# Improving productivity for IBM eServer iSeries traditional applications

*Using WebSphere Remote System Explorer*

*Garry Kipfer*
*IBM eServer Solution Enablement*
*October 2005*

## Table of contents

## Abstract

*Remote System Explorer (RSE), a perspective within IBM® WebSphere® Development Studio Client, provides a visual interface for the developer. It is designed to give access to all of the resources required to develop and maintain a traditional green-screen application on a desktop. As such, developers can manage IBM® iSeries™ resources, as well as edit, maintain, compile, and debug code. This white paper focuses on introducing RSE's resource management capabilities, as well as editing and verification functions. Thus, this paper serves as an overview of the tool and its ability to deliver increased productivity to WebSphere application developers on the iSeries family.*

## Introduction

When modernizing an application, one of the first and most important decisions to make is which tools are best suited for the task.

Ideally, the tools will address most, if not all of the stages of the modernizing process. This greatly enhances productivity by giving a common user experience to each of the developer's activities, and it removes the frustrations involved in learning several  tool interfaces to address different requirements.

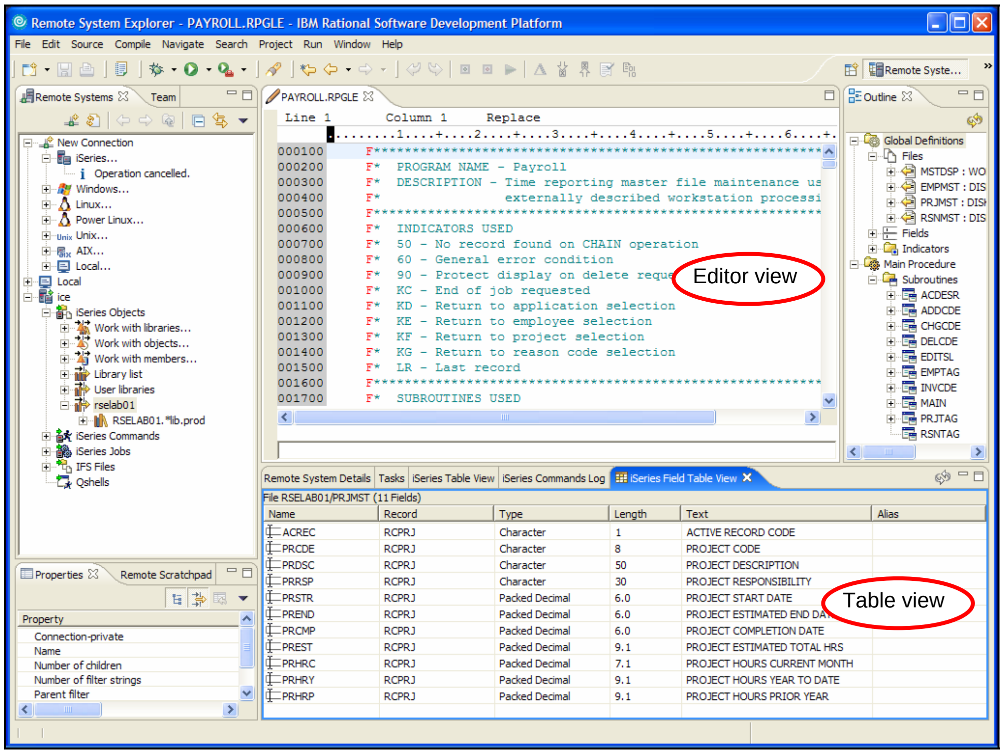
IBM WebSphere Development Studio Client has been designed to address these concerns, and it offers an integrated development environment (IDE) that can be used to build the various types of applications presented in the IBM eServer™ iSeries Developer Roadmap. This paper addresses the fist step in the modernizing process by discussing how the Remote System Explorer (RSE), a perspective within WebSphere Development Studio Client, provides a better and more productive tool for creating and maintaining traditional green-screen applications.

WebSphere Development Studio Client (IBM product number 5722-WDS) is composed of two parts. One part is installed on the iSeries system, and the other is installed on the developer's workstation. This paper discusses the workstation portion of the tool.

.

## Perspectives and views

As just mentioned, WebSphere Development Studio Client is composed of a number of perspectives, each designed to address a particular type of development effort by delivering access to the resources required for that task. It contains an RSE perspective that provides a visual interface to a more productive development environment for building traditional, green-screen applications.



```
Line 1        Column 1      Replace
000100    F***********************************************************
000200    F*  PROGRAM NAME - Payroll
000300    F*  DESCRIPTION - Time reporting master file maintenance us
000400    F*                externally described workstation processi
000500    F***********************************************************
000600    F*  INDICATORS USED
000700    F*  50 - No record found on CHAIN operation
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
001500    F*  LR - Last record
001600    F***********************************************************
001700    F*  SUBROUTINES USED
```

File RSELAB01/PRJMST (11 Fields)

| Name | Record | Type | Length | Text | Alias |
|---|---|---|---|---|---|
| ACREC | RCPRJ | Character | 1 | ACTIVE RECORD CODE | |
| PRCDE | RCPRJ | Character | 8 | PROJECT CODE | |
| PRDSC | RCPRJ | Character | 50 | PROJECT DESCRIPTION | |
| PRRSP | RCPRJ | Character | 30 | PROJECT RESPONSIBILITY | |
| PRSTR | RCPRJ | Packed Decimal | 6.0 | PROJECT START DATE | |
| PREND | RCPRJ | Packed Decimal | 6.0 | PROJECT ESTIMATED END DA | |
| PRCMP | RCPRJ | Packed Decimal | 6.0 | PROJECT COMPLETION DATE | |
| PREST | RCPRJ | Packed Decimal | 9.1 | PROJECT ESTIMATED TOTAL HRS | |
| PRHRC | RCPRJ | Packed Decimal | 7.1 | PROJECT HOURS CURRENT MONTH | |
| PRHRY | RCPRJ | Packed Decimal | 9.1 | PROJECT HOURS YEAR TO DATE | |
| PRHRP | RCPRJ | Packed Decimal | 9.1 | PROJECT HOURS PRIOR YEAR | |

***Figure 1: Default RSE perspective***

The default perspective for RSE, shown in Figure 1, contains several windows (called views). A view displays a resource and allows the developer to manage and manipulate that resource. The **Editor** view (top center of the screen capture in Figure 1) allows the developer to manipulate the source code. The **Table** view (bottom of the screen capture) shows the characteristics of the fields within a file. A perspective, which is essentially a collection of views, then defines the collection and the actions that are typically required for a particular environment. These perspectives can be modified, allowing the developer to include or exclude views as needed.

