Transition Guide: SEU to Remote System LPEX Editor

Transition Guide: SEU to Remote System LPEX Editor

Contents

Chapter 1. About this document

This document is intended for application programmers or analysts working in an IBM iSeries (AS/400) environment, who are interested in using the better toolset provided in WebSphere Development Studio Client for iSeries (Development Studio Client). It explains how to use the Remote Systems LPEX Editor, one of the tools provided in Development Studio Client, and compares this tool to the more traditional Source Entry Utility (SEU).

The Remote System Explorer is a flexible, enhanced workstation version of Programming Development Manager (PDM). It provides easy ways to perform everyday task when you use it with iSeries projects:

- Create and manage development projects on your iSeries system from your Windows-based workstation
- Connect to a remote iSeries server
- View iSeries libraries, files and members
- Launch the server compilers, the workstation editor, a program verifier, and various debuggers

This document introduces you to the similarities between the Remote Systems LPEX Editor and SEU. It also explains enhancements in the Remote Systems LPEX Editor that are not available in SEU, and describes how to accomplish the most commonly performed editing tasks on your iSeries system using both methods.

The comparison between the two tools also helps you with the first step in the iSeries Developer Roadmap: a learning path for iSeries developers who want to adopt new technologies and improve their skills. For more information about the iSeries Developer Roadmap, see http://www.ibm.com/servers/eserver/iseries/roadmap/index.html.

To use this document effectively, you need to be familiar with basic Windows tasks, such as file navigation and cursor manipulation.

Chapter 2. WebSphere Development Studio Client for iSeries overview

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 designed to ease the entire life cycle of developing and maintaining iSeries applications. Whether you work on traditional green-screen applications, or are responsible for making existing iSeries programs accessible as Web applications, your productivity is greatly increased as you work on the following tasks.

You can create and manage development projects on your iSeries server from your Windows-based workstation with the Remote System Explorer, one of the tools provided in Development Studio Client. With Remote System Explorer, you can view iSeries libraries, files, and members. You can also launch the server compilers, the workstation editor, and various debuggers. Your program editing tasks are simplified with the Remote Systems LPEX editor. The editor can access source files on your workstation or your iSeries server.

With the integrated iSeries debugger you can debug your program running on the iSeries server from a graphical user interface on your workstation. You can also set breakpoints directly in your source before running the debugger. 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, even if they are written in different languages, from a single debugger window.

For more information about Development Studio Client for iSeries, please visit http://www.ibm.com/software/awdtools/iseries/

Remote System Explorer overview

The Remote System Explorer (RSE), a perspective in Development Studio Client, provides the framework, user interface, editing capability, and performable actions on iSeries objects, commands, and jobs. It supports views, editors, and tool extensions, to help you develop and maintain iSeries applications. The Remote System Explorer is the iSeries programming environment you use to complete these tasks. The RSE guides you through various connections to your remote systems, give you access to files and resources, and help you monitor jobs and run commands.

When you first open the Remote System Explorer, you are not connected to any system except your local workstation. To connect to a remote iSeries, UNIX, Linux, or Windows server, you need to define a connection. A connection is a TCP/IP network connection to your iSeries server, that enables you to access, edit, run, compile, and debug all items on the server. When you define a connection, you specify the name or IP address of the remote system and you also give the connection itself a unique name that acts as a label in your workspace so that you can easily connect and disconnect. These actions are performed in the Remote System Explorer, the workbench perspective for iSeries development tools.

LPEX Editor overview

The Remote System Explorer contains the Remote Systems LPEX editor, so that you can edit source code directly in the workbench. The interface provides right-click actions so that you can compile, run, and debug your programs.

The Remote Systems LPEX editor also supports:

- Language parsing using automatic indenting, color, and text effects to emphasize different parts of your source program, such as programming language keywords, logic structures, and comment lines
- Many rich editing features for RPG, COBOL, CL, and DDS such as syntax checking and program verification
- Concurrent multiple edit sessions
- The Outline view and Content assist for ILE COBOL and ILE RPG source
- Local verification and error feedback
- Command shells for running local or server commands with logged output

The LPEX Editor can be used to create and edit many kinds of files, including program source files, and documentation.

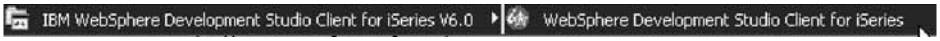
In addition to basic editing functions, the LPEX Editor offers the following features:

- Location marking facilities to define bookmarks so you can move quickly from one location to another within your document.
- Elaborate search facilities to specify the precise scope of the search. In addition to locating specific character strings, you can use other search facilities to find specific locations within your document.
- Keystroke recording facilities can record a set of keystrokes to replay later.

Starting Development Studio Client

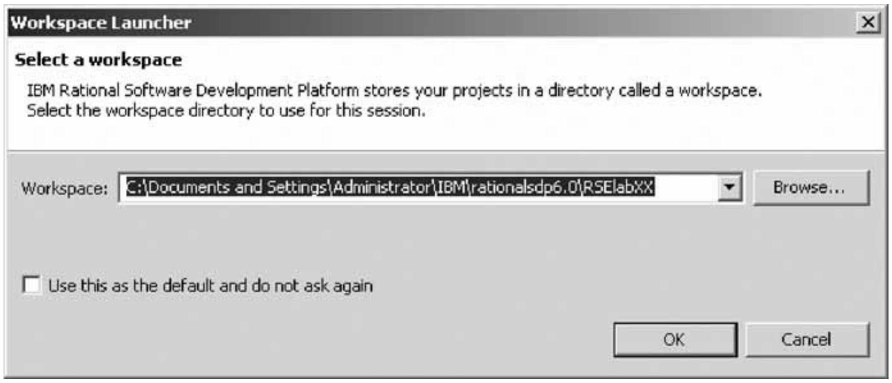
To start Development Studio Client:

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

   Note: If the Advanced Edition of the product is installed on your workstation you will see the words ″Development Studio Client Advanced Edition for iSeries″ in the Start menu.
3. If a window opens, specify the name of the workspace where you want to keep your projects and other resources such as folders, subfolders, and files that you develop with the Development Studio Client workbench.

   (Optional) Type a unique directory name, for example, `RSElabXX`.



Workspace Launcher

Select a workspace

IBM Rational Software Development Platform stores your projects in a directory called a workspace.
Select the workspace directory to use for this session.

Workspace: C:\Documents and Settings\Administrator\IBM\rationalsdp6.0\RSElabXX

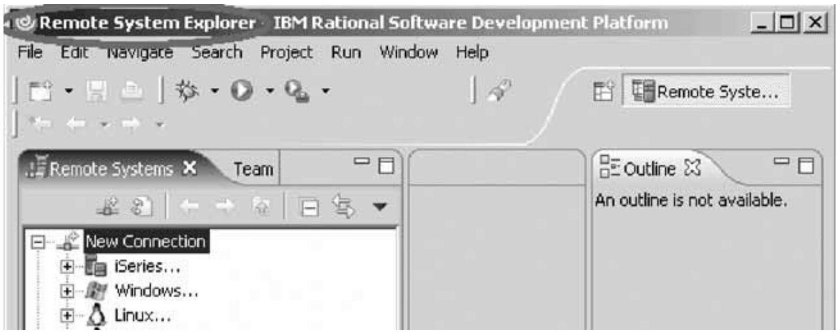
☐ Use this as the default and do not ask again

4. Click **OK** to start the workbench.
5. Click the X in the **Welcome** tab to close the Welcome view. You should see the Remote System Explorer perspective.

The *workbench* refers to the desktop development environment. The workbench aims to achieve seamless tool integration and controlled openness by providing a common interface for creating, managing, and navigating workbench resources. Each workbench window contains one or more *perspectives*. 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 commonly use while editing Java source files, while the Debug perspective contains views that you use while debugging programs. Perspectives contain *views* and *editors* and control what appears in certain menus and tool bars.

Opening the Remote System Explorer perspective

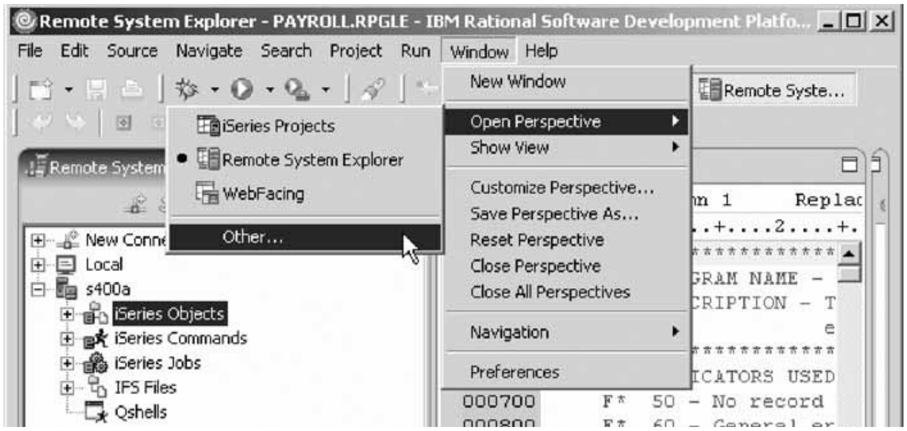
Check for the name of the perspective.



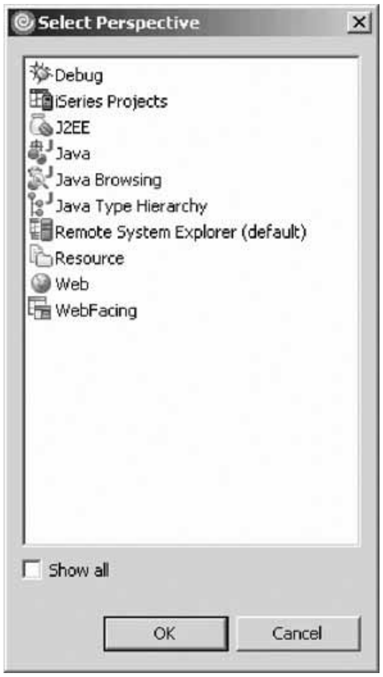
Remote System Explorer - IBM Rational Software Development Platform

An outline is not available.

1. If you see a different perspective from the **Remote System Explorer**, or no perspective at all, select **Window > Open Perspective > Other** from the workbench menu bar.
   -



2. Select **Remote System Explorer**.



For the exercises in this document, you will work in the Remote System Explorer perspective, which is designed for iSeries programmers. In this perspective, you can create new connections to an iSeries server, work with iSeries files, commands, jobs, and integrated file system files.

This perspective is active when you start Development Studio Client, loaded with a new workspace. If you have used a workspace before, the workbench opens with the perspective that you last used. You use this perspective to launch the iSeries programmer tools and views in the workbench.

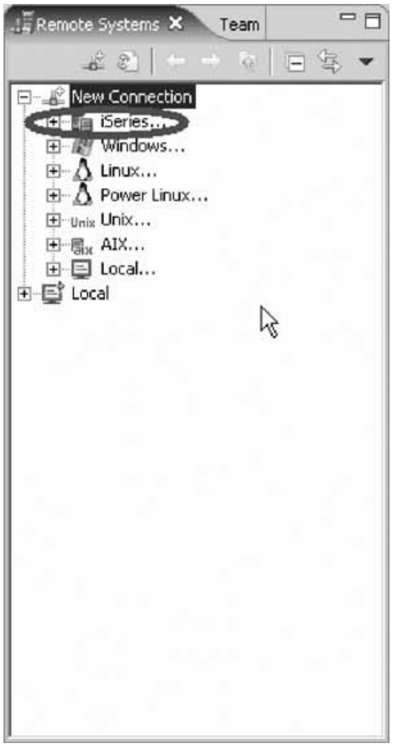
Configuring a connection and connecting to an iSeries system

When you first open the Remote System Explorer, you are not connected to any system except your local hard drive on your workstation. Before you can connect to a remote iSeries server, 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 the connection a unique name that acts as a label in the workbench, 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 server.

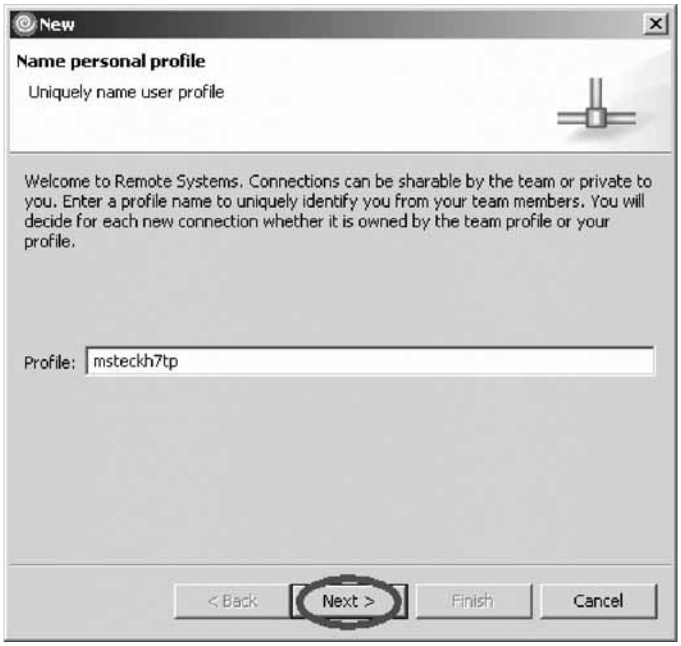
The first time you detect to an iSeries server, you need to specify a profile. All connections, filters, and filter pools belong to profiles. We will discuss profiles after you create your first connection.

To configure a connection:

1. In the Remote Systems view, **New Connection** is automatically expanded to show the types of remote systems you can connect to through the Remote System Explorer. Click the plus sign (+) beside **iSeries** to configure a connection to an iSeries server.



2. In Name personal profile page, leave the default value in the **Profile** field and click **Next**.



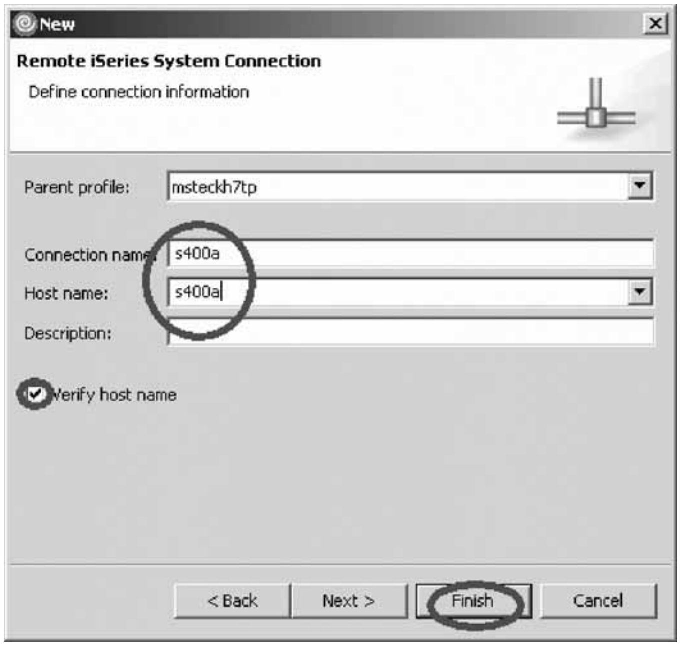
Welcome to Remote Systems. Connections can be sharable by the team or private to you. Enter a profile name to uniquely identify you from your team members. You will decide for each new connection whether it is owned by the team profile or your profile.

