Tim Hahn - Chief Architect for Enterprise Modernization Tools
28 February 2011

Application Development for z/OS

Not your Father's Green Screen

SHARE Session 8656

Abstract

Ask most people how they write and maintain applications on z/OS and you hear "oh, you use this thing called a green screen" followed by a chuckle.

In reality, application development for zEnterprise applications has been transformed over the past several years to the point where application developers enjoy the same or better features from integrated development environments as programmers who work on other platforms.

Advances in remote system communication and interaction, syntax-highlighting, parsing, and code understanding for Assembler, PL/I, C/C++, and COBOL source code, as well as programming assists such as code snippets and templates are all available to application programmers. Interactive debug of applications, written in multiple programming languages and running in various runtime environments is also possible and can greatly boost programmer productivity.

Come and learn about how these features can enable application developers who are new to the mainframe to interact with, update, and efficiently enhance mainframe applications.

Agenda

- Application Development is Hard
- Tools to the rescue!
- Using tools is Hard
- Continual Discovery
- Reprise: Application Development is Hard

Agenda

- Application Development is Hard
- Tools to the rescue!
- Using tools is Hard
- Continual Discovery
- Reprise: Application Development is Hard

Traditional Application Development for z/OS

- study compiler listings (green bar printout) or use ISPF

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       DDS0001.TEST.COBOL(HOSPEDIT) - 27.26            Columns 00001 00072
****** ******************************** Top of Data ********************************
000001        ***************************************************************************
000002         IDENTIFICATION DIVISION.
000003         PROGRAM-ID.  HOSPEDIT.
000004         AUTHOR. JON SAYLES.
000005         INSTALLATION. COBOL DEVELOPMENT CENTER.
000006         DATE-WRITTEN. 01/01/08.
000007         DATE-COMPILED. 01/01/08.
000008         SECURITY. NON-CONFIDENTIAL.
000009
000010        ***************************************************************************
000011        *  A new comment ...
000012        *  Jon's new comment
000013         ENVIRONMENT DIVISION.
000014         CONFIGURATION SECTION.
000015         SOURCE-COMPUTER. IBM-390.
000016         OBJECT-COMPUTER. IBM-390.
000017         INPUT-OUTPUT SECTION.
Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

Multiple Edit windows are possible - but limited

- ISPF split-screen mode allows this ... but it is far from sufficient for complex, multi-module application programming problems

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       DDS0001.TEST.COBOL(HOSPEDIT) - 27.26            Columns 00001 00072
****** ******************************** Top of Data ********************************
000001        ************************************************************************
000002         IDENTIFICATION DIVISION.
000003         PROGRAM-ID.  HOSPEDIT.
000004         AUTHOR. JON SAYLES.
Command ===> ____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       DDS0001.TEST.COBOL(HOSPCALC) - 06.05            Columns 00001 00072
****** ******************************** Top of Data ********************************
000001         IDENTIFICATION DIVISION.
000002         PROGRAM-ID.  HOSPCALC.
000003         AUTHOR. JON SAYLES.
000004
000005         ENVIRONMENT DIVISION.
Command ===> ____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

And it's not just the basic tools ...

- Existing systems have grown by evolution over many years
- Many documented (and un-documented) dependencies
- Sheer volume of applications
  - thousands of batch jobs
  - thousands of programs
  - billions of lines of COBOL code run daily, not to mention on other schedules (e.g. end of quarter, end of year).
- Online transaction processing also factors in

- Without application analysis tools, teams have difficulty understanding even where to start

Agenda

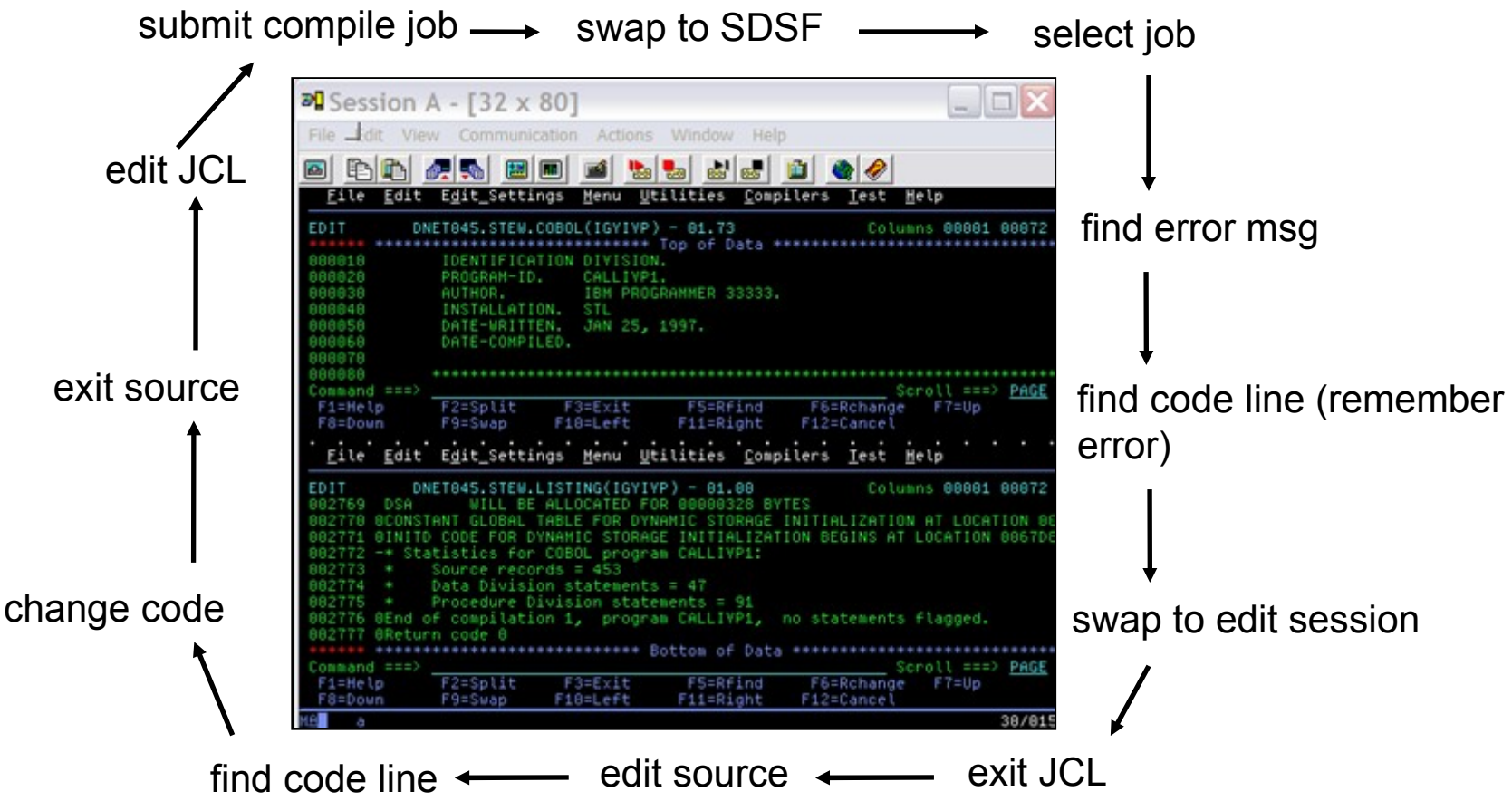
- Application Development is Hard
- Tools to the rescue!
- Using tools is Hard
- Continual Discovery
- Reprise: Application Development is Hard

Rational Developer for System z and zEnterprise

- Practitioner tools for application development and enhancement
  - Java
  - COBOL
  - PL/I
  - C/C++
  - Assembler
  - JCL
- Supporting tasks of
  - Remote access to files and jobs
  - Analyze, Understand, Edit, Build, and Unit Test of applications
  - Remote interactive debug of applications running in multiple environments
  - Integration with SCMs including Team Concert and Endevor
- Support for several source code location models
  - “remote” source code (source code held on development system)
  - “local” source code (source code held on system where IDE is running)

