

## The ABCs of RDi shortcuts



**Charles Guarino**

**Twitter @charlieguarino**  
**Central Park Data Systems, Inc.**

## About The Speaker

With an IT career spanning over 30 years, Charles Guarino has been a consultant for most of them. Since 1995 he has been founder and President of Central Park Data Systems, Inc., a New York area based IBM midrange consulting and corporate training company. In addition to being a professional speaker across the United States and Europe, he is a frequent contributor of technical and strategic articles and webcasts for the IT community. He is a member of COMMON's Speaker Excellence Hall of Fame and also Long Island Software and Technology Network's Twenty Top Techies. In 2015 Charles became the recipient of the Al Barsa Memorial Scholarship Award. Additionally, he serves as a member of COMMON's Strategic Education Team (SET) and is also a past president and monthly Q&A host of LISUG, a Long Island IBM i User's Group [www.lisug.org](http://www.lisug.org).

Charles can be reached at [cguarino@centralparkdata.com](mailto:cguarino@centralparkdata.com).

LinkedIn - <http://www.linkedin.com/in/guarinocharles>

Twitter - @charlieguarino

## What We'll Cover ...

- Remote Systems Explorer
- LPEX Editor Shortcuts
- LPEX Commands
- RDi and Modernization
- Wrap-up

## Ctrl-F6 Jump to next editor

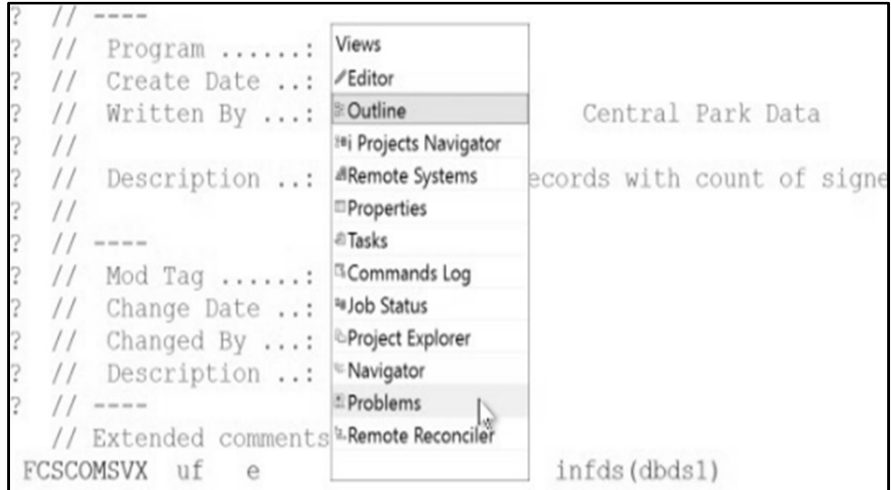
```
// Use contact tier information to add phone tier record
Exec SQL Select c15u20, c1ckey
           Into :c25u20, :c2ckey
           From zcontn a where c1ckey = :N_c3ckey;

c25u20 = c25u20
c2ckey = c2ckey
c2ptyp = N_c3ptyp
c2phno = N_c3phno
c2phex = N_c3phex
c2ausr = N_c3ausr
c2adte = %date;
c2atim = %time;

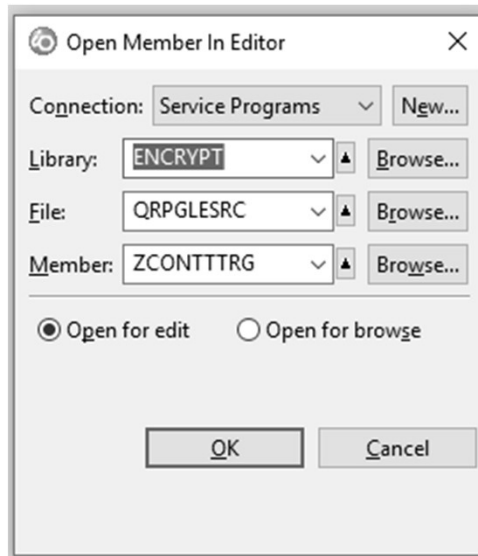
Exec SQL Insert Into
ZCONTR / c25u20 c2ckey c2ptyp c2phno c2phex
```



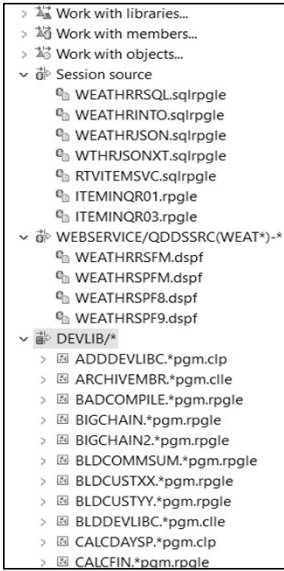
## Ctrl-F7 Switches between views



## Ctrl-Alt-Q Quick Member Open



## Right arrow to quickly close a filter

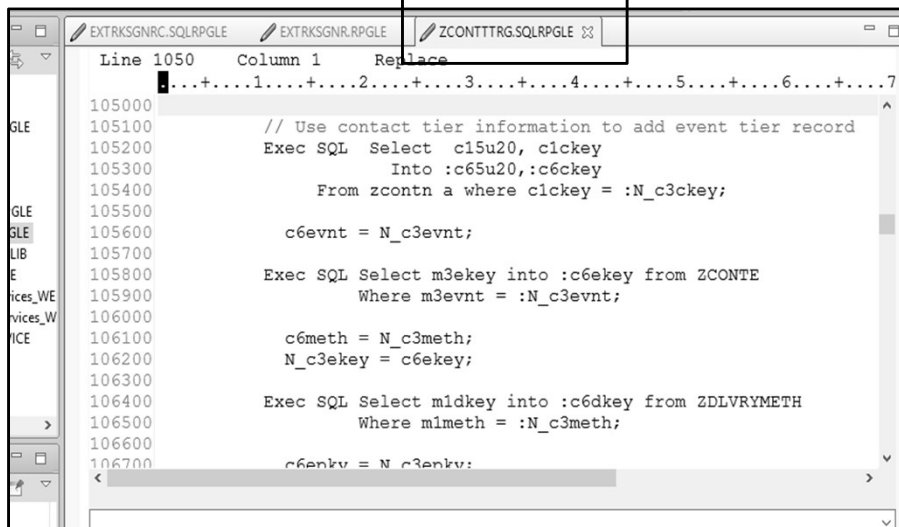


- > Work with libraries...
- > Work with members...
- > Work with objects...
- Session source
  - WEATHRRSQL.sqlrpgle
  - WEATHRINTO.sqlrpgle
  - WEATHRJJSON.sqlrpgle
  - WTHRJSONXT.sqlrpgle
  - RTVITEMSVC.sqlrpgle
  - ITEMINQR01.rpgle
  - ITEMINQR03.rpgle
- WEBSERVICE/QDDSSRC(WEAT\*)-\*
  - WEATHRRSFM.dspf
  - WEATHRSPFM.dspf
  - WEATHRSPF8.dspf
  - WEATHRSPF9.dspf
- DEVLIB/\*
  - ADDDEVLIBC.\*pgm.clp
  - ARCHIVEMBR.\*pgm.cle
  - BADCOMPILE.\*pgm.rpgle
  - BIGCHAIN.\*pgm.rpgle
  - BIGCHAIN2.\*pgm.rpgle
  - BLDCOMMSUM.\*pgm.rpgle
  - BLDCUSTXX.\*pgm.rpgle
  - BLDCUSTYY.\*pgm.rpgle
  - BLDDEVLIBC.\*pgm.cle
  - CALCDAYSP.\*pgm.clp
  - CALCFIN.\*pgm.rpgle

## F10 – Activate the menu options on the toolbar



## Ctrl-F4 (or CTRL-0) Quickly closes a member



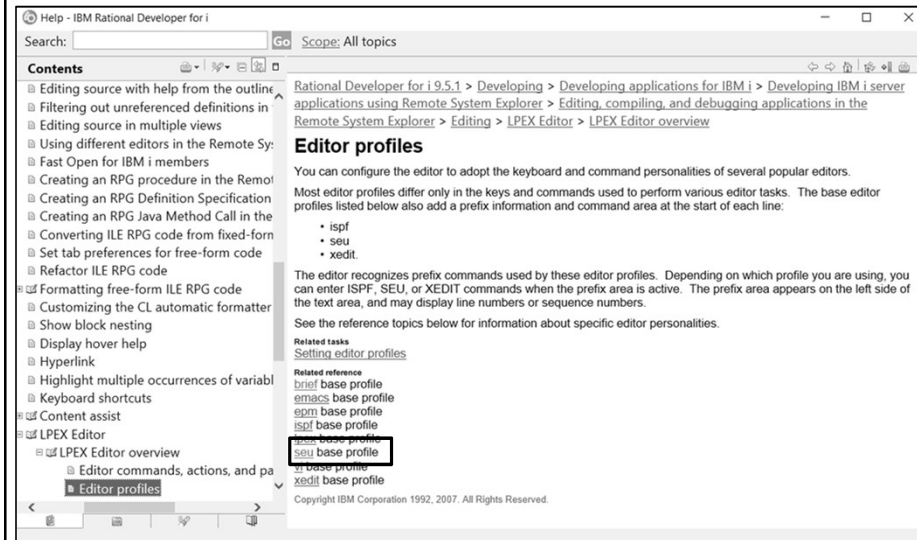
The screenshot shows a code editor window with three tabs: EXTRKSGNRC.SQLRPGLE, EXTRKSGNR.RPGLE, and ZCONTTTRG.SQLRPGLE. The ZCONTTTRG.SQLRPGLE tab is active and has a search bar highlighted with a red box. The search bar contains the text "...+...1...+...2...+...3...+...4...+...5...+...6...+...7". The code in the editor is as follows:

```
Line 1050 Column 1 Replace
...+...1...+...2...+...3...+...4...+...5...+...6...+...7
105000
105100 // Use contact tier information to add event tier record
105200 Exec SQL Select c15u20, c1ckey
105300 Into :c65u20,:c6ckey
105400 From zcontn a where c1ckey = :N_c3ckey;
105500
105600 c6evnt = N_c3evnt;
105700
105800 Exec SQL Select m3ekey into :c6ekey from ZCONTE
105900 Where m3evnt = :N_c3evnt;
106000
106100 c6meth = N_c3meth;
106200 N_c3ekey = c6ekey;
106300
106400 Exec SQL Select m1dkey into :c6dkey from ZDLVRYMETH
106500 Where m1meth = :N_c3meth;
106600
106700 c6enkv = N_c3enkv;
```