Profile: msteckh7tp

3. Remote iSeries System Connection opens:

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

   In the **Host name** field, type the IP address or the name of your server system, for example s400a.


New
Remote iSeries System Connection
Define connection information

Parent profile: msteckh7tp

Connection name: s400a

Host name: s400a

Description:

☑ Verify host name

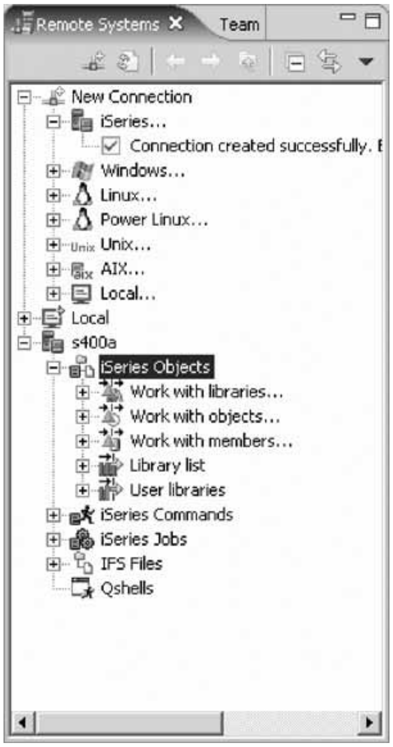
< Back
Next >
Finish
Cancel

4. The Connection name is automatically filled with the server name for your convenience. This name is listed in your Remote Systems view and must be unique to the profile, so you probably want to accept the default value.
5. Leave the default value for the **Parent profile** field. For the purposes of this exercise, you can accept the default parent profile to keep your files private. For future development, you can choose to share your files and resources by selecting the Team profile.

6. Leave the **Verify host name** check box selected.
7. Click **Finish** to define your system.

After you configure a connection to an iSeries server, you can easily connect and expand your new connection to reveal your subsystems. Subsystems are pre-defined filter groupings of the various types of remote resources that can be explored in the remote system. Your iSeries connection gives you the use of four subsystems:

- **iSeries Objects** is a PDM-like group, enabling access to libraries, objects, and members
- **iSeries Commands** contains predefined commands you can run against remote objects. You can also create command sets, each of which contain one or more frequently 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** displays various jobs, subset by job attributes, and enables you to perform a number of operations on those jobs.
- **IFS Files** helps you explore folders and files in the Integrated File System of the remote iSeries server.



To connect to an iSeries system:

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

- **Work with members** (similar to WRKMBRPDM)

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

- **Library list** (to simulate PDMs STRPDM option 12, where you can start with the pre-defined Library list filter, that when expanded lists all libraries in your library list.)
- **User libraries** (enables you to work with all user libraries on your iSeries server.)

Chapter 3. Working with members using Remote System Explorer

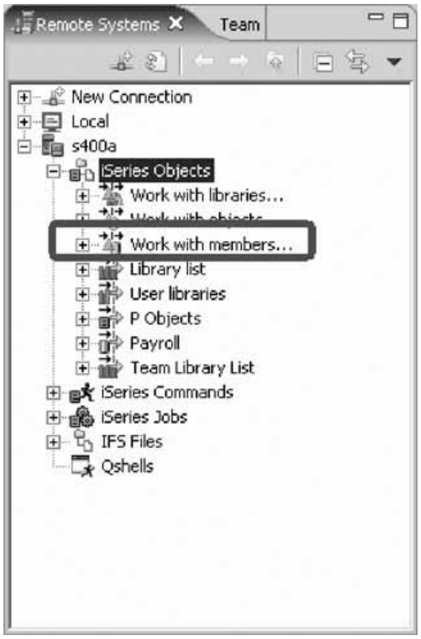
Viewing a list of members

To view a list of members in SEU using the STRSEU command:

- Specify a source file for the SRCFILE parameter but do not specify a member for the SRCMBR parameter.
- Specify the *SELECT value for the SRCMBR parameter.
- Enter the STRSEU command without any parameters if the Work with Members Using SEU display was chosen in the previous session.

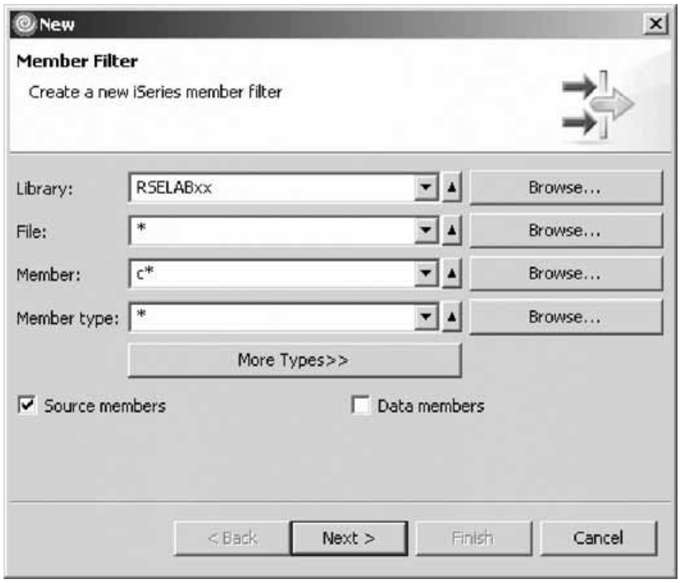
In the Remote System Explorer, member filters list a set of source and data members from your iSeries server in the Remote System view. With the filter, you can use this list again later.

1. In the Remote Systems view, expand your connection and then expand **iSeries Objects**.
2. Expand **Work with members**. (You can also right-click **iSeries Objects** and select **New > Member filter**.)



Note: Expanding **Work with members** corresponds to the WRKMBRPDM command in PDM.

3. Browse, specify, or select from the **Library** drop-down list the library that contains your source members.



New

Member Filter

Create a new iSeries member filter

Library: RSELABxx

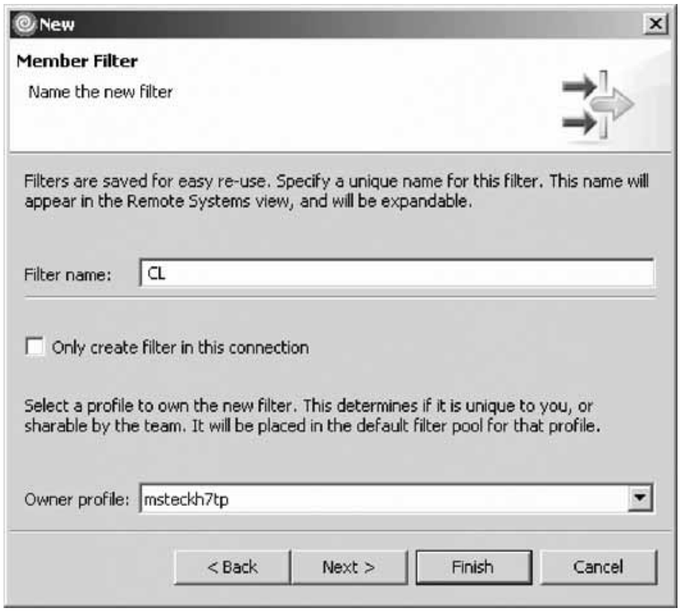
File: *

Member: c*

Member type: *

☑ Source members ☐ Data members

4. Browse, specify, or select from the **File** drop-down list the file within the library that contains the appropriate source members.
5. Enter a generic or specific member name in the **Member** field. The default is * (asterisk), and you can use up to 10 characters. You can, for example, enter c* to list all members that start with c.
6. (Optional) To select a specific member type, enter the type in the **Member type** field, or click **More Types** to select from all known object types.
7. Use the check boxes to select whether you want to list source members, data members, or both. Select at least one check box, and then click **Next**.
8. Enter a name for your filter.



New

Member Filter

Name the new filter

Filters are saved for easy re-use. Specify a unique name for this filter. This name will appear in the Remote Systems view, and will be expandable.

Filter name: CL

☐ Only create filter in this connection

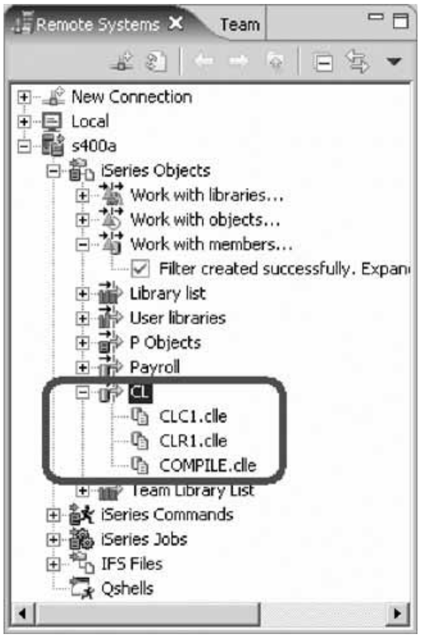
Select a profile to own the new filter. This determines if it is unique to you, or sharable by the team. It will be placed in the default filter pool for that profile.

Owner profile: msteckh7tp

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

9. (Optional) Check the **Only create filter in this connection** check box if you want the filter to display only in your current connection. Otherwise, it displays in all your connections.
10. Select the profile you want to use for this filter. You can use a personal profile to keep your work private or the Team profile to share with others. In this instance, use your default private profile.
11. Click **Next** to see more text information about the filter, or click **Finish**.

Your new member filter now displays in the Remote Systems view under **iSeries Objects**.



Changing the List of Members

You can change the list of members that appears on the Work with Members Using SEU screen so that only members that meet certain criteria are included. You can change the list so it shows only members that have the same:

- Partial name
- Member type
- Date of creation or last change
- Text or partial text

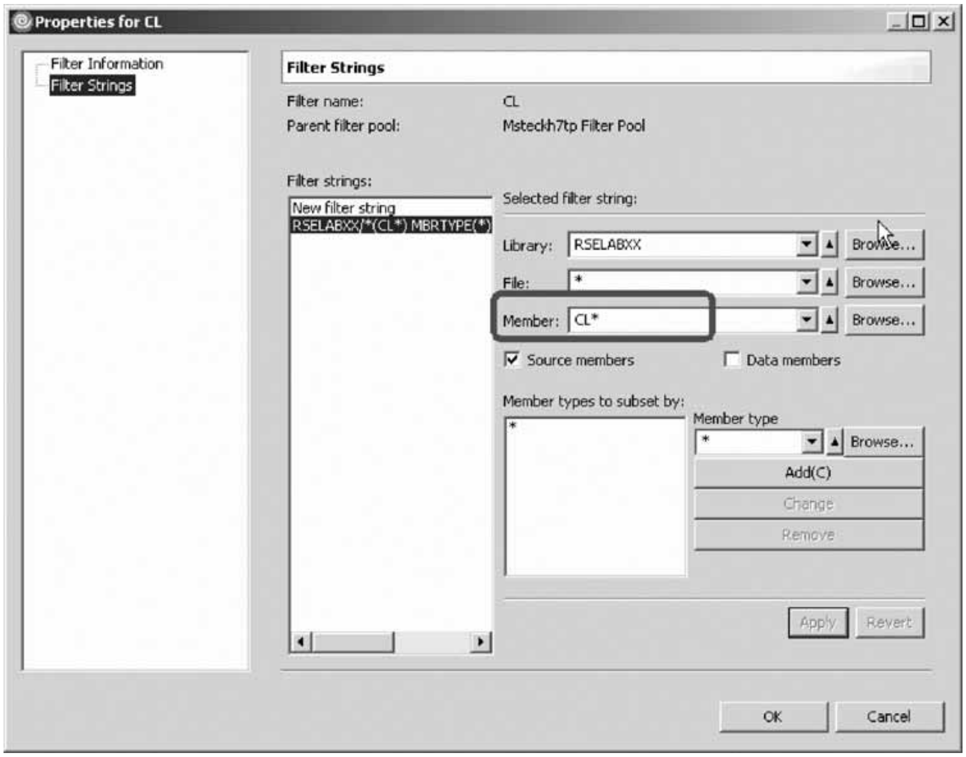
To change the list of members using SEU:

1. Press F17 (Subset). The Subset Member List display appears.
2. Specify the criteria for the members that you want to be in the list and press Enter. The changed list of members is displayed.

In Development Studio Client, you can change the properties for any filter by opening the filter’s properties page. You can also add new filter strings to a filter from this page. For example, if you have a filter that lists all RPGLE source members, you can add an additional filter string that will display all CBLLE source members as well.

To change a filter:

1. In the Remote Systems view, expand your connection name, and then expand **iSeries Objects** to view the list of your filters.
2. Right-click the filter you want to change and select **Properties**.
3. In the Properties window, click **Filter Strings** to change the filter's specifications (what the filter lists from the iSeries server). For example, change the value in the **Member** field. Then click **Apply**.



Filter Strings

Filter name: CL

Parent filter pool: Msteckh7tp Filter Pool

Filter strings:
New filter string
RSELABXX/*(CL*) MBRTYPE(*)

Selected filter string:

Library: RSELABXX

File: *

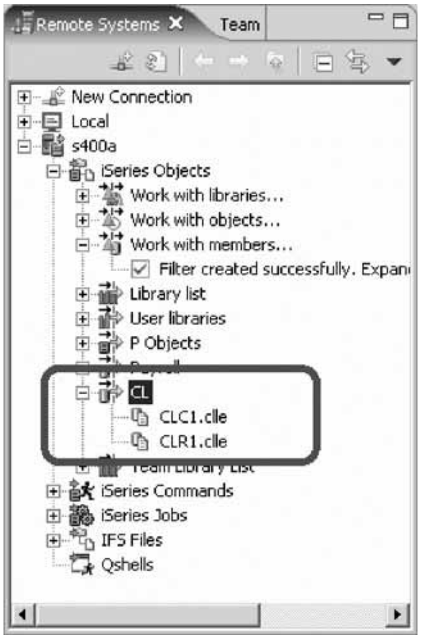
Member: CL*

☑ Source members ☐ Data members

Member types to subset by: *

Member type: *

4. If you want to add extra filter strings for your filter, click **New filter string** and complete the dialog. Then click **Apply**.
5. Click **OK**.



