

# A System z Developer's Journey Through the Application Lifecycle



Liam Doherty [dohertl@au1.ibm.com](mailto:dohertl@au1.ibm.com)



Gary Mazo [mazo@us.ibm.com](mailto:mazo@us.ibm.com)

3 March, 2015

Session: 16517



#SHAREorg



SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.



# Trademarks and Legal Notes

## Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml): AS/400, DBE, e-business logo, ESCO, eServer, FICON, IBM, IBM Logo, iSeries, MVS, OS/390, pSeries, RS/6000, S/30, VM/ESA, VSE/ESA, Websphere, xSeries, z/OS, zSeries, System z, z/VM

The following are trademarks or registered trademarks of other companies

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation  
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries  
LINUX is a registered trademark of Linux Torvalds  
UNIX is a registered trademark of The Open Group in the United States and other countries.  
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.  
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.  
Intel is a registered trademark of Intel Corporation  
\* All other products may be trademarks or registered trademarks of their respective companies.

## NOTES:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.



# Purpose and Presentation flow

- Purpose ... to present a System z developer's use of tools that help manage the software development cycle: "Day in a Life"
- Flow
  - Overview of
    - What is DevOps and its role in current state of Software development
    - Tools for Software development to support the DevOps story
  - Walk through the dev cycle and tools that supports each step in the cycle

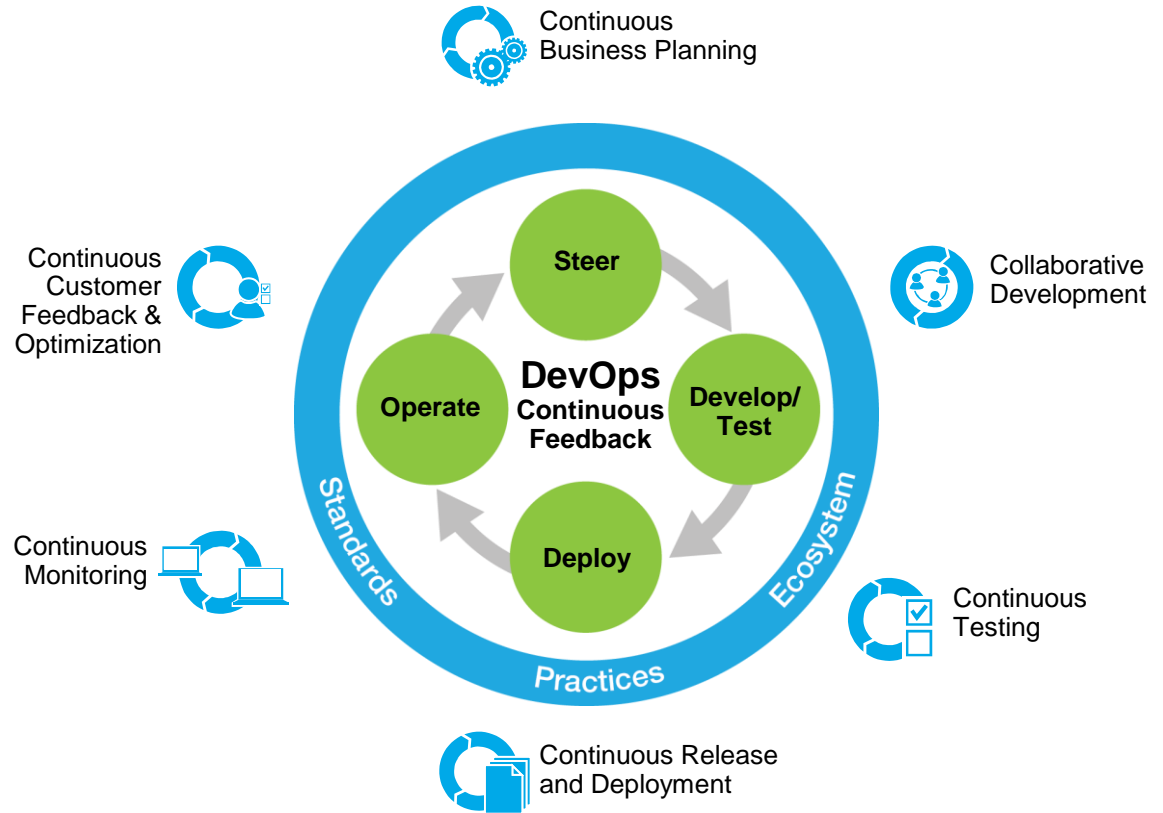
# IBM DevOps point of view

Enterprise capability for continuous software delivery that enables organizations to seize market opportunities and reduce time to customer feedback

**Accelerate software delivery –**  
*for faster time to value*

**Balance speed, cost, quality and risk –**  
*for increased capacity to innovate*

**Reduce time to customer feedback –**  
*for improved customer experience*



Our DevOps POV is resonating with clients and they are delivering **measurable business outcomes with DevOps**

# Overview of Supported Production Scenario

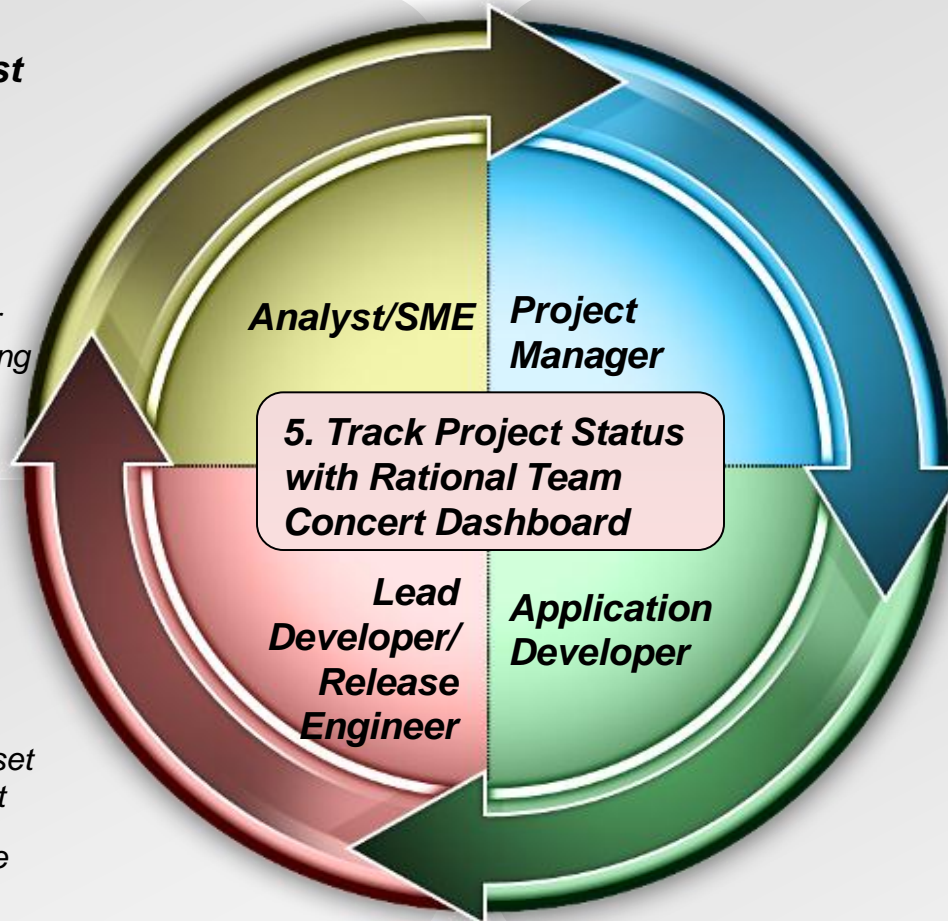
*Project Manager or Support Team has submitted Project Change Request ...*

## 1: Review Change Request

- Analyze application to be changed
- Size/scope effort and risk of change
- Submit to Project Manager for review, approval and scheduling

## 2: Review and Approve Change Request

- Review analysis for change request and approve for scheduling
- Create development work item(s) for implementation
- Add work to project plan



## 4: Promote and deploy enhancement

- Promote changes from development to test
- Create update package with set of changes from development
- Deploy update package to the test environment

## 3: Implement required changes, build and deliver

- Analyze source to identify modifications
- Implement and test modifications
- Perform personal build and deliver new features

# Development Life Cycle

## Planning

## Source Dev

## Governance/Unit test

## Build

- Define the tasks
- Create a plan
- Create a work item
- Assign the work item to a developer

- Load the project/source artifacts from SCM
- Navigate, Analyze, Edit, Syntax check source code

- Compile
- Quality assurance
  - Debug
  - Code Coverage
  - Code review
  - Unit Testing

- Check-in/Deliver the source code
- Build

**CLM**

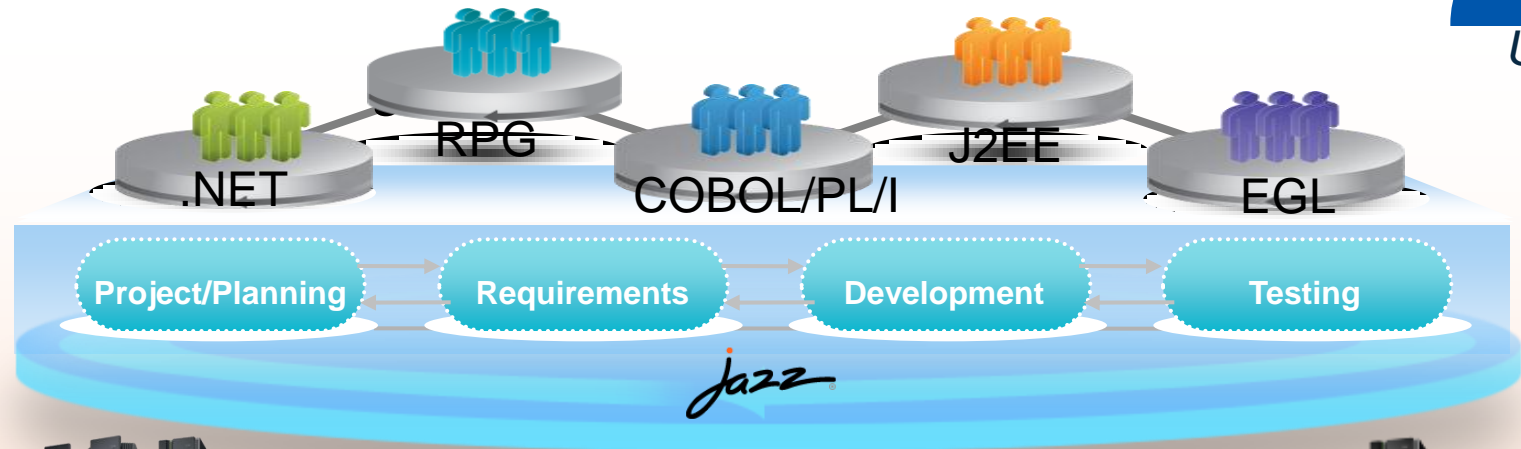
**RDz  
RTC**

**RDz  
RD&T  
RTC**

**RTC  
RDz**

# Collaborative application lifecycle management

**Deploy new, common team infrastructure for source control, change management and build that empowers your team with integrated collaboration, process automation and reporting**



- Rational Team Concert
- Rational Quality Manager
- Rational ClearCase
- Rational Requirements Composer
- Rational Build Forge
- Rational ClearQuest

*“Building an agile development team requires a multiplatform approach, and Sodifrance uses Rational Developer for System z and Rational Team Concert for System z to help application teams synchronize their efforts and improve collaboration.*

*Rational on System z offers a powerful and valuable combination for any company that wants to boost its development team’s productivity.”*

— Hugh Smith, Project Manager, Sodifrance



# Rational Team Concert – A single tool, many capabilities



## Work Items

55: Improve documentation for 4.4

Story 55

Summary: Improve documentation for 4.4

Details

Type: Story

Story Points: 1

Progress: 100%

Created: [Date]

Created By: Bill

Team Area: JUnit

Filed Against: JUnit

Tags:

Owned By: Bill Cassavelli

Context Menu: Add Work Item (Ctrl+Enter), Folders, Move Up (Alt+Up), Story, Track Build Item, Plan Item, Retrospective, Defect, Enhancement, Task, Set Default...

## Planning

JUnit Plan [4.4 m2]

JUnit Plan

Team Area: JUnit Team | Iteration: 4.4 m2 (5/2/09 - 6/1/09) | 10 Close

Bill Cassavelli | Closed items: 5 | Open items: 3

- Improve documentation for 4.4
- timeout doesn't work properly for >=2 cases in junit4.3? (2 hours)
- Jason Mitchell | Closed items: 2 | Open items: 5
- assertEquals throws NPE while comparing null elements (30 minutes)
- Improve documentation for 4.4 (1 hour)
- assertEquals array comparison doesn't handle null elements (2 hours)

## Source Control

Team

- Show Pending Changes
- Apply Patch...
- Accept... (Ctrl+Shift+F10)
- Deliver... (Ctrl+Shift+F9)
- Check-in (Ctrl+Shift+F11)
- Change File Properties...
- Add to Ignore List...

JUnit

- Jason Mitchell - 3: FileNotFoundException on junitvmwatcher.p
- JUnit Examples/src/junit/samples/money
  - Money.java
  - MoneyBag.java

Show change sets

## Builds – Continuous

Build junit I20080708-1553

Completed

Duration: 33 seconds

Start Time: July 8, 2008 3:53:58 PM

Completed: July 8, 2008 3:54:31 PM

Status Trend: [Green bars]

Reported Work Item: None reported

Contribution Summary

Downloads: 1 download

External Links: 1 link

Logs: 1 log

Repository Workspace: JUnitBuild

Work items: None included

Compile: 0 errors, 15 warnings

JUnit: 31 tests, 0 failures, 0 errors

General Informal

Requested by: Ja

Build Definition: JUnit

Build Engine: JUnit

Build History: 41

Tags: [None]

Deletion allowed: [Checked]

## Dashboards & Reporting

JUnit Project

Welcome to JUnit Project

JUnit Project Teams (1)

JUnit Project Queries

JUnit Project Description

Server Status

Current JUnit Project PL (2)

JUnit Project Teams (1)

JUnit Team

JUnit Project Queries

JUnit Project Description

Server Status

Current JUnit Project PL (2)

JUnit Plan

JUnit Release Plan

Recently modified (3) Category

JUnit

JUnit

Recently modified (3) Priority

High

Unassigned

## Method Enforcement and Automation

**Problem**  
A work item must be associated with the change set or a comment must be set

**Deliver (Failed)**

**Reason**  
All change sets should be associated with a work item or comment is planned for the current iteration

**Missing work item or comment**

This makes it easier for the team to track its progress through the iteration and allows other users to understand why your changes were made.

### Solutions

- Associate Existing Work Item
- Associate New Work Item
- Associate and Try Again (experimental)
- Override 'Descriptive Change Sets' Precondition

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)





# Multiple plan views facilitate continuous planning



**BRM Sprint 2 (1.0) Plan** | 28 items: 25 open, 3 closed | Ends in: 10 days

Plan Details | Edit

Planned Items | Links | Snapshots | Dashboard | Notes

View As: Roadmap | (3 items filtered)

Actions	Summary	Effective E:	Owned By	Rank
▶	Donors Can Choose to Support Multiple Organization	-	Deb	1
▶	Donor Dividend Allocation Criteria	-	Marco	3
	Widget Disposed Exception	1.2 weeks	Bob	-
	Browser Exception	2 hours	Bob	-
▶	Requests sent in form of email	-	Tanuj	4
	Some messages are not externalized	3 days	Bob	-
	SWT Exception	1.5 days	Marco	-
▶	Frequency of dividend transfer	-	Deb	-

**BRM Sprint 2 (1.0) Plan** | 30 items: 25 open, 5 closed | Ends in: 6 days

Plan Details

Planned Items | Links | Snapshots | Dashboard | Notes

View As: Taskboard

Story	Open	In Progress
Donor Dividend Allocation Criteria	Implement - Donor Dividend Allocation Criteria	
Frequency of dividend transfer		Implement - Frequency of dividend transfer

30 items: 25 open, 5 closed | Ends in: 6 days

Plan Details

Planned Items | Links | Snapshots | Dashboard | Notes

View As: Planned Time

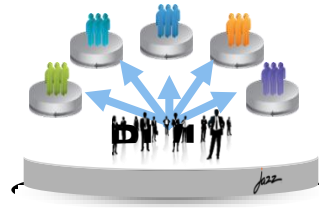
Person	Closed Items	Open Items	Load	Time
Bob	0	1	0/8	+8 h
cp	0	1	0/8	+8 h
Deb	3	5	168/8	-160 h
Marco	2	14	92/8	-84 h
Tanuj	0	0	0/8	+8 h

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



# Rational Developer for System z: An Integrated Development Environment for System z

Integration with Team  
Concert for Lifecycle and  
Source Management

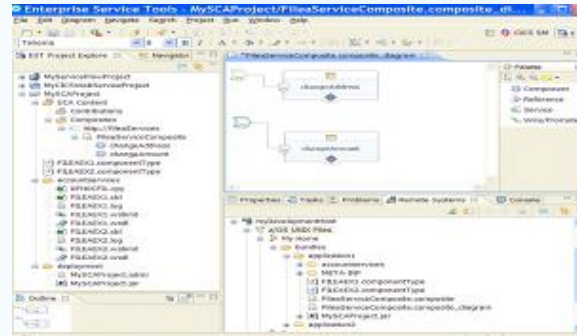


Access to typical System  
z sub-system functionality  
in z/OS, CICS, IMS, DB2,  
WAS

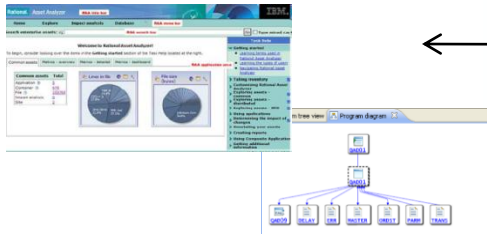
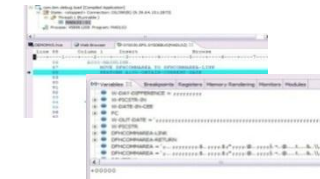