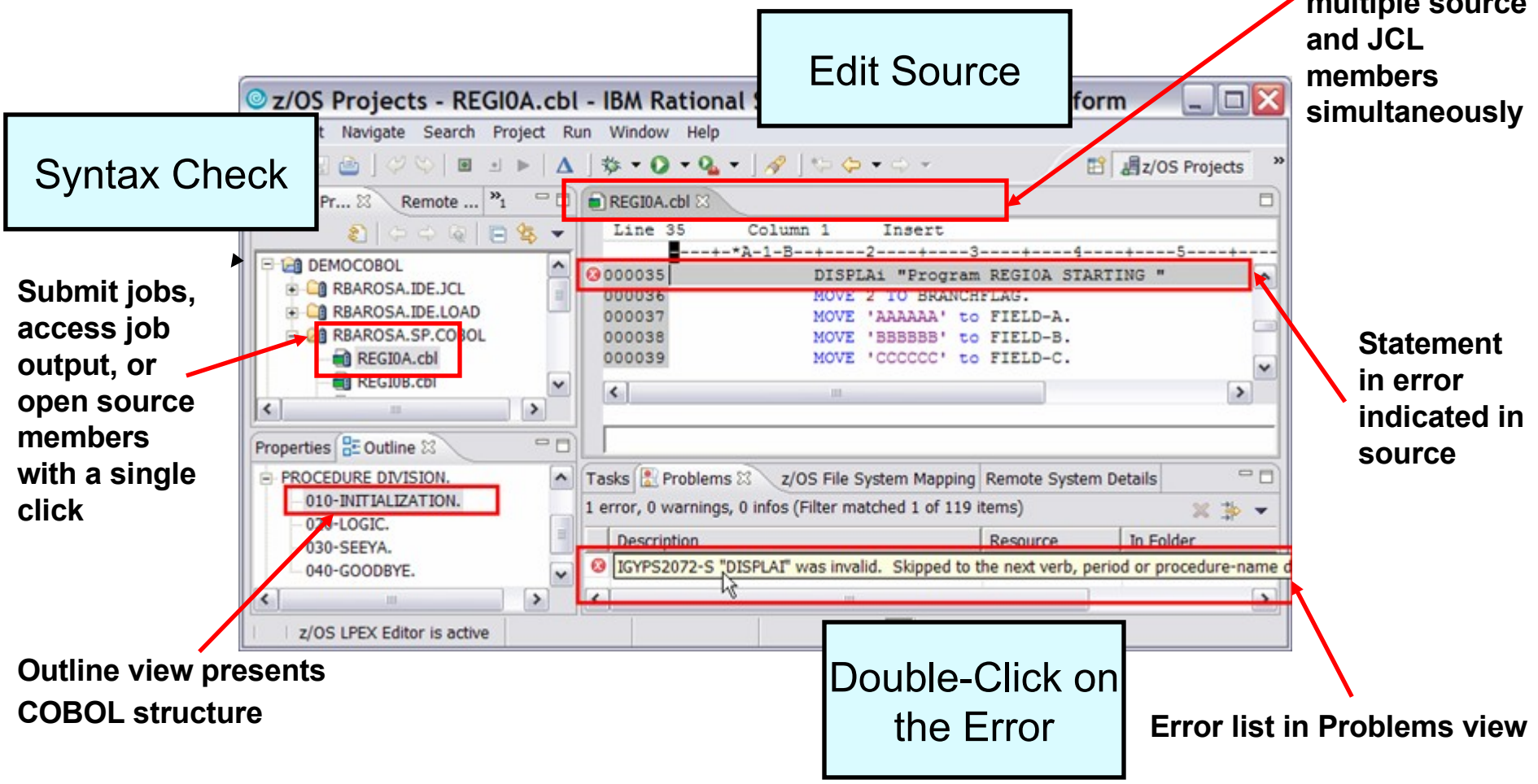
ISPF-based development

- Multiple screens/sessions and multiple disparate tools
- 20 x 80 characters of content

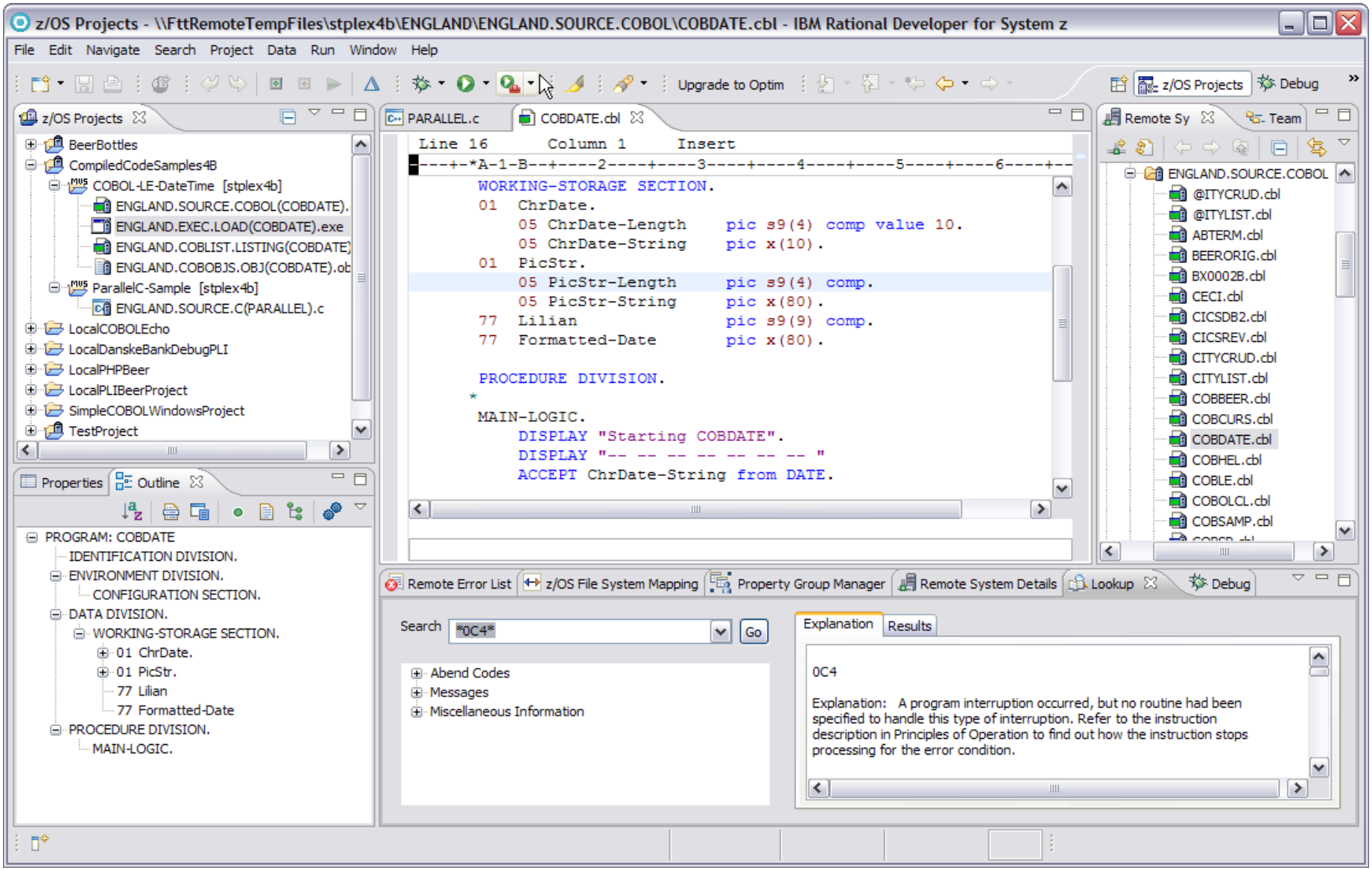


IDE-based development

- Common development environment for COBOL, PL/I, C/C++, and Java
- Simplified development with more information at your fingertips