- New Connection
- Local
- s400a
  - iSeries Objects
    - Work with libraries...
    - Work with objects...
    - Work with members...
      - Filter created successfully. Expan
    - Library list
    - User libraries
    - P Objects
    - Payroll
    - CL
      - CLC1.clle
      - CLR1.clle
    - Team Library List
  - iSeries Commands
  - iSeries Jobs
  - IFS Files
  - Qshells

Creating a Member

To create a member in SEU:

- From the command line, or by using command prompting, specify the name of the member to create for the SRCMBR parameter of the STRSEU command. If you use command prompting, type a valid source type in the Source type prompt. If you do not, the default is used.
- On the Work with Members Using SEU screen, type the name of the member to create in the New member prompt, type the source type in the Type for new member prompt, and press Enter.

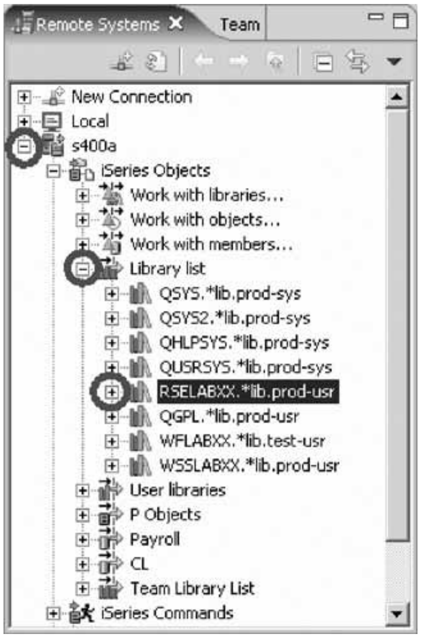
You can access the Work with Members Using SEU display by specifying *SELECT for the SRCMBR parameter of the STRSEU command.

On the Edit screen, you can start entering source.

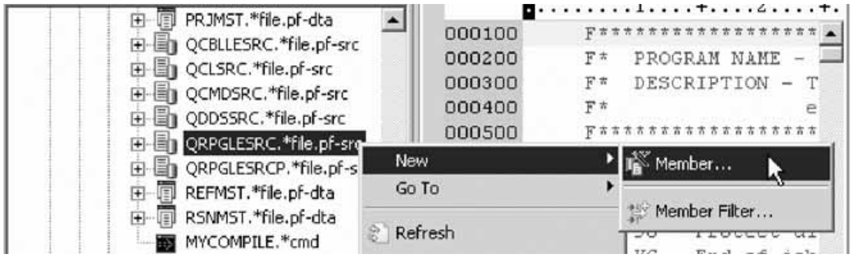
You can create a new member in the Remote System Explorer using either the iSeries Objects subsystem or the iSeries Commands subsystem. The advantage of the iSeries Objects subsystem is that you can navigate through your libraries and files to the location where you want to create the member, giving the member location a context. The advantage of the iSeries Commands subsystem is that you can specify more parameters, (useful for more advanced users).

Method 1: To create a member from the iSeries Objects subsystem:

1. In the Remote Systems view, expand your connection, and expand **iSeries Objects > Library list**.
2. Select and expand the library and source physical file in which you want to create a member.



3. Right-click the source physical file and select **New > Member**.



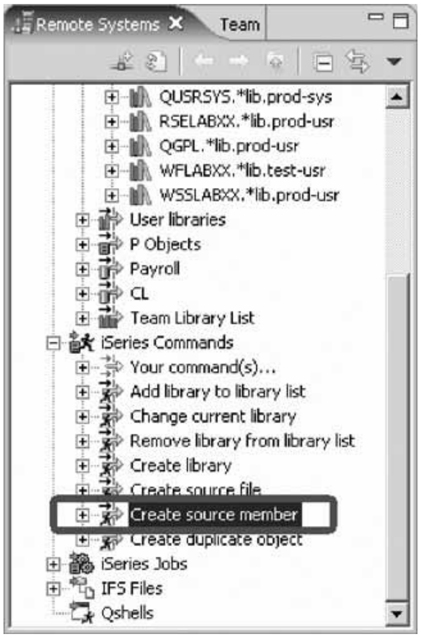
```
000100     F******************
000200     F*  PROGRAM NAME -
000300     F*  DESCRIPTION - T
000400     F*                e
000500     F******************
```

4. Enter the member type or select from the available items in the list.
5. (Optional) Add a description in the **Text** field.
6. Click **Finish**.

Your new member now displays under your selected source physical file.

Method 2: To create a member from the iSeries Commands subsystem:

1. In the Remote Systems view, expand your connection, and expand **iSeries Commands > Create source member**.



2. Specify or select parameters for the member. Click **Advanced** for extra parameters.
3. Click **OK**.

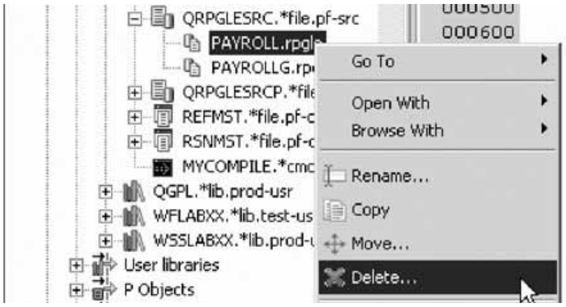
Deleting a Member

To delete a member in SEU:

1. Select option 4 (Delete) on the Work with Members Using SEU screen.
2. The Confirm Delete of Members screen opens with a list of all the members that you marked to be deleted. Check the list carefully to ensure that you are deleting the correct members.

In Development Studio Client, to delete a member:

1. In the Remote Systems view, select the member file to delete.
2. Right-click on the member file.



3. Select **Delete** from the pop-up menu.

Chapter 4. Editing members using the LPEX Editor

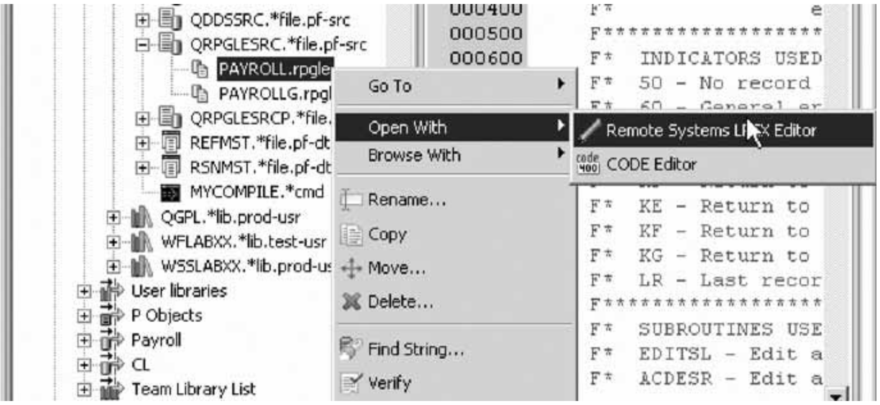
Editing a Member

To edit a member in SEU:

- Specify 2 (Edit) for the OPTION parameter of the STRSEU command.
- Specify option 2 (Edit) in the Opt prompt on the Work with Members Using SEU display.
- Specify option 2 (Edit) on the Work with Members Using PDM display.

To open source in the Remote Systems LPEX editor:

1. Expand your connection in the Remote Systems view until you find the file you want to edit.
2. Double-click the file, or right-click the file and select **Open With > Remote Systems LPEX Editor**.



```
000400   F*                e
000500   F******************
000600   F*  INDICATORS USED
         F*  50 - No record
         F*  KE - Return to
         F*  KF - Return to
         F*  KG - Return to
         F*  LR - Last recor
         F******************
         F*  SUBROUTINES USE
         F*  EDITSL - Edit a
         F*  ACDESR - Edit a
```

Browsing Members

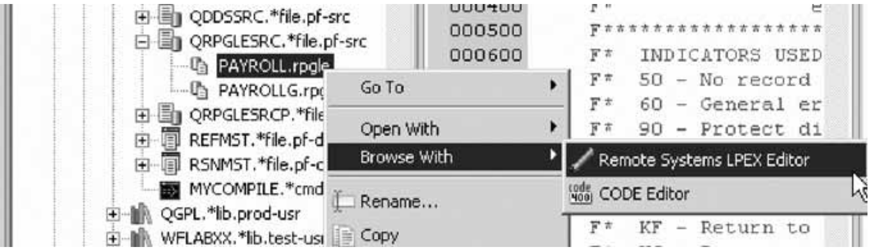
You can browse in SEU to view a member without the risk of accidentally changing it. You can use the scanning and positioning operations (such as Page Down and Page Up).

To browse a member in SEU:

- Specify 5 (Browse) for the Option parameter on the STRSEU command.
- Select option 5 (Browse) on the Work with Members Using SEU display.

In Development Studio Client, to browse a member:

1. Right-click the (member) file in the Remote Systems view and then s
2. Selecting **Browse with > Remote Systems LPEX Editor** from the pop-up menu.



```
000400   F*                     e
000500   F******************
000600   F*  INDICATORS USED
         F*  50 - No record
         F*  60 - General er
         F*  90 - Protect di
         F*  KF - Return to
```

Browsing in full screen mode

Full screen mode in SEU you can remove the format line from the top and the function key list from the bottom of the Browse display, giving you four additional Browse records to work with.

To switch to full screen mode:

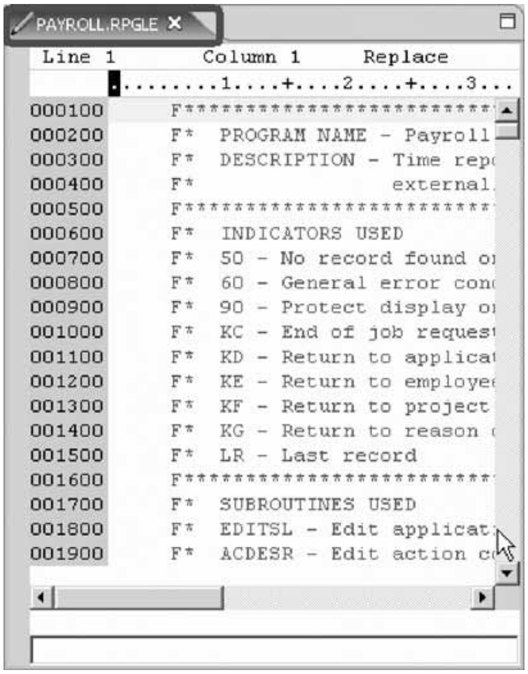
1. On the Browse screen, press F13 (Change session defaults). The Change Session Defaults screen appears.
2. Type Y at the Full Screen mode prompt and press Enter. The browser display changes to full screen mode.

To cancel full screen mode and return to normal operation:

1. On the Browse screen, press F13 (Change session defaults). The Change Session Defaults screen appears.
2. Type N at the Full screen mode prompt and press Enter. The normal Browse screen appears.

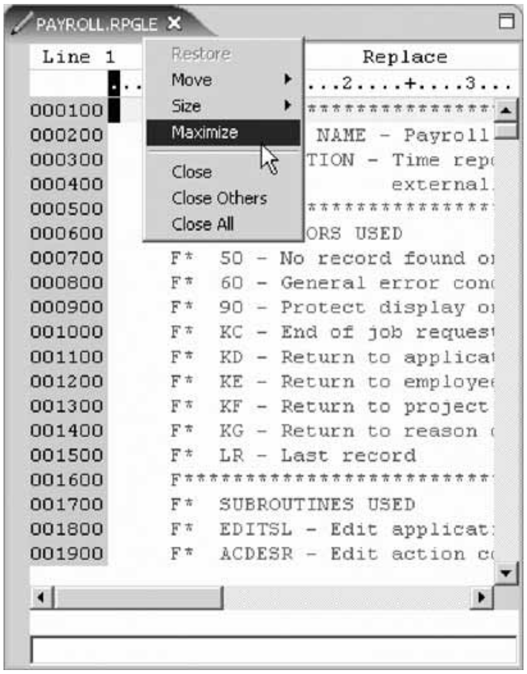
In Development Studio Client, you can switch to full screen mode in two ways:

- On the LPEX Editor view, double-click the resource name tab.



```
Line 1        Column 1     Replace
      .........1....+....2....+....3...
000100      F*************************
000200      F*  PROGRAM NAME - Payroll
000300      F*  DESCRIPTION - Time repo
000400      F*                external.
000500      F*************************
000600      F*  INDICATORS USED
000700      F*  50 - No record found or
000800      F*  60 - General error cond
000900      F*  90 - Protect display or
001000      F*  KC - End of job request
001100      F*  KD - Return to applicat
001200      F*  KE - Return to employee
001300      F*  KF - Return to project
001400      F*  KG - Return to reason c
001500      F*  LR - Last record
001600      F*************************
001700      F*  SUBROUTINES USED
001800      F*  EDITSL - Edit applicati
001900      F*  ACDESR - Edit action co
```

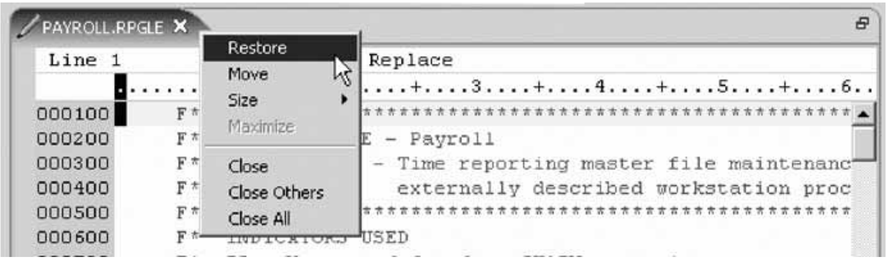
- In the LPEX Editor view, right-click the resource name tab and select **Maximize**.



```
Line 1                           Replace
     ..                   ...2....+....3...
000100                    ****************
000200                     NAME - Payroll
000300                    TION - Time repo
000400                            external
000500                    ****************
000600                    ORS USED
000700        F*   50 - No record found o
000800        F*   60 - General error cond
000900        F*   90 - Protect display o
001000        F*   KC - End of job request
001100        F*   KD - Return to applicat
001200        F*   KE - Return to employee
001300        F*   KF - Return to project
001400        F*   KG - Return to reason
001500        F*   LR - Last record
001600        F************************
001700        F*   SUBROUTINES USED
001800        F*   EDITSL - Edit applicat
001900        F*   ACDESR - Edit action c
```

To cancel full screen mode:

1. Double-click the resource name tab.
2. Right-click the resource name tab and select Restore.



```
Line 1                       Replace
      ......             ....+....3....+....4....+....5....+....6..
000100    F*             ****************************************
000200    F*             E - Payroll
000300    F*              - Time reporting master file maintenanc
000400    F*                externally described workstation proc
000500    F*             ****************************************
000600    F*             USED
```

Splitting the Browse Display

In SEU, while browsing a member, you can split the browse screen to browse another member or a spooled file. To split the Browse screen:

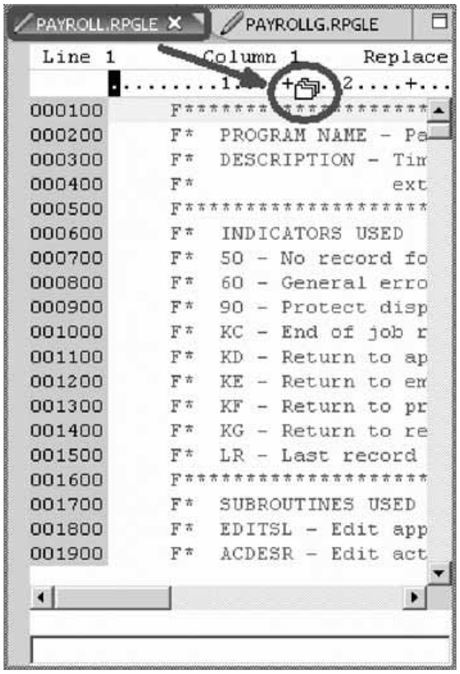
1. Press F15 (Browse options). The Browse Options screen appears.
2. Browse for the member you want:
   - To browse another member in the same file, type the member name at the Browse member prompt, and press Enter.
   - To browse another member in a different file, type the file name in the File prompt, type the member name at the Browser member prompt, and press Enter.