Integration with Asset  
Analyzer for Application  
Understanding and Impact  
Analysis

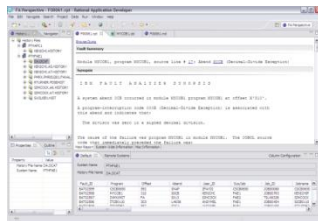
Rational Developer for System z



**New** Out of the Box debugger  
and code coverage  
capabilities



Integration with Fault  
Analyzer for Dump Analysis

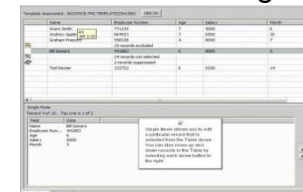


A modern IDE for productive development of  
cross-platform applications written in COBOL,  
PL/I, ASM, Java, EGL or C/C++ in System z  
CICS, IMS, DB2, Batch applications



Integration with RD&T for flexible  
access to System z environment

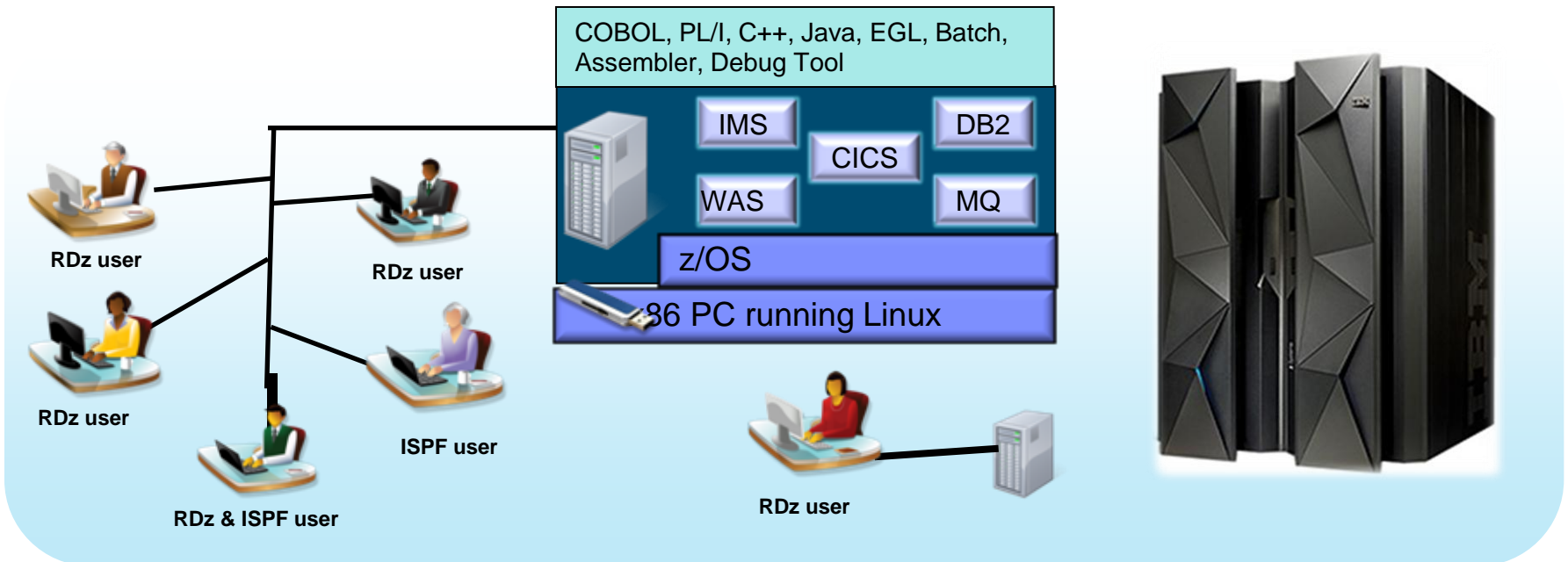
Integration with File  
Manager for file and test  
data handling



Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

# Rational Development and Test Environment for System z

*The ultimate in modern application development for System z*



- Increase availability of z/OS testing environment and resources
  - Liberate developers to rapidly prototype new applications
  - Develop and test System z applications anywhere, anytime!
  - Eliminate costly delays by reducing dependencies on operations staff
- Improve quality and lower risk via automation, measurement, and collaboration
- Focus on what is required for the change at hand, then scale

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.

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



# Development Life Cycle

## Planning

## Source Dev

## Governance/Unit test

## Build

- Define the tasks
- Create a plan
- Create a work item
- Assign the work item to a developer

- Load the project/source artifacts from SCM
- Navigate, Analyze, Edit, Syntax check source code

- Compile
- Quality assurance
  - Debug
  - Code Coverage
  - Code review
  - Unit Testing

- Check-in/Deliver the source code
- Build

**CLM**

**RDz  
RTC**

**RDz  
RD&T  
RTC**

**RTC  
RDz**

# Demo: What we will show

- Business Analyst
  - opening a new work item for Developer using RTC
- Software Developer
  - discovering the new work item in his work queue in RTC
  - accessing and loading his code into his work env using RTC
  - working to setup work using integrated RDz/RTC
  - editing the COBOL source file and figuring out how to code the requirement stated in the work item
  - code the requirement, personal build and test
- Build/Release Admin
  - team build, test and deploy
  - mark the work item Complete in RTC
- Business Analyst
  - Verify that requirement is complete and close the work item in RTC

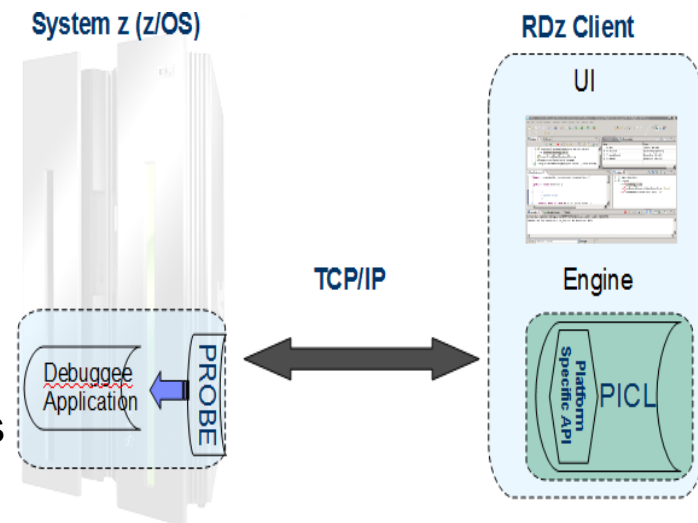
# Integrated Debugger

## ✓ Host-offload architecture:

- ✓ Remote debugger with only a small footprint on the mainframe:
  - Leverages workstation CPUs enabling faster processing of debug information
  - Enables scalability and reliability
- ✓ Debugger client is supported on Windows and Linux

## ✓ Simple and Secure Connections:

- ✓ Single client can handle multiple debug sessions on multiple hosts or an application the spans multiple systems
- ✓ Client initiated debug – no need to specify client IP address and port (v9.0.1.2)
- ✓ SSL/TLS support





# Enhanced Application Quality – Code Coverage



- Line Level Code Coverage - provides tools to measure and report on test coverage of an application
  - Leverages the Integrated Debugger technology
  - Indicating what source code lines were tested and remain to be tested

**Code Coverage Report**

Code coverage report, generated Oct 18, 2011 12:19:59 PM

Element	Coverage	Covered Lines	Total Lines
SAM1	75%	117	156
SAM1	75%	117	156
VENKATU.COBOL.SYSDEBUG(SAM1).cob	75%	117	156
SAM1()	75%	117	156

Launch Name	Launch Date
SAM1-2011_10_18_121959	Oct 18, 2011 12:19:59 PM
SAM1-2011_10_17_163634	Oct 17, 2011 4:36:34 PM
SAM1-2011_10_17_160130	Oct 17, 2011 4:01:30 PM
SAM1-2011_10_14_132043	Oct 14, 2011 1:20:43 PM
SAM1-2011_10_14_130134	Oct 14, 2011 1:01:34 PM
SAM1-2011_10_14_124502	Oct 14, 2011 12:45:02 PM

```

374 -----*A-1-B-----2-----3-----4-----5-----6-----7-----
375          PERFORM 210-PROCESS-ADD-TRAN
376          WHEN 'DELETE'
377          PERFORM 220-PROCESS-DELETE-TRAN
378          WHEN OTHER
379          IF TRAN-COMMENT NOT = '*'
380          MOVE 'INVALID TRAN CODE:' TO ERR-MSG-DATA1
381          MOVE TRAN-CODE TO ERR-MSG-DATA2
382          PERFORM 299-REPORT-BAD-TRAN
383          END-IF
384          END-EVALUATE
385          END-IF
386          MOVE TRAN-KEY TO WS-PREV-TRAN-KEY
387          IF WS-TRAN-OK = 'Y'
388          PERFORM 830-REPORT-TRAN-PROCESSED
389          END-IF
390          END-IF .
391
392 200-PROCESS-UPDATE-TRAN.
393  ADD +1 TO NUM-UPDATE-REQUESTS.
394  PERFORM 720-POSITION-CUST-FILE.
395  IF CUST-KEY NOT = TRAN-KEY OR WS-CUST-FILE-EOF = 'Y'
396  MOVE 'NO MATCHING KEY:' TO ERR-MSG-DATA1
397  MOVE TRAN-KEY TO ERR-MSG-DATA2
398  PERFORM 299-REPORT-BAD-TRAN
399  ELSE
400  *
401  *   Subroutine SAM2 will apply an update to a customer record
402  *
403  CALL 'SAM2' USING CUST-REC, TRANSACTION-RECORD,
404                  WS-TRAN-OK, WS-TRAN-MSG
405  IF WS-TRAN-OK NOT = 'Y'
406  MOVE WS-TRAN-MSG TO ERR-MSG-DATA1
407  MOVE SPACES TO ERR-MSG-DATA2
408  PERFORM 299-REPORT-BAD-TRAN
    
```

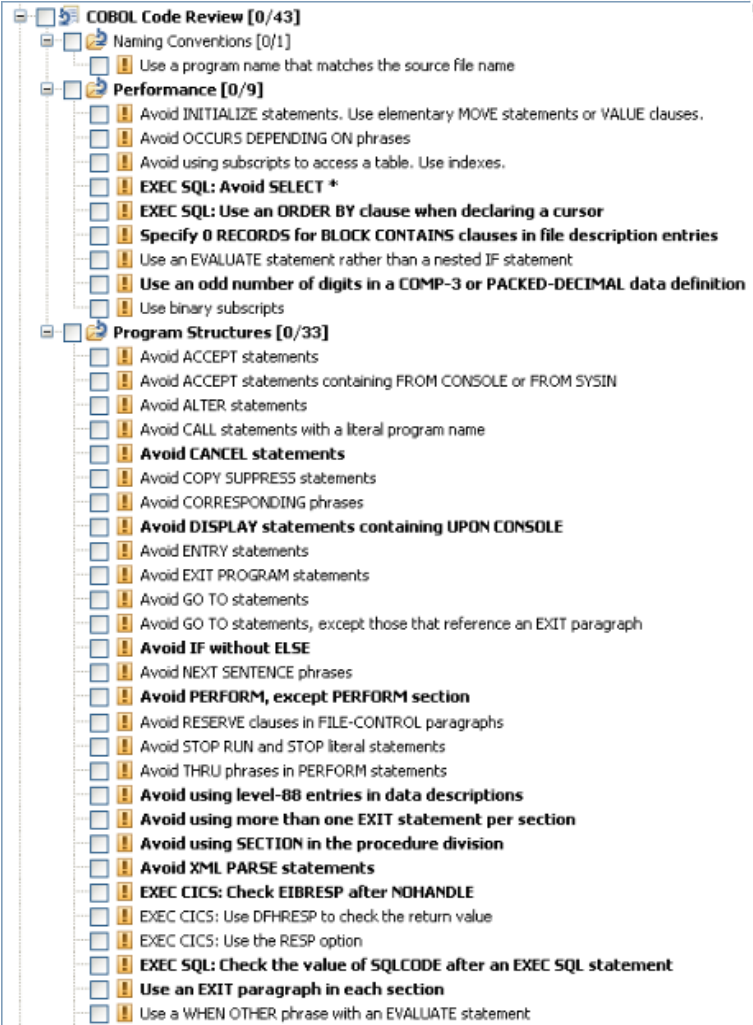
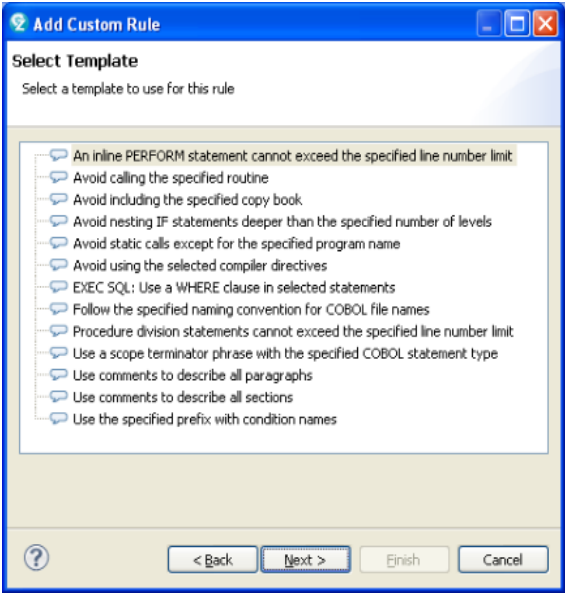
Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)





# Enhanced Quality & Structure Analysis – Code review

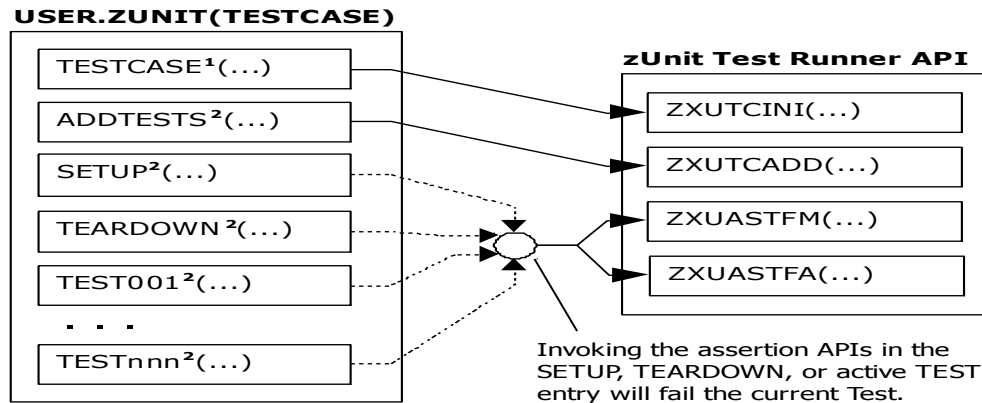
- Code Review/Governance - provides predefined rules and templates for COBOL and PL/I applications
  - Ensure adherence to corporate standards
  - Custom rules for COBOL and PL/I



Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

# zUnit – Unit testing framework for z/OS

- Frameworks that assist developers in writing code to perform repeatable, self-checking unit tests are collectively known as xUnit.
- [xUnit](#) defines a set of concepts that together provide a light-weight architecture for implementing unit testing frameworks.
  - JUnit, for example, is a very popular instance of the xUnit architecture.
- **zUnit** is a xUnit instance for System z
- Goal is to encourage the **continuous integration and continuous testing** methodology for System z Application development and maintenance



<sup>1</sup>Language-specific details:

- In COBOL, this is the first program appearing in the Test Case source file and it will be invoked by the Test Runner for Test Case initialization.
- In PL/I, this is the procedure declared with option(fetchable) in the Test Case source file and it will be invoked by the Test Runner for Test Case initialization.

<sup>2</sup>Language-specific details:

- In COBOL, these are expected to be subprograms (non-nested and therefore compatible with FUNCTION-POINTER).
- In PL/I, these are expected to be internal procedures that are declared at the package level (non-nested).

# zUnit Capabilities

- zUnit Test Runner
  - Runs on z/OS
    - Installed and configured on z/OS as part of RDz Host install and customization
  - Fetches and runs the Test Suite referred to in a zUnit configuration file
  
- zUnit Wizard used to generate Test Cases
  - RDz client feature
  - Eclipse based wizards allow creation of:
    - Template Test Cases are generated in COBOL or PL/I
      - Simple pass/fail **assertion** API

## (RDz v9.1) Complete COBOL test cases:

- Identify the interface or set of copy book(s)
- Generate XML Schema to represent the interface
- Generate XML files where you would specify test input and expected output
- Generate a Test Case based on the XML file
- (Optionally) Generate stubs for called programs



- RDz viewers/editors for unit test XML results

# What have we learned

- What is Software DevOps and its importance in today's Enterprise
- The tools that help manage every day life of Software DevOps practitioners
- How to use some of the tools and where to find more...