The RSE perspective has many functions to manage development resources, as well as a powerful LPEX (live parsing and extensible) editor. The following is a brief review of some of these features and how they are used together to deliver the integrated development environment (IDE).

## Managing resources

RSE provides a number of tools that deliver a structured approach to managing resources. One key element of the Remote System view for the developer is subsystems (not to be confused with iSeries subsystems).

## Subsystems

- The **Remote Systems** view (see screen capture in Figure 2) shows the following subsystems for various functions:
- iSeries objects
- iSeries commands
- iSeries jobs
- IFS files
- Qshells

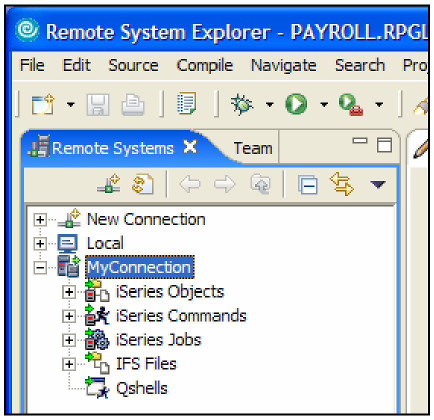


***Figure 2: Remote Systems view***

### iSeries objects

With this subsystem, the developer can create, copy, rename, delete, and retrieve information about objects, as well as compile and execute programs. Also, filters can be created for viewing specific objects. This is very similar to the iSeries **Work with libraries using Programming Development Manager** (**WRKLIBPDM**) CL command, except that filters can be saved with custom names and reused later. The developer can also drag-and-drop or cut-and-paste objects, subject to iSeries system rules, greatly reducing the time required to manipulate them.

### iSeries commands

This subsystem is used for executing iSeries commands that have been defined by the developer. It also allows for the manipulation of libraries, files, members, and library lists. The example in Figure 3 shows the options available within the subsystem.

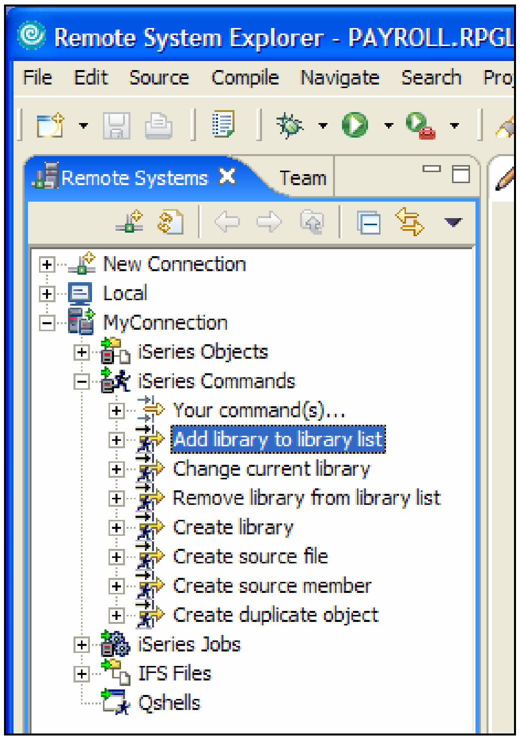
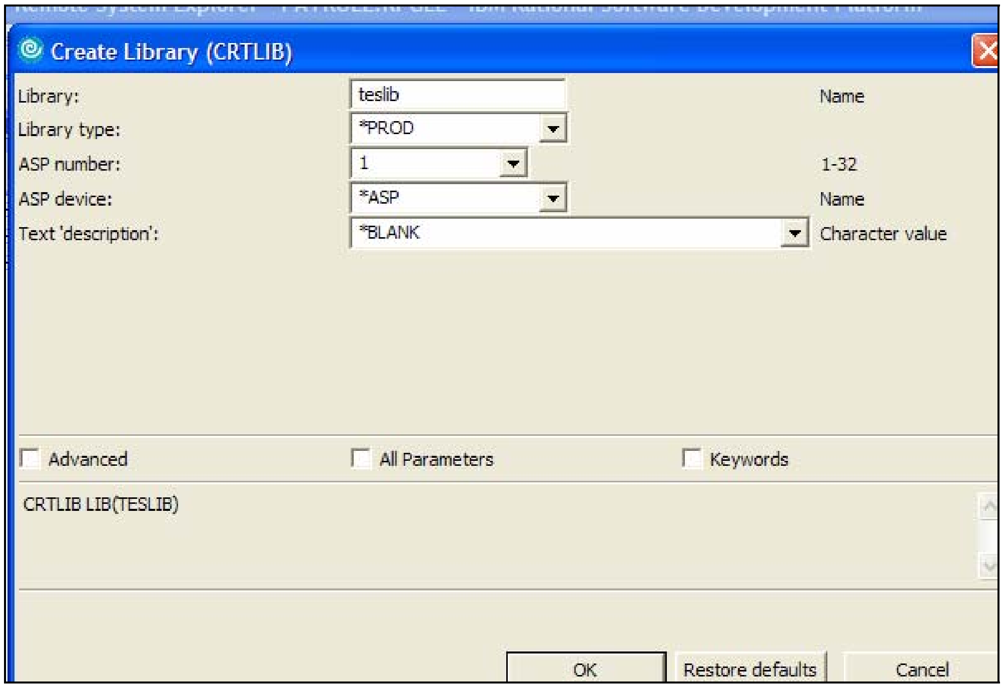


***Figure 3: Subsystem options***

Wizards are provided for each operation, as shown in Figure 4, for the iSeries **Create Library** (**CRTLIB**) CL command.



|  |  |  |
|---|---|---|
| Library: | teslib | Name |
| Library type: | *PROD |  |
| ASP number: | 1 | 1-32 |
| ASP device: | *ASP | Name |
| Text 'description': | *BLANK | Character value |

```
CRTLIB LIB(TESLIB)
```

*Figure 4: The Create Library (CRTLIB) CL command*

## iSeries jobs

This subsystem allows the developer to manage jobs on the iSeries system. These actions include the ability to hold, end, and debug jobs, as well as view logs.