A split line (the line separating the two browse sessions) is placed in the middle of the Browse screen, and you can view the additional member. You can move the split line by using F6 (Move split line).

In Development Studio Client, by default, editors are stacked in the editor area, but you can choose to tile them in order to view source files simultaneously.

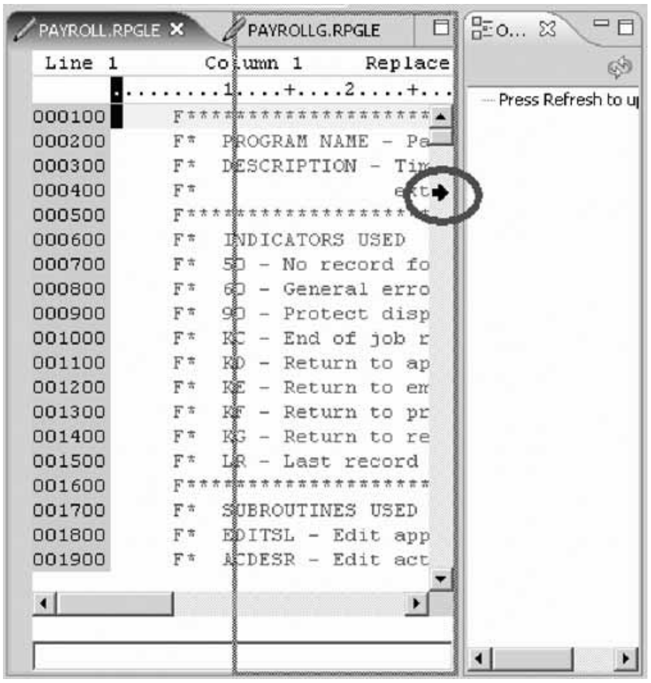
To tile the editors:

1. With two or more files open in the editor area, select one of the editor tabs.
2. Holding down the left mouse button, drag that editor over the left, right, top or bottom border of the editor area.



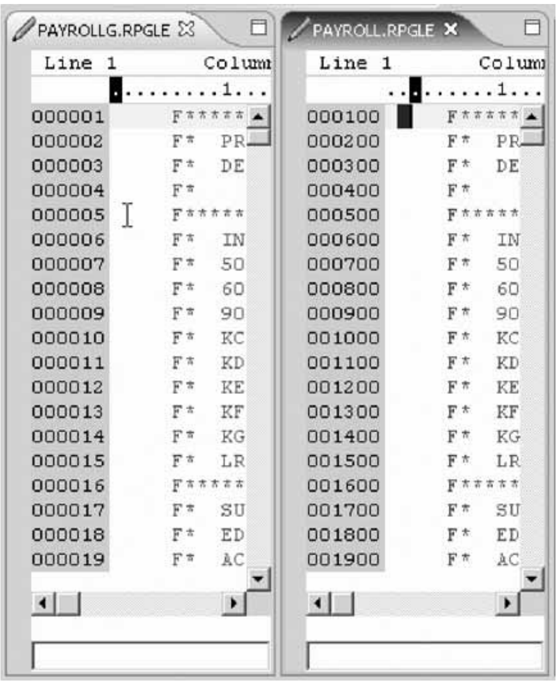
```
Line 1        Column 1      Replace
       .........1....+....2....+...
000100         F*******************
000200         F*  PROGRAM NAME - Pa
000300         F*  DESCRIPTION - Tim
000400         F*                ext
000500         F*******************
000600         F*  INDICATORS USED
000700         F*  50 - No record fo
000800         F*  60 - General erro
000900         F*  90 - Protect disp
001000         F*  KC - End of job r
001100         F*  KD - Return to ap
001200         F*  KE - Return to em
001300         F*  KF - Return to pr
001400         F*  KG - Return to re
001500         F*  LR - Last record
001600         F*******************
001700         F*  SUBROUTINES USED
001800         F*  EDITSL - Edit app
001900         F*  ACDESR - Edit act
```

Notice that the mouse pointer changes to a ″drop cursor″ that indicates where the editor session will be moved when you release the mouse button.



```
Line 1        Column 1      Replace
       .........1....+....2....+...
000100         F*******************
000200         F*  PROGRAM NAME - Pa
000300         F*  DESCRIPTION - Tim
000400         F*                ext
000500         F*******************
000600         F*  INDICATORS USED
000700         F*  50 - No record fo
000800         F*  60 - General erro
000900         F*  90 - Protect disp
001000         F*  KC - End of job r
001100         F*  KD - Return to ap
001200         F*  KE - Return to em
001300         F*  KF - Return to pr
001400         F*  KG - Return to re
001500         F*  LR - Last record
001600         F*******************
001700         F*  SUBROUTINES USED
001800         F*  EDITSL - Edit app
001900         F*  ACDESR - Edit act
```

3. (Optional) To resize, drag the borders of the editing area of each editor.



```
Line 1         Colum
 .........1...
000001         F*****
000002         F*  PR
000003         F*  DE
000004         F*
000005         F*****
000006         F*  IN
000007         F*  50
000008         F*  60
000009         F*  90
000010         F*  KC
000011         F*  KD
000012         F*  KE
000013         F*  KF
000014         F*  KG
000015         F*  LR
000016         F*****
000017         F*  SU
000018         F*  ED
000019         F*  AC
```

```
Line 1         Colum
  ..........1...
000100         F*****
000200         F*  PR
000300         F*  DE
000400         F*
000500         F*****
000600         F*  IN
000700         F*  50
000800         F*  60
000900         F*  90
001000         F*  KC
001100         F*  KD
001200         F*  KE
001300         F*  KF
001400         F*  KG
001500         F*  LR
001600         F*****
001700         F*  SU
001800         F*  ED
001900         F*  AC
```

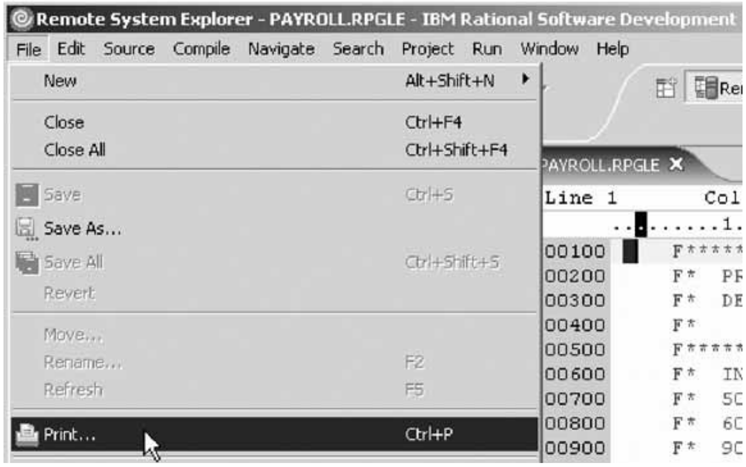
Printing a Member

To print a member in SEU:

- Specify 6 (Print) for the OPTION parameter of the STRSEU command.
- Select option 6 (Print) on the Work with Members Using SEU screen.
- When you exit from an edit session, type Y in the Print member prompt on the Exit screen.
- When you browse or edit a member, use the Line Print line command.

In Development Studio Client, to print a given source file in the editor:

1. Select the Resource Name Tab in the editor.
2. Select **File > Print** from the main menu bar.



```
Line 1       Col
  ..........1.
00100        F*****
00200        F*  PF
00300        F*  DE
00400        F*
00500        F*****
00600        F*  IN
00700        F*  5C
00800        F*  6C
00900        F*  9C
```

Changing the Edit Session Environment

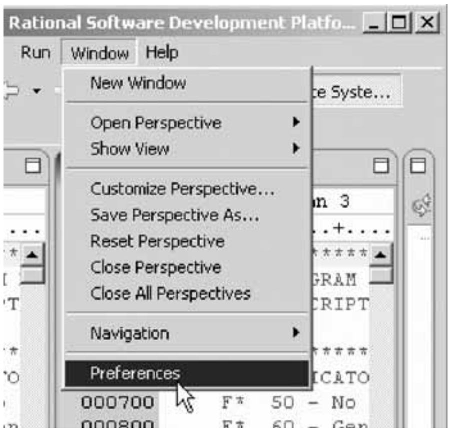
Changing the default case for a member type

The default case depends on the source type. To change the default case setting for a member type in SEU:

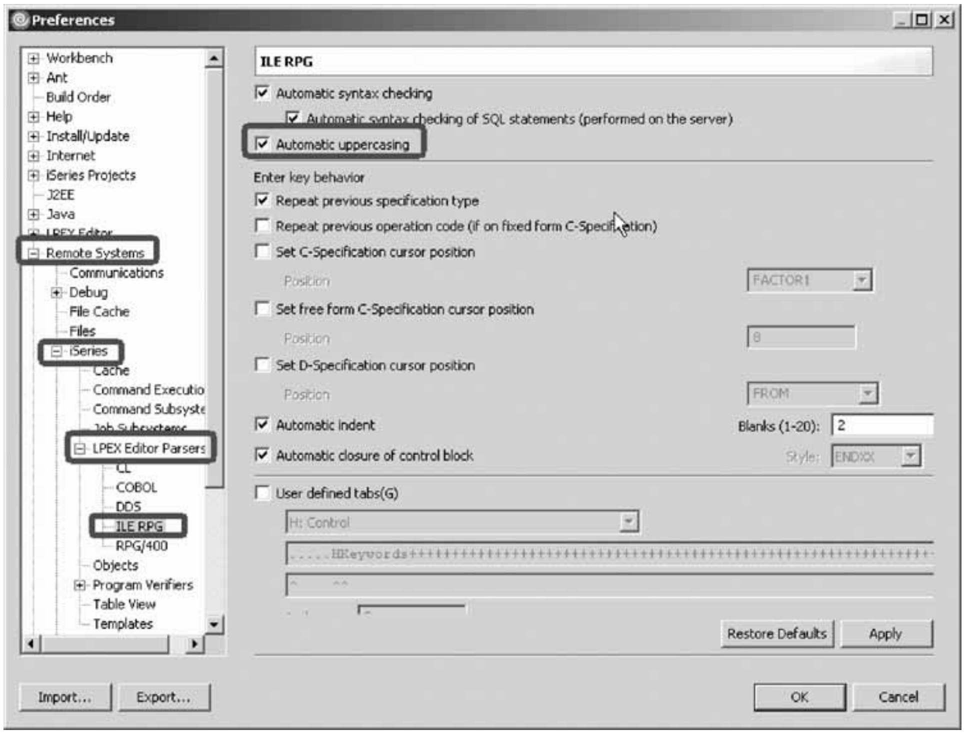
1. Edit a member of the type that you want to change.
2. Press F13 (Change session defaults).
3. Press the Page Down key to access the second part of the Change Session Defaults screen.
4. Type a response at the prompt in the Default to uppercase input for this source type:
   - Y (Yes) to assign the current source member type an uppercase input default for the current and subsequent edit sessions
   - N (No) to assign the current source member type a mixed-case input default for the current and subsequent edit sessions

To change the default case settings for a member type in LPEX:

1. **Window** > **Preferences** > **Remote Systems** > **iSeries** > **LPEX Editor Parsers**.



Select the **Automatic uppercasing** check box on the appropriate source type to update the default settings.



ILE RPG

☑ Automatic syntax checking

☑ Automatic syntax checking of SQL statements (performed on the server)

☑ Automatic uppercasing

Enter key behavior

☑ Repeat previous specification type

☐ Repeat previous operation code (if on fixed form C-Specification)

☐ Set C-Specification cursor position

Position: FACTOR1

☐ Set free form C-Specification cursor position

Position: 8

☐ Set D-Specification cursor position

Position: FROM

☑ Automatic indent — Blanks (1-20): 2

☑ Automatic closure of control block — Style: ENDXX

☐ User defined tabs(G)

H: Control

```
.....HKeywords+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

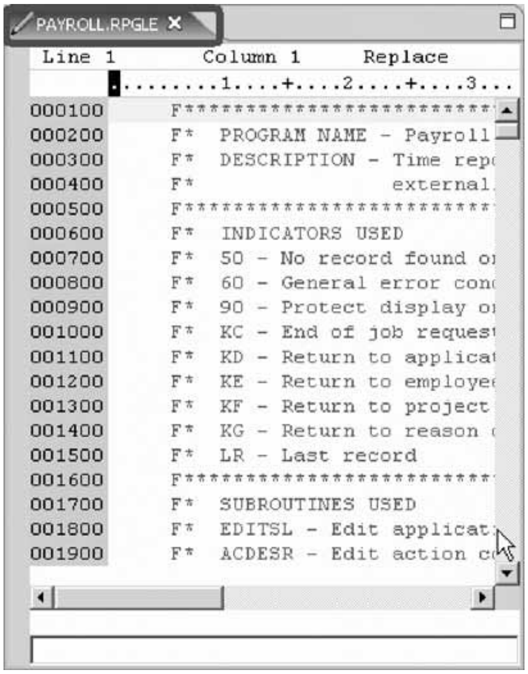
Editing in full screen mode

In SEU, to switch to full screen mode, press F13 (Change session defaults) on the Edit display to access the Change Session Defaults display, and type Y in the Full screen mode prompt.

To cancel full screen mode and return to the normal mode of operation in SEU, press F13 (Change session defaults) to access the Change Session Defaults display, type N at the Full screen mode prompt, and press Enter. SEU returns the normal Edit screen. If you use SEU language prompts, full language prompts appear.

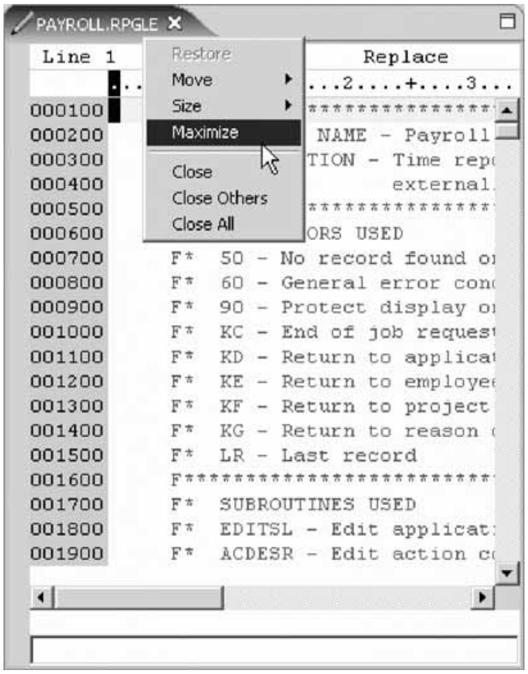
In the LPEX editor, you can switch to full screen mode in two ways:

- On the LPEX Editor view, double-click the resource name tab.



