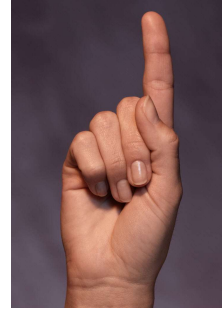## A Walk Through RDi Starting at Go!
(Rational Developer for i)

Charles Guarino
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 company. In addition to being a professional speaker, he is a frequent contributor of technical and strategic articles and webcasts for the IT community. He is a proud member of COMMON's Speaker Excellence Hall of Fame and also Long Island Software and Technology Network's Twenty Top Techies of 2009. Charles currently serves as a member of COMMON's Strategic Education Team (SET) and is also Immediate Past President and monthly Q&A host of LISUG, a Long Island IBM i User's Group www.lisug.org.

Charles can be reached at cguarino@centralparkdata.com.

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

In This Workshop …

It was an application developer revolution when IBM first introduced WDSC. However there was a slow adoption rate because of its poor client side performance.

Through the years the product evolved and took a giant step forward when it was taken over by the Rational team. Re-introduced as Rational Developer for i (Rdi) it performed significantly better than WDSC.

As the product evolution continued with new functionality the case for not using it becomes weaker and weaker. The name changed to keep pace with the platform itself, morphing into Rational Developer for Power Systems Software (RDP) and now once again RDi.

The current offering easily outperforms and outmaneuvers the PDM and SEU green screen development environments. Studies have shown time and time again significant gains in development and deployment time when using the graphical workbench. NOW is the time for you to step up and code like you never have before.

www.eclipse.org

New Terms and Concepts (cont.)

- **Workspace**
  - Folder where your local settings are stored
  - You can use multiple workspaces one at a time
  - Recommendation – don't use default setting, create your own
- **Perspective**
  - Your current desktop each with its own toolbars
  - RSE is a perspective within RDi
- **LPEX**
  - Live Parser Extensible Editor
  - A Rich UI that significantly outperforms SEU
  - The default editor of RDi

RDi – What it is

- **Your desktop or laptop is the LOCAL system**
  - The IBM i is the REMOTE system!

Your laptop or desktop

Remote IBM i
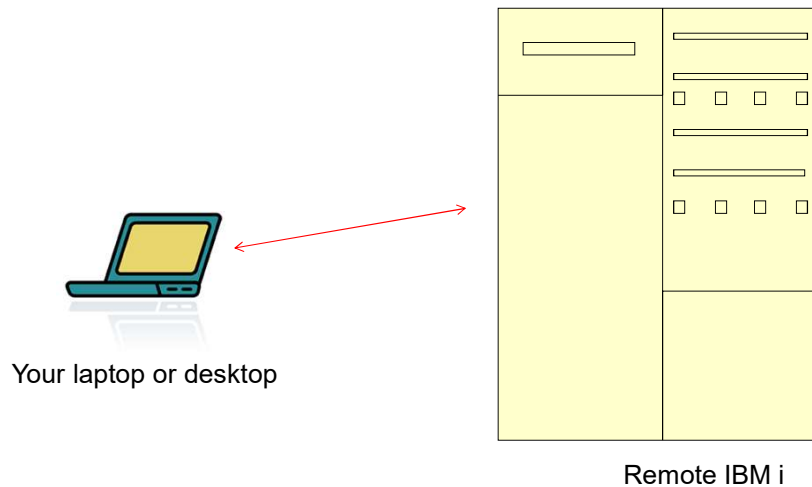
RDi – What it is NOT

- · **RDi is NOT a 5250 EMULATOR**
 - • **Entering interactive commands such as WRKSPLF will result in an error window appearing**

EVFC5001:I

No interactive job associated with this connection. Associate an interactive job with this connection to continue.

Cancel   << Details

A request to run an interactive command or program was received for the RSE connection CPDS, but no interactive job is currently associated with this connection. To associate an interactive job with this connection:
1. Open a 5250 emulator,
2. Run the CL command STRRSESVR NAME(CPDS)

Firing up RDi

IBM Rational Developer for i

Integrated IBM Products
- IBM® Data Studio Version 4.1.0.0
- IBM Rational Developer for i

Loading Workbench
Licensed Materials - Property of IBM Corp.© IBM Corporation and other(s) 1992, 2010.
Copyright and trademark information

