐑 Refresh pf Expand *file- Collapse	+ -
I Properties ⊠	Remote Scratch	npad Paste	
I Properties X Property	Remote Scratch	apad Paste	
	Remote Scratch	ypad Paste → → → → → → → → → → → → →	
Property Attribute Name	Remote Scratch	pad ▲ Paste → Move ★ Delete	
Property Attribute	Remote Scratch	pad Paste A Move Copy Paste A Move Show in Table Cache File Desc	Stions
Property Attribute Name Number of chi Source	Remote Scratch	pad ▲ Paste → Move ★ Delete	Stions
Property Attribute Name Number of chi Source Status	Remote Scratch	pad Paste A Move Copy Paste A Move Show in Table Cache File Desc	Stions
Property Attribute Name Number of chi Source Status Text	Remote Scratch	pad Paste → Move → Move → Delete Cache File Desc © Find String Change(B)	Actions
Property Attribute Name Number of chi Source Status	Remote Scratch	pad Paste	Ditions

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.

	QCMDSRC.*file.pf- CDDSSRC.*file.pf- CDDSSRC.*file.pf- CDDSSRC.*file.pf CDDSP.dspf CDDSP.dspf CDDP.dspf		Remote System File RSELABXX/QDDS	Tasks 🕞 iSeries ISRC (5 Members)	Table View 23 iSeries C
	REFMST.pf		Name	Туре	Attribute
	L RSNMST.pf		C EMPMST	PF	SRC
•		<u> </u>	C MSTDSP	DSPF	SRC
	Remote Scratchpad	-8	C PRJMST	PF	SRC
			C REFMST	PF	SRC
	いた	· •	C RSNMST	PF	SRC
Property	Value	•	4		
Attribute	PF-SRC		*		
Name	QDDSSRC		- 1722		
Number of chi	5		Command Norm	al 💌	
Source	RSELABXX				
Status	OK	-	Messages		▼ Detailei
•		+	incoodges [
Status		► table vi	Messages w		

The top of the iSeries Table view contains a lock icon **b** that controls the correlation between the Remote Systems view and the iSeries Table view. If the lock is disabled 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

content of the iSeries Table view. To enable or disable the lock, you can click it once to change its state. You can click on the columns heading to sort the view by column.

5. In the iSeries Table view toolbar make sure the lock/unlock button is in the unlock position. Leave the mouse pointer over the tool button for a second or two to display the flyover help. That way you can check if the view is locked or unlocked.

This means now the table will automatically be updated when a different object is selected in the Remote Systems view. This is a shortcut to open the pop-up menu for an object in the Remote Systems view and to select Show in Table.

You can also modify which specific columns you want to see in the iSeries Table view.

- 6. To modify the iSeries Table properties:
 - a. Click **Window > Preferences** from the workbench menu.

The Preferences Window opens.

+ Workbench	Table View		
∃ Ant Build Order	Re-open to previous s	tate	
⊡ Help ⊡ Install/Update	☑ Use the command line	for action parame	eters
 Internet iSeries Projects 	Specify columns and colum	n order	
-J2EE	Type: Member		
- Java	rype: Imember		
Remote Systems Communications Debug File Cache Files Cache Cache Command Executio	Available columns Name Type Attribute Text Status Last modified Creaced	Move Up Move Down <remove Add(B)></remove 	Selected columns Name Type Attribute Text Status
Command Subsyste Job Subsystems LPEX Editor Parsers Objects Program Verifiers Table View Templates Logging		Restore Defa	ults Apply

- b. In the left pane of the Preferences window, expand Remote Systems.
- c. Expand iSeries under Remote Systems.
- d. Click Table View under iSeries.
- e. In the right pane of the Preferences window, select **Last modified** in the **Available columns** list.
- f. Click the Add button.
- g. Click OK.

Now, let's update the iSeries Table view.

h. Click the down arrow on the iSeries Table view title bar.

Remote System	Tasks 🔒	iSeries Table	View 🗙	iSeries Comma	- 0
File RSELABXX/QDD	SSRC (5 Mem	nbers)		# & + > 6	2 🗸
Name	Туре		Attribut	Work with	•
C EMPMST	PF		SRC	Subset	-
C MSTDSP	DSPF		SRC	Show Al	
C PRJMST	PF	Custon	ined.		
REFMST	PF	✓ Default			
RSNMST	PF		- 10	Position to	
•	1	All		Print	•

i. Click **Show columns > Customized** in the pop-up menu. Now you'll see the extra column that you've added.

File RSELABXX/QI	DDSSRC (5 M	embers)			
Name	Туре	Attribute	Text	Status	Last modified
C EMPMST	PF	SRC		ОК	December 15, 2005 10:59:00 AM
Ca MSTDSP	DSPF	SRC		ОК	December 15, 2005 10:59:01 AM
C PRJMST	PF	SRC		OK	December 15, 2005 10:59:03 AM
REFMST	PF	SRC		OK	December 15, 2005 10:59:04 AM
RSNMST	PF	SRC		OK	December 15, 2005 10:59:05 AM

You can also sort the objects in the iSeries Table view by column.

j. To sort the objects in ascending order by Last modified, click on the heading.

File RSELABXX/QU	DDSSRC (5 M	embers) 👡		·····
Name	Туре	Attribute Text	Status	Last modified
RSNMST	PF	SRC	OK	December 15, 2005 10:59:05 AM
REFMST	PF	SRC	OK	December 15, 2005 10:59:04 AN
C PRJMST	PF	SRC	OK	December 15, 2005 10:59:03 AM
C MSTDSP	DSPF	SRC	OK	December 15, 2005 10:59:01 AM
C EMPMST	PF	SRC	ОК	December 15, 2005 10:59:00 AM

k. If you click the heading the second time, it will sort it in descending order.

7. In the Remote Systems view, select QRPGLESRC.

The table shows the members in QRPGLESRC.

		Remote System D	Tasks 🔒 iSerie:	s Table View 🕄	iSeries Comman
	C.*file.pf-src	File RSELABXX/QRPGL	ESRC (2 Members	5)	j≣ - ⊕ - +
•		Name	Туре	Attribute	Text
	»	PAYROLL	RPGLE	SRC	
Properties 🖾 🔪	»ı 🗖	PAYROLLG	RPGLE	SRC	
	日本 マー				
Property	Value 🔺]			
Attribute	PF-SRC				
Name	QRPGLESRC	•			•
Number of chi	2	T			
Source	RSELABXX				
Status	OK	Command Norma	al 🔻		▼ F
Text			_,		
Туре	*FIĻE	Messages		*	Details Show Log
1	•				
RSELABXX/QF	RPGLESRC OBJTYPE	(*FILE::PF-SRC)	1		

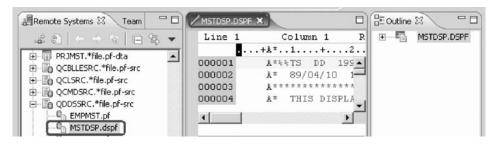
Now you are ready to use the Remote Systems LPEX Editor to edit the member MSTDSP found in QDDSSRC.

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

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

The Remote Systems LPEX Editor opens. It is built right into the workbench, with rich editing functions and is iSeries aware! It is a superset of SEU! The syntax checker is ported from SEU, and the reference manuals are built-in and F1 cursor sensitive.

- 9. Double-click the MSTDSP tab to maximize the Editor window.
- 10. Double-click the MSTDSP tab again to return the view to its original size.



You have viewed and accessed objects in your library and now you are ready to begin "Exercise 2.4: Opening a second source member and multiple views."

Exercise 2.4: Opening a second source member and multiple views

Before you begin, you must complete "Exercise 2.3: Viewing and accessing objects in the Remote System Explorer perspective" on page 14.

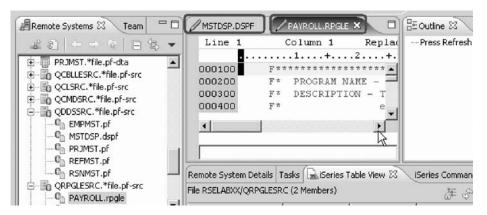
Next let's open a second member in the editor.

To open a second source member:

1. In the Remote Systems view, 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:



Notice the two tabs in the Editor window.

2. Click on each tab to switch from one edit session to another edit session.

Opening multiple views of a source member

You can open multiple views of the same source member while editing in the Remote Systems LPEX editor.

To see multiple views of your RPG source:

- 1. Double-click the **PAYROLL** tab in the editor to maximize the Editor window.
- 2. Right-click this source in the Editor view and click View > Open new view.

Note: You can open a maximum of five views of the same source.

MSTDSP.DSPF	PAYROLL.RPGLE ×			
Line 1	Column	1 R	eplace	
	1 +	2	+3	+ 4
000100	F*******	* * * * * * *	*****	*****
000200	New		Payroll	
000300	14014		ime repo	rting maste
000400	Prompt	F4	externall	y described
000500	Syntax Check Line	9	******	* * * * * * * * * * *
000600	Cut	Ctrl+X	p	
000700	Сору	Ctrl+Insert	found on	CHAIN oper
00800	Paste	Ctrl+V	ror cond	ition
000900		100000000	splay on	delete rec
001000	Select Selected		p request	ed
001100	Deselect	Alt+U	applicat.	ion selecti
001200		AICTU	employee	selection
001300	Filter view		project :	selection
001400	Show all	Ctrl+₩	reason c	ode selecti
001500	Source		, rd	
001600	View		 Open new view 	Ctrl+2
001700			Next view	Ctrl+Alt+Right
001800	Add to Snippets	03	Previous view	Ctrl+Alt+Left
•	Save		Horizontal split	

Right-click a source view and select View > Next view or View > Previous view to navigate among the views.

RSELABXX/QRPGLE		oll):1 Column	1		RSELABXX/QRPGL	cco.c/navn		
Line 1		Column	1		Karren and Skrar	ESKC(PATH	ROLL): 2	
			<u>. т</u> .	Rej	Line 1	(Colum	n 1
		. 1	+	.2			. 1	. + .
000100	F××	* * * * * *	* * * * * *	****	000100	E * *	* * * * *	* * *
000200	Ε×	PROGE	RAM NA	AME -	000200	F *	PROG	RAM
000300	Ε×	DESCH	RIPTIC	DN -	000300	F*	DESC	RIP
000400	E+				000400	F*		
000500	F××	* * * * * *	*****	****	000500	E * *	\star \star \star \star \star	* * *
000600	Ε×	INDIC	CATORS	3 U S	000600	F *	INDI	CAT
000700	F+	50 -	No re	ecoi	000700	F *	50 -	No
000800	Ε×	60 -	Gener	ral	000800	F *	60 -	Ge
000900	Ε×	90 -	Prote	ect	000900	F +	90 -	Pr
001000	Ε×	KC -	End o	of :	001000	F *	KC -	En
001100	Ε×	KD -	Retu	rn t	001100	F *	KD -	Re
001200	Ε×	KE –	Retui	rn t	001200	F*	KE -	Re
001300	Ε×	KF -	Retui	rn t	001300	F *	KF -	Re
001400	Ε×	KG -	Retur	rn t	001400	F *	KG -	Re
001500	Ε×	LR -	Last	rec	001500	F *	LR -	La
001600	F * *	*****	*****	***?	001600	E * *	* * * * *	* * *
001700	F *	SUBRO	DUTINE	es t	001700	F*	SUBR	OUT
<u> </u>				Ľ	•			

4. Right-click a source view and select **View > Horizontal split** to change from a vertical split of the views to a horizontal split of the views.

ESRC(PAVROLL): 1	
	Replace
	2+

F* PROGRAM NA	ME - Pavroll
	N - Time reporting maste
۲ [×]	externally described
F * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *
	1
FSRC(PAVROLL): 2	
restruction and the second s	
Column 1	Replace
Column 1	Replace 2+3+4
Column 1	
Column 1	2+3+4
Column 1 1+ F**************** F* PROGRAM NA	2+4
Column 1 1+ F**************** F* PROGRAM NA	2+3+4 **************************
Column 1 F***************** F* PROGRAM NA F* DESCRIPTIC F*	243+4 **************************

 Right-click the source view that you want to close (RSELABXX/ QRPGLESRC(PAYROLL): 2) and select View > Close view.

Note: This option is not available on the first view.

You have opened another member for edit and seen multiple views of a member and now you are ready to begin "Exercise 2.5: Displaying an outline of a source member."

Exercise 2.5: Displaying an outline of a source member

Before you begin, you must complete "Exercise 2.4: Opening a second source member and multiple views" on page 19.

The Outline view acts as an excellent resource when you want to edit RPG, COBOL and DDS source in the Remote Systems LPEX editor. The Outline view displays a structural outline of items defined in the file that you currently have open in the Remote Systems LPEX editor window. With the editor active, you can expand the file structure in the Outline view, and click various elements in the view to jump to that location in the source itself.

To see an Outline view of your RPG source:

- 1. Click the **PAYROLL** tab in the editor and click **Refresh** on the Outline view toolbar.
 - **Note:** If you have closed the Outline view, you can reset the perspective by selecting **Window > Reset perspective** from the workbench menu or **Show view > Other** then expand **Basic** and click **Outline** in the Show view dialog.

The Outline view contains your source program in a tree view without the lines containing logic.



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

- 2. Expand Files.
- 3. Expand the MSTDSP workstation file.
- 4. Expand the EMPSEL record format.
- 5. Double-click on any of the entries in the Outline view. This will position the source editor accordingly.

Line 187	Column 1	Replace	E Piles	
	CLON01Factor1	l++++++Opcode	e (E) + 📴 🆓 MSTDSP : WORKST	N (
018000	C*		🔳 🗎 🗄 🗄 SELECT	
018100	C* Housekeep	oing, clear di		
018200	C*			he
018300	C EMPTAG	BEGSR	187 (M)	
018400	С	dou		
018500	С	MOVE	ACODE : Cl	
018600	С	MOVE	EMESS : Ch	ara
018700	С	Z-ADD	200	
010000	С	MOVE	EMPMNT	
018900	C* error loop) start	电 登 PRJSEL	
019000	С	dou	电 🗄 PRJMNT	
019100	C	MOVE	电 🗄 RSNSEL	
019200	C	MOVE	E B RSNMNT	
019300	С	MOVE	EMPMST : DISK (Ex	
019400	С	MOVE		
			E CEX RSNMST : DISK (Ex	ter
4			Fields	

6. Stay in the **PAYROLL** tab to get the PAYROLL Editor window in focus for the next exercise.

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 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. Subsystems include:
 - a. iSeries Objects
 - b. iSeries Jobs
 - c. IFS Files
 - d. iSeries Commands
 - e. All of the above
- 3. The subsystem iSeries objects includes:
 - a. Work with libraries
 - b. Work with objects
 - c. Work with members
 - d. Library list
 - e. All of the above
- 4. 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
- 5. The lock icon controls the correlation between the Remote Systems view and the iSeries Table view. (T, F)
- 6. 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).
- 7. The Outline view:
 - **a**. Displays a structural outline of items defined in the file that you currently have open in the editor
 - b. Lists structural elements
 - c. Shows editor specific contents and toolbar
 - d. All of the above

More 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 properties of the connection. Then use the iSeries objects subsystem to list the libraries in your library list. Use the iSeries Table 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 product online help to assist you in these tasks.

Module recap

You have completed Chapter 2, "Module 2. Configuring a connection to an iSeries system and connecting to an iSeries," on page 7. You have learned how to:

- · Create a connection to an iSeries system
- Connect to an iSeries system
- Add a library to your library list
- View libraries in your job's library list from the Remote Systems view

- Find a source physical file in your library
- View members in a source physical file using the iSeries Table view
- · Customize the columns in the iSeries Table view
- Open a member for edit from the iSeries Table view or the Remote Systems view
- Maximize the editor window
- Open another member for edit
- Open multiple views of the same source member
- Switch from one edit session to another edit session
- Display a structural outline of items defined in a source member

Now that you have a connection to an iSeries server and have opened a source member for edit, you can learn more about the Remote Systems LPEX Editor. Continue to the next module Chapter 3, "Module 3. Editing source," on page 25.

Chapter 3. Module 3. Editing source

This module teaches you how to edit ILE RPG source member PAYROLL, which should already be open, and learn about some of the Remote Systems LPEX Editor's language features.

In this module, you will:

- Change the default settings of the LPEX Editor Parsers
- Change the color settings and font used by the Editor
- Change the default behavior of the Enter key
- Use SEU commands to edit source
- Undo and redo source changes
- View language sensitive help for the MOVE operation code
- View a list of all help contents
- · Limit the search of help to specific documents
- Search the help
- Request a prompt for a specification line
- Display context sensitive help for any field in the iSeries Source Prompter
- View the beginning and ending of constructs in your source
- Use the Find and Replace window to search for an item in your source
- Filter or subset your source
- Filter lines based on line type
- Search through members in a source physical file
- · Compare different versions of a program and identify the differences
- Syntax check source by line
- View help on syntax errors

Exercises

The exercises within this module must be completed in order:

- "Exercise 3.1: Introducing the editor" on page 26
- "Exercise 3.2: Changing default editor settings" on page 26
- "Exercise 3.3: Entering SEU commands" on page 33
- "Exercise 3.4: Requesting undo and redo operations" on page 34
- "Exercise 3.5: Invoking language-sensitive help" on page 35
- "Exercise 3.6: Prompting language specifications" on page 40
- "Exercise 3.7: Indenting source" on page 42
- "Exercise 3.8: Finding and replacing text" on page 44
- "Exercise 3.9: Filtering lines by string" on page 46
- "Exercise 3.10: Filtering lines by type" on page 47
- "Exercise 3.11: Searching multiple files" on page 48
- "Exercise 3.12: Comparing file differences from the Remote Systems view" on page 50
- "Exercise 3.13: Comparing files in the CODE Editor (optional)" on page 54
- "Exercise 3.14: Checking syntax" on page 56

Time required

This module will take approximately 45 minutes to complete.

Tip: You might want to maximize the Editor window during this module.

Exercise 3.1: Introducing the editor

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 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

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

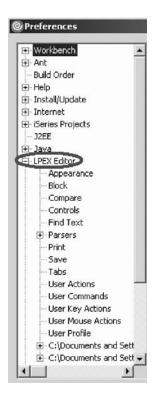
- Token highlighting where different language constructs are highlighted using different colors to help identify them in a program
- SEU-like format-line rulers to show the purpose of each column for column-sensitive languages like RPG and DDS. These rulers can automatically update themselves to reflect 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
- Settings for column-sensitive languages that simplify text insertions and deletions
- On-line language reference

Now you know the features of the editor, you are ready to begin "Exercise 3.2: Changing default editor settings."

Exercise 3.2: Changing default editor settings

Before you begin, you must complete "Exercise 3.1: Introducing the editor."

The LPEX Editor has predefined settings, but also has an associated preferences page containing settings that you can modify. The name of the category is LPEX Editor and it appears in the left pane of the Preferences window.



You will change the default settings of LPEX Editor Parsers, Appearance and User Key Actions.

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.

Setting column sensitive editing

To set preferences:

1. Click **Window > Preferences** from the workbench menu.

The Preferences window opens.

© Preferences		×
Workbench	LPEX Editor Parsers	
Ant Build Order	Show date area	
	Column sensitive editing	
⊕ Install/Update	Add signature to changed lines	
庄 Internet		
😟 iSeries Projects	Signature	
J2EE I+ Java	Resequence start at (1-9999):	100
tiava tiava		
- Remote Systems	Resequence increment by (1-9999):	100
Communications	Resequence lines at save	
. Debug		
File Cache		
Files		
Command Executio		
- Command Subsyste		
- Job Subsystems		
LPEX Editor Parsers		
Objects		
Program Verifiers		
Table View Templates		
Logging -		
		Restore Defaults Apply
Import Export		OK Cancel

- 2. In the left pane of the Preferences window, expand Remote Systems.
- 3. Expand **iSeries** under Remote Systems.
- 4. Select LPEX Editor Parsers under iSeries.

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

5. In the right pane of the Preferences window, select the **Column sensitive** editing check box.

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

6. Click Apply.

The Preferences window shows.

Other interesting preference settings are located under LPEX Editor. Appearance allows you to modify color settings and font used by the Editor.

Changing the editor appearance

- 1. In the left pane of the Preferences window, expand LPEX Editor.
- 2. Select Appearance under LPEX Editor.
- 3. Select formatLine under the Styles list.
- 4. Change the **Foreground** color to dark green.
- 5. Change Font to 12.
- 6. Change **Background** color to light green.

Notice how your changes are reflected in the sample edit view.

🗈 Workbench 📃 🔺	Appearance
🗄 Ant	
Build Order	Palette Font
🗄 Help	white Courier New-regular-12 Change
- Install/Update	
Internet	Styles
- iSeries Projects	formatLine
- J2EE	messageLine
Java	parserDefault Background
LPEX Editor	parserMessage
- Appearance	prefixText
Block	selection
Compare	statusLine
Controls	Preview
- Find Text	Line 1 Column 1 Replace
🕂 Parsers	
Print	+
Save	Emphasis, current line, default te
Tabs	** ERROO1 Embedded parser message.
User Actions	Selection Text in parser default a
User Commands	-
User Key Actions	
User Mouse Actions	LPEX 3.0.3 (build 16-Nov-2005)
- User Profile	
C:\Documents and Sett	J
	Reset Restore Defaults Apply
	Reset Restore Derauits Apply

7. Select currentLine under the Styles list.

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

- 8. Change the **Background** color to light yellow.
- **9**. If you don't like the changes you made, you can click **Restore Defaults** to return to the original settings.

Modifying the default behavior of the Enter key

To modify the default behavior of the Enter key to split the current line in the source instead of creating a new line:

1. Expand LPEX Editor if not already expanded then select User Key Actions.

E Workbench 📃	User Key Action	ns	
Build Order	Кеу	Action	
E Help	enter	splitLine	(Set
🗉 Install/Update	encer	spircente	Delete
Internet			
🗄 iSeries Projects		6	
J2EE		•	
E- Java			
LPEX Editor			
Appearance			
Block			
Compare			
- Controls			
Find Text			
+ Parsers			
Print			
Save			
Tabs			
User Actions		—)	
	Key enter	Enter	
User Key Actions	Action splitLine		
User Mouse Actions			
User Profile			
E-C:\Documents and Sett -	1		
		Reset Restore D	efaults Apply

2. Type enter in the Key field.

Note: The Key and Action fields are case sensitive. Make sure that the values typed in the Key and Action fields are exactly as shown above.