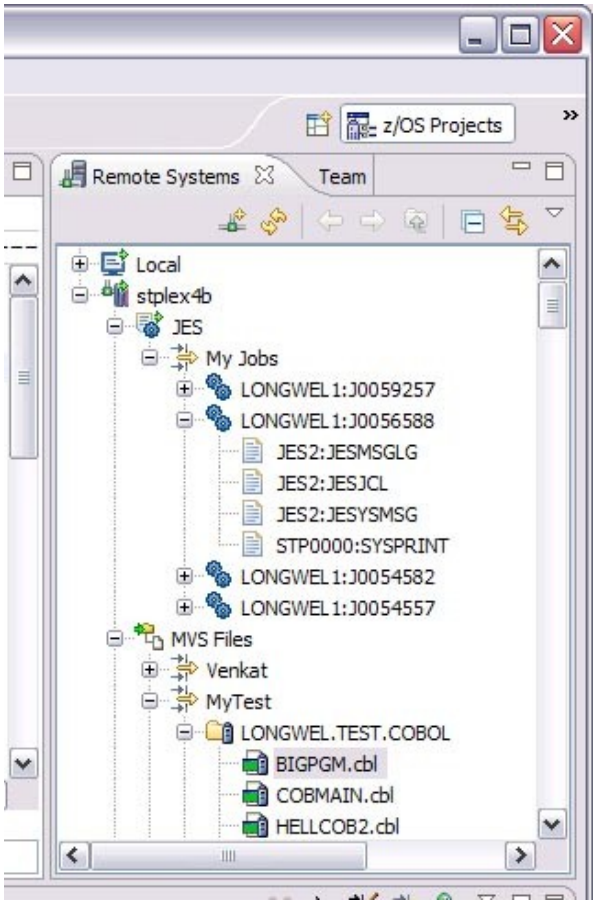


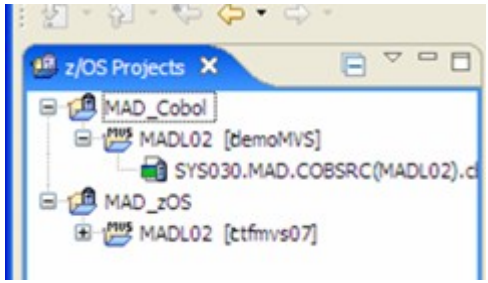
IDE-based Development



Navigate datasets or jobs live on z/OS

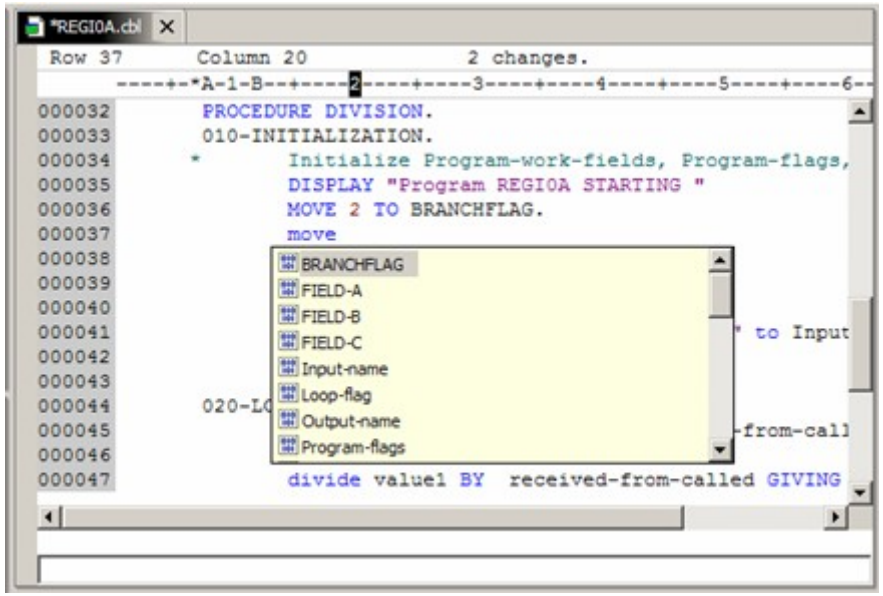
- Connect to multiple hosts concurrently
- Respects existing security configurations and user IDs
- Search, filter, browse, edit, compare, migrate, and allocate new MVS datasets and USS files
- Copy source code, members, or datasets between systems with a few mouse clicks.
- Access JES queues submit jobs, view job state, and open output spools
- Submit TSO or USS commands
- Add datasets and members into projects to group applications and work items together logically
- Open an emulator in the IDE to configured hosts

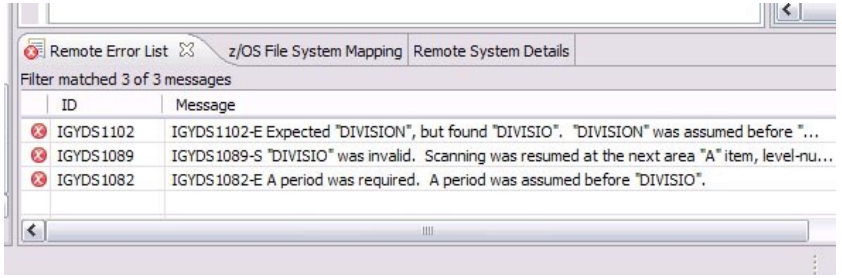




Edit and syntax check source code

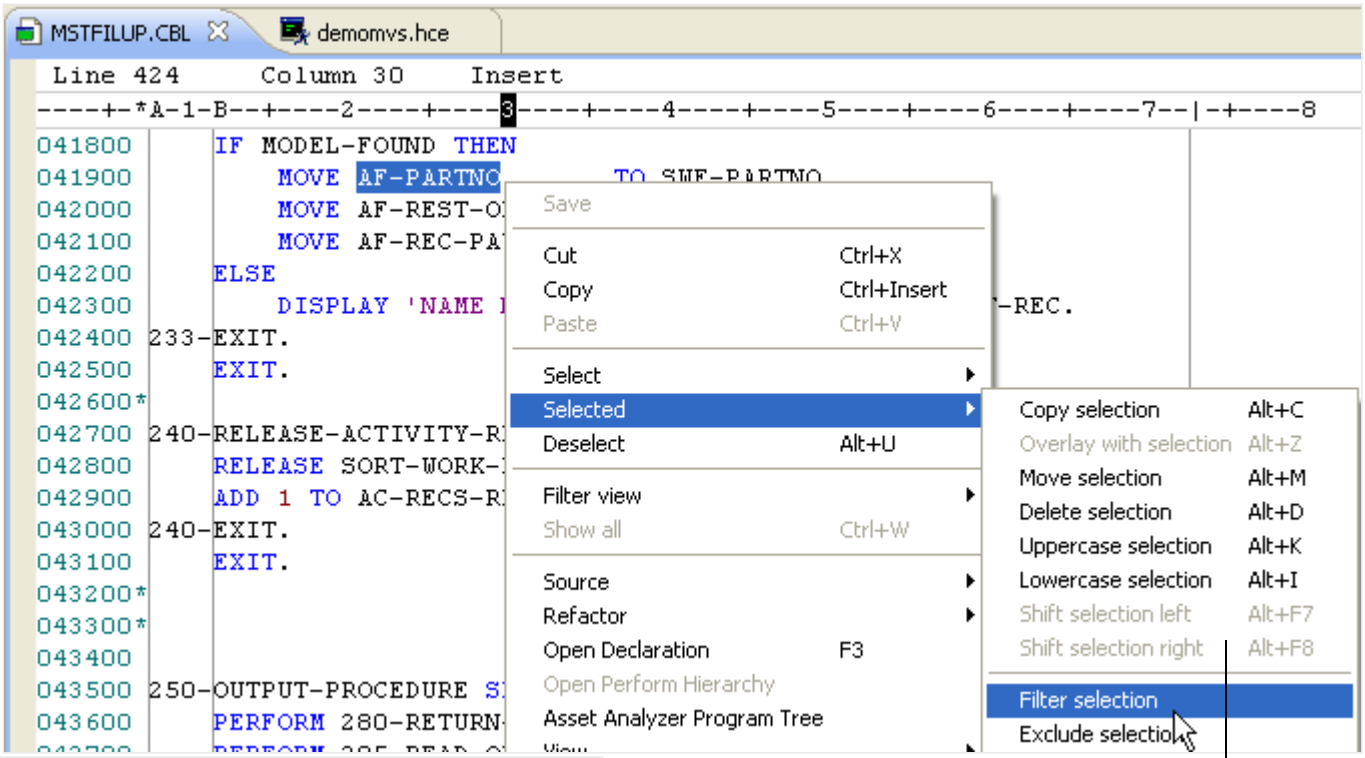
- Use advanced editing technology to:
  - Work with multiple source and JCL members concurrently from different systems
  - Perform ISPF-like commands in the workstation editor (e.g, FIND, CHANGE, INSERT LINE, etc)
  - Use syntax highlighting and code-completion to gain insight into available variables, verbs, and keywords
  - Quickly create programs from code templates, pattern definitions, or UML
  - Verify COBOL syntax with feedback as you type in real-time
- Issue syntax check commands against project source code
  - Syntax check remotely to ensure proper code structure before compilation
  - Syntax check locally ensure proper code structure and reduce server usage. RDz will download code and dependencies (e.g., copybooks) to the workstation as necessary
  - Syntax Errors are listed in the Remote error list. Double-click on the error to open the dataset and move to the line where the error was found