```
 Line 1          Column 1      Replace
          .........1....+....2....+....3...
000100          F***************************
000200          F*  PROGRAM NAME - Payroll
000300          F*  DESCRIPTION - Time rep
000400          F*                external.
000500          F***************************
000600          F*  INDICATORS USED
000700          F*  50 - No record found o
000800          F*  60 - General error con
000900          F*  90 - Protect display o
001000          F*  KC - End of job reques
001100          F*  KD - Return to applica
001200          F*  KE - Return to employe
001300          F*  KF - Return to project
001400          F*  KG - Return to reason
001500          F*  LR - Last record
001600          F***************************
001700          F*  SUBROUTINES USED
001800          F*  EDITSL - Edit applicat
001900          F*  ACDESR - Edit action c
```

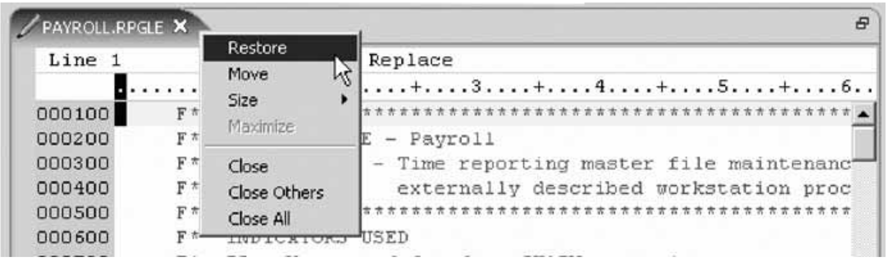
- In the LPEX Editor view, right-click the resource name tab and select **Maximize**.



```
 Line 1                        Replace
                          ...2....+....3...
000100                    ****************
000200                     NAME - Payroll
000300                    TION - Time rep
000400                          external.
000500                    ****************
000600                    ORS USED
000700          F*  50 - No record found o
000800          F*  60 - General error con
000900          F*  90 - Protect display o
001000          F*  KC - End of job reques
001100          F*  KD - Return to applica
001200          F*  KE - Return to employe
001300          F*  KF - Return to project
001400          F*  KG - Return to reason
001500          F*  LR - Last record
001600          F***************************
001700          F*  SUBROUTINES USED
001800          F*  EDITSL - Edit applicat
001900          F*  ACDESR - Edit action c
```

To cancel full screen mode in two ways:

- Double-click the resource name tab.
- Right-click the resource name tab and select **Restore**.



Finding and changing character strings

Finding and changing strings

To find a string in SEU:

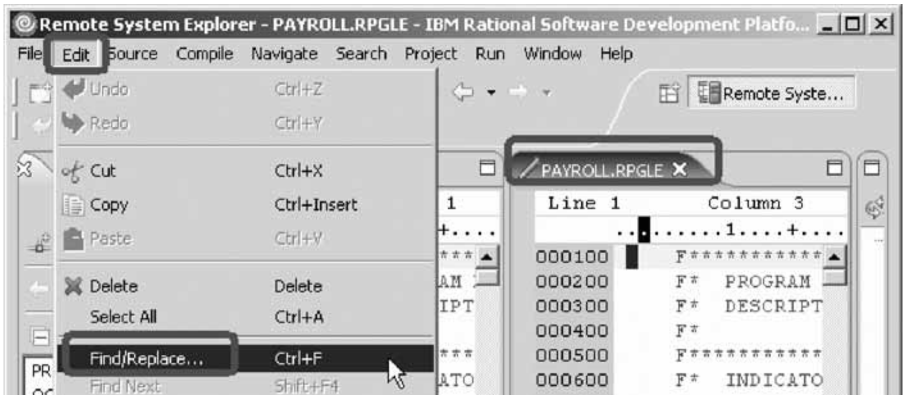
1. Type the string you want to find at the Find prompt of the Find Options screen or the Find / Change Options screen.
2. Change the other prompts, if necessary.
3. Press F16 (Find) to find the string.

To find and change a string in SEU:

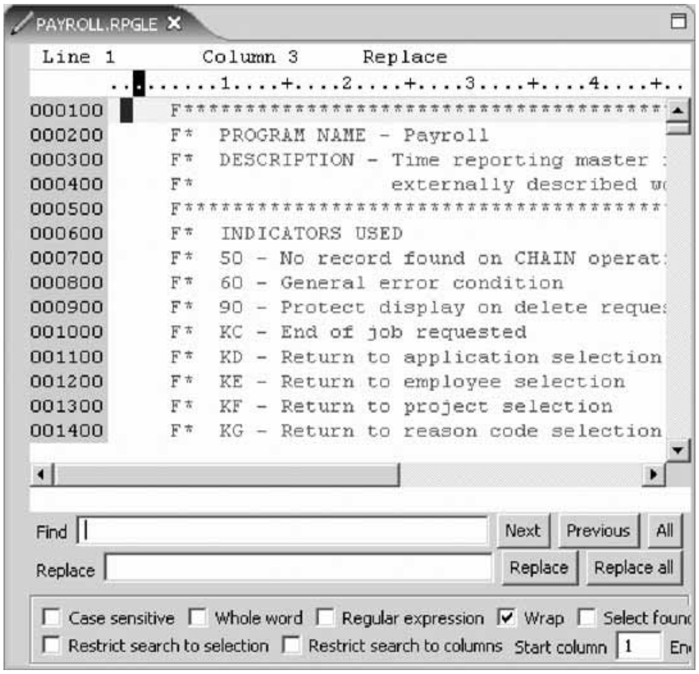
1. Type the string you want to change at the Find prompt of the Find / Change Options screen.
2. Type the string you want to replace it with at the Change prompt.
3. Change the other prompts, if necessary.
4. Press F17 (Change). The occurrences of the string that you specified are changed.

In the LPEX Editor, to locate a text string in the file you are editing and replace it with another string if you want to::

1. Select **Edit > Find/Replace** (or **Ctrl+F**) to open the Find / Replace window.



2. Enter the text you want to find in the **Find** field, and then enter the replacement text in the **Replace with** field.



```
Line 1        Column 3     Replace
       .........1....+....2....+....3....+....4....+..
000100      F****************************************
000200      F*  PROGRAM NAME - Payroll
000300      F*  DESCRIPTION - Time reporting master :
000400      F*                externally described w
000500      F****************************************
000600      F*  INDICATORS USED
000700      F*  50 - No record found on CHAIN operat:
000800      F*  60 - General error condition
000900      F*  90 - Protect display on delete reque:
001000      F*  KC - End of job requested
001100      F*  KD - Return to application selection
001200      F*  KE - Return to employee selection
001300      F*  KF - Return to project selection
001400      F*  KG - Return to reason code selection
```

□ Case sensitive □ Whole word □ Regular expression ☑ Wrap □ Select founc
□ Restrict search to selection □ Restrict search to columns Start column 1 En

3. By default, the Find/Replace window searches the entire file. You can optionally limit the search to selected text by selecting the Selected Lines radio button.
4. Click the **Find** button to find and highlight the first occurrence of the string, and any succeeding occurrence.
5. To replace the string and go to the next occurrence of the text in the Find field, click **Replace/Find**. Repeat this step until you have replaced all the occurrences of this text string that need to be replaced. To replace all the occurrences of the text string with just one command, click **Replace All** button.

Finding a string in a list of members

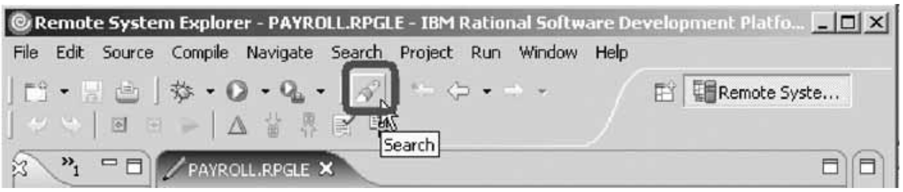
In SEU, you can use the find string function (option 25 on the Work with Members Using PDM display) of the programming development manager (PDM) to search selected members in a list or pair of a list for a character string. You can use an option of the find string function to use SEU to edit members that contain the string you specify.

If you specify 2 (Edit) for the Option prompt on the PDM Find String screen, PDM calls SEU to edit the selected members in the list that contain the string you specify. Also, PDM passes the values for the Find Columns to search and Kind of Match prompts to the Find/Change Options display of SEU each time it calls SEU.

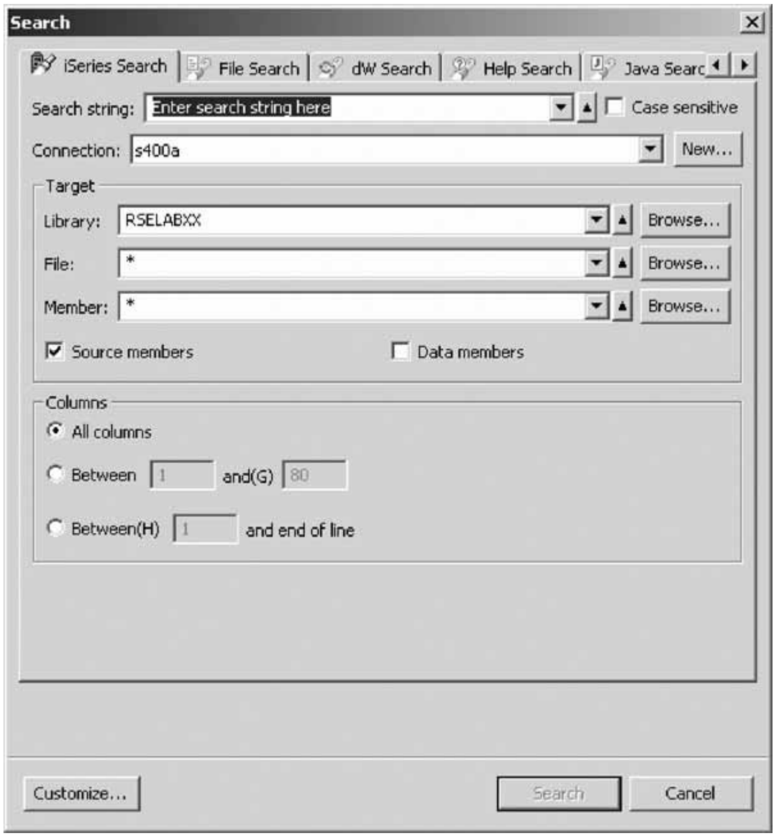
When SEU starts, the cursor is positioned on the first occurrence of the find string. Press F16 (Find) to find each subsequent occurrence.

In Development Studio Client, to search for text strings:

1. In the main workbench toolbar, click search button .



2. Click the **iSeries Search** tab
3. In the **Search string** field, type the string. You can also select it from a list of recently performed searches.



Search string: Enter search string here ☐ Case sensitive

Connection: s400a

Library: RSELABXX

File: *

Member: *

☑ Source members ☐ Data members

● All columns

○ Between 1 and(G) 80

○ Between(H) 1 and end of line

4. Select or clear the **Case sensitive** check box, depending on whether or not you want to perform a case sensitive or insensitive search.
5. The Scope area displays the targets that you have selected. If you are searching libraries or library filters, two check boxes beneath this area are enabled: **Source members** and **Data members**. Source members is checked by default. Check Data members if you want you also want to search data members.
6. If applicable, specify columns to search with the **Columns** radio buttons. By default, all columns are searched.
7. Click **Search**. Note that you can click **Cancel** to cancel the search while it is in progress and that the Search window shows the progress of the search.

Entering and editing text using the LPEX Editor

You can edit source physical file members and display, change, and save source members in your source physical files through the iSeries Objects subsystem, using the Remote Systems LPEX editor.

Note: During development, your file is locked on the remote system so that others cannot change it. Accordingly, if you attempt to edit a locked member, the workbench informs you that the file is in use and allows you to open the member in browse (read-only) mode. Also note that the workbench releases these locks on remote members if you disconnect from a subsystem or shut down the workbench while remote members are open in the editor.

Entering text

- Click in the working area of the editor and begin typing.
- To type on the command line, click in the command line area or press the Esc key and then begin typing.

Replacing and inserting text

- The status line indicates which mode the editor is in. When the editor is in Replace mode, the cursor appears as a solid block, one character wide. Text overlaid by this block will be replaced by any new text that you type. When the editor is in Insert mode, the cursor appears as a thin vertical line. Any new text that you type is inserted into the file at the cursor position, and existing text after the cursor will be shifted.
- To toggle back and forth between the two modes, press the Insert key.

Deleting text

There are many ways to delete unwanted text.

- To delete single characters:
  - Press Delete. If you are in Insert mode, the character to the right of the cursor is deleted; if you are in Replace mode, the character that is overlaid by the cursor is deleted. Text after the cursor is shifted to fill the resulting gap.
  - Press Backspace. The character before the cursor is deleted. Text after the cursor is shifted to fill the resulting gap.
- To delete all the text between the cursor and the end of the current line, press Ctrl+Delete.
- To delete a complete line, move the cursor to the unwanted line and press Ctrl+Backspace.

Moving the cursor

To move the cursor:

- Position the mouse pointer where you want the cursor to be, and click mouse button 1.
- Press the Up, Down, Left, or Right arrow.
- Press Home to move the cursor to the beginning of a line, or End to move it to the end of a line.
- Press Ctrl+Left arrow to move the cursor one word left, or Ctrl+Right arrow to move it one word right.
- Press Page Up or Page Down to move the cursor up or down one window at a time.
- Press Ctrl+Home to move the cursor to the beginning of the file, or Ctrl+End to move it to the end of the file.
- Press Ctrl+J to return the cursor to the place in the file where you last entered text.

Note: If you inadvertently select text by dragging the mouse, unselect it by pressing Alt+U, or, click the right-click and select **Deselect** from the pop-up menu.

Inserting a blank line

- Move the cursor to any point on a line and press Ctrl+Enter.
- Move the cursor to the end of a line and press Enter. A blank line is created after the current line, and the cursor moves to the first column position of this new line.

Splitting and joining lines of text

- To split a line, place the cursor where you want the break to occur, and press either Alt+S or Enter. Pressing Alt+S leaves the cursor at its current position; pressing Enter moves the cursor to the beginning of the new line.

  When you split a line, all the text to the right of the cursor is moved down to a new line.
- To join two lines, with the cursor anywhere on a line, press Alt+J to join the current line with the next line together.

Content assist

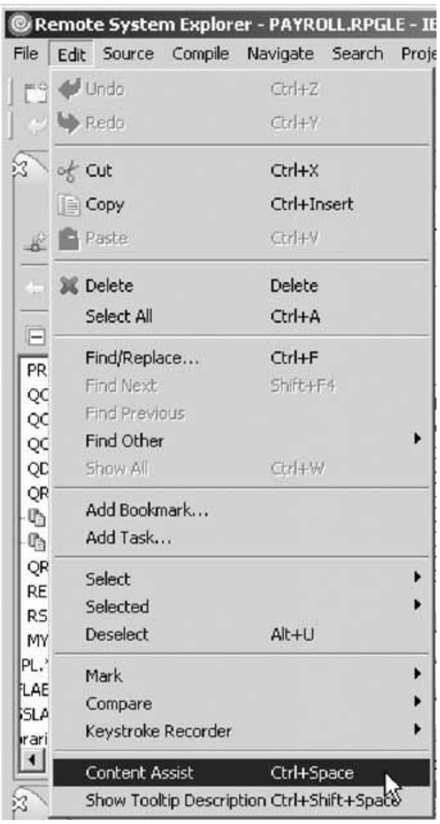
The purpose of content assist is to propose, display, and insert completions at the cursor position. In addition to handling completions, content assist can also be requested to provide context information. Context information is shown in a tooltip-like pop-up dialog box.

When you activate content assist from a valid line of code in the Remote Systems LPEX Editor, the editor opens a list of valid code completions.

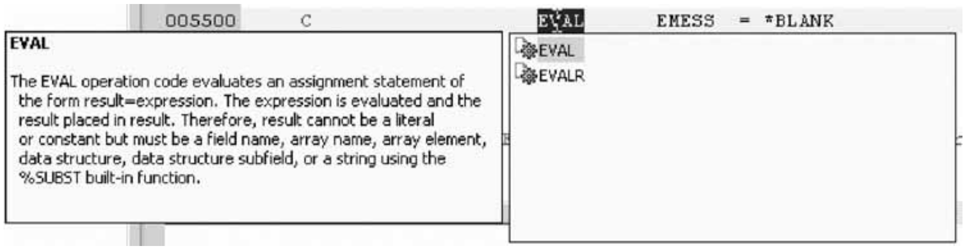
Content assist is available for ILE RPG, ILE COBOL, and C++ programs. You can use it along with specialized templates to help you complete your code. ILE COBOL provides content assist for all its four divisions - Identification, Environment, Data, and Procedure. ILE COBOL also extends the scope of content assist to include support for compiler-directing statements.

To use content assist:

1. In the Remote System Explorer, open a file you want to edit.
2. From the **Edit** menu, select **Content Assist** or press Ctrl+Space.



3. Double-click on a possible completion from the list.