## IFS files

This subsystem allows for the manipulation of the iSeries integrated file system (IFS). The developer can view, create, delete, move, rename, and open files and folders. The developer can also see properties. All of these functions, as shown in Figure 5, are accomplished with mouse actions as opposed to typing.

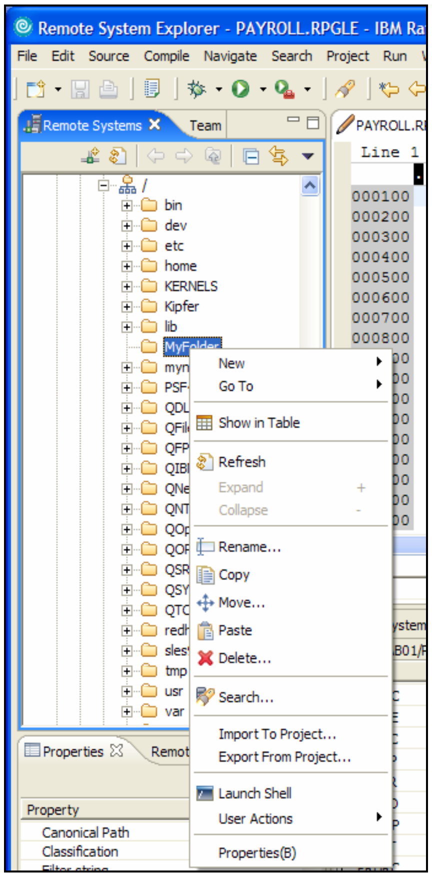


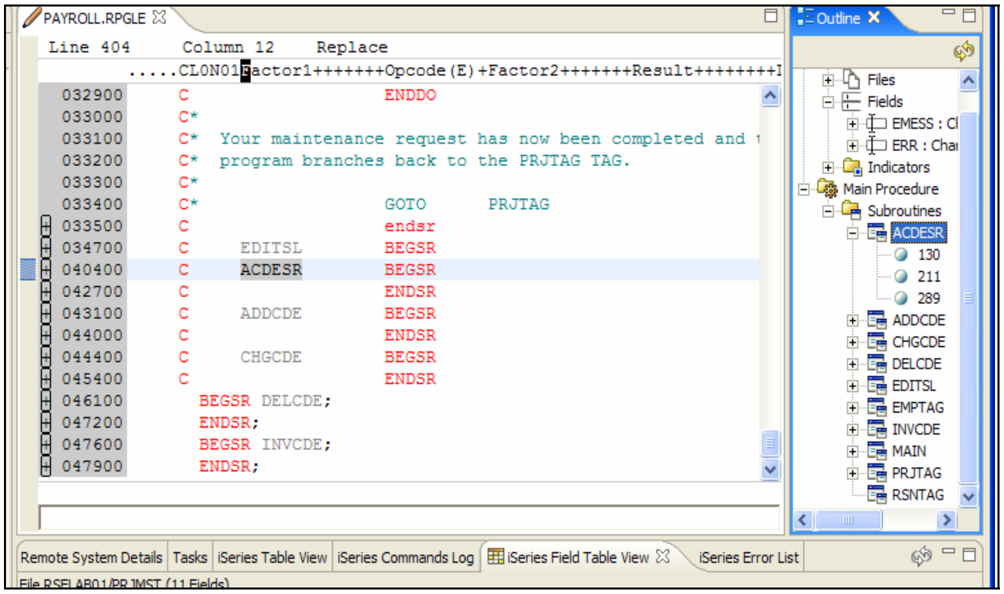
*Figure 5: IFS mouse actions*

## Editing and verifying the source code

RSE is extremely efficient for developing and maintaining application code. In this case, a number of views and actions work together. Two of the common views are the **Outline** view and the **LPEX Editor** view.

### Outline view

After the application program has been opened with the LPEX editor, and the outline view is refreshed, the information will be displayed as illustrated in Figure 6.



```
Line 404      Column 12    Replace
      .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++I
 032900     C                         ENDDO
 033000     C*
 033100     C*  Your maintenance request has now been completed and t
 033200     C*  program branches back to the PRJTAG TAG.
 033300     C*
 033400     C*                        GOTO      PRJTAG
 033500     C                         endsr
 034700     C        EDITSL           BEGSR
 040400     C        ACDESR           BEGSR
 042700     C                         ENDSR
 043100     C        ADDCDE           BEGSR
 044000     C                         ENDSR
 044400     C        CHGCDE           BEGSR
 045400     C                         ENDSR
 046100       BEGSR DELCDE;
 047200       ENDSR;
 047600       BEGSR INVCDE;
 047900       ENDSR;
```

***Figure 6: The outline view***

The **Outline** view displays the names of all files, fields, indicators, and subroutines that are used in the program. They are displayed in a tree structure, which, when expanded as shown in Figure 6, will also indicate the location where the item is defined and list every line of code that references it. The **Subroutines** tree structure (shown in the right panel of the screen capture in Figure 6) has been expanded to display the name of all subroutines that are used in the program. The structure for the **ACDESR** subroutine has also been expanded to display every line of code where it (the ACDESCR subroutine) is referenced. If the developer clicks any of these line numbers, the editor positions the cursor to that line of code. In Figure 6, the developer clicked **ACDESR** in the outline view, and the cursor is automatically positioned at the line of code where ACDESR is defined in the LPEX editor under the **PAYROLL** tab.

Compared to the typical search features that are currently available, this is a simple yet powerful productivity aid.

