

IBM WebSphere Development Studio Client for iSeries

Using the Remote System Explorer (RSE) COBOL application

Version 6.0



IBM WebSphere Development Studio Client for iSeries

Using the Remote System Explorer (RSE) COBOL application

Version 6.0

Contents

	V
Chapter 1. Module 1. Starting the product and the Remote System	4
Explorer. Exercise 1.1: Starting the product	1 1
Exercise 1.2: Opening the Remote System Explorer	3
	5
Chapter 2. Module 2. Configuring a connection to an iSeries system and	
connecting to an iSeries	7
Exercise 2.1: Configuring a connection to an iSeries	-
	10
Exercise 2.2: Connecting to an iberies system Exercise 2.3: Viewing and accessing objects in the	10
Remote System Explorer perspective	14
Exercise 2.4: Opening a second source member 2 Exercise 2.5: Displaying an outline of a source	20
member	20
Chapter 3. Module 3. Editing source 2	5
Exercise 3.1: Introducing the editor	26
Exercise 3.2: Changing default editor settings 2	26
Exercise 3.3: Entering SEU commands	<u>2</u> 9
Exercise 3.4: Requesting undo and redo operations	30
Exercise 3.5: Invoking language-sensitive help	31
Exercise 3.8: Finding and replacing text	36
Exercise 2.0: Filtering lines by string	20
Exercise 3.9. Filtering lines by stilling	20
Exercise 5.10: Filtering lines by type	20 40
Exercise 3.11: Searching multiple files.	ŧU
Exercise 3.12: Comparing file differences from the	
Remote Systems view	ŧ1
Exercise 3.13: Comparing files in the CODE Editor	
(optional)	1 5
Exercise 3.14: Checking syntax	17
Chapter 4. Module 4. Verifying and	
compiling source 5	3
Exercise 4.1: Verifying the source	53
Exercise 4.2: Compiling source remotely	55
Exercise 4.3: Submitting iSeries commands in the	,0
iSeries table view	59
Exercise 4.4: Rupping commands and programs	51
Exercise 4.4. Running commands and programs)1
Chapter 5. Module 5. Debugging a	-
program 6	1
Exercise 5.1: Introducing the Integrated iSeries	
Debugger	58
Exercise 5.2: Starting the integrated debugger 6	58
Exercise 5.3: Setting breakpoints	73
Exercise 5.4: Monitoring variables	76

Exercise 5.5: Stepping into a program.					82
Exercise 5.6: Listing call stack entries					84
Exercise 5.7: Setting breakpoints in PAYRO	LL	D			85
Exercise 5.8: Removing a breakpoint in PAY	YR	OL	LD		87
Exercise 5.9: Monitoring variables in PAYR	OL	LD	١.		88
Exercise 5.10: Adding a memory monitor					90
Exercise 5.11: Setting watch breakpoints .					91
Exercise 5.12: Closing the debug session.				•	94

Chapter 6. Module 6: Exploring Remote

System Explorer	97
Exercise 6.1: More about the Remote System	
Explorer	97
Exercise 6.2: Creating a library filter	98
Exercise 6.3: Creating an object filter	102
Exercise 6.4: Creating a user action	106
Exercise 6.5: Running commands from the Remote	
System Explorer	111

Chapter 7. Module 7. Designing

screens
Exercise 7.1: Opening a DDS member in the
Remote Systems view
Exercise 7.2: Viewing the DDS tree
Exercise 7.3: Selecting the DDS object
Exercise 7.4: Designing the DDS screen
Exercise 7.5: Creating groups from existing records 120
Exercise 7.6: Creating new screens
Exercise 7.7: Adding fields to the subfile record 124
Exercise 7.8: Switching between multiple records 127
Exercise 7.9: Adding field error handling 129
Exercise 7.10: Accessing field properties 131
Exercise 7.11: Adding new keywords
Exercise 7.12: Verifying the source changes 135
Exercise 7.13: Switching between designing and
editing the screen
Exercise 7.14: Compiling your source changes and
closing the Designer

hapter 8. Module 8. Introducing the oduct and Remote System Explorer	
ptional)	
troducing Development Studio and Development	
udio Client	
troducing iSeries Application Development Tools 142	
hapter 9. Summary147	
ppendix. Notices	ļ
ogramming interface information	ļ
ademarks	

Introduction

This tutorial teaches you how to maintain a payroll application written in ILE COBOL 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 iSeries T^{M} 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 COBOL
- 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
- · 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 a list of all help contents
- Limit the search of help to specific documents
- Search the help
- 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 53 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
- 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 67 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 97 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 115 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 141 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 141.

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.

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:

2. 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 Systems 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
- · 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" on page 20
- "Exercise 2.5: Displaying an outline of a source member" on page 20

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.

Such	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.	ections can be sharable by t y identify you from your tea ther it is owned by the team	he team or private to m members. You will n profile or your
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					×
Remote iSeries S Define connection	ystem Connec information	tion		-6-	-
Parent profile:	msteckh7tp				•
Connection name:	s400a				
Host name: Description:	s400a				
🔽 Verify host nan	ne				
			\bigtriangledown		
	< Back	Next >	Finish	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 view.

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:

- 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**.

and Execution	Initial Library L	ist	
Library List	Library list		
con -	Library:		(Add(B)
	Library	Library Position	
	RSELABXX	*LAST	
			Remove
			Move Up
			Move Down
	Current library:	*USRPRF	-
		Restore Defaul	ts 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.

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 up through the files in the Remote Systems view until you find QCBLLESRC 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 COBOL 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.

	CODSSRC. *file. EMPMST.pf MSTDSP.dsp PRJMST.pf REFMST.pf RSNMST.pf	of Go To Refresh Expand + Collapse -	•
Properties 🖾	日韓国	🕶 🛴 Rename	
Property	Value	Copy	
Attribute	PF-SRC	🕆 Paste	
Name	QDDSSRC		
Number of chi	5	+∓+ Move	
Source	RSELABXX	💥 Delete	
Status	OK		_
Text		III Show in Table	
Туре	*FILE	Carbo File Descriptions	
		SY Find String	
		Change(B)	
		Save	

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-s QDDSSRC.*file.pf-s CDDSSRC.*file.pf-s CDDSSRC.*file.pf-s CDDSSRC.*file.pf-s CDDSSRC.*file.pf CDDSSRC.*fil	STC 1	iasks Tivise ile RSELABXX	ries × VQDDSSR«	iSeries C (5 Members)	» ₁ — 🗖
•			Name	Туре	Attribute	Text 🔺
	1 1 1 T		EMPMST	PF	SRC	
			MSTDSP	DSPF	SRC	
Property	Value		PRJMST	PE	SRC	
Attribute	PF-SRC		D DEEMST	DE	SPC	-
Name	QDDSSRC			FI	JUNC	
Number of chi	5		-			<u> </u>
Source	RSELABXX					
Status	OK			I the second		
Text			Command	Jivormal	<u> </u>	
Туре	*FILE					
1446/2007			Messages	1		Details,
1						

The top of the iSeries Table view contains a lock icon it 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 Help Install/Update Internet Iseries Projects	Re-open to previous state Command line for action parameter Specify columns and column order	s
J2EE	Type: Member	-
Lava LPEX Editor Remote Systems Communications Debug File Cache Files Series Cache Command Executio Command Subsyste Job Subsystems LPEX Editor Parsers Objects Program Verifiers Table View Templates	Available columns Name Type	elected columns Vame (ype Attribute fext itatus the selected colu

- 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.

5 @ ¢		Work with 🕨 🕨			
Name	PF	Subset			
Customized PF Derault Re All		Show columns Position to			
		Print Export to File Hide Commands			
Comman	d Normal	Preferences			

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

ile RSELABXX/QI	DDSSRC (5 Mem	bers)		Ĝ	F \$\$ \$ \$ \$ \$ \$
Name	Туре	Attribute	Text	Sta	Last modified
EMPMST	PF	SRC		OK	January 3, 2005 3:3
MSTDSP	DSPF	SRC		OK	January 3, 2005 3:3
🖞 PRJMST	PF	SRC		OK	January 3, 2005 3:3
REFMST	PF	SRC		OK	January 3, 2005 3:3
RSNMST	PF	SRC		OK	January 3, 2005 3:3

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.

ile RSELABXX/QD	DDSSRC (5 Mem	bers)	-		<u>F 6 6 6 6</u>
Name	Туре	Attribute	Text		Last modified
EMPMST	PF	SRC		OK	January 3, 2005 3:3
MSTDSP	DSPF	SRC		OK	January 3, 2005 3:3
🖞 PRJMST	PF	SRC		OK	January 3, 2005 3:3
la refmst	PF	SRC		OK	January 3, 2005 3:3
T RSNMST	PF	SRC		OK	January 3, 2005 3:3.

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

7. In the Remote Systems view, select QCBLLESRC.

The table shows the members in QCBLLESRC.

PRJMST.*file.pf-dta QCBLLESRC.*file.pf-src QCMDSRC.*file.pf-src QMDSSRC.*file.pf-src QDDSSRC.*file.pf-src		iSeries × iSeries Remote [≫] 1 [□] □ Ne RSELABXX/QCBLLESRC (3 Members)				
E E	Name	Туре	Attribute	Te:		
	PAYROLLC	CBLLE	SRC	wit		
	PAYROLLC2	CBLLE	SRC	wit		
Value	Th PAYROLLD	CBLLE	SRC	no		
PF-SRC			1.000			
QCBLLESRC						
3				-		
RSELABXX						
OK	Coursed In					
	Command Nor	mai 📩				
*FILE	Messages		•	Details.		
	PRJMST.*file.pf-dta QCBLLESRC.*file.pf-src QCLSRC.*file.pf-src QCMDSRC.*file.pf-src ODDSSRC.*file.pf-src Value PF-SRC QCBLLESRC 3 RSELABXX OK *FILE	PRJMST.*file.pf-dta QCBLLESRC.*file.pf-src QCMDSRC.*file.pf-src QCMDSRC.*file.pf-src QDDSSRC.*file.pf-src Name PAYROLLC PAYROLLC	PRJMST.*file.pf-dta CellESRC.*file.pf-stc QCLSRC.*file.pf-stc CellESRC.*file.pf-stc QDDSSRC.*file.pf-stc Mame QDDSSRC.*file.pf-stc PAYROLLC Name Type PAYROLLC CBLLE Value PAYROLLC PF-SRC PAYROLLD QCBLLESRC SELABXX OK Command *FILE Messages	PRJMST.*file.pf-dta Imit Series X iSeries Remote *1 QCBLLESRC.*file.pf-src Imit Series X iSeries Remote *1 QCUSRC.*file.pf-src Imit Series X iSeries Remote *1 QCMDSRC.*file.pf-src Imit Series X iSeries Remote *1 QCMDSRC.*file.pf-src Imit Series X iSeries Remote *1 QCMDSRC.*file.pf-src Imit Series X iSeries Remote *1 Name Type Attribute Imit Series Imit Series Imit Series Name Type Attribute Imit Series Imit Series SRC Imit Series Imit Series SRC		

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" on page 20.

Exercise 2.4: Opening a second source member

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 PAYROLLC in the QCBLLESRC 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.

You have opened another member for edit 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."

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 COBOL source:

- 1. Look at the Outline view to the right of the editor window.
 - **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 ENVIRONMENT DIVISION.
- 3. Expand the CONFIGURATION SECTION.
- 4. Expand DATA DIVISION.
- 5. Expand FILE SECTION.
- 6. Double-click on any of the entries in the Outline view. This will position the source editor accordingly.



7. Stay in the **PAYROLLC** tab to get the PAYROLLC Editor window in focus for the next module.

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
- 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 COBOL source member PAYROLLC, 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
- 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 29
- "Exercise 3.4: Requesting undo and redo operations" on page 30
- "Exercise 3.5: Invoking language-sensitive help" on page 31
- "Exercise 3.8: Finding and replacing text" on page 36
- "Exercise 3.9: Filtering lines by string" on page 38
- "Exercise 3.10: Filtering lines by type" on page 38
- "Exercise 3.11: Searching multiple files" on page 40
- "Exercise 3.12: Comparing file differences from the Remote Systems view" on page 41
- "Exercise 3.13: Comparing files in the CODE Editor (optional)" on page 45
- "Exercise 3.14: Checking syntax" on page 47

Note: Exercises 3.6 and 3.7 do not appear as they apply only to RPG.

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.
	In which as much	-	1
	orkbench		-
E A	uld Order		
	ale order		
	eip stall/Llod	ato	
	ternet	ale	
	eries Proi	octe	
10	STES FIU	60.5	
- 12 	ava		
E-I	EX Editor	5	
and the owned	Appear	ance	
	Block		
	Compa	re	
	Contro	ls	
	Find Te	ext	
E	Parser:	5	
	Print		
	Save		
	Tabs		
	User A	ctions	-
	- User C	ommands	
	- User Ke	ey Action	s
	- User M	ouse Acti	ons
	User Pr	rofile	
E	E-C:\Doc	uments a	nd Sett
E	E C:\Doc	uments a	nd Sett
4			F

You will change the default settings of Appearance and User Key Actions.