This is where the workspace is selected at startup
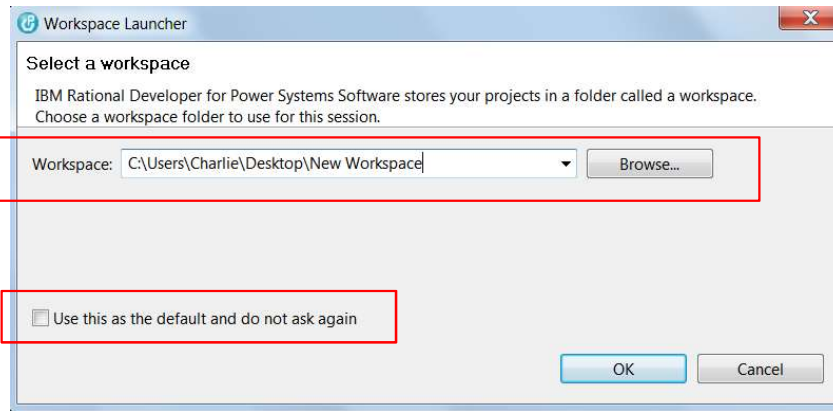
"Out-of-the-box" RSE perspective

- **Default settings shown**
  - **Perspective is completely customizable**
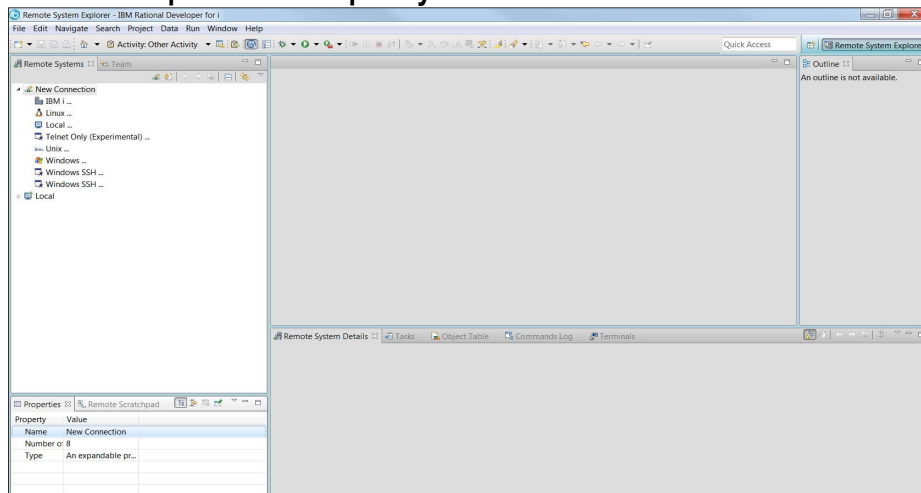
Creating a new connection to the IBM i

- **Connection names can be renamed if necessary**
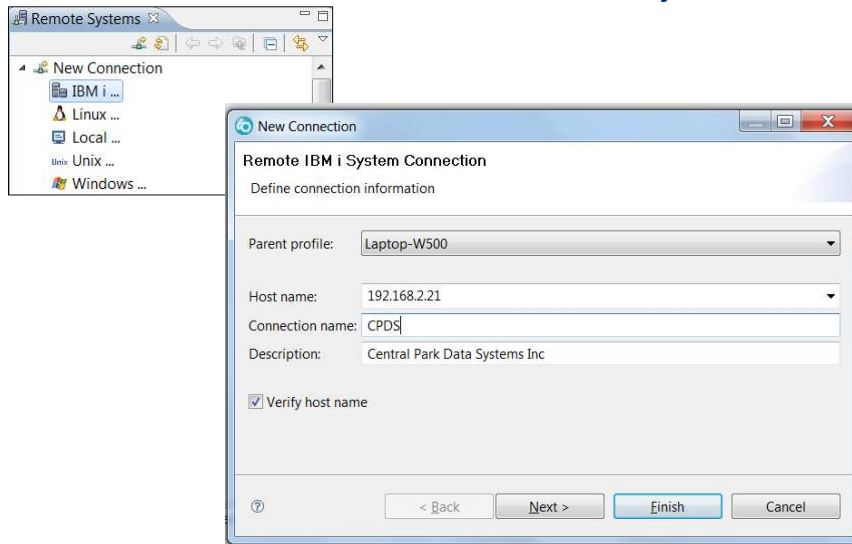
Connecting to your new connection is very simple