- **3**. Type splitLine in the **Action** field.
- 4. Click Set.

The Remote System Explorer has a predefined behavior for the Enter key. For the Enter key to use the new split line action you need to change the default Remote System Explorer behavior for the ILE RPG language.

🗄 Workbench 📃	ILE RPG
E Ant	Automatic syntax checking
Build Order	
∃-Help ∃-Install/Update	Automatic syntax checking of SQL statements (performed)
- Internet	Automatic uppercasing
- iSeries Projects	Enter key behavior
J2EE	Repeat previous specification type
- Java	Repeat previous operation code (If on fixed form C-Specifical
- LPEX Editor	
Remote Systems	E Set C-Specification cursor position
Communications Debug	Position
File Cache	🔲 Set free form C-Specification curse position
Files	Position
🔁 iSeries	
Cache	Set D-Specification cursor position
- Command Executio	Position
- Command Subsyste	Automatic indent
Job Subsystems	Me Automatic Frideric
LPEX Editor Parsers	Automatic closure of control block
COBOL	User defined tabs(G)
DDS	
(LE RPG)	H: Control
RPG7400 Objects	HKeywords++++++++++++++++++++++++++++++++++++
Program Verifiers	A AA
- Table View	
Templates	nod auanu III
	Restore Defaults Apply

- 5. Expand **Remote Systems**, then **iSeries** if not already expanded and then **LPEX Editor Parsers**.
- 6. Select ILE RPG.
- 7. Clear the **Repeat previous specification type** check box.
- 8. Click OK on the Preferences window.

Return to the Editor window.

Seeing the results of column sensitive editing

Notice that the line the cursor is on is now highlighted in the color you selected for it in the preference setting for highlight current line.

Line 27	Columr	n 10	Insert			
	FFilenam	e++IPE	ASFRlen	+LK1	en+AIDev	ice
002100	F* This	progr	am uses	all	externa	113
002200	F* used	are -	MSTDSP	-	maintens	ince
002300	Ε×	-	EMPMST	-	employee	ma
002400	E×	-	PRJMST	-	project	mas
002500	F×	-	RSNMST		reason c	
002600	F*****	*****	******	****	******	* * * :
002700	FMSTDSP	CF	E		WOF	KS
•						F

To see what column sensitive editing does:

- 1. Move the cursor to line 27, column 10.
- 2. Make sure the editor is in **Insert** mode. If the status area shows **Replace** press the **Insert** key.
- 3. Press the spacebar 3 times.

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

MSTDSP.DSPF	*PAYROLL.RPGLE ×	
Line 27	Column 13 Inser	rt
	FFilename++IPEASFRle	n+LKlen+AIDevice
002100	F* This program use	s all externally
002200	F* used are - MSTDS	P - maintenance
002300	F* - EMPMS	T – employee ma
002400	F* - PRJMS	T - project mas
002500	F* - RSNMS	T - reason code
002600	F******	****
002700	FMST DSP CF E	WORKS
	<u> </u>	
<u> </u>		<u>•</u>

4. Press the **Backspace** 3 times.

Once again the filename is shifted but no other columns are effected.

Seeing the results of splitLine

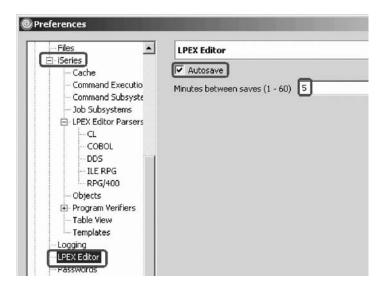
To see what splitLine does:

1. Place the cursor somewhere on a line and press **Enter**.

The text to the right of the cursor is moved to the next line.

Autosaving while editing source

To enable or disable autosave while editing source in the Remote Systems LPEX Editor, you need to change an editor preference. You use the **Window** > **Preferences** > **Remote Systems** > **LPEX Editor** and click the check box **Autosave**. The default value for the minutes is set to 5. You can specify a value between 1 and 60 minutes.



You have changed the default editor settings and now you are ready to begin "Exercise 3.3: Entering SEU commands."

Exercise 3.3: Entering SEU commands

Before you begin, you must complete "Exercise 3.2: Changing default editor settings" on page 26.

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 a 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.

To enter SEU commands:

1. Move the cursor into the gray sequence number area to the left of the edit area.

Line 31	Calumn	1	Turners	. 1	
TTUE 21	Column		Insert	a	
• • • •	1	+	2+.	3	+ 4
102600	F******	****	* * * * * * * *	* * * * * * * * *	*****
id I	FMSTDSP	CF	Е		WORKS
002800	FEMPMST	UF .	A E	K	DISK
002900	FPRJMST	UF .	ΑE	K	DISK
003000	FRSNMST	UF .	ΑE	K	DISK
ld	D*				
03200	D* Compi	le ti	ne arrag	y contai	ning e
					1000 C
4					<u> </u>

- 2. On any sequence number type dd.
- Go down a few lines and type 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.

You have learned how to use SEU commands in the editor and now you are ready to begin "Exercise 3.4: Requesting undo and redo operations."

Exercise 3.4: Requesting undo and redo operations

Before you begin, you must complete "Exercise 3.3: Entering SEU commands" on page 33.

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 Undo operation by using the Redo operation. Finally you will reload the source so that it is back to its original content.

To undo and redo edit changes:

1. Click **Edit > Undo** from the workbench menu.

Notice that the 5 new lines disappear.

- 2. Press **Ctrl+Z** to undo the last change. Notice that the deleted lines reappear.
- **3**. Click **Edit > Redo** from the workbench menu.

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.

Click File > Close on the workbench menu.
 Tip: You can also click the X on the PAYROLL tab.

MSTDSP.DSPF	*PAYROLL.RPGLI ×
Line 26	Column 10 Insert 1 change
	······ 1 .···+····2····+····3····+····4.
002100	F* This program uses all externally
002200	Rt used are - METTED - maintenance

A Save Resource dialog opens asking if you want to save the latest changes.

Save Resource		×
(?) 'PAYROLL.RPGLE	' has been modified. Save changes?	Þ
~		~
	Yes No	Cancel

- 5. Click No.
- **6**. Go back to the workbench to the Remote Systems view and open the PAYROLL member in the QRPGLESRC file.

You have learned how to undo and redo changes that you made to a file and now you are ready to begin "Exercise 3.5: Invoking language-sensitive help."

Exercise 3.5: Invoking language-sensitive help

Before you begin, you must complete "Exercise 3.4: Requesting undo and redo operations" on page 34.

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 allowed values for a variable field. This help is available from the LPEX Editor window.

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

Accessing language sensitive help

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.

Text marked in blue in the Help window contains the link to detailed information about the topic in blue.

Help - IBM Rational Softw	are Develo	pment Platfo	rm			
earch:	GO	Search scope	: All topics			
ontents 🖧 🗄				() ()	1 18	1 🗄
Introduction to WebSp Installing and migratin Navigating and custom Connecting to Linux, U Working in a team eny		PG Lang	juage	Refere	ence	
Developing iSeries ser Developing Web applic Developing Java applic Developing data acces Developing EGL applica	Free- Form Syntax	operatio <u>%CHAR</u> <u>GRAPH</u>	ns, or built-in , <u>%DATE, %</u> <u>%INT, %INT</u>	he <u>EVAL</u> or n functions : <u>DEC</u> , % <u>DEC</u> (H, %TIME, 9 (H, %TIME, 9 (H, %UNS, 0)	such as <u>3H, %</u> %	
Building XML applicatio						
Detecting and analyzir Testing and publishing	Code	Factor 1	Factor 2	Result Field	Indic	ators
Troubleshooting and so Debugging application:	MOVE (P)	Data Attributes	<u>Source</u> <u>field</u>	<u>Tarqet</u> <u>field</u>	+ -	ZB
Reference Glossary	to the res	/E operation sult field. Mor r of factor 2.				
		oving Date, e blank unle				

3. Click the link Date Data Type.

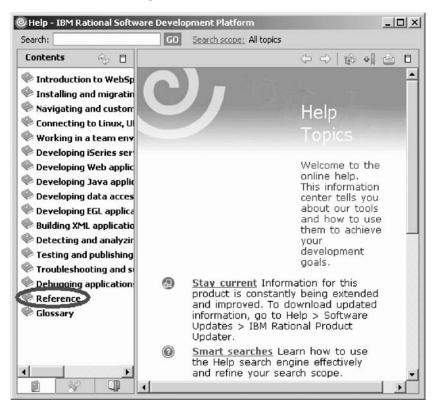
The Help page for Date Data Type is displayed.

-			
Search:		GO Search scope: All topics	
Contents 6	9 8		
Introduction to	websp Date	e Data Type	
linstalling and m	nigratin		
Navigating and	custom Date	fields have a predetermined size and format. They	
Connecting to Li		be defined on the definition specification. Leading	
🏶 Working in a tea	anui	railing zeros are required for all date data.	
Developing iSer		constants or variables used in comparisons or	
Developing Web		inments do not have to be in the same format or use	i.
Developing Java	a applic the s	ame separators. Also, dates used for I/O operations	
Developing data		as input fields, output fields or key fields are also	
Developing EGL	applica conve opera	erted (if required) to the necessary format for the	
🂚 Building XML ap	plicatio	alon.	
Detecting and a	nalyzir The c	lefault internal format for date variables is *ISO. This	
🧇 Testing and pub		It internal format can be overridden globally by the	
🧼 Troubleshooting		ol specification keyword DATFMT and individually	
🧇 Debugging appli	ication by th	e definition specification keyword DATFMT.	
🌳 Reference			
🂚 Glossary		nierarchy used when determining the internal date at and separator for a date field is	
		From the DATFMT keyword specified on the	
	1.	definition specification	
	2.		
•	•	control specification	
		+100	٠ſ

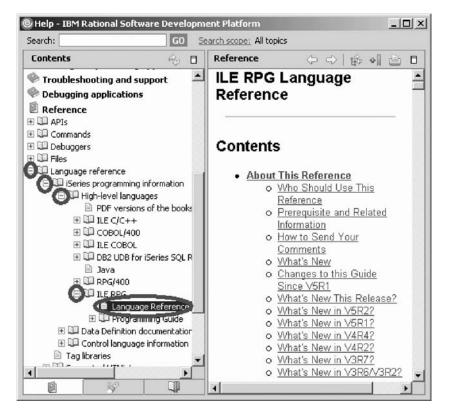
- 4. Play around in the Help window to see what else is available.
- 5. Minimize the Help window.

Note: If you are using WDSC Lite, the following steps do not apply.

6. Select **Help > Help Contents** on the workbench menu to see a list of all help that is available in the product.



- 7. In the left pane of the Help window, click Reference.
- 8. Expand Language Reference.
- 9. Expand iSeries programming information.
- 10. Expand High-level languages.
- 11. Expand ILE RPG.
- 12. Select Language Reference.
- 13. Scroll down to Moving Date-Time Data.



Having the latest version of the manuals at your fingertips will make it easier to find programming information. There is also the option to search the help by specifying a search string. By default, the complete help will be searched.

Limiting the search scope

Note: If you are using WDSC Lite, you can limit the search scope, however, you must choose another topic instead of the Reference topic shown.

To limit the search to specific documents:

1. Click Search scope.

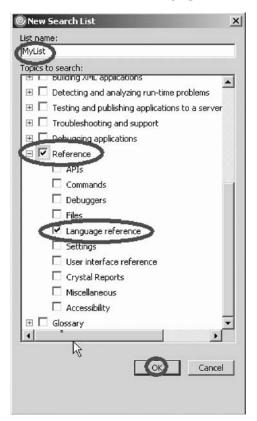
The Select Search Scope dialog opens.

2. Select Search only the following topics radio button.

Help - IBM Rational	Software Developm	ent Platform			<u>× ا</u>
Search:	G0 🤮	earch scope: All topi	cs		
Contents		Reference	<u>~</u> ~	🕸 📲 🗁	٥
Noubleshootin	Select Search Sco	be	x	e	•
🐨 Debugging appi 🚂	Search all topics				
Reference	bearch only the follow	ving topics:			
🗉 💷 APIs					
🗉 💷 Commands					
E Debuggers					
🗄 💷 Files					
🗄 💷 Language refer				e	
🖃 💷 iSeries prog				se This	
🖂 💷 High-ler				00 11110	
🖹 PDI				d Related	
🕀 💷 ILE		475			
E 🛄 co 🌘	New Edit	Remove		our	
E 🛄 ILE 🎽					
E 💷 DB:					
av 🗐				s Guide	
E D RP		OK	Cancel	1	5
				is Release?	~
	ogramming Guide		VITALS INEVAL		
	ition documentation		What's New in		
	iguage information	N1250 ST	<u>Mhat's New ir</u>		
Tag libraries	igaage in ernieden		<u>Mhat's New in</u>		
		2000 C	<u>Mhat's New ir</u>		
1	•		what's New Ir	V3R6/V3R2?	
	ų.	4			

3. Click New.

The New Search List dialog opens.



4. In the List name field, type MyList for example.

- 5. Expand Reference.
- 6. Select the Language Reference check box.
- 7. Click **OK** on the New Search List dialog.

The Select Search Scope dialog reopens again with MyList selected in the topic list.

© Select Searc	cs	-	×
Search only the MyList	e following l	opics:	
New	Edit	Remove	
			Cancel

- 8. Click OK on the Select Search Scope dialog.
- 9. In the Search field, type iSeries and programming for example.

lep - IBM Rational Software Development Platform				_ 🗆 🗵
Search:	iseries and programming	GO	Search scope: All topics	

10. Searching requires a help index and it takes approximately 10 minutes to create the index in your workspace. If you want to complete the search query, click **GO**.

The search results display.

Sear	ch Results 🔅
100%	iSeries programming information
24%	ILE C/C++ Programmer's Guide
23%	DB2 Universal Database for iSer
19%	ILE COBOL Programmer's Guide
19%	WebSphere(R) Development Stu
19%	ILE C/C++ Compiler Reference
18%	ILE C/C++ Compiler Reference
18%	ILE C/C++ Programmer's Guide
15%	ILE C/C++ Programmer's Guide
15%	DB2 Universal Database for iSer
13%	ILE COBOL Programmer's Guide

You have accessed language sensitive help and now are ready to begin "Exercise 3.6: Prompting language specifications."

Exercise 3.6: Prompting language specifications

Before you begin, you must complete "Exercise 3.5: Invoking language-sensitive help" on page 35.

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.

Accessing language prompts

To access prompts:

- 1. Return to the workbench. In the Editor window move your cursor to the D-spec on line 33.
- 2. Press F4 (You can also click **Source** from the workbench menu and then click **Prompt**.)

You see the iSeries Source Prompter at the bottom of the workbench. The iSeries Source Prompter shows the specification line broken down into its individual fields.

MSTDSP.DSPF	PAYROLL.RPGLE	23			2 -0
Line 33	Column	6	Replace		Ś
	DName++++	++++	++ETDsFrom) Global Definit
003100	D*				Files
003200	AND A STREET STREET STREET	le ti	me array c		
003300	DERR		S		Main Procedu
003400	D EMESS		S		G Subroutir
•			•		
Tasks iSeries Table \	/i iSeries Comma	Remote S	ystem 🗹 iSeries	Source	×
					**
Line type: D: Defin	nition				3
Alama	Externally Da			Те	A anath
Name	Externally Da described	ta structu type	type From	To,	/Length
Name ERR				то,	/Length
ERR		type •	type From		5
ERR	described	type •	type From		5
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw
ERR	described	type T ETDsF	type From	.+++ID	5 c.Keyw

On the iSeries Source Prompter toolbar you can use the three push buttons to: Disable source prompt view, Disable syntax checking, and Change to insert mode.

Displaying context sensitive help

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

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

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

Help - IBM Rational Soft	ware Development Platform
Search:	GO Search scope: NyList
Contents 💫 🗄	() () () () () () () () () () () () () (
Introduction to WebSp	ILE RPG Language Reference
🂚 Installing and migratir	
Navigating and custom	Positions 44-80 (Keywords)
Connecting to Linux, U	
learning in a team env	Positions 44 to 80 are provided for definition specification
Developing iSeries ser	keywords. Keywords are used to describe and define
Developing Web applic	data and its attributes. Use this area to specify any
Developing Java appli	Reywords necessary to long define the field.
Developing data acces	Deminior-Opecification regwords
Developing EGL applica	
Building XML application	AL TOFOCHIONED
Detecting and analyzing	A ASCEND
Testing and publishing	 BASED(basing_pointer_name)
Troubleshooting and s	
Debugging application	V VLADU VATA class-hamer
Reference	CONST{(constant)} CTDATA
-	 DATEMT(format(separator))
🆤 Glossary	• DESCEND
	 DIM(numeric_constant)
	 DTAARA((*VAR:)data_area_name}
	 EXPORT{(external_name)}
	EXTFLD(field_name) EXTEMT(code)

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.
- 5. In the workbench, double-click the **PAYROLL** tab to return the Editor window to its original size.

You have used prompts to enter code and now are ready to begin "Exercise 3.7: Indenting source."

Exercise 3.7: Indenting source

Before you begin, you must complete "Exercise 3.6: Prompting language specifications" on page 40.

When editing ILE RPG source, it can be difficult to determine the beginning and ending of constructs. The iSeries Indent view allows you to see your source with constructs in an indented mode.

To indent source:

 Click Source > Show Indentation from the workbench menu. The iSeries Indent view opens at the bottom of the workbench.

		~			
MSTDSP.DSPF	PAYROLL.RI	PGLE 🕅			2 -0
Line 32	Colu	umn 3	Replac	e	Ś
	1 .	+	. 2 +	3 🖂 🤅	👌 Global Definit
003100	D*				Files
003200	D* Co	mpile t	ime array		Fields
003300	D ERR		S		🗄 🖳 Indicator: 🗟 Main Procedu
003400	D EMES	s	S	and the second sec	😸 Main Procedu 🗄 🕮 Subroutir
•					
				_	
				•	
Tasks iSeries Tab	iSeries Co R	emote Syst	iSeries Sou	iSeries Ind	×
PAYROLL.RPGLE		omoto bystini	Dones Doarn	- Tibertes stitem	
Line 32	Colum		Insert		
	+				
-	******	*****	******		
FMSTD		E		WORKSTN	1
FEMPM		AE		DISK	
FPRJM		0.0		DISK	
FRSNM D*	ST UF	ΑE	K	DISK	
-	ompile ti			ing or	cox doc
D ERR	Contract A Contract Contract Contract	s s	y concari	50	DIM(1
D ENK		s		50	DIN(1
E*	55	2		00	
C****	****	*****	*****	******	*****
C* M	AINLINE C	ALCULAT	IONS		
C****	******	*****	******	******	******
•					•
					1

You can display the iSeries Indent view as a full view.

2. Double-click the **iSeries Indent** tab.

asks i Sěries Ta AVROLL.RPGLE	ble iSeries Com	Remote Syst	e	iSeries Sourc	Series Indent 🗙 🦵 🕯
	C . lum		т		
Line 15	and and a second s	n 11		nsert -3+	8: 45
+	- T+	2+-		-3+	
C*				1	
C				IF	not *INKC
C	RSCDE		1	CHAIN	
C				EXSR	ACDESR
C C				ELSE	-D
c				LEAVES	ĸ
			i la	END	
C* C	end of erro	r Toob	- 6	ENDDO	-
C*			I.	ENDDO	
C*	Display we		-l -		
C*	Display re	ason co	de	maintenanc	e iormat
C			e e	EXFMT	RSNMNT
C*		I		EXEMT	RENMAT
c			- 6	TF	*INKD
C				LEAVESR	TINED
c				END	
c					*INKG
c				I ITER	THKG
c				END	
C*		I.	5	END	
C*	Determine	undata	mo	de and nerf	orm record a
C*	Decermine	update	mo	de and perr	orm record a
C		i.	1	IF	not *inkc
		1		1 L	noc inc
					<u>ار</u>

3. Scroll down to line 150. The Row shown on the status bar is the cursor position.

In the iSeries Indent view you see some nested conditions with indented lines. As you will notice this helps to recognize the beginning and ending of these conditions.

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

- 4. Click the X in the top right corner of the view to close the iSeries Indent view.
- 5. This returns you to the Editor window with the PAYROLL program.

You have used the Indent view to see the beginning and ending of constructs and you are ready to begin "Exercise 3.8: Finding and replacing text."

Exercise 3.8: Finding and replacing text

Before you begin, you must complete "Exercise 3.7: Indenting source" on page 42.

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:

1. Press **Ctrl+Home** to go to the top of the file.

Tip: When you press **Ctrl+Home** to go to the top of a file or **Ctrl+End** to go to the bottom of a file, a quick mark is set at your cursor position. This allows you to return to that line by pressing **Alt+Q**. **Ctrl+Q** will set a quick mark.

 Click Edit > Find/Replace from the workbench menu or press Ctrl+F. The Find/Replace window opens at the bottom of the Editor window.

MSTDSP.DSF				Ð
Line !	52	Column 26	Replace	
3	CLC	N01Factor1	+++++++ <mark>O</mark> pcode (E)+Factor2+++
005200	С	MAIN	BEGSR	
005300	C		aou	*inkc
005400	С		EVAL	*IN60 =
005500	C		EVAL	EMESS =
005600	C		EVAL	EMPAPL =
005700	С		EVAL	PRJAPL =
005800	С		EVAL	RSNAPL =
005900	C*			
006000	C*	Write the	SELECT format	to display
006100	C*			
006200	C*			
006300	C		DOU	not *IN6
006400	С		EXFMT	SELECT
006500	С	*INKC	IFEQ	'1'
006600	C		LEAVESR	
006700	С		ELSE	
006800	С		EXSR	EDITSL
006900	С		END	
007000	C*			
007100	C*	IF the ge	neral error ind	icator *IN
007200	C*	program c	ontinues to loo	p 🚽
•				•
Find BEGS	ર			ext Pre <u>v</u> ious <u>A</u> ll
Replace				Replace Replace all
the second s	Cold States and the states of the		expression ♥ Wrap Sele ch to columns Start column 1	and the second

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.
- 4. Make sure the **Replace** field is blank.

You would use this field for text replacement.

The Editor moves the active line to line 52, which contains the first BEGSR phrase in the file.

- 5. Click Next to go to the next location of BEGSR in the file.
- 6. Click in the Editor window to close the Find/Replace window.

You have searched for an item in your source using the Find/Replace window and now you are ready to begin "Exercise 3.9: Filtering lines by string" on page 46.