## Summary

- Many companies spend more than 70% on keeping lights on, and that amount is increasing
- IT organizations have problems modifying applications at speed of business
- IBM provides a structured approach to incrementally modernize your portfolio based on business priorities
- Change without a Plan is chaos
- A Plan without change is stagnation
- Business goals change
  - applications need to change to address them
- Continual renewal is required
  - tools help to guide, govern, drive, and accomplish this change

# Getting started

## Next steps to modernize your enterprise applications

[www.ibm.com/rational/modernization](http://www.ibm.com/rational/modernization)



- [Try latest System z software for free](#)
- [Sign up for free web-based training](#)
- [Join IBM Rational Cafe Communities](#)
- [Get prescriptive service solutions](#)
- [Success stories](#)
- [Latest news on System z twitter](#)
- [Latest customer videos](#)
- [Latest skills: System z job board](#)

# QUESTIONS

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



# Back up and Reference

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

# Development Life Cycle

Planning

Source Dev

Governance/Unit test

Build

- Define the tasks
- Create a plan
- Create a work item
- Assign the work item to a developer

- Load the project/source artifacts from SCM
- Navigate, Analyze, Edit, Syntax check source code

- Compile
- Quality assurance
  - Debug
  - Code Coverage
  - Code review
  - Unit Testing

- Check-in/Deliver the source code
- Build

**CLM**

RDz  
RTC

RDz  
RD&T  
RTC

RTC  
RDz

# Any process: Executable and repeatable

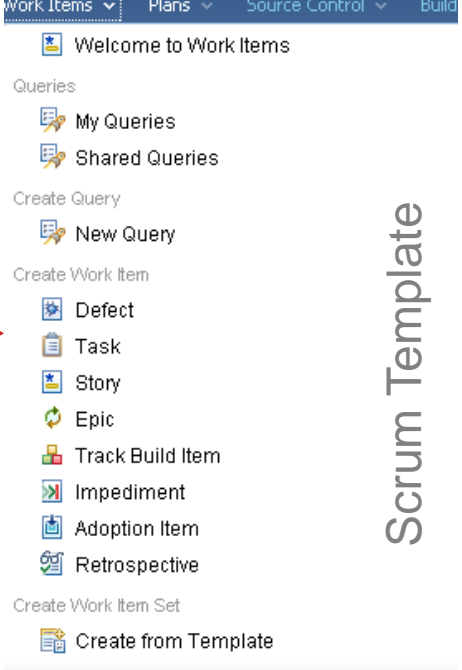
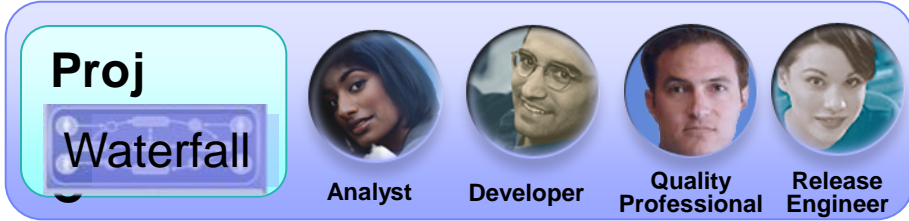
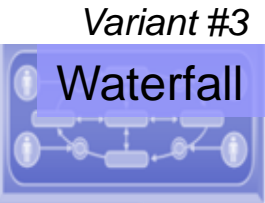
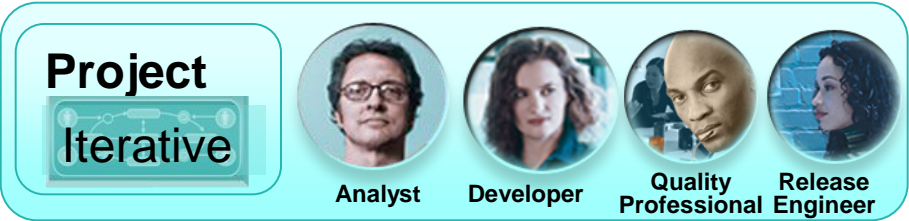
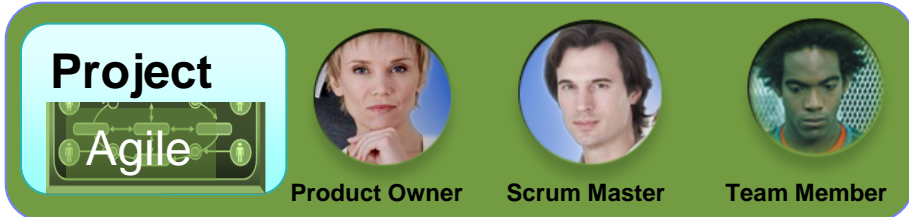
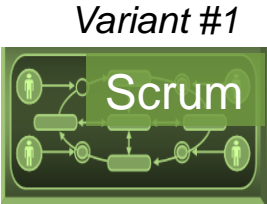
## Use ONE tool to support both agile and non-agile



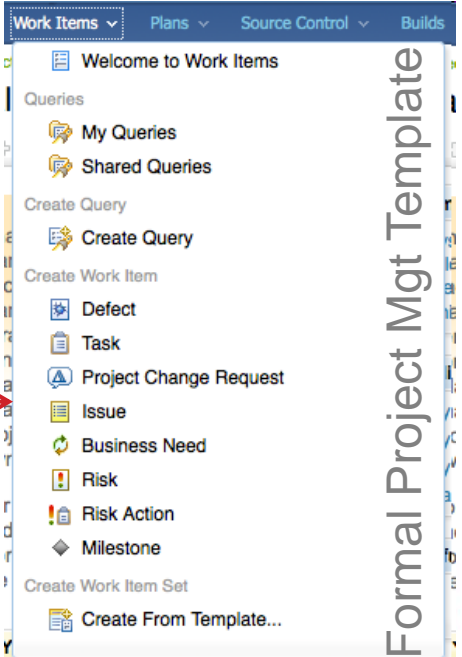
**Process Architect**



**Project Manager**



Scrum Template



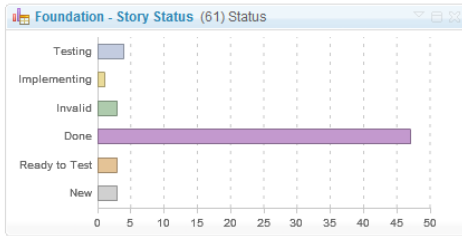
Formal Project Mgt Template

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

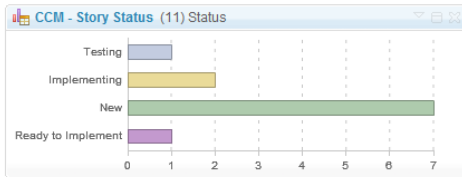
# Progress Tracking - Everyone can see live project status

## Story Status Current Iteration

Shows the status of all stories planned for the current iteration



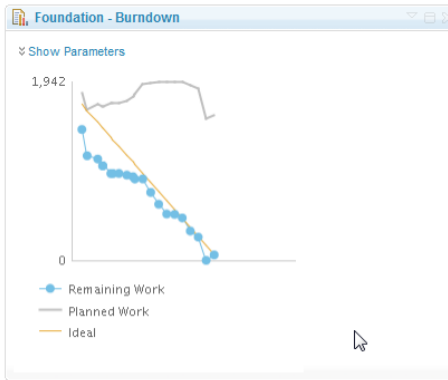
- ### Foundation - Current Stories (61)
- 155918: Provide TVT testcases and testcase updates to TVT team
  - 155399: [Repository - M14] Performance, quality and robustness enhancements
  - 154407: [Continuation] Manage and repair DB inconsistencies during 3.0.1 M14
  - 155024: Track the fixing of hangs occurring on shutdown of the CLM server
  - 155033: Explore servlet dispatching enablement as a backup option
  - 155030: [Application SDK - 14] Performance, quality and robustness enhancements
  - 155120: LPA Web UI fixes and polish
  - 111848: User can validate consistency of configuration information across contexts
  - 155168: Fix bugs and improve the quality of LPA services
  - 155624: update JREs for 3.0 iFix1 and 3.0.1
- Page 1 of 7



- ### CCM - Current Stories (11)
- 150244: address "run as administrator" issues on win7
  - 142127: As a user I want to see all extended state groups in taskboards
  - 150958: Improve the performance of loading a work item

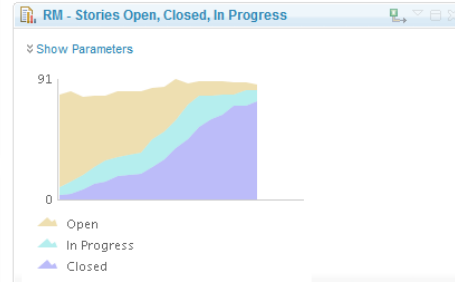
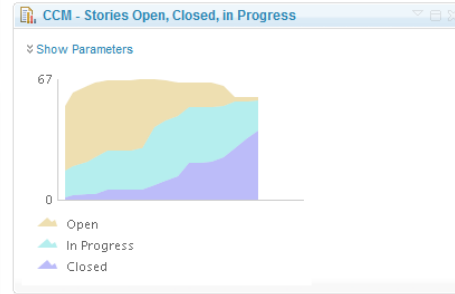
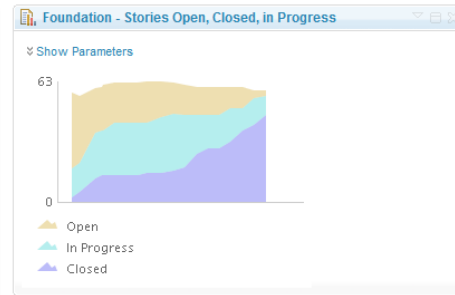
## Burndown Current Iteration

Shows the remaining amount of estimated work in hours of work items planned for the current iteration.



## Stories Open/Closed/In Progress

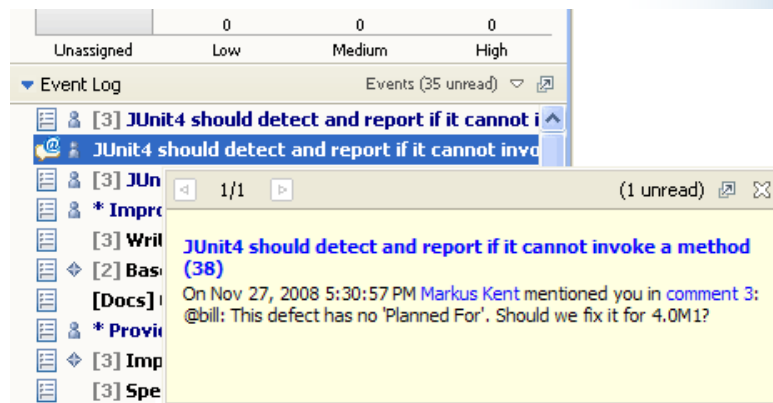
Shows the number of stories which are open, in progress, done during the iteration.



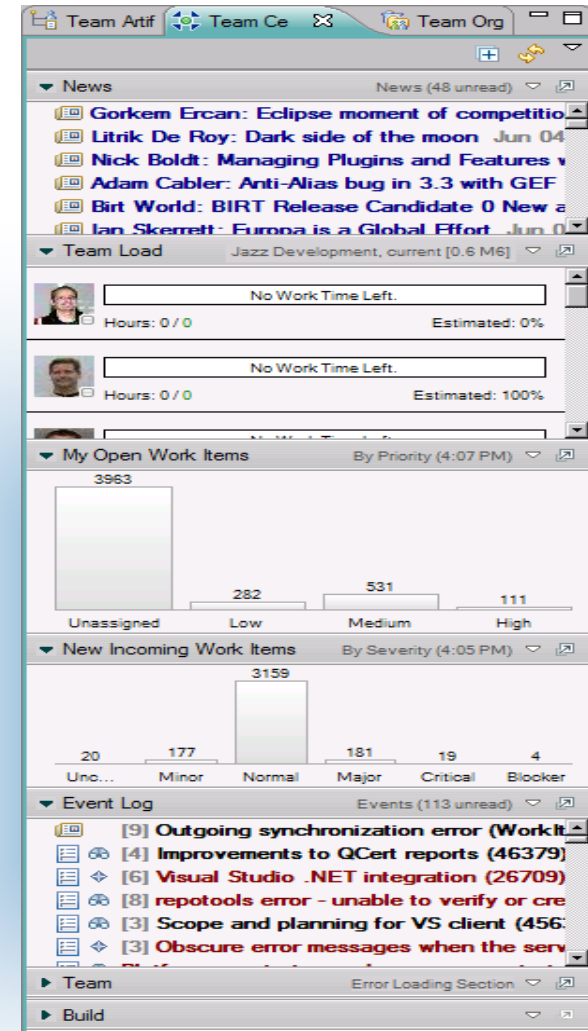
# In-context Collaboration – Team View

## Team Central

- Shows what is happening on project:
  - News & events
  - What's being worked on
  - Changes
- Configurable (RSS feeds) - New kinds of information easily added
- Personalized, Persistent - Each team member can tailor to their needs



The screenshot shows a table with columns for 'Unassigned', 'Low', 'Medium', and 'High' priority items, all with a count of 0. Below the table is an 'Event Log' section with 35 unread events. One event is selected and expanded, showing a title: 'JUnit4 should detect and report if it cannot invoke a method (38)'. The event details include a timestamp 'On Nov 27, 2008 5:30:57 PM', the author 'Markus Kent', and the content: '@bill: This defect has no 'Planned For'. Should we fix it for 4.0M1?'.



The screenshot shows a dashboard with several widgets. At the top is a 'News' section with 48 unread items, listing articles by Gorkem Ercan, Litrik De Roy, Nick Boldt, Adam Cabler, Birt World, and Ian Skerrett. Below that is a 'Team Load' section for 'Jazz Development, current [0.6 M6]' showing 'No Work Time Left' for two team members. The 'My Open Work Items' section shows a bar chart with 3963 total items, broken down by priority: Unassigned (282), Low (531), Medium (111), and High. The 'New Incoming Work Items' section shows a bar chart with 3159 total items, broken down by severity: Unc... (20), Minor (177), Normal (181), Major (19), and Blocker (4). At the bottom is an 'Event Log' with 113 unread events, listing items like 'Outgoing synchronization error (Workit)', 'Improvements to QCert reports (46379)', 'Visual Studio .NET integration (26709)', 'reptools error - unable to verify or cre', and 'Scope and planning for VS client (456)'. The bottom navigation bar shows 'Team' (Error Loading Section) and 'Build'.

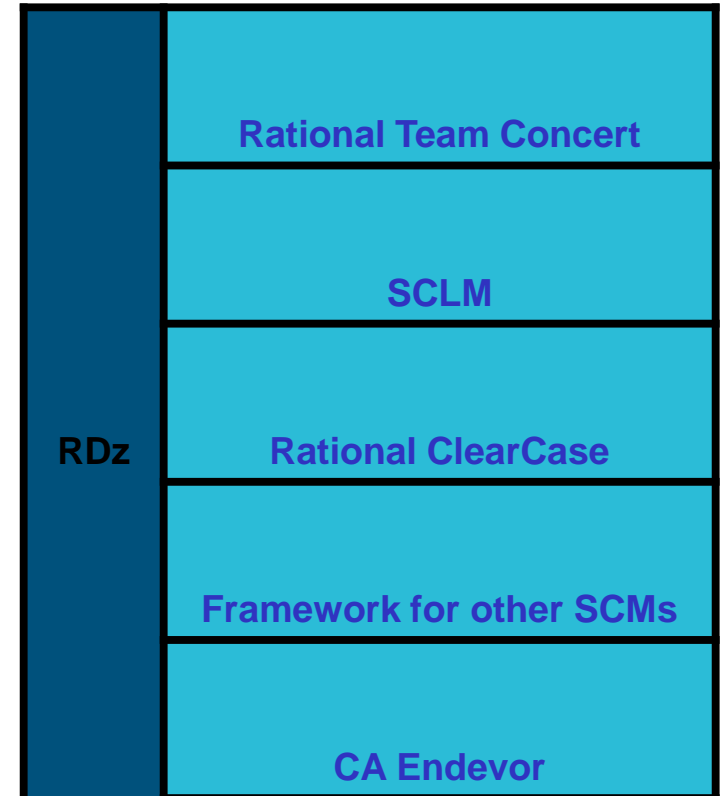
Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

# Development Life Cycle

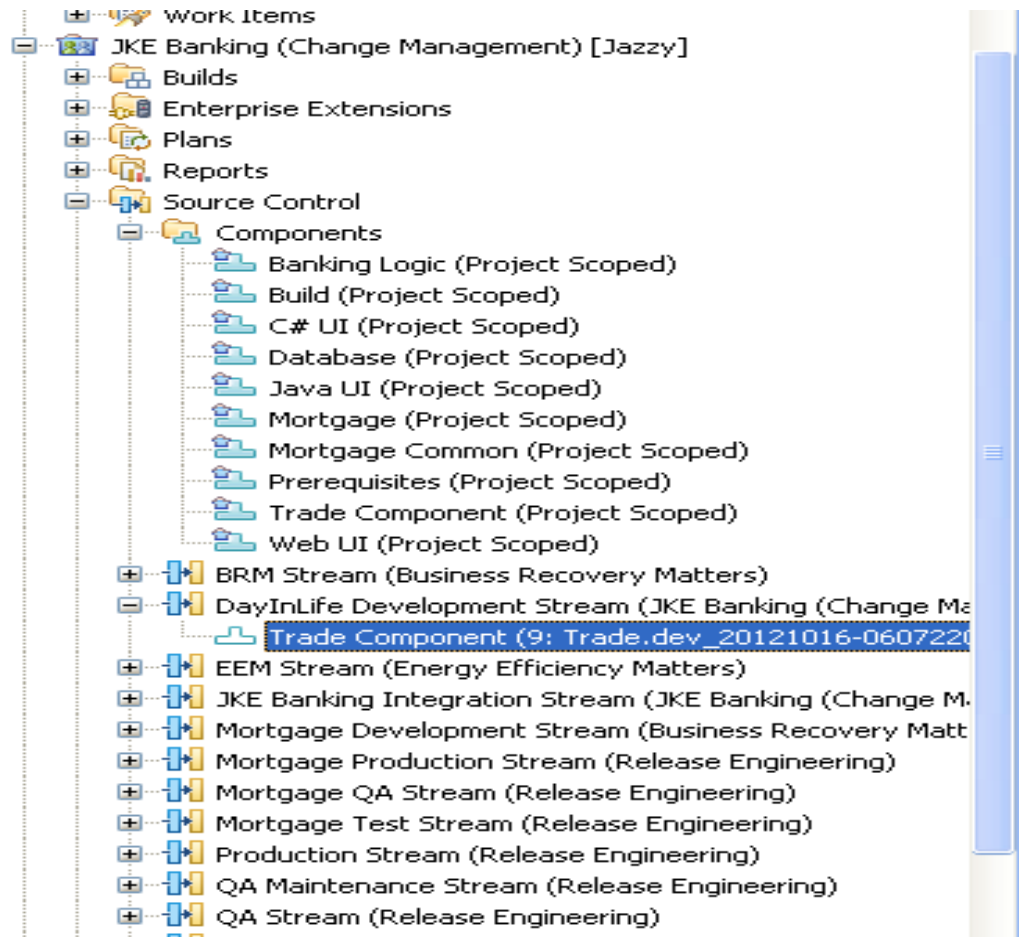
Planning	Source Dev	Governance/Unit test	Build
<ul style="list-style-type: none"> <li>• Define the tasks</li> <li>• Create a plan</li> <li>• Create a work item</li> <li>• Assign the work item to a developer</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Load the project/source artifacts from SCM</b></li> <li>• <b>Navigate, Analyze, Edit, Syntax check source code</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Compile</b></li> <li>• <b>Quality assurance</b> <ul style="list-style-type: none"> <li>• Debug</li> <li>• Code Coverage</li> <li>• Code review</li> <li>• Unit Testing</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <b>Check-in/Deliver the source code</b></li> <li>• <b>Build</b></li> </ul>
CLM	RDz RTC	RDz RD&T RTC	RTC RDz