- **Right click on Connect**
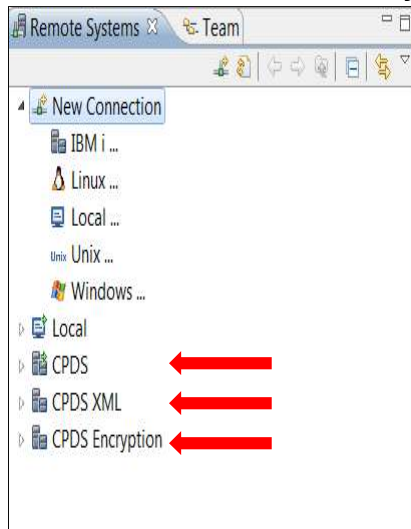- **Same user authority applies while navigating in RDi**
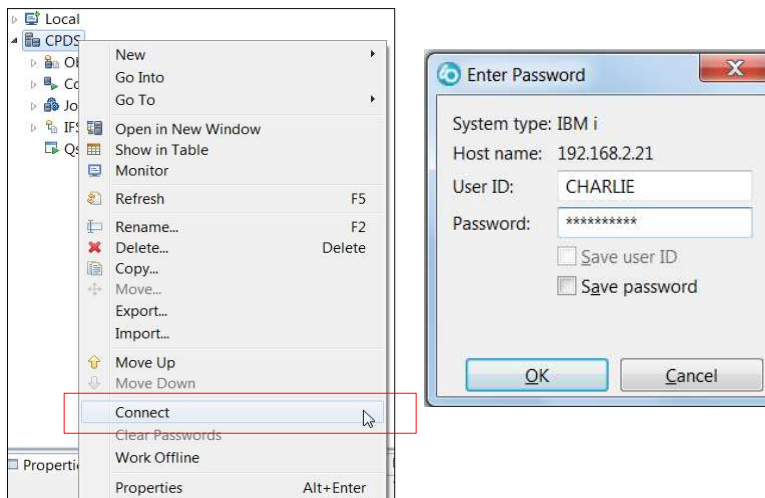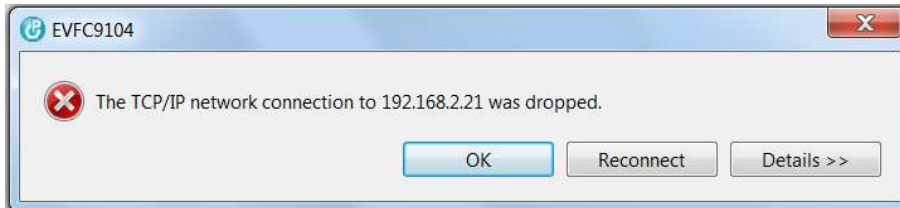
Never underestimate the power of the right click
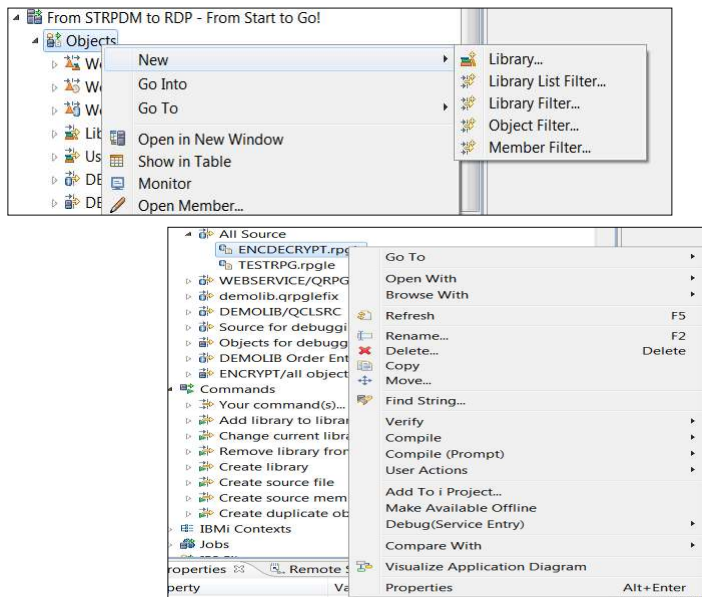
Multiple Connections to the IBM i

- **Each connection is like an active session in IBM i Access**
  - **Each with it's own library list, job descriptions, filters, etc.**

What happens if the connection drops?

- **This is the NUMBER ONE asked question!**
- **RDi has remarkable restartability**
- **Your environment is stored in your workspace**

Connection Subsystems

- **Automatically inserted into each connection you create**
- **Absolutely nothing to do with IBM i subsystems QCTL, QBATCH...**
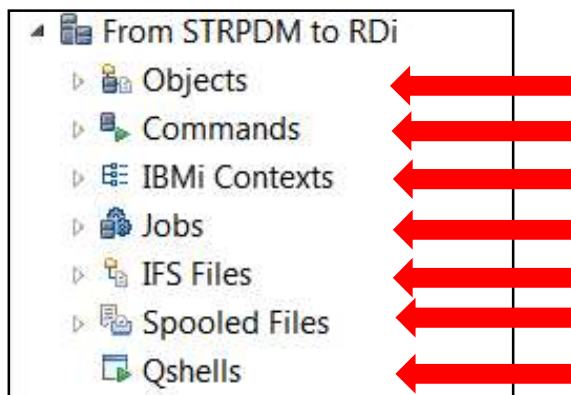
Setting up your initial connection environment