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.
- Change Background color to light green. Notice how your changes are reflected in the sample edit view.

Workhench	•
	Appearance
Build Order	Palette Foot
Help	
Install/Update	white 🗾 Courier New-regular-12
Internet	Styles
iSeries Projects	formatione
- J2EE	messageLine
Java	parserDefault Background
LPEX Editor	parserMessage
Appearance	prefixarea
BIOCK	selection Outline
Compare	statusLine
Controls	Preview
Find Text	Line 1 Column 1 Poplage
🕂 Parsers	Line i column i kepiace
Print	
Save	Emphasis, current line, default text
Tabs	** ERROO1 Embedded parser message.
User Actions	Selection Text in parser default sty
User Commands	
User Key Actions	۲ () () () () () () () () () (
	LPEX 3.0.0 (build 22-Sep-2004)
User Profile	
C: Documents and Sett	,
	Reset Restore Defaults 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.

Workbench	User Key Action	ns	
Build Order	Key	Action	
∃ Help	enter	splitLine	Set
-Install/Update			Delete
E iSeries Projects	h h	\$	
J2EE			
- Java			
LPEX Editor			
Appearance			
Block			
Compare			
Controls	-		
Find Text			
连 - Parsers			
Print			
Save			
- Tabs			
User Actions	L V Luchur	— - Ì	
	Key lenter	Enter	
User Key Actions	Action splitLine		
User Mouse Actions			
User Profile			
• C:\Documents and Sett			
+ C:\Documents and Sett -	d	11	
	-	Reset Restore De	faults 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.
- 5. Click **OK** on the Preferences window. Return to the Editor window.

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.

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.

MSTDSP.DSPF	PAYROLLC.CBLLE ×			
Line 148	Column 1 Replace			
	-+-*A-1-B+2+3-	+	4+	5
dd	01 WS-RSCDE	PIC	X(8).	
000145	01 WS-ACODE	PIC	A.	
000146	01 WS-ACREC	PIC	X.	
000147	01 WS-EMESS	PIC	X(50).	
dd				
000149				
000150	PROCEDURE -DIVISION.			
000151	T			
000152	0000-MAIN.			
000153				-
•				•

- 2. On any sequence number type dd.
- **3**. 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 29.

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.

4. Click **File > Close** on the workbench menu.

Tip: You can also click the X on the PAYROLLC tab.

MSTDSP.DSPF	/ *PAYROLLC.CBL	I X				E
Line 144	Column 1	Replace	1	change		
	+-*À-1-B+	-2+	-3-	+	-4	-+5
000140	05 IND-P	ROTECT		PIC	9(2)	VALUE _

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

Save Resource		A	×
? 'PAYROLLC.CBLLE' has b	een modified. Sav	ve changes?	
4			
	Yes		Cancel

- 5. Click No.
- 6. Go back to the workbench to the Remote Systems view and open the PAYROLLC member in the QCBLLESRC 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 30.

Inside the editor, there is cursor-sensitive language-reference help available.

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 231 of the ILE COBOL 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 So	itware Development Platform
Search:	GO Search scope: NyList
Contents 🖧 🗄	(⇒) (p) (⇒) (⇒) (⇒) (⇒) (⇒) (⇒) (⇒) (⇒) (⇒) (⇒
 Introduction to Web Installing and migra Navigating and custo Connecting to Linux, Working in a team e Developing iSeries s Developing Web app Developing Java app Developing GEL appl Building XML applica Detecting and publishing Troubleshooting and Debugging application Reference Glossary 	ILE COBOL Language Reference MOVE Statement The MOVE statement transfers data between areas of storage. The MOVE statement transfers data between areas of storage. The MOVE statement has two formats: Format 1 - MOVE • Format 2 - MOVE CORRESPONDING Related Information: • MOVE Statement Rules • Elementary Moves • Group Moves • WHEN-COMPILED Special Register [Top of Page Previous Page Next Page Table of
Glossary	Contents Index] (C) Copyright IBM Corporation 1992, 2005. All Rights Reserved.

3. Click the link Format 1- MOVE.

The Help page for Format 1- MOVE is displayed.



- 4. Play around in the Help window to see what else is available.
- 5. Minimize the Help window.
- 6. Select **Help > Help Contents** on the workbench menu 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 COBOL
- 12. Select Language Reference.
- 13. Scroll down to MOVE Statement Format 1.



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

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		_	
Search:	G0 🤇	earch scope NyLis	t		
Contents	â. H	Reference	<u>~</u>	1 🖗 🗐 🗁	
Testing and pul	Select Search Scop	e	×	Jage	
🌳 Troubleshootin 🏅	Search all topics				-
🌳 Debugging appl 🖇	Dearch only the follow	ing topics:			
Reference					
🗉 💷 APIs					
🗄 💷 Commands					
E 🛄 Debuggers					
E E Files				e	
Language refer				se this	
ISeries prog				N Dalatad	
				nd Related	
🗉 💷 ILE	New Edit	Remove		four	
🗉 💷 co	\sim $-$				
🗆 💷 ILE				L	
				his Release?	
		OK	Cancel	<u>s Guide</u>	
⊞ Ц⊒ DB;				VEDOO	
E CE PPG/4	100		What's New i	<u>m VSR27</u>	
E 🛄 ILE R	PG	0	What's New i	$n \sqrt{4P42}$	
🕀 💷 Data Defi	nition documentation		What's New i	$n \sqrt{4R2?}$	
🗉 💷 Control la	nguage information 🖵	0	What's New i	n √3R7?	
d	<u> </u>	0	What's New i	n V3R6/V3R2?	2
		4	1.6.0 .1 61 .	10000	٠ſ

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 Search Sco	×
 Search only the follow MyList 	ig topics:
New Edit	Remove
	Cancel

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

C Help	- IBM Rational Software	Develo	pment 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 3.8: Finding and replacing text."

Exercise 3.8: Finding and replacing text

Before you begin, you must complete "Exercise 3.5: Invoking language-sensitive help" on page 31.

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.

2. Click **Edit > Find/Replace** from the workbench menu or press **Ctrl+F**.

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

ace 3+45
+3+4+5+++ intenance display file ployee master file oject master file ason code master file -FILE
intenance display file
ployee master file oject master file ason code master file
oject master file ason code master file
ason code master file
-FILE
TO WORKSTATION-MSTDSP-SI
ZATION IS TRANSACTION
IS SEQUENTIAL.
-FILE
TO DATABASE-EMPMST
CATION IS INDEXED
KEY IS EXTERNALLY-DESCRIBED-KEY
IS DYNAMIC
Next Previous All

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 MSTDSP.

- 3. In the Find field, enter MSTDSP to find the start of a file section.
- 4. Make sure the **Replace** field is blank.

You would use this field for text replacement.

5. Click **Next** to go to the next location of MSTDSP in the file.

The Editor moves the active line to line 37, which contains the first MSTDSP phrase 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 38.

Exercise 3.9: Filtering lines by string

Before you begin, you must complete "Exercise 3.8: Finding and replacing text" on page 36.

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 EMPNO in the Editor window.
- 2. Select Edit > Selected > Filter Selection from the workbench menu.

6		PAYROLLC.CBLLE ×	MSTDSP.DSPF
	ce	Column 11 Repl	Line 142
4+5+	+3+	-*A-1_B+2	
			Ð
(6).	PIC	01 WS-EMPNO	번 000142
S-EMPNO.	TO	MOVE EMPNO	000302
F RCEMP	MPNO TO EMPNO	MOVE WS	000307
O EMPNO OF EMPMNT-	IS-EMPNO	MOVE	A 000320
OF RCEMP.	IO TO EMPN	MOVE US-EM	A 000353

- **3**. Move the cursor down a few lines to line 302.
- 4. Expand line 302. This expands the section up to the next instance of EMPNO.

MSTDSP.DSPF	PAYROLLC.CBLLE ×
Line 302	Column 1 Replace
-	+-*A-1-B+2+3+4+5+
H 000142	01 WS-EMPNO PIC 9(6)
(A) 000302	MOVE EMPNO TO US-EMPNO.
000303	MOVE ACODE OF EMPSEL-I TO WS-ACODE.
000305	IF IND-OFF(IND-MAINT) AND IND-OFF(IND-EOJ) SET IND-OFF(IND-MOT-FOUND) TO TRUE
000307	MOVE WS-EMPNO TO EMPNO OF RCEMP
H 000320 H 000353	NOVE US-EMPNO TO EMPNO OF EMPMNI-C MOVE US-EMPNO TO EMPNO OF RCEMP.

Now you want to show the entire source again.

5. 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."

To help you navigate quickly through your ILE COBOL source the editor allows you to filter lines based on the line type. Imagine you want to see where all the divisions are defined in your source.

Line 317 000304 000305 000306 000307 000308	Column -*A-1-B+ IF	38 2- IND-OF	Replace	34	<mark>-</mark> -4·	
+ 000304 000305 000306 000307 000308	-*A-1-8+ IF	IND-OF	+	3+		
000304 000305 000306 000307 000308	IF	IND-OF				
000.309		MOVE U START	ID-OFF (IN IS-EMPNO EMPMST-F INVALID	INT) A D-NOT- TO EMH ILE KEY	AND IN FOUND PNO OF	D-O T RC
000310	Prompt	F4	SET I	ND-ON	(IND-N	-TC
000311 000312	Syntax Check Line Copy Book		PFF (IND	-NOT-E	OUND)	
000313 000314 000315	Cut Copy Paste	Ctrl+X Ctrl+Insert Ctrl+V	EMPMST /ALID K READ	-FILE EY SEI	TIND-0	ON (
000316 000317 000318	Select Selected	oltau	CORR	OF RCE RCE	MP TO	WS_ EM
000319	Filter view	ALCTO	Date	EMPMNI	-0	м
000321 000322 000323 000324	Show all Source View	Ctrl+W	Divisions Comments Outline Embedded Errors	SQL/CICS/I	Ctrl+G	50 R ()
000326 000327 000328	Save Show Date Area		WS-EME WS-ACO	SS TO DE TO	EMESS ACODE	OF OF
-	Add Breakpoint					•

To filter lines by type:

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

All source lines with divisions are displayed allowing you to move quickly and easily to the desired division in your file.

MSTDSP.DSPF	PAYROLLC.CBLLE X
Line 150	Column 1 Replace
-	+-*A-1-B+2+3+4+5+
H	TRENTTETCATION REVISION
000005	PROGRAM-ID. PAYROLL.
000018	ENVIRONMENT DIVISION.
000064	DATA DIVISION.
000150	PROCEDURE DIVISION.
	Ĩ

- 3. Move your cursor to the line with the DATA DIVISION (line 64).
- 4. Expand the division to show all lines in this division.
 - Now you could work with the source inside this division.
- 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" on page 40.

Exercise 3.11: Searching multiple files

Before you begin, you must complete "Exercise 3.10: Filtering lines by type" on page 38.

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.

File Edit Source G	ompile Navigate	earch Project	Run Window	Help
🖻 • 🗄 🖆] 🕏	s • O • Q • Z	🖗 Search	Ctrl+H	Remote Syste
1 🗸 😒 🛛 🔹 I	▶ A 📝 🖻 🛓	P File		EniSeries Projects
MSTDSP.DSPF	PAYROLLC.CB	iSeries		Ð
Line 150	Column 1	V Remote	•	

The Search window opens.

2. In the Search string field, type ENHRS.

onnection: s400a	New
Library RSELABXX	Browse
File: QDDSSRC	Browse
Member *ALL	▼ · Browse
Source members	🗖 Data members
Columns	
All columns	
C Between 1 and(G) 80	
C Between(H) 1 and end of	fline

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.	. iSerie	s iSeries I	Remote S	×	-	° 🗆
"ENHRS" - 3 Occur	rences in 3 f	Members				23	Se	•
E-C RSELABXX	(QDDSSRC)	EMPMST) (1	match)					
Line :	18) A	ENHRS R						
E- C RSELABXX	(/QDDSSRC(MSTDSP) (1 r	natch)					
Line 9	99) A	ENHRS R	B 11	37REFFLD(RCEMP	/ENHRS *LIBL/E	MPMST	()	
E- C RSELABXX	(/QDDSSRC(REFMST) (1 r	natch)					
(Line 2	27) A	ENHRS	31	TEXT('EMPLOYEE	NORMAL WEEK H	OURS	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	0	PAYROLLC.CB	LLE 💋	REFMST.PF	X	_	5
Line 27		Column 1	Rep	lace			
	. A		.Name+-	++++RLer	1++T	ЪрВ	Functions++++++
000023	A		ELOCN		30		TEXT (' EMPLOYEE 🔺
000024	À						COLHDG ('EMPLOY
000025	A		EUSRI		8		TEXT ('EMPLOYEE
000026	A						COLHDG ('EMPLOY
000027	A		ENHRS		3	1	TEXT ('EMPLOYEE
000028	À						COLHDG ('NORMAL
000029	A		EPHRC		5	1	TEXT ('PROJECT
000030	A						COLHDG('PRJ HR
000031	A		EPHRY		7	1	TEXT ('PROJECT
000032	A						COLHDG ('PRJ HR
000033	A		EPHRP		7	1	TEXT ('PROJECT
000034	A						COLHDG('PRJ HR-
000035	A		EPNRC		5	1	TEXT ('NON PROJ
000036	A						COLHDG ('NON PR
000037	A		EPNRY		7	1	TEXT (NON PROJ
000038	A						COLHDG ('NON PR
000039	A		EPNRP		7	1	TEXT ('NON PROJ
000040	A						COLHDG ('NON PR
000041	A		EHWRK		5	1	TEXT ('EMPLOYEE
000042	A						COLHDG('EMP HR
000043	A *	PROJECT	MASTER	RELATED	FIE	LDS	
000044	A		PRCDE		8		TEXT ('PROJECT
000045	A						COLHDG (' PROJEC 🚽
•							Image: A start of the start
_							

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 40.

