- **Royal Cyber**

- **Development Modernization**

- **Newsletter**

"Technical Content For A Rational World"

In this issue:

- *COBOL Standards Checking with RDz Code Review Part 2 - Chris Leland*

- *Where in the world is that RDz plug-in? – Bill Napolitano*

- *The RDz Top 10 "How To" Mysteries Explained – Chris Leland*

- *The Royal Cyber Rational EM "App Store"*

- *Royal Cyber @Innovate*

- *Royal Cyber in the News*

COBOL Standards Checking – RDz's "Code Review" Feature – Part 2

*This article is the second in a two-part series that explores the use of RDz's "Software Analyzer" for analyzing **COBOL**** Performance and Maintainability.*

By Chris Leland, Royal Cyber

L ast issue we introduced you to RDz's Software Analyzer/Code Review feature – which provides an automated approach to COBOL, PL/I and Java standards checking for performance. In the last issue we discussed how important standards checking was – especially for reviewing ways to optimize your COBOL, PL/I and Java source. In this issue we will finish off code review by discuss three topics:

1. Parameter Driven Rules Customization
2. Batch Code Review Execution
3. Completely Custom Rules

Full disclosure, in this issue we will focus primarily on Parameter Driven Rules and Batch Execution. Completely Custom Rules development and deployment is a huge topic, and probably best learned by utilizing expert help. Royal Cyber specializes in developing custom code review rules - and you can learn more about what we've done and see a few samples by checking out our offerings on the EM App Store; http://www.royalcyber.com/em-app-store/

Parameter Driven Rules Customization

The out-of-the-box default (IBM-supplied) come in two flavors – 1. those that check for the existence or non-existence of something in your code you want highlighted when running Code Review – and 2. Code Review rules that require you to define parameterized input to the rule. The first category was covered in the last issue – so please refer to Newsletter Four – for details.

The second category (parameteter driven rules) is accessed from: Window > Preferences > Software Analyzer > Custom Rules and Categories – as shown in figure 1 below.

From this dialog you can see that you can:

- **Define new Categories** – to organize your rulesets by – well – anything that would be meaningful; Business Areas, Technologies (Hogan COBOL Rules, DB2 SQL Rules, etc.)
- **Define new Rules** – which extend the standards rules for COBOL, PL/I and Java we looked at in the last issue.

Defining new Categories

Adding a new custom ruleset is a simple a matter of Clicking:

- Selecting a language and clicking; Add Category...
- Typing the Category name into a dialog (see figures 2 and 3 below).

You can see from the screen captures that any named rule (including rules with special characters) are allowed.
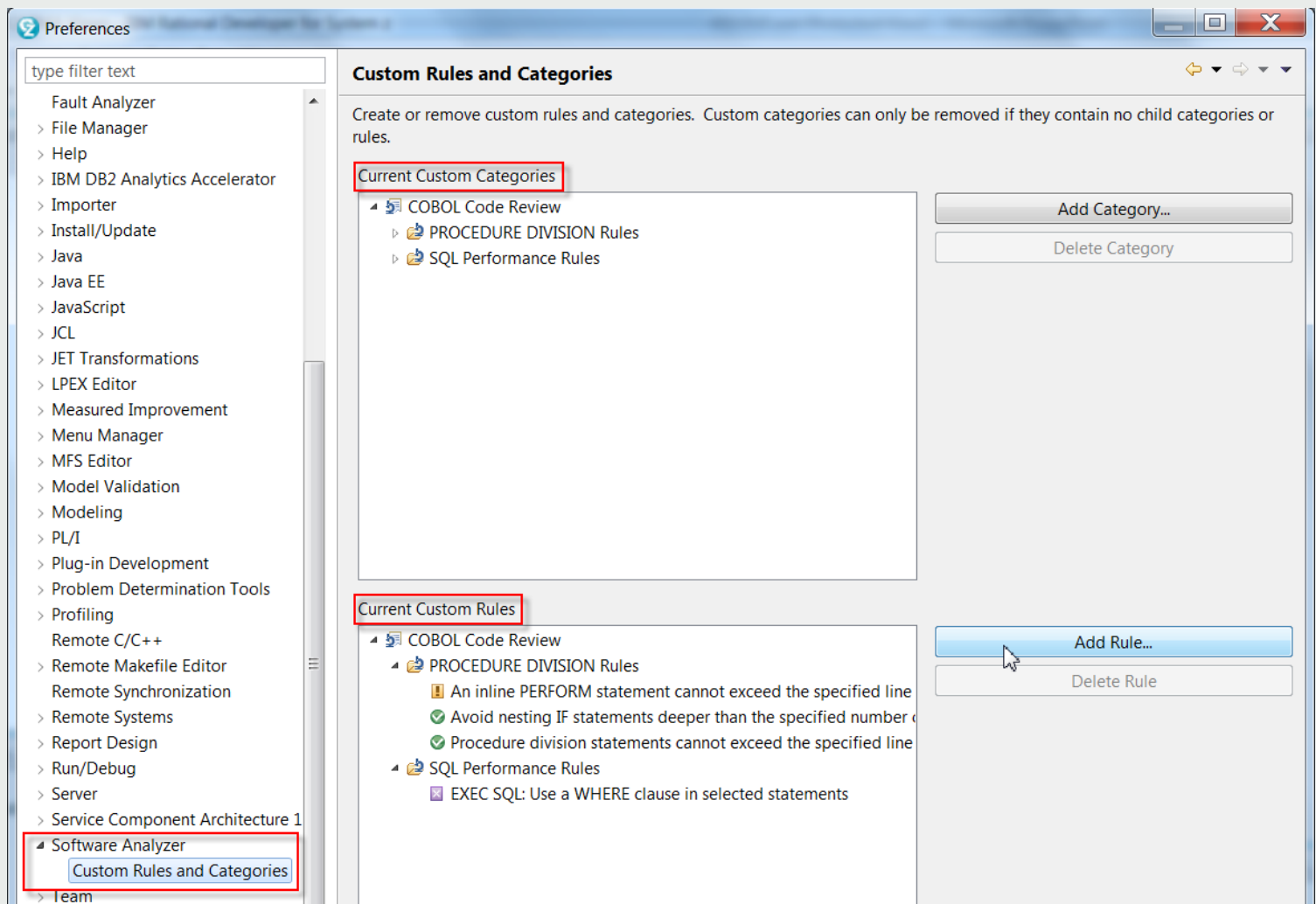
Figure 1. The Software Analyzer Custom Rules and Categories dialog

Defining new Custom Rules

Adding new rules is a 4-part process:

1. From Custom Rules and Categories click Add Rule... Figure 1
2. Select the (Parent) Category the new rule belongs to – note that you can add new Custom Rules to any category, not just new Custom Categories - Figure 4
3. Define the parameters for the Custom Category (Figure 5) and select the severity to assess for breaking the rule (Recommendation, Warning or Severe) – Figure 6
4. Add new Parameter Driven rules to an existing Ruleset – Figure 7.

Note from Figure 5 that there are quite a few high-value Parameter Driven rules – for COBOL, such as:

- Determining an acceptable level of IF Nesting and Inline Perform
- Catching specified SQL statements that do not have a WHERE clause
- Finding Paragraphs and Sections that are not commented
- Enforcing naming conventions for conditionals and file names