- **Initial library list, current library, initial command to run**
  - **Initial command helpful for dynamic library lists**
  - **Right click on connection name**
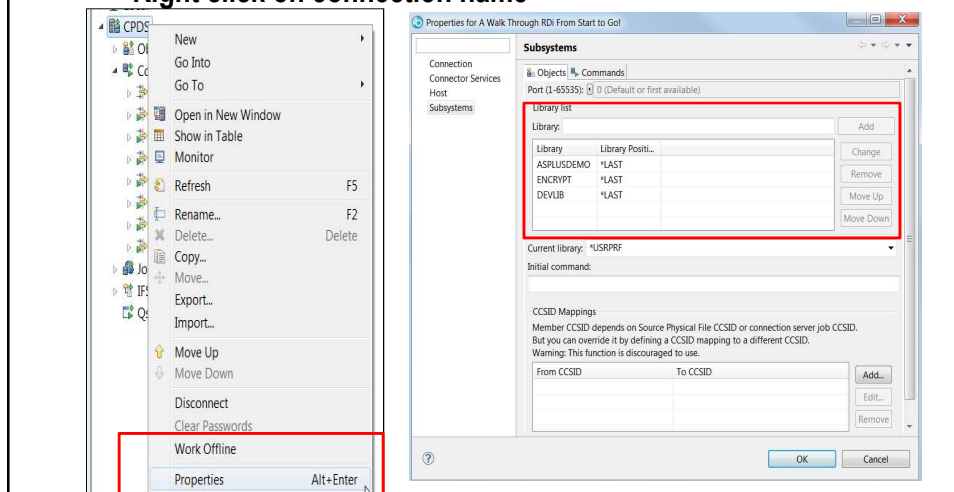
A FILTER limits what objects appear in your current selection

- **You've been using filters in SEU all along**
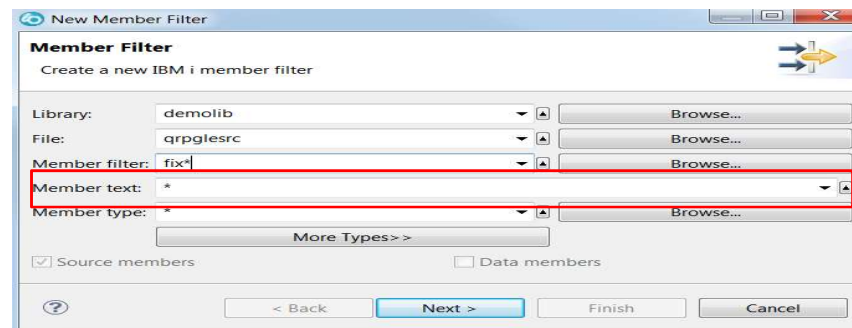
demolib.qrpglefix is the name chosen for this filter

- Click on any filter and its contents open

A production scenario of using a multi-string filter

- Having all related source members in one convenient spot eliminates the need for constant "F12-ing" and retyping
- This filter shows RPGLE, CLLE, DSPF and a TXT member

10

Show in Table is VERY versatile!

RDi is aware of each object table entry
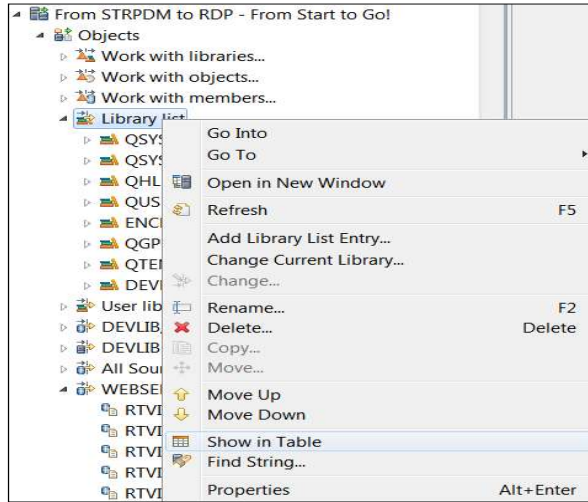
Show in Table is VERY versatile!