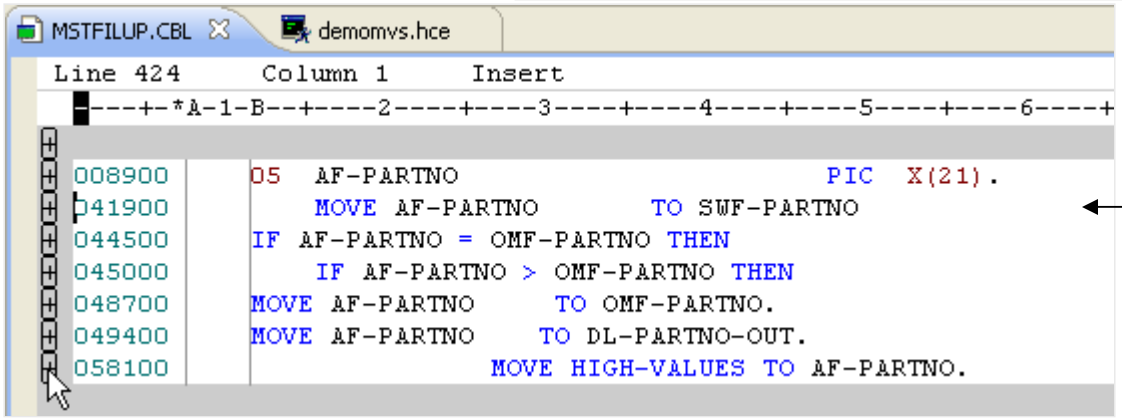




Isolating Code Elements

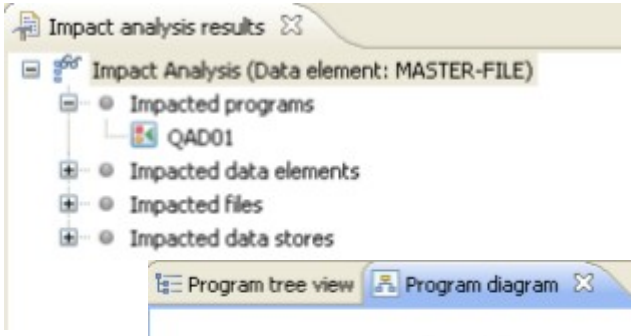
- Isolate and filter source code
- Multiple editor styles
  - “Eclipse”-style
  - ISPF-style
- Bookmark, and/or Expand & Collapse code details

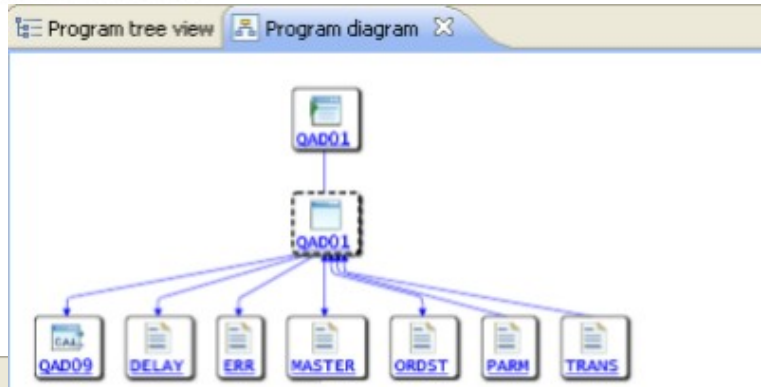


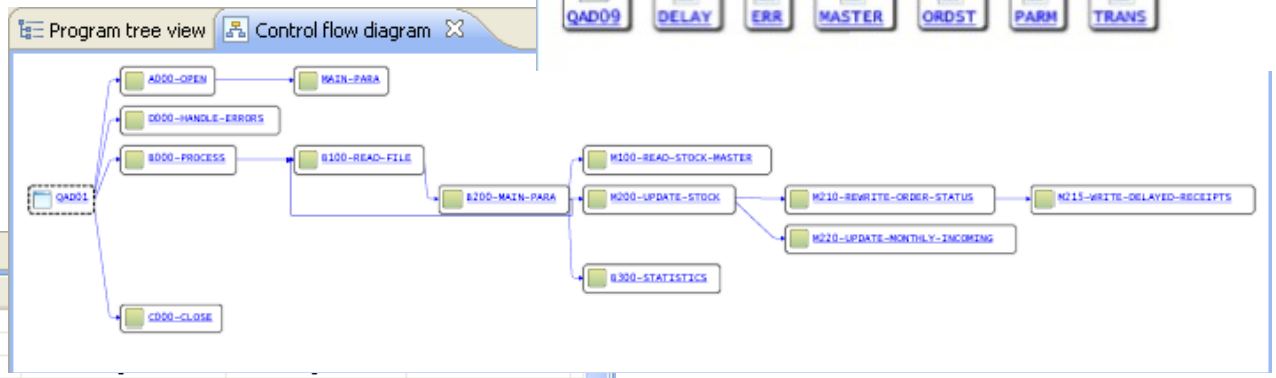


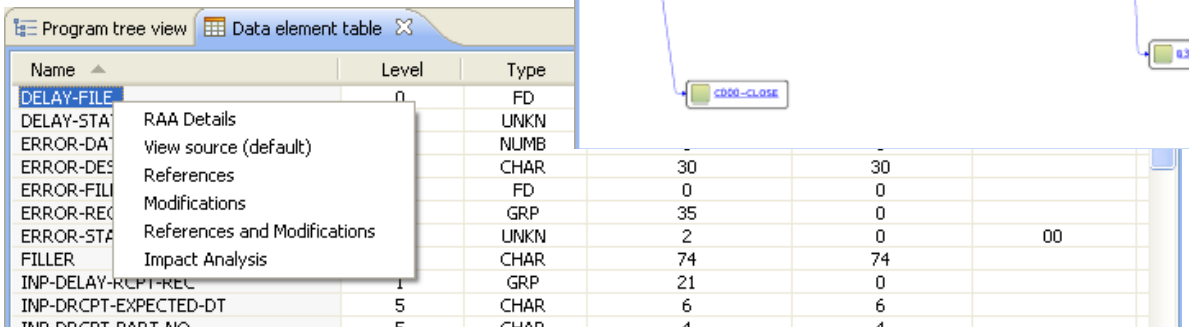
Analyze applications using graphical diagrams

- Bring application analysis information into the IDE to aid in program development and understanding
  - Link code to data and runtime resources
  - Visualize code structure and flow
- Understand the effect of changes made in the IDE when deployed into production
  - Run impact analysis on code to determine affected modules
  - Size testing efforts and create workspaces for changes