## LPEX Editor view

Remote System Explorer (RSE) contains the Remote Systems LPEX editor for editing source code directly in the workbench. The interface provides right-click actions for compiling and running programs. With the many functions available with this LPEX Editor, the developer can perform the following operations:

- Cut, copy, and paste operations.
- Block mark lines, characters, or rectangles with copy, move, and delete operations.
- Find and replace character strings quickly and easily.
- Undo and redo operations.
- Highlight various language constructs using different colors and fonts.
- Show the purpose of each column using a source entry utility (SEU)-like format line ruler that is automatically updated to reflect current specifications.
- Utilize SEU-like source line prompting for RPG and DDS source members.
- Employ sequence numbers that allow SEU-style commands in the prefix area.
- Tab between columns for column-sensitive languages.
- Automatically convert to uppercase for languages that expect it.
- Use commands to simplify text insertions and deletions for column-sensitive languages.
- Display specific source member content in the editor using filtered views.
- Display control structures for columnar languages.
- View field details of any database, workstation, or printer files referenced in the source member using a **show fields** function.
- Verify the source code.
- Receive online language reference help.
- Check the syntax for RPG, COBOL, or CL.

In addition, the editor can be customized to meet the developer’s requirements.

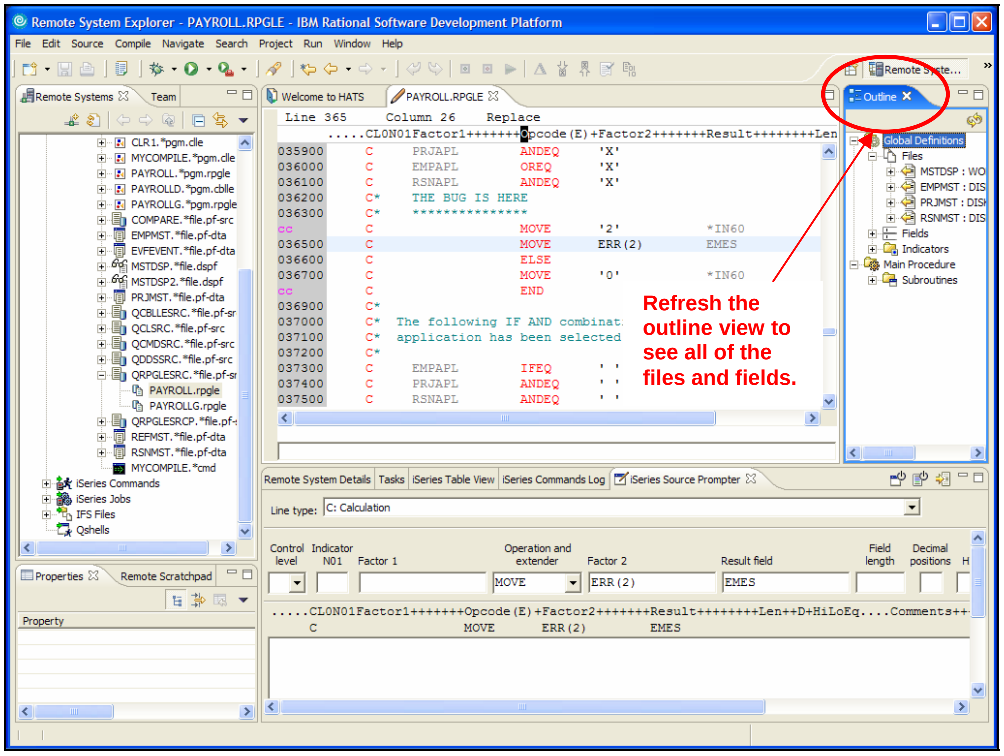
## PAYROLL example

Some of the more pertinent functions that impact the developer’s productivity and demonstrate the ease-of-use of the LPEX editor can be demonstrated through a sample RPG program (PAYROLL). Because PAYROLL contains several errors, it will be easy to illustrate LPEX editing features, such as its ability to integrate with the other views, to provide a complete development environment.

In Figure 7, the developer has clicked the program name in the **Remote Systems** view, and the program has been opened and displayed in the **PAYROLL** tab in the **Editor** view. The **Outline** view has been refreshed and all the files and fields have been displayed. In the **PAYROLL** tab of the **Editor** view, the SEU commands can be used for editing by the block copy designated by CC in the sequence number. Hence there are no new commands to learn. Also, the highlighted line contains the MOVE op code. The prompt key (F4) was pressed, and in the bottom panel, the **iSeries Source Prompter** tab displays the prompt required for this op code.

## Online manuals

In addition, if the help key (**F1**) is pressed, the RPG manual will open and display in a separate window (see Figure 7) to explain the **MOVE** op code. This removes the need to search printed manuals.



```
 Line 365      Column 26     Replace
       .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len
035900       C      PRJAPL          ANDEQ     'X'
036000       C      EMPAPL          OREQ      'X'
036100       C      RSNAPL          ANDEQ     'X'
036200       C*     THE BUG IS HERE
036300       C*     ***************
cc           C                      MOVE      '2'           *IN60
036500       C                      MOVE      ERR(2)        EMES
036600       C                      ELSE
036700       C                      MOVE      '0'           *IN60
cc           C                      END
036900       C*
037000       C*   The following IF AND combinat
037100       C*   application has been selected
037200       C*
037300       C      EMPAPL          IFEQ      ' '
037400       C      PRJAPL          ANDEQ     ' '
037500       C      RSNAPL          ANDEQ     ' '
```