Exercise 3.9: Filtering lines by string

Before you begin, you must complete "Exercise 3.8: Finding and replacing text" on page 44.

The 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.

To filter source by string:

- 1. Double-click the variable EMPAPL in the Editor window.
- 2. Select Edit > Selected > Filter Selection from the workbench menu.

ć		RPGLE 🗙	PAYROLL.	MSTDSP.DSPF
	Replace	lumn 36	Co	Line 56
E)+ <mark>E</mark> xtende	+++++Opcode (E	N01Factor1+	CLO	
			т	Ð
EMPAPL	EVAL		- <u>C</u>	+ 005600
itenance)	(employee main	If EMPAPL	C*	007900
EMPAPL	when		С	🗐 008800
'X'	IFEQ	EMPAPL	С	035800
'X'	OREQ	EMPAPL	С	036000
, ,	IFEQ	EMPAPL	С	037300
	IFNE	EMPAPL	С	038300
'X'	ANDNE	EMPAPL	С	₦ 038400

- 3. Move the cursor down a few lines to line 79.
- 4. Expand line 79. This expands the section up to the next instance of EMPAPL.

MSTDSP.DSPF	PAYROLL.	RPGLE ×		B
Line 79	Col	lumn 1	Replace	
		. 1 +	2+3	+4
l A				
🗄 005600	С		EVAL	EMPAPL =
(E) 607900	C*	If EMPAR	PL (employee mai	ntenance) e
💛 ឋ០៩០០០	C*	branches	to label EMPTA	.G.
008100	C*		PL (project main	
008200	C+	branches	to label PRJTA	G.
008300	C*	If the p	rior two tests	were not si
008400	C*	reason c	ode maintenance:	. The proc
008500	C^{\star}	next exe	cutable operati	on.
008600	C*			
_ 008700	С		SELECT	
008800	С		when	EMPAPL =
035800	С	EMPAPI	, IFEQ	'X'
036000	С	EMPAPI	OREQ	'X'
037300	С	EMPAPI	, IFEQ	т т
038300	С	EMPAPI	IFNE	т т
038400	С	EMPAPI	ANDNE	'X'

Now you want to show the entire source again.

 Click Edit > Show all from the workbench 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. You have filtered your source so that you see only lines containing a given string and now you are ready to begin "Exercise 3.10: Filtering lines by type."

Exercise 3.10: Filtering lines by type

Before you begin, you must complete "Exercise 3.9: Filtering lines by string" on page 46.

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.

Line 24		Column 46	Repla	ice	
• *•		1+2	+	3 + 4 .	+ <mark>.</mark>
000100	F***	* * * * * * * * * * * *	* * * * * * *	* * * * * * * * * * * * * * * * *	* * * * *
000200	Ε×	PROGRAM NAM	E - Pay	roll	
000300	F *	DESCRIPTION	- Time	e reporting mas	ster f
000400	F *		exte	ernally describ	oed wo
000500	F * :	New		************************************	* * * * * *
000600	F×	Prompt	F4		
000700	F *	Syntax Check Line	- 20 Mar 1997	i on CHAIN op	perati
00800	F* -			-condition	
000900	F×	Cut	Ctrl+X	y on delete n	ceques
001000	F×	Сору	Ctrl+Insert	lested	
001100	F* _	Paste	Ctrl+∀	cation selec	ction
001200	Ε×	Select		> yee selection	n
001300	F *	Selected		▶ect selection	n
001400	F×	Deselect	Alt+U	on code selec	ction
001500	F*	Filter view		Date	
001600	F + :	Support International	Ctrl+W	Code	*****
001700	F* -			Comments	
001800	F*	Source		Control t	cion d
001900	F*	View		FIOLEGUIES	ll mai
002000	E+:	Add to Snippets		SOL Statements	*****
002100	F* _		2	Subroutines	/ desc
002200	F *	Save			e disp
002300	F*	Show Date Area		employee ma	
002400	F×	-	PRJMST	- project mas	ster f
•					<u> </u>

To filter lines by type:

- 1. Right-click in the Editor window with the PAYROLL program.
- 2. Click **Filter view > Subroutines** on the pop-up menu.

All source lines with BEGSR or ENDSR are displayed allowing you to move quickly and easily to the desired subroutine in your file.

I	ine 1	02 0	Column	1	Replace	
		C	LONO1F	actor1	++++++Opcode (E))+Factor2
Ð						
Đ	00520	0 C	М	AIN	BEGSR	
Ð	01020	0 C			endsr	
Ð	01100	0 C	R	SNTAG	BEGSR	
Ð	01760	0 C			ENDsr	
Ð	01830	0 C	E	MPTAG	BEGSR	
Ð	02570	0 C			ENDSR	
Ē	02640	0 C	P	RJTAG	begsr	
Ð	03350				endsr	
Ð	0347)0		E	DITSL	BEGSR	
H	04040	0 C	A	CDESR	BEGSR	
Ð	04270				ENDSR	
Η	04310		A	DDCDE	BEGSR	
ŧ	04400				ENDSR	
H	04440		C	HGCDE	BEGSR	
H	04540				ENDSR	
ŦŦŦŦ	04610		BEGSR	DELCD	Ε;	
H	04720		ENDSR	;		
Ш	04760			INVCD	Ε;	
H	04790	0	ENDSR	;		

- **3.** Move your cursor to the line with the subroutine declaration CHGCDE (line 444).
- 4. Expand the declaration to show all lines in this subroutine. Now you could work with the source inside this subroutine.
- 5. Right-click in the Editor window and click Show all on the pop-up menu.

You have filter lines in your source by line type and now you are ready to begin "Exercise 3.11: Searching multiple files."

Exercise 3.11: Searching multiple files

Before you begin, you must complete "Exercise 3.10: Filtering lines by type" on page 47.

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. Click **Search > iSeries** from the workbench menu.

Remote System	Explorer - PAYRO	L.RPGLE - IBM	Rational Softw	are Development 💶 🗙
File Edit Source	Compile 🛛 Navigate 🕻	Search Project	Run Window	Help
] 11 • 13 14] • *•			Ctrl+H	😰 🔚 Remote Syste
	PAYROLL.RPG	iSeries		8
Line 446	Colun			1- (D) 1 D - 1 - 0 - 1 -
043200	CLUNUIFa	ICTOLT+++	TP	de (E) +Factor2+++

The Search window opens.

2. In the **Search string** field, type ENHRS.

earch string	h \$\$° dW Search \$\$° Help Search \$\$° Java Searc ▼ + □ Case sensitive
onnection: s400a	
Target	
Library RSELABXX	Browse
File: QDDSSRC	Browse
Member *ALL	Browse
Source members	Data members
Columns	
 All columns 	
C Between 1 and(G)	30
C Between(H) 1 and er	id of line

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

- 3. Under Target in the Library field, type RSELABxx.
- 4. Under **Target** in the **File** field, type QDDSSRC to search all members in this source physical file.
- 5. Under Target in the Member field, select *ALL.
- 6. Click Search.

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

Tasks	iSeries	iSeries	Remote	s	iSeries	iSerie	s I 🕈	Remote S	. ×	-	E
"ENHR	S" - 3 Occur	rences in 3	Members					C	23	× ·	-
	RSELABXX	(QDDSSRC)	(EMPMST)	(1 m	atch)						
	(Line :	18) A	ENHRS	R							
E-0	SELABXX	(/QDDSSRC((MSTDSP)	(1 ma	atch)						
1 1	(Line 9	99) A	ENHRS	R	B 11 3	37REFFLD(F	RCEMP/EN	HRS *LIBL/E	MPMST)	
ė-0	RSELABXX	(QDDSSRC)	REFMST)	(1 ma	atch)						
1	(Line 2	27) A	ENHRS	3	1 T	EXT('EMPL)	OVEE NOR	MAL WEEK H	IOURS'	5	

7. Double-click the last line in the list.

A ENHRS 3 1 TEXT('EMPLOYEE NORMAL WEEK HOURS')

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

MSTDSP.DSPF	PAYR	OLL.RPGLE	REFMST.PF	25			
Line 27	1	Column 1	Rep	lace			
	A		Name+-	++++RLei	1++T	DpB	
000026	А						
000027	А		ENHRS		3	1	
000058	A						
000029	А		EPHRC		5	1	
000030	А						
000031	А		EPHRY		7	1	
000032	А						
000033	А		EPHRP		7	1	
000034	А						
000035	А		EPNRC		5	1	
000036	А						
000037	А		EPNRY		7	1	
000038	А						
000039	А		EPNRP		7	1	
000040	А						
000041	А		EHWRK		5	1	
000042	А						
000043	A*	PROJECT		RELATED	FIE	LDS	
000044	А		PRCDE		8		
000045	А						
000046	А		PRDSC		50		
000047	А						
000048	А		PRRSP		30		
000049	А						
000050	А		[PRSTR	R			
•			-				
							<u> </u>

8. Click the X in the **REFMST** tab to close the REFMST file.

You have searched through members in a source physical file and now you are ready to begin "Exercise 3.12: Comparing file differences from the Remote Systems view."

Exercise 3.12: Comparing file differences from the Remote Systems view

Before you begin, you must complete "Exercise 3.11: Searching multiple files" on page 48.

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 two ways to do a 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.

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.

Note: Make sure all lines show in the source before starting the Compare tool.

To compare files in the workbench:

1. Click **Window > Preferences** from the workbench menu.

The Preferences window opens.

2. In the left pane of the Preferences window, expand LPEX Editor.

© Preferences		_ 🗆 ×
	Compare	
- Appearance Block	Ignore case	
Compare	AIgnore leading blanks	
Controls		
- Find Text	Ignore trailing blanks	
Parsers	Ignore all blanks	
Print	Ignore sequence numbers	
Save	Ignore comments	\mathbb{R}
Tabs	1 Ignore commerce	· v
User Actions		
User Commands	Column restricted comparison	
User Key Actions	Start column 1	
User Mouse Actions		
User Profile	End column 80	
● C:\Documents and Sett		
Remote Systems		
⊕ Run/Debug		
Server		
Validation		
Web and XML		
Web Diagram		
WebFacing		
		Reset Restore Defaults Apply
Import Export		OK Cancel

3. Click Compare under LPEX Editor.

In the right pane of the Preferences window make sure that the **Ignore blanks** check boxes are selected.

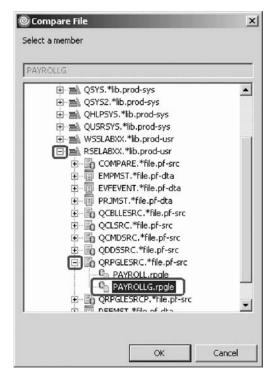
- 4. Click **OK** in the Preferences window.
- 5. Back in the Editor window of the PAYROLL member double-click the **PAYROLL** tab.

1	Undo	Ctrl+Z	18	ES EBRer	note Syste
4	Redo	Ctrl+Y			
108	" Cut	Ctrl+X	REFMST.PF ×		6
	Copy	Ctrl+Insert	Replace		
	Paste	Ctrl+V	.Name+++++	RLen++TI)pB
ι ι 🕷	Delete	Delete	ENHRS	3	1
C	Select All	Ctrl+A		_	-
0	Find/Replace	Ctrl+F	EPHRC	5	1
L c	Find Next	Shift+F4	EPHRY	7	1
r -	Find Previous		LERKI	1	1
č	Find Other Show All	Ctrl+W	EPHRP	7	1
۲—			-		
C	Add Bookmark Add Task		EPNRC	5	1
2-			EPNRY	7	1
r r	Select		DENKI	,	1
C	Selected Deselect	Alt+U	EPNRP	7	1
۲.		ALCTO .	-		
-	Mark		EHMRK		1
	Compare Keystroke Recorder	•	Compare to file	CtN+Shift+N	_
C C	Show Tooltip Descrip	tion F2	Next Mismatch Previous Mismatch	Ctrl+Shift+P	
c	Content Assist	Ctrl+SPACE	Refresh	Ctrl+Shift+R	
000	0046 A		Clear		
001	0047 A				
	0048 A		PRRSP	30	
	0049 A				
000	0050 A		PRSTR	R	
•					D
					_

6. Click **Edit > Compare > Compare to file** on the workbench menu.

The Compare window opens.

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



The editor now will show the differences of these two members PAYROLL and PAYROLLG .

You can move from mismatch to mismatch by using the Compare menu under the Edit menu.

Mismatches in PAYROLL and PAYROLLG are highlighted in different colors so that you know where the mismatched lines are in each file.

MSTDSP.DSPF		REFMST.PF	8
Line 52	Column 1	Replace	
	+1+	.2+	+4
004200	C* another.	The action co	de you sele <mark>.</mark>
004300	C* determine	es if the progr	am will add
004400		n existing reco	
004500	C*******	****	*********
004600	C* Housekeep	oing, clear dis	play fields
004700	C*		
004800	C	EXSR	MAIN
004900	C* If MAIN is	done program	ends
005000	C	eval	*INLR =
005100	* MAIN SUBRO	DUTINE	
005200	C MAIN	BEGSR	
000053	C	dou	*INKC
005300	C	dou	*inkc
005400	C	EVAL	*IN60 =
005500	C	EVAL	EMESS =
005600	C	EVAL	EMPAPL =
005700	С	EVAL	PRJAPL =
005800	C	EVAL	RSNAPL =
005900	C*		
006000	C* Write the	e SELECT format	to display
006100	C*		
006200	C*		
006300	C	DOU	not *IN6
006400	C	EXFMT	SELECT
006500	C *INKC	IFEQ	'1'
1			<u>.</u>
One or mo	ore mismatches f	ound.	
1			

- **13**. Click **Ctrl + Shift + N** to find the next mismatch. Next, end the compare session.
- 14. Click **Edit > Compare > Clear** from the workbench menu.
- **15**. Double-click the **PAYROLL** tab to return the Editor window to its original size.

You have compared different versions of the program and found the differences and now you are ready to begin "Exercise 3.13: Comparing files in the CODE Editor (optional)."

Exercise 3.13: Comparing files in the CODE Editor (optional)

Before you begin, you must complete "Exercise 3.12: Comparing file differences from the Remote Systems view" on page 50.

The CODE tool provides a side-by-side view of the members being compared. If you prefer this type 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.

To compare files in the CODE Editor:

- 1. In the Remote Systems view, right-click member PAYROLL in QRPGLESRC.
- 2. Click **Open With > CODE Editor** on the pop-up menu.

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

- 3. In the CODE Editor, open the PAYROLLG member.
- Click File > Open from the CODE Editor menu. The Select file - Open for Edit dialog opens.

🔀 Select file - Open for Edit		×
File name	<pre><r0s400a>RSELABX</r0s400a></pre>	•
Contains sequence <u>p</u> umbers	Subset by source type < All Files >	•
<pre><r0s400a>RSELABXX/QRPGLES</r0s400a></pre>	Contents of ' <r0s400a>RSELABXX/QRPGLESRC(")'</r0s400a>	
	PAYROLLG RPGLE	<u>त</u>
ROS400A>RSELABXX/QRP	aleanoparnoleaj	

- 5. In the left pane of the Select file Open for Edit dialog expand the **R0S400A** connection (the Remote System Explorer connection).
- 6. Expand library RSELABxx.
- 7. Select file QRPGLESRC.
- 8. In the right pane of the Select file-Open for Edit dialog, select member PAYROLLG from the member list.
- 9. Click OK.
- Click Actions > Compare from the CODE Editor menu. The Compare dialog opens:

Compare			×
<r0s400a>RSE</r0s400a>	LABXX/QRPG	LESRCIPAYR	OLLG]
Select or enter a	i file name		
<r0s400a>RSE</r0s400a>	LABXX/QRPO	GLESRC(PAYF	(OLL)
<r0s400a>RSE</r0s400a>	LABXX/QRP	GLESRC(PAYF	ROLLJ
•			•
Compare	Cancel	Help	

All entries are preloaded.

11. 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.

	R05400A>RSELAE /iew Actions Comp		RC(PA Wind		-		<u>_ ×</u>
han	B * * *	Da. (B. \ +	1.0		989	19 18 0	
< 805400A	RSELABXX/ORP	GLESRC(PA		< R 05400A	RSELA	BXX/ORPGL	ESRC(PA
Row 1	Column 1	Repla		Row 1	C	olumn 1	Rep1.
	1 +	2				.1+	2
000001	F******	******		000100	F**	****	*****
000002	F* PROGRA	M NAME 🗍	111	000200	F*	PROGRAM	NAME
000003	F* DESCRI	PTION -	1 H	000300	F*	DESCRIPT	ION -
000004	F*		18.8	000400	F*		
000005	F******	******		000500	F**	*******	****
000006	F* INDICA	TORS US		000600	F*	INDICATO	RS US
000007	F* 50 - N	o recor		000700	F*	50 - No	recor
000008	F* 60 - G	eneral		008000	F*	60 - Gen	eral
000009	F* 90 - P	rotect		000900	F*	90 - Pro	tect
000010		nd of j		001000	F*	KC - End	l of j
000011		eturn t		001100	F*	KD - Ret	urn t
000012		eturn t	LН	001200	F*	KE - Ret	urn t
000013	F* KF – R	eturn t		001300	F*	KF - Ret	urn t
000014		eturn t		001400	F*	KG - Ret	urn t
000015	F* LR – L	ast rec		001500	F*	LR - Las	t rec
000016	F*****	*****		001600	F**	*******	****
000017		TINES U		001700	F*	SUBROUTI	
000018		- Edit		001800	F*	EDITSL -	Edit
000019		- Edit		001900	F*	ACDESR -	Edit
000020	F*****		li i	002000	F**	*******	
000021		rogram		002100	F*	This pro	
000022		re - MS	1 i i	002200	F*	used are	
000023	F*	- EM	IН	002300	F*		- EM
000024	F*	- PR		002400	F*		- PR
•		F	1-	•			•
IBM Live	Parsing Edit	or		IBM Live	Parsi	ng Editor	

12. 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. You will fix these in a few moments.

- 13. Click Ctrl + Shift + N to find the next mismatch.
- 14. Click **Compare > Exit compare** from the CODE Editor menu (which was inserted while performing this action).

The CODE Editor window re-opens.

Close the CODE Editor.

15. Click **File > Exit** from the CODE Editor menu. The workbench re-opens. Continue working in the workbench.

You have compared files using the CODE Editor and now you are ready to begin "Exercise 3.14: Checking syntax."

Exercise 3.14: Checking syntax

Before you begin, you must complete "Exercise 3.13: Comparing files in the CODE Editor (optional)" on page 54.

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 source member.

Now you will create a syntax error and watch for the prompt to correct it.

To syntax check the file:

1. In the PAYROLL Editor window move the cursor to line 211 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.

MSTDSP.DSF	PF /*PA	VROLL.RPGLE	×	REFMST.P	F		8
Line 2		Column			ace 1		
2	CL	ON01Eact	tor	1++++++	-Opcod <mark>e</mark>	(E) + Fac	ctor2+++
020100	C*	leave s	sub	routine			-
020200	C*	T					
020300	C				IF	*II	NKD
020400	С				leaves:	r	
020500	C				END		
020600	C*						
020700	C*	Access	s ei	mployee	master	to va	lidate
020800	C*						
020900	C				IF		r *ink
021000	C	EMPI	NO		CHAIN		PMST
021100	С			1967 - 197	EXSRX		DESR
	RNF5014	E Operat	tio	n code i	and the second sec	valid;	speci
021200	-				ELSE		
021300	С				LEAVESI	R	
021400	C				END		
021500	C*	end of	er	ror loop			
021600	C				ENDdo		
021700	C*			1			-
021800	C* C*	Displa	ay (employee	e mainte	enance	rorma
021900					DVDMD	DM	DMAL
022000	C				EXFMT IF		PMNT
022100		end of s			ΤF	^ I I	NED
022200	c c	end of s	sub.	routine	LEAVESI	D	
022300					END	r.	
022400					PIAD		-1
•							
10	Т	1		1			

- 4. Move the cursor onto the pink error message.
- 5. Press F1.

-	vare Development Platform	11.2
Search:	GO Search scope: NyList	
Contents 🗞 🗄	クウ 鈴・1 色	٥
 Introduction to WebSp Installing and migratin Navigating and custon Connecting to Linux, UI Working in a team env Developing iSeries ser Developing Web applic Developing data acces Developing EGL application Detecting and analyzir Testing and publishing Troubleshooting and si Debugging application: Reference Glossary 	RNF5014 Operation code is not valid; specification is ignored. 30 Cause : The Operation code is not valid. No further checking is done on the specification. The operation must start with a valid Operation Code and can be followed by an optional Operation Extender. If specified the Operation Extender must be one or more of A, D, E, H, M, N, P, R, T, or Z, and enclosed within parentheses. Recovery : Specify a valid operation, or omit the specification. Compile again. (C) Copyright IBM Corporation 1992, 2005. All Rights Reserved.	

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. (Similarly, you can use **Edit > Undo** to correct this).
- 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. Click **Window > Preferences** from the workbench menu and then expand **Remote Systems, iSeries, LPEX Editor Parsers** and select the language you want to change the settings for in the left pane of the Preferences window, select the **Automatic syntax checking** check box and then click **OK**.

Tip: You can syntax check the whole source member currently in the editor by clicking **Source > Syntax Check All**.

-Save	ILE RPG	
Tabs User Actions	Rutomatic syntax checking	
- User Commands	Automatic syntax checking of SQL statements (performed on the server	
User Key Actions		ments (performed on the serve
- User Mouse Actions	Automatic uppercasing	
User Profile	Enter key behavior	
← C:\Documents and Sett	Repeat previous specification type	
E C:\Documents and Sett	Repeat previous operation code (if on fixed form C-Specification)	
C:\Documents and Sett	Set C-Specification cursor position	on management of the second of the
Remote Systems	 Der cropeningenom carsor bosition 	
Communications Debug	Position	
- File Cache	Set free form C-Specification cursor position Position	
Files		
😑 Series		
Cache	Set D-Specification cursor position	
- Command Executio	Position	
- Command Subsyste	Automatic indent	N. 1. 4
- Job Subsystems	Macomatic Indenc	Blanks (1-
E- LPEX Editor Parsers	Matematic closure of control block	S
COBOL	E there defend to be (2)	
DDS	User defined tabs(G)	
ILE RPG	H: Control	Ψ.
RPG/400	HKevwords++++++++++++++++++++++++++++++++++++	+++++++++++++++++++++++++++++++++++++++
- Objects	4	
Program Verifiers	A AA	
- Table View - Templates	And every: 0	
Logging		
	Res	store Defaults Apply
→ → → → → → → → → → → → → → → → → → →		

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 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 preferences 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)

- 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 allowed 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 the _____ key in an Editor window. Name the key:
 - a. F2
 - b. F3
 - **c**. F1
 - **d**. F4
- 9. 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. The _____ view allows you to see your source with constructs in an indented mode. By default, the _____ view will show the indented view in the bottom pane.
 - a. Outdent
 - b. iSeries Indent
 - **c**. Edit
- 12. The iSeries Indent view is browse mode only and cannot be edited. (T, F)
- **13**. You use the _____ window to search for an item in the current source. Choose the best answer.