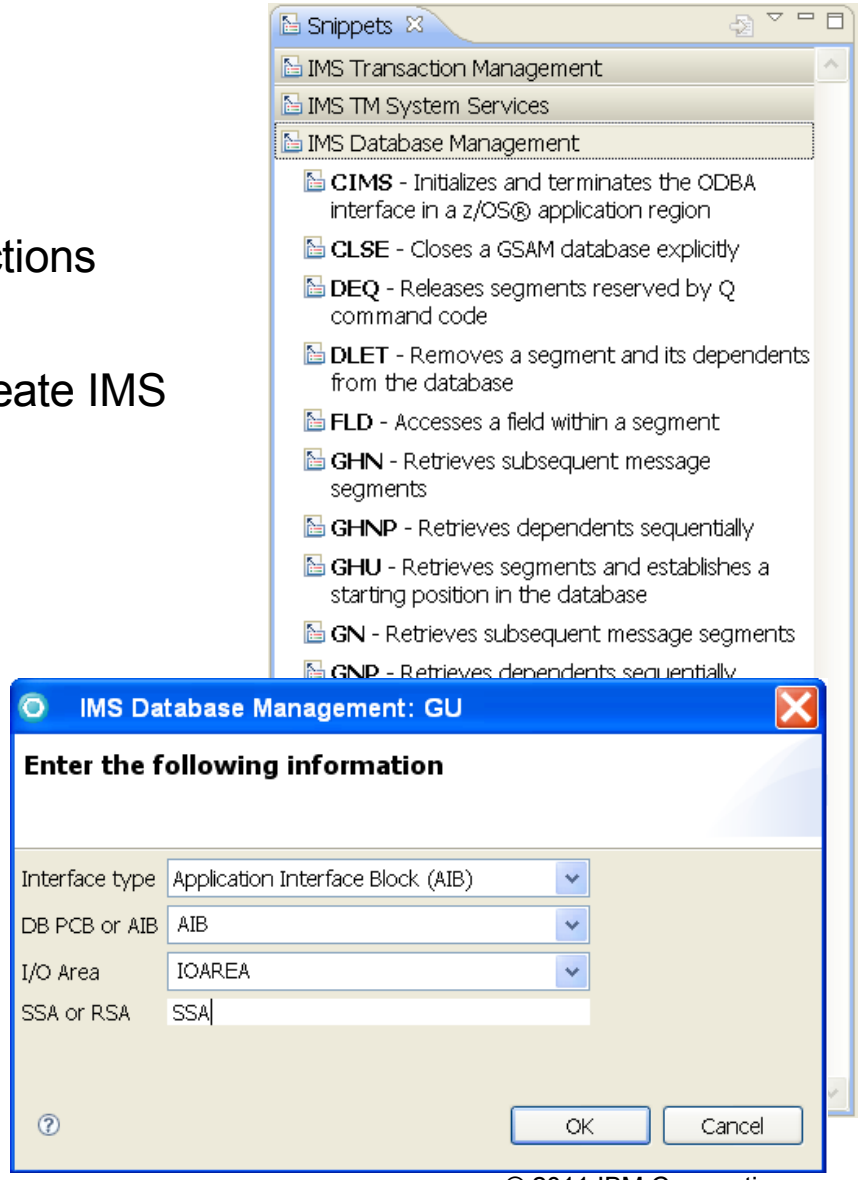




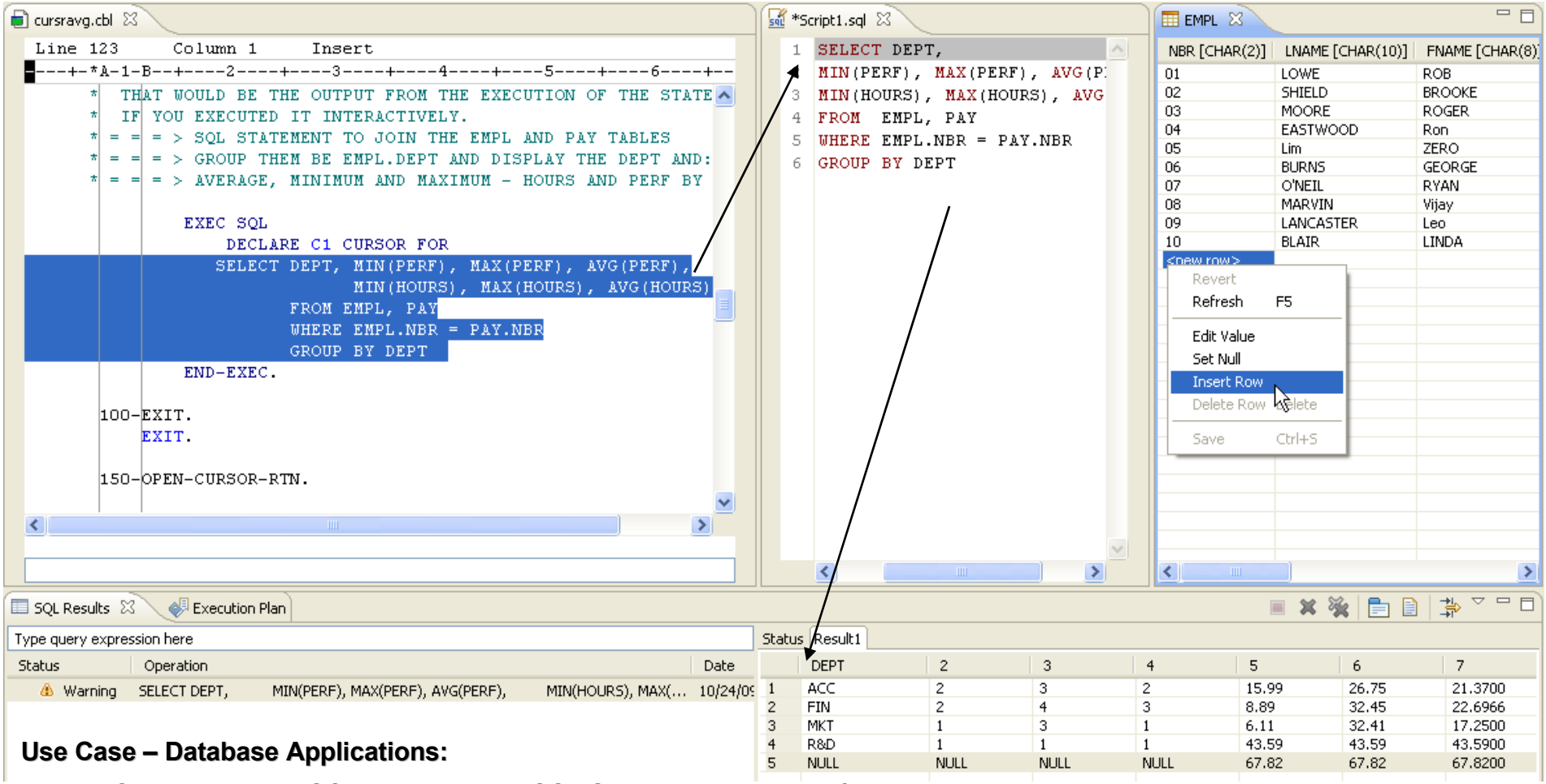
NOTE: Features on this page require usage of Rational Asset Analyzer in conjunction with Rational Developer for System z or zEnterprise

Speed Development with Code generation

- Model Driven Development
  - use UML to generate COBOL code
- CICS
  - Create working CICS-DB2 CRUD transactions
- IMS statement insertion
  - 71 IMS code generation wizards aid to create IMS COBOL code inline
- DB2
  - Stored Procedure wizards
- Batch applications
  - VSAM / QSAM access program creation
  - Pattern-based code creation preview