Can add new objects to a library or manipulate the library itself
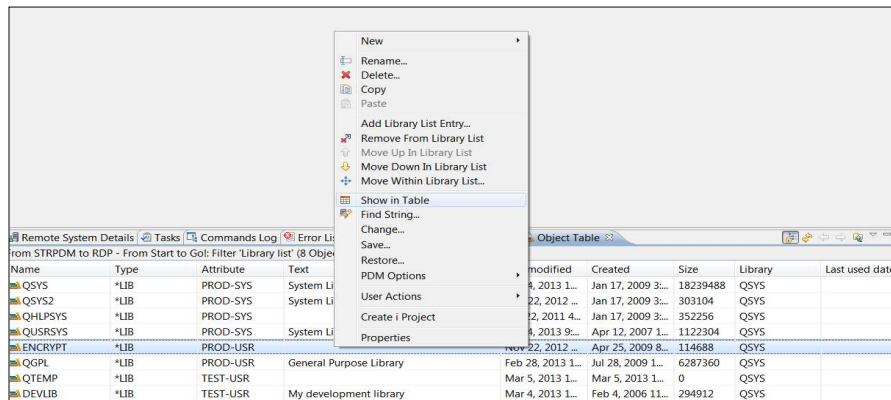
RDi respects source member locking

- **The reverse situation is also true**
  - **Source members are NOT locked when downloaded to work offline**
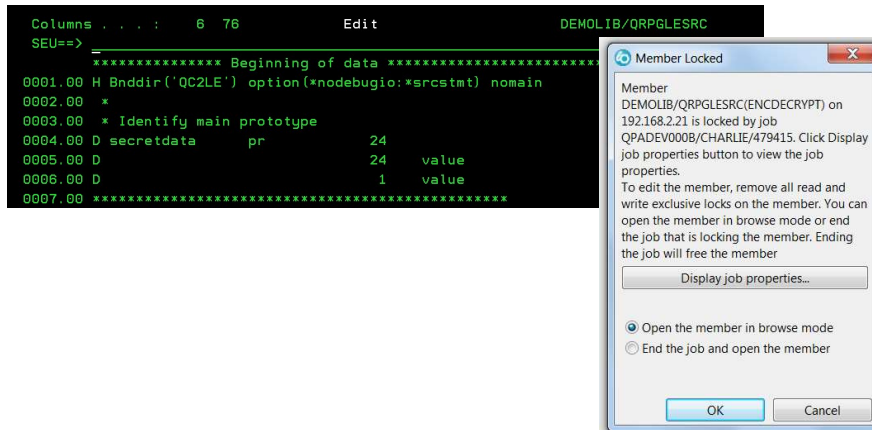
Opening up source member ENCDECRYPT in LPEX

- **Can select "Open with LPEX" or just double click on member**
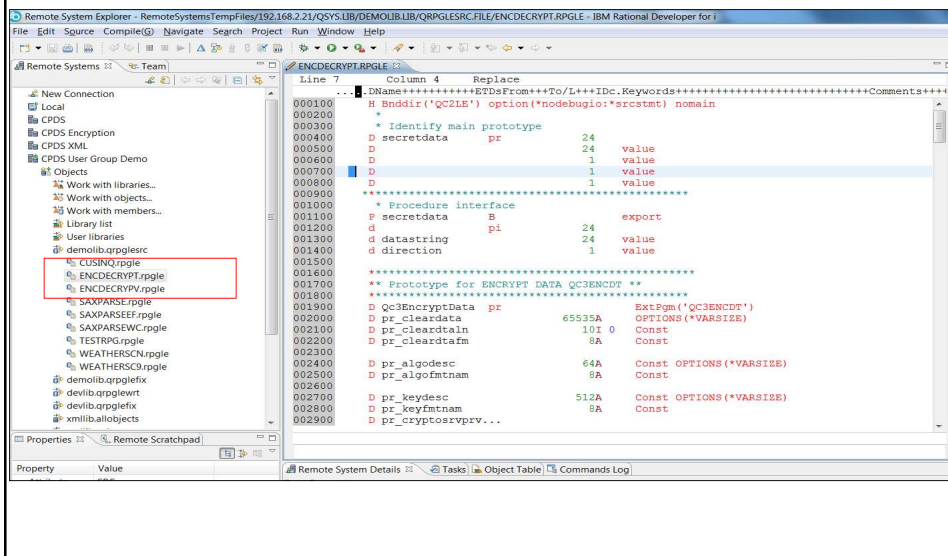
Alternate method of opening a source member

- · **Don't always need a filter to edit / browse a member**
- · **Can also create a new connection on the fly**
- · **Press CTRL-ALT-Q to display this window**
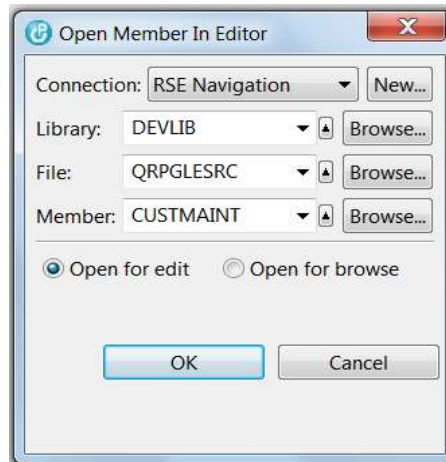
Help is always available at your fingertips