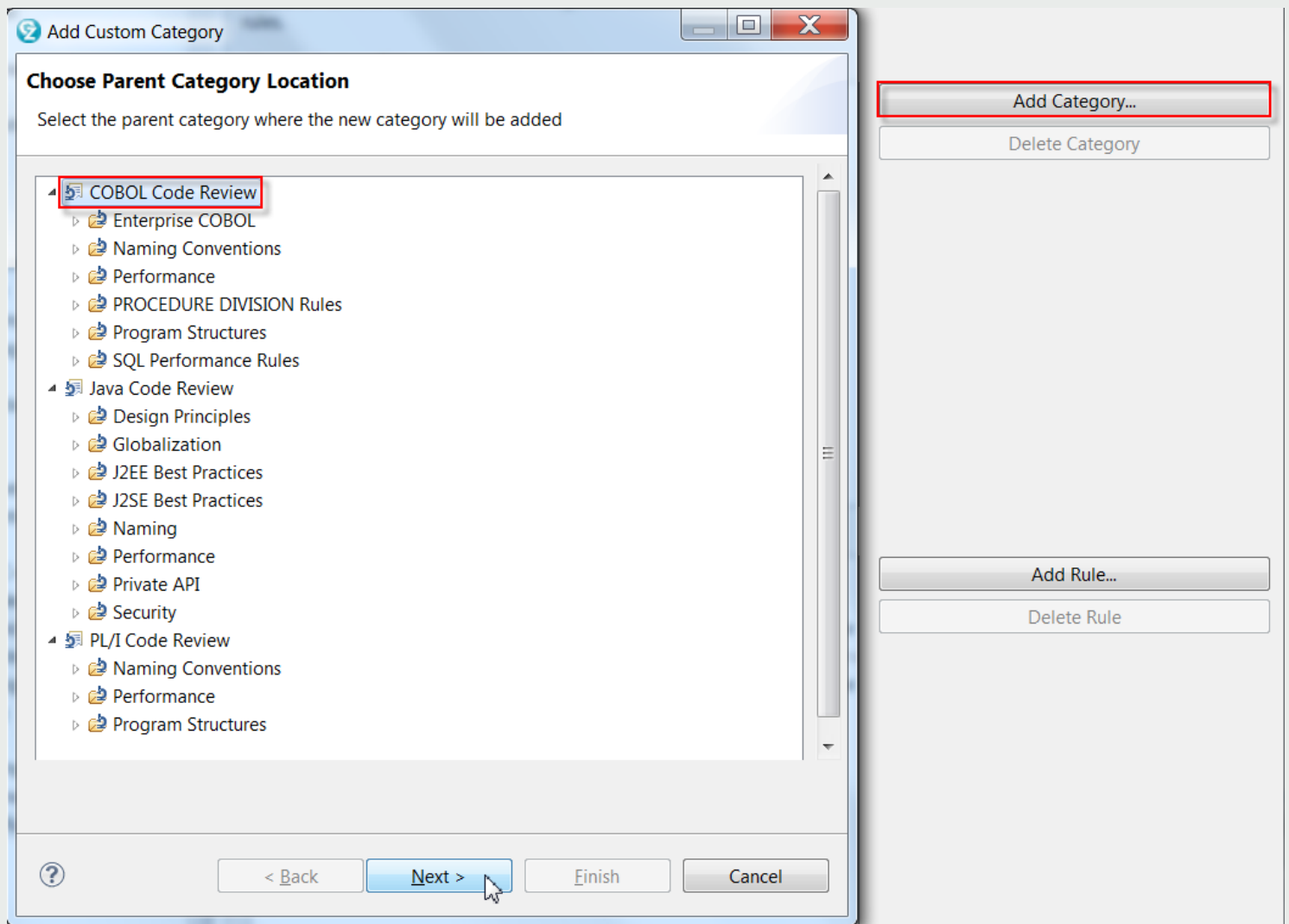Similar quality Parameter Driven rules exist for PL/I and Java.
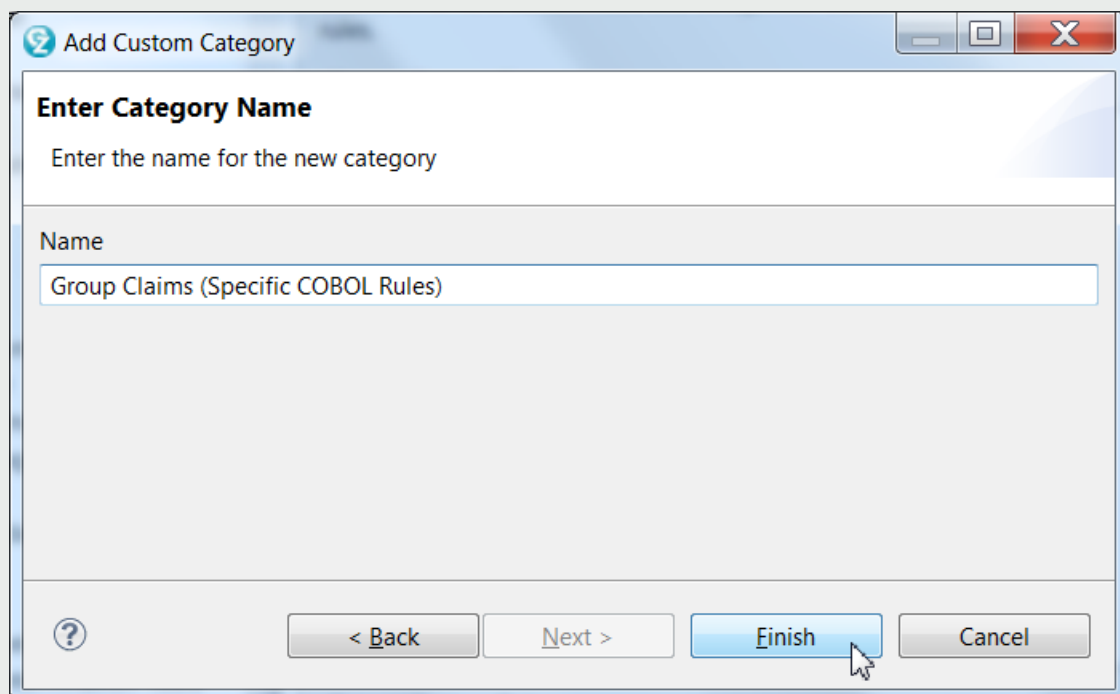
Figure 2. Adding a Rule Category – part 1

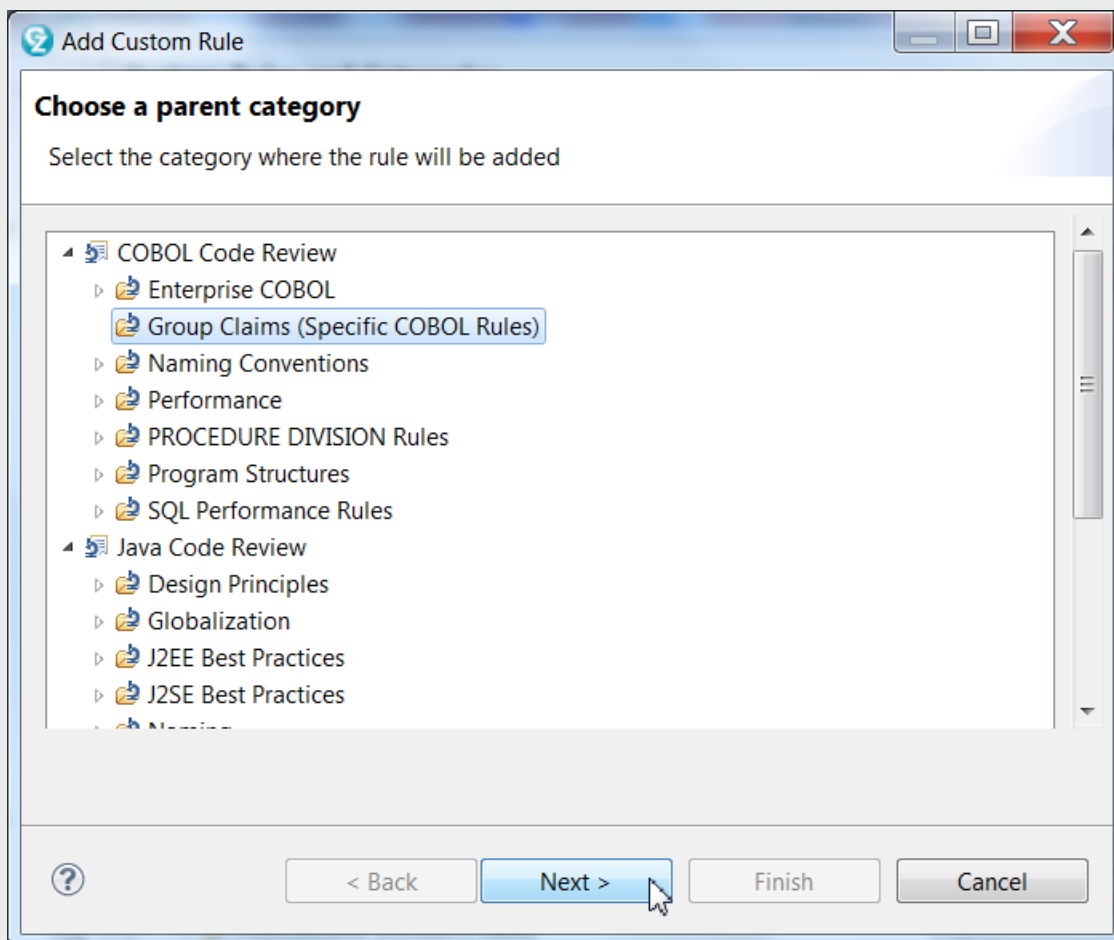**Figure 3. Adding a Rule Category – part 2**

Figure 4. Adding a new Rule – part 1

While everything is not shown above – Figure 4 describes the RDz Custom Rules Categories:

- Language (COBOL, Java and PL/I):
    - Existing Ruleset categories
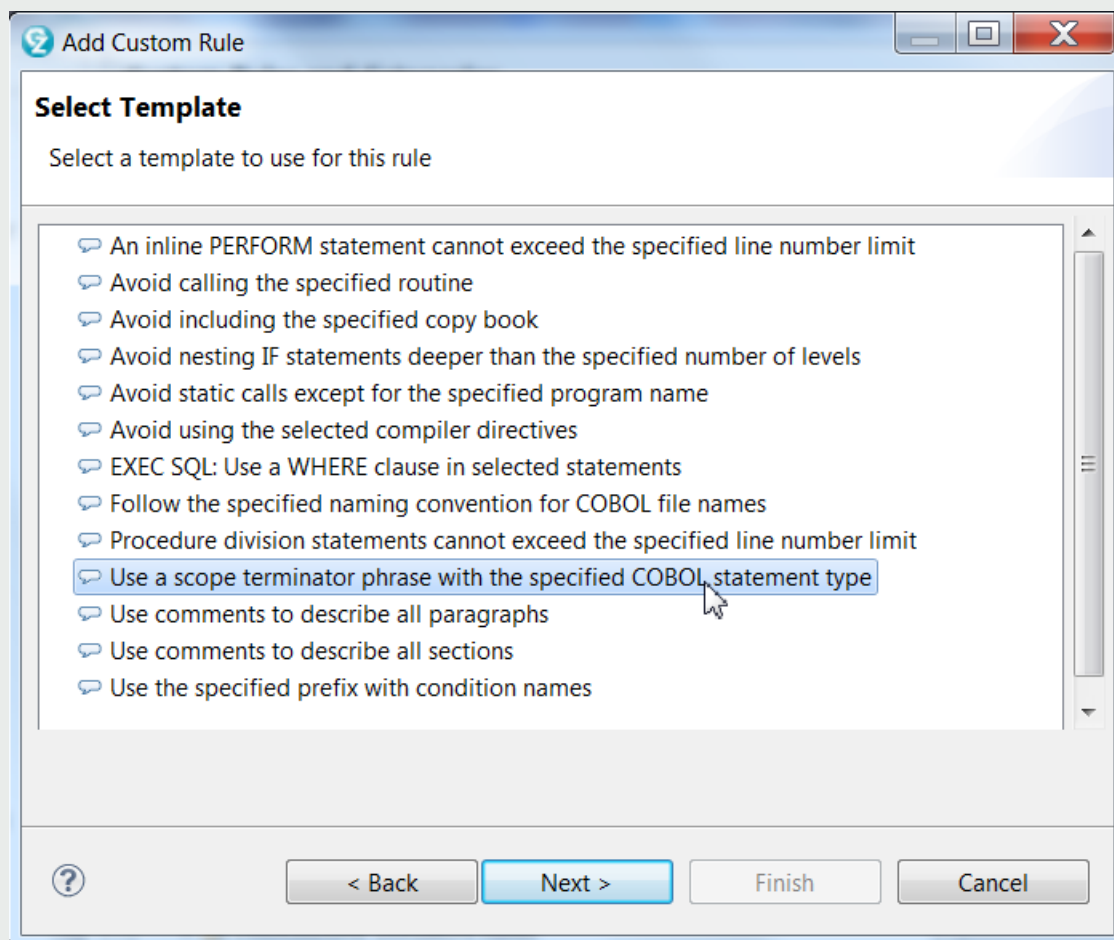   - Any custom Ruleset categories you've defined
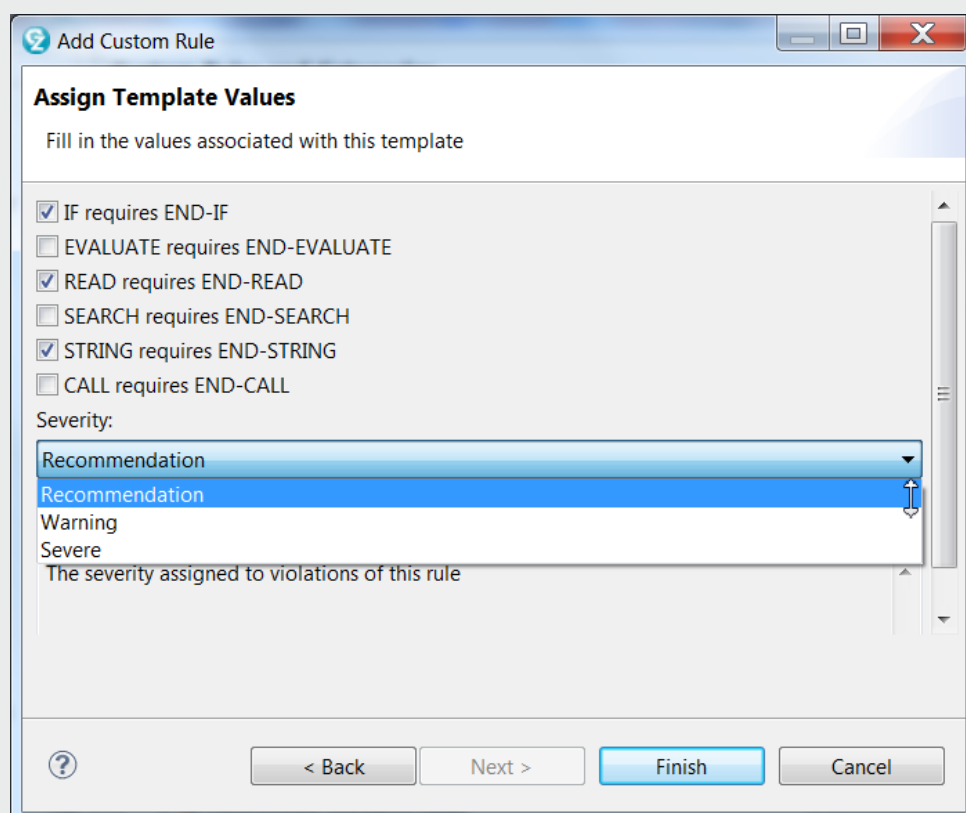
Figure 5. Adding a new Rule – part 2

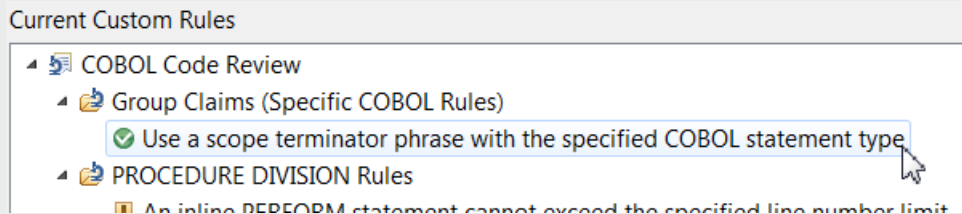**Figure 6a. Adding a new Rule – part 3**

Current Custom Rules

- COBOL Code Review
  - Group Claims (Specific COBOL Rules)
    - ✓ Use a scope terminator phrase with the specified COBOL statement type
  - PROCEDURE DIVISION Rules
    - An inline PERFORM statement cannot exceed the specified line number limit

Figure 6b. Parameter driven rule added

Software Analyzer Configurations

Create, manage, and run configurations

Select a range of inputs from the scope tab and at least one rule from the rules tab to create an analysis configuration.

type filter text

- Software Analyzer
  - Claims Processing Rules
  - Development Rulesets
  - Production Level Rules
  - QA Code Review Level Rules

Name: Claims Processing Rules

Scope | **Rules**

Rule Sets: [ ] Set Import... Export...

Analysis Domains and Rules

- ✓ COBOL Code Review [20/50]
  - ✓ Enterprise COBOL [2/2]
  - ✓ Group Claims (Specific COBOL Rules) [1/1]
    - ✓ Use a scope terminator phrase with the specified COBOL statement type
  - ✓ Naming Conventions [1/1]
  - ▪ Performance [5/9]
  - ✓ PROCEDURE DIVISION Rules [3/3]
  - ▪ Program Structures [7/33]

Properties

- ✓ IF requires END-IF
- ☐ EVALUATE requires END-EVALUATE
- ✓ READ requires END-READ
- ☐ SEARCH requires END-SEARCH
- ✓ STRING requires END-STRING
- ☐ CALL requires END-CALL

Filter matched 5 of 5 items

Apply   Revert
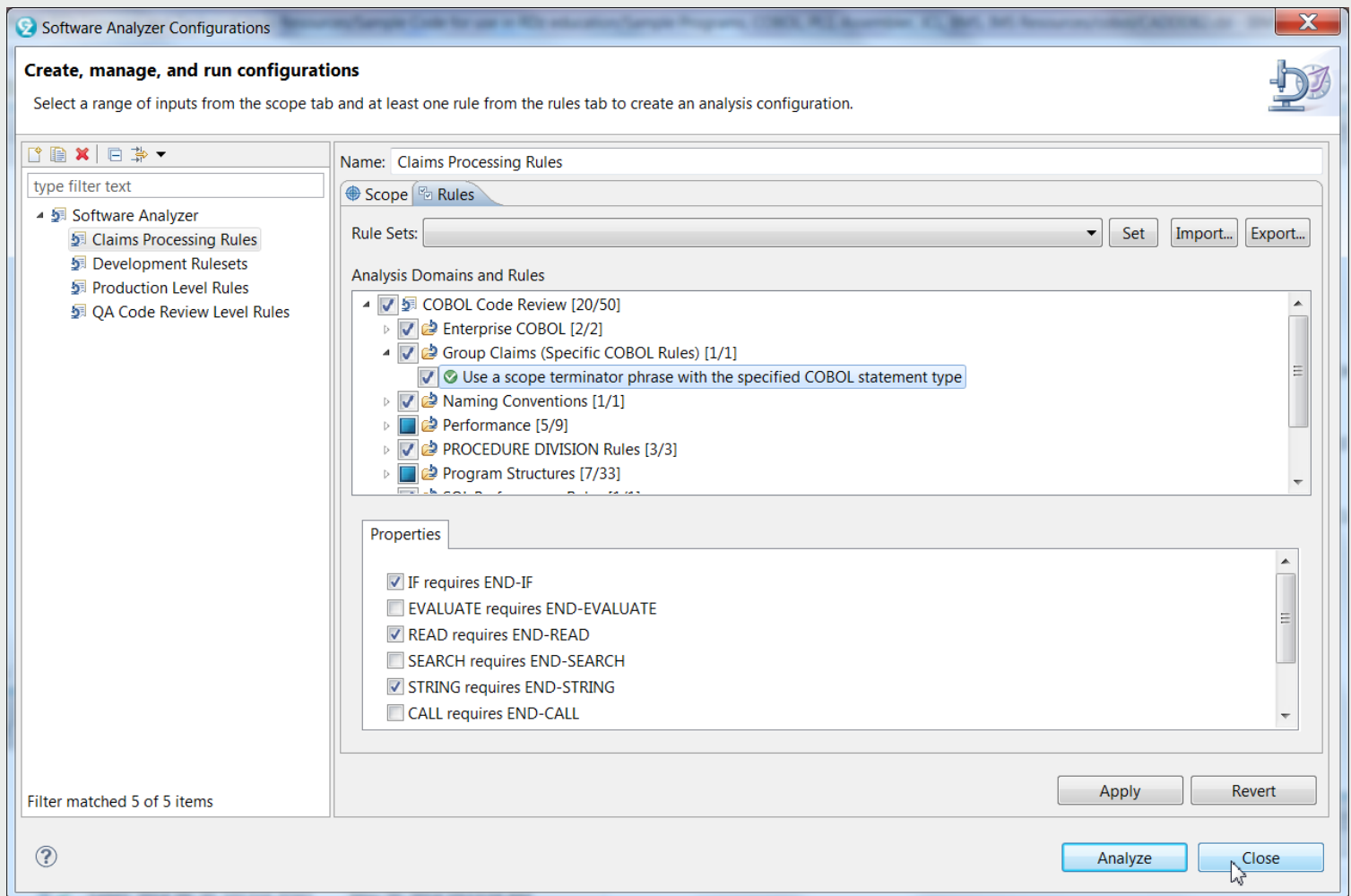
Analyze   Close

Figure 7. Adding a new Rule – part 4

From Figure 7 you can see that, after editing the Software Analyzer Configuration, you can select your new Parameter Driven Rule(s) – and Rule Categories. After you've selected the rules you want to run against your code: Click **Close**, and at "**Save changes**?" click: **Yes**.