Line type: C: Calculation

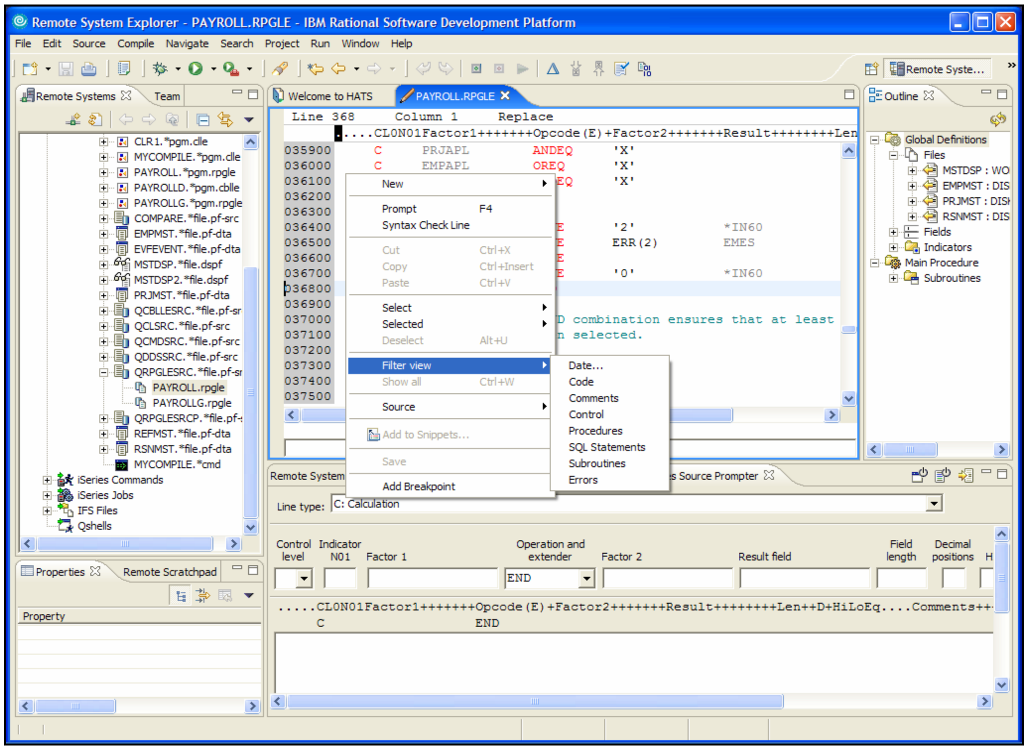
| Control level | Indicator N01 | Factor 1 | Operation and extender | Factor 2 | Result field | Field length | Decimal positions | H |
|---|---|---|---|---|---|---|---|---|
| | | | MOVE | ERR(2) | EMES | | | |

```
.....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len++D+HiLoEq....Comments++
     C                   MOVE      ERR(2)        EMES
```

*Figure 7: Remote Systems view and options*

## Filter views

Figure 8 shows what occurs if the developer right-clicks the **PAYROLL** tab window. The resulting pop-up menu offers a number of actions, such as **Prompt** and **Syntax Check Line**. In particular, this menu supplies **Filter view,** which as its name implies, allows the developer to filter the view by date, code, or whatever the situation requires. In this case, **Subroutines** was chosen.

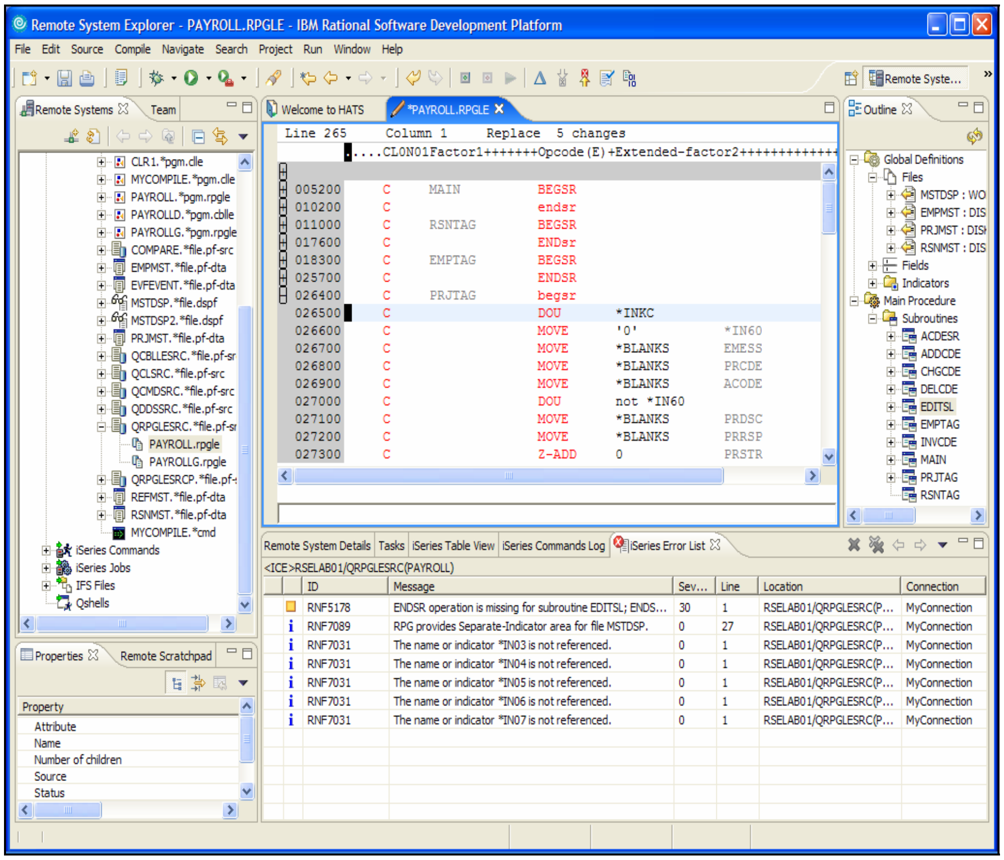


```
Line 368      Column 1      Replace
.....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len
035900     C      PRJAPL            ANDEQ     'X'
036000     C      EMPAPL            OREQ      'X'
```

Figure 8: PAYROLL tab options

## Subroutine views

Once the **Subroutine** folder is chosen (under the **Main Procedure** folder), the view changes and only the subroutines are displayed (one line for each). The complete subroutine can be displayed by clicking the **+** sign next to the folder. Refer to the illustration below in Figure 9 for an example.