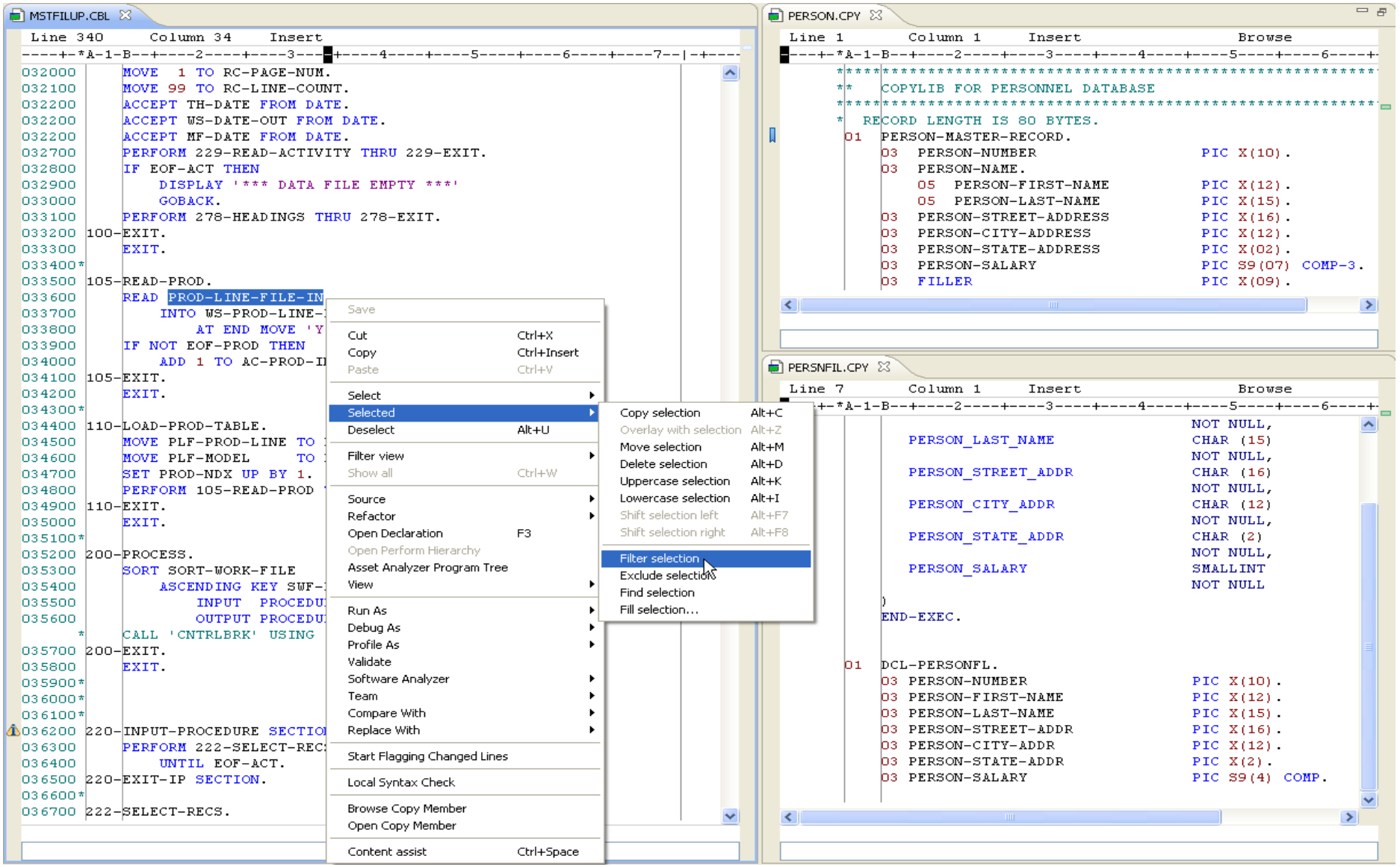
Interactive test of database queries



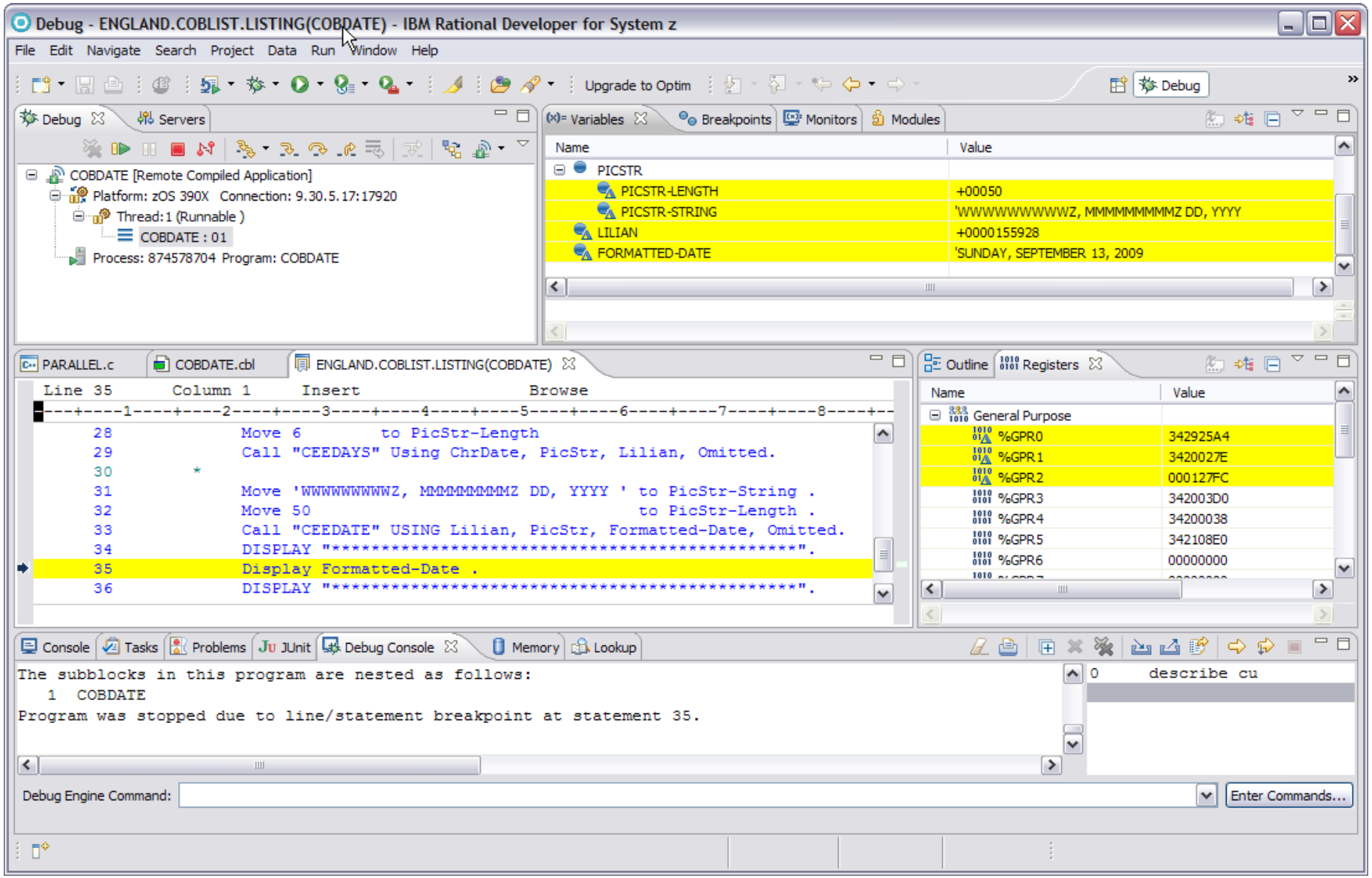
Use Case – Database Applications:

1. Copy/Paste your SQL Declare into a SQL Script and run it – verify results
2. Without doing any other navigation open a test table, and edit row values – or modify the statement (or both)
3. Re-run the SQL Script – verify results
4. Return to step 2 – repeat until satisfied with functionality

Edit a program and edit its copybooks (all at the same time)