```
005500      C                          EVAL      EMESS    = *BLANK
```

EVAL

The EVAL operation code evaluates an assignment statement of the form result=expression. The expression is evaluated and the result placed in result. Therefore, result cannot be a literal or constant but must be a field name, array name, array element, data structure, data structure subfield, or a string using the %SUBST built-in function.

If you are editing COBOL member in the LPEX editor, you can type the partial text for a template alias and press Ctrl+Space to view a list of valid completions. For example, if you type `if` in ILE COBOL, followed by Ctrl+Space a list of completions appears in the content assist popup. One of the completions is the IF - ELSE - END-IF template. When you select that template, the actual text is inserted in the LPEX editor.

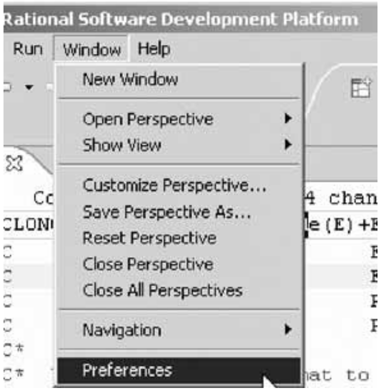
Creating a new template

Template support is available for both ILE RPG, ILE COBOL, and C++ programs. You can use templates to predefine a structured description of coding patterns that reoccur in your source code. The LPEX editor supports the use of templates to fill in common source patterns. For example, if you often use a particular coding pattern, you can avoid typing it each time that you want to use it by starting with a template. Invoking content assist at the point of this pattern in your code gives you a list of possible templates. Selecting a template inserts the code into the LPEX editor.

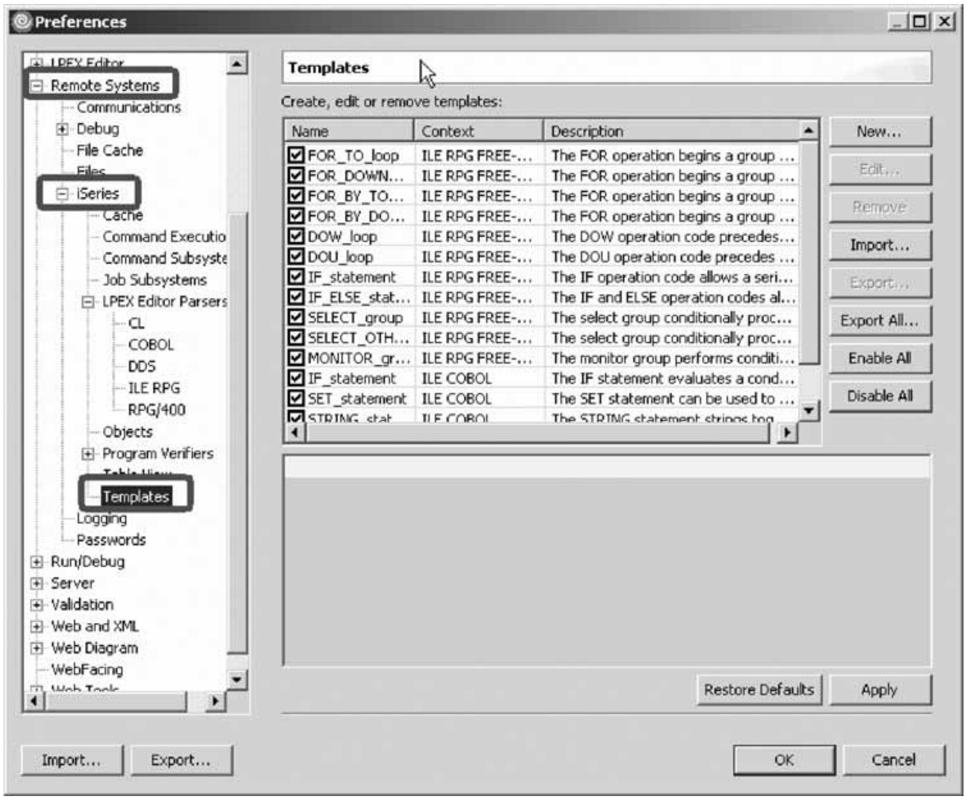
Restriction: For RPG source, you can only use templates for free-form RPG.

To create a template:

1. Select **Window** and click **Preferences**.



2. Expand **Remote Systems > iSeries**.



Templates

Create, edit or remove templates:

Name	Context	Description
☑ FOR_TO_loop	ILE RPG FREE-...	The FOR operation begins a group ...
☑ FOR_DOWN...	ILE RPG FREE-...	The FOR operation begins a group ...
☑ FOR_BY_TO...	ILE RPG FREE-...	The FOR operation begins a group ...
☑ FOR_BY_DO...	ILE RPG FREE-...	The FOR operation begins a group ...
☑ DOW_loop	ILE RPG FREE-...	The DOW operation code precedes...
☑ DOU_loop	ILE RPG FREE-...	The DOU operation code precedes ...
☑ IF_statement	ILE RPG FREE-...	The IF operation code allows a seri...
☑ IF_ELSE_stat...	ILE RPG FREE-...	The IF and ELSE operation codes al...
☑ SELECT_group	ILE RPG FREE-...	The select group conditionally proc...
☑ SELECT_OTH...	ILE RPG FREE-...	The select group conditionally proc...
☑ MONITOR_gr...	ILE RPG FREE-...	The monitor group performs conditi...
☑ IF_statement	ILE COBOL	The IF statement evaluates a cond...
☑ SET_statement	ILE COBOL	The SET statement can be used to ...
☑ STRING_stat	ILE COBOL	The STRING statement strings tog

3. Select **Templates** and click **New**.
4. In the new template window, type the name of the template, and select a language from the Context list.
5. In the **Content** field, type the code that you want content assist to insert.
6. Click **OK** twice.

To insert this new template, type the template name in your code, and press Ctrl+Space. The new template name that you created appears in a pop-up window. Double-click on this completion to insert the contents of the template into your code.

Editor profiles

You can configure the editor to adopt the keyboard and command personalities of many popular editors.

Most editor profiles differ only in the keys and commands used to perform various editor tasks. Some base editor profiles, listed below, also add a prefix information and command area at the start of each line:

- ispf
- seu
- xedit

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

The prefix area appears on the left side of the text area, and may display line numbers or sequence numbers. Line numbers are maintained by the editor.

You can configure the editor to adopt the keyboard and command personalities of many popular editors. To set an editor profile:

1. Select Window -> Preferences from the main menu
2. Select LPEX Editor
3. Select the editor profile that you want to use
4. Select Apply
5. Select OK

SEU base profile

The seu base profile has default key assignments and prefix commands that should be familiar to SEU users.

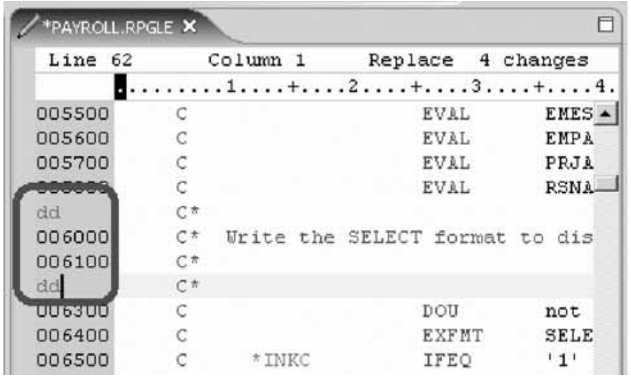
Key Settings

Here are some of the common keyboard shortcuts settings:

Shortcut	Action
Ctrl+F	find
Ctrl+Home	top
Ctrl+End	bottom
Ctrl+J	findLastChange
Ctrl+L	locateLine
Ctrl+N	findNext
Ctrl+P	print
Ctrl+Shift+F	findAndReplace
Ctrl+Shift+Z	redo
Ctrl+S	save
Ctrl+Y	redo
F1	help
Alt+S	split a line
Alt+J	join current line with the next line

Prefix commands (line commands)

Prefix commands work the same way as line commands in SEU.