- a. Search
- b. Find
- c. Edit
- d. Find/Replace
- 14. 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
- **15**. 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
- **16.** 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. both
- 17. 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
- **18**. 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
- **19**. There are two ways to compare files. They are Compare tool in the workbench and the Compare tool in the CODE Editor. (T, F)
- **20.** 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 source member. (T, F)

More practice

Given your experience 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 product online help to assist you in these tasks.

Module recap

You have completed Chapter 3, "Module 3. Editing source," on page 25. You have learned how to:

- Change the default settings of the LPEX Editor Parsers
- Change the color settings and font used by the Editor
- Change the default behavior of the Enter key
- Use SEU commands to edit source
- Undo and redo source changes
- · View language sensitive help for the MOVE operation code
- View a list of all help contents
- · Limit the search of help to specific documents
- Search the help
- Request a prompt for a specification line
- Display context sensitive help for any field in the iSeries Source Prompter
- · View the beginning and ending of constructs in your source
- · Use the Find and Replace window to search for an item in your source
- Filter or subset your source
- · Filter lines based on line type
- Search through members in a source physical file
- · Compare different versions of a program and identify the differences
- Syntax check source by line
- View help on syntax errors

Now that you have mastered editing source, you can move on to verifying 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! Continue to Chapter 4, "Module 4. Verifying and compiling source," on page 63.

Chapter 4. Module 4. Verifying and compiling source

This module teaches you how to verify and compile RPG in the Remote Systems LPEX Editor. When errors are found by either the verify or the compile step, the iSeries Error List appears. The iSeries Error List is a powerful tool that manages errors found by verify and compile utilities. You will become familiar with these tools, the various capabilities of the iSeries Error List and the RPG program that you have created.

In this module, you will:

- · Check for semantic errors on your workstation
- Start the Program Verifier tool
- Use the iSeries Error list to locate each error in the source
- Use content-assist to fix an error
- Save your source
- Re-verify source
- Change compile preferences
- Invoke the compile command
- · Change the current library using the Command field in the iSeries Table view
- Start an interactive connection
- Invoke the payroll program

Exercises

The exercises within this module must be completed in order:

- "Exercise 4.1: Verifying the source"
- "Exercise 4.2: Compiling source remotely" on page 67
- "Exercise 4.3: Submitting iSeries commands in the iSeries Table view" on page 71
- "Exercise 4.4: Running commands and programs" on page 73

Time required

This module will take approximately 20 minutes to complete.

Exercise 4.1: Verifying the source

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 view 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:

 Click Source > Verify from the workbench menu. (Similarly, you can also use the pop-up menu for the source member in the Remote Systems View or the Verify tool button — you need the source in the editor for the button to appear.)

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

Series 1	Tabl	iSeries Com	iSeries Sour	Remote 9	Search Ta	sks 🏭 i S	eries Erro 🗙	Remote Syst	- 6
<toras< th=""><th>507M></th><th>RSELABXX/QRP</th><th>SLESRC(PAYROL</th><th>L)</th><th></th><th></th><th></th><th>***</th><th>⇒ ▼</th></toras<>	507M>	RSELABXX/QRP	SLESRC(PAYROL	L)				***	⇒ ▼
	ID		Message		Severity	Line	Location	Connection	
	RNES	5178	ENDSR operati	on is	30	1	RSELAB	s400a	
	RNF7	060-	The name or in	dicato	30	365	RSELAB	s400a	
	RNF7	030	The name or in	dicato	30	95	RSELAB	s400a	
	RNF7	018	The operand R	SNTA	30	95	RSELAB	s400a	
	RNFS	5184	Factor 2 must	be '0'	30	364	RSELAB	s400a	
	RNF7	515	Move operand:	s ERR	30	365	RSELAB	s400a	
i	RNF7	089	RPG provides S	Separa	0	27	RSELAB	s400a	
i	RNF7	031	The name or in	dicato	0	110	RSELAB	s400a	
i	RNF7	031	The name or in	dicato	0	1	RSELAB	s400a	
i	RNF7	031	The name or in	dicato	0	1	RSELAB	s400a	
i	RNF7	031	The name or in	dicato	0	1	RSELAB	s400a	
i	RNF7	031	The name or in	dicato	0	1	RSELAB	s400a	
i	RNF7	031	The name or in	dicato	0	1	RSELAB	s400a	

The error list shows you:

- The error message itself
- The severity
- The line number
- The source location
- The connection name

Fixing errors

Next you will fix the errors in your source.

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

1. Double-click 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 should be EMESS. One good way of finding the correct name is the content-assist.

MSTDSP.DSPF	/*PAYROLL.RPGLE ×			8
Line 366	Column 51	Replace	3 changes	
or1+	+++++Opcode (E)+Factor2++	+++++Result+	+++
036300 ****	****			-
036400	MOVE	'1'	*IN60	
036500	MOVE	ERR (2)	EMES	
EMESS char(50)	TEEO	Demess Dempno Dempapl Demcat Dedept Delocn Dename Dename		
037300 PL	IFEQ	14		
\mathbf{R}	# N I I W I I	T. T.		

Note: The Outline view must be populated to use content-assist.

2. Select the misspelled variable and press CRTL+spacebar.

If the variable starts correctly the selection presented contains the correct name.

3. Double-click the variable EMESS in the list to correct the variable name. Another way to find the variable name is to use the Outline view and see what variables are declared.

E Outline ×	69 -	-
🖯 🎯 Global Definitions		+
i ⊕ - C Files		
Fields		
🗄 🕀 ACODE : Cha		
🕀 🕀 ACREC : Cha	racter (1)	
📋 🗄 🖆 EDEPT : Char	acter (5)	_
ELOCN : Cha	racter (30)	
EMCAT : Cha		
EMESS : Cha	and the second se	
EMPAPL : Ch	• •	
EMPNO : Pac		1
ENAME : Cha		
i ∰ ∰ ENHRS : Pad		
EPHRC : Pack	그만 집 것 같아? 집 같이 많았	-

The next error is a RNF7030 as well.

- 4. Double-click RNF7030. Fix it in the editor.
- 5. RSNTAX should really be RSNTAG.
 - Make the appropriate change.
- 6. Go to the next error RNF5184.

The next error is actually RNF7018 but it is related to the one you just fixed and can be ignored. It shows the same line number, which is an indication that both errors are related.

- 7. Double-click RNF5184.
- 8. Fix this by replacing the 2 with a 1.

Error RNF7515 is related to the first error you fixed. It has the same line number. You can ignore it. The only serious remaining error is RFN5178. This error is caused by a missing ENDSR.

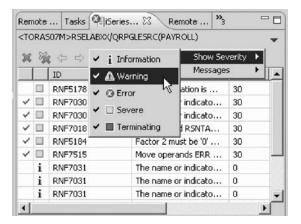
- **Note:** The verifier could not determine where the ENDSR is missing so the line number reported is 1, for this reason you can't just double-click on the error message. You need to investigate where the missing statement belongs. You can use the Indent view to determine where the ENDSR is missing.
- 9. Move to line 398 in the editor.

Line 398	Co	l 11m	n 8	R	eplac	e f	i ch	ang	28	-
				2						
039700	C				EN					
039800	C*D				EN	DSR				
039900	<u> </u>									
040000	C****	***	****	* * * * * *	****	* * *	* * * *	* * *	* * *	* * *
040100	C* A	CDE	SR	subrou	tine	ver	ifie	s t	he ·	tim
040200	C* a	11	mai	ntenan	ice se	lec	tion	s.		
040300	C****	***	***	*****	****	* * *	* * * *	* * *	* * *	* * *
040400	С	AC	CDESI	R	BE	GSR	~			
040500	C*									
040600	С* Н	ous	seke	eping,	clea	r d	ispl	ay	fie	lds
040700	C*			T. 1970						
040800	C				EV	AL		EM	ESS	=
040900	C				EV	AL		* I	N60	=
041000	C				EV	AL		* I	N90	=
041100	C*									
041200	С* Т	he	fol	lowing	sta	tem	ents	pe	rfo	rm
041300	C* t	wo	fun	ctions	. Fi	rst	the	y d	ete	rmi
041400	C* r	equ	lest	ed and	bran	che	to	the	ap	pro
041500	C* t	hey	/ det	termin	e if	the	mai	nte	nan	ce
041600	C*									
041700	С				SE	LEC	Т			
041800	C				WΗ	EN		AC	ODE	=
041900	С				EX	SR		AD	DCD	Е
042000	С				τJΗ	EN		AC	ODE	=
042100	С				EΧ	SR		CH	GCD	Е
4										+

10. Remove the asterisk (*) from the C-spec.

You can use the **Tab** key to quickly move to the appropriate column. All the non-informational errors are now fixed.

You can filter out messages according to these severities by using the filter menu.



- 11. Click the arrow in the iSeries Error List title bar.
- 12. Click Show Severity on the pop-up menu.
- 13. Clear the severities you don't want to see in the list (Warning for example).

Saving the source member

Now before you loose any of your changes, it's a good idea to save them. Make sure the 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. Click **File > Save** from the workbench menu.
- 2. Click the Save icon 📓 in the workbench toolbar.
- 3. Press Ctrl+S.

Changes are uploaded to the iSeries.

4. Verify your source again.

iSeri	ies 1	Tabl	iSeries Com	iSeries Sour	Remote S	Search '	Tasks	× iSe	eries Erro 🗴	Remote Syst.	5
<tc< th=""><th>RAS</th><th>507M>I</th><th>RSELABXX/QRPG</th><th>SLESRC(PAYROL</th><th>L)</th><th></th><th></th><th>1</th><th></th><th>***</th><th>\$ -</th></tc<>	RAS	507M>I	RSELABXX/QRPG	SLESRC(PAYROL	L)			1		***	\$ -
		ID		Message		Severit	y Lir	ne	Location	Connection	
	i	RNF7	089	RPG provides S	epara	0	27	7	RSELAB	s400a	
	i	RNF7	031	The name or in	dicato	0	1		RSELAB	s400a	
	i	RNF7	031	The name or in	dicato	0	1		RSELAB	s400a	
	i	RNF7	031	The name or in	dicato	0	1		RSELAB	s400a	
	i	RNF7	031	The name or in	dicato	0	1		RSELAB	s400a	
	i	RNF7	031	The name or in	dicato	0	1		RSELAB	s400a	

Everything should be ok. You should see only severity 0 messages. You are ready to compile the program.

You have verified your source and fixed any errors and now you are ready to begin "Exercise 4.2: Compiling source remotely."

Exercise 4.2: Compiling source remotely

Before you begin, you must complete "Exercise 4.1: Verifying the source" on page 63.

The remote compile capability is part of the Remote System Explorer. It gives you a workstation interface to submit requests to compile, bind, or build objects on the iSeries 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 view, 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.

Changing compile preferences

To change the preferences to run the compile interactive:

1. Click **Window > Preferences** from the workbench menu.

- Workbench	Command Execution
E Ant	Command Executive Preferences
Build Order	Command Executely Preferences
⊡-Help ⊡-Install/Update	
- Internet	Preferences for compiles and user action variables
- iSeries Projects	Object library: *SRCLIB
J2EE	Replace object
E-Java	A
E-LPEX Editor	C Compile in batch
Remote Systems	
Communications	Preferences for batch compiles, commands, and user action variabl
⊕ Debug	Job description library:
File Cache	
Files iSeries	Job description: *USRPRF
Cache	SBMJOB additional parameters:
- Command Executio	
- Command Subsyste	Add batch compiles to the iSeries Job Status view
- Job Subsystems	Add batch commands to the iSeries Job Status view
- LPEX Editor Parsers	
μa	Preferences only for compile commands
COBOL	Precerences only for complie commands
DDS	Compile member types in this order:
ILE RPG	PF
RPG/400	LF Move Up
Objects	DSPF
	ICEE
- Templates	RPGLE Move Down
	Restore
	Restore

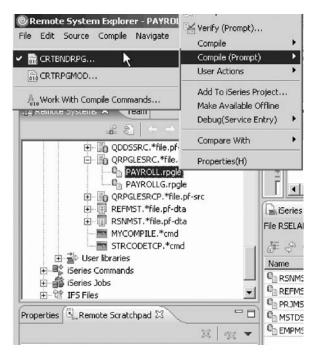
- 2. In the left pane of the Preferences window, expand Remote Systems.
- 3. Expand iSeries under Remote Systems.
- 4. Click Command Execution under iSeries.
- 5. In the right pane of the Preferences window, clear the **Compile in batch** check box.
- 6. Click OK to return to the Remote System Explorer perspective.

Invoking the compile command

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.

To compile source:

1. Right-click the PAYROLL member in QRPGLESRC.



- 2. Click **Compile (Prompt) > CRTBNDRPG** on the pop-up menu. The Create Bound RPG Program (CRTBNDRPG) dialog opens.
- 3. In the **Debugging views** list, select the *ALL parameter.

Create Bound RPG Program	(CRTBNDRPG)	x
Program:	> PAYROLL 💌	Name
Library:	> RSELABXX	Name
Source file:	> QRPGLESRC	Name, QRPGLESRC
Library:	> RSELABXX	Name
Source member:	> PAYROLL 💌	Name
		×
Source stream file:		
		v
Generation severity level:	10	0-20
Text 'description':	*SRCMBRTXT	Character value
Default activation group:	*YES 💌	
	Advanced Parameters	
Compiler options:	> Add	
	*EVENTF Remove(B)	
	Move up	
	Move down	
Debugging views:	> * ALL	
Replace program:	> *YES 💌	
Advanced(5)	All Parameters(6)	Keywords(7)
CRTBNDRPG PGM(RSELABXX/PAYR DBGVIEW(*ALL) REPLACE(*YES)	OLL) SRCFILE(RSELABXX/QRPGLESRC) SRCMBR(PAYRO	LL) OPTION(*EVENTF)
		estore defaults Cancel

If you want to see the other parameters available, click **Advanced**.

4. Click **OK** when you are finished.

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 information messages.

If you are not sure that the compile was successful, you can check the iSeries Commands log.

			- G	. •
s400a				
1				-
CDTONO				
		rselabxx/payroi X/orpglesrc))	
) OPTION(*EVENT	F)	
		EPLACE(*YES) placed in library R:		
		reated on 01/04/0		
14:03:3				
		ogram PAYROLL w d in library RSELA		
		everity that result		
	ram creat	ion date and time	are 01/04/05	in and a

5. 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 returned for each command.

You have set compile preferences, invoked the compile command, checked for a successful compile and now you are ready to begin "Exercise 4.3: Submitting iSeries commands in the iSeries Table view."

Exercise 4.3: Submitting iSeries commands in the iSeries Table view

Before you begin, you must complete "Exercise 4.2: Compiling source remotely" on page 67.

You can use the iSeries Table view inside the Remote System Explorer to submit commands to the iSeries. You can run commands from 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 click either **Prompt** to specify parameters and then **Run** or just click **Run**. When you run a command, the **Messages** field is populated with the messages from the command. When you select a message, the **Details** button is enabled. When you click this button, the message and its help is displayed.

Also note that besides the iSeries Table view, you can also use the Remote Systems view to run commands and programs. Which one you choose depends on your personal preference. 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.

In the **Commands** field, you select where you want to run the command. The choices are Normal, which means that the command will run in the RSE communication server job, Batch or Interactive.

To change the library list:

asks Remote	iSeries	Remote	iSeries	iSeries	Series	X iSeries 🗖 🗖	
ile RSELABXX/C	QDDSSRC (5 Me	mbers)			ä	E 🗇 🗘 🗢 🖻 🔻	
Name	Туре	Attrib	ute T	ext	Sta	Last modified	
EMPMST	PF	SRC			ОК	January 3, 2005 3:3	
MSTDSP	DSPF	SRC			OK	January 3, 2005 3:3	
PRJMST PF		SRC			OK	January 3, 2005 3:3	
REFMST	FMST PF SRC				ОК	January 3, 2005 3:3	
🖞 RSNMST	PF	SRC			OK	January 3, 2005 3:3	
✓ Command [r Command]	Normal 🗾	(chgcurlib rs	selabxx			Prompt Run	

- 1. Click the **iSeries Table View** tab from the views at the bottom of the workbench.
- 2. In the **Command** field type, CHGCURLIB RSELABxx for example.

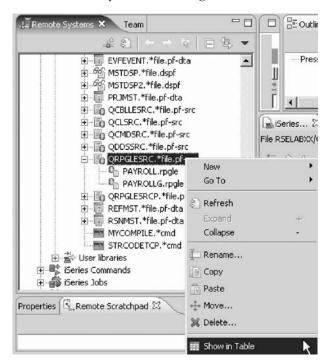
Note: Use a library that is on your iSeries system.

3. Click Run.

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

If you see this message, click **OK**.

a. In the Remote Systems view, right-click QRPGLESRC.

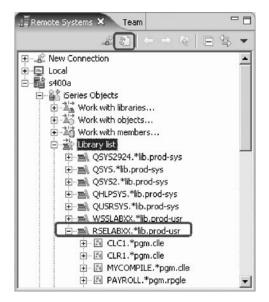


- b. Click Show in Table on the pop-up menu. The iSeries Table view is now populated with the member in the selected source file.
- c. Run the CHGCURLIB command again.

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

rasks 🛛	Remote	iSeries	Remote	iSeries	iSeries	(iSeries	. 🗶 iSeries 🗖 🖡
ile RS	ELABXX/QRP	GLESRC (2 M	1embers)			6	王命○○王 ▼
Name	,	Туре	Attribu	ite Te	xt	Sta	Last modified
È PA	YROLL	RPGLE	SRC			OK	January 4, 2005 1:4
PAYROLLG	RPGLE	SRC			OK	January 3, 2005 3:3	
					R		
<u> </u>				-			
⊆o	mmand Norr	nal 💌					Prompt Run
C Ma	ssages Curr	oot libraru o	banged to DS	EL ADVV		-	Details Show Log

4. Back in the workbench in the Remote Systems view, right-click **Library list** and click **Refresh**.



- 5. You will see a small green asterisk beside the RSELABxx library to indicate it as your current library.
- 6. 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[™].

You have submitted a command to change the current library in the command line of the iSeries Table view now you are ready to begin "Exercise 4.4: Running commands and programs."

Exercise 4.4: Running commands and programs

Before you begin, you must complete "Exercise 4.3: Submitting iSeries commands in the iSeries Table view" on page 71.

As you know already, 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. This is the one you are using currently.
- 2. In a batch job.
- 3. In an interactive job (to test 5250 applications).
- 4. In a server 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, and you need to first run a STRRSESVR connectionName command to associate the emulator with a particular connection in the Remote System Explorer communications server. A multi-threaded debug session creates a new server job and this way keeps the RSE communications server job free for other tasks.

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.

	- 1 1	tion Actions ¹⁰ Wi		: B @ 4	2	
MAIN			/400 Main Menu			
Select or	ne of the fol	lowing:			System:	TORAS644
3. 0 4. F 5. F 6. 0 7. C 8. F 9. C 10. 1 11. 0 90. S Selectior	Programming Communication Define or cha Problem handl Display a mer	ies, and fold is inge the syste ing issistant opt: /400 tasks	em			
	F4=Prompt	F9=Retrieve	F12=Cancel	F13=Informa	tion Assi	istant

3. In the command line, type the command

STRRSESVR connectionName

4. Press Enter.

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.

Running the payroll program

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

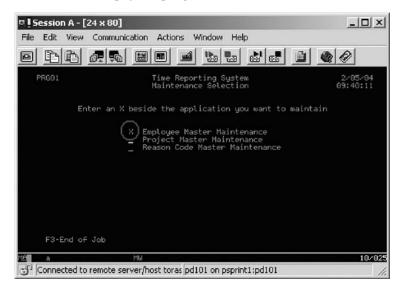
Return to the workbench.

To run the program:

Remote Systems X	Update Change Save Restore User Actions	•	Press Ra	efresh to up	alate
⊕- <u>=</u> > QHLPSYS ⊕- <u>=</u> > QUSRSY: ⊕- <u>=</u> > WSSLAB: ⊕- <u>=></u> RSELAB> ⊕⊡ CLCI	Run As Run(Prompt)(B) Debug(Service Entry) Debug As(J) Debug (Prompt)(K)	Bat	eractive	in RSE Job	
È-E CLRI-	Dura autor du D	Na	me	Туре	At
	Properties(H) LL.*pgm.rpgle:		PAYROLL	RPGLE	SF
PAYRO PAYRO P-E PAYRO	LLC2.*pgm.cblle LLC2.*pgm.cblle LLC.*pgm.cblle LLG.*pgm.rpgle DECPP.*pgm.clp	e b	PAYROLLG	RPGLE	SF

- 1. In the Remote Systems view, locate the PAYROLL program that you created.
- 2. Right-click the PAYROLL program.
- 3. Click **Run As > Interactive** on the pop-up menu.
- 4. Switch to your 5250-emulation session.

You will see the payroll program menu.



- 5. Type x beside **Employee Master Maintenance**.
- 6. Press Enter.
- 7. Type 234 for the Employee Number.
- 8. Type A for the Action Code to add employee 234.
- 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 applications.
- 14. To get control of the interactive job, right-click **iSeries Objects** and click **Release Interactive Job** on the pop-up menu.

You can also choose to disconnect a session. You would right-click the connection and click **Disconnect** on the pop-up menu.

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 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 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. From the iSeries Table view you can:
 - a. List and sort libraries, objects, and members
 - b. Copy, rename, delete, edit, compile and run (for programs), open with and browse (for members) in the view from the pop-up menu
 - c. Transfer files from one system to another
 - d. All of the above
- 7. 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. Multi-threaded
- e. All of the above
- 8. Interactive programs require a 5250 emulator, and you need to first run a STRRSESVR connectionName command to associate the emulator with a particular connection in the Remote System Explorer. (T, F)

More practice

Given your experience using the Program Verifier and Compile command, and that you have your own 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 product online help to assist you in these tasks.