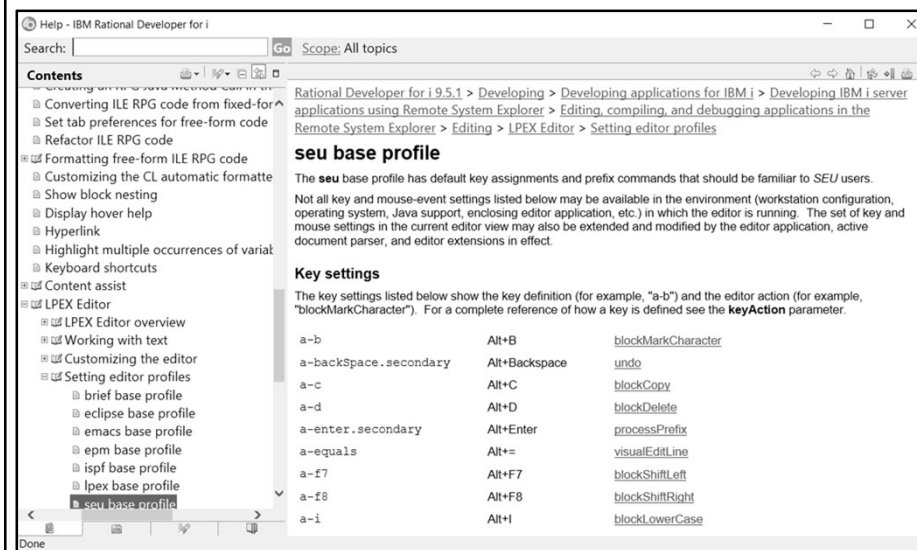
## What We'll Cover ...

- Remote Systems Explorer
- LPEX Editor Shortcuts
- LPEX Commands
- RDi and Modernization
- Wrap-up

# Editor Profiles



# Help SEU



## Display all current shortcuts – SHIFT-CTRL-L

```

Exec SQL Update ZCONTP
Set c2ptyp = :N_c3ptyp,
c2phno = :N_c3phno,
c2phex = :N_c3phex,
c2cusr = :N_c3user,
c2cdte = :c2cdte,
c2ctim = :c2ctim
Where c2pkey = :N_c3Pkey;

```

```
Endsr;
```

```
// *****
```

```
// *****
```

```
//          Delete Phone Tier
```

```
// *****
```

```
Begsr          DeletePhoneTier;
```

```
// Check to see if this is a phone tier delete request
```

```
if N_c3pkey > *zeros;
exec sql delete from zcontp where c2pkey = :N_c3pkey;
endif;
```

Find Previous	Ctrl+Shift+K
Find Text in Workspace	Ctrl+Alt+G
Find and Replace	Ctrl+F
Force Return	Alt+Shift+F
Format	Ctrl+Shift+F
Forward History	Alt+Right
Generate Code	Alt+Shift+G
Go to Line	Ctrl+L
Go to Matching Fence	Ctrl+Alt+P
Incremental Find	Ctrl+J
Incremental Find Reverse	Ctrl+Shift+J
Indent Selection	Ctrl+I
Inline	Alt+Shift+I
Insert Line Above Current Line	Ctrl+Shift+Enter
Insert Line Below Current Line	Shift+Enter
Inspect	Ctrl+Shift+I
Join Lines	Ctrl+Alt+J
Last Edit Location	Ctrl+Q
Line End	End
Line Start	Home
Maximize Active View or Editor	Ctrl+M

## Alt+Shift+Q to Show Views

Show System Menu	Alt+-
Show View	Alt+Shift+Q, Q
Show View ()	Alt+Shift+Q, D
Show View (Breakpoints)	Alt+Shift+Q, B
Show View (Cheat Sheets)	Alt+Shift+Q, H
Show View (Console)	Alt+Shift+Q, C
Show View (Error Log)	Alt+Shift+Q, L
Show View (History)	Alt+Shift+Q, Z
Show View (Javadoc)	Alt+Shift+Q, J
Show View (Outline)	Alt+Shift+Q, O
Show View (Package Explorer)	Alt+Shift+Q, P
Show View (Problems)	Alt+Shift+Q, X
Show View (Search)	Alt+Shift+Q, S
Show View (Synchronize)	Alt+Shift+Q, Y
Show View (Type Hierarchy)	Alt+Shift+Q, T
Show View (Variables)	Alt+Shift+Q, V
Show View Menu	Ctrl+F10
Surround With Quick Menu	Alt+Shift+Z
Switch to Editor	Ctrl+Shift+E
Terminate Result	Ctrl+Break
Toogle Breakpoint	Ctrl+Shift+B

## Position to beginning of line – Home button

```
270400 // *****  
270500 // Forgot Password  
270600 // *****  
270700 Dcl-Proc UpdateFGTPWD;  
270800 Dcl-Pi UpdateFGTPWD varchar(128);  
270900 End-Pi;  
271000  
271100 Dcl-s error varchar(128);  
271200  
271300 // This is a place holder routine pending API to initiate a forgotten  
271400 // password reset  
271500 N_c3stat = 'In Process';  
271600 N_c3cpl = '0';  
271700 TKeyN = N_c3tkey;  
271800 TKey = %editc(TKeyN:'X');  
271900 // Cmd = 'SBMJOB CMD(CALL ASYNC2CL PARM('' + TKEY + ''))';  
272000 Cmd = 'CALL ASYNCCL PARM('' + TKEY + ''))';  
272100 CmdLen = %len(%trim(Cmd));  
272200 Qcmdexec(Cmd:CmdLen);  
272300  
272400 error = *blanks;  
272500  
272600 return error;  
272700  
272800 End-Proc;
```

## Position to end of line – End button

```
// *****  
// Forgot Password  
// *****  
Dcl-Proc UpdateFGTPWD;  
Dcl-Pi UpdateFGTPWD varchar(128);  
End-Pi;  
  
Dcl-s error varchar(128);  
  
// This is a place holder routine pending API to initiate a forgotten  
// password reset  
N_c3stat = 'In Process';  
N_c3cpl = '0';  
TKeyN = N_c3tkey;  
TKey = %editc(TKeyN:'X');  
// Cmd = 'SBMJOB CMD(CALL ASYNC2CL PARM('' + TKEY + ''))';  
Cmd = 'CALL ASYNCCL PARM('' + TKEY + ''))';  
CmdLen = %len(%trim(Cmd));  
Qcmdexec(Cmd:CmdLen);  
  
error = *blanks;  
  
return error;  
  
End-Proc;
```



## Jump to top of source – CTRL-Home

```
Line 1      Column 1      Replace 20 changes
.....1.....2.....3.....4.....5.....6.....7.....+
000100  ctl-opt option(*nodebugio) dftactgrp(*no) actgrp(*caller)
000200      usrprf(*owner) alwnull(*usrctl);
000300
000301      dcl-f custmast disk(*ext) keyed usage(*update);
000302      dcl-f itemmastn keyed;
000303
000400      // The following data structures are used to provide field attributes
000500      // File I/O done through SQL
000600
000700      // Contact Tier (Primary level)
000800  D zcontNds      e ds      extname(zcontN)
000900
001000      // Phone tier (Child of contact)
001100  D zcontPds      e ds      extname(zcontP)
001200
001300      // e-Mail tier (Child of contact)
001400  D zcontEMLds    e ds      extname(zcontEML)
001500
001600      // Communication events tier (Child of Contact)
001700  D zEVENTCOMMds  e ds      extname(zEVENTCOMM)
001800
001900      // Attribute tier (Child of Contact)
002000  D zcontATTRds   e ds      extname(zcontATTR)
```

## Jump to bottom of source – CTRL-End

```
Line 2779   Column 81   Replace 20 changes
.....1.....2.....3.....4.....5.....6.....7.....
269900
270000      End-Proc;
270100
270200      // *****
270300
270400      // *****
270500      //          F o r g o t   P a s s w o r d
270600      // *****
270700  Dcl-Proc UpdateFGTPWD;
270800      Dcl-Pi UpdateFGTPWD varchar(128);
270900      End-Pi;
271000
271100      Dcl-s error varchar(128);
271200
271300      // This is a place holder routine pending API to initiate a forg
271400      // password reset
271500      N_c3stat = 'In Process';
271600      N_c3cmpl = '0';
271700      TKeyN = N_c3tkey;
271800      TKey = %editc(TKeyN:'X');
271900      // Cmd = 'SBMJOB CMD(CALL ASYNCL PARM('' + TKEY + ''))';
272000      Cmd = 'CALL ASYNCL PARM('' + TKEY + ''))';
272100      CmdLen = %len(%trim(Cmd));
272200      Qcmdexec(Cmd:CmdLen);
272300
272400      error = *blanks;
272500
272600      return error;
272700
272800      End-Proc;
272900
273000      // *****
```

## Ctrl-A to Select All source

```
Line 29      Column 91      Replace 4 changes
.....1.....2.....3.....4.....5.....6.....7.....8.....
000600      //      items found in the item master table.
000700
000800      //      Note - If you wanted to have a 'conversation' and continue
000900      //      returning each subsequent set of 200 you would need to
001000      //      receive the last item sent to the consumer and use
001100      //      that as your starting point.
001200
001300      //      Data structure webordersdata's only purpose is to define table fields.
001400      dcl-ds webordersdata extname('WEBITEMS') template end-ds;
001500
001600      //      If you want to return more than 200 items, just change the value here.
001700      dcl-c maxelements const(200);
001800
001900      dcl-ds itemlistsds qualified;
002000      itemnumber like(acitemno);
002100      description like(acitemdes);
002200      unitofmeasure like(acunitofms);
002300      itemclass like(acitemclas);
002500      end-ds;
002600
002700
002800      //      Subprocedure getitems starts here.
002900      //      Since this is a local subprocedure we don't need a prototype.
003000
003100      dcl-proc getitems export;
003200
003300      dcl-pi *n;
003400      itemclassin char(2) const;
003500      success char(5);
003600      classdesc char(30);
003700      numberofitems int(10);
```