# RDz Source Code Integration

- Rational's Strategic Source Code tooling is RTC and RDz provides tight integration
- RDz offers integration into a variety of other Source Code Management (SCM) tools as well as a framework for creating SCM integration on your own (CARMA)
- Variety of vendors supply plug-ins to RDz to provide easy access to processes and source code controlled by their products

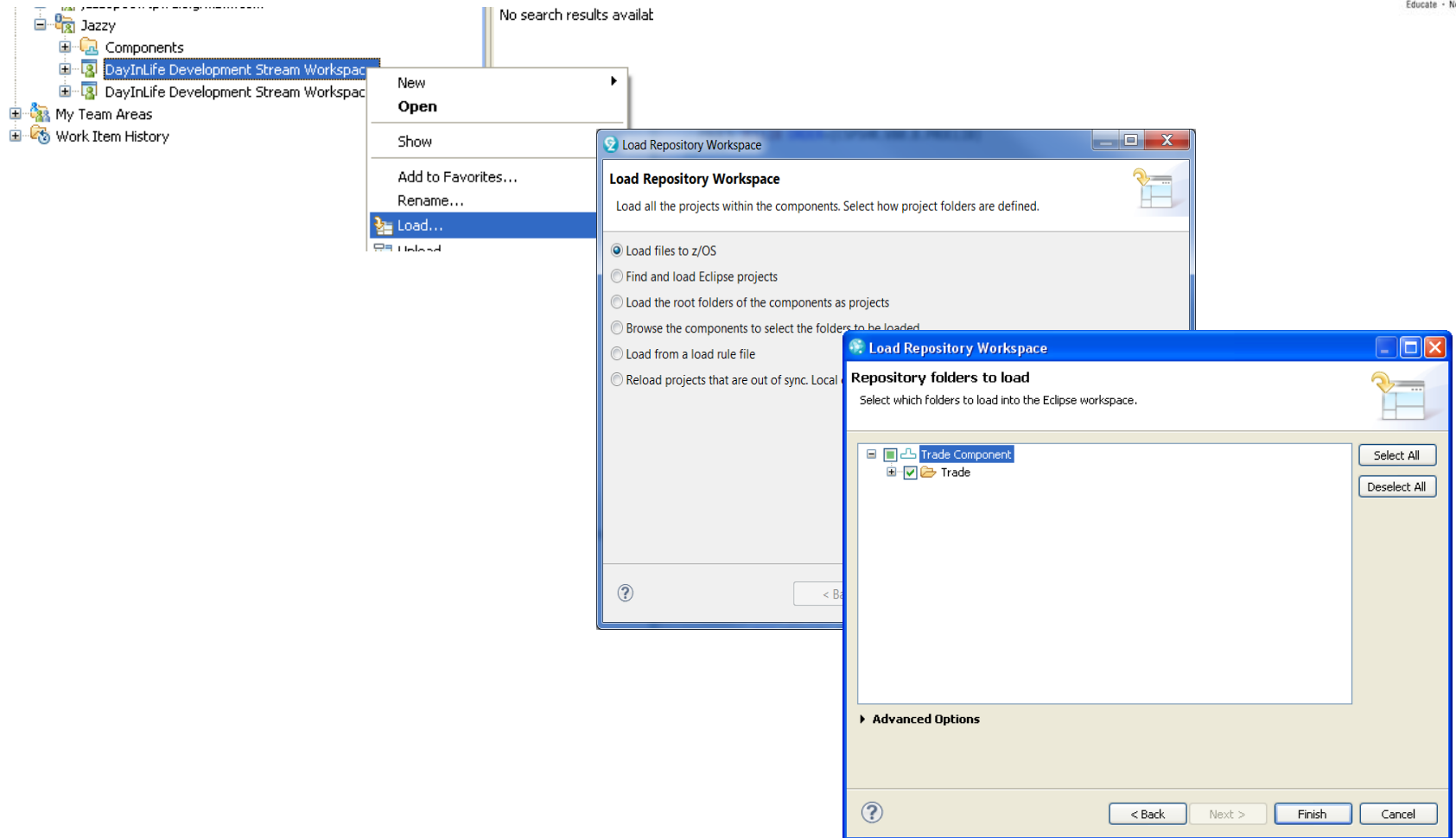


# Source Control Management



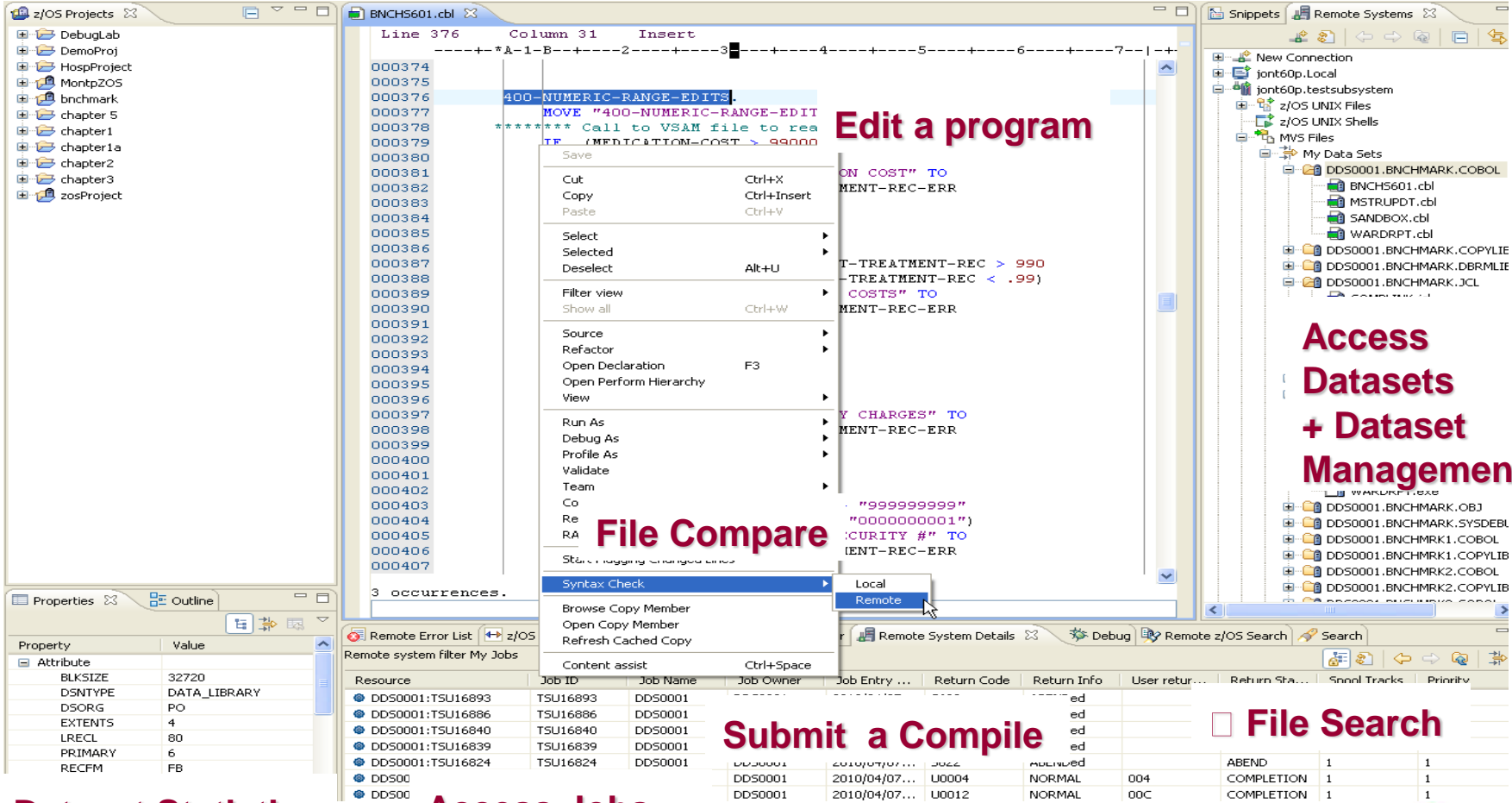


# Load the source artifacts



# The Benefits of RDz

Instead of maneuvering to access panels and working **sequentially**, in RDz the functionality you need is always in-focus – you work **concurrently**



**Edit a program**

```

Line 376 Column 31 Insert
-----+*A-1-B-----2-----3-----4-----5-----6-----7-|-+
000374
000375
000376
400-NUMERIC-RANGE-EDITS
000377 MOVE "400-NUMERIC-RANGE-EDIT
000378 ***** Call to VSAM file to rea
000379 IE (MEDICATION-COST > .99000
000380
000381
000382
000383
000384
000385
000386
000387
000388
000389
000390
000391
000392
000393
000394
000395
000396
000397
000398
000399
000400
000401
000402
000403
000404
000405
000406
000407
  
```

**File Compare**

**Access Datasets + Dataset Management**

**Submit a Compile**

Resource	Job ID	Job Name	Job Owner	Job Entry ...	Return Code	Return Info	User retur...	Return Sta...	Spool Tracks	Priority
DDS0001:TSU16893	TSU16893	DDS0001								
DDS0001:TSU16886	TSU16886	DDS0001								
DDS0001:TSU16840	TSU16840	DDS0001								
DDS0001:TSU16839	TSU16839	DDS0001								
DDS0001:TSU16824	TSU16824	DDS0001								
DDS0001	2010/04/07...	U0004			ABEND			ABEND	1	1
DDS0001	2010/04/07...	U0004			NORMAL		004	COMPLETION	1	1
DDS0001	2010/04/07...	U0012			NORMAL		00C	COMPLETION	1	1

**File Search**

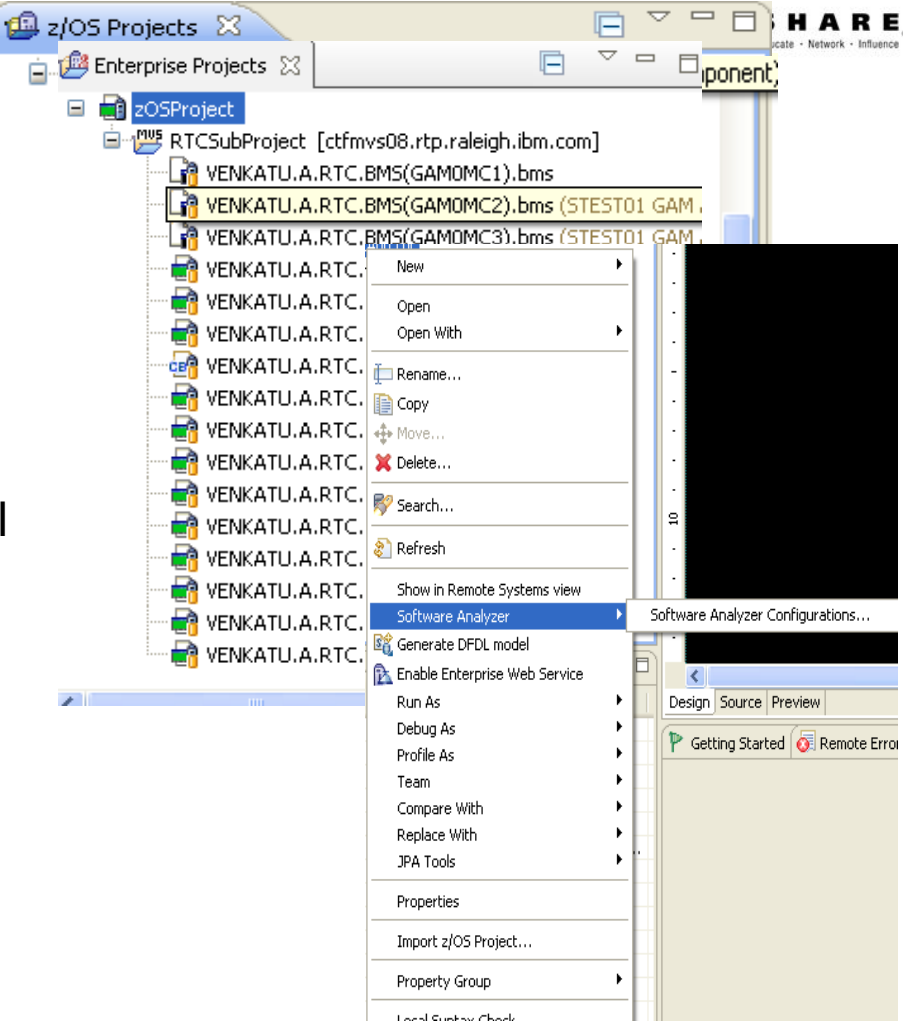
**Dataset Statistics**

Property	Value
Attribute	
BLKSIZE	32720
DSNTYPE	DATA_LIBRARY
DSORG	PO
EXTENTS	4
LRECL	80
PRIMARY	6
RECFM	FB

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)

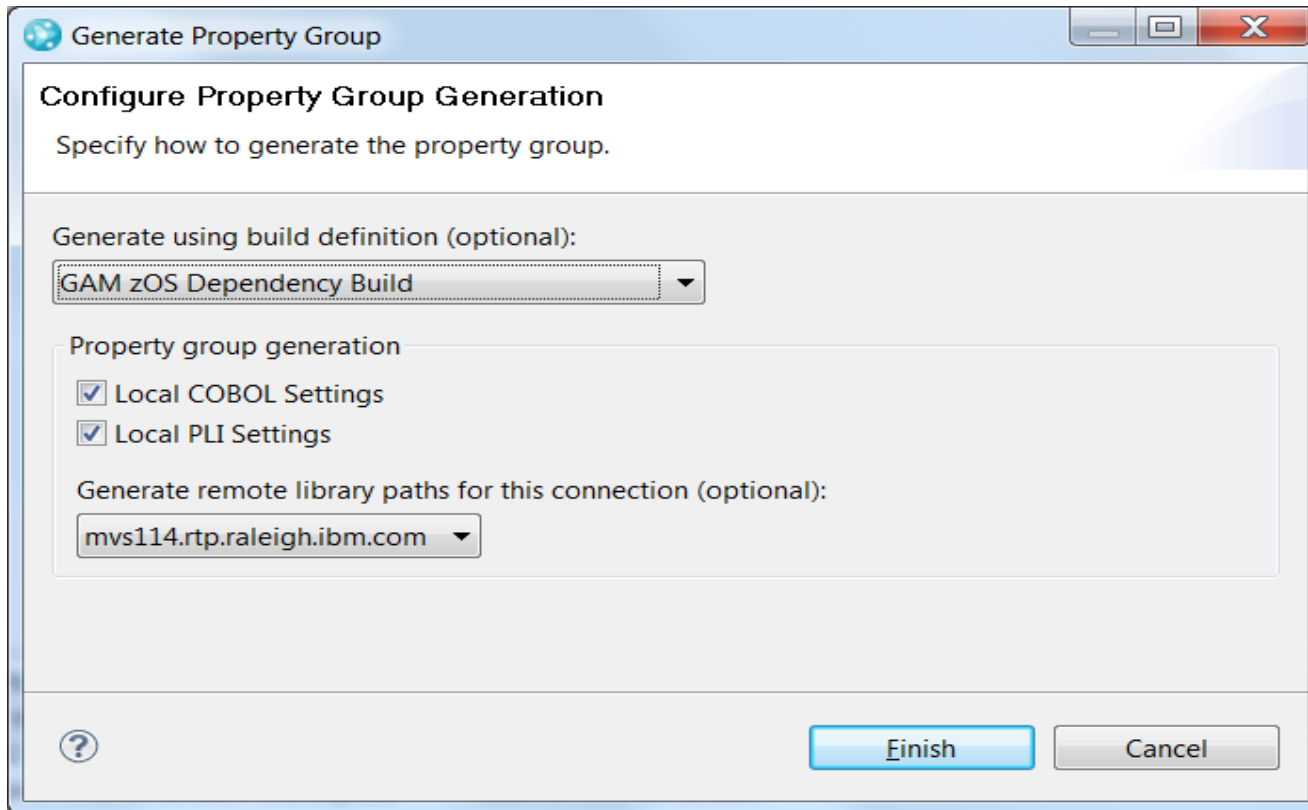
# RDz and RTC together

- Once the project is loaded, it will appear in the RDz z/OS projects view
- RDz augments the development productivity & experience
  - Appropriate editors (COBOL, maps, etc.) and functions (content assist, real time syntax check, etc.)
  - High value functions (Enterprise web services, SFM, Code review, Unit testing, program analysis/control flow etc.)