(Review) Running the Code Review Rules

When you've saved your changes and are ready to run (again - assuming that you're in Edit) select:

Software Analyzer > Claims Processing Rules   *(or the name of your Configuration)*

This will invoke Code Review on your program. Code Review analyzes your program and flags any statement that violates one or more of the rules you've selected with a Code Review annotation (see **Figures 11 and 12**).
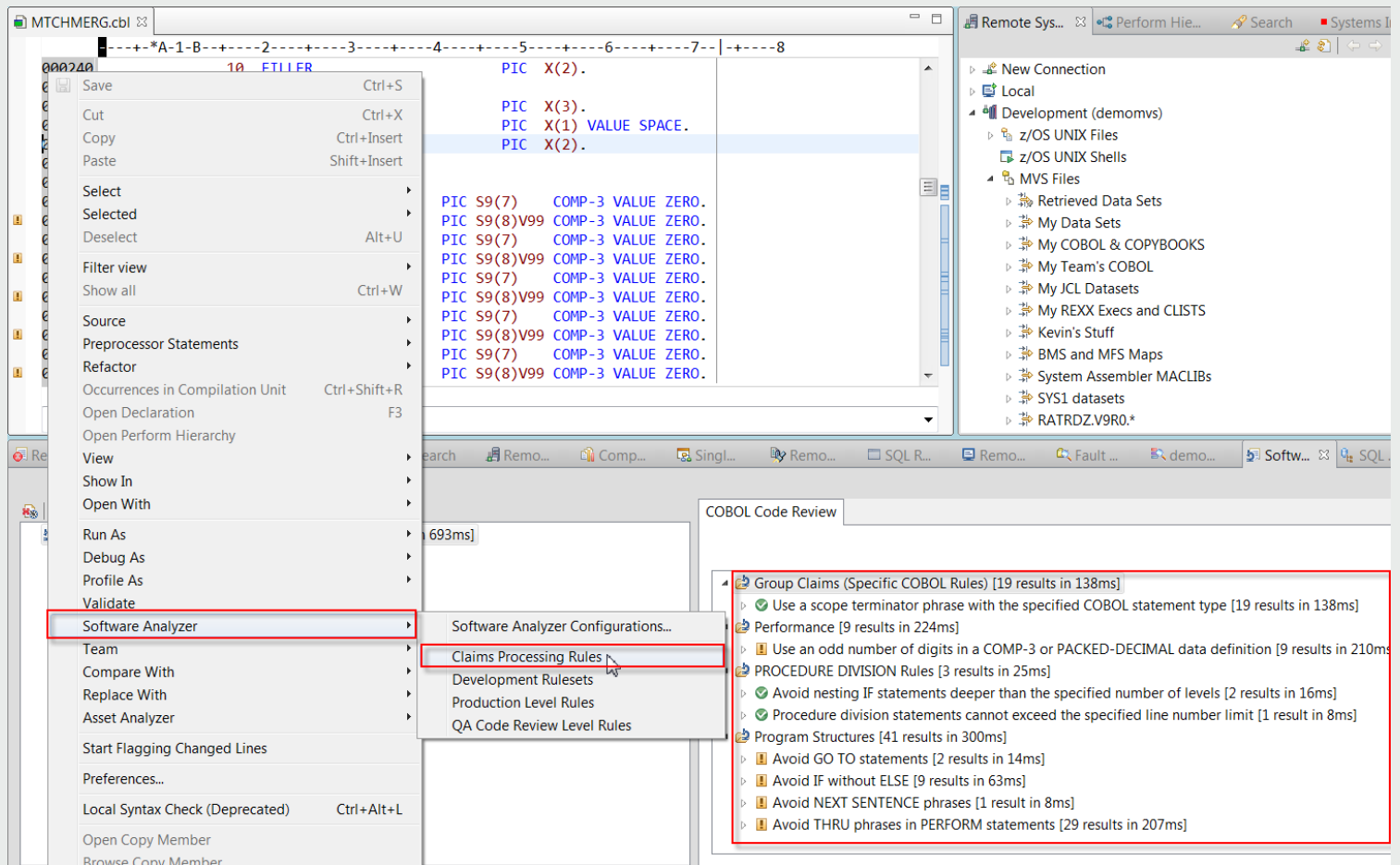

Figure 11. Rule Violation annotations – and hyperlinks from a COBOL Code Review view

Running Code Review against an entire PDS in one action

Interactive Code Review runs locally against individual programs opened in either the LPEX, COBOL, PL/I or Java editors. However, it can also run against an entire TSO library (PDS) if you download the library to a Local Workstation Project (you can drag & drop or copy & paste). This will create a folder in your Workspace. You can then Right-click over the entire folder, and from the Context Menu select: Software Analyzer and the Ruleset you wish to run against the PDS (See Figure 12).
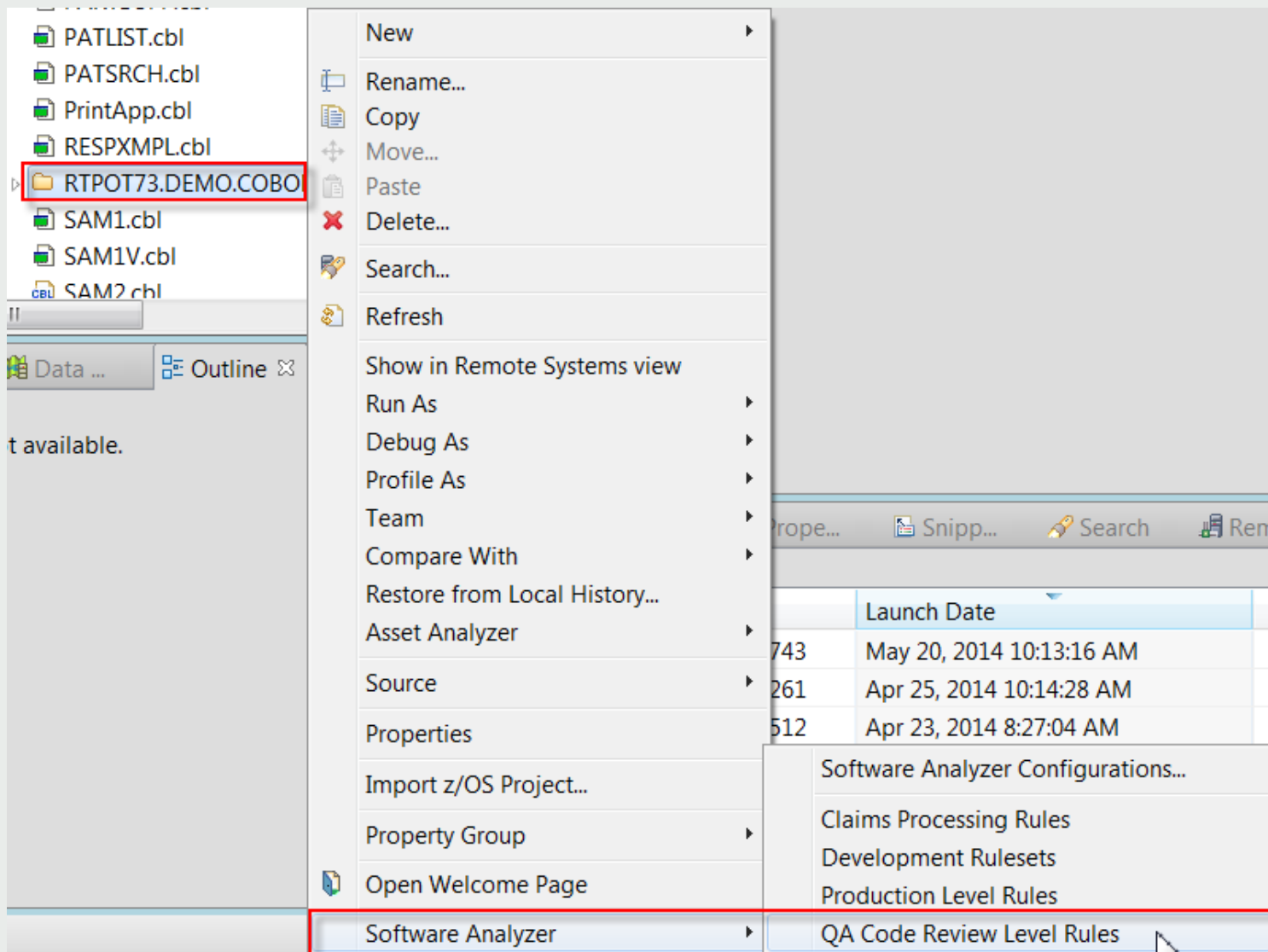
Figure 12. Running Code Review on an entire PDS (source library)

Batch (JCL-driven) Code Review

Code Review prior to RDz v9.0 was both interactive, and based on the West Point "honor system" – in that, if your program statements broke a Code Review rule you might (or you might not) be interested enough in the results to change code that is flagged. In other words there was way to enforce conformity to standards. Also Code Review prior to v9.0 was entirely an IDE-based/manual process, and many shops these days are looking forward towards the "continuous delivery" app-dev model – where, as part of a check-in and Build process, Code Review iruns and reports are generated showingi the general health of the program and level of standardization achieved. Enter RDz v9.0 and Code Review running in Batch.

Starting in Q3 / 2013 RDz's v9.0.1 allowed Batch-driven Code Review – where a REXX is called from MVS JCL and the REXX in turn access various z/OS UNIX files. IBM ships a PROC for this; AKGCR – which ships in the AKG.SAKGSAMP files. Your Systems Programmer will have to decide how to configure the samples for this as AKG.SAKGSAMP is typically not in JES's default search path.