## Alt-U to UnSelect All source

```
37      Column 89      Replace 4 changes
.....1.....2.....3.....4.....5.....6.....7.....8.....
//      items found in the item master table.

//      Note - If you wanted to have a 'conversation' and continue
//      returning each subsequent set of 200 you would need to
//      receive the last item sent to the consumer and use
//      that as your starting point.

//      Data structure webordersdata's only purpose is to define table fields.
dcl-ds webordersdata extname('WEBITEMS') template end-ds;

//      If you want to return more than 200 items, just change the value here.
dcl-c maxelements const(200);

dcl-ds itemlistsds qualified;
itemnumber like(acitemno);
description like(acitemdes);
unitofmeasure like(acunitofms);
itemclass like(acitemclas);

end-ds;

//      Subprocedure getitems starts here.
//      Since this is a local subprocedure we don't need a prototype.

dcl-proc getitems export;

dcl-pi *n;
itemclassin char(2) const;
success char(5);
classdesc char(30);
numberofitems int(10);
```

## Block select – CTRL-M

```
? // Check if contact file has record already if so skip add
Exec SQL Select c15u20, clickey
        Into :c15u20,:clickey
        From zcontn a where clickey = :N_c3ckey;

if sqlcode = 100;
    clickey = N_c3ckey;
    c15u20 = N_c35u20;
    cljobt = N_c3jobt;
    cllong = N_c3long;
    clfrnm = N_c3frnm;
    cl1snm = N_c31snm;
    cl1sfx = N_c31sfx;
    clausr = N_c3user;
    cladte = %date;
    clatim = %time;
Exec SQL Insert Into
ZCONTN ( c15u20, cljobt, cllong, clfrnm, cl1snm,
        c11sfx, clausr, cladte, clatim, clickey)
Values (:c15u20,:cljobt,:cllong,:clfrnm,:cl1snm,
```

## Jump to end of block – CTRL-Shift-M

```
        From zcontn a
        where

        c15u20 = :c15u20
        and cljobt = :cljobt
        and cllong = :cllong
        and clfrnm = :clfrnm
        and cl1snm = :cl1snm
        and cl1sfx = :cl1sfx
        and clausr = :clausr
        and cladte = :cladte
        and clatim = :clatim;

endif; // End of SQLCODE = 100

// When we add a new contact record to ZCONTN we need to insert a
// ZCONTR which records the relationship between customer number
```

## Select an individual line – ALT-L

```
.c1sfx, :c1usr, :c1dte, :c1ctm, :c1ckey),  
  
// Get the key value of the record just inserted to pass back in  
// transaction file record
```

```
Exec SQL
```

```
  Select Max(c1ckey)  
  Into :N_c3ckey  
  From zcontn a  
  where
```

```
  c15u20 = :c15u20  
and c1jobt = :c1jobt  
and c1long = :c1long  
and c1frnm = :c1frnm  
and c1lsnm = :c1lsnm  
and c1lsfx = :c1lsfx  
and c1ausr = :c1ausr  
and c1adte = :c1adte  
and c1atim = :c1atim;
```

## Or select a block of code – ALT-L 2X

```
// Get the key value of the record just inserted to pass back in  
// transaction file record
```

```
Exec SQL
```

```
  Select Max(c1ckey)  
  Into :N_c3ckey  
  From zcontn a  
  where
```

```
  c15u20 = :c15u20  
and c1jobt = :c1jobt  
and c1long = :c1long  
and c1frnm = :c1frnm  
and c1lsnm = :c1lsnm  
and c1lsfx = :c1lsfx  
and c1ausr = :c1ausr  
and c1adte = :c1adte  
and c1atim = :c1atim;
```

```
endif; // End of SQLCODE = 100
```

## Block lowercase – ALT-I

```
// Delete attribute tier
Exsr DeleteAttrTier;

// delete event tier
exsr deleteeventtier;

// if n_c3udi = 'r' this is a remove
when n_c3udi = 'r';

// if a remove is requested it only applies to the relationship record
// in zcontr. this allows for reassigning contacts to different custom
// since a single contact can be assigned to multiple customers

// remove (delete) zcontr record
exsr removezcontr;

// #####
// begin new action codes for individual insertions
// #####

// *****
// Create transaction codes
// *****
```

## Block uppercase – ALT-K

```
// Delete attribute tier
Exsr DeleteAttrTier;

// DELETE EVENT TIER
EXSR DELETEEVENTTIER;

// IF N_C3UDI = 'R' THIS IS A REMOVE
WHEN N_C3UDI = 'R';

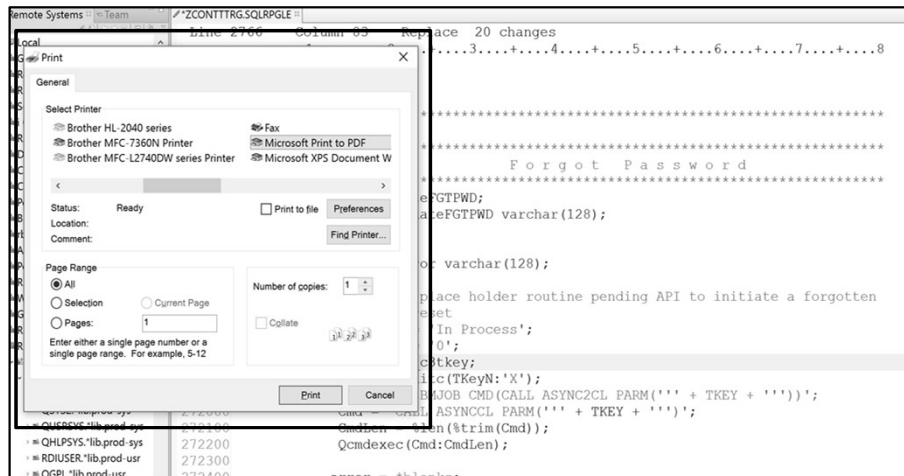
// IF A REMOVE IS REQUESTED IT ONLY APPLIES TO THE RELATIONSHIP RECORD
// IN ZCONTR. THIS ALLOWS FOR REASSIGING CONTACTS TO DIFFERENT CUSTOM
// SINCE A SINGLE CONTACT CAN BE ASSIGNED TO MULTIPLE CUSTOMERS

// REMOVE (DELETE) ZCONTR RECORD
EXSR REMOVEZCONTR;

// #####
// BEGIN NEW ACTION CODES FOR INDIVIDUAL INSERTIONS
// #####

// *****
// Create transaction codes
// *****
```

## Print a PDF of your source code – CTRL-P



## Undo / Redo CTRL-Z, CTRL-Y

```
002900 // Since this is a local subprocedure we don't need a prototype.
003000
003100 dcl-proc getitems export;
003200
003300 dcl-pi *n;
003400 itemclassin char(2) const;
003500 success char(5);
003600 classdesc char(30);
003700 numberofitems int(10);
003800 itemlist likeds(itemlistds)
003900 dim(maxelements);
004000 end-pi;
004100
004200 dcl-s maxfetch packed(3:0) inz(maxelements);
004300
004400
dd // Reset important indicator fields, this program is stateless
004600 numberofitems = *zero;
004700 success = 'False';
dd
004900 // Retrieve item class description from master table
005000 exec sql
005100 select ahclasdesc into :classdesc
005200 from webclsdes
005300 where ahitemclas = :itemclassin;
```

## Double click - Quickly select a word

```
// Since this is a local subprocedure we don't need a prototype

dcl-proc getitems      export;

dcl-pi *n;
itemclassin          char(2)  const;
success              char(5);
classdesc            char(30);
numberofitems       int(10);
itemlist             likeds(itemlistds)
                    dim(maxelements);

end-pi;

dcl-s maxfetch packed(3:0) inz(maxelements);

// Reset important indicator fields, this program is stateless
numberofitems = *zero;
success = 'False';

// Retrieve item class description from master table
exec sql
select ahclassdesc into :classdesc
from webclsdes
where ahitemclas = :itemclassin;
```

## Insert / Replace toggle

Column 73  Replace 4 changes  
.....1.....2.....3.....4.....5.....6.....7.....

```
// Since this is a local subprocedure we don't need a prototype.

dcl-proc getitems      export;

dcl-pi *n;
itemclassin          char(2)  const;
success              char(5);
classdesc            char(30);
numberofitems       int(10);
itemlist             likeds(itemlistds)
                    dim(maxelements);

end-pi;
```

Column 73  Insert 4 changes  
.....1.....2.....3.....4.....5.....6.....7.....

```
// Since this is a local subprocedure we don't need a prototype.

dcl-proc getitems      export;

dcl-pi *n;
itemclassin          char(2)  const;
success              char(5);
classdesc            char(30);
numberofitems       int(10);
itemlist             likeds(itemlistds)
                    dim(maxelements);

end-pi;
```

## Ctrl-D Instantly duplicate a line

```
dcl-proc getitems      export;

dcl-pi *n;
itemclassin          char(2)  const;
success              char(5);
classdesc            char(30);
numberofitems       int(10);
itemlist             likeds(itemlistds)
                    dim(maxelements);

end-pi;
```

```
dcl-s maxfetch packed(3:0) inz(maxelements);
dcl-s maxfetch packed(3:0) inz(maxelements);|
```

## Ctrl-Backspace Instantly delete a line

```
dcl-proc getitems      export;

dcl-pi *n;
itemclassin          char(2)  const;
success              char(5);
classdesc            char(30);
numberofitems       int(10);
itemlist             likeds(itemlistds)
                    dim(maxelements);

end-pi;
```

```
dcl-s maxfetch packed(3:0) inz(maxelements);
```



## Ctrl-DELETE Partially delete a line

```
dcl-proc getitems      export;

dcl-pi *n;
itemclassin          char(2)  const;
success              char(5);
classdesc            char(30);
numberofitems        int(10);
itemlist              likeds(itemlistds)
                    dim(maxelements);

end-pi;

dcl-s maxfetch        packed(3:0) . . .
```