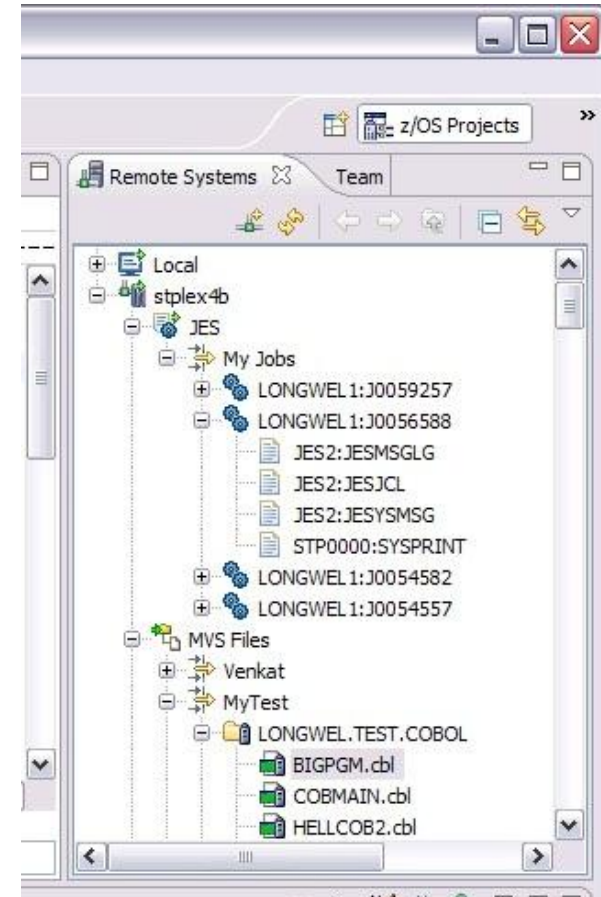
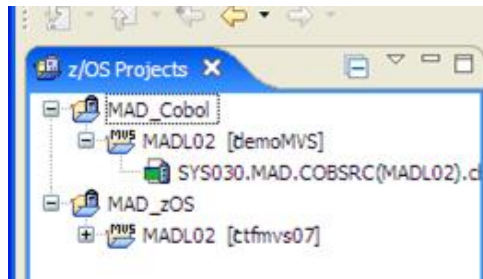
# Create a Property Group

- Generate property groups for your project based on RTC build definition
- Allows RDz to resolve the dependencies and thus offer all the tooling



# Navigate datasets and jobs live on zOS

- 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 capabilities in RDz

RDz at a high level has different types of editors

- LPEX Editor
  - Supports editing of COBOL, PLI, HLASM, JCL, C/C++, Rexx etc.
  - Provides ISPF like edit experience including prefix commands, command line and even look and feel
  - Supports advanced edit functions for COBOL, PLI and HLASM like real time syntax checking, content assist
- COBOL, PLI, and JCL advanced editors
  - Based on the Eclipse editor infrastructure, provide more advanced edit capabilities like quick fixes, hyper-linking, hover, easy navigation between various edit sessions or within the same edit session.
  - Supports real time syntax checking, content assist, key word highlighting etc.

Line	Column	Insert
000027	*	
000028	*****	
000029	IDENTIFICATION DIVISION.	
000030	PROGRAM-ID. GAMOVDB.	
000031	DATA DIVISION.	
000032	WORKING-STORAGE SECTION.	
000033		
000034	COPY GAMOBMD.	
000035	COPY GAMOBDD.	
000036	COPY GAMOBED.	
000037	COPY GAMOBEC.	
000038	COPY GAMOBPD.	
000039	01 DATABASECONNECTION.	
000040	49 SCHEMA	PIC
000041	49 PIC X VALUE '...'	
000042	01 ITABLEPREPARE.	
000043	49 ITABLE-VAR-LEN	PIC S
000044	49 ITABLE-VAR-TXT	PIC M
000045	01 INVENTORYPREP.	
000046	02 PIC X(12) VALUE 'INSERT'	
000047	02 INVNTDBCON PIC X(9).	
000048	02 TABLENAME PIC X(11).	

```

PROCEDURE DIVISION.

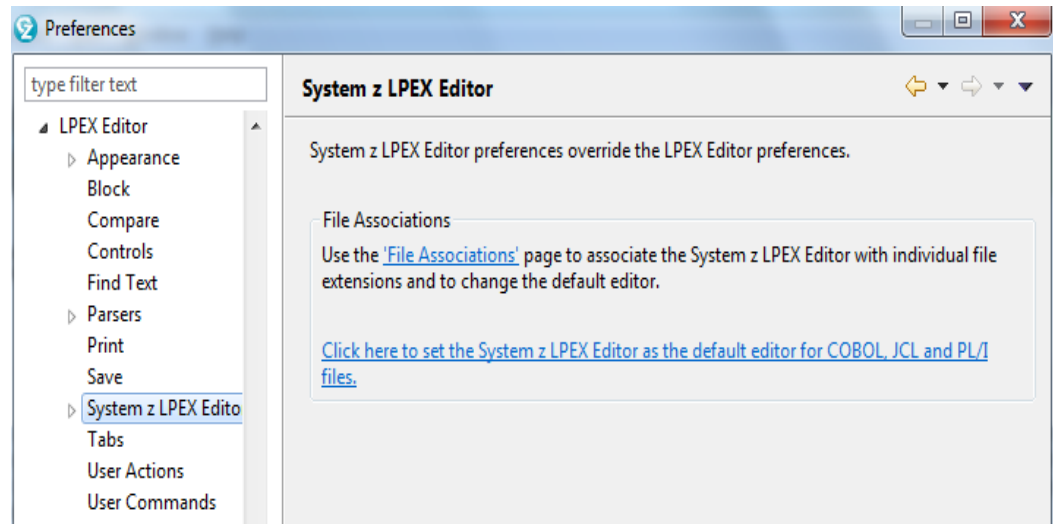
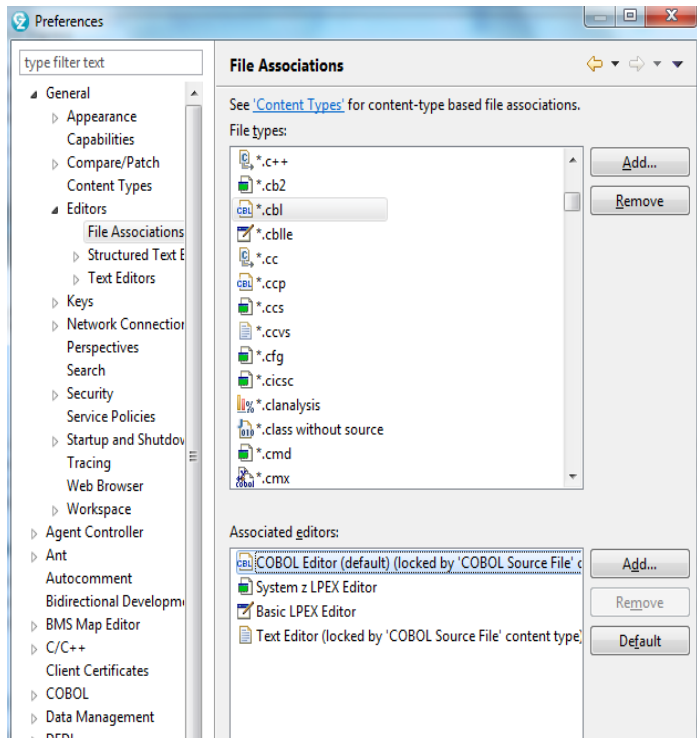
INITIALIZE INPUTS-OUTPUTS.
INITIALIZE INVO.
INITIALIZE LOCINVO.
MOVE LOW-VALUE TO LOCINVO.
MVE LOW-VALUE TO INVO.
  
```

Change to "MOVE"

# COBOL, PL/I and JCL Editor – the new default

- New workspaces created in RDz 9.x
  - New editors are the default
  - Single click switch to LPEX is available

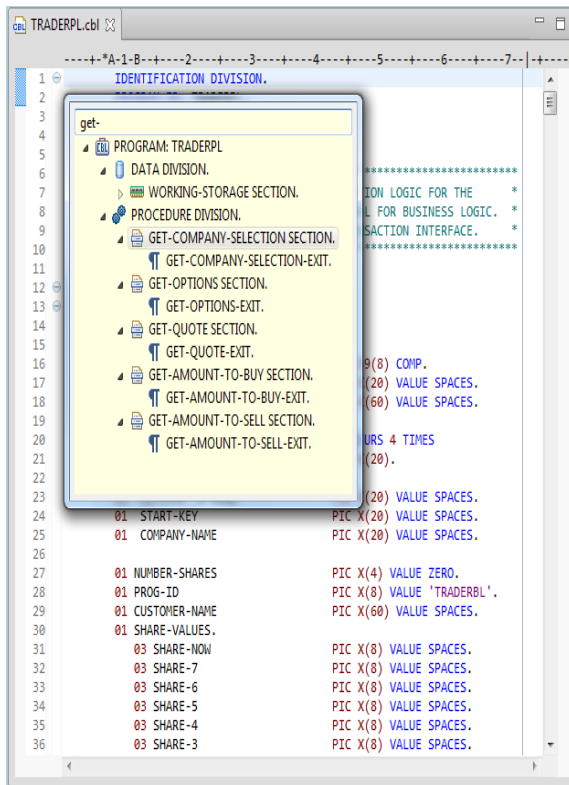
Applicable to source and include/copybooks



# COBOL, PL/I and JCL Editor improvements

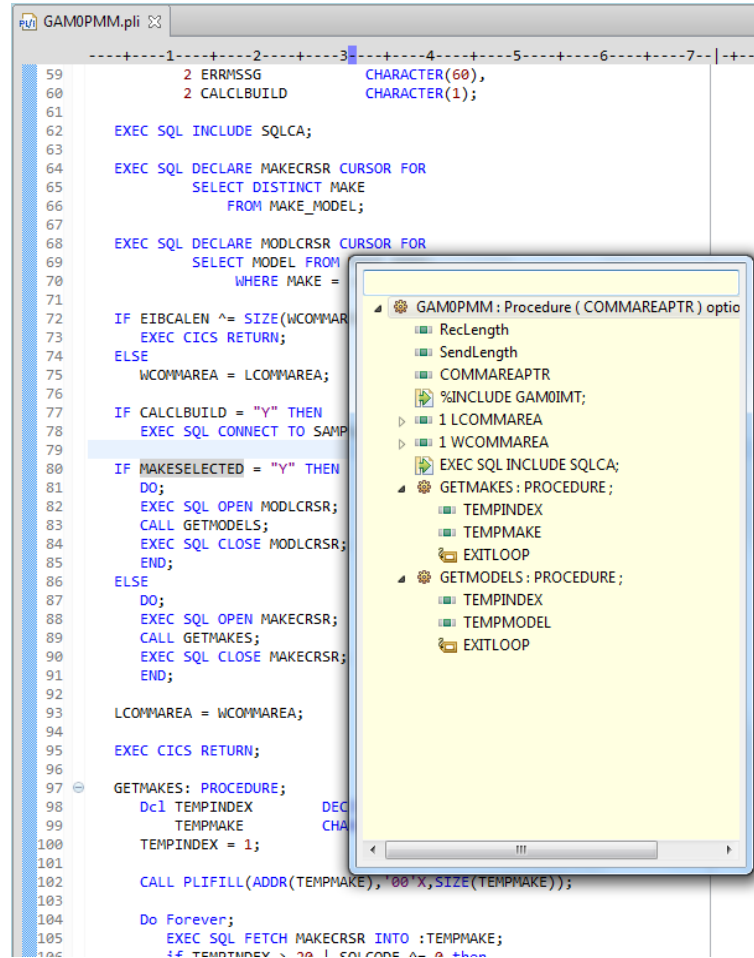
## ■ Quick Outline

- Press Ctrl+O to activate



The screenshot shows a COBOL editor window titled 'TRADERPL.cbl'. A project tree is visible on the left side, showing the following structure:

- PROGRAM: TRADERPL
  - DATA DIVISION.
  - WORKING-STORAGE SECTION.
  - PROCEDURE DIVISION.
    - GET-COMPANY-SELECTION SECTION.
      - GET-COMPANY-SELECTION-EXIT.
    - GET-OPTIONS SECTION.
      - GET-OPTIONS-EXIT.
    - GET-QUOTE SECTION.
      - GET-QUOTE-EXIT.
    - GET-AMOUNT-TO-BUY SECTION.
      - GET-AMOUNT-TO-BUY-EXIT.
    - GET-AMOUNT-TO-SELL SECTION.
      - GET-AMOUNT-TO-SELL-EXIT.



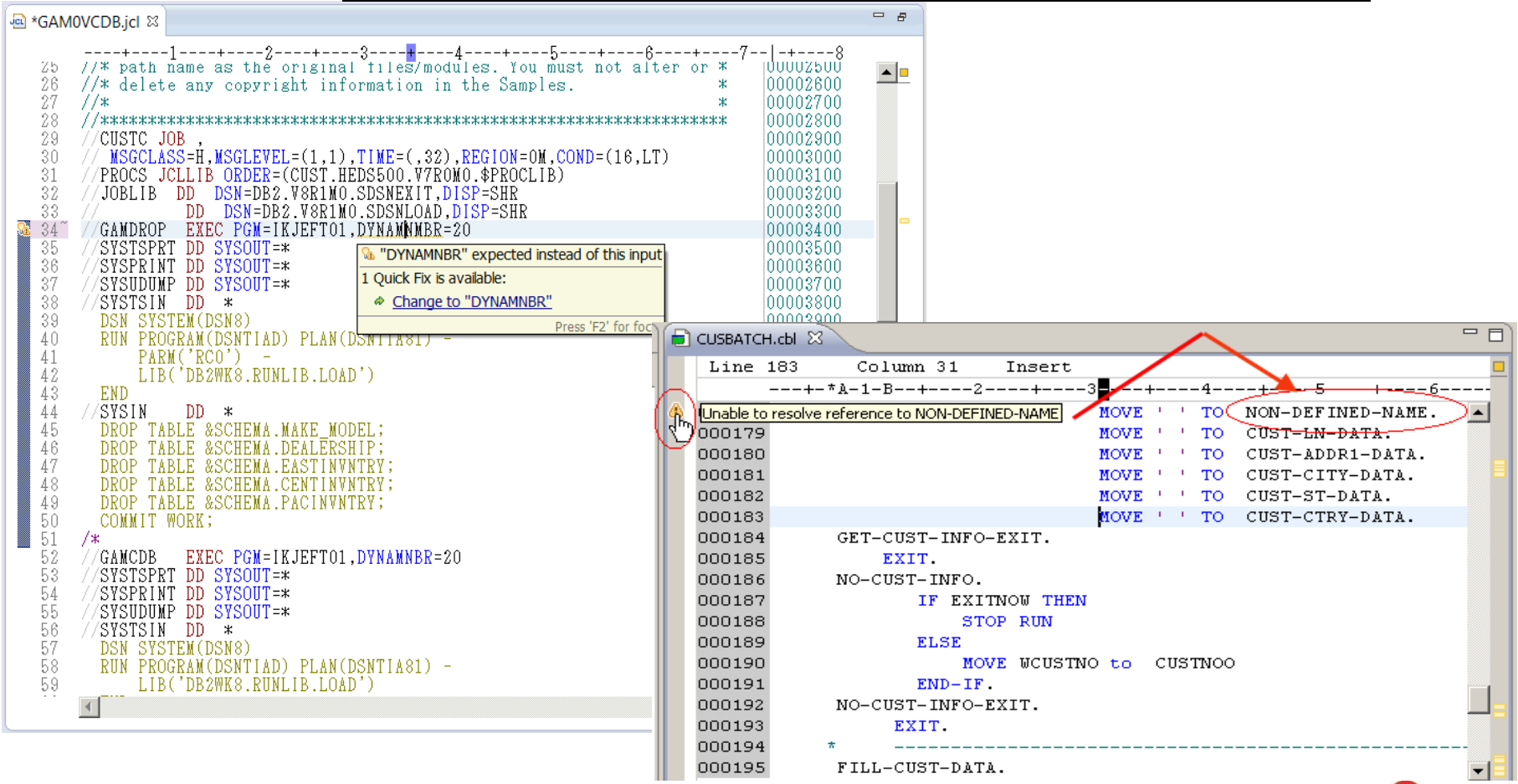
The screenshot shows a COBOL editor window titled 'GAM0PMM.pli'. A project tree is visible on the right side, showing the following structure:

- GAM0PMM : Procedure ( COMMAREAPTR ) optio
  - RecLength
  - SendLength
  - COMMAREAPTR
    - %INCLUDE GAMOIMT;
    - 1 LCOMMAREA
    - 1 WCOMMAREA
      - EXEC SQL INCLUDE SQLCA;
      - GETMAKES : PROCEDURE;
        - TEMPINDEX
        - TEMPMAKE
        - EXITLOOP
      - GETMODELS : PROCEDURE;
        - TEMPINDEX
        - TEMPMODEL
        - EXITLOOP