The REXX returns a single over-loaded MVS Return Code (see Figure 13) that describes both the status of the process and the highest Code Review error level (read the comments from the PROC to ascertain what happened during the run).

```
//* The tool can end with following return codes:                          *
//* -  0: processing completed successfully                                 *
//*        there are no code review rule violations                         *
//*        the generated code review output is valid                        *
//*        the (optional) post-processing script executed successfully *
//* -  1: processing completed successfully                                 *
//*        the most severe review violation is a recommendation (4)   *
//*        the generated code review output is valid                        *
//*        the (optional) post-processing script executed successfully *
//* -  2: processing completed successfully                                 *
//*        the most severe review violation is a warning (8)          *
//*        the generated code review output is valid                        *
//*        the (optional) post-processing script executed successfully *
//* -  3: processing completed successfully                                 *
//*        the most severe review violation is a severe violation (12) *
//*        the generated code review output is valid                        *
//*        the (optional) post-processing script executed successfully *
//* -  4: warning, non-critical issues were encountered                     *
//*        the state of the code review violations is unknown          *
//*        the generated code review output is valid                        *
//*        the (optional) post-processing script executed successfully *
//* -  5: the most severe post-processing message is a warning (4)     *
//*        the state of the code review violations is unknown          *
//*        the state of the generated code review output is unknown    *
//* -  6: the most severe post-processing message is an error (8)      *
//*        the state of the code review violations is unknown          *
//*        the state of the generated code review output is unknown    *
//* -  7: the most severe post-processing message is a failure (12)    *
//*        the state of the code review violations is unknown          *
//*        the state of the generated code review output is unknown    *
//* -  8: error, critical issues were encountered                           *
//*        the generated code review output is not valid                    *
//* - 12: system or environment error                                       *
//*        the generated code review output is not valid                    *
//*                                                                         *
```

Figure 13. Batch Code Review MVS Return Codes

Running Code Review in Batch

To run Code Review from JCL you need to (minimally):

- Find and customize the AKGCR JCL PROC (see Figure 14)
- Find where your Java home_directory is under UNIX z/OS
- Transfer your Code Review Rules to z/OS UNIX:
    - Export the Rules to a PC File
    - Import the Rules to a Local Workstation project
    - Drag & Drop the Rules to a z/OS UNIX directory where you have R/W access
- Transfer your Property Group file to z/OS UNIX:
    - Export the Property Group to a PC File – ending on .xml
    - Import the Property Group file to a Local Workstation project
    - Drag & Drop the Property Group file to a z/OS UNIX directory where you have R/W access

Over-ride and/or validate DD statements in the AKGCR Proc for:

- JAVA
- PDS – which names the program you wish to run Batch Code Review against
- RULES – point to the UNIX directory where you've transferred your custom rules
- PROPERTY – point to the UNIX direectory where you've transferred your Property Group

```
//AKGCR    PROC PRM='',                      * input overrides
//             PATH='/rc/v90/usr/lpp/rdzutil',
//             RULE='/shared/test/cr',
//             HLQ='RC001'
//*
//AKGCR    EXEC PGM=IKJEFT01,REGION=0M,PARM='%AKGCRX &PRM'
//SYSPROC  DD DISP=SHR,DSN=&HLQ..SAKGPROC
//SYSTSIN  DD DUMMY
//SUMMARY  DD SYSOUT=*,RECFM=VB,LRECL=2051
//CSV      DD SYSOUT=*,RECFM=VB,LRECL=2051
//XML      DD SYSOUT=*,RECFM=VB,LRECL=2051
//WORKSPCE DD SYSOUT=*,RECFM=VB,LRECL=2051
//MSGS     DD SYSOUT=*,RECFM=VB,LRECL=2051
//ERRMSGS  DD SYSOUT=*,RECFM=VB,LRECL=2051
//PMSGS    DD SYSOUT=*,RECFM=VB,LRECL=2051
//PERRMSGS DD SYSOUT=*,RECFM=VB,LRECL=2051
//PATH     DD DUMMY SYSOUT=*,RECFM=VB,LRECL=2051
//SYSTSPRT DD SYSOUT=*
//PDS      DD DISP=SHR,DSN=RC0001.TEST.COBOL(WARDRPT)
//MEMBERS  DD DUMMY
//HOME     DD PATH='&PATH.'
//JAVA     DD PATH='/usr/lpp/java/J6.0'
//TMPDIR   DD PATH='/tmp'
//RULES    DD PATH='/u/rc0001/claims_rules.dat'
//CUSTRULE DD PATH='&RULE./custom.rules.ccr'
//PROPERTY DD PATH='/u/rc0001/db2work.xml'
//EXTMAP   DD PATH='&RULE./extension.mappings.zip'
//SCRIPT   DD DUMMY PATH='&PATH./samples/cr.post.sh'
//OPTIONS  DD DUMMY *          z/OS 1.13 supports instream data in PROC
```

Figure 14. Batch Code Review MVS Return Codes

There are many additional options for running Batch Code review – including the integration or use of your own Custom-created Code Review rules (next topic in this Newsletter), reviewing multiples PDS members in one run, providing the Batch process with custom z/OS File System Mapping and others described in Figure 15.

```
//* SYSTSPRT                              TSO output, required *
//* - DD definition only                                      *
//* - holds the log of the tool that triggers the code review *
//*                                                            *
//* PDS                             PDS holding source, required *
//* - when concatenating multiple DDs, only the first one is used *
//* - must reference a PDS(E)                                  *
//* - when DSN(MEMBER) is provided, and MEMBERS is empty, then *
//*   MEMBER will be used as member filter                     *
//*                                                            *
//* MEMBERS                          member filter, optional (all) *
//* - when concatenating multiple DDs, all are used           *
//* - comma separated list of members to review               *
//*                                                            *
//* RULES                           *.dat, validation rules, required *
//* - when concatenating multiple DDs, only the first one is used *
```

```
//* - must reference a z/OS UNIX file                       *
//*                                                         *
//* PROPERTY                       *.xml, property group, required *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX file                       *
//*                                                         *
//* HOME                          base directory for code, required *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX directory                  *
//*                                                         *
//* JAVA                          base directory for Java, required *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX directory                  *
//*                                                         *
//* TMPDIR          base directory for workfiles, optional ($HOME) *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX directory                  *
//*                                                         *
//* SCRIPT                      post processing script, optional () *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX file                       *
//*                                                         *
//* EXTMAP                    *.zip, extension mapping, optional () *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX file                       *
//*                                                         *
//* CUSTRULE            *.ccr, custom validation rules, optional () *
//* - when concatenating multiple DDs, only the first one is used  *
//* - must reference a z/OS UNIX file                       *
//*                                                         *
//* CODEPAGE                       convert UTF-8, optional (IBM-037) *
//* - when concatenating multiple DDs, all are used         *
//* - only first word of provided data is used              *
//*                                                         *
//* OPTIONS            additional Java startup options, optional () *
//* - when concatenating multiple DDs, all are used         *
//*                                                         *
//* ENVVARS          additional environment variables, optional () *
//* - when concatenating multiple DDs, all are used         *
//*                                                         *
//* CFG        alternative method to provide input data, optional () *
//* - when concatenating multiple DDs, all are used         *
//*                                                         *
//* SUMMARY               content of Analysis Summary.xml, output *
//*                                                         *
//* CSV                  content of CSV Analysis Results.csv, output *
//*                                                         *
//* XML                  content of XML Analysis Results.xml, output *
//*                                                         *
//* WORKSPCE                     content of workspace.log, output *
//*                                                         *
//* MSGS                            content of stdout.log, output *
//*                                                         *
//* ERRMSGS                         content of stderr.log, output *
//*                                                         *
//* PMSGS                       content of post.stdout.log, output *
//*                                                         *
//* PERRMSGS                    content of post.stderr.log, output *
//*                                                         *
//* PATH                            content of path.log, output *
```

Figure 15. Additional DD Card Customization and Options for Batch Code Review

Batch Code Review Outputs

Outputs from Batch Code review include Log files and the same CSV (spreadsheet format) and XML output summaries as are produced. Figure 16 shows a series of XML results which could be input into any XML reporting product or hardcopy utility.

Figure 16. Batch Code Review Output Report – Generated XML

Custom Code Review Rule Development

As soon as IBM announced support for Code Review customers began asking for the ability to "roll their own" – Custom Code Review Rules. So in v9.0 IBM provided a Java/Eclipse wizard-driven development framework for creating custom COBOL Code Review Rules – and in RDz v9.1 IBM offered the same sort of process for custom PL/I Code Review Rules development and deployment. The Custom Code Review Rules are generated as standard Eclipse/Java plug-ins that must be installed (not copied… installed) into use.

As mentioned at the beginning of this article, this is a deep and complex topic – and set of steps. It pre-req's both the target language (COBOL or PL/I) and a deep understanding of how to create and deploy Eclipse/Java plug-ins. Typically one single person does not have both skills/experience to the technical depths necessary so shops often pair up a Java/Eclipse developer with a z/OS COBOL programmer to do the work.

And as mentioned earlier, Royal Cyber has a year+ of experience doing this work. It would make sense if you're interested in this – to contact us and find out how we can jumpstart your efforts: http://royalcyber.com/em-app-store/custom-cobol-code-review-plug-ins/

Next steps…

RDz is one of the primary strengths of Royal Cyber. In 2014 IBM chose Royal Cyber to manage its RDz "Distance Learning" program - out of all the RDz business partners world-wide. We offer deep and quality service engagements in every phase or RDz – from Installation and Deployment, to Training/Mentoring and adoption, to administering and supporting both the RDz client and server – and finally, to evaluating your RDz Adoption and Return-on-Investment.

- For details on the Royal Cyber RDz Rollout and project/task usage modeling contact us: inquires@royalcyber.com
- To sign up for Royal Cyber RDz Free Distance Learning: http://royalcyber.com/royal-cyber-rdz-distance-learning-training-schedule/
- To get a copy of the January 2014 Newsletter – including additional deep-dive articles on RDz and RAAi: http://royalcyber.org/mn/index.html

Chris Leland: Chris is Royal Cyber's primary RDz technical instructor - having worked full-time with RDz for almost four years; developing applications in COBOL, teaching RDz classes and mentoring in COBOL and Assembler, installing and configuring RDz, delivering custom workspace design sessions, writing an RDz evaluation exam, and integrating RDz with Rational Asset Analyzer. Chris passed the IBM/RDz Certification exam with flying colors in 2011.

Where in the world is my RDz Plug-in?
Finding that elusive plug-in for RDz

By Bill Napolitano

Lately we've gotten a number of requests by customers to pull together a table of RDz plug-ins and where to get them. Done. In the table below you will find RDz plug-ins from IBM, from Compuware and from CA. We do not claim that this table is complete – nor can we be 100% certain that every plug-in listed will work with your version of RDz. As one of my COBOL instructors once told me about the importance of testing, "It works when it works."   As bromides go … not bad.