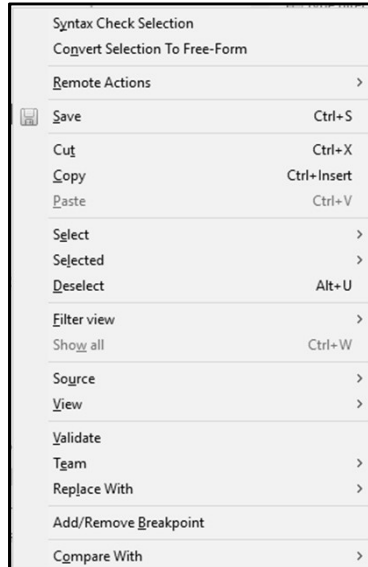
## Ctrl-ENTER Instantly Insert a Blank Line

```
//  UPDATTR will update an attribute tier and only an attribute tier.
//  If N_c3UDI = 'UPDATTR' this is a update of an attribute tier
When N_c3UDI = 'UPDATTR';

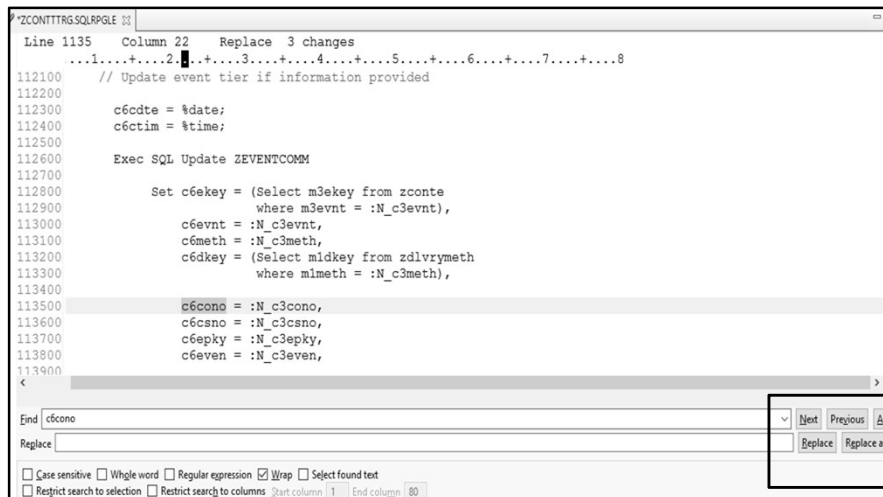
    N_c3errm = UpdateUPDATTR;

//  UPDEVENT will update an event tier and only an event tier.
//  If N_c3UDI = 'UPDEVENT' this is a update of an event tier
When N_c3UDI = 'UPDEVENT';
```

## Right click or Shift-F10



## Ctrl-F to find a string



## Ctrl-Alt-F to find \*ALL occurrences of a string

```
*ZCONTTRG.SQLRPGLE
Line 1135 Column 28 Replace 3 changes
...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
*
* 107700      c6cono = N_c3cono;
* 108700      ZEVENTCOMM ( c6cono, c6csno, c65u20, c6ckey, c6ekey, c6evnt, c6dkey,
* 108900      Values      (:c6cono, :c6csno, :c65u20, :c6ckey, :c6ekey, :c6evnt, :c6dkey,
* 109900      and c6cono = :c6cono
* 113500      c6cono = :N_c3cono,
* 230900      From zeventcomm where c6cono = :N_c3cono and
* 232400      c6cono = N_c3cono;
* 233400      ZEVENTCOMM ( c6cono, c6csno, c65u20, c6ckey, c6ekey, c6evnt, c6dkey,
* 233600      Values      (:c6cono, :c6csno, :c65u20, :c6ckey, :c6ekey, :c6evnt, :c6dkey,
* 235300      and c6cono = :c6cono
* 242400      c6cono = :N_c3cono,
```

## Ctrl-Shift-F to Reformat

### Before

```
if a = 6;
if b = 5;
select;
when customer = 123;
b = 4;
when customer = 456;
b = 7;
other;
b = 10;
endsl;
endif;
else;
a = 11233;
endif;
```

### After

```
if a = 6;
  if b = 5;
    select;
    when customer = 123;
      b = 4;
    when customer = 456;
      b = 7;
    other;
      b = 10;
    endsl;
  endif;
else;
  a = 11233;
endif;
```

## Alt-S to Split a line

**Before**

```
if c6epky = *zeros;  
  Exec SQL Select c2pkey into :c6epky from ZCONTP a  
    Where c25u20 = :c65u20  
    order by rrn(a) desc Fetch first row only;p  
Endif;
```

**After  
Box  
Checked**

```
if c6epky = *zeros;  
  Exec SQL Select c2pkey into :c6epky  
    from ZCONTP a  
    Where c25u20 = :c65u20  
    order by rrn(a) desc Fetch first row only;p  
Endif;
```

**After  
Box  
Unchecked**

```
if c6epky = *zeros;  
  Exec SQL Select c2pkey into :c6epky|  
    from ZCONTP a  
    Where c25u20 = :c65u20  
    order by rrn(a) desc Fetch first row only;p  
Endif;
```

## RPG Preferences for Alt-S

### ILE RPG

- Automatic syntax checking
  - Automatic syntax checking of SQL statements (performed on the server)
- Automatic uppercasing
- Show lowercase values in dropdown menus in Source Prompter
- Alt-S action splits the line and shifts the text on the new line

#### Enter key behavior

- Repeat previous specification type
- Repeat previous operation code (if on fixed form C-Specification)
- Set C-Specification cursor position

Position

FACTOR1 ▾

Preferences>RPG>ILE RPG

## Ctrl-J Returns to last line changed

```
        xmlstring = %scanrpl(stalltag:'<x':xmlstring);
        xmlstring = %scanrpl(eldtag:'</x':xmlstring);
    endfor;

// Prep table to hold parsed XML data
// Delete all rows

    exec sql
        delete from weatherdata;

// Call the webservice, parse and insert rows in WEATHRDATA
    processXML(xmlstring);
    *inlr = *on;
    return;

//Convert the xml data to row data and insert into weatherdata
dcl-proc processXML;

    dcl-pi *n ;
        XmlParm char(32766);
    end-pi;

    exec sql
```

## Ctrl-left and Ctrl-right to advance through words

```
// The JSON to XML routine sometimes creates invalid XML tags.
// Next to "fix" the new XML data, XML tag values cannot start with a numeric
// This loop will replace any starting number with lower case "x"
    for ctr = 1 to 10;
        starttag = '<' + %char(ctr-1);
        endtag = '</' + %char(ctr-1);
        xmlstring = %scanrpl(stalltag:'<x':xmlstring);
        xmlstring = %scanrpl(eldtag:'</x':xmlstring);
    endfor;

// Prep table to hold parsed XML data
// Delete all rows
```

## Ctrl-/ and \ Comment and un-comment a line of code

```
Begsr          UpdateAttrTier;

// Update attribute tier if information provided
// c5cdte = %date;
c5ctim = %time;

Exec SQL Update ZCONTATTR
Set c5attr = :N_c3attr,
  c5akey = (Select m2akey from zattrmst
            Where m2attr = :N_c3attr),
  c5cono = :N_c3cono,
  c5csno = :N_c3csno,
  c5cusr = :N_c3user,
  c5cdte = :c5cdte,
  c5ctim = :c5ctim
Where c5rkey = :N_c3Rkey;
```

## Ctrl “/” or “\” Comment and un-comment a block to text

```
// *****
//          Update Attribute Tier
// *****
Begsr          UpdateAttrTier;

// Update attribute tier if information provided
// c5cdte = %date;
// c5ctim = %time;

Exec SQL Update ZCONTATTR
Set c5attr = :N_c3attr,
  c5akey = (Select m2akey from zattrmst
            Where m2attr = :N_c3attr),
  c5cono = :N_c3cono,
  c5csno = :N_c3csno,
  c5cusr = :N_c3user,
```

## Shift-Ctrl-D Quick search by date navigation

The screenshot displays two windows of search results in an IDE. The top window shows search results for the range 150901 to 150930, with lines 031700 through 032300 highlighted. The bottom window shows search results for the range 150901 to 150930, with lines 127000 through 129400 highlighted. Both windows include search filters and navigation buttons.

```

031700 15/09/21
031800 16/11/21 // If N_c3UDI = 'D' this is a delete
031900 16/11/21 // When N_c3UDI = 'D';
032000 15/09/21
032100 16/11/21 // If a delete is requested then we must call all delete
032200 15/09/21 // subroutines because there is a potential to delete multiple tiers
032300 16/11/21 // in a single transaction

```

Find From (YMMDD): 150901  
To (YMMDD): 150930  
Equal to Less than Greater than **Range** Wrap

```

127000 15/09/21
127100 15/09/21 // When we add a new contact record to ZCONTN we need to insert a record into
127200 15/09/21 // ZCONTR which records the relationship between customer number and or link
127400 15/09/21 // We will only do this if a customer number is supplied on the transaction.
127500 15/09/21 // We will also only add the record if the customer relationship does not already
127600 15/09/21 // exist. A relationship is defined as Link Code Identifier, Company Number, Customer
127800 15/09/21
127900 15/09/21 // When we request an insert "I" transaction type and the contact record
128200 15/09/21
128800 15/09/21
129400 15/09/21

```

Find From (YMMDD): 150901  
To (YMMDD): 150930

## How to display source member line dates

The screenshot shows a source code editor with a context menu open. The code includes comments and an SQL query. The context menu options are:

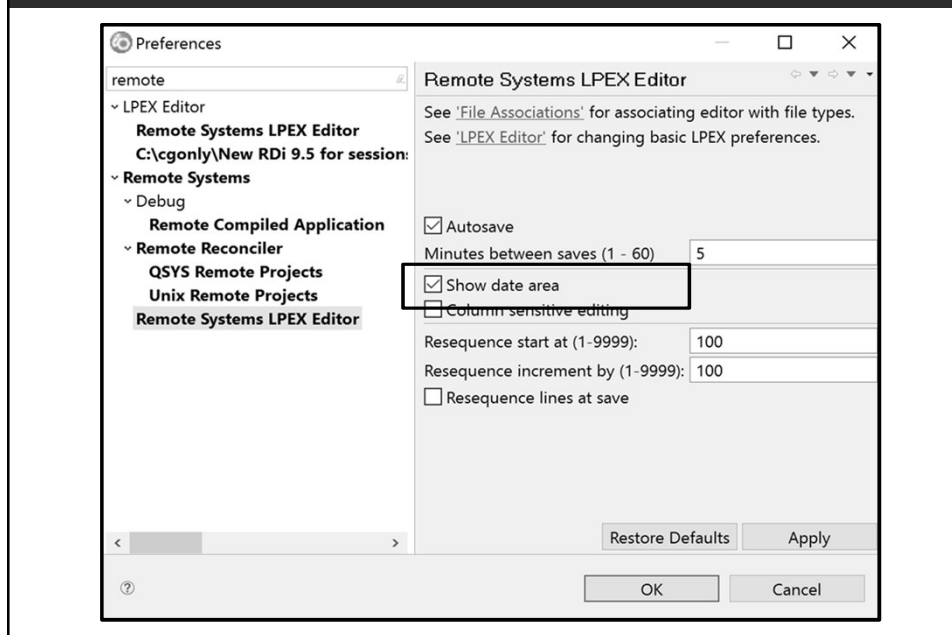
- New
- Prompt F4
- Syntax Check Line
- Save Ctrl+S
- Cut Ctrl+X
- Copy Ctrl+Insert
- Paste Ctrl+V
- Select
- Selected
- Deselect Alt+U
- Filter view
- Show all Ctrl+W
- Refactor
- Source
- View
- Add/Remove Breakpoint
- Set Service Entry Point...
- Hex edit line Alt+Shift+H
- Show Date Area

```

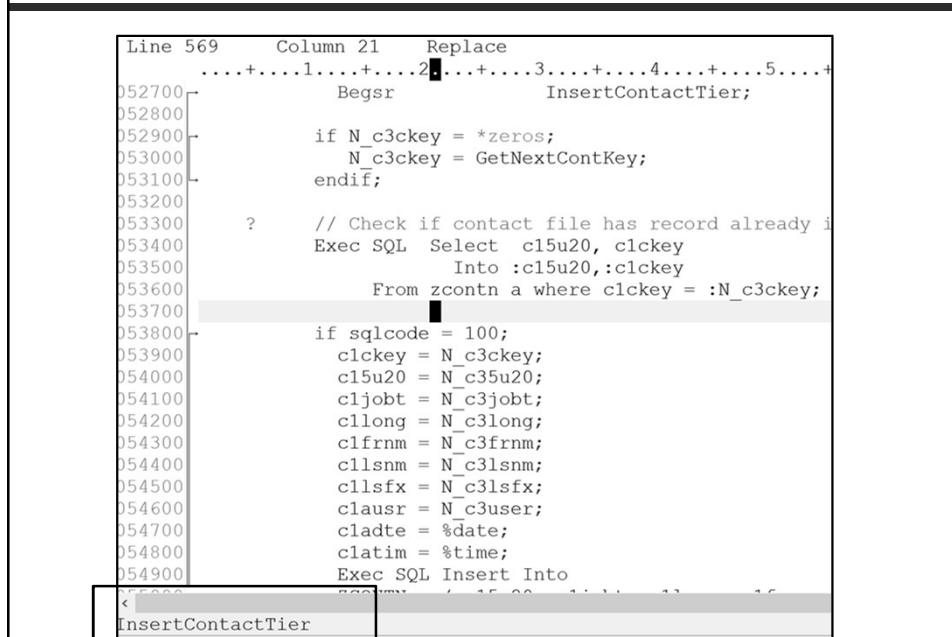
055400
055500 // Get the key value of the record just inserted to pass back
055600 // transaction file record
055700
055800 Exec SQL
055900 Select Max(c1ckey)
056000 Into :N_c3ckey
056001 From zcontn a
056002 where
056100
056101
056102 c15u20 = :c15u20
056200 and c1jobt = :c1jobt
056300 and c1long = :c1long
056400 and c1frnm = :c1frnm
056500 and c1lsnm = :c1lsnm
056600 and c1lsfx = :c1lsfx

```

## How to \*always\* have the date shown



## Ctrl-Shift-O Show Block Nesting





## Ctrl-F5 to remove block nesting arrows

```
RDIShorts.hce /CUSINQ.RPGLE /ZCONTTTRG.SQLRPGLE
Line 560 Column 13 Replace
.....1.....2.....3.....4.....5.....6.....
052700 Bgstr InsertContactTier;
052800
052900 if N_c3ckey = *zeros;
053000 N_c3ckey = GetNextContKey;
053100 endif;
053200
053300 ? // Check if contact file has record already if so ski
053400 Exec SQL Select c15u20, clickey
053500 Into :c15u20,:clickey
053600 From zcontn a where clickey = :N_c3ckey;
053700
053800 if sqlcode = 100;
053900 clickey = N_c3ckey;
054000 c15u20 = N_c35u20;
054100 c1jobt = N_c3jobt;
054200 c1long = N_c3long;
054300 c1frnm = N_c3frnm;
054400 c1lsnm = N_c3lsnm;
054500 c1lsfx = N_c3lsfx;
054600 clausr = N_c3user;
054700 cladte = %date;
054800 clatim = %time;
054900 Exec SQL Insert Into
055000
```

## Ctrl-2 Split screen processing

```
CITYDATA.RPGLE
CiconlyNew RD 95 for sessions/Get With The Program_XMLLIB/GRPGLESRC/CITYDATA.RPGLE 1
Line 13 Column 6 Replace
.....Name*****ETdsFrom**To/L**IDc.Keywords*****
001300 D MonthlyData ds
001400 D Month 9
001500 D Low 3
001600 D High 3
001700
001800 D optional1 s 100a inz('/xmldocs/citydata1.
001900 D filename s 25a inz('/xmldocs/citydata1.
002000 D x s 2 0
002100 D y s 2 0
002200
002300
002400 /free
002500
002600 // Setup parsing options
002700 options1 = 'doc=file +
002800 allowextra=yes allowmissing=yes case=any';
002900
003000 // Initialize path to file in IFS
003100 filename = '/xmldocs/citydata1.xml';
003200
003300 // Capture header data into data structure Cities
003400 xml-into Cities %xml(filename:options1);
003500
003600 // Capture detail data, load subfile
003700 for x = 1 to 2;
003800 cityname = cities.citydata(x).cityname;
003900 region = cities.citydata(x).region;
004000 for y = 1 to 3;
004100 monthname = cities.citydata(x).monthlydata(y).month;
004200 low = cities.citydata(x).monthlydata(y).low;
004300 high = cities.citydata(x).monthlydata(y).high;
004400 write citydata;
004500 endfor;

CiconlyNew RD 95 for sessions/Get With The Program_XMLLIB/GRPGLESRC/CITYDATA.RPGLE 2
Line 13 Column 6 Replace
.....Name*****ETdsFrom**To/L**IDc.Keywords*****
001300 D MonthlyData ds
001400 D Month 9
001500 D Low 3
001600 D High 3
001700
001800 D optional1 s 100a
001900 D filename s 25a inz('/xmldocs/citydata1.
002000 D x s 2 0
002100 D y s 2 0
002200
002300
002400 /free
002500
002600 // Setup parsing options
002700 options1 = 'doc=file +
002800 allowextra=yes allowmissing=yes case=any';
002900
003000 // Initialize path to file in IFS
003100 filename = '/xmldocs/citydata1.xml';
003200
003300 // Capture header data into data structure Cities
003400 xml-into Cities %xml(filename:options1);
003500
003600 // Capture detail data, load subfile
003700 for x = 1 to 2;
003800 cityname = cities.citydata(x).cityname;
003900 region = cities.citydata(x).region;
004000 for y = 1 to 3;
004100 monthname = cities.citydata(x).monthlydata(y).month;
004200 low = cities.citydata(x).monthlydata(y).low;
004300 high = cities.citydata(x).monthlydata(y).high;
004400 write citydata;
004500 endfor;
```

## Copy (Alt-C) or Move (Alt-M) source code

```

CITYDATA.RPGL2
C:\egonly\New RD: 95 for sessions\Get With The Program_XMLLIB\QRPGLSRC\CITYDATA.RPGL2:1
Line 46      Column 23  Replace
.....1.....2.....3.....4.....5.....6.....
001900      D filename  S      25a  inz('/xmldocs/citydata.
002000      D x         S         2  0
002100      D y         S         2  0
002200
002300
002400      //free
002500
002600      // Setup parsing options
002700      optional = 'doc=file +
002800      allowextra=yes allowmissing=yes case=any';
002900
003000      // Initialize path to file in IFS
003100      filename = '/xmldocs/citydata.xml';
003200
003300      // Capture header data into data structure Cities
003400      xml-into Cities %xml(filename:optional);
003500
003600      // Capture detail data, load subfile
003700      for x = 1 to 2;
003800          cityname = cities.citydata(x).cityname;
003900          region = cities.citydata(x).region;
004000          for y = 1 to 3;
004100              monthname = cities.citydata(x).monthlydata(y).month;
004200              low = cities.citydata(x).monthlydata(y).low;
004300              high = cities.citydata(x).monthlydata(y).high;
004400              write citydata*;
004500          endfor;
004600
004700
004800      *inlr = *onz;
004900      return;
005000
005100      /end-free
  
```

```

C:\egonly\New RD: 95 for sessions\Get With The Program_XMLLIB\QRPGLSRC\CITYDATA.RPGL2:2
Line 25      Column 9    Replace
DN*****ETDFrom+++To/L+++IDC.Keywords+++++
001300      D MonthlyData  ds
001400      D Month        9
001500      D Low          3
001600      D High         3
001700
001800      D optional    s      100a
001900      D filename    S      25a  inz('/xmldocs/citydata.
002000      D x           S         2  0
002100      D y           S         2  0
002200
002300
002400      //free
002500
002600      // Setup parsing options
002700      optional = 'doc=file +
002800      allowextra=yes allowmissing=yes case=any';
002900
003000      // Initialize path to file in IFS
003100      filename = '/xmldocs/citydata.xml';
003200
003300      // Capture header data into data structure Cities
003400      xml-into Cities %xml(filename:optional);
003500
003600      // Capture detail data, load subfile
003700      for x = 1 to 2;
003800          cityname = cities.citydata(x).cityname;
003900          region = cities.citydata(x).region;
004000          for y = 1 to 3;
004100              monthname = cities.citydata(x).monthlydata(y).month;
004200              low = cities.citydata(x).monthlydata(y).low;
004300              high = cities.citydata(x).monthlydata(y).high;
004400              write citydata*;
004500          endfor;
  