Module recap

You have completed Chapter 4, "Module 4. Verifying and compiling source," on page 63. You have learned how to:

- Check for semantic errors on your workstation
- Start the Program Verifier tool
- Use the iSeries Error list to locate each error in the source
- Use content-assist to fix an error
- Save your source
- Re-verify source
- Change compile preferences
- Invoke the compile command
- Change the current library using the Command field in the iSeries Table view
- Start an interactive connection
- Invoke the payroll program

Now that you know how to verify and compile source from the Remote Systems view and run the payroll program interactively, you can move onto debugging your payroll program. Continue to Chapter 5, "Module 5. Debugging a program," on page 79.

Chapter 5. Module 5. Debugging a program

This modules teaches you how to debug a CL and ILE RPG program.

You will learn how to start the debugger, set breakpoints, monitor 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.

In this module, you will:

- · Invoke the debugger from the Launch Configurations window
- Add a breakpoint
- Add a conditional breakpoint
- Edit a breakpoint
- Monitor a variable through the Monitors view
- Step into your payroll program
- Show a Listing view
- List the call stack entries in the Debug view
- View all breakpoints
- Remove a breakpoint
- Monitor storage
- Set a Watch breakpoint
- Close the debugger

Exercises

The exercises within this module must be completed in order:

- "Exercise 5.1: Introducing the Integrated iSeries Debugger" on page 80
- "Exercise 5.2: Starting the integrated debugger" on page 80
- "Exercise 5.3: Setting breakpoints" on page 85
- "Exercise 5.4: Monitoring variables" on page 88
- "Exercise 5.5: Stepping into a program" on page 93
- "Exercise 5.6: Listing call stack entries" on page 95
- "Exercise 5.7: Setting breakpoints in PAYROLLG" on page 95
- "Exercise 5.8: Removing a breakpoint in PAYROLLG" on page 97
- "Exercise 5.9: Monitoring variables in PAYROLLG" on page 98
- "Exercise 5.10: Adding a memory monitor" on page 99
- "Exercise 5.11: Setting Watch breakpoints" on page 100
- "Exercise 5.12: Closing the debug session" on page 102

Time required

This module will take approximately 30 minutes to complete.

Exercise 5.1: Introducing the Integrated iSeries Debugger

The Integrated iSeries Debugger is a source-level debugger that enables you to debug and test an application that is running on an iSeries system. 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 view.
- Set Watch breakpoints to make the program stop whenever a specified variable changes.
- View the call stack of your program in the Debug view. 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 view. 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 Modules/Programs view.
- Debug multi-threaded 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 $RPG/400^{\otimes}$ and ILE RPG, COBOL and ILE COBOL, C, C++ and CL.

In the following exercises you will be given the opportunity to learn about some of the basic features of the Debugger. For the purpose of these exercises you will debug a CL and an ILE RPG program. Don't worry if you don't know RPG.

You know the basic features of the debugger and now you are ready to begin "Exercise 5.2: Starting the integrated debugger."

Exercise 5.2: Starting the integrated debugger

Before you begin, you must complete "Exercise 5.1: Introducing the Integrated iSeries 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 exercises in Chapter 3, "Module 3. Editing source," on page 25.

You can start the Debugger in several ways: directly from the pop-up menu of a program or service program in the Remote Systems view, or from a Launch Configurations window. Starting directly from the Remote Systems view doesn't

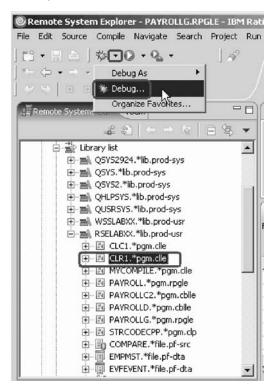
allow you to specify parameters to be passed to the program. The Launch Configurations window allows you to modify how the program is invoked and to specify parameters.

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.

To start the debugger:

1. In the Remote Systems view expand the **Library list** filter, if it isn't expanded already.



- 2. Expand library RSELABxx, if it isn't expanded already.
- 3. Select program CLR1 in library RSELABxx.
- 4. Click the arrow beside the **DEBUG** icon ***** on the workbench toolbar.
- Select **Debug** from the list.
 The Debug Launch Configurations window opens.
- 6. Select iSeries: Debug Interactive Application under the Configurations list.

Debug	A LANGE THE REAL PROPERTY OF THE REAL PROPERTY	
reate, manage, and run confi	gurations	Ť
Apache Tomcat Attach to a Running Pro Compiled Application Host Java Application HTML Script Application Series: Debug Batch Ap Series: Debug Job Series: Debug Job Series: Debug Multi-Thr Java and Mixed Langua Java Applet Java Application Java Application Java Application Java Bean Ju JUnit Remote Java Application Script Host Application Script Host Application WebSphere v5.0 Applic. WebSphere v6.0 Applic.	Perspectives These settings associate a perspective with iSeries: launch configurations. A different perspective may launch mode, and can optionally be activated when when a breakpoint is encountered via the Debug p perspective switch should not occur, select "None". Debug: Debug Restore Defaults	be associated with each supported a configuration is launched or eferences. To indicate that a
New Delete		Apply Revert
		Debug Close

- 7. Click New.
 - **Note:** You could also use **Debug(prompt) > Interactive** from the pop-up menu.

The right pane of the Debug Launch Configurations window opens.

What To Debu		tar⊵ ¥∕ Source	Advanced
ograms or servic	e program to be d	ebugged	Add
RSELABXX	CLR1	*PGM	Edit
			Remove
Step into Terminate deb Ipdate produc	ug session on prog ition files	ram completion	

- 8. In the Name field, type the program name CLR1.
- 9. Select the **Update production files** check box.

10. Click the How To Start tab.

The How To Start page opens.

e: CLR1		
What To Debug	How To Start	Advanced 🖂 Common
ommand to start appl	ication:	
CALL PGM(RSELABXX)	(CLR1)	2
		2
	Prompt	

By default, the page contains a call for the program selected in the Remote Systems view.

11. Click Prompt.

The Call Program (CALL) window opens.

Call Program	n (CALL)							x
Program:	> CLR1			Name				
Library:	> RSEL	ABXX	- 1	Name				
Parameters:		>					Add J	
CALL PGM(RSEL/	ABXX/CLR1) P		arameters		∏ Keyw	ords		-
								K F
			OK	Re	estore default	s	Cancel	

12. In the **Parameters** field, type 'XX' where 'XX' is your workstation number. Click **Add**.

The parameter value will appear in the lower list.

If you forget to click **Add**, the parameter will be missing from the CALL program command and you will receive an error when the program tries to call the payroll program.

13. Click OK.

The complete start command for the program appears.

un configurations	-
ion What To Debug How To Start Source stion cation Command to start application: tch Ap teractic Command to start application: tch Ap ization D b b b b b lib:Thr angua n iSeri Altication Applic.	e Advanced Common
<u> </u>	Apply Revert
	Name: CLR1

14. Click Debug.

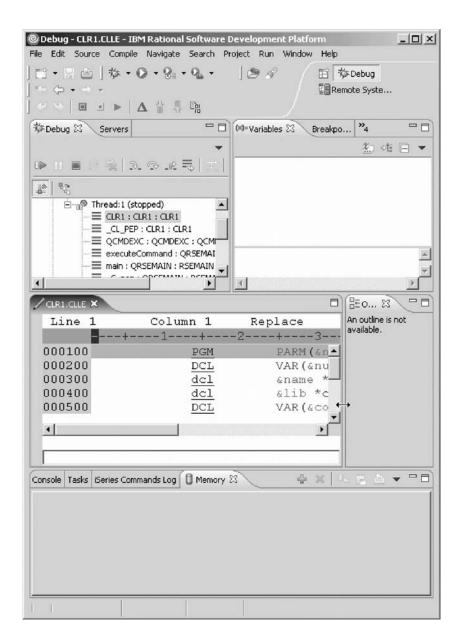
The Debug perspective opens.

If not, you may see this error message.

		nection. Associate an in	iteractive
job with this connec	tion to continue.	Cancel	<< Details
A request to run an interact connection s400a, but no in To associate an interactive 1. Open a 5250 em	teractive job is current job with this connectio	tly associated with this o	
This will associate	nand STRRSESVR NAME e the interactive job will eractive command or pr	th this connection, caus	ing this

The Remote System Explorer communications server has been shut down in the meantime. Go to your 5250 emulator and restart the Remote System Explorer communications server following the instructions in the message. You don't have to cancel the message. It will be removed as soon as the connection between the Remote System Explorer communications server and the interactive session has been established. Now the Debug perspective is loaded in the workbench.

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



You have started an interactive debug session and now you are ready to begin "Exercise 5.3: Setting breakpoints."

Exercise 5.3: Setting breakpoints

Before you begin, you must complete "Exercise 5.2: Starting the integrated debugger" on page 80.

You can only set breakpoints at executable lines. All executable lines are displayed in blue. One way to set a breakpoint is to right-click on the line in the Source view.

To set a breakpoint:

- 1. Position the cursor on line 11.
- 2. Right-click anywhere on line 11.

	Add Breakpoint Add watch Breakpoint	4
Line 11	Run To Location	8+4
000800	Monitor Expression	10) then (go
000900	Monitor Memory	IAME) VALUE
001000	Switch View	, ib) value(
001100	CALL	GM (alib/ PAYROLL
001200		
	1	
4		<u> </u>

3. Click Add breakpoint on the pop-up menu.

A dot with a checkmark in the prefix area indicates that a breakpoint has been set for that line. The prefix area is the small grey margin to the left of the source lines.

Now you add a conditional breakpoint to stop in the loop when it loops the 99th time.

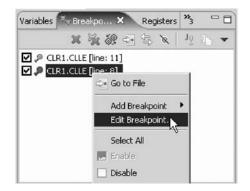
Adding a conditional breakpoint

To add a conditional breakpoint:

- 1. Position the cursor on line 8.
- 2. Right-click on line 8.
- 3. Click Add breakpoint on the pop-up menu.

Line 7	Column 1 Rep	place
+	1+2	-+4
000700 LOOP.	chaver	ver(scount) velu
000800	if (&count	*lt 100) then(gc
000900	CHGVAR	VAR (GNAME) VALUE
001000	chgvar	var(&lib) value(
001100	CALL	PGM (&lib/PAYROLL
<u>•</u>		<u>•</u>

4. Click the **Breakpoints** tab in the upper right pane of the Debug perspective. The Breakpoints view opens.



5. In the **Breakpoints** view right-click the breakpoint for line 8.

6. Click **Edit Breakpoint** on the pop-up menu. The Edit a Line Breakpoint window opens.

🖉 Edit a Line Breakpoint	x
Required information	-
Sets a breakpoint to stop execution at a specific source line	Sor
Executable (Optional)	
J Object (Optional)	
l Source:	
Line:	
8	
< Back Next > Finish	Cancel

7. Click Next.

ptional paramet				-
Make the breakpoi	ic condicional dp	on the followi	ng parameters	3Ôr
Frequency				
From: 99				
To: Infinity			-	
Every: 1			- 1	
210171] 2				
xpression:				

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.

- 8. Under Frequency in the From field, type 99.
- 9. Click Finish.

You have added a breakpoint including a conditional breakpoint to your source and now you are ready to begin "Exercise 5.4: Monitoring variables" on page 88.

Exercise 5.4: Monitoring variables

Before you begin, you must complete "Exercise 5.3: Setting breakpoints" on page 85.

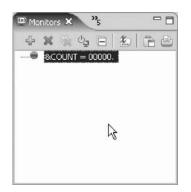
You can monitor variables in the Monitors view. Now you will monitor the variable &count.

To monitor a variable:

1. In the Source view, double-click the variable &count.

CLR1.CLLE ×	P	Ionitor Expression		l
Line 7	Þ	Ionitor Memory	NG 🕨	
	-+·	iwitch View	▶ 8-	<mark>+</mark> 4-
000700 LO	OP:	cngvar	vartaco	unt) valu
000800		if (&count	*lt 100) then (go
000900		CHGVAR	VAR (&NA	ME) VALUE
001000		chgvar	var(&li	b) value(
001100		CALL	PGM (&li	b/ PAYROLL
41				
<u> </u>				<u>_</u>

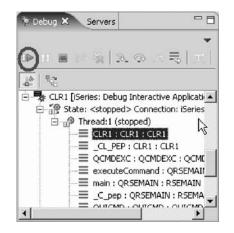
- 2. Right-click &count.
- **3**. Click **Monitor Expression** on the pop-up menu. The Monitors view opens.



The variable appears in the Monitors view. 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.



4. Click the **Resume** icon 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 the **Resume** icon again.

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

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

If you do not see the error message below, go to "Exercise 5.5: Stepping into a program" on page 93.

Error Handling

If you forget to add the parameter to the CALL program command when you debug the program, you will see this error message.



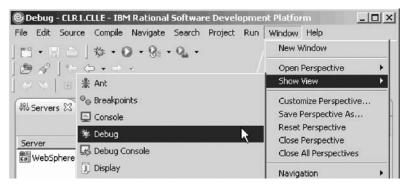
- a. Click OK.
- b. Click the **Terminate** icon on the Debug toolbar.

The debug session terminates on the workstation but the exception waits for input from the 5250 emulation session.

If you closed the Debug view by mistake, you will need to re-open the Debug view and then terminate the debug session on the workstation.

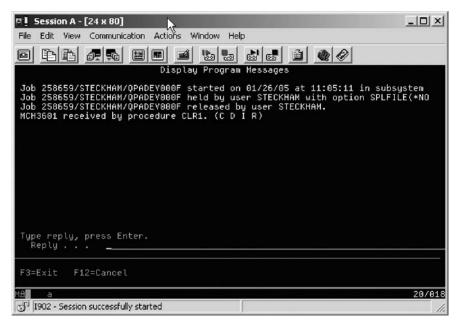
Opebug - CLR1.CLLE - IBM Rational	Software Develop	ment Platfo	rm	_ 🗆 ×
File Edit Source Compile Navigate	Search Project I	Run Window	Help	
🖆 • 🗄 🖻 🅸 • O • 9a •	Q 109	8	E A Debug	
$ \leftarrow \diamond \cdot \rightarrow \cdot$			Remote Syste	
] <> > ■ ■ ▶ ∆ 凿 Ŗ	Foi			
			rs 🕄 "4	
AN Servers 🖾		rs 💷 Monito		
\$ 0 ¢ %			× 4 E X	
Server	Host name	&COUNT = 0	0100.	
l.				
			E Outline 🕅	
			An outline is not avai	
Line 10 Colum	ın ⊥ – R€	eplace	All outline is not avai	aule.
	100 million (100 million)	1922		
000900	CHGVAR	VAR		
2 001100	<u>chgvar</u> CALL	var PGM -		
4	CALL	FGIT		
		and \$2.	a a s	
Console X Tasks iSeries Comma	and the second se	- 24		* U
Process: 587975/STECKHAM/QPADEV000	D Program: CLR1			
				_
				-
II I				

Click Window > Show View > Debug.



Remember to terminate the Debug view if you haven't done so already.

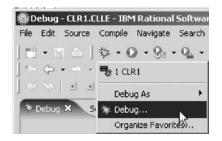
c. Go to your 5250 emulator and press Enter until the program messages complete and the Remote System Explorer communications server screen appears again.



d. In the workbench, click the **Remove all terminated launches** icon on the Debug toolbar to clean up the Debug view.



- e. Click the arrow beside the Debug icon on the workbench toolbar to start a new debug session.
- f. Select **Debug** from the list.



The Debug Launch Configurations window opens.

Connection: s40			▼ New
tor motilor in Tarte			
	e program to be d		
Library	Program	Туре	Add
RSELABXX	CLR1	*PGM	Edit
			Remove

g. Click the How To Start tab.

The How To Start page opens.

e: CLR1			
What To Debug	How To Start	Advanced	Common
ommand to start app	lication		
CALL PGM(RSELABX)			2
	Prompt		
	-		

By default, the page contains a call for the program selected in the Remote Systems view.

h. Click Prompt.

The Call Program (CALL) window opens.

🔘 Call Program	n (CALL)					x
Program:	> CLF	21	Name			
Library:	> RSI	ELABXX	 Name 			
Parameters:		>		*	Add	
Advanced		T All P	arameters	🗌 Keywords		
CALL PGM(RSELA	ABXX/CLR1)	PARM('XX')				*
			(OK)	Restore defaults	Cancel	1

i. In the **Parameters** field, type 'XX' where 'XX' is your workstation number. Click **Add**.

The parameter value will appear in the lower list.

- j. Click OK.
- k. Click Debug.

If you did not terminate the debug session, you will see this error message.



Complete the steps shown earlier in the section called **Error Handling** to terminate the debug session then start the debug session again by clicking the arrow beside the Debug icon on the workbench toolbar and selecting **CLR1** from the list. Breakpoints and monitors are restored.

You have monitored the variable &count and now you are ready to begin "Exercise 5.5: Stepping into a program."

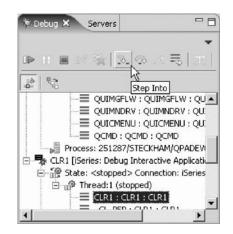
Exercise 5.5: Stepping into a program

Before you begin, you must complete "Exercise 5.4: Monitoring variables" on page 88.

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 the PAYROLLG program.

To step into a program:

1. Click the **Step into** icon on the Debug toolbar.



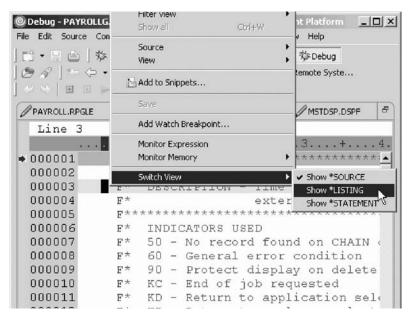
The source of PAYROLLG is displayed. Depending on the option you used to compile the program (*SRCDBG or *LSTDBG for RPG, or *SOURCE, *LIST, or *ALL for ILE RPG), this window displays either the Source or Listing View.

If you specified an incorrect parameter for the CALL program command, you will see this error message.



Complete the same steps as covered in the section called **Error Handling** in "Exercise 5.4: Monitoring variables" on page 88.

 Right-click anywhere in the Source view and click Switch view > Show *LISTING on the pop-up menu.



- **3**. Page down in the source and take a look at the expanded file descriptions. You don't have any /Copy member in your PAYROLLG program but these would also be shown in a Listing view. Switch back to the Source view.
- 4. Right-click anywhere in the Source view.

5. Click **Switch view > Show *SOURCE** on the pop-up menu.

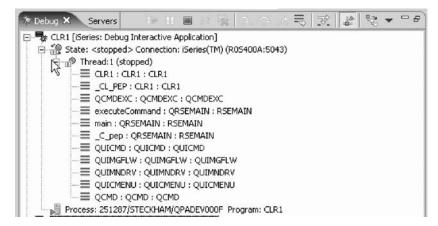
You have stepped into PAYROLLG program, switched the view from source to listing and back to source and now you are ready to begin "Exercise 5.6: Listing call stack entries."

Exercise 5.6: Listing call stack entries

Before you begin, you must complete "Exercise 5.5: Stepping into a program" on page 93.

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, and procedure 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.



The stack entry allows you to work with and switch between different programs and/or ILE modules.

You have viewed the call stack entries of your program and now you are ready to begin "Exercise 5.7: Setting breakpoints in PAYROLLG."

Exercise 5.7: Setting breakpoints in PAYROLLG

Before you begin, you must complete "Exercise 5.6: Listing call stack entries."

Now you add some breakpoints in PAYROLLG.

To add breakpoints:

- 1. Select PAYROLLG in Thread1.
- 2. In the source view (also called the iSeries default editor) scroll to line 57.
- **3**. Double-click the prefix area of line 57.

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

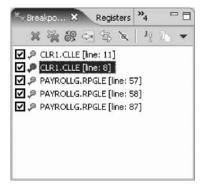
4. Repeat the above step for line 58.

olace ++Opcode (E) +H EVAL
EVAL 🔺
EVAL _
CT format t
DOU

5. Right-click in the prefix area of line 87 and click **Add Breakpoint** on the pop-up menu.

5	RPGLE X	/ PAYROL	CLR1.CLLE
place	olumn 1 Rep	37	Line 8
++Opcode (E)+)	01Factor1+++++	CLO	
ject mainte	If PRJAPL (pro	C*	000085
	branches to lal	C*	000086
wo tests we	If the prior to	C*	000087
	reason code ma:	C*	000088
e operation	next executable	C*	000089
		C*	000090
SELECT		С	₽ 0 00091
			-
<u> </u>			<u> </u>

To view all breakpoints, select the Breakpoints tab from the top left pane.



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

You have added several breakpoints to PAYROLLG and now you are ready to begin "Exercise 5.8: Removing a breakpoint in PAYROLLG" on page 97.

Exercise 5.8: Removing a breakpoint in PAYROLLG

Before you begin, you must complete "Exercise 5.7: Setting breakpoints in PAYROLLG" on page 95.

It is also easy to remove breakpoints from the Source view.

To remove a breakpoint:

- 1. Right-click the prefix area of line 58.
- 2. Click **Remove Breakpoint** on the pop-up menu.

🕲 Debug - P	Show all Ctrl+W	Ve	lopment Platfo	m 💷	
File Edit Sc	Source View	• n	Window Help	2	
	Add to Snippets	1	Remote Syste		
PAYROLL.	Save		X / MSTDS	P.DSPE 6	
	Edit Breakpoint	1			
Line	Remove Breakpoint		ace		
	Disable Breakpoint 🥠 🖓	1	Opcode (E) +Exten	
00005:	Add Watch Breakpoint		BEGSR	1000000000	
00005:	Run To Location		dou	*INK	
00005			EVAL	*IN6	
00005.	Monitor Expression		EVAL	EMES:	
00005	Monitor Memory	•	EVAL	EMPA:	
₽ 00005 [.]	Switch View		EVAL	PRJA:	
»00005-	Switch New		EVAL	RSNA:	
000059	C*				
000060	C* Write the SH	ELEC'	format .	to disi	

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

Now you are ready to run the PAYROLLG program.

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

You have removed a breakpoint from PAYROLLG and now you are ready to begin "Exercise 5.9: Monitoring variables in PAYROLLG."