# Editor Productivity Features – real time syntax checking

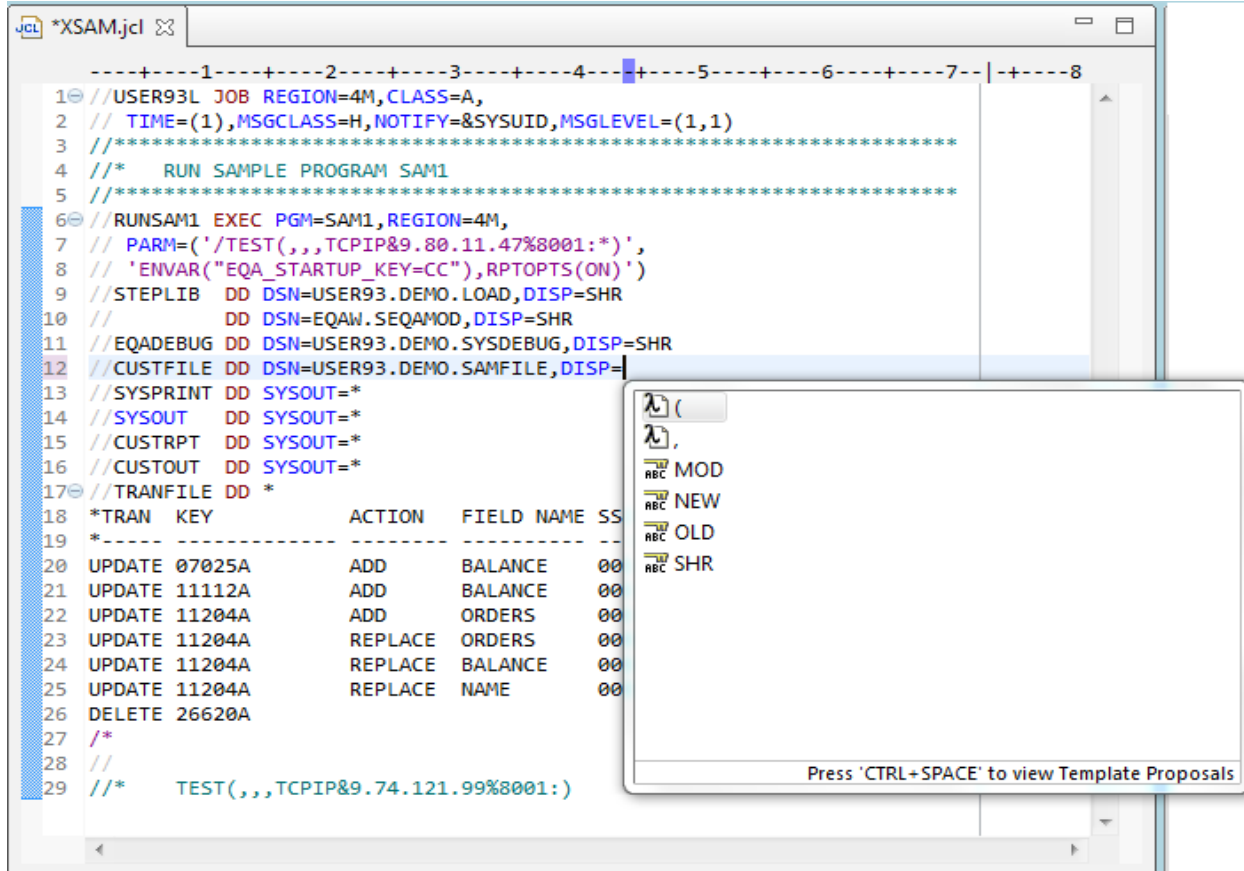
Real-time syntax check without requiring code compile or save



The screenshot displays two windows from an IDE. The main window, titled '\*GAM0VCDB.jcl', shows JCL code. Line 34 is highlighted, and a tooltip indicates an error: '"DYNAMNBR" expected instead of this input'. A 'Quick Fix' is available: 'Change to "DYNAMNBR"'. The secondary window, titled 'CUSBATCH.cbl', shows COBOL code. Line 183, column 31 has an error: 'Unable to resolve reference to NON-DEFINED-NAME'. A red circle highlights this error, and a red arrow points from the tooltip in the first window to the error in the second window.

# JCL Editor Content Assist

- **Keyword syntax proposals**
  - Press Ctrl+SPACE to activate



The screenshot shows a JCL editor window with a list of JCL statements. A tooltip is displayed over the text 'MOD' in the following statement:

```
12 //CUSTFILE DD DSN=USER93.DEMO.SAMFILE,DISP=
```

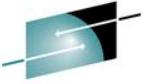
The tooltip lists the following template proposals:

- MOD
- NEW
- OLD
- SHR

At the bottom of the tooltip, it says: "Press 'CTRL+SPACE' to view Template Proposals".

```
1 //USER93L JOB REGION=4M,CLASS=A,
2 // TIME=(1),MSGCLASS=H,NOTIFY=&SYSUID,MSGLEVEL=(1,1)
3 //*****
4 //*   RUN SAMPLE PROGRAM SAM1
5 //*****
6 //RUNSAM1 EXEC PGM=SAM1,REGION=4M,
7 // PARM=(' /TEST(,,TCPIP&9.80.11.47%8001:*)',
8 // 'ENVAR("EQA_STARTUP_KEY=CC"),RPTOPTS(ON)')
9 //STEPLIB DD DSN=USER93.DEMO.LOAD,DISP=SHR
10 // DD DSN=EQAW.SEQAMOD,DISP=SHR
11 //EQADEBUG DD DSN=USER93.DEMO.SYSDEBUG,DISP=SHR
12 //CUSTFILE DD DSN=USER93.DEMO.SAMFILE,DISP=
13 //SYSPRINT DD SYSOUT=*
14 //SYSOUT DD SYSOUT=*
15 //CISTRPT DD SYSOUT=*
16 //CUSTOUT DD SYSOUT=*
17 //TRANFILE DD *
18 *TRAN KEY          ACTION  FIELD NAME SS
19 *-----
20 UPDATE 07025A      ADD     BALANCE  00
21 UPDATE 11112A      ADD     BALANCE  00
22 UPDATE 11204A      ADD     ORDERS   00
23 UPDATE 11204A      REPLACE ORDERS   00
24 UPDATE 11204A      REPLACE BALANCE  00
25 UPDATE 11204A      REPLACE NAME    00
26 DELETE 26620A
27 /*
28 //
29 //*   TEST(,,TCPIP&9.74.121.99%8001:)
```

# JCL Editor improvements

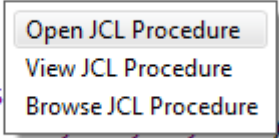


## ■ JCL Outline shows all DDs

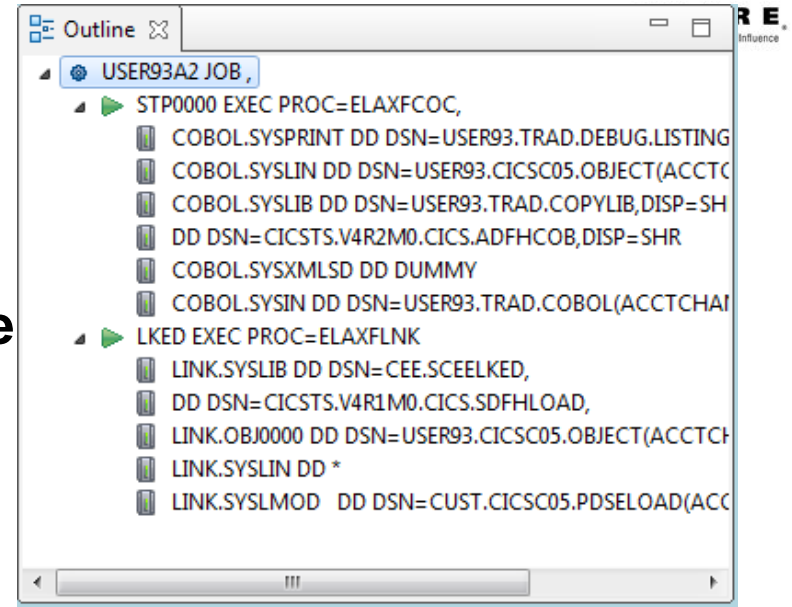
- No longer only instream

## ■ Open actions allow Open, View, Browse

```
5 //STP0000 EXEC PROC=ELAXFCOC,  
6 // CICS=,  
7 // DB2=,  
8 // COMP=,  
9 // PARM.COBOL=('CICS  
10 // 'LIB,TEST,SOURCE,  
11 //COBOL.SYSPRINT DD DSN=USER93.TRAD.DEBUG.LISTING(ACCT  
12 //COBOL.SYSLIN DD DSN=USER93.CICSC05.OBJECT(ACCTC
```

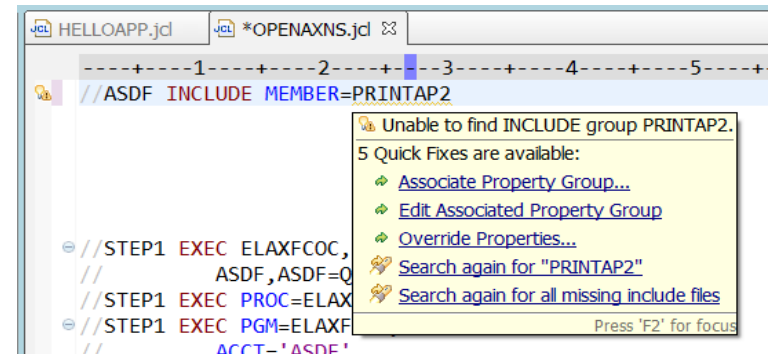


- Open JCL Procedure
- View JCL Procedure
- Browse JCL Procedure



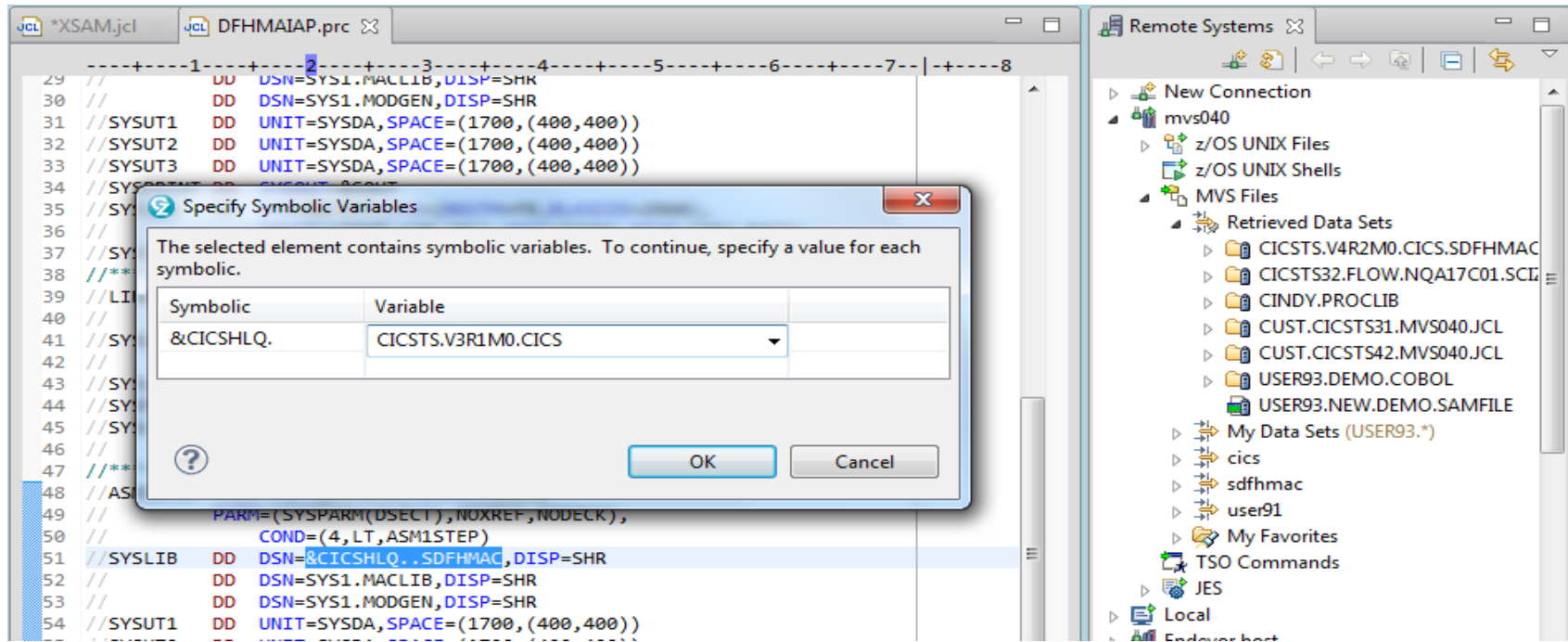
## ■ Improvements for INCLUDE members

- Hover to see contents
- Open actions support
- Syntax check with Quick fixes



# JCL Editor Symbolic resolution

- Open Declaration (F3) on data set with Symbolic variables



The screenshot shows the JCL Editor interface with a dialog box titled "Specify Symbolic Variables". The dialog box contains the following text:

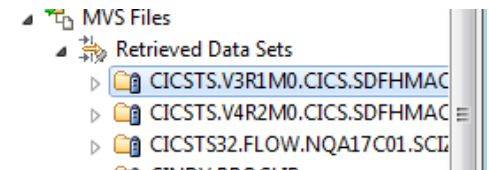
The selected element contains symbolic variables. To continue, specify a value for each symbolic.

Symbolic	Variable
&CICSHLQ.	CICSTS.V3R1M0.CICS

The dialog box has "OK" and "Cancel" buttons. The background shows the JCL editor with the following code:

```
29 // DD DSN=SYS1.MACLIB,DISP=SHR
30 // DD DSN=SYS1.MODGEN,DISP=SHR
31 //SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(400,400))
32 //SYSUT2 DD UNIT=SYSDA,SPACE=(1700,(400,400))
33 //SYSUT3 DD UNIT=SYSDA,SPACE=(1700,(400,400))
34 //SYSUT4 DD UNIT=SYSDA,SPACE=(1700,(400,400))
35 //SY
36 //
37 //SY
38 //**
39 //LI
40 //
41 //SY
42 //
43 //SY
44 //SY
45 //SY
46 //
47 //**
48 //AS
49 // PARM=(SYSPARM(DSECT),NOXREF,NODECK),
50 // COND=(4,LT,ASM1STEP)
51 //SYSLIB DD DSN=&CICSHLQ..SDFHMAC,DISP=SHR
52 // DD DSN=SYS1.MACLIB,DISP=SHR
53 // DD DSN=SYS1.MODGEN,DISP=SHR
54 //SYSUT1 DD UNIT=SYSDA,SPACE=(1700,(400,400))
```

- Data set is Retrieved using specified value

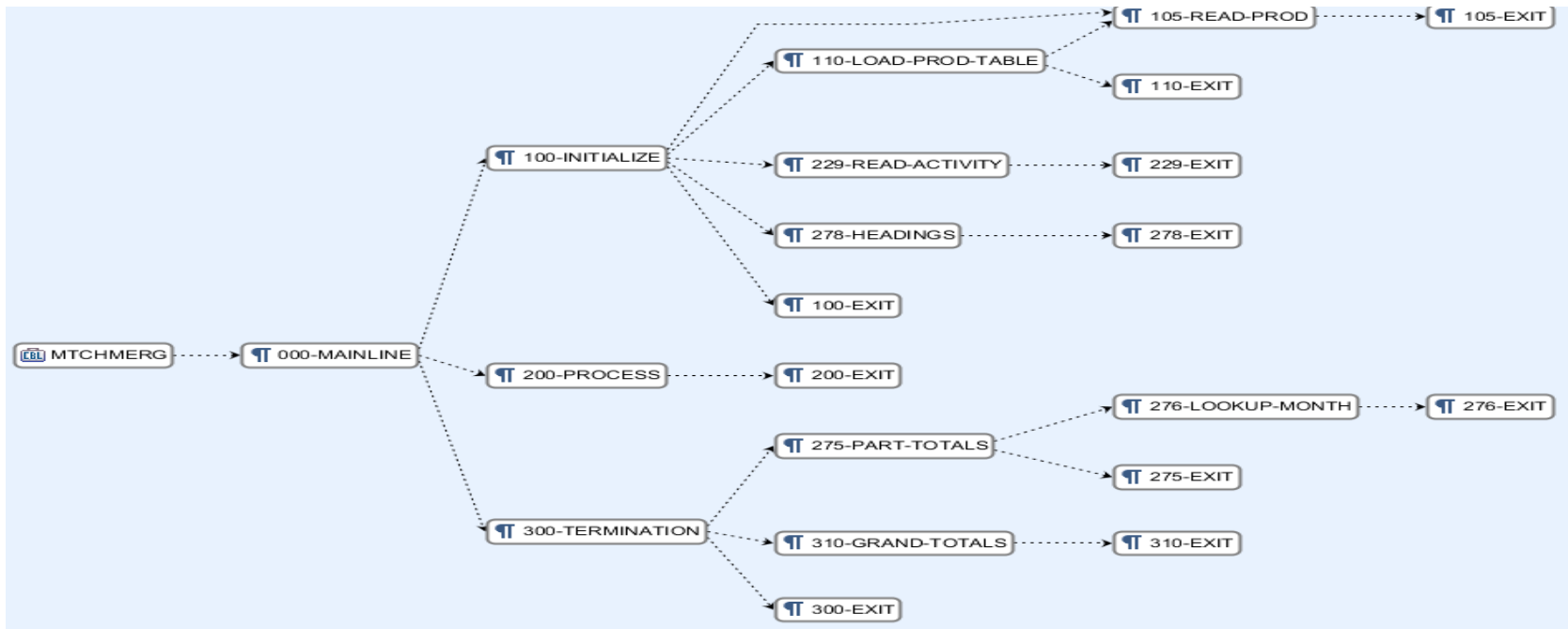


The screenshot shows the "Remote Systems" tree view with the following structure:

- Remote Systems
- Local
- Enduser host
- JES
- TSO Commands
- My Favorites
- user91
- sdfhmac
- cics
- My Data Sets (USER93.\*)
- USER93.DEMO.COBOL
- USER93.NEW.DEMO.SAMFILE
- MVS Files
  - Retrieved Data Sets
    - CICSTS.V3R1M0.CICS.SDFHMAC
    - CICSTS.V4R2M0.CICS.SDFHMAC
    - CICSTS32.FLOW.NQA17C01.SCI
    - CINDY.PROCLIB
    - CUST.CICSTS31.MVS040.JCL
    - CUST.CICSTS42.MVS040.JCL

# Enhanced Application Quality & Structure Analysis

- Application Analysis
  - Control flow diagrams for COBOL and **PLI** programs,
    - **Graphical representation of the program flow with links to the source**
  - Helps identify and highlight potential unreachable code



# Enhanced Structure Analysis – Data Element Table

- A table representation of the user-defined data items and symbols in a program
  - ▶ Hyperlinks in the table are integrated with the editor allowing easy access to the declaration of the data items.
- Generated by showing the “symbol table” generated when RDz real-time syntax check parses the program