```

## Ctrl-0 to close additional source views

```

CITYDATA.RPGL2
C:\egonly\New RD: 95 for sessions\Get With The Program_XMLLIB\QRPGLSRC\CITYDATA.RPGL2:1
Line 13      Column 6    Replace
.....1*****ETDFrom+++To/L+++IDC.Keywords+++++
001300      D MonthlyData  ds
001400      D Month        9
001500      D Low          3
001600      D High         3
001700
001800      D optional    s      100a
001900      D filename    S      25a  inz('/xmldocs/citydata.
002000      D x           S         2  0
002100      D y           S         2  0
002200
002300
002400      //free
002500
002600      // Setup parsing options
002700      optional = 'doc=file +
002800      allowextra=yes allowmissing=yes case=any';
002900
003000      // Initialize path to file in IFS
003100      filename = '/xmldocs/citydata.xml';
003200
003300      // Capture header data into data structure Cities
003400      xml-into Cities %xml(filename:optional);
003500
003600      // Capture detail data, load subfile
003700      for x = 1 to 2;
003800          cityname = cities.citydata(x).cityname;
003900          region = cities.citydata(x).region;
004000          for y = 1 to 3;
004100              monthname = cities.citydata(x).monthlydata(y).month;
004200              low = cities.citydata(x).monthlydata(y).low;
004300              high = cities.citydata(x).monthlydata(y).high;
004400              write citydata*;
004500          endfor;
  
```

```

C:\egonly\New RD: 95 for sessions\Get With The Program_XMLLIB\QRPGLSRC\CITYDATA.RPGL2:2
Line 13      Column 6    Replace
.....1*****ETDFrom+++To/L+++IDC.Keywords+++++
001300      D MonthlyData  ds
001400      D Month        9
001500      D Low          3
001600      D High         3
001700
001800      D optional    s      100a
001900      D filename    S      25a  inz('/xmldocs/citydata.
002000      D x           S         2  0
002100      D y           S         2  0
002200
002300
002400      //free
002500
002600      // Setup parsing options
002700      optional = 'doc=file +
002800      allowextra=yes allowmissing=yes case=any';
002900
003000      // Initialize path to file in IFS
003100      filename = '/xmldocs/citydata.xml';
003200
003300      // Capture header data into data structure Cities
003400      xml-into Cities %xml(filename:optional);
003500
003600      // Capture detail data, load subfile
003700      for x = 1 to 2;
003800          cityname = cities.citydata(x).cityname;
003900          region = cities.citydata(x).region;
004000          for y = 1 to 3;
004100              monthname = cities.citydata(x).monthlydata(y).month;
004200              low = cities.citydata(x).monthlydata(y).low;
004300              high = cities.citydata(x).monthlydata(y).high;
004400              write citydata*;
004500          endfor;
  
```

## F3 – Quick navigation to a subroutine or procedure

```

When      N_c3UDI = 'I';      // New record insert requested

// When insert requested always insert a contact record
// or get the contact record information to use when populating
// additional tier information
Exsr InsertntContactTier;

// If interface file has phone information insert a phone tier
// record
if N_c3PHNO <> *blanks;
Exsr InsertPhoneTier;
Endif;

// If interface file has e-mail information insert a e-mail
// tier record
if N_c3EMAD <> *blanks;
Exsr InsertEmailTier;
Endif;

```

```

063400 16/11/21      // *****
063500 16/11/21      //              Insert Contact Tier
063600 16/11/21      // *****
063700 16/11/21      Begsr          InsertntContactTier;
063800 15/09/21
063900 16/11/21      if N_c3ckey = *zeros;
064000 16/11/21          N_c3ckey = GetNextContKey;
064100 16/11/21      endif;
064200 15/09/21

```

```

EVFP0020 Press Alt + Left/Right arrow to switch between the previous and the current location.

```

## Ctrl-F5 Remove embedded warnings and errors after verify

```

        when TrigEvent = TriggerEventInsert;
            // This is a transaction table and will only have inserts
            // The UDI operation code will determine what happens to the
            // contact, phone, email, attribute or event tier information

            Select;
            When      N_c3UDI = 'NEW';      // New contact key requested
RNF7503: Expression contains an operand that is not defined.
RNF7030: The name or indicator N_C3UDI is not defined.

            // When NEW is specified we are looking for the next contact record
            // key value to use. This is contained in the dtaara RBSLSTCKEY
            N_c3ckey = GetNextContKey;
RNF7503: Expression contains an operand that is not defined.
RNF7030: The name or indicator N_C3CKEY is not defined.

            When      N_c3UDI = 'I';      // New record insert requested
RNF7503: Expression contains an operand that is not defined.

            // When insert requested always insert a contact record
            // or get the contact record information to use when populating
            // additional tier information
            Exsr InsertntContactTier;

```

```

when TrigEvent = TriggerEventInsert;
// This is a transaction table and will only have inserts
// The UDI operation code will determine what happens to the
// contact, phone, email, attribute or event tier information

Select;
When      N_c3UDI = 'NEW';      // New contact key requested

// When NEW is specified we are looking for the next contact record
// key value to use. This is contained in the dtaara RBSLSTCKEY
N_c3ckey = GetNextContKey;

When      N_c3UDI = 'I';      // New record insert requested

// When insert requested always insert a contact record
// or get the contact record information to use when populating
// additional tier information

```

## What We'll Cover ...

- Remote Systems Explorer
- LPEX Editor Shortcuts
- **LPEX Commands**
- RDi and Modernization
- Wrap-up

## LPEX Commands

The screenshot shows the Help window for IBM Rational Developer for i. The search bar is empty, and the scope is set to 'All topics'. The 'Contents' pane on the left shows a tree view with 'Default editor commands' selected. The main pane displays the following content:

Rational Developer for i 9.5.1 > Developing > Developing applications for IBM i > Developing IBM i server applications using Remote System Explorer > Editing, compiling, and debugging applications in the Remote System Explorer > Editing > LPEX Editor

### Default editor commands

The editor is fully programmable through the use of an extensive set of editor commands, actions, and parameters. You can use commands and parameters to customize the editor window, search for or change text in your document, or perform many other functions.

Commands may be issued from the editor command line, or run programmatically in editor extensions. Select a command from the list below to display reference information for that command.

Command Name	Description
<a href="#">action</a>	Runs an editor action.
<a href="#">add</a>	Adds one or more blank lines.
<a href="#">block</a>	Sets or manipulates a block of text.
<a href="#">calc</a>	Performs simple calculations.
<a href="#">compare</a>	Examines two documents for differences in their content.
<a href="#">delete</a>	Deletes one or more lines.
<a href="#">deleteText</a>	Deletes text at the current cursor position.
<a href="#">expandAll</a>	Brings hidden lines back into view.
<a href="#">findText</a>	Finds and optionally replaces text in the document.
<a href="#">get</a>	Imports a file into the document.
<a href="#">help</a>	Displays help information on a requested editor topic.
<a href="#">text</a>	Gets user input to use with an editor command.

## D Specs out of order? Sort them!

### Before

```
d Pos          s          4s 0
d hldcsnr      s          like(cocsno)
d Chr          s          4s 0
d Len          s          4s 0
d SameCsnr     s          n
d Alpha        c          const('ABCDEFGHJKLMNOPQRSTUVWXYZ')
d SignerRec    s          n
```

### After

```
d Alpha        c          const('ABCDEFGHJKLMNOPQRSTUVWXYZ')
d Chr          s          4s 0
d Len          s          4s 0
d Pos          s          4s 0
d SameCsnr     s          n
d hldcsnr      s          like(cocsno)
d SignerRec    s          n
```

LPEX Command SORT

## Escape quickly to the command line

```
000100      ctl-opt pgminfo(*pcml:*module) nomain debug
000200          option( *srcstmt : *nodebugio );
000300
000400
000500 // This program receives an item class and returns the first 200
000600 // items found in the item master table.
000700
000800 // Note - If you wanted to have a 'conversation' and continue
000900 // returning each subsequent set of 200 you would need to
001000 // receive the last item sent to the consumer and use
001100 // that as your starting point.
001200
001300 // Data structure webordersdata's only purpose is to define table fields.
001400 dcl-ds webordersdata extname('WEBITEMS') template end-ds;
001500
001600 // If you want to return more than 200 items, just change the value here.
001700 dcl-c maxelements const(200);
001800
001900 dcl-ds itemlistds qualified;
002000         itemnumber like(acitemno);
002100         description like(acitemdes);
002200         unitofmeasure like(acunitofms);
002300         itemclass like(acitemclas);
002400 end-ds;
002500
002600
```

## Compare

```

ZCONTITRGSQRPGLE
Line 111 Column 1 Replace 2 changes
.....1.....2.....3.....4.....5.....6.....7.....8.....9.....0
009600 Phone Packed(10:0) Const;
009700 PhoneExt Char(4) Const;
009800 PhoneCC Char(3) Const;
009900 Email Char(256) Const;
010000 Verbose Char(1) Const Options(*NOPASS);
010100 END-PR;
010200
010300 DCL-PR MACHGOPT EXTPGM('TSCHGOPT');
010400 ReturnCode Char(6); // returns 'OK' if successful
010500 UserId Char(256) Const;
010600 Opt_Type Char(50) Const;
010700 Opt_Value Char(500) Const;
010800 Verbose Char(1) Const Options(*NOPASS);
010900 END-PR;
011000
011100
011200 DCL-PR RESETEML EXTPGM('SNDPAERET');
011300 UserId Char(256) Const;
011400 Token Char(60) Const;
011400 Emad Char(128) Const;
011500 ReturnCode Char(1); // returns 1 if successful '0' if not
011600 END-PR;
011700
011800
011900
012000
012100
012200 D Main pr extpgm('ZCONTITRG')
012300 D TrgBuffer likeds(TrgParm)
012400 D TrgBufferLen 10i 0
012500
012600 D Main pi
012700 D TrgBuffer likeds(TrgParm)
012800 D TrgBufferLen 10i 0
One or more mistakes found.
compare