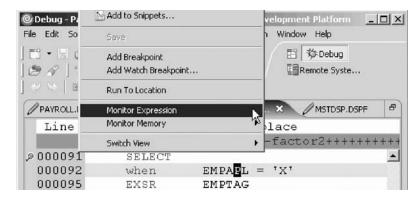
Exercise 5.9: Monitoring variables in PAYROLLG

Before you begin, you must complete "Exercise 5.8: Removing a breakpoint in PAYROLLG" on page 97.

Now lets monitor variables and change them in PAYROLLG.

To monitor variables:

- 1. In the source view, double-click the variable EMPAPL on line 88.
- 2. Right-click the variable.
- 3. Click Monitor Expression on the pop-up menu.



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.

- 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.
- 6. In the Monitors view, double-click the variable RSCDE.

The value changes into an entry field.

7. In the entry field, type in the new value X for the variable.



8. Press Enter.

The variable is successfully changed.

You have monitored several variables in PAYROLLG and now you are ready to begin "Exercise 5.10: Adding a memory monitor."

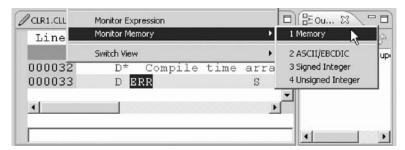
Exercise 5.10: Adding a memory monitor

Before you begin, you must complete "Exercise 5.9: Monitoring variables in PAYROLLG" on page 98.

Adding a memory monitor for a variable allows you to view the memory starting with the address where the variable is located. The memory is displayed in hexadecimal and text format.

To add a memory monitor:

- 1. In the Source view, double-click the variable ERR in line 33.
- 2. Right-click and select **Monitor Memory > Memory** on the pop-up menu.



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

RR : 0xc6ba71b7f7017e80 <	<hex></hex>				
Address	0 - 3	4 - 7	8 - B	C - F	
C6BA71B7F7017E80	404040D		CENERAND	C3C540E2	
C6BA71B7F7017E90	C5D3C5C	🔩 Reset to Base	Address	C540D5D6	
C6BA71B7F7017EA0	E340C5D8	Go to Address.	13	E77F4040	
C6BA71B7F7017EB0	4040D4D	Column Cine		D6D5C540	
C6BA71B7F7017EC0	C1D7D7D	Column Size	P	E2C5D3C5	
C6BA71B7F7017ED0	C3E3C5C4	Reformat	P	D5E3C5D5	
C6BA71B7F7017EE0	C1D5C3C	🚊 Copy To Clipbo	pard	C1D7D7D3	
C6BA71B7F7017EF0	C9C3C1E:		5	C3E3C5C4	
C6BA71B7F7017F00	40C6D6D5	😐 Print	5	C1D5C3C5	
C6BA71B7F7017F10	4040404(Dereference P	ninter	E3C9D6D5	
C6BA71B7F7017F20	40C3D6C-		Lo tocopo	E4C1D340	
0000100000000000000	FOREAODE	01776010	0190090	DEDOA07E	

- 3. Use the scroll bar on the right of the Memory view to scroll down.
- You can see the current content of the memory.
- 4. Right-click in the view area.
- 5. Click **Reset to Base Address** on the pop-up menu to return to the starting address.
- 6. Click the X on the Memory title bar to remove the memory monitor.

Console Tasks iSeries Commands Log T Memory X	÷ 🗙	0a 🔒 🖨	
---	-----	--------	--

You have added a memory monitor for the variable ERR and now you are ready to begin "Exercise 5.11: Setting Watch breakpoints."

Exercise 5.11: Setting Watch breakpoints

Before you begin, you must complete "Exercise 5.10: Adding a memory monitor" on page 99.

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

To set a Watch breakpoint:

1. Go to the **Line number** field at the bottom of the source area. In this field enter 116 to go to that line.

CLR1.CLLE	PAYROLLG.RPGLE ×			8
Line 33	Column	11	Replace	e i
	DName <mark>+</mark> +++	++++++	ETDsFro	m+++To/I
000027	FMSTDSP	CF	Е	
000028	FEMPMST	UF A	E	к
000029	FPRJMST	UF A	E	К
000030	FRSNMST	UF A	Е	К
000031	D*			
000032	D* Compi.	le time	e array	contai
000033	D ERR		S	
•				I
<u> </u>				<u> </u>
114				
TIO				

- 2. Double-click variable *IN60 to highlight it.
- 3. Right-click and click Add Watch Breakpoint on the pop-up menu.

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

By default the **Number of bytes to watch** field is set to zero, which means the variable will be watched in its defined length.

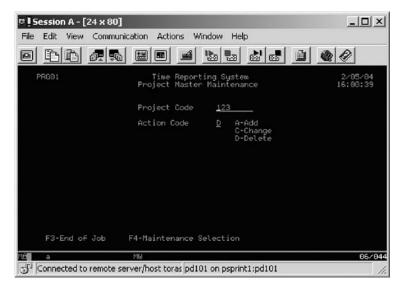
Add a Watch Breakpo				
equired information Sets a breakpoint to stop e	xecution whe	en data change	es at a specific add	tress
Address or expression:	*IN60			1
Number of bytes to watch:	0			<u> </u>
			\Im	
Ĩ	< Back	Next >	Finish	Cancel

4. Click Finish.

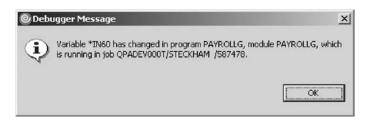
The Watch breakpoint is now set.

5. Click the **Resume** button on the Debug toolbar.

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



- 6. In the 5250 emulation session, 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.



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

You have added a Watch breakpoint for the variable *IN60 and run the program to see the notification that the variable has changed and you are ready to begin "Exercise 5.12: Closing the debug session."

Exercise 5.12: Closing the debug session

Before you begin, you must complete "Exercise 5.11: Setting Watch breakpoints" on page 100.

To close the debugger:

1. Click the **Resume** icon 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.
- 4. A message Program terminated appears:

<u>UK</u>

Click **OK**.

lopment Platform	
in Window Help	
*= \$ - + → -	Close All te
Variables	
×	✓ Show Text J Show Text

- 5. Right-click the **Debug icon** on the top right of the workbench.
- 6. Click Close on the pop-up menu to close the Debug perspective.

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 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. To set a breakpoint you:
 - a. Right-click on the line and click Add Breakpoint on the pop-up menu.
 - b. Double-click in the prefix area.
 - c. Right-click in the Breakpoints view and click Add Breakpoint on the pop-up menu.
 - d. Right-click in the prefix area and click Add Breakpoint on the pop-up menu.
 - e. All of the above
- 4. You can change variables and indicators in the:
 - a. Remote Systems view
 - b. Debug view
 - c. Monitors view
 - d. Source view
 - e. All of the above
- 5. The debugger allows you to:
 - a. Step over a program call
 - b. Step into a called program
 - 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/Enable
 - d. Edit
 - e. 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 text
 - b. Character
 - c. Text only
 - d. Decimal
 - 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.
 - a. Watch
 - b. Line
 - c. Conditional

More 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 your program. Use the product online help to assist you in these tasks.

Module recap

You have completed Chapter 5, "Module 5. Debugging a program," on page 79. You have learned how to:

- · Invoke the debugger from the Launch Configurations window
- · Add a breakpoint
- · Add a conditional breakpoint
- · Edit a breakpoint
- · Monitor a variable through the Monitors view
- Step into your payroll program
- Show a Listing view
- List the call stack entries in the Debug view
- · View all breakpoints
- Remove a breakpoint
- Monitor storage
- Set a Watch breakpoint
- Close the debugger

You have mastered debugging your program and now you learn how to define filters and perform actions. Continue with Chapter 6, "Module 6: Exploring Remote System Explorer," on page 105.

Chapter 6. Module 6: Exploring Remote System Explorer

This module teaches you how to use the Remote System Explorer perspective to work with the iSeries objects that you used in the previous modules. You will learn 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.

In this module, you will:

- · Know the features of Remote System Explorer
- Create a filter to show specific iSeries libraries
- Change the filter to add more iSeries libraries
- Create a filter to show all the source files in a library
- · Access members to edit from your filter
- Create a user action that copies a source file with data to a new source file in the same library
- Specify user action parameters
- · Specify a restriction on a user action
- Try the user action
- Run an OS/400 command from the iSeries Table view

Exercises

- "Exercise 6.1: More about the Remote System Explorer"
- "Exercise 6.2: Creating a library filter" on page 106
- "Exercise 6.3: Creating an object filter" on page 109
- "Exercise 6.4: Creating a user action" on page 113
- "Exercise 6.5: Running commands from the Remote System Explorer" on page 117

Time required

This module will take approximately 15 minutes to complete.

Exercise 6.1: More about the Remote System Explorer

Most of the functions of CODE Project Organizer have been replaced by WebSphere Studio functions 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 over time eventually be a full replacement for PDM.

Remote System Explorer allows you to:

- 1. Simplify your work by giving you quick access to lists of iSeries libraries, objects, members, IFS files, UNIX[®] files, and local files.
- 2. Use the context-sensitive pop-up menus on these lists to perform actions such as start the Remote Systems LPEX Editor, CODE Designer, or Integrated Debugger or other common iSeries actions.

- **3**. 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.
- 4. 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.

You have read the list of Remote System Explorer capabilities and you are ready to begin "Exercise 6.2: Creating a library filter."

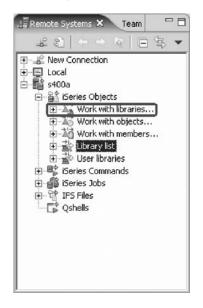
Exercise 6.2: Creating a library filter

Before you begin, you must complete "Exercise 6.1: More about the Remote System Explorer" on page 105.

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

In the previous modules you have worked with the Library list. 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.

Filters allow you to easily organize elements within your system. You use the filter function to list iSeries native file system objects (such as libraries, objects, or members).



To create a library filter:

- 1. In the Remote Systems view expand the connection that connects to your iSeries system if its not already expanded.
- 2. Expand iSeries Objects if its not already expanded.
- Expand Work with Libraries. (You can also right-click iSeries Objects and click New > Library Filter on the pop-up menu).

Expanding Work with libraries corresponds to the WRKLIBPDM command, plus creates the filter in the Remote Systems view.

The Create a new iSeries library filter page opens:

© New	×
Library Filter Create a new iSeries library filter	
Library: RSE*	Browse

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 **RSExxxxx** 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.

You specify the first filter string that selects the libraries starting with RSE.

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

The Name the new filter page opens.

@New			x
ibrary Filter			
Name the new filte	r		
Filters are saved fo appear in the Remo			lter. This name will
Filter name:	RSE and VARP	Glibraries	
Only create filte	r in this conne	ction	
Select a profile to o sharable by the tea			
Owner profile:	eckh7tp		-

- **Note:** You can choose between creating the filter for all connections or for this specific one only.
- 6. In the Filter name field, type All RSE and VARPG libraries.

You give your filters a name because the Remote System Explorer saves them for future use, unlike PDM, which does not save filters.

7. Click Finish.

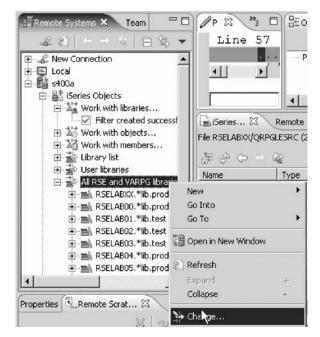
Back in the Remote Systems view under iSeries Objects you will see the new filter expanded, listing all RSE* libraries.

Now you need to add the VARPG libraries.

Changing the library filter

To change the library filter:

1. Right-click the filter All RSE and VARPG libraries and click Change.



The Change Library Filter window opens.

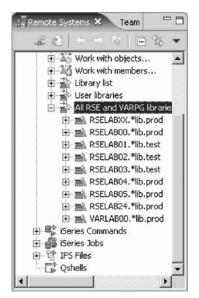
Change Library Filter	× ×
Filter name:	All RSE and VARPG libraries
Parent filter pool:	Msteckh7tp Filter Pool
Filter strings:	New filter string:
New filter string	Library VAR* I . Browse
Rec	Test Create Revert
	Corco Cancel

- 2. Select New filter string from the Filter strings list.
- 3. In the Library field, type VAR*.
- 4. Click **Create**.

The **VAR**^{*} filter string is added to the list.

5. Click OK.

You are now back in the Remote Systems view.



You will see the list expanded to include your filter. Now you can work with the libraries directly and can drill down to the object you want to work with.

You have created a filter to show a specific iSeries library and changed that filter to add more iSeries libraries and you are ready to begin "Exercise 6.3: Creating an object filter."

Exercise 6.3: Creating an object filter

Before you begin, you must complete "Exercise 6.2: Creating a library filter" on page 106.

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

To create an object filter:

1. In the Remote Systems view, expand your connection and then expand **iSeries Objects** if not already expanded.

.] Remote Systems 🗙 Team 🗖 🗖	1
▲ 20 (10 (45 ▼	
🕒 🚠 Work with objects 📔	J
E→Aj Work with members	
由 🗟 Library list	
🖅 🚔 User libraries	
🖨 🚔 All RSE and VARPG librarie	
🕀 🛋 RSELABXX.*lib.prod	I
🕀 🛋 RSELABOO.*lib.prod	I
庄 🛋 RSELAB01.*lib.test	I

 Expand Work with objects. You can also right-click iSeries Objects and click New > Object filter on the pop-up menu.

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

The Create a new iSeries object filter page opens: Now create a filter to show all your source files in your RSELABxx library.

	eries object filter			→] [~]
Library:	RSELABXX		•	Browse
Object:	*		•	Browse
Object type:	*		•	Browse.
Object attribute:	*		•	Browse
	Mor	e Types>>		

- 3. In the Library field, type RSELABxx.
- 4. Click **Browse** beside the **Object type** field. The Select Object Type window opens.

Select Object Type		×
Select an object type:		
*CTLD *CRG *CRQD *DEVD *DOC *DTAARA *DTAQ *EDTD *EXITRG		
*FCT *FILE		
*FLR *FNTTBL *FNTSRC *FORMDF *FTR *GSS *IMGCLG *IPXD *JOBD *JOBD *JOBD *JOBSCD *JRN *IRNRCV		×
	ОК	Cancel

- 5. Select *FILE under the Select an object type list.
- 6. Click OK.

The Create a new iSeries object filter page displays with the object type updated.

Object type: *FILE Image: Browse Object attribute: * Image: Browse	Library:	RSELABxx	<u> </u>	Browse
Object attribute: *	Object:	*	••	Browse
	Object type:	*FILE	V.	Browse
More Types>>	Object attribute:	*	¥.	Browse
Hore Types 22		More Types>>		

- Click Browse beside the Object attribute field. The Select Object Attribute window opens.
- 8. Select **PF-SRC** under the **Select an object attribute** list.
- 9. Click OK.

Select Object Attribute		×
Select an object attribute:		
BEER an object admotte: BSCF38 CMNF38 DDMF DFU DFUEXC DFUNOTEXC DKTF DSPF38 ICFF LF LF LF LF LF LF PF38-DTA PF38-DTA PF38-SRC PRTF PRTF38 SAVF TAPF		
	OK	Cancel

10. Click Next.

The Name the new filter page opens.

New			×
Object Filter			
Name the new f	lter		
	for easy re-use. Specify mote Systems view, and		ilter. This name will
Filter name:	My source files		
Only create I	ilter in this connection		
	o own the new filter. This eam. It will be placed in t		
Owner profile:	nsteckh7tp		•
	< Back N	ext >	Cancel

- 11. In the Filter name field, type My source files.
- 12. Click Finish.

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

le 🔊	(+) =)	6	- \$
1 1 1	terms to	LAB02.*lit	********
- St. (S.E.)		LABO3.*lit	
1 1 1		LAB04.*lit	같은 감사가 가지 말을 했다.
1 1 1		LABO5.*lib	- AL - 20 - 20
1 1 1		LAB24.*lit	19 1 9 19 19 19 10 10 10
1 1 1		LABOO.*li	o.prod
	My sour		
15 A 17 A 17		IPARE.*fi	
		ILLESRC.*	352 102
1 E E		SRC.*file	· · · · · · · · · · · · · · · · · · ·
1 1		1DSRC.*fi	
1 () () () () () () () () () (SSRC.*fil	C2 (20)
		GLESRC.	
		GLESRCP	.*file.pt-
1	ries Com	mands	
	ries Jobs		
1 1	5 Files		
Qs Qs	hells		

Note: If you end up with too many filters, you can create filter pools. They allow you to group filters.

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.

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

Let's assume you want to edit the member PAYROLL in the source file QRPGLESRC using this object filter.

Editing a member from your own object filter

To edit a member:

- 1. Expand QRPGLESRC.
- 2. Right-click member PAYROLL.
- 3. Click Open With > Remote Systems LPEX Editor on the pop-up menu.

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.

You have created a filter to show all the source files in your library and accessed members to edit from your filter and you are ready to begin "Exercise 6.4: Creating a user action."

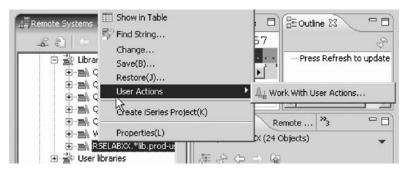
Exercise 6.4: Creating a user action

Before you begin, you must complete "Exercise 6.3: Creating an object filter" on page 109.

In PDM you can create user actions in addition to using the pre-supplied system actions. In Remote System Explorer you can do the same. You define user actions through the Work With User Actions window. User actions can be defined for iSeries libraries, objects, members and jobs as well as folders and files in any remote UNIX, Windows, Linux, Local, or IFS system.

To open the Work with User Actions wizard:

1. Expand your iSeries connection and expand **iSeries Objects** if not already expanded.



- 2. Expand the Library list filter if not already expanded.
- 3. Right-click RSELABxx.
- Click User Actions > Work with User Actions on the pop-up menu. The Work with User Actions window opens.



- 5. In the right pane of the Work with User Actions window, expand **New** in the list, if it is not expanded already.
- 6. Select Object action.

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.

arent profile: msteckh7tp			
New	Action name:	Copy source f	ile
Member action	Comment:	Copy source f	ile with data
	Command:	Normal comma	nd 💌
	CRTDUPOBJ		*
	I		
	Insert variable	Edit	Browse
	Prompt	1	
	Prompt first	□ Refresh only □ Invoke o	
		only 🔲 Invoke d	once
	Single selection	only 🔲 Invoke d	
	Single selection Resource types for Defined Types	only 🔲 Invoke d	once will appear
	Single selection Resource types for Defined Types ALL CMD FILE	only 🥅 Invoke o	will appear Selected Types
	Single selection Resource types for Defined Types ALL CMD FILE FILE FILE_DATA	only T Invoke of which this action	will appear Selected Types
	Constant Single selection	only TInvoke of which this action Add> <remove(g)< td=""><td>will appear Selected Types</td></remove(g)<>	will appear Selected Types
	Single selection Resource types for Defined Types ALL CMD FILE FILE_DATA FILE_DSPF	only TInvoke of which this action Add> <remove(g)< td=""><td>will appear Selected Types</td></remove(g)<>	will appear Selected Types
	Constant Single selection	only TInvoke of which this action Add> <remove(g)< td=""><td>will appear Selected Types</td></remove(g)<>	will appear Selected Types
	Single selection Resource types for Defined Types ALL CMD FILE FILE FILE DATA FILE_DATA FILE_DSPF FILE_MBRS FILE_PRTF	only TInvoke of which this action Add> <remove(g)< td=""><td>will appear Selected Types</td></remove(g)<>	will appear Selected Types

- 7. In the Action field, type Copy source file for the user action name.
- 8. In the **Comment** field, type Copy source files with data.
- 9. In the **Command** field, type CRTDUPOBJ for the command to execute.
- 10. Click **Prompt** to open the command prompter for this command.

© Create Duplicate Ob	oject (CRTDUPOBJ)	×
From object:	8N	▼ Name, generic*
From library:	84	▼ Name
Object type:	&T	▼ <u>A</u> dd
To library:	*FROMLIB	▼ Name
New object:	*OBJ	▼ Name
From ASP device:	*	▼ Name
To ASP device:	*ASPDEV	▼ Name
Duplicate data:	*YES	•
		2
☐ Advanced	🔽 Aļl Parameters	™ ⊑ Keywords
CRTDUPOBJ OBJ(8N) FRO	DMLIB(&L) OBJTYPE(&T) DATA(*YES)	-
		-
		×
	<u>OK</u> <u>R</u> es	store defaults

This is the command you will be running:

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

Specifying user action parameters

To specify user action parameters:

- 1. In the **From Object** field, type &N to indicate to use the name of the selected object in the Remote Systems view.
- 2. In the **From Library** field, type &L to pick up the library name from the selected object.
- **3**. In the **Object Type** field, type &T to pick up the object type from the selected object.
- 4. In the New Object field, type QJUNKSRC.
- **5**. Select the **All parameters** check box to see the additional Duplicate data parameter.

Now the Duplicate data parameter is also shown on the prompt window.

- 6. Select ***YES** from the **Duplicate data** list.
- 7. Click OK.

You return to the Work with User Actions window.

8. Select the **Refresh after** check box, so that the Remote Systems view gets refreshed after the action has been run.

E & New	Action name:	Copy source f	le
Object action	Comment:	Copy source f	ile with data
	Command:	Normal comma	nd j
		(&N) FROMLIB(&L)	DBJTYPE(&T) NEW
	Insert variable.	Edit	Browse
	Prompt first Single selection	Refresh n only I Invoke o	
	Single selectio	and the second se	
	Single selectio	n only 📃 Invoke o	nce will appear
	Single selection	n only I Invoke o or which this action Add> <remove(g) Edit(H)</remove(g) 	nce will appear Selected Types

Note: Clicking the **Insert variable** push button displays a list of valid replacement variables with the explanation of what they do.

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.

Specifying a restriction on a user action

To specify this restriction:

- 1. Under the **Defined Types** list box, click **FILE_SRC**.
- 2. Click Add beside the Defined Types list box.

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

3. Click Create then Close.

Now, only when you right-click on a source file, will this user action appear on the pop-up menu. For any other object type it will not appear. Back in the workbench and the Remote System view, give it a try.

Note: Remember to close the payroll source member if you opened it earlier.