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:

- Click Window > Preferences from the workbench menu. The Preferences window opens.
- 2. In the left pane of the Preferences window, expand LPEX Editor.

Preferences		_ [] >
ELPEX Editor	Compare	
Appearance Block Compare - Controls - Find Text - Parsers - Print - Save - Tabs - User Actions - User Actions - User Commands - User Key Actions - User Mouse Actions - User Profile - C:\Documents and Sett - C:\Documents and Sett	Ignore case Ignore leading blanks Ignore trailing blanks Ignore all blanks Ignore sequence numbers Ignore comments Column restricted comparison Start column In column 80	Ŗ
Remote Systems Remote Systems Run/Debug Server Validation Web and XML Web Diagram WebFacing WebFacing		
	Res	et Restore Defaults Apply

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 PAYROLLC member double-click the **PAYROLLC** tab.
- 6. Click Edit > Compare > Compare to file on the workbench menu.

			E Kemote Syste	B
ledo.	Cbri+Y		,	
Iut	Ctrl+X			
Iopy	Ctrl+Insert	Replace		
Paste	Ctrl+∀	.Name+++++	RLen++TDpB	• •
Delete	Delete	ENHRS	3 1	
Select All	Ctrl+A			
Find/Replace	Ctrl+F	EPHRC	5 1	
Find Previous		EPHRY	7 1	
ing Other Show All	Ctrl+W	EPHRP	7 1	
Add Bookmark Add Task		EPNRC	5 1	
Select	•	EPNRY	7 1	
Selected Deselect	► Alt+U	EPNRP	7 1	
Mark	۲.	EHMRK	5 1	
Iompare	•	🗥 Compare to file		
(eystroke Recorder	•	Next Mismatch	୯tିନ୍ଧି+Shift+N	
now Tooltip Descript	ion F2	Previous Mismatch	Ctrl+Shift+P	
Loncent Assist	CONTOPACE	Refresh	Ctrl+Shift+R	
046 A		Clear		
047 A		nppen	20	
149 A		EKKOP	50	
050 A		PRSTR	R	
				Þ
	Lut Lut Lopy Paste Select All Select All Select All Select All Select All Select All Select All Add Bookmark Add Task Select Select All Select All Add Bookmark Select All Select All	Lut Ctrl+X Lopy Ctrl+Insert Lopy Ctrl+Insert Raste Ctrl+V Delete Delete Lind/Replace Ctrl+F Lind/Replace Ctrl+F Lind Next Shift+F4 Lind Other + Lind Other + Lind Other + Lind Resk Ctrl+W Add Bookmark - Add Task - Lielect + Lielectd + Lielectd + Lielectd + Lielextroke Recorder + Lihow Toolbip Description F2: - Compare + D46 A D47 A D48 A D49 A D50 A	Lut Ctrl+X REFMST.PF × Lopy Ctrl+Insert Replace Lopy Ctrl+Insert Replace Late Delete Name + + + + + + + Delete Delete ENHRS Lind/Replace Ctrl+A E PHRC Lind/Replace Ctrl+F E PHRC Lind Next Shift+F4 E PHRC Lind Other Image: Ctrl+W E PHRY Lind Other Image: Ctrl+W E PNRY Lind Task E PNRY E PNRY Lielect Image: Ctrl+U E PNRY Lielect Image: Ctrl+SPACE Next Mismatch Deselect Alt+U E HMIRK Compare Image: Ctrl+SPACE Next Mismatch Previous Mismatch Previous Mismatch Previous A PRR S P D46 A PRS TR D48 A PRS TR D49 A PR STR	Ctrl+X REFNST.PF X Copy Ctrl+Insert Replace Addeed Delete Image: Ctrl+Y Delete Delete ENHRS 3 ind/Replace Ctrl+A EPHRC 5 1 ind/Replace Ctrl+F EPHRC 5 1 ind/Replace Ctrl+F EPHRY 7 1 ind Other + + EPHRY 7 1 ind Other + + EPHRY 7 1 idect - - EPHRY 7 1 idect - - EPNRY 7 1 idect - - - Compare to file - idect d - - - - - - idect d - - - - <td< td=""></td<>

The Compare window opens.

- 7. Expand your connection.
- 8. Expand *LIBL.
- 9. Expand RSELABXX.
- 10. Expand QCBLLESRC.
- 11. Select member PAYROLLC2
- 12. Click OK.



The editor now will show the differences of these two members PAYROLLC and PAYROLLC2 .

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

Mismatches in PAYROLLC and PAYROLLC2 are highlighted in different colors so that you know where the mismatched lines are in each file.

MSTDSP.DSPF	PAYROLLC.CBLLE ×
Line 87	Column 1 Replace
	+-*A-1-B+2+3+4+5+
000078	COPY DDS-ALL-FORMATS OF PRJMST.
000079	
000080	FD RSNMST-FILE.
000081	01 RSNMST-REC.
000082	COPY DDS-ALL-FORMATS OF_RSNMST.
000083	L
000084	WORKING-STORAGE SECTION.
000085	
000086	01 ERROR-MESSAGES.
000087	OS MSG-TORIF
000088	10 MSG-1 PIC X(50) VALUE
000088	10 MSG-1 PIC X50) VALUE
000089	MAINTENANCE SELECTION CODE NOT EQUAL
000090	10 MSG-2 PIC X(50) VALUE
000091	'MORE THAN ONE APPLICATION SELECTED FOR M
000092	10 MSG-3 PIC X(50) VALUE
000093	' NO APPLICATION SELECTED FOR MAINTEN.
000094	10 MSG-4 PIC X(50) VALUE
000095	ACTION CODE NOT EQUAL TO "A", "C" OR
000096	10 MSG-5 PIC X(50) VALUE
000097	' ADD REQUESTED BUT RECORD ALREADY EXISTS
000098	10 MSG-6 PIC X(50) VALUE
000099	' WARNING - RECORD WAS PREVIOUSLY DELE
Property of the local data and t	

 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 **PAYROLLC** 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 41.

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 PAYROLLC in QCBLLESRC.
- 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 PAYROLLC2 member.
- 4. Click **File > Open** from the CODE Editor menu.

The Select file - Open for Edit dialog opens.

Select file - Open for Edit						×
File name	<r0s400a>RSELABXX/</r0s400a>	QCBLLESRO	(PAYROLLC2)	Source type	CBLLE	•
Contains sequence <u>p</u> umbers			Subset	by source type	< All Files >	•
<r0s400a>RSELABXX/QCBLLESI</r0s400a>	R Contents of ' <r0s40< td=""><td>0A>RSELAB</td><td>X/QCBLLESRC(*)</td><td></td><td></td><td></td></r0s40<>	0A>RSELAB	X/QCBLLESRC(*)			
 → <local> C:\Program Files'</local> → <ros400a></ros400a> → ODEVTOOLS → ODEVTOOLS → OGPL → OHLPSYS → OSYS → OSYS2 → OUSRSYS → RSELABXX → OMPABE → ODEMESEC 	PAYROLLC PAYROLLC2 PAYROLLD	CBLLE CBLLE CBLLE	with syntax & veri with verify errors no errors	fy errors		
QLLSRL QCMDSRC QDDSSRC	Befresh Sort	Subset:	Eind:		4	1
<r0s400a>RSELABXX/QCB</r0s400a>	LLESRC[PAYROLL	C2]				

- 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 QCBLLESRC.

- 8. In the right pane of the Select file-Open for Edit dialog, select member PAYROLLC2 from the member list.
- 9. Click OK.
- Click Actions > Compare from the CODE Editor menu. The Compare dialog opens:

Compare			×
<r0s400a>RSE</r0s400a>	LABXX/QCBLI	LESRC(PAYRO	LLC2]
Select or enter a	file name		
<r0s400a>RSE</r0s400a>	LABXX/QCBL	LESRC(PAYR)	DLLC]
<r0s400a>RSE</r0s400a>	LABXX/QCBL	LESRC(PAYR)	OLLC)
			•
			_
Compare	Cancel	Help	

All entries are preloaded.

11. Click Compare.

The editor now has the PAYROLLC2 member loaded on the left and member PAYROLLC 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.



12. Use the vertical scroll bars to move within the files.

As you scroll, you will see where the differences are in the members. 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 45.

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. Click the PAYROLLC Editor window, click **Source** and then click **Syntax Check All** on the workbench menu.

Remote	System Explorer - PAYROLLC.CBLL	E - IBM Rational Software Development Pl 💶 🕽
e Edit 🤤	Source Compile Navigate Search	Project Run Window Help
9 • G	Syntax Check All	🔹 🗘 🔹 🚽 👘 🖬 🖬 Remote Syste
~ ~ ~ I		
	g venity Ctri+Shirt+V	
MSTDS	Yerify (Prompt)	6
Line	Show fields	ace
	Copy Book	·3+4+5+
00001-		TS OF PRJMST.
00001	ILE COBOL Help	•
30000	Prompt E4	
_ 30000	Prompt 14	
30000	Resequence All Lines	TS OF RSNMST.
30000	Refresh Ctrl+F5	
-30000	wormente prototop	we CTION.
000085		
000086	OF WSC TABLE	5.
000008	10 MSG-1	PIC VEO) VALUE
000000	10 M3G-1	NTENANCE SELECTION CODE NOT FOUND T
000090	10 MSG-2	PIC X (50) VALUE
000091	MORE T	HAN ONE APPLICATION SELECTED FOR MA
000092	10 MSG-3	PIC X(50) VALUE
000093	' N	O APPLICATION SELECTED FOR MAINTENA
000094	10 MSG-4	PIC X(50) VALUE
000095	5 ' AC	TION CODE NOT EQUAL TO "A", "C" OR
000096	5 10 MSG-5	PIC X(50) VALUE
000097	' ADD R	EQUESTED BUT RECORD ALREADY EXISTS
000098	10 MSG-6	PIC X(50) VALUE
000099) ' WA	RNING - RECORD WAS PREVIOUSLY DELET
000100	10 MSG-7	PIC X(50) VALUE
4		<u>•</u>
-		
1		
1		

Error messages appear to draw attention to the errors.

MSTDSP.DSPF	PAYROLLC.CBLLE X	Ð
Line 88	Column 1 Replace	
	+-*A-1-B+2+3+4+5+	
000078	COPY DDS-ALL-FORMATS OF PRJMST.	
000079		
000080	FD RSNMST-FILE.	
000081	01 RSNMST-REC.	
000082	COPY DDS-ALL-FORMATS OF RSNMST.	_
000083		
000084	WORKING-STORAGE SECTION.	
000085		
000086	01 ERROR-MESSAGES.	
000087	05 MSG-TABLE.	
000088	10 MSG-1 PIC X50) VALUE	
LNC2	530E Unmatched right parenthesis. Right parenthesi	s i
LNC:	248E PICTURE string 'X50)' does not describe valid	dat
000089	MAINTENANCE SELECTION CODE NOT EQUA	LT
000090	10 MSG-2 PIC X(50) VALUE	
000091	'MORE THAN ONE APPLICATION SELECTED FOR	AM S
000092	10 MSG-3 PIC X(50) VALUE	
000093	NO APPLICATION SELECTED FOR MAINT	ENA
000094	10 MSG-4 PIC X(50) VALUE	
000095	ACTION CODE NOT EQUAL TO "A", "C"	OR
000096	10 MSG-5 PIC X(50) VALUE	
000097	' ADD REQUESTED BUT RECORD ALREADY EXIS	TS
000098	10 MSG-6 PIC X(50) VALUE	-
•		
Syntax che	ck action completed with informational and/or error	me: Z

- 2. Move the cursor onto the pink error message.
- **3**. Press **F1**.



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

- 4. Minimize the Help window.
- 5. Add the required left parenthesis to correct the error.
- 6. 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**, select the language you want to change the settings for in the left pane of the Preferences window and select the **Automatic syntax checking** check box and then click **OK**.