```

## Compare clear

```

ZCONTITRGSQRPGLE
Line 127 Column 61 Replace 2 changes
.....DName+++++ETDsFrom++To/L+++IDc.Keywords+++++Comments+++++
009600 Phone Packed(10:0) Const;
009700 PhoneExt Char(4) Const;
009800 PhoneCC Char(3) Const;
009900 Email Char(256) Const;
010000 Verbose Char(1) Const Options(*NOPASS);
010100 END-PR;
010200
010300 DCL-PR MACHGOPT EXTPGM('TSCHGOPT');
010400 ReturnCode Char(6); // returns 'OK' if successful
010500 UserId Char(256) Const;
010600 Opt_Type Char(50) Const;
010700 Opt_Value Char(500) Const;
010800 Verbose Char(1) Const Options(*NOPASS);
010900 END-PR;
011000
011100
011200 DCL-PR RESETEML EXTPGM('SNDPAERET');
011300 UserId Char(256) Const;
011400 Token Char(60) Const;
011500 Emad Char(128) Const;
011600 ReturnCode Char(1); // returns 1 if successful '0' if not
011700 END-PR;
011800
011900
012000
012100
012200 D Main pr extpgm('ZCONTITRG')
012300 D TrgBuffer likeds(TrgParm)
012400 D TrgBufferLen 10i 0
012500
012600 D Main pi
012700 D TrgBuffer likeds(TrgParm)
012800 D TrgBufferLen 10i 0
012900
013000 ? // Pointer based data structures for the input buffer.
compare clear

```

## Compare prompt

The screenshot shows a mainframe editor window with a list of lines and columns. A 'Compare File' dialog box is open, showing a file selection process. The dialog box title is 'Compare File' and it shows the current directory as 'COMMON\_JO\_ > QRPGLSRC'. The file list includes:

Name	Date modified	Type
COPYJOHNG	6/15/2016 12:03 PM	SQLRPGLE
EXTRKSGNR.RPGLE	5/9/2016 8:38 PM	RPGLE File
EXTRKSGNRA.RPGLE	5/9/2016 8:38 PM	RPGLE File
EXTRKSGNRB.RPGLE	5/9/2016 8:38 PM	RPGLE File
EXTRKSGNRC	5/9/2016 8:38 PM	SQLRPGLE
ZCONTITRG	11/21/2016 10:05 ...	SQLRPGLE

The 'File name' field contains 'EXTRKSGNRB'. The 'Open' button is highlighted. Below the dialog box, the mainframe editor shows the following code:

```

Line 35      Column 1      Replace
.....DName+++++++ETDsFrom+++To/L+++IDc.Keywords+++++++Comments+++++++
000200      h option(*nodebugio) dftactgrp(*no) actgrp(*caller) alwnull(*usrctl)
000300      h usrprf(*owner)
000400
000500      ? // ----
000600      ? // Program
000700      ? // Create Da
000800      ? // Writtes B
000900      ? //
001000      ? // Descripti
001100      ? //
001200      ? // ----
001300      ? // Mod Ta
001400      ? // Change Da
001500      ? // Change E
001600      ? // Descripti
001700      ? // ----
001800      ? // Extend
001900      FSCOMSVX  if
002000
002100      D Main
002200      D      CSNO
002300      D      Count
002400
002500      D Main
002600      D      CSNO
002700      D      Count
002800
002900      d Alpha      c      const('ABCDEFGHIJLKNOPQRSTUVWXYZ')
003000      d Pos        s      4s 0
003100      d Chr        s      4s 0
003200      d Len        s      4s 0
003300      d hldcsnr    s      like(cocsnr)
003400      d SameCsnr s      n
003500      d SignerRec  s      n
003600
  
```

At the bottom of the editor window, the text 'compare prompt' is visible.

## Jump to a specific line number

The screenshot shows a mainframe editor window with a list of lines and columns. Line 22 is highlighted. The code listing is as follows:

```

Line 22      Column 1      Replace      4 changes
.....1.....2.....3.....4.....5.....6.....7.....8.....
000200      option(*srcstmt : *nodebugio);
000300
000400
000500      // This program receives an item class and returns the first 200
000600      // items found in the item master table.
000700
000800      // Note - If you wanted to have a 'conversation' and continue
000900      // returning each subsequent set of 200 you would need to
001000      // receive the last item sent to the consumer and use
001100      // that as your starting point.
001200
001300      // Data structure webordersdata's only purpose is to define table fields.
001400      dcl-ds webordersdata extname('WEBITEMS') template end-ds;
001500
001600      // If you want to return more than 200 items, just change the value here.
001700      dcl-c maxelements const(200);
001800
001900      dcl-ds itemlistds qualified;
002000      itemnumber like(acitemno);
002100      description like(acitemdes);
002200      unitofmeasure like(acunitofms);
002300      itemclass like(acitemclas);
002500      end-ds;
002600
002700
002800      // Subsequent activities have
  
```

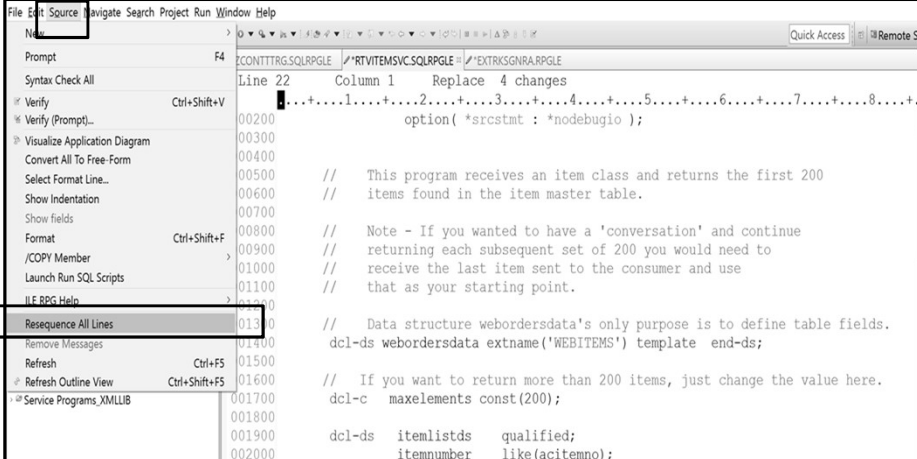
The line number '22' is shown in the bottom left corner of the editor window.

## Jump to a specific line number – shortcut – CTRL-L

```
003200 D zcontr e ds extname(zcontr)
003300
003400 // Customer link code view
003500 D M_LnkV01 e ds extname(m_lnkV01)
003600
003700 D QcmdExec pr extpgm('QCMDEXC')
003800 D Cmd 1024a options(*varsize)
003900 D CmdLen 15p 5
004000
004100
004200 D Main pr extpgm('ZCONTTTRG')
004300 D TrgBuffer likeds(TrgParm)
004400 D TrgBufExt 10i 0
```

Line number:

## Resequence All Lines



The screenshot shows an IDE window with the 'Edit' menu open. The 'Resequence All Lines' option is highlighted. The background shows a code editor with the following content:

```
002000 option(*srcstmt : *nodebugio);
003000
004000
005000 // This program receives an item class and returns the first 200
006000 // items found in the item master table.
007000
008000 // Note - If you wanted to have a 'conversation' and continue
009000 // returning each subsequent set of 200 you would need to
010000 // receive the last item sent to the consumer and use
011000 // that as your starting point.
012000
013000 // Data structure webordersdata's only purpose is to define table fields.
014000 dcl-ds webordersdata extname('WEBITEMS') template end-ds;
015000
016000 // If you want to return more than 200 items, just change the value here.
017000 dcl-c maxelements const(200);
018000
019000 dcl-ds itemlistds qualified;
020000 itemnumber like(acitemno);
```

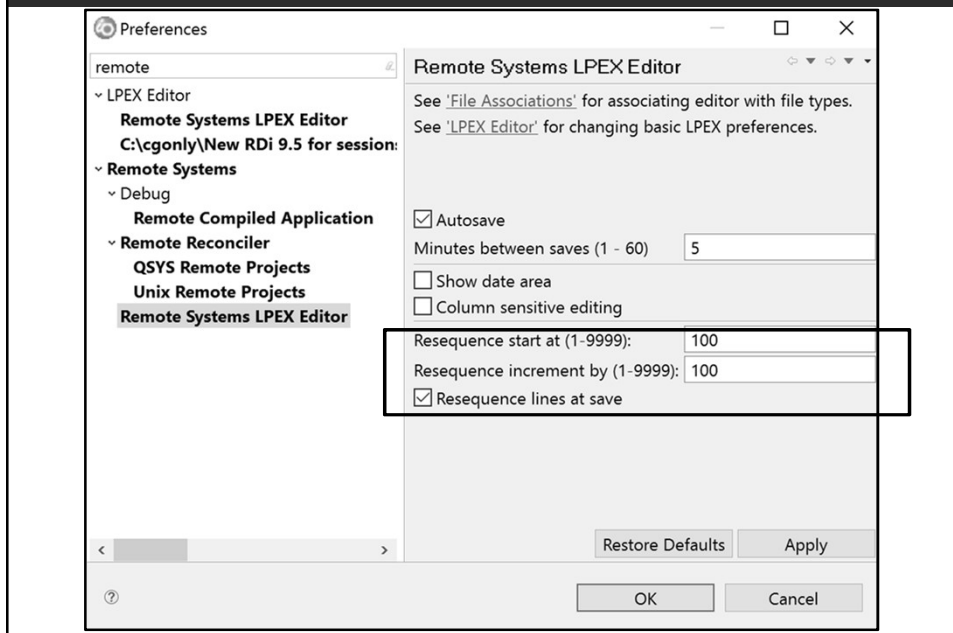


## Resequence

```
*ZCONTTRGSQLRPGLE **
Line 119      Column 1      Replace 3 changes
009600      Phone Packed(10:0) Const;
009700      PhoneExt Char(4) Const;
009800      PhoneCC Char(3) Const;
009900      Email Char(256) Const;
010000      Verbose Char(1) Const Options(*NOPASS);
010100      END-PR;
010200
010300      DCL-PR MACHGOPT EXTPGM('TSCHGOPT');
010400      ReturnCode Char(6); // returns 'OK' if successful
010500      UserId Char(256) Const;
010600      Opt_Type Char(50) Const;
010700      Opt_Value Char(500) Const;
010800      Verbose Char(1) Const Options(*NOPASS);
010900      END-PR;
011000
011100      DCL-PR RESETEML EXTPGM('SNDFAERET');
011300      UserId Char(256) Const;
011400      Token Char(60) Const;
011500      Emad Char(128) Const;
011600      ReturnCode Char(1); // returns 1 if successful '0' if not
011700      END-PR;
011800
011900
012000
012100
012200      D Main          pr          extpgm('ZCONTTRG')
012400      D TrgBuffer          likeds(TrgParm)
012500      D TrgBufferLen      10i 0
012600
012700      D Main          pi          likeds(TrgParm)
012800      D TrgBuffer          10i 0
012900      D TrgBufferLen
013000
```