Remote System Explorer - PAYROLL.RPGLE - IBM Rational Software Development Platform

```
Line 265     Column 1     Replace   5 changes
     .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2++++++++++++
005200     C     MAIN          BEGSR
010200     C                   endsr
011000     C     RSNTAG        BEGSR
017600     C                   ENDsr
018300     C     EMPTAG        BEGSR
025700     C                   ENDSR
026400     C     PRJTAG        begsr
026500     C                   DOU       *INKC
026600     C                   MOVE      '0'           *IN60
026700     C                   MOVE      *BLANKS       EMESS
026800     C                   MOVE      *BLANKS       PRCDE
026900     C                   MOVE      *BLANKS       ACODE
027000     C                   DOU       not *IN60
027100     C                   MOVE      *BLANKS       PRDSC
027200     C                   MOVE      *BLANKS       PRRSP
027300     C                   Z-ADD     0             PRSTR
```

<ICE>RSELAB01/QRPGLESRC(PAYROLL)

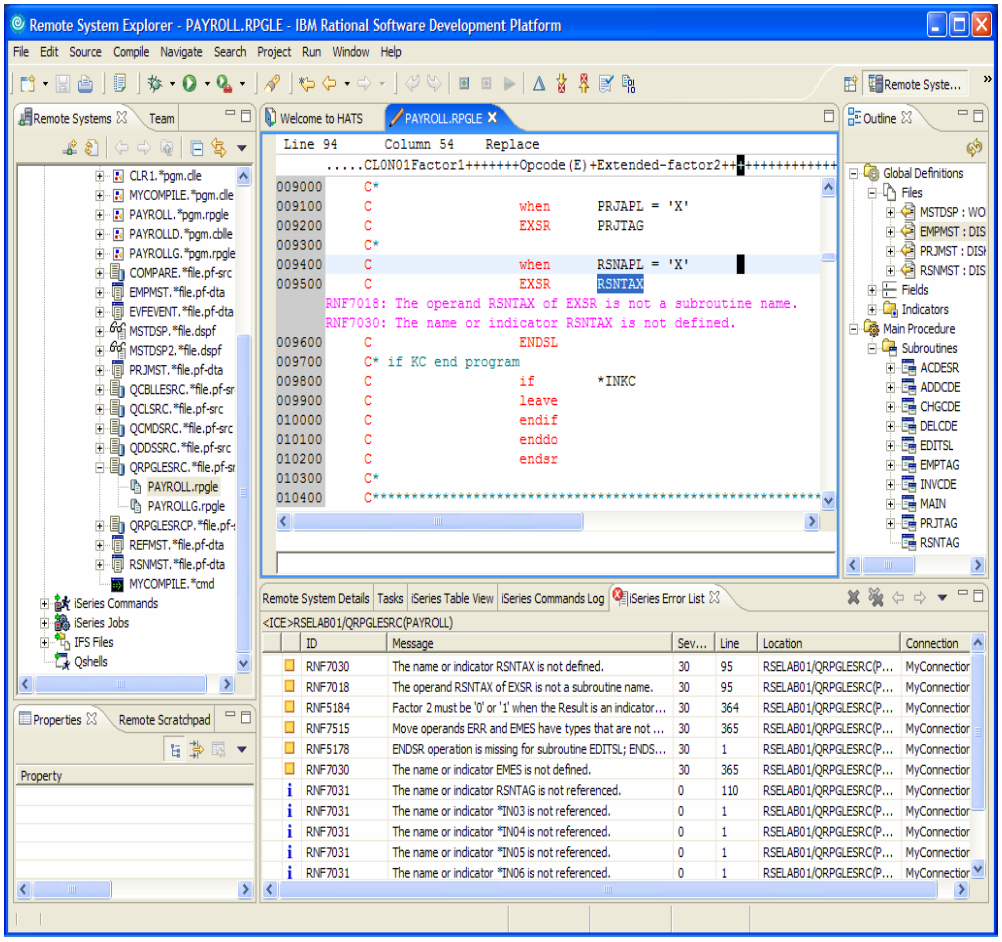
| | ID | Message | Sev... | Line | Location | Connection |
|---|---|---|---|---|---|---|
| 🟨 | RNF5178 | ENDSR operation is missing for subroutine EDITSL; ENDS... | 30 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| i | RNF7089 | RPG provides Separate-Indicator area for file MSTDSP. | 0 | 27 | RSELAB01/QRPGLESRC(P... | MyConnection |
| i | RNF7031 | The name or indicator *IN03 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| i | RNF7031 | The name or indicator *IN04 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| i | RNF7031 | The name or indicator *IN05 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| i | RNF7031 | The name or indicator *IN06 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| i | RNF7031 | The name or indicator *IN07 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |

*Figure 9: Subroutine tab views*

## Source verification

One of the most powerful features in RSE is the ability to verify source code on the developer’s workstation. In the example shown in Figure 10, the **PAYROLL** program was verified, and the results are displayed below.

A new tab in the bottom panel, called **iSeries error list**, is opened and all the compile time errors are listed there. In Figure 10, the RNF7030 error has been double-clicked. The cursor is then positioned at the line of code in error in the **PAYROLL** tab, and the error message text is displayed.


Remote System Explorer - PAYROLL.RPGLE - IBM Rational Software Development Platform
File Edit Source Compile Navigate Search Project Run Window Help
Remote Systems
Team
Welcome to HATS
PAYROLL.RPGLE
Outline
Properties
Remote Scratchpad
Property

```
Line 94       Column 54    Replace
     .....CL0N01Factor1+++++++Opcode(E)+Extended-factor2+++++++++++++++
009000      C*
009100      C                   when      PRJAPL = 'X'
009200      C                   EXSR      PRJTAG
009300      C*
009400      C                   when      RSNAPL = 'X'
009500      C                   EXSR      RSNTAX
     RNF7018: The operand RSNTAX of EXSR is not a subroutine name.
     RNF7030: The name or indicator RSNTAX is not defined.
009600      C                   ENDSL
009700      C* if KC end program
009800      C                   if        *INKC
009900      C                   leave
010000      C                   endif
010100      C                   enddo
010200      C                   endsr
010300      C*
010400      C****************************************************************
```