Preferences	
🛨 Workbench 📃	COBOL
E Ant	Automatic syntax checking
	Automatic syntax and any
E-Install/Undate	I Automatic syntax checking or SQL statements (iseries connect —
	Automatic uppercasing
iSeries Projects	User defined tabs
J2EE	General
🕀 Java	+ 1 + 2 + 3 + 4
E LPEX Editor	
Remote Systems	AA A
Communications	And every: 4
	No. Staty 1
	43
Aiseries	
Cache	
Command Executio	
- Command Subsyste	
Job Subsystems	
ELPEX Editor Parsers	
	Restore Defaults Apply
Tunnaut Evenaut	OK Creat
Export	

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 LPEX Editor has predefined settings, but also has an associated preference page containing settings that you can define. (T, F)
- 2. LPEX Editor preferences are set in the:
 - a. Preferences window
 - b. Editor window
 - c. New wizard
 - d. Remote Systems view
- **3**. 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
- 5. 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

- 6. To receive language sensitive help, press the _____ key in an Editor window. Name the key:
 - a. F2
 - b. F3
 - **c**. F1
 - d. F4
- 7. 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
- 8. 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
- 9. 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
- 10. 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
- 11. 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
- 12. 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
- **13**. 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
- 14. There are two ways to compare files. They are Compare tool in the workbench and the Compare tool in the CODE Editor. (T, F)
- **15**. 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)

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
- 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 53.

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
- 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 55
- "Exercise 4.3: Submitting iSeries commands in the iSeries table view" on page 59
- "Exercise 4.4: Running commands and programs" on page 61

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.)

You will need to open the COBOL source file named PAYROLLC2 to complete this verify step.

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

Tasks	iSeries Table	iSeries Comm	Remote Search	Remo	te Syste	× iSeries Error	X 🖓
<tor <="" th=""><th>AS07M>RSELAB</th><th>X/QCBLLESRC(PAY</th><th>ROLLC2)</th><th></th><th></th><th>36</th><th>× + + •</th></tor>	AS07M>RSELAB	X/QCBLLESRC(PAY	ROLLC2)			36	× + + •
	ID	Message	Severity	Line	Location		Conn
	LNC1904	Program PAYROLL	40	1	RSELABXX,	QCBLLESRC(s400a
	LNC1326	PRCD OF PRJSEL-I	' 30	402	RSELABXX/	QCBLLESR.C(s400a
1	LNC1329	Subscript value '14'	30	606	RSELABXX/	QCBLLESRC(s400a
	LNC1463	'EMPNO' is not uniq	30	302	RSELABXX/	QCBLLESR.C(s400a
0	LNC0407	AT END phrase mis.	20	399	RSELABXX/	QCBLLESRC(s400a
Ø	LNC0407	AT END phrase mis.	20	448	RSELABXX/	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis.	20	499	RSELABXX/	QCBLLESR.C(s400a
0	LNC0407	AT END phrase mis.	20	548	RSELABXX/	QCBLLESRC(s400a
Ø	LNC0407	AT END phrase mis.	20	184	RSELABXX/	QCBLLESRC(s400a
C	LNC0407	AT END phrase mis.	20	299	RSELABXX/	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis.	20	348	RSELABXX/	QCBLLESRC(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 LNC1463.

You are automatically brought back into the Editor window to the line where the error occurred. The error is caused by EMPNO not being uniquely defined. Go to line 302 where EMPNO is defined. The variable EMPNO should be EMPNO OF EMPSEL-I.

2. Make the change.

The next error is LNC1326.

- 3. Double-click LNC1326. Fix it in the editor.
- 4. PRCD should really be PRCDE.

Make the appropriate change on line 402.

5. Double-click LNC1329.

This error is because the array was declared with 4 elements. Go to line 606.

6. Change the index of the array from 14 to 4.

The next errors are LNC0407. You can ignore these errors as they are severity 20.

All severity 30 errors and above are now fixed. You can filter out different severities by using the filter menu.

Task	s i	Series Table	iSeries Comm	Remo	ote Search	Re	emo	te Syste	×ais	eries Error	X	-
<to< th=""><th>RAS</th><th>507M>RSELAB</th><th>X/QCBLLESRC(PA</th><th>YROLL</th><th>C2)</th><th></th><th></th><th></th><th></th><th>x</th><th>× (+ +)</th><th>8</th></to<>	RAS	507M>RSELAB	X/QCBLLESRC(PA	YROLL	C2)					x	× (+ +)	8
		ID	Message	1	Severity	L	~	i Informati	ion	Show	Severity 🕨	E
		LNC1904	Program PAYROLL		40	1		A Warning		Messa	iges 🕨	
~		LNC1326	PRCD OF PRISEL	-I'	30	4			A-	SRC(s400a	
-		LNC1329	Subscript value '1-	4'	30	e	~	C Error	0	SRC(s400a	
~		LNC1463	'EMPNO' is not uni	iq	30	З	~	Severe		SRC(s400a	
1	0	LNC0407	AT END phrase mi	is	20	З	~	Terminati	na	SRC(s400a	
	0	LNC0407	AT END phrase mi	is	20	4	10			SRC(s400a	
	0	LNC0407	AT END phrase mi	is	20	49	99	RSELABXX	/QCBLL	ESRC(s400a	
	0	LNC0407	AT END phrase mi	is	20	54	18	RSELABXX	/QCBLL	ESRC(s400a	
	0	LNC0407	AT END phrase mi	is	20	18	34	RSELABXX	/QCBLL	ESRC(s400a	
	0	LNC0407	AT END phrase mi	is	20	29	99	RSELABXX	/QCBLL	ESRC(s400a	
	0	LNC0407	AT END phrase mi	s	20	34	18	RSELABXX	/QCBLL	ESRC(s400a	

- 7. Click the arrow in the iSeries Error List title bar.
- 8. Click **Show Severity** on the pop-up menu.
- 9. 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.

asks	iSeries Table .	iSeries Comm	Remote Search	Remo	te Syste	Xa iSeries Error	× 🖓
TORA	AS07M>RSELA	3XX/QCBLLESRC(PA	YROLLC2)			36	× + +
	ID	Message	Severity	Line	Location		Conn
0	LNC0407	AT END phrase mis	s 20	184	RSELABXX	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis	s 20	299	RSELABXX,	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis	s 20	348	RSELABXX,	QCBLLESRC(s400a
3	LNC0407	AT END phrase mis	s 20	399	RSELABXX,	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis	s 20	448	RSELABXX,	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis	s 20	499	RSELABXX,	QCBLLESRC(s400a
0	LNC0407	AT END phrase mis	s 20	548	RSELABXX	QCBLLESRC(s400a
					1		

Everything should be ok. You should see only severity 20 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 53.

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	
± Ant	Compared States and States and	
Build Order	Command Execution Preferences	
∃-Help		
Install/Update	Preferences for compiles and user action variables	
H Internet	Object library:	*SRCLIB
12EE		1
T. Java	Replace object	
En Java	(Compile in batch	
Remote Systems	•	
Communications		
⊕ · Debug	Preferences for batch compiles, commands, and user	action variabl
- File Cache	Job description library:	*LIBL
Files	Job description:	*LISPPR
⊟-iSeries	Job description.	
Cache	SBMJOB additional parameters:	
- Command Executio	Add batch compiles to the iSeries Job Status view	
- Command Subsyste	Add batch compands to the Carios Job Status via	
- Job Subsystems	Add bacch commands to the idenes bob blacus vie	WV
LPEX Editor Parsers		
	Preferences only for compile commands	
TEPPC	Compile member types in this order:	
	PF 🔺	
	LF Move Up	
+ Program Verifiers	PRTF	
- Table View	ICFF Mayo Dorug	
- Templates	RPGLE	
Logging		Restore
	2 <u></u>	

- 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 CRTBNDCBL 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 PAYROLLC2 member in QCBLLESRC.



- Click Compile (Prompt) > CRTBNDCBL on the pop-up menu. The Create Bound COBOL Program (CRTBNDCBL) dialog opens.
- 3. In the **Debug view** list, select the *ALL parameter.

Create Bound COBOL Program (CR	TBNDCBL)	×
Program:	> PAYROLLC2	Name
Library:	> RSELABXX 💌	Name
Source file:	> QCBLLESRC	Name
Library:	> RSELABXX 💌	Name
Source member:	> PAYROLLC2	Name
Output:	*PRINT	
Generation severity level:	30	0-30
Text 'description':	*SRCMBRTXT	Character value
Compiler options:	Advanced Parameters Add *EVENTF Remove Move up Move down	
Debug view:		
Replace program:	> *YES •	
Advanced(X)	All Parameters(Y) Keywor	rds(Z)
CRTBNDCBL PGM(RSELABXX/PAYROLLC2) DBGVIEW(*ALL) REPLACE(*YES)	SRCFILE(RSELABXX/QCBLLESRC) SRCMBR(PAYROLLC2) OF	PTION(*EVENTF)
	CK Restor	e 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.

Tasks	iSeries Table Vi	BiSeries Comma	X Remote Search	Remote System	Series Error List	- 8
					4	2 💌
S400 CR OP Pro Cai libr Com	a TBNDCBL PGM(RSEL TION(*EVENTF) DBG gram PAYROLLC2 c use : The p ary on the stated d mand Normal	ABXX/PAYROLLC2) SR SVIEW(*ALL) REPLACE reated in library RSELA rogram has been comp ate and time.	CFILE(RSELABXX/QCBLI (*YES) BXX on 01/11/05 at 14: iled and translated succ	LESRC) SRCMBR(PA) 14:42. essfully and placed i	(ROLLC2) n the indicated Prompt	I I I III

5. Click the **iSeries Commands Log** tab from the view 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 55.

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.

PAYROLLC CBLLE	SRC	with syntax & verify errors	OK
PAYROLLC2 CBLLE			OK
	SRC	with verify errors	OK
PAYROLLD CBLLE	SRC	no errors	ОК
4			

To change the library list:

- 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 table view is not linked to an active connection.

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

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



b. Click Show in Table on the pop-up menu.

The 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.

Tasks 🕞 ISeries Ta	ble View 🗙 iS	eries Commands Re	mote Search Remote System Det iSe	eries Error List 🗁 🗗
File RSELABXX/QCBLI	ESRC (3 Members))
Name	Туре	Attribute	Text	Status
PAYROLLC	CBLLE	SRC	with syntax & verify errors	OK
PAYROLLC2	CBLLE	SRC	with verify errors	OK
PAYROLLD	CBLLE	SRC	no errors	OK
•				
•				
Command Norm	al 💌		•	Prompt Run
Messages Curre	nt library changed	to RSELABXX.	Detail	s 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 59.

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.



- 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 🗙	Team	- 0	PAYROLL	2" ³²		Ho 8 - 0
& @ ← ⇒	~ E	\$3 ▼	Line	448	С	ණ
	MYCOMPILE PAYROLL.*P PAYROLLG. PAYROLLG. COMPARE.* EMPMST.*Fi EVFEVENT.* MSTDSP.*fi MSTDSP2.*fi MSTDSP2.*fi QCBLLESRC C PAYROL QCLSRC.*Fi QCMDSRC.* QCMDSRC.* MSTDSF PRJMST PAJROL QCDSSRC.* MSTDSF PRJMST REFMST	© Go To gm.rr Go To Show Collap Collap Copy Paste → Move. Copy Paste Copy Nove. Copy Paste Copy Refre. Copy Paste Copy Refre. Copy Nove. Collap	000437 000437 in Table sh nd - me me te ge in table	· int · REC · LE · Ogr		IDENTIFICATION D ENVIRONMENT DIV DATA DIVISION. PROCEDURE DIVIS ote *1
•		User /	Actions			[
Properties & Re	emote Scra	Run A Run(P Debug Debug	ls Prompt)(B) g(Service Entry) g As(F))類 iSe)	ries app tch :eractive	lication in RSE Job
Attribute	CBLLE	Debug	g (Prompt)(J)	🕨 🕃 Mu	lti-threa	ided
Name Number of children	PAYROLL(Prope	erties(K)		1	

- 1. In the Remote Systems view, locate the PAYROLLC2 program that you created.
- 2. Right-click the PAYROLLC2 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
- 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 53. 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 67.

Chapter 5. Module 5. Debugging a program

This module teaches you how to debug a CL and ILE COBOL 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 68
- "Exercise 5.2: Starting the integrated debugger" on page 68
- "Exercise 5.3: Setting breakpoints" on page 73
- "Exercise 5.4: Monitoring variables" on page 76
- "Exercise 5.5: Stepping into a program" on page 82
- "Exercise 5.6: Listing call stack entries" on page 84
- "Exercise 5.7: Setting breakpoints in PAYROLLD" on page 85
- "Exercise 5.8: Removing a breakpoint in PAYROLLD" on page 87
- "Exercise 5.9: Monitoring variables in PAYROLLD" on page 88
- "Exercise 5.10: Adding a memory monitor" on page 90
- "Exercise 5.11: Setting watch breakpoints" on page 91
- "Exercise 5.12: Closing the debug session" on page 94

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

The Debugger supports 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 COBOL program PAYROLLD.

Note: PAYROLLD is the same COBOL program as PAYROLLC2 but without compile errors. You are using it instead of PAYROLLC2 in this exercise, to accommodate anyone who decided to skip right to this exercise without completing the exercises in Chapter 4, "Module 4. Verifying and compiling source," on page 53.

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 CLC1 to call PAYROLLD and you will pass one parameter to CLC1.