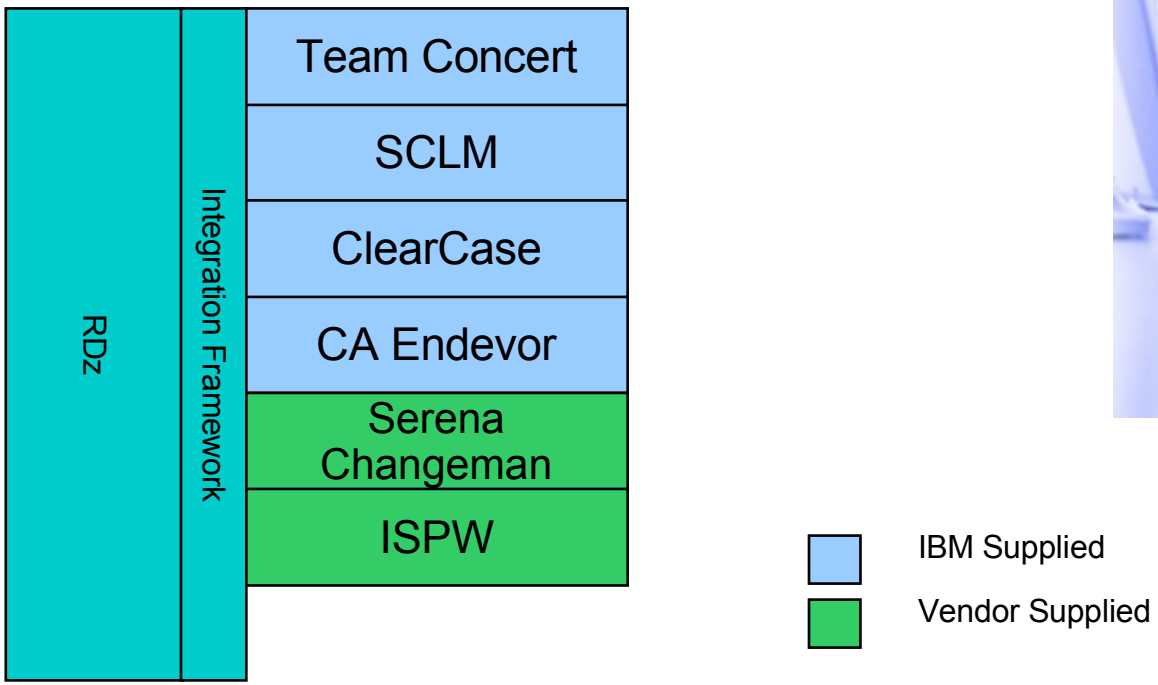


Interactive Debug of applications running on z/OS



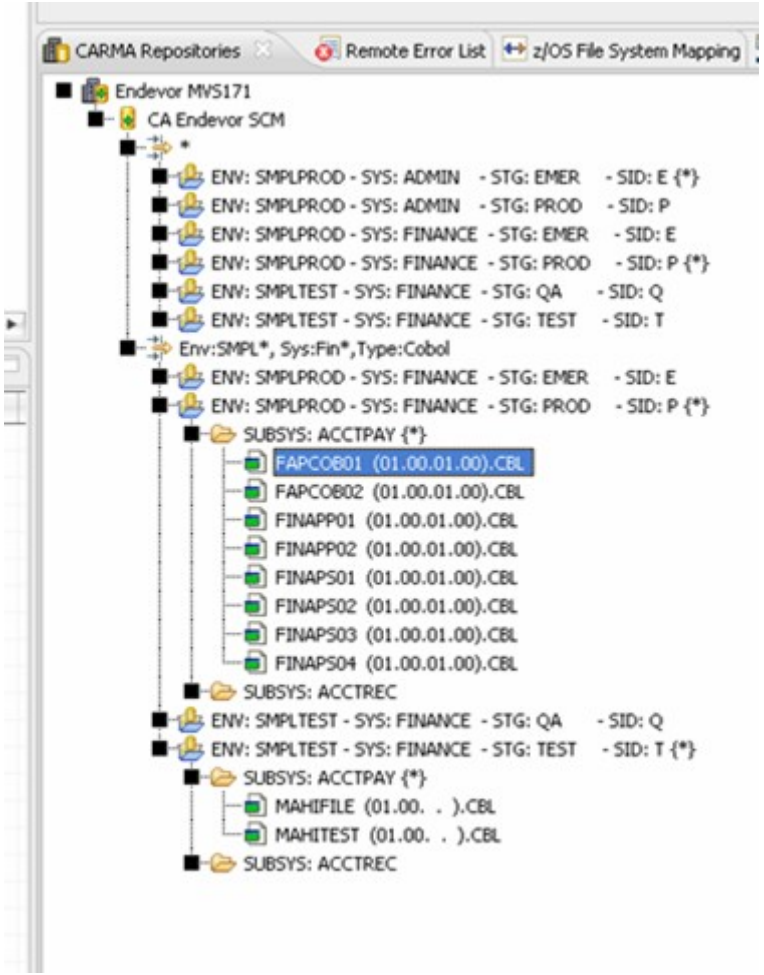
Access source code…

- RDz offers integration into a variety of Source Code Management (SCM) tools as well as a framework for creating SCM integration on your own
- Several vendors supply plug-ins to RDz to provide easy access to processes and source code controlled by their products



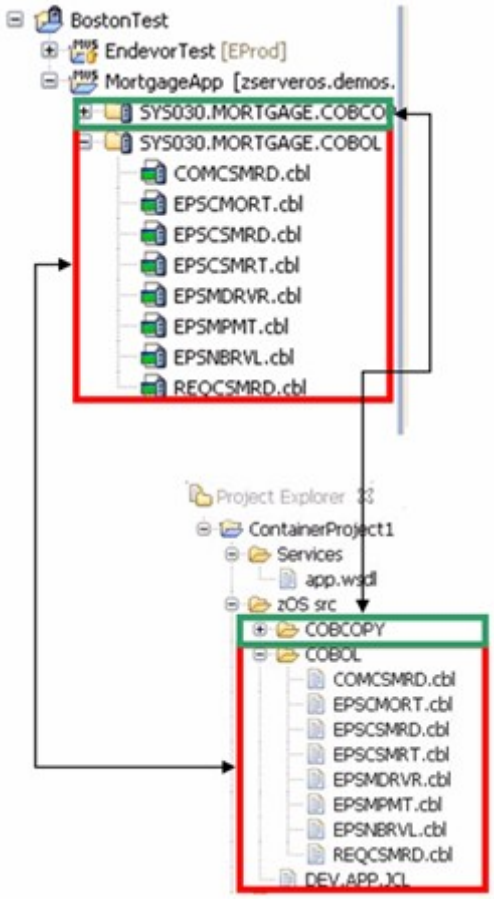
Endevor Integration

- Filter and search through environments, systems, subsystems, members, and stages based on queries (equivalent to DISPLAY)
  - Filters saved across z/OS sessions
  - Easy access to common searches and members
  - Drill down into subsystems
- RETRIEVE members to z/OS projects
  - Access to typical RDz functionality like syntax check, content assist, debug, etc
- ADD/UPDATE members with single click
  - RDz remembers Endevor location for retrieve and adds back
- QuickEdit (browse) members from CARMA interface
- Integration with existing GENERATE configuration



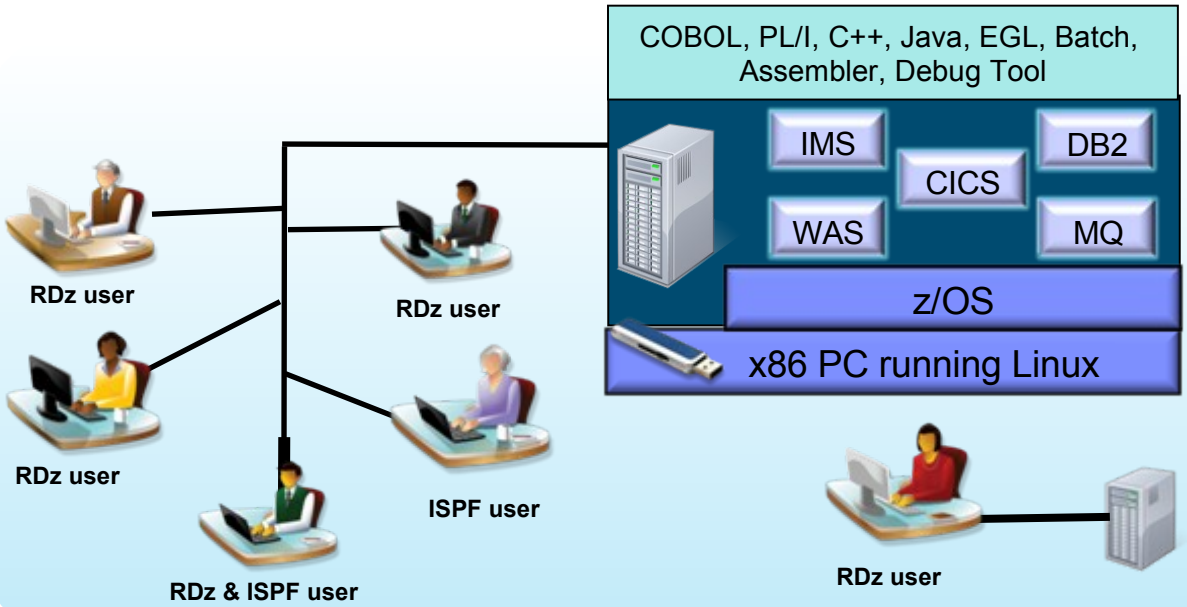
RTC integration with RDz - all tasks within a single IDE

- RTC provides
  - agility, collaboration and process
  - Work item planning and coordination
  - SCM and Build functions for z/OS (and other platforms)
- RDz augments the development productivity & experience
  - files act as if on the host
  - Appropriate editors (COBOL, maps, etc.) and functions (content assist, syntax check, etc.)
  - High value functions (XML enablement, SFM, code generation from models, from UML, etc)
- RDz projects in RTC
  - RDz projects are a view into the RTC project
  - RDz projects provide a working set for the developer



RDz Unit Test Feature

Assisting application development for System z



For more information:
Session 8369
Monday – 4:30-5:30PM
Room 201A

- Liberate developers to rapidly prototype new applications
- Develop and test System z applications anywhere, anytime!
- Free up mainframe development systems for production capacity
- Eliminate costly delays by reducing dependencies on operations staff

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

Agenda

- Application Development is Hard
- Tools to the rescue!
- Using tools is Hard
- Continual Discovery
- Reprise: Application Development is Hard

Using Tools - There is a Learning Curve

- It's true.
- Any new tool is, at first, overwhelming.