- · **All of the RPG manuals are loaded with RDi**
- · **Help is also cursor sensitive – more on this later!**
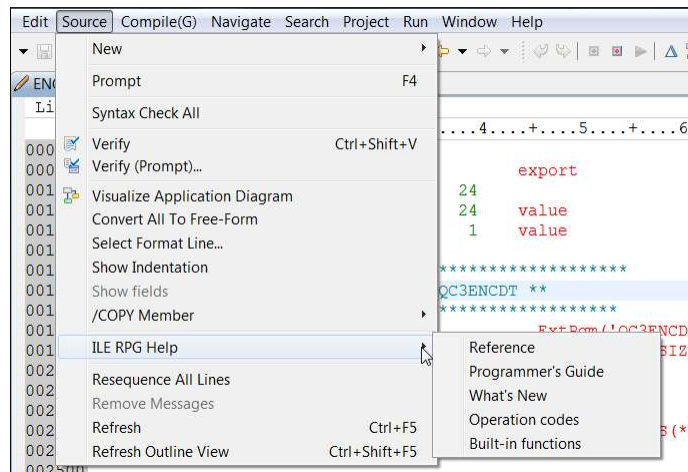
Help is always available at your fingertips

- **All of the RPG manuals are loaded with RDi**
- **Help is also cursor sensitive – more on this later!**
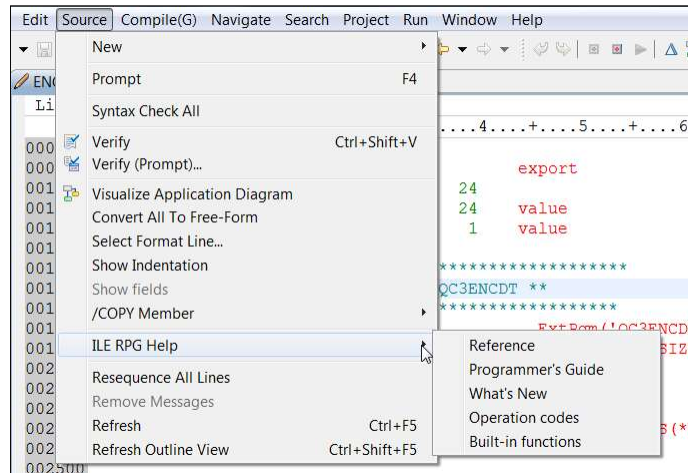
RDi (LPEX) shows extended information superior to SEU
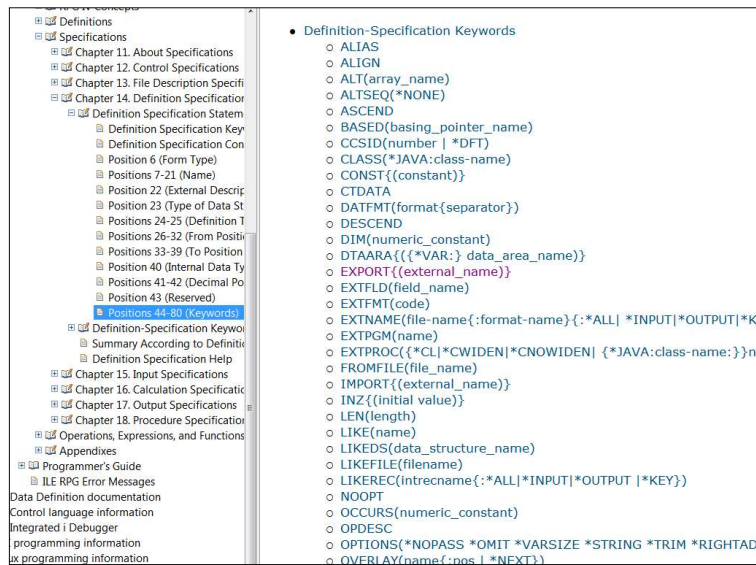
To FIND a string in LPEX, press CTRL-F

SEU example **FIND const**

Popular Keyboard Shortcuts

- CTRL-Home = SEU TOP

- CTRL-End = SEU BOTTOM

- CTRL-Z = Undo

- CTRL-Y = Redo

- CTRL-P = Print

- CTRL-A = Select All

- CTRL-Q and ALT-Q – Quick Mark

- F3 = Open Declaration

Source code filtering helps you quickly locate what you need

- **Right click anywhere in the source code**

When you select a block of code other options are available

- **Select the block of code, right click and select your function**

Other popular options available with selected blocks of code

- Upper to lower conversion and vice-versa
- Free-form code conversion

Block nesting displays arrows at start and end of logic blocks

- Position on start of block>Right click>Source>Show Block Nesting

Reformatting of nested code!