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 CLC1 in library RSELABXX.
- 4. Click the arrow beside the **DEBUG** icon **b** 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	and the second state of the second states and the second states of the	
reate, manage, and run conf	gurations	1. Ale
Attach to a Running Pro Attach to a Running Pro Compiled Application Host Java Application Series: Debug Batch Ap Series: Debug Job Series: Debug Job Series: Debug Job Series: Debug Multi-Thr Java and Mixed Langua 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 whe when a breakpoint is encountered via the Debug perspective switch should not occur, select "None" Debug: Debug Restore Defaults	s: Debug Interactive Application y be associated with each supported en a configuration is launched or preferences. 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.

connection: 1540	Jua			• <u>New</u>
roorams to be de	buaaed.			
he checked prog	ram or service proc	gram will be used as in	nitial debug objec	:t.
Library	Program	Туре		Add
RSELABXX	CLC1	*PGM	R	Edit
				Remove
Chan into				
Step into		ware assemblishing		

8. In the Name field, type the program name CLC1.

- 9. Select the Update production files check box.
- 10. Click the **How To Start** tab.

The How To Start page opens.

ne: CLC1				
What To Debug	How To Start	Source	Advanced	🗆
Command to start appl	ication:			
CALL PGM(RSELABXX)	/CLC1)			-
	-			4
	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	m (CAI	LL)							×
Program:	>	CLC1		Name					
Library:	>	RSELABXX	-	Name					
Parameters:	(XX				4	Ad	ۍ کړ	
F Advanced		II All	Parameter	rs		🗆 Keywor	ds		
CALL PGM(RSEL	ABXX/C	LC1) PARM('XX')							-
									<u>*</u>
		[OK	>	Restor	e defaults		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.

🕑 Debug	and the second system of the second system of the	2
Create, manage, and run con	ìgurations	Ť
Configurations: Apache Tomcat Attach to a Running Prc Compiled Application Host Java Application HTML Script Application Kiseries: Debug Batch Ac Kew_configuration Series: Debug Multi-Thr Java and Mixed Langua Java Applet Java Application Java Application Java Application Script Host Application Script Host Application KebSphere v5.0 Applic. WebSphere v6.0 Applic	Name: CLC1 What To Debug How To Start Command to start application: FALL PGM(RSELABXX/CLC1) PARM(XX')	urce Advanced Common
New Delete		Apply Revert
		Debug Close

14. Click Debug.

The Debug perspective opens.

If not, you may see this error message.

No interactive job associated with this iob with this connection to continue	connection, Associate an i	nteractive
	Cancel	<< Details
A request to run an interactive command or pr connection s400a, but no interactive job is cur To associate an interactive job with this conn 1. Open a 5250 emulator, 2. Run the CL command STRRSESVR N 3. This will associate the interactive jo	ogram was received for the rrently associated with this ection: VAME(s400a) be with this connection, cause	e RSE <u>*</u> connection.

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.

Oebug - CLC	1.CLLE - IBM Rationa	l Software Develo	pment Platform	_ D ×
File Edit Source	e Compile Navigate	e Search Project	Run Window Help	
10.08	☆・0・8	• Q. •	28	E A Debug
$] \Leftrightarrow \Leftrightarrow \bullet \Rightarrow$	*] 🖉 😒 🗉		R 19	Remote Syste
₩ Debug 🖾	Servers	- 6	Variables 🔍 Breakpo 🕅	Registers 3 🗖 🗖
		•	× % &	\$\$ \$ 19 h +
	8 🛬 3. 3. 10. 10			
	eries: Debug Interactiv	e Application]		
🗐 🔐 State	e: <stopped> Connecti</stopped>	on: iSeries(TM) ("		
	Thread:1 (stopped) = act:act:act			
	CL_PEP : CLC1 : C	LC1		
	QCMDEXC : QCMDE			
(RPAVPOLL	/ PAVPOLI		× F	
Tipo 1	Colu			An outline is not available.
TILE 1	+1	+2	+3+-	
♦ 000100		PGM	PARM (#)	
000200		DCL	VAR (#)	
000300		dcl	&name *cha	
000400		<u>dcl</u>	&lib *char	
000500		DCL	VAR (&count	
000600	TOOD	CHGVAR	VAR (&NAME)	
000700	TOOA:	<u>engvar</u>	var («counc	
Canada Tarka		(B M	.r.	
Console Tasks	iSeries Commands Log		W	
	1			
1 1				

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 68.

You can only set breakpoints at executable lines. All executables 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.

1			NU ELEC	200000
Debug - CLC File Edit Sour	1.CLLE - 1 ce Comp	IBM Rational Software Dev ile Navigate Search Proje	Select Selected))
	1 1/2 -	0 • % • % •]	Deselect	Alt+U
] *= 🗇 🔸 🔿	*		Filter view	•
] 🖉 😒 🗉	_ ▶		Show all	Ctrl+₩
PAYROLLC2.	CBLLE	PAYROLLC.CBLLE	Source	•
Line	11	Column 1	View	•
Dille .	+	1	Add to Snippets	
000100		PGM	Sava	
000200		DCL	Show Date Area	
000300		dcl -	Dilow Date Area	
000400		dcl	Add Breakpoint	
000500		DCL	Add Watch Break	pint
000600		CHGVAR	Run To Location	
000700	LOOP	: <u>chgvar</u> -	Monitor Expression	
000800		<u>if</u> (&cc	Monitor Expression Monitor Memory	
000900		CHGVAR _	Monitor Memory	
001000		chgvar	Switch View	•
001100		CALL -	PGM (&Lib/	PAYROLLD)
001200				
001300		ENDPGM		

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.

PAYROLLC2	CBLLE	PAYROLLC.CBLLE	CLC1.CLE X MSTDSP.DSPF
Line	8	Column 2	Replace
	+	1+	-2+4
000100		PGM	PARM (#)
000200		DCL	VAR (#) TYPE (*char
000300		dcl	&name *char 10
000400		dcl	&lib *char 10
000500		DCL	VAR (&count) TYPE (*DE
000600		CHGVA	R VAR (&NAME) VALUE ('Cl
000700	LOOP	: chgva	ir var(&count) value(&c
2000800		<u>if</u> (8	count *lt 100) then(goto 1
000900		CHGVA	R VAR (&NAME) VALUE ('Cl
001000		chgva	r var(&lib) value('RSE
2001100		CALL	PGM (&lib/ PAYROLLD)
001200			
001300		ENDPG	-M

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

☑ 2 Line [Clu	1.CUE:11]	
✓ P Line [CL		
	<>> Go to File	
	Add Breakpoint	
	Edit Breakpoint	
	Select All	
	💹 Enable	
	Disable	
	🐹 Remove	
	Semove All	

- 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.

C Edit a Line Bre	akpoint			×
Sets a breakpoint	to stop executi	ion at a specific :	source line	\$
Executable (Option	nal)			
Object (Optional)				
Source:				
CLC1.CLLE	_			
	< Back	Next >	Finish	Cancel

7. Click Next.

Make the breakpo	int conditional up	on the followir	ng parameters	Th
				200
Frequency				
From: 99			-	
To: Infinity			-	
Europu 1			-	
Every: 11				
EverA: 1 T				
Every:] 1				
xpression:				
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."

Exercise 5.4: Monitoring variables

Before you begin, you must complete "Exercise 5.3: Setting breakpoints" on page 73.

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.

© Debug - CLC1.CLLE - IBM Rational File Edit Source Compile Navigate	Software Develop Search Project F	Cut Copy Paste	Ctrl+X Ctrl+Insert Ctrl+V
│覧・╗┢│歩・○・9; • │┶ ⇔ • → - │ ୰ ७ │ ⋓ ◎		Select Selected) () ()
Servers		Filter view Show all	Ctrl+W
		Source	,
CLC1 [iSeries: Debug Interactive /	Application]	Save	
El CLC1 : CLC1 : CLC1 = □ CL_PEP : CLC1 : CLC1 = □ CL_PEP : CLC1 : CLC1 = □ QCMDEXC : QCMDEXC	1 C : QCMDEXC	Edit Breakpoint Remove Breakpoint Disable Breakpoint Add Watch Breakpoint	
PAYROLL	CLC1.CLE X	Run To Location	
Line 8 Colum	n 24 Re	Monitor Expression Monitor Storage	┣,
000700 LOOP:	chgvar	Switch View	•
000900 001000 2 001100 001200 001300	CHGVAR Chgvar CALL ENDPGM	VAR(&NAME) var(&lib) PGM(&lib/P.	
•			

- 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** lie 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 82.

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.

© Debugger Message	×
Program received unmonitored exception N location referenced.	CH3601: Pointer not set for
	OK

- 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.

Oebug - CLC1.CLLE - IBM Ration	nal Software Developmen	nt Platform	_ 🗆 🗙
File Edit Source Compile Naviga	te Search Project Run	Window Help	
11 · II ▲ ☆ · O · 2	• Q. • 1 (29 A)	Fi 🏂 Debug	1
$ \leftarrow \leftarrow \bullet \bullet \bullet$		Remote Sys	ste
	R D	/	
]			
해 Servers 없	Breakpo	P Monitors X 4	
\$ Q Q	- 10 P	4 X 🖄 🖕 🖃	260
Server	Host name &CC	OUNT = 00100.	
PAYROLL PAYROLL	CICI.CLE X	MSTDSP	023
Line 10 Col	umn 1 Repl	.ace	An outline is
+1-	+2+	+3+-	not available.
000600	CHGVAR \	/AR (&NAME) 📥	
000700 LOOP:	chgvar v	var (&count)	
P000800	if (&count *	tlt 100) tl	
000900	CHGVAR V	/AR (@NAME)	
001000	chgvar v	var(&lib)	
		14	
	1		
Console Tasks iSeries Commands Lo	g 🚺 Memory 🖾	출 💥 🔍 🖺	
			1

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.

역 📕 Session A - [24 x 80]	_O×
File Edit View Communication Actions Window Help	
Display Program Messages	
Job 589773/STECKHAM/QPADEY000T started on 05/03/11 at 12:11:06 in subs Job 589773/STECKHAM/QPADEY000T held by user STECKHAM with option SPLFI Job 589773/STECKHAM/QPADEY000T released by user STECKHAM. MCH3601 received by procedure CLC1. (C D I R) C	ystem LE(*NO
Job 589773/STECKHAH/QPADEV000T held by user STECKHAM with option SPLFI Job 589773/STECKHAH/QPADEV000T released by user STECKHAM. MCH3601 received by procedure CLC1. (C D I R)	LE(*NO
Type reply, press Enter. Reply	
F3=Exit F12=Cancel	
1002 - Section successfully started	20/018
I I I I I I I I I I I I I I I I I I I	11.

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.

_			
onnection: s40	10a		▼ <u>New</u>
ograme or cervic	e program to be d	ebunned	
ibrary	Program	Type	Add
RSELABXX	CLC1	*PGM	Edit
			Remove
Step into	ua session on prod	ram completion	
Terminace Geo	og session on prog	ram completion	

g. Click the **How To Start** tab.

The How To Start page opens.

What To Debug	How To Start	Advanced
Command to start appl CALL PGM(RSELABXX)	ication: /CLC1)	E
		<u>r</u>
	Prompt	
		Apply Devert

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.

Program: Library: Parameters:	> CLC1 > RSELABXX	Name Vame	_	Add
C Advanced	 	arameters	<u>.</u> ∐ Keywords	
CALL PGM(RSELA	BXX/CLC1) PARM('XX')			

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 **CLC1** 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 76.

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 PAYROLLD 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 76.

2. Right-click anywhere in the Source view.

CLC1.CLLE	PAYROLLD.CB	LLE ×					ć	3
Line	152 _Col	umn	8	Replace			Browse	
	+-* <mark>A</mark> -1-	B+	2-	+	3+	4	-+5	-
• 000150	PROC	EDUR	E DIVI	ESION.			-	•
000151								
000152		Prompt		F4				
000153		Syntax C	Theck Line					
000154		Сору Во	ok		• E			
000155		Cut		Ctrl+X	E			
000150		Сору		Ctrl+Insert	Ē			1
000157		Paste		Ctrl+∀	E.			_
000150		Select			• OR) TO	TRUE		
000160		Selected			· BC.	1100.		
000161		Deselect		Alt+U				
000162	*	Filter vie	W		. ection	until	end o:	
000163		show all		Ctrl+W				
000164		Source			S-MAIN	THRU	0100-E2	
000165		Jource			EOJ).			
000166		Add to S	nippets					
000167		Save			E			
000168								
000109		Add Brea add Wat	akpoint ch Preakooir		E			
000171			сп ы вакрон	16				
000172		Run To L	ocation					
000173		Monitor E	Expression					
000174		Monitor S	Storage		•			
000175		Switch Vi	ew		Show *SO	URCE	1	
000176					Show *LIS	TING		
000177	* Wri	te t	he Mai	intenance	Show *STA	ATEMANT	en to	-1
•								-
								1
								٦

3. Click **Switch view > Show *LISTING** on the pop-up menu.

CLC1.CLLE	PAYROLLD LISTING	3			8
Line 390	Column	1 1	Insert		Browse
+	-12-	+		4	-5+
➡ 390	201	000150	PROCEDURE	DIVISION.	
391		000151			
392		000152	0000-MAIN.		
393		000153			
394	202	000154	OPEN	I-O MSTDSE	FILE
395		000155		EMPMST	-FILE
396		000156		PRJMST	-FILE
397		000157		RSNMST	-FILE.
398		000158		т	
399	203	000159	SET 1	IND-OFF (IND	-ERROR)
400	204	000160	INITI	IALIZE MSTE	SP-REC.
401		000161			
402		000162*	• Display M	Maintenance	Selecti
403		000163			
404	205	000164	PERFO	ORM 0100-PR	OCESS-MA
405		000165	UNT	TIL IND-ON(IND-EOJ)
406		000166			
407	206	000167	CLOSE	E MSTDSE	FILE
408		000168		EMPMST	-FILE
409		000169		PRJMST	-FILE
410		000170		RSNMST	-FILE.
411		000171			
412	207	000172	STOP	RUN.	
413		000173			
414		000174			
415		000175	0100-PROCH	ESS-MAIN.	
416		000176			
417		000177*	Write the	e Maintenan	ce Selec
418		000178			
					-
<u> </u>					

- 4. Page down in the source and take a look at the expanded file descriptions. You don't have any /Copy member in your PAYROLLD program but these would also be shown in a Listing view. Switch back to the Source view.
- 5. Right-click anywhere in the Source view.
- 6. Click Switch view > Show *SOURCE on the pop-up menu.