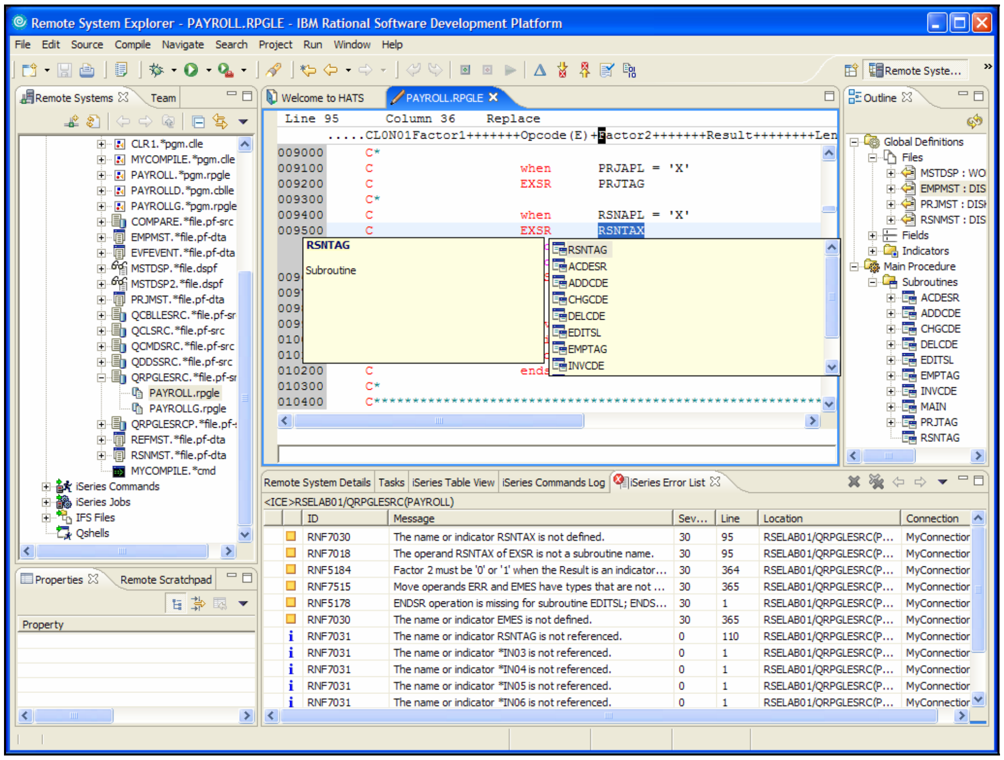
<ICE>RSELAB01/QRPGLESRC(PAYROLL)

| | ID | Message | Sev... | Line | Location | Connection |
|---|---|---|---|---|---|---|
| 🟧 | RNF7030 | The name or indicator RSNTAX is not defined. | 30 | 95 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| 🟧 | RNF7018 | The operand RSNTAX of EXSR is not a subroutine name. | 30 | 95 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| 🟧 | RNF5184 | Factor 2 must be '0' or '1' when the Result is an indicator... | 30 | 364 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| 🟧 | RNF7515 | Move operands ERR and EMES have types that are not ... | 30 | 365 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| 🟧 | RNF5178 | ENDSR operation is missing for subroutine EDITSL; ENDS... | 30 | 1 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| 🟧 | RNF7030 | The name or indicator EMES is not defined. | 30 | 365 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| i | RNF7031 | The name or indicator RSNTAG is not referenced. | 0 | 110 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| i | RNF7031 | The name or indicator *IN03 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| i | RNF7031 | The name or indicator *IN04 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| i | RNF7031 | The name or indicator *IN05 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnectior |
| i | RNF7031 | The name or indicator *IN06 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnectior |

*Figure 10: PAYROLL program verified and the iSeries error list*

## Error correction

Because the subroutine name is incorrect in this instance, the developer will position the cursor on the field and press **CTRL + spacebar**. The editor will then display all the subroutines as shown in Figure 11. The developer then clicks the correct name to place it in the source code. This is much simpler than the existing process of displaying the spool file, finding the error number and line number, and then the actual line of code.



```
 Line 95       Column 36     Replace
      .....CL0N01Factor1+++++++Opcode(E)+Factor2+++++++Result++++++++Len
009000       C*
009100       C                   when      PRJAPL = 'X'
009200       C                   EXSR      PRJTAG
009300       C*
009400       C                   when      RSNAPL = 'X'
009500       C                   EXSR      RSNTAX
010200       C                   ends
010300       C*
010400       C**********************************************************
```

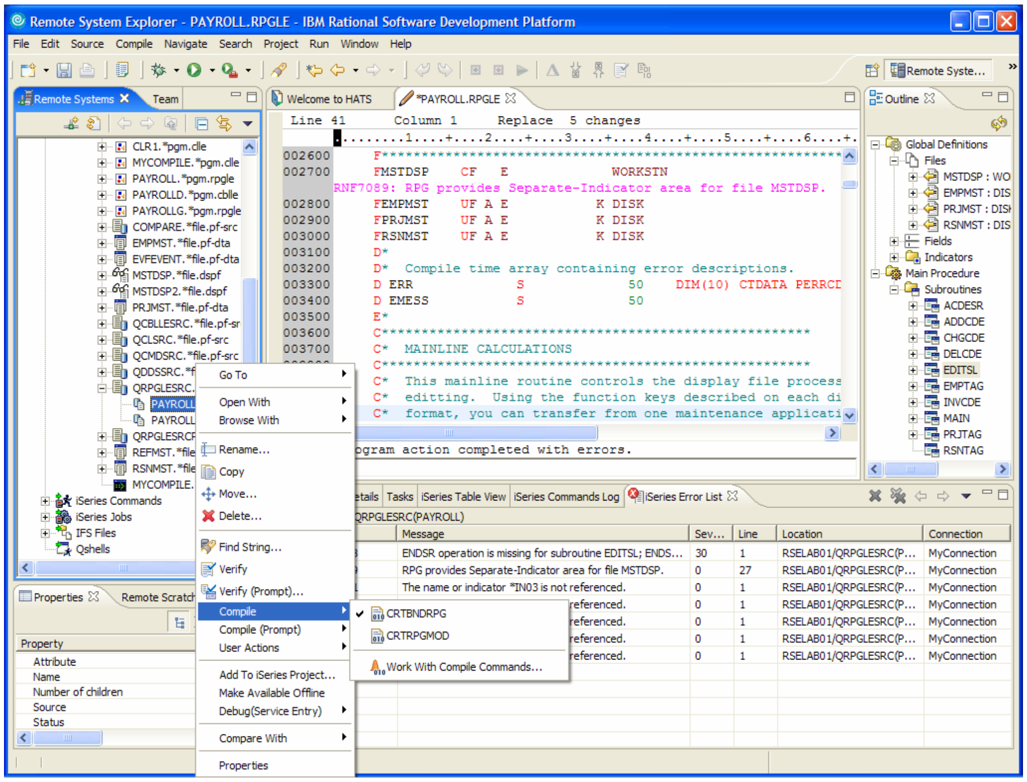
<ICE>RSELAB01/QRPGLESRC(PAYROLL)