- **Position on start of block>Right click>Source>Format**

```
        if N_c3errm = *blanks and N_c3stat <> 'In Process';
        N_c3cmpl = '1';
        else;
v84up   N_c3cmpl = '0';
        endif;

        N_c3cmts = %timestamp();

                        if a = 1;
                         x = 5;
                        if b = 2;
```

|			
	Syntax Check Selection		Open Declaration	F3
	Convert Selection To Free-Form		Comment	Ctrl+/
💾 Save	Ctrl+S			
Cut	Ctrl+X		Format	Ctrl+Shift+F
Copy	Ctrl+Insert		Show Block Nesting	Ctrl+Shift+O
Paste	Ctrl+V		Select Block	Ctrl+M
Select	▸		Jump To Block End	Ctrl+Shift+M
Selected	▸		Select Format Line...	
Deselect	Alt+U		Hex edit line	Alt+Shift+H
Filter view	▸		Show Date Area	
Show all	Ctrl+W			
Source	▸			
View	▸		24 changes	

Source formatting

BEFORE

```
if hours > 40;
overtime = hours - 40;
overtimepay = overtime * overtimerate;
if hours > 50;
overtimepay = overtimepay + bonus;
endif;
gross = regularpay + overtimepay;
else;
grosspay = regularpay;
endif;
```

AFTER

```
if hours > 40;
  overtime = hours - 40;
  overtimepay = overtime * overtimerate;
  if hours > 50;
    overtimepay = overtimepay + bonus;
  endif;
  gross = regularpay + overtimepay;
else;
  grosspay = regularpay;
endif;
```

Block commenting

- **Position on start of block>Right click>Source>Comment**
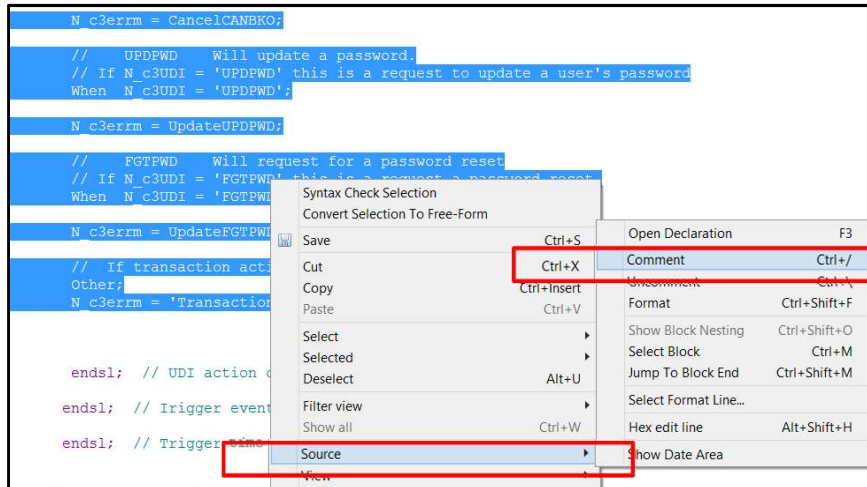
Cursor sensitive prompting and help features are available

- **Press F4 on any line to display the source code prompter**
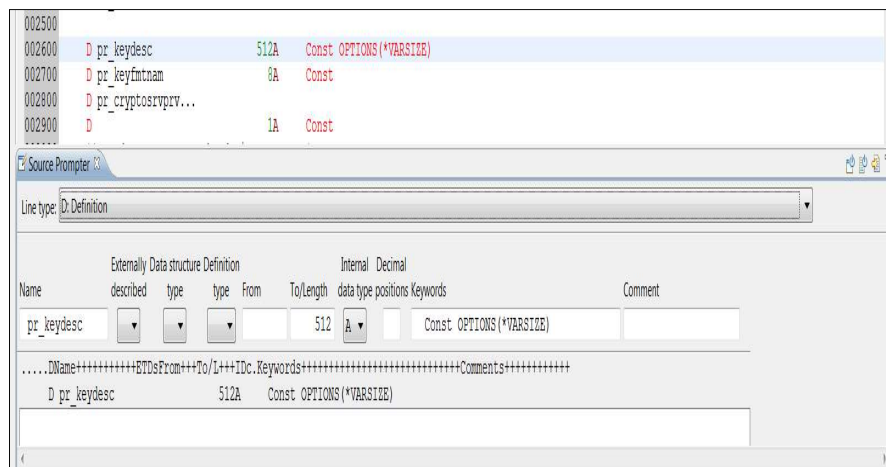
Cursor sensitive prompting and help features are available (cont.)