You have stepped into PAYROLLD 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 82.

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 PAYROLLD."

Exercise 5.7: Setting breakpoints in PAYROLLD

Before you begin, you must complete "Exercise 5.6: Listing call stack entries" on page 84.

Now you add some breakpoints in PAYROLLD.

To add breakpoints:

- 1. Select PAYROLLD in Thread1.
- 2. In the source view (also called the iSeries default editor) scroll to line 201.
- **3**. Double-click the prefix area of line 201.

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 206.

CLC1.CLLE	PAYROL	.LD.CBLLE 🗙				B
Line 3	206	Column	1	Replace		Browse
	+-*j	A-1-B	+	-23	4	-+5-
000198	*	0500-R	SNMS	I-SELECT.	Ţ	
000199						
000200		IF	IND-	-OFF (IND-ERR	OR)	
<i>p</i> 000201			IF	(EMPAPL OF S	ELECT-I = 'X')
000202				INITIALIZE	EMPSEL-O	
000203				PERFORM 030	U-EMPMST-SELE	CT THR
000204			ET CI	UNTIL IND	-ON (IND-MAINT) OR II
000203			гирі	TE (DETADT.	OF SELECT-T =	1 1 1
000200				TNTTTAL	TZE PRISEL-O	
000208				PERFORM	0400-PRJMST-	SELECT
000209				UNTIL	IND-ON (IND-M	AINT)
000210				ELSE		
000211				INITIAL	IZE RSNSEL-O	
000212				PERFORM	0500-RSNMST-	SELECT
000213				UNTIL	IND-ON (IND-M	AINT)
000214				END-IF		
000215			END-	-IF		
000216		EN.	D-IF.	•		
000217		TP	TND	ON (THE MATH	m \	
000210		TE	CE4	TND-OFF (TND	יי הטממא. מש הט גמטאר	TIF
000210			TNT	TALTZE MSTD	SP-REC	.015
000221		EN	D-IF.			
000222						
000223	(0100-EX	IT.			
000224		EX	IT.			
000225						
•					1	
<u> </u>						

5. Repeat the above step for line 218.

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 PAYROLLD and now you are ready to begin "Exercise 5.8: Removing a breakpoint in PAYROLLD" on page 87.

Exercise 5.8: Removing a breakpoint in PAYROLLD

Before you begin, you must complete "Exercise 5.7: Setting breakpoints in PAYROLLD" on page 85.

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

To remove a breakpoint:

- 1. Right-click the prefix area of line 206.
- 2. Click **Remove Breakpoint** on the pop-up menu.

		/ PAYROLI	.D.CBLLE 🗙							B
Line	206	5 1	Column	1	Rej	place			Brow	se
		+-*A	-1-B	+	2	-+3	+	- 4	+	-5-
000204	1				UN	TIL IND	-ON (IND-	MAINT)	OR	11 -
000205				ΕL	SE					
2000206	Ē	Prompt			F4 /	TJAPL	OF SELEC	T-I =	'X')	
000207	1	Syntax	Check Line			ITIAL	IZE PRJS	EL-O		
000208	j n	Copy Bo	ook			REORM	0400-PR	JMST-S	ELEC	T
000209	? —	Cut			CENTER	UNTIL	TND-ON (IND-MA	(TNT)	(
000210	1	Copy			Ctrl+Insert	TUTAT	TTP DONG	RT-O		
000211		Paste			Ctrl+V	REORM	126 KaNa 0500-89		FLEC	Ψ
000212		Foloct				UNTT.	TND-ON	TND-MA	TNT)	-
000214	1	Selecter	4			, on the	1110 011,	1110 111		
000215	5	Deselec	t		Alt+U					
000216	5	ente				-				
000217	7	Show al	BW I		CELLU					
2000218	3	Di love di	2	10	CONTRACT	-)-MAIN	т)			
000219	9	Source				F (IND	-ERROR)	TO TRU	ΓE	
000220)	Add to S	5nippets			MSTD	SP-REC			
000221										
000222	<u> </u>	Save								
000223	5	Edit Bre	akpoint							
000229		Remove	e Breakpoint							
000220	5	Disable	Breakpoint		M					
000220	7	Add Wa	tch Breakpoir	nt			N			
000228	3	Run To	Location			01110				
000229		Monitor	Expression			ar di	splay fi	elds a	ind r	res
000230)	Monitor	Storage			•				
000231	L —	Switch (liaus) EMES	S OF SEI	ECT-0.		-1
		Switch	//eva				1			1
<u> </u>										

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

Now you are ready to run the PAYROLLD program.

3. Click the **Resume** icon from the Debug toolbar.

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



- 4. Type an X beside the Project Master Maintenance option.
- 5. Press **Enter** in the emulation session.

The program runs to the breakpoint at line 201.

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

Exercise 5.9: Monitoring variables in PAYROLLD

Before you begin, you must complete "Exercise 5.8: Removing a breakpoint in PAYROLLD" on page 87.

Now lets monitor variables and change them in PAYROLLD.

To monitor variables:

- 1. In the source view, double-click the variable EMPAPL on line 201.
- 2. Right-click the variable.
- 3. Click Monitor Expression on the pop-up menu.

0		/ PAY	ROLLD.CBLLE	×						8
	Line	201	Colum	nn	26		Repla	ce	Brow	wse
		+-	*A-1-B		+	-2-	+-	3+-	4+	5-
	000200)		ΙF	INI	-OF	F (IND	-ERROR)		-
ø	000201				IF	(EM	PAPI	Syntax Check S		1
	000202	2				IN	ITIA-	2,1121. 210212		_
	000203	3				PE	RFOF	Cut	Ctrl+X	1
	000204	1					UNTI	Сору	Ctrl+Insert	l l
	000205	5			ELS	δE		Paste	C01+V	_
	000206	5				IF	(PF	Select		+
	000207	7					IN	Selected		•
	000208	3					PE	Deselect	Alt+U	
	000209	3						Filter view		• -
	000210)				EL	SE	Show all	Ctrl+W	
	000211						IN_			
	000212	2					PE_	Source		•
	000213	3						Add to Snippets		(
	000214	ł				EN	D-IF_			_
	000215	5			ENI	D-IF		Save		
	000216	5		ENI	D-IF	•		Edit Breakpoint.		
	000217	7						Remove Breakp	oint	
2	000218	3		ΙF	INI	D-ON	(INE	Disable Breakpo	int	
	000219	3			SET	' IN	D-OF	Add Watch Brea	akpoint	
	000220	J			INI	TIA	LIZE-	Pup To Location		
	000221			ENI	D-TF		_	Kun to cocador		_
	000222		0100					Monitor Express	sion	
	000223	5	0100-	EX.	LT.			Monitor Storage	a hS	•
	000224	ŧ		EX.	гт.			Switch View		+
	000223	-								
	000226	2	0200	~	POR	CET		DUTON		
	000227		0200-	СНІ	SCK-	SEL	ECT-0	PTION.		-
	4									×

4. Click the Monitors tab in the upper right pane.

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

5. In the same way add the variables PRJAPL on line 206 and RSNAPL on line 244 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 RSNAPL. The value changes into an entry field.
- 7. In the entry field, type in the new value X for the variable.

Variables	Breakpoints	Monitors ×	Registers	Memory Rende	ering	Prog	grams		- 8
				с ^П а	×	*	¢,	2	60
• 8 • E • P	COUNT = 001 MPAPL = RJAPL = X SNAPL = X	00.	\mathbb{Q}						

8. Press Enter.

The variable is successfully changed.

You have monitored several variables in PAYROLLD and now you are ready to begin "Exercise 5.10: Adding a memory monitor" on page 90.

Exercise 5.10: Adding a memory monitor

Before you begin, you must complete "Exercise 5.9: Monitoring variables in PAYROLLD" on page 88.

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 IND-ERROR in line 219.
- 2. Right-click and select Monitor Storage > Memory on the pop-up menu.



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

D-ERROR : 0xdab1b99084	0021dc <hex></hex>			
lddress	0 - 3	4 - 7	8 - B	C - F
DAB1B99084002		000	00000000	00000000
DAB1899084002 - F	eset to Base Addr.	ess 000	00000000	00000000
DAB1B99084002 G	io to Address	N 000	DAB1B990	84002660
DAB1899084002	aluma Cina	000	DAB1B990	84002660
DAB1B99084002	Jului in 5128	000	00000000	00000000
DAB1899084002 P	eformat	000	00000000	00000000
DAB1899084002 🚔 😋	opy To Clipboard	000	3077EE90	OC0014FE
DAB1899084002		000	F323D4BE	39000080
DAB1899084002	nnc	000	00000000	00000000
DAB1899084002 D	ereference Pointe	, 000	00000000	00000000
DAB1899084002270	UUU7FUFI	FOF IF OF 1	F6000000	00000000
DAB1B99084002300	4040F0F0	40404040	40404040	40404040
DAB1B99084002310	40404040	40404040	40404040	40404040
DAB1B99084002320	40404040	40404040	40404040	40404040
DAB1899084002330	40404040	40404040	404040F0	40404040
DAB1B99084002340	40404040	40404040	40404040	40404040
DAB1B99084002350	40404040	40404040	40404040	40404040
DAB1B99084002360	40404000	00000000	00000000	00000000
DAB1B99084002370	40404040	40404040	40404040	40404040
DAB1B99084002380	40404040	40404040	40404040	40404040
DAB1B99084002390	40404040	40404040	40404040	40404040
DAB1B990840023A0	40400000	00000000	00000000	00000000
DAB1B990840023B0	80000000	00000000	3077EE90	0C001238
DAB1B990840023C0	00000000	00000000	00000000	00000000
DAB1B990840023D0	00000006	00000000	00000000	00000000
DAB1B990840023E0	00000000	00000000	00000000	00000000
DAB1B990840023F0	00000000	00000000	00000000	00000000
DAB1B99084002400	00000000	00000000	00000000	00000000
DAB1B99084002410	00000000	00000000	00000000	00000000
PAB1899084002420	40405050	40404040	40404040	40404040

- **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.



You have added a memory monitor for the variable IND-ERROR 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 90.

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 118 to go to that line.

CLC1.CLLE	/ PAYRO	.LD.CBLLE 🗙						5
Line 2	244	Column	1	Replac	ce			
	+-*;	A-1-B	+2	+	3	+	4	
000240								
000241		IF	((EMP.	APL OF	SELEC	r -1 =	'X	')
000242			(PRJ.	APL OF	SELEC	r -1 =	'X'	')
000243			((EMP.	APL OF	SELEC	r-1 =	'X	•)
000244			(RSN.	APL OF	SELEC	r -1 =	'X	•)
000245			SET I	VD-ON ()	IND-ER	ROR)	TO	F R
000246			MOVE :	ERRMSG	(2) TO	EMES	S O	E
000247		EL	SE					
000248			SET I	VD-OFF	(IND-E	RROR)	ΤO	ΤÌ
000249		EN	D-IF.					
000250							10	
000251	*	The fo.	llowin	g IF A	ND Com	binat	ion	e.
000252	*	applic	ation :	has be	en sel	ected		
000253								
000254		IF	((EMP.	APL OF	SELEC	$\mathbf{r} - \mathbf{I} =$		')
000255			(PRJ.	APL OF	SELEC	r- 1 =	. 1	')
000256			(RSN.	APL OF	SELEC	r -1 =	. 7	')
000257			SET I	ND-ON (IND-ER	ROR)	TO	rr'
000258			MOVE :	ERRMSG	(3) TO	EMES	S O	F
000259		EN	D-IF.					
000260			8					
000261	*	The fo.	llowin	g code	check	s eac	h aj	pp.
000262	*	ensure	that .	it is d	either	T T	(bl	an)
000263								
000264		IF	((EMP.	APL OF	SELEC	r-i N	OT	=
000265			(EMP.	APL OF	SELEC	r-i N	OT :	=
000266			SET I	ND-ON (IND-ER	ROR)	TO	F R
				1				, È
-								

- 2. Double-click variable IND-TABLE 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 IND-TABLE.