Plug-in	Vendor	RDz version	Comments
Property Group Composer	Royal Cyber	V8.x V9.x	This plug-in automates the tedious, time-consuming and error-prone process of creating the SYSLIB entry for the RDz editors. It solves one of the most challenging aspects of rolling out RDz successfully.
RAAi	IBM	RAA v6.x RDz v8.5 and v9.x	This plug-in to RDz allows you to access the full capabilities of RAA for application analysis from inside RDz. See the Royal Cyber Newsletter #2 for details: http://webspherehatsguru.com/Royal-Cyber-Development-Modernization-Newsletter-Volume1-1.pdf The plug-in is part of the RAA DVD set. You can also obtain it through the standard IBM support site.
Debug Tool ** No plug-in is necessary. This functionality ships with RDz	IBM	All releases	No plug-in is necessary. This functionality ships with RDz
PD Tools plug-ins: • File Manager • Fault Analyzer • Application Program Analyzer • CICS Tools	IBM	PD Tools v13.1 RDz v8.5 RDz v9.x	This link is to the Installation Manager Note – if you are using RDz v8.5 and/or PD Tools v12 or v11 go here: http://www-01.ibm.com/support/docview.wss?uid=swg24036304
CA-Intertest	Computer Associates	Unknown	You will have to get this plug-in from your Computer Associates sales-rep. It is not available for download.
Compuware Suite: • Xpeditor • File-Aid • Abend-Aid	Compuware	RDz v8.x RDz v9.x	You will have to get this plug-in from your Compuware sales-rep. It is not available for download. If you need assistance with this, please contact Jim Ixxx from Compuware.
Serena Changeman ZMF Client Pack	Serena	V8.5 V9.0	You will have to get this plug-in from your Compuware sales-rep. It is not available for download.
CA-Endevor plug-in	Computer Associates	V9.0	RDz ships with a free framework for integrating with CA-Endevor called CARMA. But Compuware also ships its own RDz/Endevor integration software. Which you can obtain from your CA rep.

Bill Napolitano is a freelance technical writer with a 30-year background in z/OS application development and tooling.

The RDz **Top 10** "How To" and "Say What?!?!" Mysteries Explained

This article describes the most common issues and answers, and technical "how to" concerns from new-to-RDz Users .

By Chris Leland, Royal Cyber

I've been teaching RDz for 3 ½ years – and in that time have seen a lot of reactions, and been asked a lot of questions… some of them over and over again. So I thought, "Hey Chris… why not get ahead of this? Why not answer questions before they're asked. So – taking a page from the [Johnny Carson "Great Carnac"](#) routine, here's a list of the most common RDz questions (and answers):

1. Why is the Perform Hierarchy disabled/grayed out?
- There are typically two reasons this happens:
    i. Your program has not been mapped to COBOL (please read up on **z/OS File System mapping**)
    ii. There are syntax errors in your code that are so severe the RDz parser can't make sense of the semantics of your PROCEDURE DIVISION – and just does a Roberto Duran, "No mas!"
- Next level explanation of the RDz's approach to syntax checking…
    i. The RDz LPEX, COBOL and PL/I editors always maintain two copies of your program:
        1. The code you see (just like ISPF =1 and ISPF =2
        2. And a in-memory/parsed version of your program that RDz uses to provide "advanced tooling features" – such as the COBOL Perform Hierarchy, Program Control Flow diagrams (for COBOL and now for PL/I), Occurrences in Compilation Unit, Content Assist, etc. etc.
    ii. If the in-memory/parsed version of your code contains enough – or certain types of syntax errors (say you misspell PROCEDURE DIVISION?) – then advanced tooling is turned off – until you fix the syntax errors.
    iii. (But what if I'm accessing production code – which does **N O T** have syntax errors?
        1. Chances are that your Property Group file is not setup correctly;
            a. Either you're missing a SYSLIB entry
            b. Or you've not mapped the Property Group to the resource you're working with.
- There is a 3rd possibility for this problem – which I've seen in couple of shop's where they put some kind of flag or annotation in the PROCEDURE DIVISION telling the build JCL that the program is either batch, online or both. These kinds of programs needs special handling (see question #9 below).

2. How do I get ISPF functionality during my edit session?
- To edit your programs using ISPF functionality you must use the LPEX (not COBOL, not JCL and not the PL/I dedicated editors)
- Plus you need to make sure that your LPEX profile is set to **ispf**
    i. From Window > Preferences > LPEX Editor, set the Profile to ispf ➔
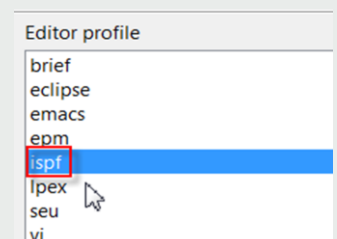
3. Why does RDz hang when I open a dataset (not a source file ... a data/dataset)?
- Most likely you're opening a large QSAM file – right? (RDZ doesn't edit VSAM or IMS Databases without help from the IBM File Manager plug-in).
- But (assuming you've said, "yes Chris… that's it!" – you have to do one more thing to avoid this problem.
- Map the dataset to a Workstation File Extension of **tdat**
    i. From the small picture below you can see that I'm accessing the z/OS File System mapping view (even though the view's full name is *ellipsed*)

ii. From this view you can right-click, and Add a Data Set Mapping to your DSN – that specifies **tdat** as the Workstation File Extension (and download the file in binary)

- The net results of doing this will be that RDz opens its dedicated QSAM file editor – displaying (by default) 200 records at a clip. Not hanging – and providing some very nice tooling.
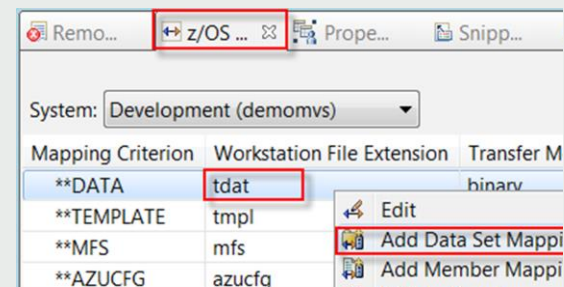
4. How does Block (rectangle) copy/paste work?

- Block copy works very well if you're using the new COBOL, JCL and PL/I editors
  - **i.** There's actually a toolbar option to select: "Block Selection Mode" toggling.
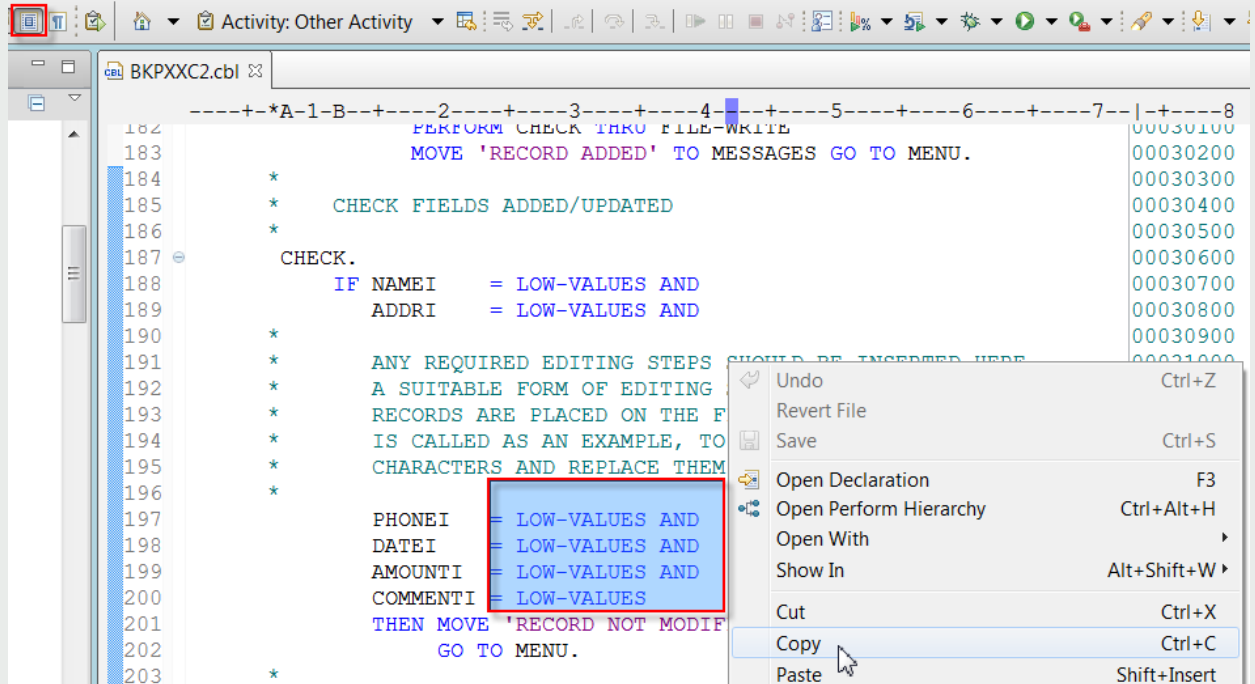
The COBOL Editor's "Block Selection" editing actions

- Block Copy works reasonably well when using the Assembler editor (see the Royal Cyber RDz Assembler Editing course manual for the steps

- Sadly, Block Copy seems broken when it comes to the LPEX editing. We are discussing with the IBM development team, and will let you know (via another Royal Cyber Newsletter) when things turn around.