- **Position the cursor in the spec press F1 for built-in help text**
  - **This is the help text for the D-spec keywords section**

Hyperlinking – A quick way to change a variable definition

Hover over any variable and LPEX displays the attributes

```
// *********************************************
     begsr encryptroutine;
// *********************************************

     cleardata = *blanks;
     cleardat cleardata : Character (24)
     encdata = *blanks;
     datastringlen = %len(%trimr(cleardata));
```

Press and hold CTRL and the field turns into a hyperlink

```
// *********************************************
     begsr encryptroutine;
// *********************************************

     cleardata = *blanks;
     cleardat cleardata : Character (24)
     encdata = *blanks;
     datastringlen = %len(%trimr(cleardata));
```

Click on the link and you are immediately brought to the definition

```
     ** The clear data variable
D cleardata          S              24
```

Multiple active views of the same editable source member

- **Edit different sections of the SAME open source member**
  - **Source>Select New Editor – or – press CTRL-2**

Dirty source code editors

- **An asterisk next to the member name indicates unsaved changes**
  - **Very useful when several members open at once**
  - **If restored to original state or saved, no longer dirty editor**

Content Assist helps you fill in the blanks

- **Ideal for built in functions (bifs)**
  - **Type a portion of the bif and press CTRL-SPACE**
  - **Also works with your own procedures**

```
25900        // *********************************************
26000           begsr getblockof8length;
26100        // *********************************************
26200        //  This routine will receives a random field length and
26300        //  returns a value which is a multiple of 8.
26400        //  Field name is datastringlen.
26500
26600           if  datastringlen <= 8;
26700              datastringlen = 8;
26800           elseif datastringlen <= 16;
26900              datastringlen = 16;
27000           elseif datastringlen <= 24;
27100              datastringlen = 24;
27200           endif;
27300
27301           a = %sub
27302
27303
27304
27305
27400        end
27500
27600        /end-free
27700        P
```

RNF0604E The token

- %subarr(array:start-index|numeric-field)
- %subarr(array:start-index|numeric-field:nu
- %subdt(date|time|timestamp:*ms|*s|*mn|*
- %subst(string:start-index)
- %subst(string:start-index:length)

%SUBST(string:start-index:length)

%SUBST (Get Substring) returns a portion of argument string.
It may also be used as the result of an assignment with the EVAL operation code.

Content Assist is also aware of RPG statements

- **Press CTRL-Space with or without statement starting letters**

```
022000
022100           Qc3EncryptData(cleardata: cleardatalen :'DATA0100':
022200           pgmalgorithm : 'ALGD0200' :
022300           pgmkeyds : 'KEYD0200' :
022400           cryptosrvprv : cryptodevice :
022500           encdata : %size(encdata) :
022600           returndatalen :
022700           apiError );
022701
022702           x
022703
022704
022705
022706
022707
022708
022709
022710
022711
```

- xml-into
- xml-sax

XML-INTO

The XML-INTO operation reads the data from an XML document directly into variable or into an array parameter that it passes to the procedure specified by handler. Various options may be specified to control the operation. The first operand specifies the target of the parsed data. It can contain a variable name or the %HANDLER built-in function. The second operand contains the %XML builtin function specifying the source of the XML document and any options to control how the document is parsed. It can contain XML data or it can contain the location of the XML data. The doc option is used to indicate what this operand specifies.

Content Assist helps you complete calls to subprocedures

- **Press CTRL-Space with or without statement starting letters**

The Outline view displays all you need to know about a program

- **Updates in Real-Time**
- **Now includes ability to subset for even better navigation**

Source verification – Out of the park!

- **Source>Verify – or – Shift-Ctrl-V**
  - **Simulates the compiler but doesn't create an object**
  - **Can be used while offline**
  - **Double click on an error and RDi goes right to the line**

Source verification - removing error messages

RDi will insert error messages directly into your source code.
Pressing CTRL-F5 will remove them when you're ready.

BEFORE:

AFTER:

RPG Procedure Wizard removes the complexity

- **Source>New>Procedure**
  - **Creates the procedure interface as well as the prototype**

D-Spec wizard – Never have to remember the keywords!

- **Source>New>D Specification**
  - **Used for standalone fields, named constants, program-described, externally-described, program-status and file-information data structures**