| | ID | Message | Sev… | Line | Location | Connection |
|---|---|---|---|---|---|---|
| 🟧 | RNF7030 | The name or indicator RSNTAX is not defined. | 30 | 95 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| 🟧 | RNF7018 | The operand RSNTAX of EXSR is not a subroutine name. | 30 | 95 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| 🟧 | RNF5184 | Factor 2 must be '0' or '1' when the Result is an indicator… | 30 | 364 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| 🟧 | RNF7515 | Move operands ERR and EMES have types that are not … | 30 | 365 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| 🟧 | RNF5178 | ENDSR operation is missing for subroutine EDITSL; ENDS… | 30 | 1 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| 🟧 | RNF7030 | The name or indicator EMES is not defined. | 30 | 365 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| i | RNF7031 | The name or indicator RSNTAG is not referenced. | 0 | 110 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| i | RNF7031 | The name or indicator *IN03 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| i | RNF7031 | The name or indicator *IN04 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| i | RNF7031 | The name or indicator *IN05 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P… | MyConnectior |
| i | RNF7031 | The name or indicator *IN06 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P… | MyConnectior |

***Figure 11: Displaying subroutines***

## Compiling the program

Once all the errors have been corrected, the developer can click the program name to compile it. This is demonstrated in the following example (Figure 12):



```
 Line 41      Column 1      Replace  5 changes
       .........1....+....2....+....3....+....4....+....5....+....6....+.
002600      F*************************************************************
002700      FMSTDSP    CF   E             WORKSTN
       RNF7089: RPG provides Separate-Indicator area for file MSTDSP.
002800      FEMPMST    UF A E           K DISK
002900      FPRJMST    UF A E           K DISK
003000      FRSNMST    UF A E           K DISK
003100      D*
003200      D*  Compile time array containing error descriptions.
003300      D ERR             S             50    DIM(10) CTDATA PERRCD
003400      D EMESS           S             50
003500      E*
003600      C*******************************************************
003700      C*  MAINLINE CALCULATIONS
            C*******************************************************
            C*  This mainline routine controls the display file process
            C*  editting.  Using the function keys described on each di
            C*  format, you can transfer from one maintenance applicati
ogram action completed with errors.
```

QRPGLESRC(PAYROLL)

| Message | Sev... | Line | Location | Connection |
|---|---|---|---|---|
| ENDSR operation is missing for subroutine EDITSL; ENDS... | 30 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| RPG provides Separate-Indicator area for file MSTDSP. | 0 | 27 | RSELAB01/QRPGLESRC(P... | MyConnection |
| The name or indicator *IN03 is not referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |
| referenced. | 0 | 1 | RSELAB01/QRPGLESRC(P... | MyConnection |

*Figure 12: Compiling the program*

## Summary

Remote System Explorer (RSE), provided with WebSphere Development Studio Client, uses many familiar commands to help reduce its learning curve, while also offering powerful new functions with a visual interface. The integrated development environment allows further advantages.

Additional functions, such as the **Outline** tab view and the **Verify** function, combined with the LPEX editor, enhance the developer's productivity by greatly reducing the number of steps required to complete a task.

Although RSE takes some discipline to learn, after the initial learning period, developers are much more productive than in traditional programming environments. Programmers find there is often a temptation to rely on the older tools with which they are already familiar. If this temptation is resisted, however, developers will soon discover increased productivity and simplified maintenance for traditional applications. In addition, when developers have mastered WebSphere Development Studio Client and RSE they can use the same tool with the other modernizing technologies on the roadmap such as IBM WebFacing for iSeries, WebSphere Host Access Transformation Services (HATS), Java™, and WebSphere Portal development.

## Resources

For more information regarding Remote System Explorer, eServer, and the eServer i5 and iSeries family of computers, refer to the following links.

- WebSphere Development Studio Client for iSeries
  **ibm.com**/software/awdtools/wdt400
- **Training courses**
  – Web-based training course for WebSphere Development Studio Client
  **ibm.com**/developerworks/websphere/library/tutorials/dl/sw738/
- **IBM Redbooks™** (**ibm.com**/redbooks)
  – WebSphere Development Studio Client for iSeries V5.1.2 (SG24-6961)
- IBM eServer i5 Information Center
  http://publib.boulder.ibm.com/infocenter/iseries/v5r3/ic2924/index.htm
- IBM eServer p5 Information Center
  http://publib.boulder.ibm.com/infocenter/pseries/index.jsp
- IBM Publications Center
  www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US

## About the author

**Garry Kipfer**

*IBM eServer Solution Enablement*

Garry is a consulting IT specialist working in the Solutions Enablement group, Canada. He has more than 30 years experience in midrange systems with IBM. During that period, he has worked in traditional application development, client/server applications, data warehousing, application modernization, iSeries performance, and IBM DB2® Universal Database™ for iSeries.

## Trademarks and special notices