resequence

## How to \*always\* resequence the line numbers at save



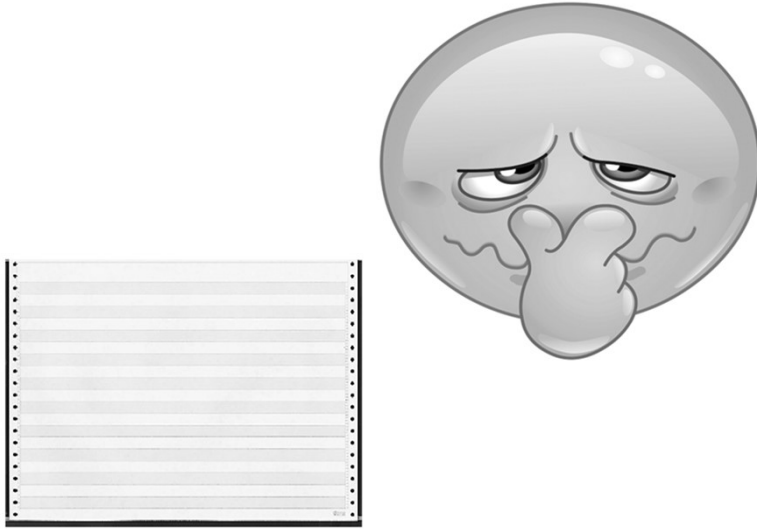
## What We'll Cover ...

- Remote Systems Explorer
- LPEX Editor Shortcuts
- LPEX Commands
- **RDi and Modernization**
- Wrap-up

## How RDi is helping with modernization



## Code Smell



## Several ways to get better code



## Several ways to get better code

- Field Renaming
- Extraction
- Generalization
- Encapsulation

## RDİ Developer Hub

The screenshot shows the IBM developerWorks website interface for the RDİ Developer Hub. The top navigation bar includes 'Technical topics', 'Evaluation software', 'Community', and 'Events'. Below this, there are tabs for 'Profiles', 'Communities', and 'Apps'. The main content area is titled 'Communities' and features a section for 'RDİ - IBM Rational Developer for i Hub'. This section includes a community description, a list of recent updates, and a list of announcements. The announcements list includes items such as 'New! Announcement: Rational Developer for i 9.5.1', 'New! Video on new Git and Orion support', and 'Announcement: Rational Developer for i 9.5.0.3'. The page also features a sidebar with navigation options like 'Overview', 'Members', 'Wiki', 'Blog', 'Feeds', 'Bookmarks', 'Surveys', and 'Tags'.

## RDi Latest Update

The screenshot shows a blog post interface. At the top, there are navigation tabs for 'My Blogs', 'Public Blogs', and 'My Updates'. The main title is 'RDi - IBM Rational Developer for i Hub'. Below this, there's a sub-header 'Rational Developer for i V9.5.1.2 fixpack is now available!' by 'eric.simpson' on 'Apr 13', with '2' comments and '2168' visits. The post content includes a title 'Rational Developer for i V9.5.1.2 fixpack for V9.5 is now available!', a paragraph stating 'The fixpack contains enhancements and fixes. For details, see the link in the Fix List.', and another paragraph: 'You can go to IBM Installation Manager and click Update or go to FixCentral to find the downloadable files (links available in the Fix List document)'. A 'Fix List' link is provided: 'http://www-01.ibm.com/support/docview.wss?uid=swg27038481'. There are social media sharing buttons for Google+, Like (0), Share, and Tweet. A sidebar on the left lists 'Overview', 'Recent Updates', 'Members', 'Wiki', 'Blog' (highlighted), 'Feeds', and 'Bookmarks'. At the bottom, there are tags 'rdi 9512 install' and a 'Like' button with the text 'Be the first of your friends to like this.' and social media icons for SU, LinkedIn, and Twitter.

## Refactor ILE RPG Code

The screenshot shows an IBM Knowledge Center article. The breadcrumb trail is: 'Rational Developer for i > Rational Developer for i 9.5.1 > Developing > Developing applications for IBM i > Developing IBM i server applications using Remote System Explorer > Editing, compiling, and debugging applications in the Remote System Explorer > Editing > Refactor ILE RPG code'. The article title is 'Refactor ILE RPG code' with a 'Version 9.5.1' dropdown menu. The content describes refactoring as 'the process of changing code without changing the function of the code' and provides instructions on how to use the 'Refactor > Rename' action. A 'Procedure' section starts with: '1. Open the source member that you want to refactor. The following example shows a source member with a declaration called name that is selected on line 8. Throughout the source member, this instance of name is spelled with several different casings that would benefit from renaming so that a consistent spelling can be applied. As well, there are additional declarations called name within another dcl-ds and a dcl-proc that will remain unaffected by renaming because they are within different scopes and are not related to the declaration that was selected.'

## Refactoring in Action

```

// *****
//                               Insert Attribute Tier CRTATT
// *****
Dcl-Proc InsertCRTATTR;
  Dcl-Pi InsertCRTATTR  varchar(128);
  End-Pi;

  Dcl-s error varchar(128);

  // First check to make sure the contact key supplied is not zero,
  // if so, return error.

  If N_c3ckey = *zeros;
    error =
    'The supplied contact key is zero and is invalid. You must supply-
the contact key when adding any other tier';
    return error;
  Endif;

  // Check if attribute passed is valid
  Exec SQL Select count(*)
    Into :RowCount
    From zattrmst where m2attr = :N_c3attr;

  If RowCount = *zeros;
    error = 'Invalid attribute entered. Value not listed in master file-
';
    return error;
  Endif;

  // Get attribute key value
  Exec SQL Select m2akey
    Into :c5akey
    From zattrmst where m2attr = :N_c3attr;

```

## RFE = Request for Enhancements

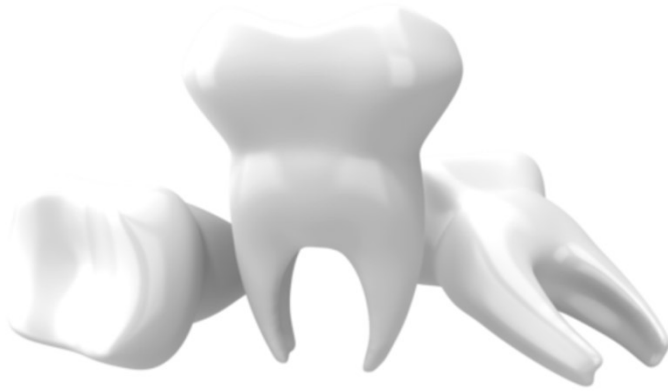
IBM Support My support Downloads Documents Tickets

9.5.1 Fix Pack 1 (9.5.1.1)

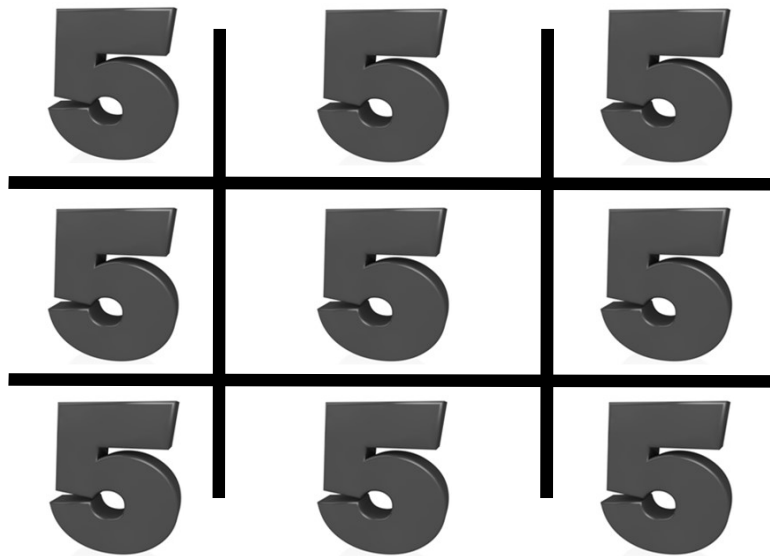
Link	Date Released	Status
<a href="#">Download 9.5.1.1</a>	21 December 2016	Current

APAR/RFE	Description
<a href="#">RFE31305</a>	Refactoring support for ILE RPG - Ability to rename declarations and all (and only) its references within the source file is delivered as documented at <a href="http://www.ibm.com/support/knowledgecenter/SSAE4W_9.5.1/com.ibm.etools.iseries.rse.doc/topics/refactor.html">http://www.ibm.com/support/knowledgecenter/SSAE4W_9.5.1/com.ibm.etools.iseries.rse.doc/topics/refactor.html</a> . - The other possible refactorings have been split into individual RFEs. Please vote on them to provide feedback on the relative demand. - Extract Constant - <a href="http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&amp;CR_ID=98365">http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&amp;CR_ID=98365</a> - Extract Local Variable - <a href="http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&amp;CR_ID=98367">http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&amp;CR_ID=98367</a> - Extract Procedure - <a href="http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&amp;CR_ID=98369">http://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&amp;CR_ID=98369</a>
<a href="#">RFE14402</a>	Add preference to default all opens to be browse only. On Remote Systems>IBM i>Objects Subsystem preference page, there is now a 'Open source in browse mode' preference.
SE66252 SE66265	<a href="#">RFE12100</a> Kerberos authentication for RDI has fixes if used on Windows in some conditions or if used for debugging: APAR SE66252 - Kerberos support does not work on Windows in some conditions APAR SE66265 - with Kerberos support setting SEP results in error In addition to RDI 9.5.1.1, the following host PTFs are required as well: V7R1M0 PTF SI53473 V7R2M0 PTF SI63271 V7R3M0 PTF SI55962

**Extract Local Variable**



**Extract Constant**



## Extract Procedure



## What We'll Cover ...

- Remote Systems Explorer
- LPEX Editor Shortcuts
- LPEX Commands
- RDi and Modernization
- Wrap-up



**The ABCs of RDi shortcuts**



**Charles Guarino**

**Twitter @charlieguarino**  
**THANK YOU !!!**