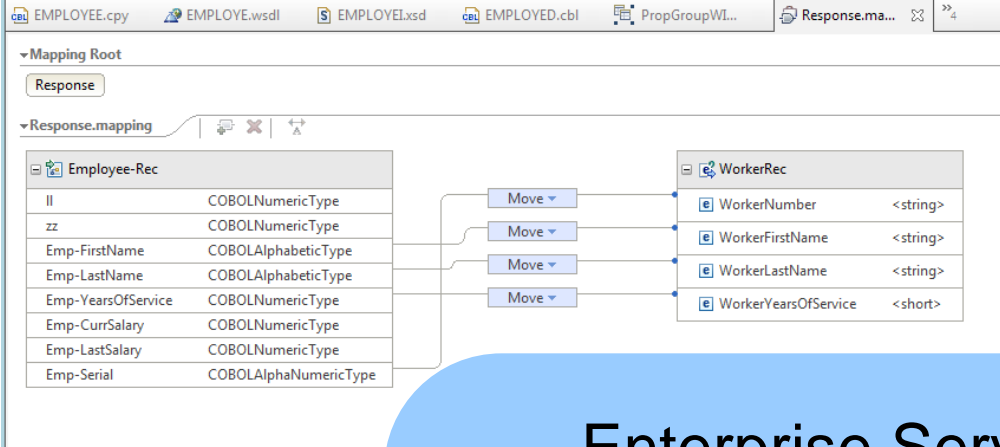
Data Element Table X

Showing data elements from WARDRPT.cbl

Search:

Data Item Name:	Data Type:	Level:	Top-Level Item:	Declaration:	Initial Value:	Line ...	Reference count:	Full Declaration:
PATLISTEST-S-ID	Data	10	PATIENT-MASTER-REC	PIC X(08)		378	0	10 PATLISTEST-S-ID PIC X
PATMSTR	File Descriptor	0	PATMSTR			116	4	FD PATMSTR RECORD CO.
PATMSTR-FOUND	Data	88	FILE-STATUS-CODES			134	1	88 PATMSTR-FOUND VALUE "0
PATMSTR-KEY	Data	5	PATMSTR	PIC X(06)		120	2	05 PATMSTR-KEY PIC X(06).
PATMSTR-REC	Data	1	PATMSTR			119	1	01 PATMSTR-REC.
PATMSTR-STATUS	Data	5	FILE-STATUS-CODES	PIC X(2)		133	3	05 PATMSTR-STATUS PIC X
PATPERSN	File Descriptor	0	PATPERSN			123	4	FD PATPERSN RECORD CO
PATPERSN-FOUND	Data	88	FILE-STATUS-CODES			136	1	88 PATPERSN-FOUND VALUE "
PATPERSN-KEY	Data	5	PATPERSN	PIC X(06)		127	2	05 PATPERSN-KEY PIC X(06).
PATPERSN-REC	Data	1	PATPERSN			126	2	01 PATPERSN-REC.
PATPERSN-STATUS	Data	5	FILE-STATUS-CODES	PIC X(2)		135	3	05 PATPERSN-STATUS PIC
PATSRCH	File Descriptor	0	PATSRCH			98	4	FD PATSRCH RECORDING
PAYMENT-METHOD-TYPE	Data	5	PATIENT-PERSONAL-...	PIC X(02)		313	0	05 PAYMENT-METHOD-TYPE
PEDIATRICS	Data	88	INPATIENT-DAILY-REC			157	0	88 PEDIATRICS VALUE "1010".

# Traditional development and Enterprise web services



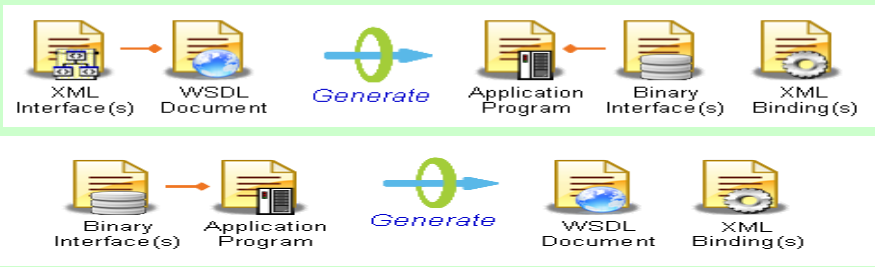
Supports traditional development/maintenance

- Cobol, PL/I, Assembler, JCL

Supports modern architecture development

## Enterprise Service Tools

Top down, bottom up, and meet in the middle web service enablement for CICS, IMS, and Batch/TSO environments.



**Enable Enterprise Applications for Mobile and Web**

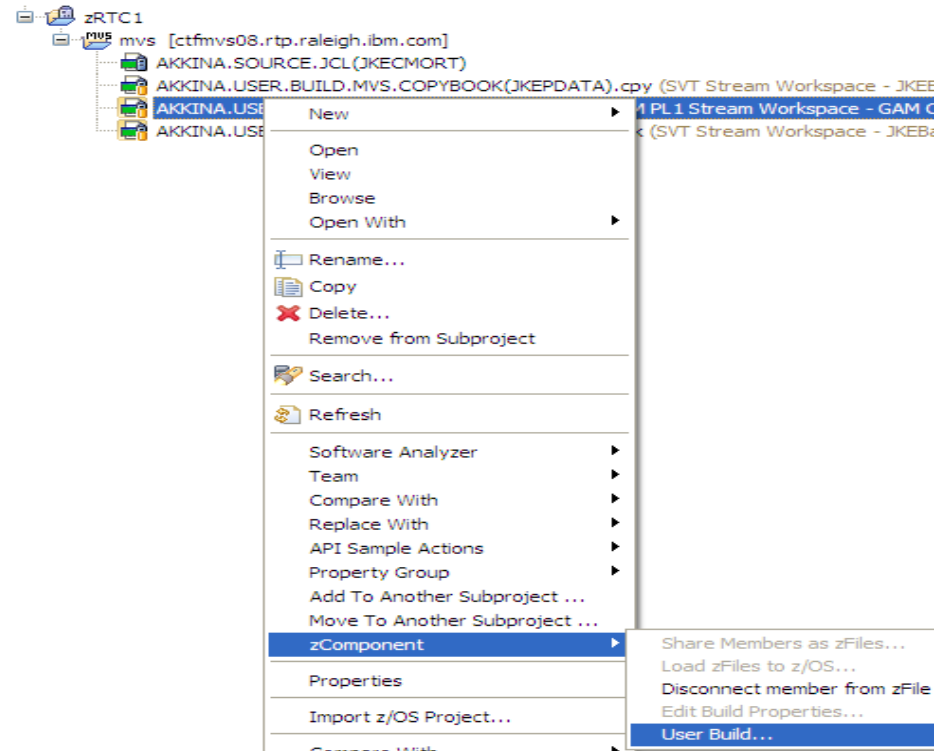
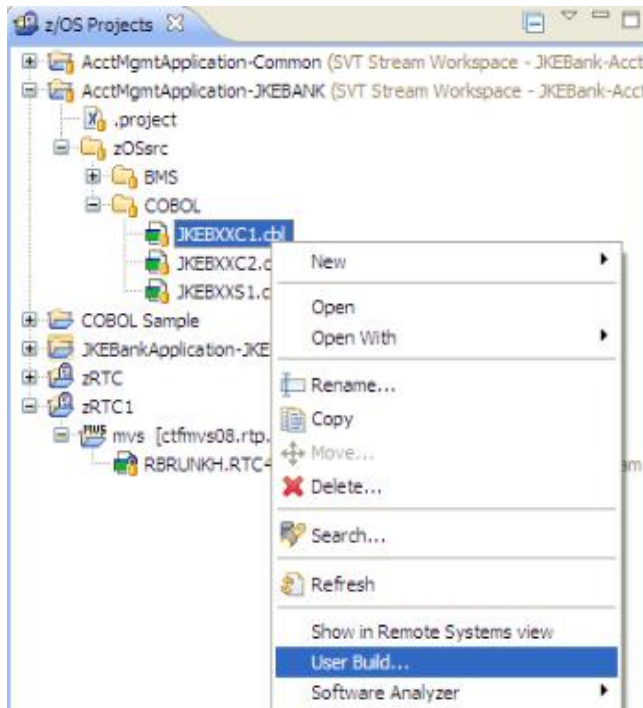
Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



# User Build from RDz

“User build”, is supported both in zComponent projects and RDz remote z/OS projects

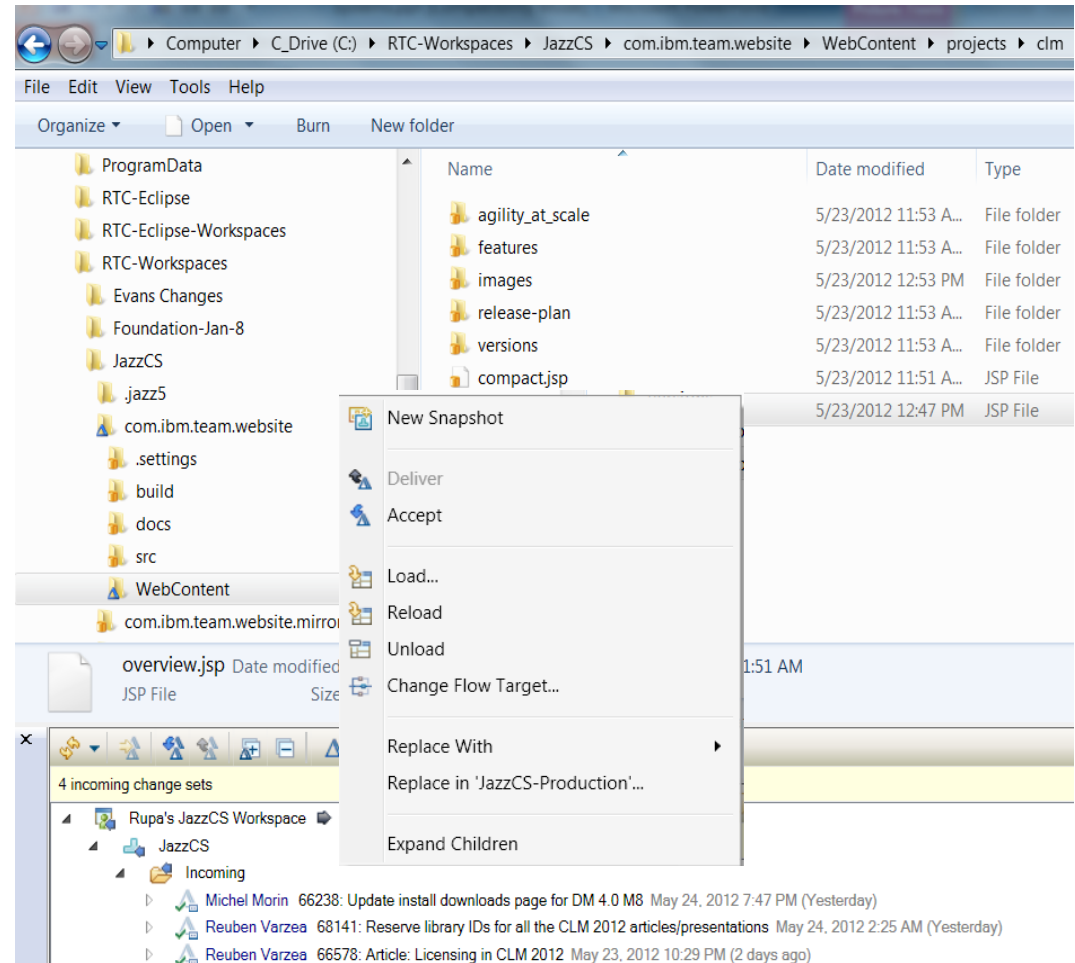
- Builds just one the single file selected, supports Error feedback
- Generates JCL based on the associated RTC Language definitions and Translators





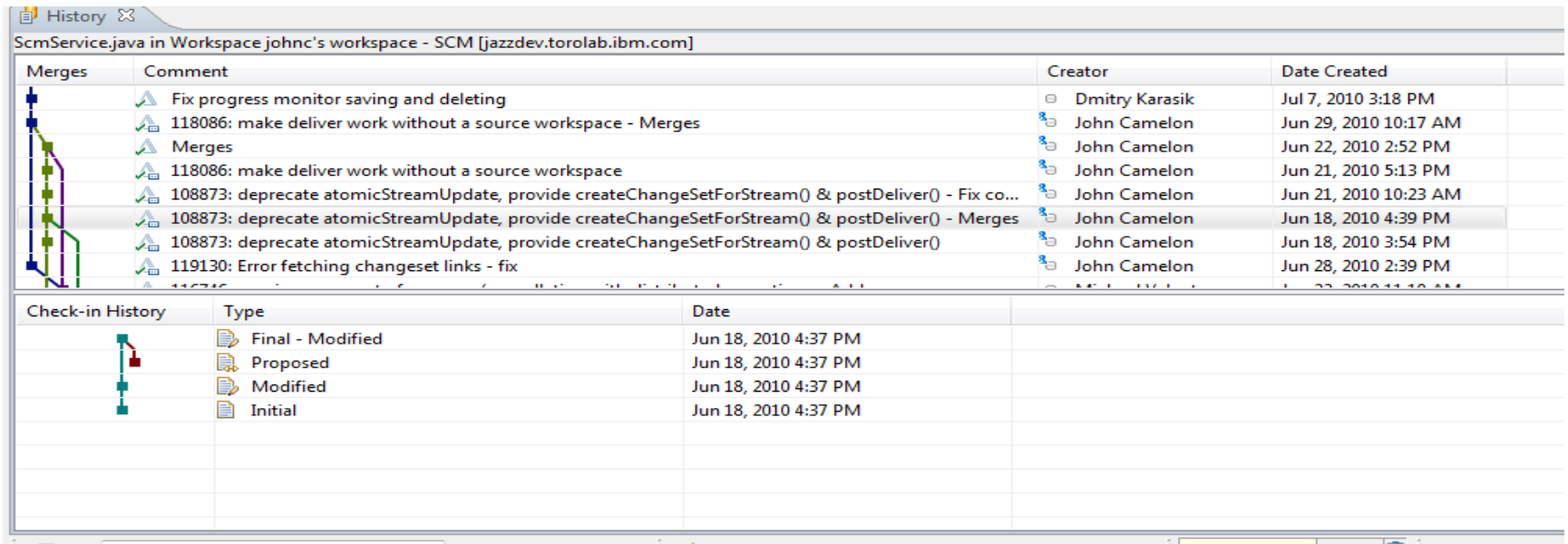
# Pending Changes

- If you want finer grained control on your SCM operations, then the Pending Changes view is for you
  - Check in, deliver, accept changes
  - Suspend, resume, discard changes
  - Replace, reload out-of-sync
  - Resolve conflicts
  - Open change sets and work items via the web client











# Traceability : Check-in History

- Someone made a costly mistake merging and you want to understand exactly where the mistake was made
  - Problem : Traditional history commands & UI only show before/after & merge states for a change set ... it does not show intermediates
  - Solution : Use Check-in history in Eclipse, CLI or .NET clients







The screenshot shows the Eclipse SCM History view for a workspace. It is divided into two main sections: 'Merges' and 'Check-in History'.