Trying the user action

To try your user action:

1. Locate your filter My source files.

RSELABU2.*ID.test RSELAB03.*Ib.test RSELAB04.*Ib.proc RSELAB05.*Ib.proc RSELAB24.*Ib.proc RSELAB24.*Ib.proc RSELAB24.*Ib.proc	Change Save Restore	Dutline 8
E COMPARE.*file.pf-s	User Actions	Copy source file
	Add To iSeries Project Make Available Offline	A: Work With User Action
D- QDD55RC.*file.pf-s	Properties(L)	SRC SRC

- 2. Expand the filter My source files, if it is not already expanded.
- **3**. Right-click the QRPGLESRC file.
- 4. Click **User Actions > Copy source file** on the pop-up menu.

The file gets duplicated and the list gets refreshed. Your new source file will show in the list. You can check the messages of the CL commands you are running in the RSE communications server job by looking at the iSeries Commands log in the right hand side bottom pane of the workbench.

🔓 iSeries Commands Log 🗙 Remote Search	∅ ▼ □ □
s400a	
NEWOBJ(QJUNKSRC) DATA(*YES) Object QJUNKSRC in RSELABXX type *FILE created	. 🔺
1 objects duplicated. Cause : All the objects with supported object be Create Duplicate Object (CRTDUPOBJ) comman See low level messages to determine if any objects	d were duplicated.
Command Normal	

- 5. To delete the source file QJUNKSRC that you just created, right-click QJUNKSRC.
- 6. Click **Delete** on the pop-up menu.

The Delete Confirmation dialog opens.

7. Click **Delete**.

You have created a user action that copies a source file with data to a new source file, specified user action parameters, specified restrictions on the user action and tried the user action and you are ready to begin "Exercise 6.2: Creating a library filter" on page 106.

Exercise 6.5: Running commands from the Remote System Explorer

Before you begin, you must complete "Exercise 6.4: Creating a user action" on page 113.

The command entry is part of the Remote Systems view.

To run a command:

- 1. Check if you have an **iSeries Table View** tab in the bottom right pane where your commands log appeared in the previous exercise.
- 2. If you have it, click on it.
- 3. If you don't have it:
 - a. In the Remote Systems view, right-click My source files filter.
 - b. Click **Show in table** on the pop-up menu.

As you have seen already you can now run commands on the iSeries server that the table is connected to. You can enter commands in the Commands field beneath the iSeries Table view, and view messages in the Messages field. You can enter a command and click either Prompt to specify parameters and then Run, or just click Run. When you run a command, the Messages field is populated with the messages from the command. When you select a message, the Details button is enabled. When you click this button, the message and its help is displayed.

4. Type an iSeries command, for example ?ADDLIBLE.

Vame	Туре	Attribute	Text
COMPARE	*FILE	PF-SRC	
QCBLLESRC	*FILE	PF-SRC	
QCLSRC	*FILE	PF-SRC	
QCMDSRC	*FILE	PF-SRC	
QDDSSRC	*FILE	PF-SRC	
QRPGLESRC	*FILE	PF-SRC	
QRPGLESRCP	*FILE	PF-SRC	
<u> </u>			
Command Normal	ADDLIBLE		▼ Pro

The question mark is there to display a prompt screen.

Instead of specifying a question mark you could use the **Prompt** push button.

5. Click Run.

The Command Prompt window for the ADDLIBLE command opens.

6. In the Library field, type RSELAB00 and click OK.

O Add Library List E	RSELABOO	Name
Library list position: List position:	*FIRST	•
Reference library:		Name
ADDLIBLE LIB(RSELABO	00)	
	,0)	

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

Note: You may need to add a different library if the library RSELAB00 does not exist on your iSeries system.

The Messages field will confirm the successful completion of this command. To get to the command history, similar to F9 on the 5250 screen, select the appropriate value from the **Command** list (down arrow beside the **Messages** field.

Name	Туре	Attribute	Text
COMPARE	*FILE	PF-SRC	
QCBLLESRC	*FILE	PF-SRC	
QCLSRC	*FILE	PF-SRC	
QCMDSRC	*FILE	PF-SRC	
QDDSSRC	*FILE	PF-SRC	
QRPGLESRC	*FILE	PF-SRC	
QRPGLESRCP	*FILE	PF-SRC	
<u> </u>			
Command Normal	-		Prompt F

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.

You have run an OS/400 command from the iSeries Table view.

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 1. The Remote System Explorer is replacing what ADTS tool:
 - a. Page Designer
 - b. CODE Designer
 - c. Screen Design Aid
 - d. Program Development Manager
- 2. ______ filters list a set of libraries from your iSeries system in the Remote Systems view.
 - a. Object
 - b. Member
 - c. Library
 - d. Job
- **3**. Filters allow you to easily organize elements within your system. You use the filter function to list iSeries native file system objects (such as libraries, objects, or members). (T, F)
- 4. Expanding Work with Libraries corresponds to the _____ command.
 - a. WRKOBJPDM
 - b. WRKLIBPDM
 - c. WRKMEMPDM
 - d. All of the above

- 5. You can create filters for all connections or for a specific connection. (T, F)
- 6. You give your filters a specific name because the Remote System Explorer saves them for future use, unlike PDM, which does not save filters. (T, F)
- 7. ______ filters list a set of objects from your iSeries host in the Remote Systems view.
 - a. Job
 - b. IFS
 - c. Object
 - d. Member
 - e. Library
- 8. If you end up with too many filters, you can create filter pools. (T F)
- 9. Filters can be specified for non iSeries servers and your local system. (T, F)
- 10. User actions can be defined for:
 - a. iSeries libraries
 - b. iSeries objects
 - c. iSeries members
 - d. iSeries jobs
 - e. files and folders in any remote UNIX, Windows, Linux, Local or IFS system
 - f. All of the above
- 11. You can specify a restriction on a user action. (T, F)
- 12. You can run OS/400 commands from the:
 - a. iSeries Table view
 - b. iSeries Commands log
 - c. Tasks view
 - d. iSeries Error List
 - e. a and b
 - f. All of the above

More 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.

Recap

You have completed Chapter 6, "Module 6: Exploring Remote System Explorer," on page 105. You have learned how to:

- Know the features of Remote System Explorer
- Create a filter to show specific iSeries libraries
- · Change the filter to add more iSeries libraries
- · Create a filter to show all the source files in a library
- · Access members to edit from your filter
- Create a user action that copies a source file with data to a new source file in the same library
- Specify user action parameters

- Specify a restriction on a user action
- Try the user action
- Run an OS/400 command from the iSeries Table view

Now that you know how to create filters and actions to manage your iSeries objects you can learn how to visually design screens. Continue with Chapter 7, "Module 7. Designing screens," on page 123.

Chapter 7. Module 7. Designing screens

This module teaches you about the various aspects of the CODE Designer while modifying a display file to add a screen. You will step through each part of the CODE Designer tool interface and update some DDS as well. In the workbench, in the Remote System Explorer perspective you will use the connection that you used in the module before.

In this module, you will:

- · Open a DDS member for edit with CODE Designer
- · Show file-level keywords and record keywords
- · View the details of records, record-level keywords and field-level keywords
- View the design of the payroll application main menu
- · Create a group from an existing record format
- Create a new group and add a subfile record and a subfile control record
- Add columns to the subfile record
- Add fields to the subfile control record
- Copy existing fields
- · Set indicators to handle field errors
- View and update record and field properties
- View keywords and the properties of a keyword
- · Insert a keyword
- View help for a keyword
- Check there are no semantic errors in the DDS source
- View help for an error
- · Launch the editor in read mode from the error list
- · Launch the editor in write mode to fix the error
- Find a keyword in the source
- Save source changes
- Compile your source changes
- Close the Designer

Exercises

The exercises within this module must be completed in order:

- "Exercise 7.1: Opening a DDS member in the Remote Systems view" on page 124
- "Exercise 7.2: Viewing the DDS tree" on page 125
- "Exercise 7.3: Selecting the DDS object" on page 126
- "Exercise 7.4: Designing the DDS screen" on page 127
- "Exercise 7.5: Creating groups from existing records" on page 128
- "Exercise 7.6: Creating new screens" on page 130
- "Exercise 7.7: Adding fields to the subfile record" on page 132
- "Exercise 7.8: Switching between multiple records" on page 135
- "Exercise 7.9: Adding field error handling" on page 137

- "Exercise 7.10: Accessing field properties" on page 139
- "Exercise 7.11: Adding new keywords" on page 141
- "Exercise 7.12: Verifying the source changes" on page 143
- "Exercise 7.13: Switching between designing and editing the screen" on page 145
- "Exercise 7.14: Compiling your source changes and closing the Designer" on page 146

Time required

This module will take approximately 30 minutes to complete.

Exercise 7.1: Opening a DDS member in the Remote Systems view

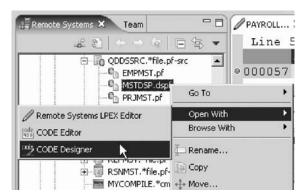
Using an editor to create and maintain DDS source for your display, printer and physical files can be a frustrating and difficult task. What would be great is a graphical design tool that let's you design your screens and reports visually and then generate the DDS source for you. Well, that's exactly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

CODE Designer helps 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.

Note: Make sure MSTDSP is closed from the previous LPEX Editor exercises.

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

- 1. Expand the Library list filter if it is not already expanded.
- 2. Expand the QDDSSRC file in library RSELABxx.



- **3**. Right-click the MSTDSP member.
- 4. Click **Open with > CODE Designer** on the pop-up menu.

The member MSTDSP will be downloaded to the workstation and loaded into CODE Designer. CODE Designer is a separate tool not integrated into the workbench.

You have opened the DDS source member MSTDSP in the Remote Systems view using CODE Designer and you are ready to begin "Exercise 7.2: Viewing the DDS tree" on page 125.

Exercise 7.2: Viewing the DDS tree

Before you begin, you must complete "Exercise 7.1: Opening a DDS member in the Remote Systems view" on page 124.

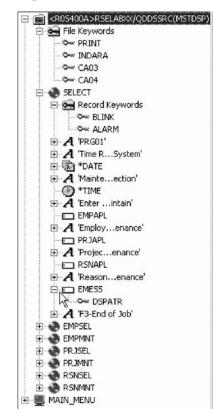
What you are looking at now is basically an Explorer view of the DDS. The DDS tree view on the left-hand side of the Designer displays the DDS source in its file, record, field, and keyword hierarchy. It is a familiar and intuitive way to see the overall structure of the DDS source and to navigate through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

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 show file-level keywords and the record SELECT in the DDS tree:

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



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

You have seen the file-level keywords and the record SELECT in the DDS tree and you are ready to begin "Exercise 7.3: Selecting the DDS object."

Exercise 7.3: Selecting the DDS object

Before you begin, you must complete "Exercise 7.2: Viewing the DDS tree" on page 125.

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, printer or physical 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 by clicking View > List from the CODE Designer menu.

In the Details page columns display information about the selected DDS object. You can use this page to display for example, details of all the fields in the record SELECT or keywords and conditions of a field or record.

The Listing page is a listing of the source statements generated by the Program Verifier.

In the bottom right-hand side of the Designer is the Utility notebook. This notebook contains several pages: Selected DDS, Web Settings, Comments and Error List. The Selected DDS page in the notebook shows the actual DDS source for the currently selected item.

Note: The Web Settings page allows you to specify attributes that are used by the WebFacing tool.

To work with the DDS record SELECT:

1. In the DDS tree click the SELECT record.

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.

Record SELECT fields						
Field	Positi	Len	Туре	Shift	Usa	Sample 🔺
A 'PRG01'	2,5	5	Text const			PRG01
A 'Time RSystem'	2, 30	21	Text const			Time Reporting Syste
🚾 "DATE	2,70	6	Date const			MM/DD/YY
A 'Mainteection'	3, 30	21	Text const			Maintenance Selectic
*TIME	3, 70	6	Time const			HH:MM:SS
A 'Enterintain'	6,14	54	Text const			Enter an X beside the
🗖 EMPAPL	9,25	1	Alphanumeric	A - Alphanum	Both	В
A 'Employenance'	9,28	27	Text const			Employee Master Mai
🗖 PRJAPL	10, 25	1	Alphanumeric	A - Alphanum	Both	В
A 'Projecenance'	10, 28	26	Text const			Project Master Mainte
	11, 25	1	Alphanumeric	A - Alphanum	Both	B

 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.

	Record SELECT keywords
Keyword	Conditioni
🗫 BLINK	
S ALARM	60
L,	<u></u>
n	•
Selected DI)S Create keywords Web Settings Comments Error list
Selected DC	DS Create keywords Web Settings Comments Error list
e	• • • • • •
<u>₽</u> À*	89/03/02 12:07:25 DOUG REL-
₽ Å* Å*	89/03/02 12:07:25 DOUG REL- THE SELECT FORMAT ALLOWS SELECTION OF
A* A* A*	89/03/02 12:07:25 DOUG REL- THE SELECT FORMAT ALLOWS SELECTION OF MASTER FILE THE OPERATOR WANTS TO MAIN
A* A* A*	89/03/02 12:07:25 DOUG REL- THE SELECT FORMAT ALLOWS SELECTION OF MASTER FILE THE OPERATOR WANTS TO MAIL R SELECT
A* A* A*	89/03/02 12:07:25 DOUG REL- THE SELECT FORMAT ALLOWS SELECTION OF MASTER FILE THE OPERATOR WANTS TO MAIN
A* A* A*	89/03/02 12:07:25 DOUG REL- THE SELECT FORMAT ALLOWS SELECTION OF MASTER FILE THE OPERATOR WANTS TO MAIN R SELECT BLJ 60 AL2 2 5'FF
₽ Å* Å*	89/03/02 12:07:25 DOUG REL- THE SELECT FORMAT ALLOWS SELECTION OF MASTER FILE THE OPERATOR WANTS TO MAIN R SELECT 60 BLJ

3. In the DDS tree click the EMESS field.

The Details page shows its field-level keywords. The Selected DDS page now shows the DDS for the EMESS field.

Select	ted DDS Create	keywords Web Settii	ngs Comments Error list	
	À À 60	EMESS	50A O 21 16 DSPATR(HI)	*
र				<u>*</u>

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 independent of the DDS required to make things appear the way they do. You need the Design page.

You have seen the details of the record SELECT, the record-level keywords and the field-level keywords and you are ready to begin "Exercise 7.4: Designing the DDS screen."

Exercise 7.4: Designing the DDS screen

Before you begin, you must complete "Exercise 7.3: Selecting the DDS object" on page 126.

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. You can 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.

1. Click the MAIN-MENU tab in the workbook.



In order to understand where MAIN_MENU came from, you need to understand the concept of a group. A group is simply a collection of one or more DDS records that represent how a screen or report would be assembled at runtime. It allows you at development time to work with screens or reports as they would appear when they get assembled by your programs at runtime. To work with groups in CODE Designer you need to tell CODE Designer which record formats make up a screen. In this case this has been done for the screen you see in the Design page. A group MAIN_MENU has been created for you and CODE Designer has saved this information in the DDS source in comment lines. Any groups that you create are persisted as comment lines in the DDS so you can re-use these groups in subsequent CODE Designer sessions.

The groups you create will appear in the tree view as well as in the workbook as a Design page tab for each group defined, to allow quick access to each group of records.

You have seen the concept of a group, specifically the MAIN_MENU group and you are ready to begin "Exercise 7.5: Creating groups from existing records."

Exercise 7.5: Creating groups from existing records

Before you begin, you must complete "Exercise 7.4: Designing the DDS screen" on page 127.

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. The screen is made up of the record format EMPSEL.

To create a new group:

1. Scroll to the bottom of the DDS tree and expand the MAIN_MENU group. The SELECT record appears as the only record in this group.



- 2. Right-click the MAIN-MENU group.
- 3. Click **Insert group** on the pop-up menu.

A Group Properties notebook opens and a blank Design page for the group SCREEN1 also opens.

Details MAIN_MENIC SCREENT Source Listing
□
Group properties ? × 5 Bycc: Styles 6 Group 7 Group
5 Bytes 6
Available Selected
Available Selected
PRJSEL (W)
Selected DDS Web Settings Comment: RSNSEL RSNMNT
À.
A A
à l

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

4. In the Group Properties notebook, select the EMPSEL record from the **Available**

list and click the 🚬 button.

For simplicity this is the only record you will add for now. The Design page now shows you what the record EMPSEL looks like.



5. In the group field, type EMPL_SELCT over SCREEN1 to rename the group.

Group	EMPL	SELCT		
-	vailable		Selected	
SELECT		EMF	SEL	
PRJSEL		(and the second s		
RSNSEL				
RSNMNT		<<		

6. Close the Group Properties notebook. Click the X in the top right corner of the Group Properties notebook.

You have finished creating a group. You could now work in the Design page with the record formats contained in this group. Instead you'll create a new record format.

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.

You have inserted a new group using the existing record EMPSEL and you are ready to begin "Exercise 7.6: Creating new screens."

Exercise 7.6: Creating new screens

Before you begin, you must complete "Exercise 7.5: Creating groups from existing records" on page 128.

To create a new record screen on the Design page you need to create a group that will create an empty page you can work with.

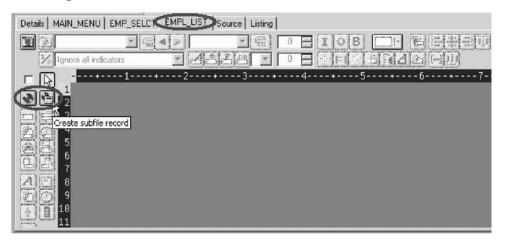
To create a new group:

- 1. Right-click the new EMPL_SELCT group in the DDS tree.
- 2. Click Insert group on the pop-up menu.

A Group Properties notebook appears and a blank Design page for the group SCREEN1 also appears.

Details MAIN_MENU EMP_SELCT	CREENT Source Listing	
Ignore all indicators		i esten I esten
	Group properties Basics Styles	+67 ?×
	Group SCREEN1 Available Selected SELECT EMPSEL EMPMNT PRJSEL >> PRJMNT RSNSEL <	
Selected DDS Web Settings Comm		

- 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 buttons available are **Create standard record** and **Create subfile record**. 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.

Details MAIN_MENU EMP_SELCT EMPL_LIST Source Listing
Ølgnore al indicators ▼ 23529 ▼ 0 = 110 E 12 E 12 E
I I <td< td=""></td<>

You have created a new record screen and you are ready to begin "Exercise 7.7: Adding fields to the subfile record."

Exercise 7.7: Adding fields to the subfile record

Before you begin, you must complete "Exercise 7.6: Creating new screens" on page 130.

Now you add some columns to the subfile using the Design page. The subfile should be positioned on row eight. You use the cursor to specify the location of the part you want to put on the screen, in this case your subfile.

To add fields on the subfile:

- 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.

	<mark>+</mark> 1+2
	8,5
	BBBBBBBBBB
	<u>BBBBBBBBBB</u>
<u></u> [] 10	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
日出11	BBBBBBBBBBB
A 12	BBBBBBBBBBB
S 3 13	BBBBBBBBBBB
H	

3. Move the cursor 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 button.



The toolbar at the top of the Design page is a very convenient place to monitor and manipulate the currently selected parts.

4. Rename the record from RECORD1 to EMPLSTSFL and the field from FIELD1 to OPCODE by typing over the text in each list.

		No	and the second s	IOB -
Ignore all indicators	•	A#6B		× II # II A

- 5. Click the Color palette button and select pink to change the color of the field.
- 6. Click the **I** button to change the usage of the field to input.



Now you will create an additional column in the subfile.

Creating an additional output column

Now you want to create an output column.

To create an additional output column:

1. Position the cursor at row 8, column 9.

Note: The bottom right of the CODE Designer window shows the current



If you can't see the field with the cursor position on your screen, click the **Maximize** button in the top right corner of the screen. You can use the cursor keys or the mouse to move the cursor.

2. 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 +0(30).

3. Click the **Convert string to field** button on the toolbar or press **F11** to convert the SDA syntax into a character output field of length 30.

III	
III	000000000000000000000000000000000000000
III	00000000.000000000000000000000000000000
III	000000000000000000000000000000000000000
III	000000000000000000000000000000000000000
III	000000000000000000000000000000000000000

EMPLSTSFL	5 <u>30</u> 4	1	'+O(30)' •	6 <u>442</u>	6		IOB	
Ignore all indicators		•	A BBB	Y	0	100	※ 1=1 米 🛛	6 MA L

4. Rename the new field to ENAME using the toolbar. This will show the name of the employee.

Details MAIN_MENU EMP_SELCT EMPL_LIST Source Listing	
lignore all indicators ▼ ▲書子碑A▼ 0 🖶 ※目線6 🕅	<u>4</u> 20 E III
□ 🕞	6+7

- 5. Position the cursor to 8, 41.
- Now you will add a field for the employee's salary. Type \$666,666.66

and then press the back arrow.

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?

7. 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!

EMPL	STSFL all indica			<mark>ю.</mark> К.С.Регу	≝ ▼ 2		B 💷 ·
	+	1+-	2	+3	+	4+	- 5 +
2 2							
4 1 8	III	00000000	00000000	00000000	000000	\$666,666	.66
5 63 9	III	00000000				\$666,666	
	III	00000000	00000000	00000000	000000	\$666,666	
11	III	00000000	00000000	00000000	000000	\$666,666	.66
12	III	00000000	00000000	00000000	000000	\$666,666	i.66
6 📑 13	III	00000000	00000000	00000000	000000	\$666,666	i.66
14							
elected DDS	Web Set	tings Commer	nts Error list				
· ·							

- 8. Rename the field to SALARY and change its color to yellow, using the toolbar.
- 9. The subfile seems compacted to the left. It would be better to space it out

evenly. Just select a field and click the space horizontally **U** button on the far right side of the toolbar. The other alignment buttons will align fields, left, right, center and top.

	PLSTSFL	SALARY -	≝ 8 ∷ IOB (Y▼ 2 ∷ 淡티翁:	
III		000000000000000000000000000000000000000	\$666,666.68	
			\$666,666.66 \$666,566.66 \$666,666.66 \$666,666.66 \$666,666.66	

Just below the palette on the left 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.

- 10. In the Subfile size field, type 300.
- 11. In the Subfile page size field, type 9.
- 12. Click in the Design page.

EMPLSTS			
1/2 Ignore all in	ndicators	- Cobe - De <u>Pers</u> is	
· 🔊	+1+	2345	+88
and a second second			
3			
. B. 9			
L 🖺 8	III	000000000000000000000000000000000000000	\$666,666.66
	III	000000000000000000000000000000000000000	\$666,666.66
B ¹⁰	III	000000000000000000000000000000000000000	\$666,666.66
11	III	000000000000000000000000000000000000000	\$666,666.66
192	III	000000000000000000000000000000000000000	\$666,666.66
00 13	III	000000000000000000000000000000000000000	\$666,666.66
3 1 14	III	000000000000000000000000000000000000000	\$666,666.66
10	III	000000000000000000000000000000000000000	\$666,666.66
0 🗄 16	III	000000000000000000000000000000000000000	\$666,666.66
17			

The Design page is updated accordingly.