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

	D		2
Address or expression:	IND-TABLE		
Number of bytes to watch:	0	 	
			De .
			-0

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 IND-TABLE has changed.



- 8. Click OK.
- 9. In the Breakpoints view, right-click the Watch breakpoint and click **Disable** on the pop-up menu.

You have added a watch breakpoint for the variable IND-TABLE 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 91.

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:

ОК

Click **OK**.

opment f	Platform		- 🗆 🗙
un Windo	w Help		
*= ¢	• • •	Close All Close te	*
Variables	∫ [©] ⊜ Breakpoints 3	Dock On ► ✓ Show Text J	

- 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 67. 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 97.

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 98
- "Exercise 6.3: Creating an object filter" on page 102
- "Exercise 6.4: Creating a user action" on page 106
- "Exercise 6.5: Running commands from the Remote System Explorer" on page 111

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 97.

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				x
Library Filter Create a new iSeries	ibrary filter			
Library: RSE*			2	Browse
	< Back	Next >	Finish	Cancel

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	r easy re-use. Ite Systems vie	Specify a unio aw, and will be	que name for this expandable.	filter. This name will
Filter name:	RSE and VARP	Glibraries		
Only create filte	er in this conne	ction		
Select a profile to o sharable by the tea	wn the new fil m. It will be pl	ter. This deter aced in the de	mines if it is uniqu fault filter pool for	e to you, or r that profile.
Owner profile: mst	teckh7tp			-

- **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.

Remote Systems 🗙 Team	-
⊕ 🛣 Work with objects	
⊞	
🕀 📸 Library list	
⊡ 👬 User libraries	
All RSE and VARPG libraries	
E-IN RSEL/ New	
E IN RSEL/ Go Into	
RSEL/ Go To	
E-W RSEL	
RSEL/ Um Open in New Window	
H IN RSEL	
REL C Refresh	
Expand RSEL	+
Collapse	-
B-M RSEU NA Change	
RSEL	
H-M RSEL T Repare	
THIN RSEL	
E-IN RSEL	
E RSEL + Move	
🗈 🐘 RSEL/ 💥 Delete	
E M RSEL	
🔁 🐘 RSEL/ 🏠 Move Up	
RSEL/ Move Down	
RSEL/	
RSEL/ Show in Table	
E-WARSEL/	
E-W RSEL - Prind Scring	
RSEL/	

The Change Library Filter window opens.

© Change Library Filte	er 🗵
Filter name:	All RSE and VARPG libraries
Parent filter pool:	Msteckh7tp Filter Pool
Filter strings:	New filter string:
New filter string	New nicer schilig.
	Library VAR* I T
	Test Create Revert
	Cancel 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.

Remote Systems 🗙 Team	- 8
	- \$ →
🗉 🖍 RSELAB41.*lib.test	
🖭 🐘 RSELAB42.*lib.test	
🕀 📊 RSELAB43.*lib.test	
🕀 🖍 RSELAB44.*lib.test	
庄 🐘 RSELAB45.*lib.test	
😟 🐘 RSELAB50.*lib.prod	
庄 🐘 RSELAB60.*lib.test	
庄 航 RSELAB61.*lib.test	
🖅 🐘 RSELAB62.*lib.test	
庄 航 RSELAB63.*lib.test	
🗈 航 RSELAB64.*lib.test	
🖅 🐘 RSELAB65.*lib.test	
庄 航 RSELAB66.*lib.test	
🕀 🐘 RSELAB67.*lib.test	
🕀 🐘 RSESERVER.*lib.prod	
🕀 🐘 RSEWULKAN.*lib.test	
🖅 航 RSE100.*lib.prod	
🕀 🐘 RSE1000.*lib.prod	
🖅 🐘 RSE10000.*lib.prod	
庄 航 RSE5000.*lib.prod	
🔃 🐘 VARLABXX.*lib.prod	
🕀 🐘 VARLABOO.*lib.prod	
🕀 🔐 VARLAB10.*lib.test	
🕀 🐘 VARLAB11.*lib.test	
I VARLAB12.*lib.test	
🖅 🐘 VARLAB13.*lib.test	
WARLAB14.*lib.test	
WARLAB15.*lib.test	
E W VARPGSETUP.*lib.prod	
E ar iSeries Commands	1
	•

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 98.

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:

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



 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.

New				2
Object Filter Create a new iSer	ries object filter			
.ibrary:	RSELABXX		••	Browse
Object:	*		•·	Browse
Object type:	*		••	Browse.
Object attribute:	*		•	Browse
	Mor	e Types>>		
	< Back	Next >	Finish	Cancel

3. In the Library field, type RSELABxx.

4. Click **Browse** beside the **Object type** field. The Select Object Type window opens.

*CTLD		
*CRG		
*CRQD		
*DEVD		
*DOC		
*DTAARA		
*DTAQ		
*EDTD		
*EXITRG		
*FCT		
FILE		
*FLR		
*FNTTBL		
FNISRC		
TIR Kaca		
MGCLC		
*IPYD		
*10BD		
*1080		
*1085CD		
*JRN		
KADNID CU		

- 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.

ibrary	RSFLABYY	T .	Browse
Object:	*		Browse
Object type:	*FILE	 	Browse
Object attribute:	*	 ▼	Browse
	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.

DDMF DFU DFUEXC DFUNOTEXC DKTF DSPF	
DFU DFUEXC DFUNOTEXC DKITF DSPF	
DFUEXC DFUNOTEXC DKTF DSPF	
DFUNOTEXC DKTF DSPF	
DKTF DSPF	
DSPF	
DEDE20	
ICEE	
IF	
LF38	
MXDF38	
PF-DTA	
PF38-DTA	
PF-SRC	
PF38-SRC	
PRIF DDTE39	
SAVE	
TAPE	

10. Click Next.

The Name the new filter page opens.

© New				x
Object Filter				
Name the new	w filter			
Filters are sav appear in the	ed for easy re-use. Remote Systems vie	. Specify a unique r ew, and will be exp	ame for this filte andable.	r. This name will
Filter name:	My source files			
Conly creat	te filter in this conne	ection		
Select a profil sharable by th	e to own the new fil ne team. It will be pl	lter. This determine laced in the default	s if it is unique to filter pool for the) you, or at profile.
Owner profile	: msteckh7tp			•
	< Deale			

- 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:

remote systems as Team	-
	*
😥 🐘 RSELAB66.*lib.test	
🕀 🐘 RSESERVER.*lib.prod	
🖅 🐘 RSEWULKAN.*lib.test	
庄 🖍 RSE100.*lib.prod	
🗈 🐘 RSE1000.*lib.prod	
🖅 🐘 RSE10000.*lib.prod	
🕀 🕪 RSE5000.*lib.prod	
🕀 🐘 VARLABXX.*lib.prod	
🕀 🗤 VARLABOO.*lib.prod	
🕀 🐘 VARLAB10.*lib.test	
🕀 🖬 VARLAB11.*lib.test	
庄 航 VARLAB12.*lib.test	
⊕… 🙀 VARLAB13.*lib.test	
🕀 🐘 VARLAB14.*lib.test	
🕀 🐘 VARLAB15.*lib.test	
E 🔐 My source files	
COMPARE.*file.pf-src	
QDDSSRC.*file.pf-src	
± are is commands	
±) ∰a iSeries Jobs	
······L≱ Usnells	
III 5400a	
	Ē

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 QCBLLESRC.
- 2. Right-click member PAYROLLC.
- 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" on page 106.

Exercise 6.4: Creating a user action

Before you begin, you must complete "Exercise 6.3: Creating an object filter" on page 102.

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.

.ERemote System	ns 🗙 Team	C	- 0	CLR1.CLLE	PAYROL	LG.RPGLE	3
	£ 8] (-	⇒ <u>a</u> B \$	-	Line 1		Column	. 1
	rs X Team Pray list QSYS.*lib.prod-s QSYS.*lib.prod-s QSYS.*lib.prod-s QSYS.*lib.prod- QUSRSYS.*lib.prod QUSRSYS.*lib.prod QUSRSYS.*lib.prod QGPL.*lib.prod-us WFLABXX.*lib.prod RSELAB.*lib.prod RSELAB.*lib.prod RSELAB.*lib.prod RSELABMJS1.*lib. RSELABMJS1.*lib. RSELABMJS1.*lib. RSELABN.*lib.prod RSELABX.*lib.prod RSELABX.*lib.prod RSELABN.*lib.tes RSELAB01.*lib.tes RSELAB03.*lib.tes RSELAB03.*lib.tes RSELAB03.*lib.tes RSELAB03.*lib.tes RSELAB04.*lib.tes RSELAB03.*lib.tes RSELAB04.*lib.tes RSELAB05.*lib.tes RSELAB06.*lib.tes RSE	→	Entry ary Listiany	CLRI.CLLE	PAYROL F** F* F* F* F* F* F* F* F* F	LG.RPGLE & Column .1 PROGH DESCF DESCF SUDIO 50 - 60 - 90 - KC - KD - KC - KC - KC - KC - KC - KC - KC - KC	AM AM AT No Ge: Pr En Re Re Re Re La: AT SR AT SR AT SR AT
Properties 23	RSELAB07.*lib.te: RSELAB08.*lib.te: RSELAB09.*lib.te: Remote Scratch	++ Move Within Lib Show in Table Find String Change Save(B) Restore	rary L	ist 3 aa	iSeries	Series 🎇	3
Attribute		User Actions		<u> </u>	Work With U	ser Actions	

- 2. Expand the Library list filter if not already expanded.
- 3. Right-click RSELABXX.
- 4. 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			1
New	Action name:	Copy source file	1
Member action	Comment:	Copy source file	with data
	Command:	Normal command	
	CRTDUPOBI		×
			*
	4		Ð
	Insert variable	Edit	Browse
		1	
	Prompt		
	Prompt first	Refresh aft	ter 🔽 Show actior e
	Prompt	Refresh afi n only Invoke onc r which this action wi	ter 🔽 Show action e II appear
	Prompt	Refresh aft	ter 🔽 Show action e II appear Selected Types
	Prompt first Single selection Resource types fo Defined Types	Refresh aft	ter 🔽 Show action e II appear Selected Types ALL
	Prompt first Prompt first Single selection Resource types fo Defined Types ALL CMD FILE FILE FILE_DATA	Refresh afi n only Invoke onc which this action wi Add> <remove(g)< td=""><td>ter 🔽 Show action e II appear Selected Types ALL</td></remove(g)<>	ter 🔽 Show action e II appear Selected Types ALL
	Prompt	Refresh afi only Invoke onc which this action wi Add> <u>Add></u> <u>CRemove(G)</u> Edit(H)	ter 🔽 Show action e II appear Selected Types ALL
	Prompt	Refresh aft only Invoke onc r which this action with Add> <remove(g) Edit(H)</remove(g) 	ter 🔽 Show action e II appear Selected Types
	Prompt	Refresh aft only Invoke onc which this action wi Add> (Remove(G) Edit(H)	ter 🔽 Show action e II appear Selected Types ALL
	Prompt	Refresh afi n only Invoke onc r which this action wi Add> (Remove(G) Edit(H)	ter 🔽 Show action e II appear Selected Types ALL Revert(J)

- 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	Object (CRTDUPOBJ)	×
From object:	8N	▼ Name, generic*
From library:	8L	▼ Name
Object type:	&T	▼ Add
To library:	*FROMLIB	▼ Name
New object:	QJUNKSRC	▼ Name
Duplicate data:		•
Advanced	Al Parameters	☐ Keywords
CRTDUPOBJ OBJ(&N) F DATA(*YES)	ROMLIB(&L) OBJTYPE(&T) NEV	
	OK Restore def	aults <u>C</u> ancel

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.

New	Action name:	Copy source f	ile
Member action	Comment:	Copy source f	ile with data
	Command:	Normal comma	nd j
	CRTDUPOBJ OBJ(&N)	FROMLIB(&L)	OBJTYPE(&T) NEW
	•		. D
	Insert variable	Edit	Browse
	Prompt		
	— — — — — — — — — —	The Definet	a fear of the second
	Prompt first	Refresh	arcer Show actio
	Single selection or	ly III Invoke c	nce
	Single selection or Resource types for wi	ly I Invoke o	nce will appear
	Resource types for wi	ly Invoke o	will appear Selected Types
	Prompt first Single selection or Resource types for wi Defined Types FILE_DATA	ich this action	will appear Selected Types
	Prompt first Single selection or Resource types for wi Defined Types FILE_DATA	Add>	will appear Selected Types
	Prompt first Single selection or Resource types for wi Defined Types FILE_DATA	Add>	will appear Selected Types
	Prompt first Single selection or Resource types for wi Defined Types FILE_DATA • FILE_DATA • FILE_DATA • FILE_MBRS FILE_SRC ILE LIB •	Add>	will appear Selected Types FILE_SRC
	Prompt first Single selection or Resource types for wi Defined Types FILE_DATA FILE_DSPF FILE_MBRS FILE_MBRS FILE_SRC: *FILE:PF*	Add>	will appear Selected Types FILE_SRC

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.