- ISPF and “green screen” users
  - Compatibility mode for editor environment
  - Remote system access
  - 3270 emulator included
- IDE-comfortable users
  - Multiple edit window support
  - multiple editor selections available based on preference
  - extensive preferences and customization
  - context menus for most common tasks

Agenda

- Application Development is Hard
- Tools to the rescue!
- Using tools is Hard
- Continual Discovery
- Reprise: Application Development is Hard

Continual Discovery

- Challenge Yourself
  - Resolve to learn something new every day
- Find a Buddy
  - learn from what each of you have found
- Impress your Friends
  - find out something cool? Share you knowledge!
- Be Social!
  - Join a user group, discussion group, or online community
  - ask questions, or just lurk and learn
  - (See links at end of presentation)

Agenda

- Application Development is Hard
- Tools to the rescue!
- Using tools is Hard
- Continual Discovery
- Reprise: Application Development is NOT SO Hard

The hardest part of Application Development is ...

- Getting started

The hardest part of Application Development is ...

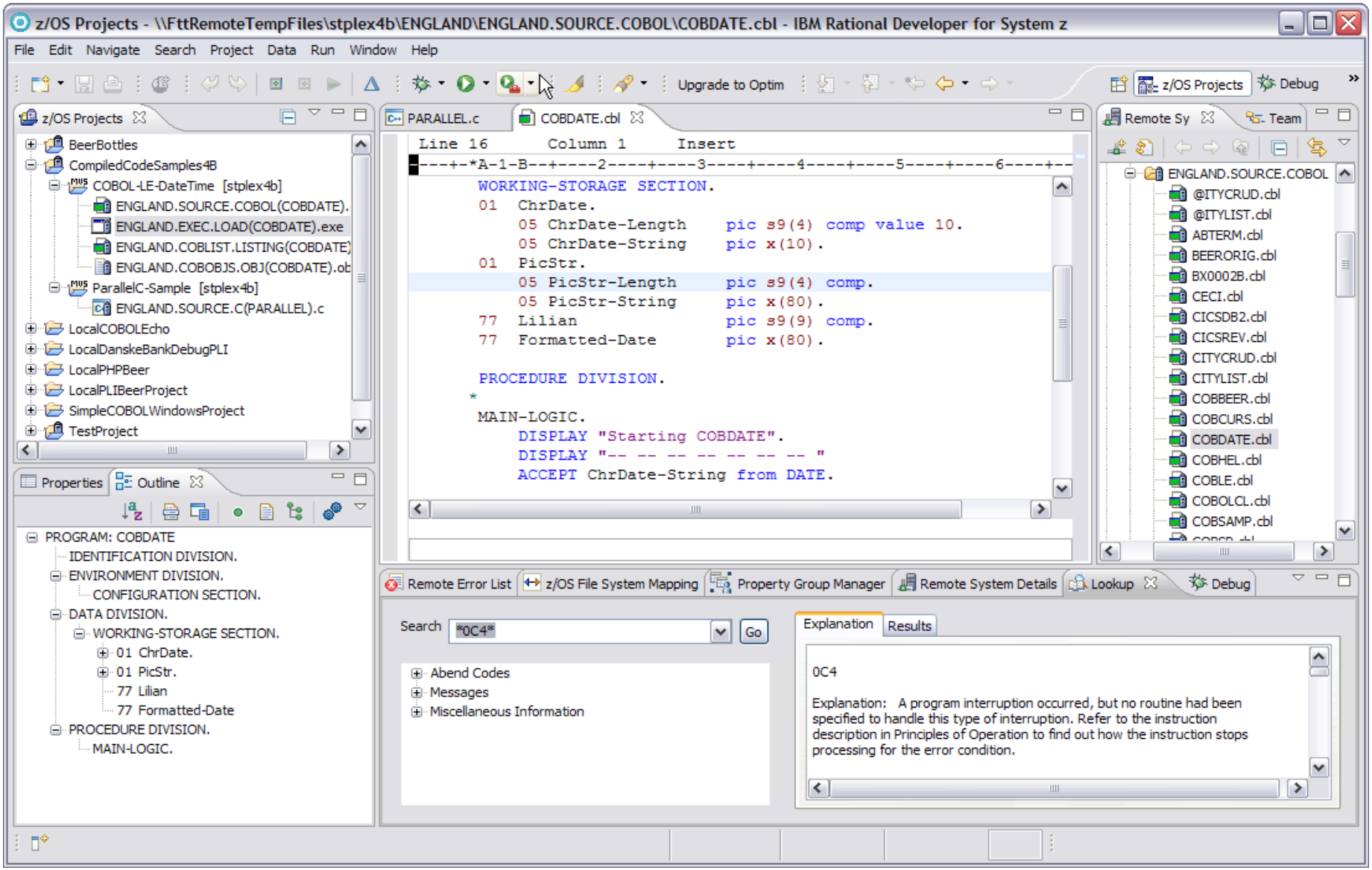
- Getting started ... and
- Staying engaged and on task

Application development for z/OS used to be like this ...

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       DDS0001.TEST.COBOL(HOSPEDIT) - 27.26            Columns 00001 00072
****** ******************************** Top of Data ********************************
000001        ***************************************************************************
000002         IDENTIFICATION DIVISION.
000003         PROGRAM-ID.  HOSPEDIT.
000004         AUTHOR. JON SAYLES.
000005         INSTALLATION. COBOL DEVELOPMENT CENTER.
000006         DATE-WRITTEN. 01/01/08.
000007         DATE-COMPILED. 01/01/08.
000008         SECURITY. NON-CONFIDENTIAL.
000009
000010        ***************************************************************************
000011        *  A new comment ...
000012        *  Jon's new comment
000013         ENVIRONMENT DIVISION.
000014         CONFIGURATION SECTION.
000015         SOURCE-COMPUTER. IBM-390.
000016         OBJECT-COMPUTER. IBM-390.
000017         INPUT-OUTPUT SECTION.
Command ===> _____________________________________________ Scroll ===> PAGE
 F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
 F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel
```

And it still can be ...

... but why do that when you can use all these features?



Rational Developer for System z and zEnterprise

- Eases developers into working with System z systems
  - **Easier to get started**

- Offers many features within the Integrated Development Environment
  - **Keeps developers on task and engaged**

www.ibm.com/software/rational

Useful Links

- Rational Developer for System z Information:
  - http://www.ibm.com/software/rational/products/developer/systemz/
- Rational Developer for System z Infocenter:
  - http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp?topic=/com.ibm.iea.rdz/rdz/wdz76.html
- Additional Video Demonstrations:
  - http://websphere.dfw.ibm.com/atdemo/atdemo_rdz.html
  - http://websphere.dfw.ibm.com/atdemo/atdemo_rdz_zosad_recorded.html
- Rational Developer for System z - Distance Learning:
  - https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Learn%20RDz-Learn%20COBOL
- Rational “COBOL Cafe” online discussion group:
  - https://www.ibm.com/developerworks/rational/community/cafe/cobol.html
- Rational Developer for System z - RDz hub:
  - https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=df67969e-ba40-44c7-a1ca-ef4a2aa99e01
- My information
  - email: hahnt@us.ibm.com
  - Blog: https://www.ibm.com/developerworks/mydeveloperworks/blogs/applicationmodernization