You have added some columns to the new subfile record screen and now you are ready to begin "Exercise 7.8: Switching between multiple records."

Exercise 7.8: Switching between multiple records

Before you begin, you must complete "Exercise 7.7: Adding fields to the subfile record" on page 132.

Now let's fix up the subfile control record. The group you created contains 2 records. You can verify this by looking at the record list in the toolbar:



 Change the current record by selecting RECORD1CTL from the record list or click next record 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.

000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66
000000000000000000000000000000000000000	\$666,666.66

- 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:

Details MAIN MENU EMP_SELCT	MPL_LIST Source Listing	
		· · · · · · · · · · · · · · · · · · ·
Ignore all indicators	• 2850 • • = ¥5857	de eu
	····2 <mark>-</mark> -···+····3-···+-···4-···++5-···+·	68
Position	to:	
A 🖤 :		\$666,666.66
		\$656,666.66
* 3 10		\$666,666.66
11 11		\$666,666.66
		\$666,666.66
300 🚍 13	000000000000000000000000000000000000000	\$666,666.66
9 14		\$656,655.66
		\$666,666.66
0 🚍 16		\$666,666.66
17		

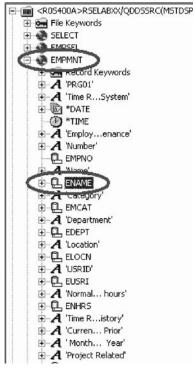
Now you need an employee name field.

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 Create source reference field button in the palette, or you could reference the original database field using either the Create database reference field button or the 🕒 Create database reference field(s) by selection 🗳 button. But there is an even simpler way. Use copy and paste!

Copying the employee name field

To copy the employee name field:

- 1. In the DDS tree expand the EMPMNT record.
- 2. Click the ENAME field and press Ctrl+C.



(The pop-up menu or Edit menu shows the Copy menu item as well).

- 3. Position the cursor to 4, 23 and press Ctrl+V. Now that was easy!
- 4. Click the field and change the name from ENAME to POS_TO.

Details MAIN_MENU	EMP_SELCT	EMPL_LIST Source Listing	
EMPLSTCTL	▼ 5 ^{x+2} /4		*E#ST
1/6 Ignore all ind	icators		
	1+-	2 <mark>-</mark> -+3+4+5+-	688
			<u> </u>
	Position	to: BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	
4 9 9 1 4 7			
A 1 8			\$666,666.66
E O 9			\$555,655.65
10			\$666,666.66
			\$666,666.66
11			\$666,666.66
300 🚍 13			\$555,555.65
14			\$666,666.66
9 1 15			\$666,666.66
0 16			\$666,666.66
17			<u> </u>

You have fixed up the subfile records to add an employee name field with a position to entry opposite this field and you are ready to begin "Exercise 7.9: Adding field error handling."

Exercise 7.9: Adding field error handling

Before you begin, you must complete "Exercise 7.8: Switching between multiple records" on page 135.

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 set on indicator 60.

In this case the field should turn red, reverse image and position the cursor to it. Now wouldn't it be better if you had something easier to remember than some arbitrary number from 1 to 99.

To set indicators:

- 1. Click the Change named indicator sets 💹 button on the Design page toolbar (or press F7.)
 - The Named indicator sets window opens.
- 2. In the Settings name field, type: Not Found.
- 3. Click Create.

Named indicat	or sets				×
	1	Indicators			
01 11 02 12 03 13 04 14 05 15 06 16 07 17 08 18 09 18	20 30 21 31 22 32 23 33 24 34 25 35 26 36 27 37 28 38 29 38	40 50 41 51 42 52 43 53 44 54 45 55 46 56 47 57 48 58 49 58	61 7 62 7 63 7 64 7 65 7 66 7 66 7 66 7 68 7	70 80 71 81 72 82 73 83 74 84 75 85 76 86 77 87 78 88 79 89	90 91 92 93 94 95 96 97 98 98 98
	Not Found			<u>C</u> rea <u>B</u> ena Dela	me
1	<u>(</u>	<u>рк А</u> рг	oly Ca		elp

4. Select 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 **Select named indicator set** list on the bottom left.

EMPLSTCTL	▼ (<u>***</u>) ≪ ≫	POS_TO 💌 🎬
Not Found	•	

- 5. Now select the **POS_TO** field.
- 6. 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:

POS_TO 💌	4 <u>440</u>			B	
▪ А #Б₽₿	Α 💌	0 -	ii 🔆		12 FU

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

A A A	60	POS_TO	R	В	4	23 COLOR(RED) REFFLD(RCEMP/ENAME *LIBL/E
A A	90 60					DSPATR(PR) DSPATR(RI)
A	60					DSPATR(PC)

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

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

EMPLSTC			
All indicate	ors are off 1+?		
	1	J J J	0
	Position to:		
	POSICION CO.		
		000000000000000000000000000000000000000	**** ***
		000000000000000000000000000000000000000	\$666,686.68 \$666,666.66
요 쇼 끄			\$666,666.66

Wow! This really works!

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

You have added error handling to the position to employee name field and you are ready to begin "Exercise 7.10: Accessing field properties."

Exercise 7.10: Accessing field properties

Before you begin, you must complete "Exercise 7.9: Adding field error handling" on page 137.

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 the 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 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-click on anything in the DDS tree or the Details page or Design page.

To open the Properties notebook:

1. In the DDS tree, click the record SELECT and press F4 to see the Record properties.

Record	SELECT		
Туре	Standard		
Modified	06/29/92		
Number of fields	14	🔲 In a <u>w</u> indow	
Description			_

As you select different items, the Properties notebook will continuously update itself to show you the properties of the selected item.

2. 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.

🖃 💼 <r1s400a>RSELABXX/C</r1s400a>
E E File Keywords
- Ow PRINT
- INDARA
- CA03
CA04
E SELECT
E 📻 Record Keyword:
BLINK
• A 'Time RSystem

- 3. Change the year from 2 to 4 digits. Select the Length of year check box.
- 4. Select 4 digits from the list.

Maintenance Select		HH: MM: SS
Basics Attributes Co	olor Editing Indicators	1
Sample	Source of date	~
MM/DD/1111	Length of year	4 digits 🗖
Position Size Row Colu	100D	
	0	
27×132 2	2=	

Notice how the sample is updated on the Properties notebook.

- 5. To test the Design page, click the **MAIN_MENU** tab in the workbook and look at the upper right corner of the screen. The year now has 4 digits.
- 6. Click the EMPAPL field in the SELECT record. On the Field properties notebook click the **Basics** tab.

On this page you can change the field's name, usage, length, type, and screen position. The other pages give you quick access to other properties of this field.

Field properties Basics Attributes Color	Indicators Styles	<u>? ×</u>
Field EMPAPL Usage IOBHPM	Type CHEREPS Shift A - Alphanumeric	-
Length Total Dec Continued (CNTFLD)	Position Size Row Column 24×80 9 25 25 27×132 2 2 2	
Description		

You have seen the record properties for the record SELECT and changed the length of year to 4 digits and you are ready to begin "Exercise 7.11: Adding new keywords."

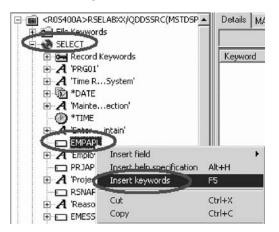
Exercise 7.11: Adding new keywords

Before you begin, you must complete "Exercise 7.10: Accessing field properties" on page 139.

CODE Designer helps you manage the visual aspects of your displays and reports. But you also need to access the full power of DDS. You need to access keywords.

To add keywords:

- 1. Click the EMPAPL field in the DDS tree.
- 2. Press F5 or right-click and click Insert keywords on the pop-up menu.



You see the Details page for the EMPAPL field and the **Create keywords** tab is added to the Utility 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 9	ubset	All					-
Swalias		CHECK(M	(11)	See CHR	ID	See FLDCSRPRG	∽wRDWRAP
SW BLANKS		SW CHECK(M	(10F)	COL	0R>>>	Second Contract Second Sec	Service Contraction of the service o
🗫 CHANGE		Service CHECK	111F)	ᄍ сом	1P	See INDTXT>>>	
🗠 CHECK(A	B)	CHECK(R	RB)	Sw DFT		SGID>>>	
CHECK[E	R)	CHECK[R	RL)	OFT'	VAL	Service Noccosid	
CHECK[F	E)	CHECKIR	RZ)	DSP	ATR>>>	Second Se	
CHECK(L	C)	CHECK(V	(N)	Sw DUP		Second Se	
CHECK(M	IE)	S-CHECKIV	NE)	Swent	FLDATR	Service PUTRETAIN	
CHECK(M	IF)	See CHGINPE	DFT	See ERR	IMSG>>>	See BANGE	
CHECK[N	(10)	CHKMSG	ID	See ERR	MSGID>>>	Servalues	

With the EMPAPL Properties notebook at the Basics page, click the numeric field button to change the field to numeric type.

Basics Attributes Color Field		
EMPAPL Usage LOBHPM	Shift A - Alphanumeric	
Total Dec	Size Row Column 24×80 9 25	
Continued (CNTFLD) Width	27×132 2 2	
Description		

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 again.

5. Click the ALIAS keyword and press F1.

The DDS Reference help for the ALIAS keyword appears.

Tip: CODE Designer has lots of on-line help. Press F1 anywhere you want to see help for an item, icon or notebook. You will see help relevant to what you are currently trying to do. From the Help menu you have quick access to the DDS Language Reference as well as several other useful sources of information.

Contents 🏟	() () () () () () () () () () () () () (
 Remote System Explorer Classic iSeries tools inforr Remote System Explorer 	DDS Reference
✓ Remote System explorer	ALIAS (Alternative Name) Keyword for Display Files
	Use this field-level keyword to specify an alternative name for a field. When the program is compiled, the alternative name is brought into the program instead of the DDS field name. The high-level language compiler in use determines if the ALIAS name is used. Refer to the appropriate high-level language reference manual for information about ALIAS support for that language.
	The format of the keyword is: ALIAS (alternative-name)
	Refer to DDS Syntax Rules for ALIAS naming conventions.

- 6. Minimize the Help window.
- 7. In the Create keywords page, 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 the INDTXT keyword again.

The keyword is created with the same default values creating a conflict.

Keyword	Parame	
See EDTCDE	3	
See INDTXT	01 '?'	
See INDTXT	01 '?'	

9. Close the Keyword Properties dialog.

You have seen the help on the ALIAS keyword, added INDTXT keywords and you are ready to begin "Exercise 7.12: Verifying the source changes."

Exercise 7.12: Verifying the source changes

Before you begin, you must complete "Exercise 7.11: Adding new keywords" on page 141.

You have just added a new record and some new fields to your 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 exercise by creating two INDTXT keywords describing the same indicator.

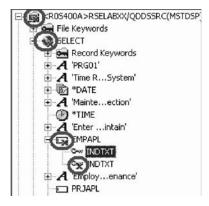
To verify your source:

1. Click **Tools** > **Verify file** (or click the verify **▶** button on the main toolbar) on the CODE Designer menu.



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.

2. 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.

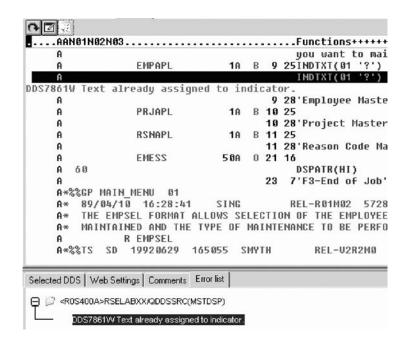
The EMPAPL field is highlighted in red.

4. 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.

- 5. 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.



You have verified your DDS source and have identified an error in the source and you are ready to begin "Exercise 7.13: Switching between designing and editing the screen."

Exercise 7.13: Switching between designing and editing the screen

Before you begin, you must complete "Exercise 7.12: Verifying the source changes" on page 143.

To switch between the Design mode and the Edit mode:

1. Click the Edit DDS source **I** button or click **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 menu items.
- 3. Press Ctrl-F to open the Find/Replace window.
- 4. In the Find field, type INDTXT and click Find.

itor - Find and Replace	
Find INDIXI Replace with	_
 Replace this occurrence 	
• Replace, then find <u>n</u> ext	E Pattern match
Replace <u>all occurrences</u>	⊠ <u>W</u> rap
Find Replace Options.	Cancel Help

- 5. Press Ctrl-N to find the next occurrence.
- 6. Delete the second INDTXT line. Type D in the number column and press Enter.

You have edited the source to fix the error by switching to the editor and you are ready to begin "Exercise 7.14: Compiling your source changes and closing the Designer."

Exercise 7.14: Compiling your source changes and closing the Designer

Before you begin, you must complete "Exercise 7.13: Switching between designing and editing the screen" on page 145.

Now you will compile the source on the iSeries just as you did in the Remote Systems LPEX Editor and then close the Designer.

To compile your source:

- 1. Click **File > Save** from the Designer menu to save your source to the iSeries.
- Click Tools > Compile from the Designer menu and then click No prompt (or click the compile button on the CODE Designer toolbar).
- **3.** A message indicates when the compile is complete. Click **OK** in the Message dialog. If you re-compile and run the payroll program, you will see the 4 digit year change you made to the opening screen of the program.

To close the Designer:

1. Click File > Exit from the Designer menu.

Now you are ready to review your knowledge of this module by taking the quiz. You can also apply what you have learned in this module by completing the practice tasks detailed in More practice.

Quiz

- 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. Match the item with its correct description.
 - a. Utility notebook
 - b. Design page
 - c. Properties notebook
 - d. Group
 - a. Contains several different tabbed pages
 - b. For visually designing screens
 - c. A collection of one or more DDS records that make up a single screen or report
 - d. Access properties of a field, record, or entire file
- 3. The Utility notebook contains:
 - a. Selected DDS page

Note: You will also see the message CPD7886W Field overlaps another field with no conditions specified. You can ignore this message.

- b. Error list page
- c. Web Settings page
- d. Comments page
- e. All of the above
- 4. 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
- 5. Grouping records together allows you to work on one record while still seeing the related records in the background. (T, F)
- **6**. The Properties notebook is modeless. When you change an object's properties, the selected object changes immediately. (T, F)
- 7. The DDS source is checked using the same verifier that the LPEX Editor or CODE Editor uses. (T, F)
- 8. You can switch between Design mode and Edit mode in CODE Designer. (T F)
- 9. You can compile source on the iSeries just as you do in the LPEX Editor. (T, F)

More practice

Given your experience in 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 the exercises in this module as you create your report. When you are familiar with the file REFMST, you can create your printer file. Use the product online help to assist you in this task.

Module recap

You have completed Chapter 7, "Module 7. Designing screens," on page 123. You have learned how to:

- Open a DDS member for edit with CODE Designer
- · Show file-level keywords and record-level keywords
- View the details of records, record-level keywords and field-level keywords
- View the design of the payroll application main menu
- Create a group from an existing record format
- Create a new group and add a subfile record and a subfile control record
- Add columns to the subfile record
- · Add fields to the subfile control record
- Copy existing fields
- Set indicators to handle field errors
- View and update record and field properties
- View keywords and the properties of a keyword

- Insert a keyword
- View help for a keyword
- Check there are no semantic errors in the DDS source
- View help for an error
- Launch the editor in read mode from the error list
- Launch the editor in write mode to fix the error
- Find a keyword in the source
- Save source changes
- Compile your source changes
- Close the Designer

You have learned about various aspects of the CODE Designer while modifying a display file and now can learn more about the product and how it is packaged. This is an optional module. Continue to Chapter 8, "Module 8. Introducing the product and Remote System Explorer (optional)," on page 149.

Chapter 8. Module 8. Introducing the product and Remote System Explorer (optional)

This modules teaches you about IBM WebSphere Development Studio for iSeries 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 iSeries application development tools included with Development Studio Client for iSeries programmers. You then are introduced to Remote System Explorer the launching point for iSeries application development tools.

In this module, you will:

- Know the goals of the product
- Know the editions of the product
- Identify the host tools and the client tools
- · List and describe the iSeries application development tools

Exercises

The exercises within this module must be completed in order:

- "Introducing Development Studio and Development Studio Client"
- "Introducing iSeries Application Development Tools" on page 150

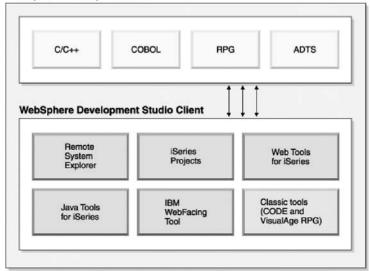
Time required

This module will take approximately 15 minutes to complete.

Introducing Development Studio and Development Studio Client

Development Studio Client is the ideal set of workstation development tools for creating, testing, deploying and maintaining traditional and e-business applications for your iSeries server. Development Studio Client is included with the compiler-based server product, WebSphere Development Studio. The following diagram illustrates the interaction between host and client tools:

WebSphere Development Studio



Development Studio Client is designed to help you:

- 1. Develop and maintain iSeries applications using the Remote System Explorer.
- 2. Develop Web-enabled front-ends to iSeries business logic.
- 3. Create GUI front-ends to iSeries business logic.

Both Development Studio Client and Development Studio Client Advanced Edition are built on the Rational Software Development Platform. This platform offers a fresh look and feel for the Eclipse workbench, and helps make it easier for you to build, integrate, and extend your applications. The Rational Software Development Platform offers a tutorials gallery and a samples gallery, to help you get up and running with the product as quickly as possible. The platform also offers user roles, which you can select from bottom-right corner of the Welcome view, (and from Window > Preferences > Workbench > Capabilities) that customize and simplify the user interface according to your programming role. These are just some of the new features provided in the Rational Software Development Platform.

The product comes in two editions for iSeries programmers. Both editions of the product are packaged with an additional base Rational product:

- Development Studio Client is packaged with Rational Web Developer. Rational Web Developer includes support for Web services, XML development tools, and core support for Java and Web development tools.
- Development Studio Client Advanced Edition is packaged with Rational 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 JavaBeans[™], EGL, and for Java messaging services

You have reviewed the goals of the product and the product editions and you are ready to begin "Introducing iSeries Application Development Tools."

Introducing iSeries Application Development Tools

Before you begin, you must complete "Introducing Development Studio and Development Studio Client" on page 149.

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. What are they and what do they do?

Remote System Explorer

You can manage your development 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 you can 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 without errors 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 Commands 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, you 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.

iSeries Debugger

With the Integrated iSeries debugger you can debug an application that is running on an iSeries system. It provides an interactive graphical interface that makes it easy to debug and test your iSeries programs. It is fully integrated into the workbench. 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.

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. CODE Designer is not fully integrated into the workbench, but you can launch it as a separate tool from the workbench.

You have identified the host tools and the client tools as well as the iSeries application development tools and now you are ready to review your knowledge of this module by taking the quiz.

Quiz

- 1. WebSphere Development Studio for iSeries:
 - a. Includes all four host compilers
 - b. Includes all four host compilers and the workstation-based tools named Development Studio Client
 - c. Includes only the workstation-based tools named Development Studio Client
- 2. WebSphere Development Studio Client for iSeries includes:
 - a. Rational Web Developer
 - 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 includes:

- a. Rational Application Developer
- b. Cooperative development environment (CODE)
- c. VisualAge RPG
- d. Java tools
- e. Web Tools
- f. WebFacing Tool
- g. All of the above
- 4. Rational 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. Rational Web Developer includes support for:
 - a. Web services
 - b. XML development tools
 - c. Java tools
 - d. Web tools
 - e. All of the above
- 6. The editor can access source files on your workstation or on your iSeries system directly. When a compilation or verify action 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)
- 7. You can debug your program running on the iSeries system from your workstation using the Integrated iSeries Debugger. (T, F)
- **8**. 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
- 9. 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
- 10. 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)
- 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

Module recap

You have completed Chapter 8, "Module 8. Introducing the product and Remote System Explorer (optional)," on page 149. You should now understand:

- · What components make up Development Studio and Development Studio Client
- What components are included in Rational Application Developer and Rational Web Developer
- What tools you use to manage your development cycle tasks
- · What the benefits are of Remote System Explorer

Finish your tutorial by reviewing the materials in Chapter 9, "Summary," on page 155.

Chapter 9. Summary

This tutorial has taught you how to maintain a payroll application using the Remote System Explorer. You learned how to start the product and open the Remote System Explorer perspective and how to use tools and views in this perspective to connect to an iSeries system and edit, verify, compile and debug the payroll application.

Completed Learning Objectives

If you have completed all of the modules, you should now be able to:

- Start the product and open the Remote System Explorer perspective
- · Create a connection to an iSeries and select iSeries objects from this connection
- Use the Remote Systems LPEX Editor to edit source
- Verify and compile source in the Remote Systems LPEX Editor
- Debug your interactive payroll application from the workstation
- Modify a display file
- · Recognize the product features and packaging

More information

For more information on the product and the Remote System Explorer, see http://ibm.com/software/awdtools/iseries.

Quiz answers

Here are the answers to each module quiz.

1	2	3	4	5	6	7	8
1d	1c	1d	1d	1c	1d	1b	1b
2e	2e	2T	2d	2T	2c	2 aa, bb, cd, dc	2g
3b	Зе	3a	3g	3e	3T	3e	3g
4 aa, bd, cb, dc	4c	4T	4T	4c	4b	4h	4e
5e	5T	5d	5T	5c	5T	5T	5e
	6T	6c	6d	6T	6T	6T	6T
	7d	7b	7e	7e	7c	7T	7T
		8c	8T	8T	8T	8T	8f
		9c		9a	9T	9T	9e
		10a		10a	10f		10T
		11b			11T		11b
		12T			12e		
		13d					
		14e					
		15d					

	16c			
	17b			
	18c			
	19T			
	20T			

Appendix. Notices

Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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 that 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 described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing 2-31 Roppongi 3-chome, Minato-ku Tokyo 106-0032, Japan

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, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF 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 include 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 serve 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. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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:

Lab Director IBM Canada Ltd. Laboratory 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports 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 used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1992, 2005. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM iSeries OS/400 RPG/400 Rational VisualAge WebSphere

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

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT[®], Win32, Win32s and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.



Printed in USA