Image: Remote Systems × Team Image: Remote Systems × Team Image: Remote Systems × Image: Remote Systems × Image: Remote S	Paste Move Delete Show in Table Cache File Descriptions Find String Change(8) Save Restore	□ □ ■ □
	User Actions Add To iSeries Project Make Available Offline	Copy source file
Wy source files COMPARE *file.of-src COMPARE *file.of-src	Properties(K)	

- 2. Expand the filter My source files, if it is not already expanded.
- 3. Right-click the QCBLLESRC 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.

By iSeries Commands Log 🗙 Remote Search	⊿
s400a	
NEWOBJ(QJUNKSRC) DATA(*YES) Object QJUNKSRC in RSELABXX type *FILE created.	<u> </u>
1 objects duplicated. Cause : All the objects with supported object the Create Duplicate Object (CRTDUPOBJ) command See low level messages to determine if any objects w	t types specified on were duplicated. vere not 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 98.

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 106.

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 command 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.

ie LE LE LE	PF-SRC PF-SRC PF-SRC PF-SRC	Text		OK OK	-
le Le	PF-SRC PF-SRC PF-SRC			ОК	
le Le	PF-SRC PF-SRC			OK	
LE	PF-SRC				
3.2				OK	
LE	PF-SRC			OK ,	•
-0	ADDLIBLE		Pro	ompt	
	-0	- (?ADDLIBLE	7ADDLIBLE	ADDLIBLE Pro Details	Prompt Run Details Show Log

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.

Library:	RSELAB00	Name
List position:	*FIRST	1
Reference library:		Name
🗖 Advanced	All Parameters	vords
Advanced	All Parameters 🗖 Keyw	vords
Advanced	T All Parameters T Keyv	vords

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.

	1760	Attribute	Text	Status
OMPARE	*FILE	PF-SRC		OK
CBLLESRC	*FILE	PF-SRC		OK
CLSRC	*FILE	PF-SRC		OK
CMDSRC	*FILE	PF-SRC		OK
CMDSRC	*FILE	PF-SRC		ОК

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 97. 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 115.

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 116
- "Exercise 7.2: Viewing the DDS tree" on page 117
- "Exercise 7.3: Selecting the DDS object" on page 118
- "Exercise 7.4: Designing the DDS screen" on page 119
- "Exercise 7.5: Creating groups from existing records" on page 120
- "Exercise 7.6: Creating new screens" on page 122
- "Exercise 7.7: Adding fields to the subfile record" on page 124
- "Exercise 7.8: Switching between multiple records" on page 127
- "Exercise 7.9: Adding field error handling" on page 129
- "Exercise 7.10: Accessing field properties" on page 131

- "Exercise 7.11: Adding new keywords" on page 133
- "Exercise 7.12: Verifying the source changes" on page 135
- "Exercise 7.13: Switching between designing and editing the screen" on page 137
- "Exercise 7.14: Compiling your source changes and closing the Designer" on page 138

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 117.

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 116.

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 117.

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	2
🔞 "DATE	2,70	6	Date const			MM/DD/YY	
A 'Mainteection'	3, 30	21	Text const			Maintenance Selection	:
*TIME	3,70	6	Time const			HH:MM:SS	
A 'Enterintain'	6,14	54	Text const			Enter an X beside the	9
EMPAPL	9,25	1	Alphanumeric	A - Alphanum	Both	В	
A 'Employenance'	9, 28	27	Text const			Employee Master Ma	i T
🗖 PRJAPL	10, 25	1	Alphanumeric	A - Alphanum	Both	В	
A 'Projecenance'	10, 28	26	Text const			Project Master Mainte	
RSNAPL	11, 25	1	Alphanumeric	A - Alphanum	Both	B	ŕ

 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.

Details	MAIN_MENU Source Listing	
	Record SELECT keywords	
Keywa	ord Conditioni	
∽~ BLI	NK	
∽~AU	ARM 60	
	N	
	4	
Selecte	ed DDS Create keywords Web Settings Comments E	rror list Ì
	1	
<u>an</u>		
	A* 89703702 12:07:25 DOUG A* THE SELECT FORMAT ALLOWS SELECT	REL-
	A* MASTER FILE THE OPERATOR WANTS	TO MAIN
	A R SELECT	
	À co	BLJ
	Δ	2 5'PF
	A	I I
	A	2 30'Ti -1

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 118.

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 119.

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
□ 🕞+1+2+3+4+5+6++6++8
Group SCREEN1
Available Selected
Selected DDS Web Settings Comment: BSNSEL
A
A A
A 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 \rightarrow 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 pro	perties		? ×
Basics Styles	1		
Group	EMPL S	FLCT	
Av	ailable		Selected
SELECT		EMPSEL	
PRJSEL			
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 120.

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		d e stad I a le u
	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_SE	LCT EMPL_LIST Source Lis	sting		
RECORD1 -				
lgnore all indicators	• 435 8		ZEMAREU]
□ □ 1	+3	+4+	5+б+	8
<u></u> 11			3	

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 122.

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
	BEBEEBEBEE BEBEEBEBEE
A D 10	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
변생 11	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
A 12	BBBBBBBBBB
5 0 13	<u>8888888888888888888888888888888888888</u>

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.

EMPLSTSFL		(OPCODE)	🖷 (3F	IOB	
1/1 Ignore all indicators	-	a#6®		* 11 *	БяA

- 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
Π	000000023000000000000000000000000000000
Ш	000000000000000000000000000000000000000
ΠŤ	000000000000000000000000000000000000000
111	000000000000000000000000000000000000000

EMPLSTSFL	▼‱∢⋗	'+0(30)'	▼ \$\$\frac{\frac{\psi + 2}{\psi 12}}{\psi 12}\$	6	 IOB		de
Ignore all indicators	•	436	20	0	 淡言業日	5 JA	20

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
1/ Ignore all indicators 🔽 🖉 🖉 🖉 🖉 📄 🔜 第二日日 日日日
□

- 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!

Details MAIN_M	IENU E	EMPL_SELCT	EMPL UST	Source	isting			
EMPLS	STSFL	▼ 5 ⁰⁺² / _{λ45} <	SALA		8	EI	OBE	I• 2
1/6 Ignore	all indica	tors	- A-	5.5.BY	- 2	EM	1 #6	A
	+	1+-	2	+3	+	- 4 +	<mark>-</mark> -5	-+6-
2								
<u>A</u> 🗳 8	III	00000000	000000000	00000000	000000	\$666,	666.66	
9	III	00000000	000000000	00000000	000000	\$666,	666.66	
A 8 10	III	000000000	000000000	000000000	000000	\$666,	666.66	
	111	00000000		00000000		Şbbb,	666.66	
	<u>111</u> TTT	00000000		000000000		\$000,	000.00 ccc cc	
	<u></u>	00000000	000000000	00000000	000000	ŞUUU,	000.00	
Calacted DDG		: le		(
Delected DUS	web Set	ings Commer	nts Error list					
<u> </u>	-	CITION		011 20				
A		SALARY		84 20	8 41EDI	CDE(I S		

- 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.

EMF 1/0 Igno	PLSTSFL 💽 🛍	SALARY -	≝ 8 ∷ IOB Y▼ 2 ∷ ※ ⊫ ≋	
<u>111</u>	000000000000	000000000C <mark>(</mark> 0000000	\$666,666.68	
			\$666,666.66 \$666,666.66 \$666,666.66	
$\frac{111}{111}$	000000000000	000000000000000000000000000000000000000	\$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.

<u>^</u>

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 124.

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	\$565,656,55
000000000000000000000000000000000000000	\$566.666.56
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_SELC	T EMPL_LIST Source Listing	
Ignore al indicators		ABE
	-+2 <mark>-</mark> +3+4+5	+6+7+8
2 2		
Page 4 CPositi	ion to:)	
as:		
		C555 555 55
		\$555,555,55
		\$666,666,66
		\$666,666.66
12 11		\$666,666.66 -
300 🚍 13	000000000000C X 000000000000000	\$666,666.66
9 14 111		\$666,666.66
15	000000000000000000000000000000000000000	\$666,666.66
		\$555,555.55
150		

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 is button in the palette, or you could reference the original database field using either the Create database reference field button or the L 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_MEN	NU EMP_SELCT	MPL_LIST Source Listing	
	TL 🖌 🏭 🛛		
1/0 Ignore all	indicators		AB FU
	+1+-	2 <mark>-</mark> -+3+4+5+	678
			<u> </u>
	Position		
5 5			
			ACCE FEE FE
			\$656,665,66
10			\$666,666.66
11			\$666,666.66
12			\$666,666.66
300 🚍 13			Ş666,666.66
9 - 14		000000000000000000000000000000000000000	\$666,665.66
15			\$555,555.65
0 🚍 16			\$666,666.65
17			-

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 127.

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 M 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				x	
		Indicators				
01 111 02 121 03 131 04 141 05 151 06 161 07 171 08 181 08 181	20 7 30 7 21 7 31 7 22 3 3 3 23 7 33 7 24 7 34 7 25 3 35 25 3 36 36 3 36 3 37 3 38 7 38 7 38 7 38 7 38 7 38 7 38	40 50 41 51 42 52 43 53 44 54 45 55 46 56 47 57 48 58 49 59	60 F 61 F 62 F 63 F 64 F 66 F 66 F 66 F 68 F 68 F	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 99	
Ch	Checked indicators are: 💿 On 🔘 Off					
Settings nam	Not Found	1		<u></u> re	ate	
Ignore all indic	ators			<u>B</u> er	ame	
All Indicators a	ie orr			De	lete	
		<u>ОК А</u> р	ply Ca	ancel H	lelp	

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 💌 🏭
1/0 Not Found	•	A#5PBA-

- 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>444</u>	30 🕂	IOB	ra Beem
•	A#5PB	A 🕶	0 📑	※ 王 米	2eu

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

Sele	cted DDS	∂ Web Settings Co	mments Ei	ror list			
A A A A A	60 90 60 60	POS_TO	R	В	4	23 COLOR(RED) REFFID(RCEMP/ENAME *LIBL/ DSPATR(PR) DSPATR(RI) DSPATR(PC)	EÌ
1							

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**.

<u></u>	EMPLSTCT	L 🔻	◄ ► POS_TO	IOB 💷 🖁
	M All indicator	s are off	■ ∠±Epsa • • =	∃ ≫ E ₩ E M Z &
Γ		12	 +	+6+1+
1		Position to:		
	5			
			000000000000000000000000000000000000000	\$666,666.66
2	L 10			\$656,666.66

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 129.

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	

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.

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

Maintenance Sélect	ion	HH: MM: SS
asics Attributes Co	lor Editing Indicator	s]
Sample MM/DD/YYYY	Source of date	🖌 🗹
Position Size Row Colu 24×80 2 70 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		? X
Basics Attributes Color	Indicators Styles	
Field EMPAPL Usage	Type	
Length Total Dec Continued (CNTFLD)	A - Alphanumeric Position Size Row Column 24×80 9 25 25 27×132 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 131.

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

Insert Subset	All			-
S ALIAS	CHECK(M11)	See CHRID	See FLDCSRPRG	Service WRDWRAP
S BLANKS	Section CHECK(M10F)	COLOR>>>	IGCALTTYP	Switch Contract Contr
🗫 CHANGE	Section CHECK(M11F)	Second Comp	∞ INDTXT>>>	
🗫 CHECK(AB)	Section CHECK(RB)	See DFT	🗫 MSGID>>>	
Section CHECK(ER)	Section CHECK(RL)	See DFTVAL	Service Noccold	
CHECK(FE)	CHECK(RZ)	DSPATR>>>	∞ OVRATR	
🗫 CHECK(LC)	Service CHECK(VN)	S- DUP	Second Se	
🗫 CHECK(ME)	Service CHECK(VNE)	Service ENTFLDATE	Service PUTRETAIN	
🗫 CHECK(MF)	See CHGINPDFT	ERRMSG>>>	🗫 RANGE	
CHECK(M10)	Sector CHKMSGID	Contemporary Conte	Serve VALUES	

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

Basics Attributes Color	Indicators Styles	
EMPAPL Usage Length Total Dec Continued (CNTFLD)	A - Alphanumeric Position Size Row Column 24×80 9 25 27×132 2 2	5.
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 infor Remote System Explorer 	DDS Reference
[2013] Jane K. Malada H. Martin K. Sanakara and K. Katalan and K. Katalan.	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	
Sector 2010 - 2010 - 2010	3	
Swindtxt	01 '?'	
Swind txt	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 133.

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 135.

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.

litor - Find and Replace	
Find INDIXI Replace with	•
• Replace this occurrence	
C Replace, then find <u>n</u> ext	Pattern match
Replace <u>all occurrences</u>	I ₩rap
Find Replace Options	s 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 137.

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 115. 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 141.

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 142

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 141.

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 141. You should now understand:

- What components make up Development Studio and Development Studio Client editions
- 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 147.

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	3е	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	6с	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