```
Line 62        Column 1      Replace   4 changes
       ........1....+....2....+....3....+....4.
005500     C                      EVAL      EMES
005600     C                      EVAL      EMPA
005700     C                      EVAL      PRJA
005800     C                      EVAL      RSNA
dd         C*
006000     C*  Write the SELECT format to dis
006100     C*
dd         C*
006300     C                      DOU       not
006400     C                      EXFMT     SELE
006500     C       *INKC          IFEQ      '1'
```

Prefix command	Result
n	Repositions the cursor on the line with the specified line number. n is a positive integer.
+	Scrolls down one line.
+n	Scrolls down n lines. n is a positive integer.
-	Scrolls up one line.
-n	Scrolls up n lines. n is a positive integer.
/	Makes the specified line the current line.
A	Copies or moves lines after the specified line. The source lines and the type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM.
An	Copies or moves lines after the specified line. The source lines and the type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM. n copies of the source will be inserted after the specified line. n is a positive integer.
B	Copies or moves lines before the specified line. The source lines and type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM.
Bn	Copies or moves lines before the specified line. The source lines and type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM. n copies of the source will be inserted before the specified line. n is a positive integer.
C	Indicates that the specified line is to be the source for a copy operation. The target location for the copy is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO.
Cn	Indicates that the specified line and the following n - 1 visible lines are to be the source for a copy operation. n is a positive integer. The target location for the copy is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO.
CC in two different lines	The CC commands delimit the source for a copy operation. Only visible lines are included in the source. The target location for the copy is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO.

CCR in two different lines	The CCR commands delimit the source for a copy operation. Only visible lines are included in the source. The target location for the copy is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO. After the copy operation has been completed, the CCR commands are not cleared from the prefix area. They are retained to allow you to specify another target.
CR	Indicates that the specified line is to be the source for a copy operation. The target location for the copy is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO. After the copy operation has been completed, the CR command is not cleared from the prefix area. It is retained to allow you to specify another target.
CRn	Indicates that the specified line and the following n - 1 visible lines are to be the source for a copy operation. n is a positive integer. The target location for the copy is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO. After the copy operation has been completed, the CRn command is not cleared from the prefix area. It is retained to allow you to specify another target.
D	Deletes the specified line.
Dn	Deletes the specified line and the following n - 1 visible lines.
DD in two different lines	Delimits a block of text that you want deleted. Only the visible lines will be deleted.
I	Inserts a new line after the specified line.
In	Inserts n new lines after the specified line. n is a positive integer.
IS	Inserts the skeleton line after the specified line. The skeleton line is specified with the S command.
ISn	Inserts n copies of the skeleton line after the specified line. The skeleton line is specified with the S command.
L	Shifts the specified line one character to the left. If there is not at least one blank at the start of the line, nothing is done.
Ln	Shifts the specified line n characters to the left. If there is are less than n spaces at the start of the line, only the spaces are deleted.
LL in two different lines	Delimits a block of text that you want to shift one character to the left. Only the visible lines will be shifted. Lines that do not have at least one space at the start of the line will be unaffected.
LLn in two different lines	Delimits a block of text that you want to shift n characters to the left. Only the visible lines will be shifted. Lines that have less than n spaces at the start of the line will only have those spaces deleted.
LLT in two different lines	Delimits a block of text that you want to shift one character to the left. Only the visible lines will be shifted.
LLTn in two different lines	Delimits a block of text that you want to shift n characters to the left. Only the visible lines will be shifted.
LP	Prints the specified line.
LPn	Prints the specified line and the following n - 1 visible lines.

LPP in two different lines	Delimits a block of text that you want to print. Only the visible lines will be printed.
LT	Shifts the specified line one character to the left.
LTn	Shifts the specified line n characters to the left.
M	Indicates that the specified line is to be the source for a move operation. The target location for the move is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO.
Mn	Indicates that the specified line and the following n - 1 visible lines are to be the source for a move operation. n is a positive integer. The target location for the move is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO.
MM in two different lines	Delimits the source for a move operation. Only visible lines are included in the source. The target location for the move is determined by the first occurrence of one of the following: A, An, B, Bn, O, On, OO or OO.
O	Causes the editor to overlay the specified line. The source lines and the type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM.
On	Causes the editor to overlay the specified line and the following n - 1 visible lines. The source lines and the type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM.
OO in two different lines	Delimits the overlay target for a move or copy operation. Only the visible lines will be overlaid. The source lines and the type of operation (move or copy) are determined by the first occurrence of one of the following commands: C, Cn, CC,CR, CRn, CCR, M, Mn, or MM.
R	Shifts the specified line one character to the right. If the line cannot be shifted without truncating a non blank character, nothing is done.
Rn	Shifts the specified line n characters to the right. If the line cannot be shifted n characters without truncating a non blank character, the line will only be shifted as far to the right as it can be without truncating a non blank character.
RP	Duplicates the specified line.
RPn	Duplicates the specified line n times.
RPP in two different lines	Delimits a block of text that you want duplicated. Only the visible lines will be duplicated.
RPPn in two different lines	Delimits a block of text that you want duplicated n times. Only the visible lines will be duplicated.
RR in two different lines	Delimits a block of text that you want to shift one character to the right. Only the visible lines will be shifted. Lines that cannot be shifted without truncating a non blank character will be unaffected.
RRn in two different lines	Delimits a block of text that you want to shift n characters to the right. Only the visible lines will be shifted. Lines that cannot be shifted without truncating a non blank character will only be shifted as far to the right as they can be without truncating a non blank character.

RRT in two different lines	Delimits a block of text that you want to shift one character to the right. Only the visible lines will be shifted.
RRTn in two different lines	Delimits a block of text that you want to shift n characters to the right. Only the visible lines will be shifted.
RT	Shifts the specified line one character to the right.
RTn	Shifts the specified line n characters to the right.
S	Saves the specified line as the skeleton line. The skeleton line is used by the IS and ISn commands.
SF in an excluded block header	Shows the first line in the excluded block. Lines can be excluded with the X, Xn and XX commands.
SFn in an excluded block header	Shows the first n lines in the excluded block. Lines can be excluded with the X, Xn and XX commands.
SL in an excluded block header	Shows the last line in the excluded block. Lines can be excluded with the X, Xn and XX commands.
SLn in an excluded block header	Shows the last n lines in the excluded block. Lines can be excluded with the X, Xn and XX commands.
W	Scrolls the text area so that column 1 is visible.
Wn	Scrolls the text area so that column n is the first visible column on the left.
X	Excludes the specified line. The line can be reshown with the SF, SFn, SL, or SLn commands.
Xn	Excludes the specified line and the following n - 1 visible lines. The lines can be reshown with the SF, SFn, SL, or SLn commands.
XX	Delimits a block of text that you want to exclude. The lines can be reshown with the SF, SFn, SL, or SLn commands.

Using source entry utility commands

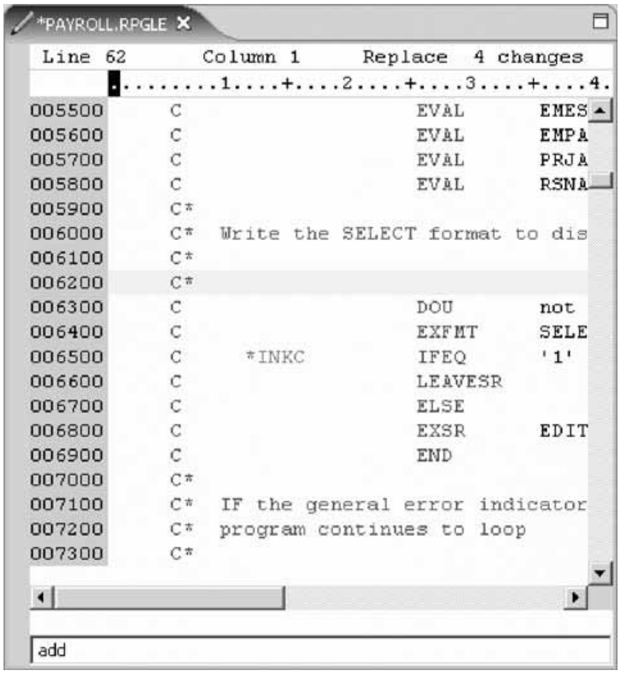
You can use editor commands to customize your editing environment, search for or change text in your document, or perform many other functions.

To issue an editor command:

1. Press **Esc** to go to the editor command line below the editing area.
2. Type your editor command on the command line, then press Enter to perform the command. For example, entering the command:

   ```
   add
   ```

   adds a new line into the file.



```
Line 62       Column 1      Replace   4 changes
       ........1....+....2....+....3....+....4.
005500      C                     EVAL      EMES
005600      C                     EVAL      EMPA
005700      C                     EVAL      PRJA
005800      C                     EVAL      RSNA
005900      C*
006000      C*  Write the SELECT format to dis
006100      C*
006200      C*
006300      C                     DOU       not
006400      C                     EXFMT     SELE
006500      C     *INKC           IFEQ      '1'
006600      C                     LEAVESR
006700      C                     ELSE
006800      C                     EXSR      EDIT
006900      C                     END
007000      C*
007100      C*  IF the general error indicator
007200      C*  program continues to loop
007300      C*
```

```
add
```

To recall a previously-used command:

1. While in the command line, press Up or Down. Previously used commands appear in the message window.
2. Select the command you want to reuse, then press Enter.

Command name	**Description**
action	Used to run an editor action.
add	Adds one or more blank lines into the current file.
block	Sets or manipulates a block of text.
compare	Examines two documents for differences in their content.
delete	Deletes one or more lines from the current file.
deleteText	Deletes text from the current file.
expandAll	Brings hidden lines back into view.
findText	Finds and optionally replaces text in a document.
get	Gets a file and inserts it after the current line.
help	Returns help information on a requested command or parameter.
input	Gets user input to use with an editor command.
insert	Inserts a new line into the current document.
insertShow	Inserts a new show element into the current document.

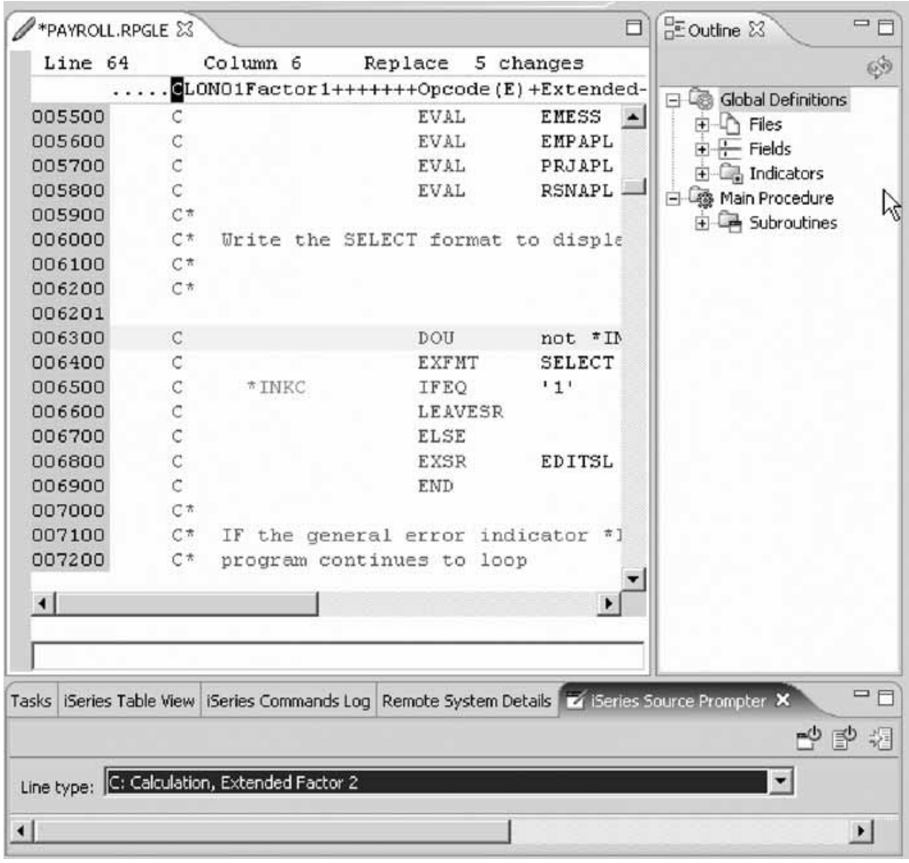
insertText	Inserts text at the current cursor position.
load	Loads or reloads a file.
locate	Moves the cursor to a specified element, line, mark, sequence number, or sequence text.
parse	Triggers the parser.
print	Prints the current document.
processPrefix	Processes prefix commands in the prefix area.
query	Returns the setting of a parameter.
replaceText	Replaces text at the current cursor position.
resequence	Resequences sequence numbers in a document.
save	Saves the contents of the current editing window into a file.
saveAsHtml	Saves the contents of the current editing window into an HTML file.
screenShow	Forces all document views to be refreshed.
set	Assigns a value to an editor parameter.
sort	Sorts editor elements.
undo	Used to undo one or more sets of document changes.
updateProfile	Updates selected editor parameters as they apply to a document view.

In addition, the following short commands can be issued from the editor command line:

Short command	**Description**
/text	Find the next occurrence of the specified text.
-/text	Find the specified text searching backwards from the current position.
’text	Find the next occurrence of the specified text. The search is a case-sensitive search.
-’text	Find the specified text searching backwards from the current position. The search is a case-sensitive search.
:lineNumber	Locate the specified line. lineNumber must be a positive integer.
lineNumber	Locate the specified line. lineNumber must be a positive integer.

Editing with prompts and format lines

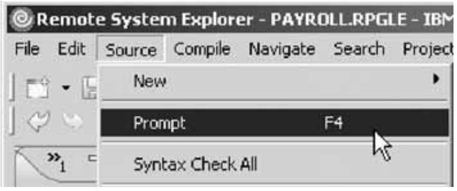
If you are using the Remote Systems LPEX editor, you might find editing easier with the source prompter.