- In the meantime use the COBOL or PL/I editor for working with code blocks. This isn't as bad as it sounds, because starting with RDz v9 IBM provided an "Open With" Context menu option for switching editors on the fly.
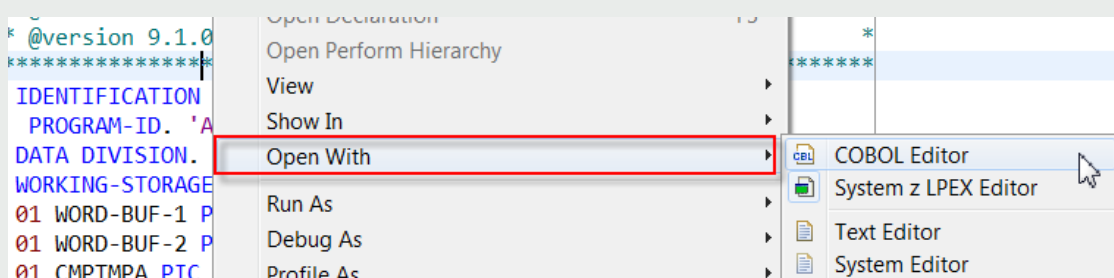
The RDz editor's Context Menu: Open With action

5. When a job finished executing how can I get notified?

- Unfortunately, RDz doesn't provide out-of-the-fox functionality for TSO Notify. There are two work-arounds:
  - **i.** Press F5 (refresh) or use the Context Menu to refresh your JES Filter
- Utilize the Poll Configuration option of "Show in Table" – to enable RDz's automatic JES poll facility
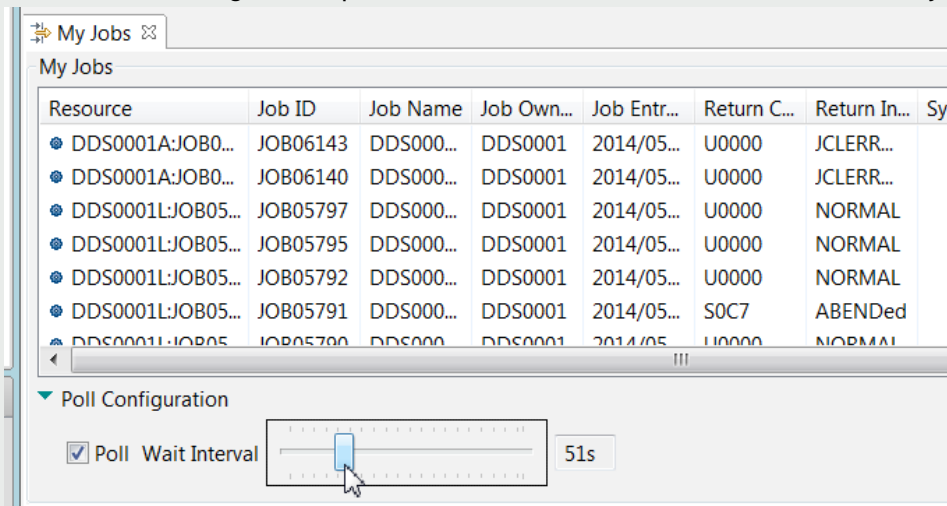
JES Polling

  - **i.** There is one additional option for this, though… an IBM consulting in Germany named: **Dennis Behm** created a plug-in that essentially pushes LISTBC commands to your workstation. These commands include TSO Notifications of Job completion.
  - **ii.** If you're interested in more information about this, please check out this link: https://www.ibm.com/developerworks/community/files/app/file/180a95a0-1dc3-489f-9cf1-2899f55c0bf7   - and feel free contact Dennis directly to obtain the plug-in: dennis.behm@de.ibm.com

6. How do I open multiple emulator sessions in version 9.x?

- From the menu, select: **Window > New Editor**
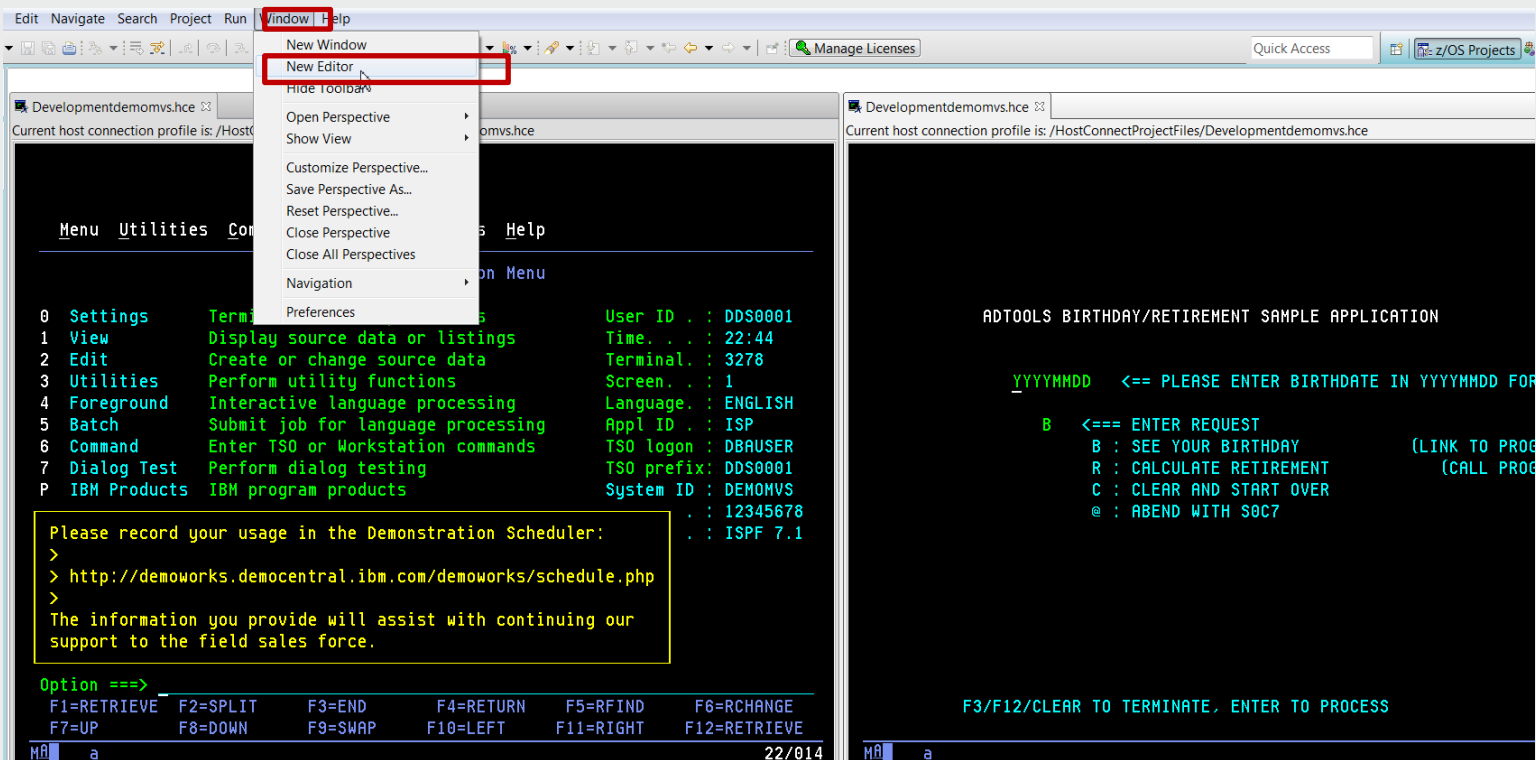
7. **Why does RDz Search take so long against large files and libraries (and what can I do about it)?**

- RDz's Search – which is a premier value-add for the product - is done using Java on z/OS – and depending on the volume of data to be searched over can be both expensive and time-consuming.
- Starting in **RDz v9.1** IBM released a version of search called "Remote z/OS File Search". This actually does a standard ISRSUPC search on the host in the background (via JCL) – making it as fast as ISPF – and as inexpensive.
    - i. And additionally, this 9.1 Remote z/OS File Search still provides the terrific hyper-text linking back to the dataset lines that caused search to test true for your text-pattern. ☺
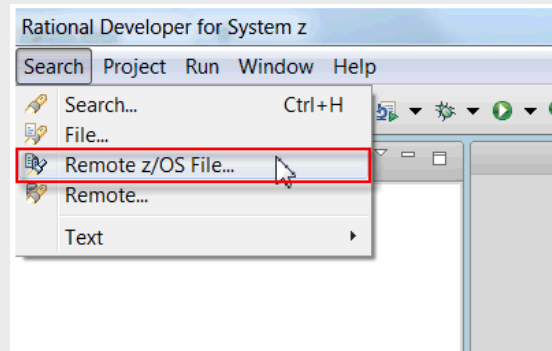
RDz v9.1 z/OS File Search

8. **How can I save JES Spool files on z/OS (like the SDSF "XDC" command)?**

- Using RDz v9.0 and earlier – you could save a JES Spool file to a local (workstation or network-server) dataset – by utilizing the menu: File > Save As
    - i. However often you won't be able to utilize a local file for your z/OS output because of binary values, or size, or … well … it's just not what you want to do – you want to save to a remote/zOS dataset.
- The good news is that in RDz v9.1 IBM has paved the way for you to do this because they've
    - i. Enabled Menu Manager actions on JES batch jobs and Spool Files (including nifty Menu Manager variables for elements of the job (jobid, jobname, etc.)
    - ii. And they've provided sample REXX execs and Menu Manager scripts to write JES Spool files to remote z/OS datasets
- In the next Newsletter we'll discuss this topic (Menu Manager on JES Spool files) in-depth.