```
 Line 64       Column 6      Replace  5 changes
        .....CLON01Factor1+++++++Opcode(E)+Extended-
005500       C                         EVAL      EMESS
005600       C                         EVAL      EMPAPL
005700       C                         EVAL      PRJAPL
005800       C                         EVAL      RSNAPL
005900       C*
006000       C*  Write the SELECT format to disple
006100       C*
006200       C*
006201
006300       C                         DOU       not *IN
006400       C                         EXFMT     SELECT
006500       C     *INKC               IFEQ      '1'
006600       C                         LEAVESR
006700       C                         ELSE
006800       C                         EXSR      EDITSL
006900       C                         END
007000       C*
007100       C*  IF the general error indicator *I
007200       C*  program continues to loop
```

Line type: C: Calculation, Extended Factor 2

To invoke the source prompter, click anywhere inside your file and press **F4**. You can also click **Source > Prompt** from the workbench menu.



Source prompter is available for RPG, DDS, and CL.

Chapter 5. Program verification

Program verifiers provide a full range of syntactic and semantic checking for your code, similar to compilers. Verifiers perform the same checking as compilers, but do not generate object code. Instead, program verifiers provide feedback and error messages you can use to fix your code before you compile. After you verify your program, you can view any information displays in the iSeries error list view so that you can fix your code before you compile.

Verification preferences provide many of the same options as the compiler. For example, you can set preferences to generate a listing for your source during verification, or whether or not to use local cache for /copy members or other externally described files.

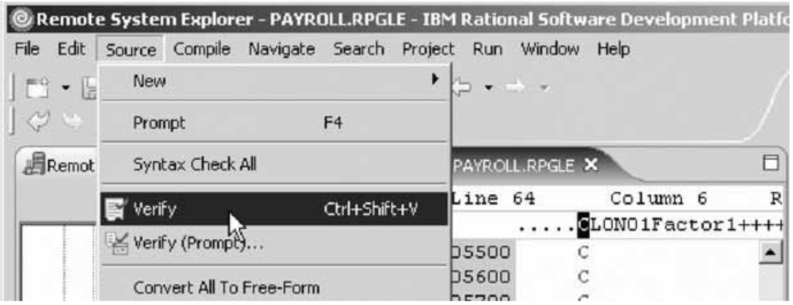
You can verify on programs written in the following languages:

- ILE RPG
- ILE COBOL
- DDS

Note: Verification is performed locally, on your workstation, whereas compiling is performed on the iSeries server.

To perform a verification in your source:

- Using the Remote Systems LPEX editor, click inside the editing area of your source member to make sure it is selected.
- From the workbench menu, select **Source > Verify** or **Source > Verify with Prompt** to specify additional parameters. Or, click Ctrl+Shift+V.



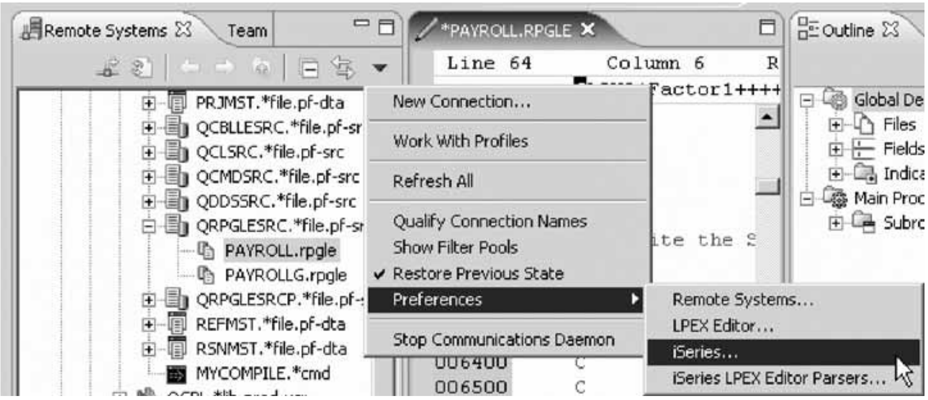
```
Line 64        Column 6      R
      .....CLON01Factor1++++
05500      C
05600      C
```

The iSeries error list view opens with feedback for your verification. Double-click an error to jump to the line in your source where the error occurs.

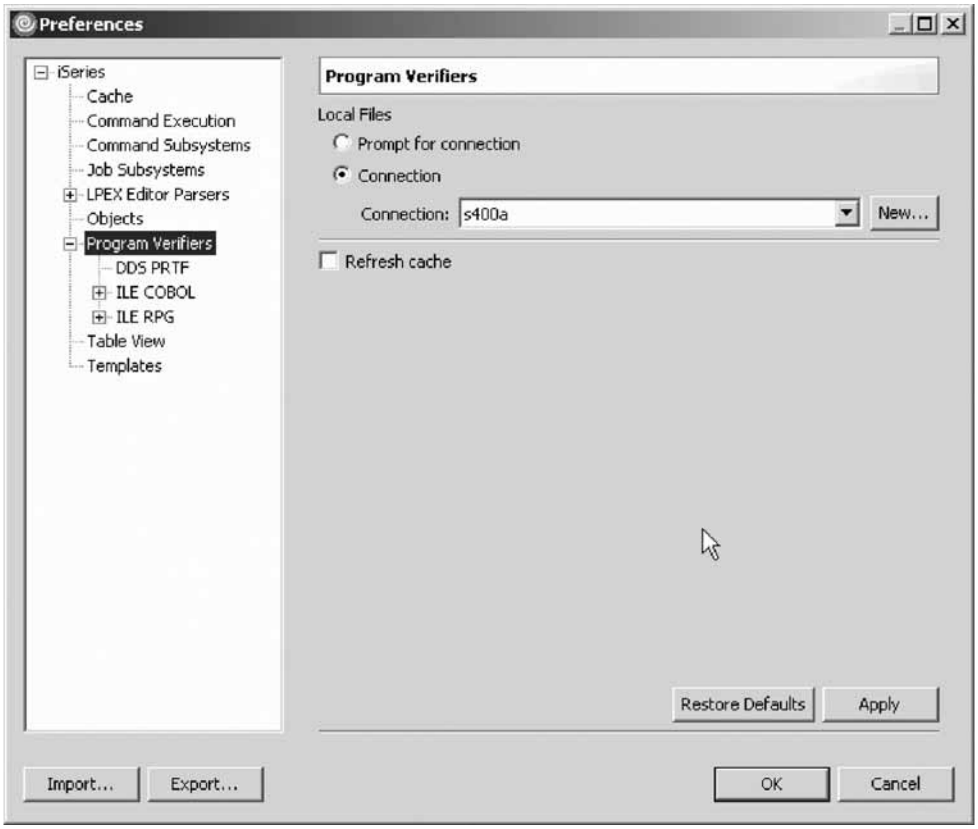
You control the behavior of the program verifier through their preference pages. The program verifier triggered during verification will correspond to the type of file your are editing. For example, if you are editing a ILE RPG file, the ILE RPG program verifier is triggered when you perform verification.

To access program verifier preferences:

- Click the drop-down menu from the Remote Systems view and select **Preferences > iSeries**.



- Expand Program Verifiers to see the preference pages for the various program verifiers.



Program Verifiers

Local Files

○ Prompt for connection

● Connection

Connection: s400a

□ Refresh cache

Chapter 6. Compiling

You can run a compile command on an iSeries server from the Remote System Explorer perspective, and you can retrieve error feedback from the compilation. When you compile, the workbench determines the member type, and then runs the last used compile command for that type. However, you can always change this behavior through Remote Systems preferences (see the related links).

You can also select multiple members from any source file from any library in your connection by holding down the CTRL key and clicking on all the members you want to compile. Then right-click to access the compile menu. Note that the workbench determines what order to compile multiple members, also controlled through the preferences. For example, you would most likely want to compile a DSPF member before a RPG member that depends on the result of the DSPF member compilation.

Note that multiple member compile is disabled if:

- One or more of the selected members types are not compatible.
- The selected members are not from the same iSeries connection.

A compile command is an identifier and is always associated with a source member type. A command string is the entire command that will be run on the server, for example, CRTRPGPGM PGM(MYLIB / MYPROG) SRCFILE(MYLIB / MYSRC) SRCMBR(MYMBR). A command string is identified with a compile command. Each profile in the Remote System Explorer has a set of source member types, and each source type has a set of compile names associated with them. You can add source types to a profile and add compile commands to a source type.

Compile commands can be defined per member type (QSYS native file system) or file type (hierarchical integrated file system), and IBM supplies a number of default compile commands for common member and file types. Compile commands can also be shared using team support.

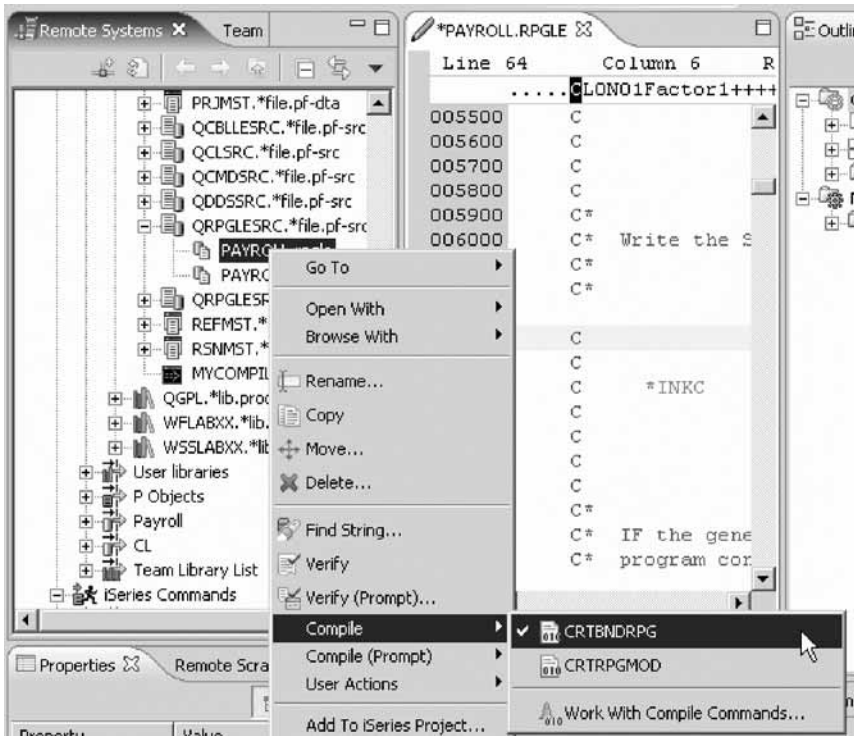
Compiling your programs

You can compile your iSeries programs from the Remote Systems view. The pop-up menu for source members includes compile submenus. The compile submenus contain several compile commands for various member types. A compile name is an identifier for a compile command to run on a remote system. For example, the compile label CRTBNDRPG tells the iSeries server to create a bound RPG program. Compile names are associated with source member types, such as *rpgle, *cl, *pf, *dspf, and so on. For this reason, you will only see the options to compile when you right-click these types of elements. There are two ways to compile: prompted and non-prompted. Both compile actions invoke a submenu with a list of compile names.

To compile a program:

1. In the Remote Systems view, drill down to a source physical file member and right-click. You can perform multiple selection for compilation by holding down the CTRL key and clicking various members (as long as they are compatible and from the same iSeries connection).

2. Select Compile > or Compile (Prompt) > and then one of the following:



```
 Line 64      Column 6      R
      .....CLON01Factor1++++
005500     C
005600     C
005700     C
005800     C
005900     C*
006000     C*  Write the S
           C*
           C*

           C
           C
           C      *INKC
           C
           C
           C
           C
           C*
           C*  IF the gene
           C*  program cor
```

- Select Appropriate Compile Command such as CRTBNDRPG: A compile command associated with your selected source member type will display in the pop-up menu for your convenience. For example, select CRTBNDRPG to create a bound RPG program for a selected RPG file. If you want to use a different compile command than one of the defaults, choose the second option, described below.
- Work With Compile Commands: Use the Work With Compile Commands window to choose the profile and the source member type that you want to work with from the drop-down lists at the top of the window, and add, change, or delete associated compile commands. This option is also helpful if you want to use a different compile command for your source member, other than the default associated compile command displayed in the pop-up menu.

If there are errors, the iSeries Error List view is automatically populated with the results of the compilation.

iSeries Error List

If a compile command (or a verification) that you run is not successful, the iSeries Error List view displays your error messages.

The view identifies each error with an icon that identifies the severity level of the error, the ID of the error, the message, the numerical 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. To show the location of the error in the source member, double-click on the error. This opens the source member in the Remote Systems LPEX editor, if it is not already open, and highlights the text in the source that caused the error. The error message might also be embedded in the document below the line that caused the error.

The iSeries Error List view sorts the entries in the current list. You can do this by clicking on the column header of the column that you want to use for sorting:

- Severity. (Click on the column labeled Severity).
- Results of more than one compile command. As each compile command produces an events file (the listing of errors), a new table is shown in the view.
- To move between different error lists, use the back and forward button near the top right corner of the iSeries Error List view.

The iSeries Error List view also filters which errors you want to see. To do this, click on the drop-down arrow near the top right hand corner of the view, and choose Show Severity, and then choose the severity type. For example, if you choose Show Severity > Severe, only messages with severity of type Severe would be shown.

You can also choose how error messages are to be displayed in the editor. This is available by selecting Messages, and then selecting one of three options: Insert None, Insert Selected Only, or Insert All.

Chapter 7. Running programs and commands

The Remote System Explorer is built so that you can run programs and commands on the iSeries server. The iSeries Commands Log view displays any returned messages as your applications are run. You can select a program or command and run it according to the type of application it invokes:

- As an iSeries Batch Application: This option uses the SBMJOB command to run your application in the batch subsystem that is specified in your user profile.
- As an iSeries Interactive Application: For this option, your application will run in the interactive job associated with the selected Remote System connection. If no interactive job is associated with the connection, you will be asked to use the STRRSESVR command to associate the interactive job with the specified connection.
- As an iSeries Application in the Remote System Explorer communication server job: This option lets you run the program in the same job as the communication server. This is useful if you use the communication server job to set up your environment, your library list for example, and you need your program to run in this environment.
- As an iSeries Multi-threaded Application: Use this option to run multi-threaded applications. It creates a BCI job in QUSRWRK and runs the application in that job. You can also use this option to run any batch application. This option is only available for objects of type *PGM.

There are two distinct ways of invoking the Run action, with and without prompt. If you choose the option without prompt, the selected object will be run in the selected type of job with default values. If your program or command requires parameters, you need to use the prompt version.

Note: When you choose the Run option without prompt, a launch configuration with default settings is created and saved temporarily with the name ″My iSeries program for run″. You can go into the Run Launch Configurations window, rename the launch configuration and save it for future use.

To run a program or command without prompt, select a program or command and do one of the following:

- In the Remote Systems view, choose Run As from the pop-up menu.
- In the iSeries Table view, choose Run As from the pop-up menu.
- From the workbench Run menu, choose Run As.
- From the down arrow next to the Run push button, choose Run As.

To invoke a Run Launch Configurations window (Prompt) for a specific program or command, select that program or command and do one of the following:

- In the Remote Systems view, choose Run (Prompt) from the pop-up menu.
- In the iSeries Table view, choose Run (Prompt) from the pop-up menu.
- From the workbench Run menu, choose Run....
- From the down arrow next to the Run push button, choose Run....

A Launch Configurations window will display, where you can specify parameters for your application.

Chapter 8. Debugging programs

While exploring iSeries program objects, jobs, and Java classes in the workbench, you can launch the Integrated iSeries Debugger to debug these items by simply right-clicking them and choosing from a pop-up menu item. In the Remote System Explorer perspective, you can start a debug session for a program object, job, or Java class. And, from a workbench project, you can start a debug session for a program object or Java class.

You can launch more than one debug session at the same time from the workbench. When you do this, the debugger user interface remains open while each program you are debugging is shown on a separately-tabbed program page.

Debugging your programs

The iSeries debugger enables you to detect and diagnose errors in code developed with OS/400 Integrated Language Environment (ILE) and Original Program Model (OPM) languages and Java. The debugger's client/server design makes it possible to debug applications running on an iSeries machine while the debugger user interface is running in the workbench IDE on a Windows workstation.

The iSeries debugger enables you to detect and diagnose errors in code developed with OS/400 Integrated Language Environment (ILE) and Original Program Model (OPM) languages and Java. The debugger's client/server design makes it possible to debug applications running on an iSeries machine while the debugger user interface is running in the workbench IDE on a Windows workstation.

When you start the debugger, you have the option of starting and debugging an application or debugging an existing job on the iSeries host. When starting an application for debugging, a job is created on the iSeries host for you.

The debugger client user interface allows you to control program execution. For example, you can run your program, set breakpoints, step through program instructions, examine variables, and examine the call stack. The debugger user interface lets you debug multiple applications, which may be written in different languages, from a single debugger window. Each session you debug is listed separately in the Debug view.

Service entry points

The service entry point feature is designed to allow easy debugging of web applications that invoke business logic written in RPG, COBOL, CL, or even C or C++. The service entry point is a special kind of entry breakpoint that can be set directly from the Remote System Explorer. It is triggered when the first line of a specified procedure is executed in a job that is not under debug. Service entry points allow you to gain control of your job at that point. A new debug session gets started and execution is stopped at that location.

While the original service entry point design was meant to assist with debugging web applications, using the feature is also beneficial when debugging applications in an environment where the job name is not known ahead of time. Some sample scenarios include:

- Debugging WebFacing applications: Typically you would want to debug your RPG/COBOL code that is driven by the JSP that is executing in the application server. You do not know the name of the job in which your RPG/COBOL program will run, ahead of time, so the use of service entry points is an ideal solution for this scenario.
- Debugging any application that uses the Toolbox program or service program calls: When a program or service program is called and the code where the service entry point is set is about to execute, the debugger can take control of the application. With this technique, you can put the program or service program invoked by the Toolbox under debug although you do not know which job it will be running in.
- If your application runs in a multi-job environment, service entry points will significantly simplify your life while debugging. You would set a service entry point in the program that is about to be spawned in a new job. When the program is spawned and the line where the service entry point is set is about to execute, the debugger will be able to gain control of the program and stop at that line.

When a service entry point is triggered and the job which is running the application is not currently under debug, a message that allows you to launch a debug session will open. If you select Yes in the message, a debug session for the program will be launched in the Debug perspective. If you select No, you will resume the program without debugging it. If you do not want to be prompted to start a debug session right away when a service entry point is hit, select the check box in the prompt dialog or use the debugger preferences settings. For further information about debug preferences, see the related topic in the online help.

Printed in USA