9. **Does RDz support my COBOL 4GL? (IBM Report Writer, Netron Cap, CA Meta-COBOL, Telon, etc.)**

- From question #1 above – recall that RDz maintains an in-memory model of your program that is syntax-parsed, and used for Content Assist, and all of the advanced COBOL, PL /I, JCL and Assembler tooling.
- When you edit a 4GL source file (from some package that generates COBOL) the non-COBOL keywords embedded in the file will give the in-memory parser fits (in other words you'll see dozens to hundreds of syntax error annotations and (most likely) your COBOL tooling will be disabled. ☹
- **Preprocessing to the Rescue!**
    - i. Since v8.0.3 RDz has provided an elegant and simple solution to this problem – called Preprocessor Integration (see figure below).
    - ii. This option – part of your norm Property Group functionality - allows you to invoke a TSO REXX or CLIST command – which invokes your 4GL preprocessor (which is part of your build JCL stream).
    - iii. The output of this preprocessing on the host is piped into the in-memory/parsed version of your code so that the editor sees only ANSI-Standard COBOL
        - 1. While you work with/edit the original 4GL source – just like using ISPF
- IBM provides a learning/example video on this functionality – here:
    https://www.ibm.com/developerworks/community/files/app/file/fb23df1b-1f39-410d-afe6-898c9095a7e0

RDz Property Group > COBOL tab > Editor Configurations > Remote Preprocessing setup

10. My SCM is not an IBM product and it's not CA-Endevor… what are my options for integration?

- Assuming your SCM provides Check-in/Check-out, RDz can support it. Here are some options

SCM Solution	Options	Comments

Serena/Changeman	• Serena Plug-ins
	• Frameworks from:
	   ○ IBM
	   ○ Q-Group
		Many customers choose Serena's own RDz plug-ins, as they tend to trust vendor-supplied solutions. However these can be pricey. Consider contacting IBM or Q-Group to see what they offer:
		http://qgrp.com/english/products/csar-for-changeman/

Panvalet, Librarian, Alchemist, Home-Grown SCM products/utilities	• Menu Manager, HATS RCP • CARMA	The best method for interfacing with these SCMs is to utilize IBM's own Menu Manager + HATS/RCP functionality. Royal Cyber is arguably the best in the world at (cost-effectively) assisting you with this work. Please check out our tutorial at: http://webspherehatsguru.com/Royal-Cyber-Newsletter-Volume-2-February-2014.pdf
ISPW	ISPW corporation's own plug-in to RDz	This is link to an ISPW flyer: www.interchip.de/filestorage/**ISPW**forMultiPlatformApplications.pdf

The better answer to all of your "Say What?!?!" questions and comments

While the above are the most frequently asked questions they are by no means the only questions we get. In fact, in the 3 years we at Royal Cyber have been doing RDz work we've compiled a large set of Frequently Asked Questions – which are free and available here: http://royalcyber.com/em-app-store/rdz-extensible-eclipse-based-help-portal/

However you might ask yourself, "*Why wait until an RDz question makes it into a Top 10 List*"? Because, by the time that happens, the question (and probably many other questions) has begun to drain your anticipated product productivity and lowered your company's return-on-investment.

Common Sense

And then the follow-on question (especially for management); If you've spent (?? How much ??) money on RDz software, and then spent additional money (probably rivaling the software costs) on your people's time installing, configuring and deploying RDz – wouldn't it make sense to get trained and mentored so that there's minimal loss of productivity and users don't find themselves going back to ISPF because they don't know how to do something using RDz?

We can help

RDz is Royal Cyber's number one technical domain – and we are IBM's premier RDz world-wide services provider. We specialize in effective (including cost-effective) methods of getting ISPF developers up and running efficiently and we provide them with multi-layered and easily consumable 'how to' technical services in creative packages such as:

- Our Distance Learning "Wrapper" program: http://royalcyber.com/em-app-store/rdz-distance-learning-wrapper-program/ - that encloses our RDz Distance Learning with just enough technical mentoring and support for your custom RDz environment
- Our extensive catalog of RDz products and services:
- http://www.royalcyber.com/wp-content/uploads/2014/You-RDz-and-Royal-Cyber-v4.pdf
- Our Distance Learning program: http://royalcyber.com/royal-cyber-rdz-distance-learning-training-schedule/ where your developers will learn "native RDz" to the point that they're both excited about using RDz and effective at using it.

Chris Leland: Chris is Royal Cyber's primary RDz technical instructor - having worked full-time with RDz for almost four years; developing applications in COBOL, teaching RDz classes and mentoring in COBOL and Assembler, installing and configuring RDz, delivering custom workspace design sessions, writing an RDz evaluation exam, and integrating RDz with Rational Asset Analyzer. Chris passed the IBM/RDz Certification exam with flying colors in 2011.

Royal Cyber's Rational EM "App Store"

One of the things that's difficult to do, when you provide many disparate solutions for the Rational space is to find the proper software delivery strategy. So we built an App Store.

The **"EM App Store"** http://www.royalcyber.com/em-app-store/ is a web-based software distribution catalog that provides a consolidated digital distribution platform for the Royal Cyber Rational EM value-add plug-ins, tools and specialty services for the Rational EM brand solutions.

Some of the solutions can be automatically downloaded and installed into Rational products.

Other solutions (typically the more complex offerings) are blended tools + services.

We will continue to build out the various departments and offerings from month-to-month.

In the meantime?

Stop by.

Rational Developer for System z – plugins, tools and components

Property Group Composer

A plug-in that automates the creation and management of your Property Group SYSLIB function – critical to your RDz rollout success. Get App

Reusable Code Snippets *Free*

A collection of COBOL, SQL, JCL, MVS Utility and Assembler code fragments with parameterized variables for quality reuse. Get App

Custom Statement Templates *Free*

A collection of COBOL, SQL, JCL, MVS Utility and Assembler RDz statement templates – Content Assist enabled. Get App

Common COBOL and JCL Regex Templates *Free*

A set of simple to sophisticated Regular Expressions, which can be used to simplify and accelerate application analysis. Get App

Custom COBOL Code Review Rules

Plug-ins that extend RDz's Eclipse and JCL/Batch code review and standards checking functionality for run-time performance and maintain... Get App

RDz Custom Workspaces *Free*

A set of our most popular RDz/Eclipse Workspace designs – from contemporary/Goth to Green-screen/Dino Both look & feel and ... Get App

Shop a little. If you see something like? Pick it up.

Migration and Transformation Solutions

Micro Focus to IBM COBOL Conversion Facility

The Micro Focus COBOL Conversion tool allows customers to convert Micro Focus COBOL source and copybooks to IBM compatible versions. Get App

HATS Fusion

IBM Rational Host Access Transformation Services (HATS) transforms traditional text-based interfaces—such as 3270 and 5250 green-screen. Get App

Mainframe to Mobile Solutions

IBM Worklight helps you extend your business to mobile devices. It provides an open, flexible and comprehensive platform, upon which to build, run and manage HTML5, hybrid and native mobile apps. Get App

Rational Developer for System z- Learning tools and services.

RDz Eclipse-based Custom Help Portal

This local Eclipse project provides fast, deep and Search-able, high-quality "How to" help on most aspects of RDz day-to-day use. Get App

RDz Learning Video Series *Free*

Check out our learning videos on RDz on RDz and RAA. Useful for post-training education and ongoing support. Get App

RDz and RAA Custom Courseware

We offer deep technical services on both RDz and RAA. Our delivery has garnered world-wide acclaim from the over 40 companies we ...

RDz Hotline" Real-time person-to-person Mentoring assistance

We are setting up a 24-hour/RDz Call Center, where you will be able to screen-share and discuss your technical issues with a ...

RDz/zOS Integration Tools and Examples

Using Menu Manager and HATS/RCP we have skeleton projects and deep expertise in this domain – critical to successful RDz ...

RDz Distance Learning Wrapper program

Since IBM awarded Royal Cyber the delivery of their monthly Distance Learning program, we have been offering short/custom ...

Royal Cyber @ Innovate

We (Syed, Anand, Ravi, and Chris will be @ Innovate from Saturday thru Thursday. You can catch with any of us, by emailing me (Syed Basheer): basheer@royalcyber.com - and we'll make the time to connect.   We are presenting at these sessions:

Session #	Title	Location	Time
2210A	COBOL Application Development for IBM AIX	Dolphin - Oceanic 2	Tue, 03-June 4:15 PM ➔ 5:45 PM
1865A	Aon Integramark Creates a Dynamic SOA Environement: Learn How	Dolphin - Oceanic 2	Wed, 04 June 3:00 PM ➔ 4:00 PM
2434A	Improving Your Time to Value with Rational System z Development Solutions: Proven Practices from the Pros	Dolphin - Oceanic 1	Wed, 04-June 10:45 AM ➔ 12:15 PM
2339A	Customer Panel: DevOps for Enterprise Systems: Customer Success Stories	Oceanic 3	Tue, 03-June 11:15 AM ➔ 12:15 PM
2331	Opening your Power Systems to Mobile	Dolphin - Oceanic 2	Thursday, 05-June 3:00 PM ➔ 4:00 PM

About Innovate 2014: http://www-01.ibm.com/software/rational/innovate/

IBM Software

Innovate2014

June 1–5 Orlando, Florida

‹ Innovate @ SPEED ›

Scott Adams   Gene Kim

| Innovate | Register | Why Attend? | Agenda | Keynotes | Sponsors | Business Partners |

Gene Kim   Scott Adams

Guest Speakers

Scott Adams, creator of the world renowned "Dilbert" comic strip, takes center stage at Innovate 2014. Using his book, How to Fail at Almost Everything and Still Win Big, as a backdrop, Adams will discuss how all of us Dilberts can succeed in a world of pointy-haired managers. You don't want to miss his inspiring story on how to Innovate@Speed.

Gene Kim is a multiple award-winning CTO, researcher and author. He was founder and CTO of Tripwire for 13 years. Gene is a huge fan of IT operations, and how it can enable developers to maximize throughput of features from "code complete" to "in production," without causing chaos and disruption to the IT environment.

Register to attend: https://www-950.ibm.com/events/wwe/innovate/inv14reg.nsf/enrollcodes?openform

Royal Cyber in the News

In February & March we delivered our first two IBM RDz Distance Learning classes – to over 230 new RDz users world-wide. As a first-time transition from IBM, which had been running Distance Learning for six years there were some hiccups with the logistics, but he results were as you can see below. By the time this Newsletter goes to print there will (likely) **not** be availability left in the April Distance Learning session, but you can sign up for May (through the end of 2014) sessions here: http://royalcyber.com/royal-cyber-rdz-distance-learning-training-schedule/

Rational EM News

* For details on the Royal Cyber RDz Rollout and project/task usage modeling contact us:
  inquires@royalcyber.com

* Details on RDz technical services:
  http://www.royalcyber.com/wp-content/uploads/2014/You-RDz-and-Royal-Cyber-v4.pdf