**Merges Table:**

Merges	Comment	Creator	Date Created
	Fix progress monitor saving and deleting	Dmitry Karasik	Jul 7, 2010 3:18 PM
	118086: make deliver work without a source workspace - Merges	John Camelon	Jun 29, 2010 10:17 AM
	Merges	John Camelon	Jun 22, 2010 2:52 PM
	118086: make deliver work without a source workspace	John Camelon	Jun 21, 2010 5:13 PM
	108873: deprecate atomicStreamUpdate, provide createChangeSetForStream() & postDeliver() - Fix co...	John Camelon	Jun 21, 2010 10:23 AM
	108873: deprecate atomicStreamUpdate, provide createChangeSetForStream() & postDeliver() - Merges	John Camelon	Jun 18, 2010 4:39 PM
	108873: deprecate atomicStreamUpdate, provide createChangeSetForStream() & postDeliver()	John Camelon	Jun 18, 2010 3:54 PM
	119130: Error fetching changeset links - fix	John Camelon	Jun 28, 2010 2:39 PM

**Check-in History Table:**

Check-in History	Type	Date
	Final - Modified	Jun 18, 2010 4:37 PM
	Proposed	Jun 18, 2010 4:37 PM
	Modified	Jun 18, 2010 4:37 PM
	Initial	Jun 18, 2010 4:37 PM

# Development Life Cycle

Planning	Source Dev	Governance/Unit test	Build
<ul style="list-style-type: none"> <li>• Define the tasks</li> <li>• Create a plan</li> <li>• Create a work item</li> <li>• Assign the work item to a developer</li> </ul>	<ul style="list-style-type: none"> <li>• Load the project/source artifacts from SCM</li> <li>• Navigate, Analyze, Edit, Syntax check source code</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Compile</b></li> <li>• <b>Quality assurance</b> <ul style="list-style-type: none"> <li>• <b>Debug</b></li> <li>• <b>Code Coverage</b></li> <li>• <b>Code review</b></li> <li>• <b>Unit Testing</b></li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Check-in/Deliver the source code</li> <li>• Build</li> </ul>
CLM	RDz RTC	RDz RD&T RTC	RTC RDz

# Integrated Debugger

- **RDz introduced a new feature called Integrated Debugger**
  - ✓ **A GUI-based multi-platform, multi-language debugger**
    - ✓ Full asynchronous mode
      - ✓ Thread-level control of multi-threaded applications
    - ✓ Automonitor support
  - ✓ **RDz v9.0.1 Supported:**
    - ✓ COBOL V5.1, V4, V3.4
    - ✓ Batch, Batch IMS, Batch DB2, CICS 5.1, 4.2, 4.1
    - ✓ Interactive Code coverage – Out of the box
  - ✓ **RDz 9.1 added support for:**
    - ✓ PLI v4.x, v3.9
    - ✓ C/C++ V1R13, V2R1
    - ✓ IMS TM
    - ✓ DB2 Stored procedures

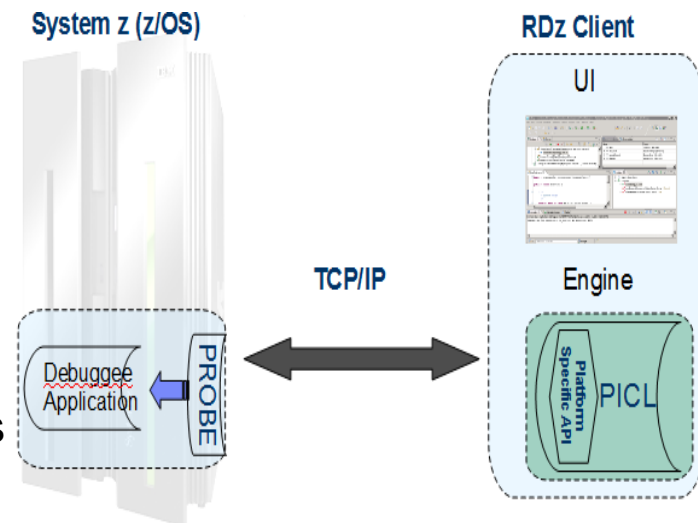
# Integrated Debugger

## ✓ Host-offload architecture:

- ✓ Remote debugger with only a small footprint on the mainframe:
  - Leverages workstation CPUs enabling faster processing of debug information
  - Enables scalability and reliability
- ✓ Debugger client is supported on Windows and Linux

## ✓ Simple and Secure Connections:

- ✓ Single client can handle multiple debug sessions on multiple hosts or an application the spans multiple systems
- ✓ Client initiated debug – no need to specify client IP address and port (v9.0.1.2)
- ✓ SSL/TLS support





# Enhanced Application Quality – Code Coverage



- Line Level Code Coverage - provides tools to measure and report on test coverage of an application
  - Leverages the Integrated Debugger technology
  - Indicating what source code lines were tested and remain to be tested

**Code Coverage Report**

Code coverage report, generated Oct 18, 2011 12:19:59 PM

Element	Coverage	Covered Lines	Total Lines
SAM1	75%	117	156
SAM1	75%	117	156
VENKATU.COBOL.SYSDEBUG(SAM1).cob	75%	117	156
SAM1()	75%	117	156

Launch Name	Launch Date
SAM1-2011_10_18_121959	Oct 18, 2011 12:19:59 PM
SAM1-2011_10_17_163634	Oct 17, 2011 4:36:34 PM
SAM1-2011_10_17_160130	Oct 17, 2011 4:01:30 PM
SAM1-2011_10_14_132043	Oct 14, 2011 1:20:43 PM
SAM1-2011_10_14_130134	Oct 14, 2011 1:01:34 PM
SAM1-2011_10_14_124502	Oct 14, 2011 12:45:02 PM

```

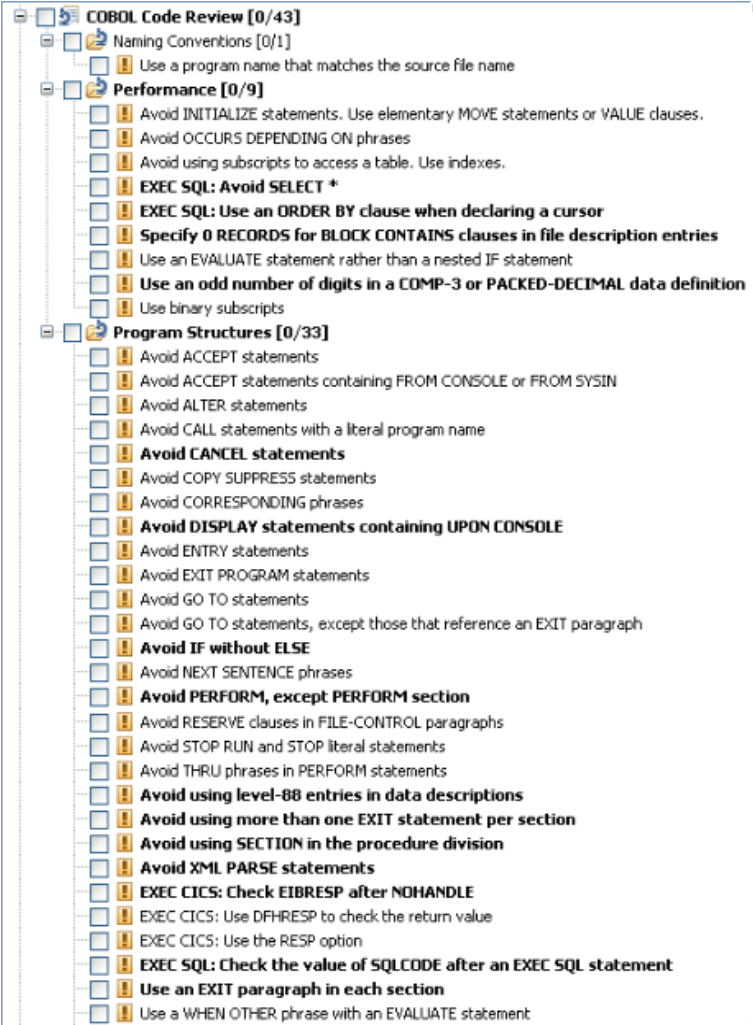
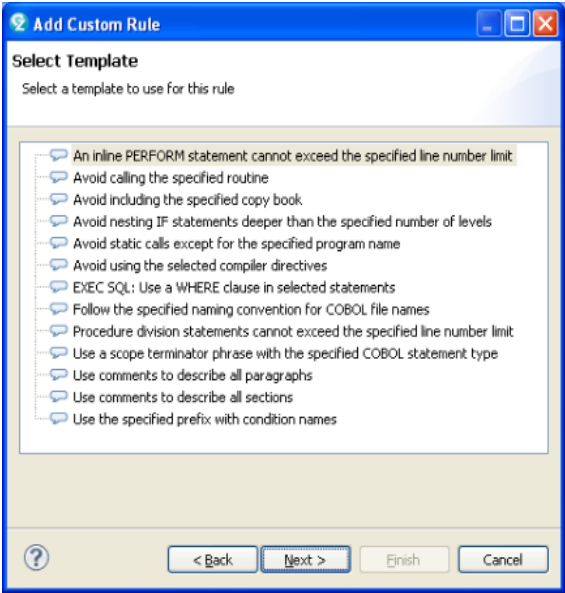
374 -----*A-1-B-----2-----3-----4-----5-----6-----7-----
375                                PERFORM 210-PROCESS-ADD-TRAN
376                                WHEN 'DELETE'
377                                PERFORM 220-PROCESS-DELETE-TRAN
378                                WHEN OTHER
379                                IF TRAN-COMMENT NOT = '*'
380                                MOVE 'INVALID TRAN CODE:' TO ERR-MSG-DATA1
381                                MOVE TRAN-CODE TO ERR-MSG-DATA2
382                                PERFORM 299-REPORT-BAD-TRAN
383                                END-IF
384                                END-EVALUATE
385                                END-IF
386                                MOVE TRAN-KEY TO WS-PREV-TRAN-KEY
387                                IF WS-TRAN-OK = 'Y'
388                                PERFORM 830-REPORT-TRAN-PROCESSED
389                                END-IF
390                                END-IF .
391
392 200-PROCESS-UPDATE-TRAN.
393  ADD +1 TO NUM-UPDATE-REQUESTS.
394  PERFORM 720-POSITION-CUST-FILE.
395  IF CUST-KEY NOT = TRAN-KEY OR WS-CUST-FILE-EOF = 'Y'
396  MOVE 'NO MATCHING KEY:' TO ERR-MSG-DATA1
397  MOVE TRAN-KEY TO ERR-MSG-DATA2
398  PERFORM 299-REPORT-BAD-TRAN
399  ELSE
400  *
401  *   Subroutine SAM2 will apply an update to a customer record
402  *
403  CALL 'SAM2' USING CUST-REC, TRANSACTION-RECORD,
404                  WS-TRAN-OK, WS-TRAN-MSG
405  IF WS-TRAN-OK NOT = 'Y'
406  MOVE WS-TRAN-MSG TO ERR-MSG-DATA1
407  MOVE SPACES TO ERR-MSG-DATA2
408  PERFORM 299-REPORT-BAD-TRAN
    
```

Complete your session evaluations online at [www.SHARE.org/Seattle-Eval](http://www.SHARE.org/Seattle-Eval)



# Enhanced Quality & Structure Analysis – Code review

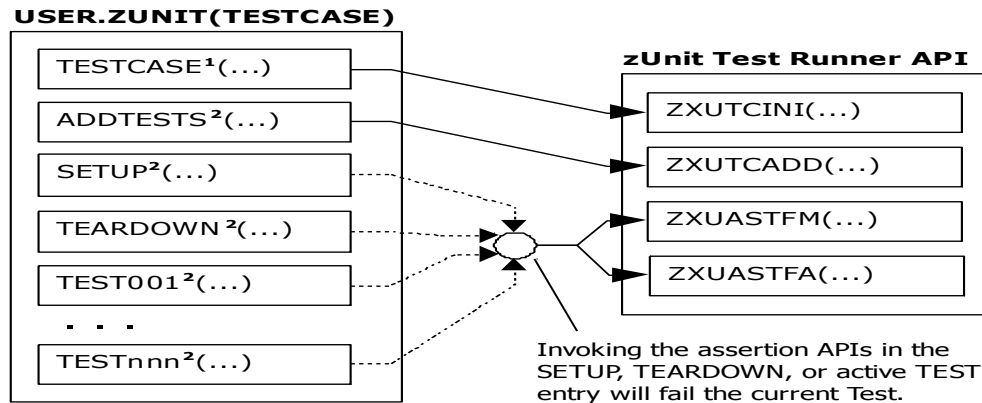
- Code Review/Governance - provides predefined rules and templates for COBOL and PL/I applications
  - Ensure adherence to corporate standards
  - Custom rules for COBOL and PL/I





# zUnit – Unit testing framework for z/OS

- Frameworks that assist developers in writing code to perform repeatable, self-checking unit tests are collectively known as xUnit.
- [xUnit](#) defines a set of concepts that together provide a light-weight architecture for implementing unit testing frameworks.
  - JUnit, for example, is a very popular instance of the xUnit architecture.
- **zUnit** is a xUnit instance for System z
- Goal is to encourage the **continuous integration and continuous testing** methodology for System z Application development and maintenance



<sup>1</sup>Language-specific details:

- In COBOL, this is the first program appearing in the Test Case source file and it will be invoked by the Test Runner for Test Case initialization.
- In PL/I, this is the procedure declared with option(fetchable) in the Test Case source file and it will be invoked by the Test Runner for Test Case initialization.

<sup>2</sup>Language-specific details:

- In COBOL, these are expected to be subprograms (non-nested and therefore compatible with FUNCTION-POINTER).
- In PL/I, these are expected to be internal procedures that are declared at the package level (non-nested).

# zUnit Capabilities

- zUnit Test Runner
  - Runs on z/OS
    - Installed and configured on z/OS as part of RDz Host install and customization
  - Fetches and runs the Test Suite referred to in a zUnit configuration file
  
- zUnit Wizard used to generate Test Cases
  - RDz client feature
  - Eclipse based wizards allow creation of:
    - Template Test Cases are generated in COBOL or PL/I
      - Simple pass/fail **assertion** API

## (RDz v9.1) Complete COBOL test cases:

- Identify the interface or set of copy book(s)
  - Generate XML Schema to represent the interface
  - Generate XML files where you would specify test input and expected output
  - Generate a Test Case based on the XML file
  - (Optionally) Generate stubs for called programs
- RDz viewers/editors for unit test XML results



# Development Life Cycle

## Planning

## Source Dev

## Governance/Unit test

## Build

- Define the tasks
- Create a plan
- Create a work item
- Assign the work item to a developer

- Load the project/source artifacts from SCM
- Navigate, Analyze, Edit, Syntax check source code

- Compile
- Quality assurance
  - Debug
  - Code Coverage
  - Code review
  - Unit Testing

- Check-in/Deliver the source code
- Build

CLM

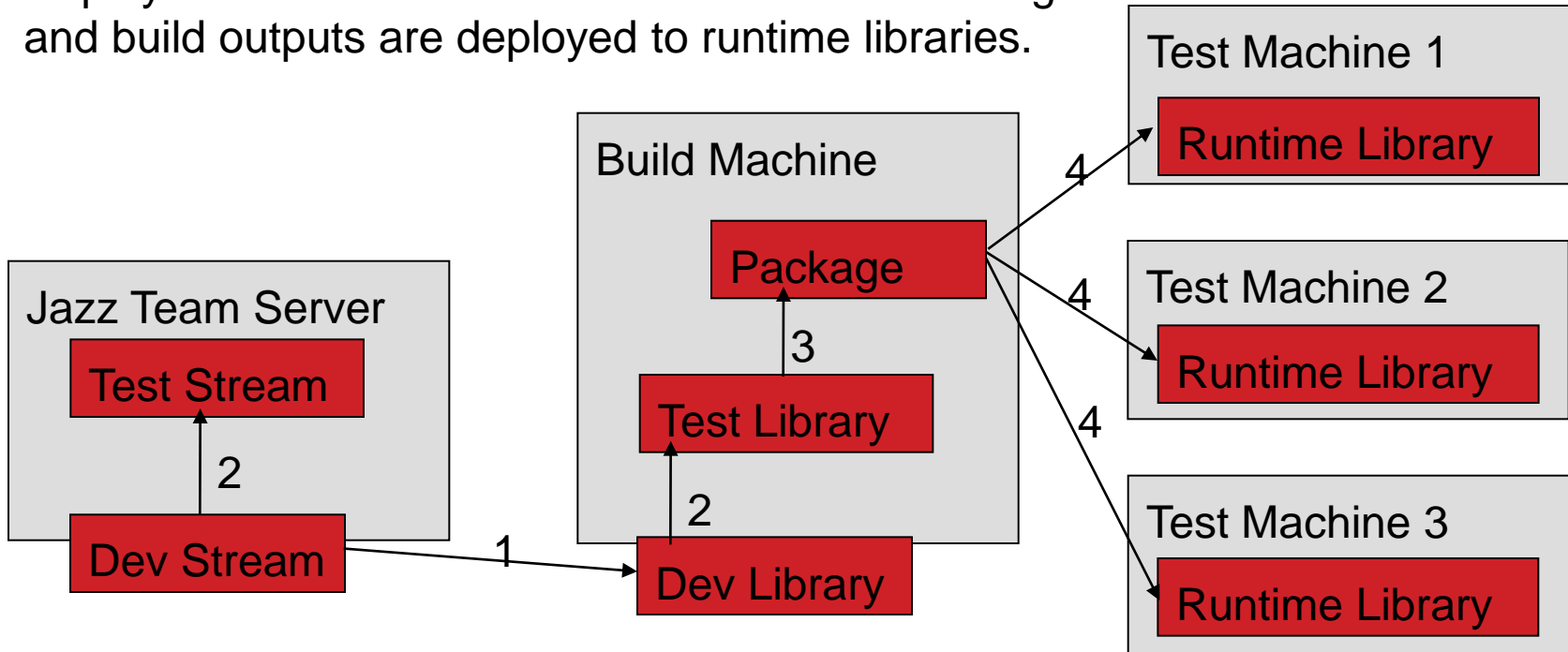
RDz  
RTC

RDz  
RD&T  
RTC

RTC  
RDz

# The big picture

1. Dependency build runs on build machine. Source is loaded from Dev Stream and outputs are built in Dev Library.
2. Promotion build runs on build machine. Source is promoted from Dev Stream to Test Stream and build outputs are copied from Dev Library to Test Library.
3. Package build runs on build machine. Test Library build outputs are archived in a package.
4. Deploy build runs on various test machines. Package is loaded to test machine and build outputs are deployed to runtime libraries.



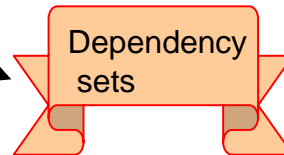
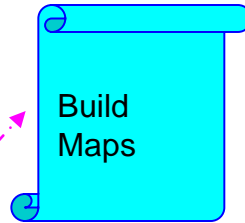
# Dependency Build Summary

Server

**1- Scan**  
Scan new or changed files  
Extract their logical information and dependencies

**2- Preprocessing**  
For changed files:  
impact on build maps

Calculate dependency sets



Mainframe (Build machine)


Build processing

# Snapshots for every build

Build MVS Dependency Build Test 20120619-0908330825

**✓ Completed**

Duration: 53 seconds  
 Start Time: June 19, 2012 9:08:33 AM  
 Completed: June 19, 2012 9:09:27 AM

Status Trend: 

**Reported Work Items**  
 Work items reported against this build to help stabilize it.

None reported against this build

[Create a new work item](#)  
[Associate an existing work item](#)

**Contribution Summary**

Changes: [Show changes](#)  
 Downloads: [7 downloads](#)  
 Logs: [1 log](#)  
 Snapshot: [MVS Dependency Build Test 20120619-0908330825](#)  
 Work items: [3 included in build](#)

**General Information**

Requested by: ADMIN  
 Build Definition: [MVS Dependency Build Test](#)  
 Build Engine: [Setup engine 15560](#)  
 Build History: [19 builds](#)  
 Tags:

**Associated Release**  
 Released builds are available as ch  
[Create a release to associate v](#)

**Snapshot**

Name: \*

[Save](#)

**Details**

Created by:  [ADMIN](#)  
 Created on: Jun 19, 2012 9:08 AM  
 Modified on: Jun 19, 2012 9:08 AM  
 Description:

**Links**

[Create a new repository workspace](#)  
[Create a new stream](#)  
[Compare with snapshot](#)  
[Compare with repository workspace or](#)

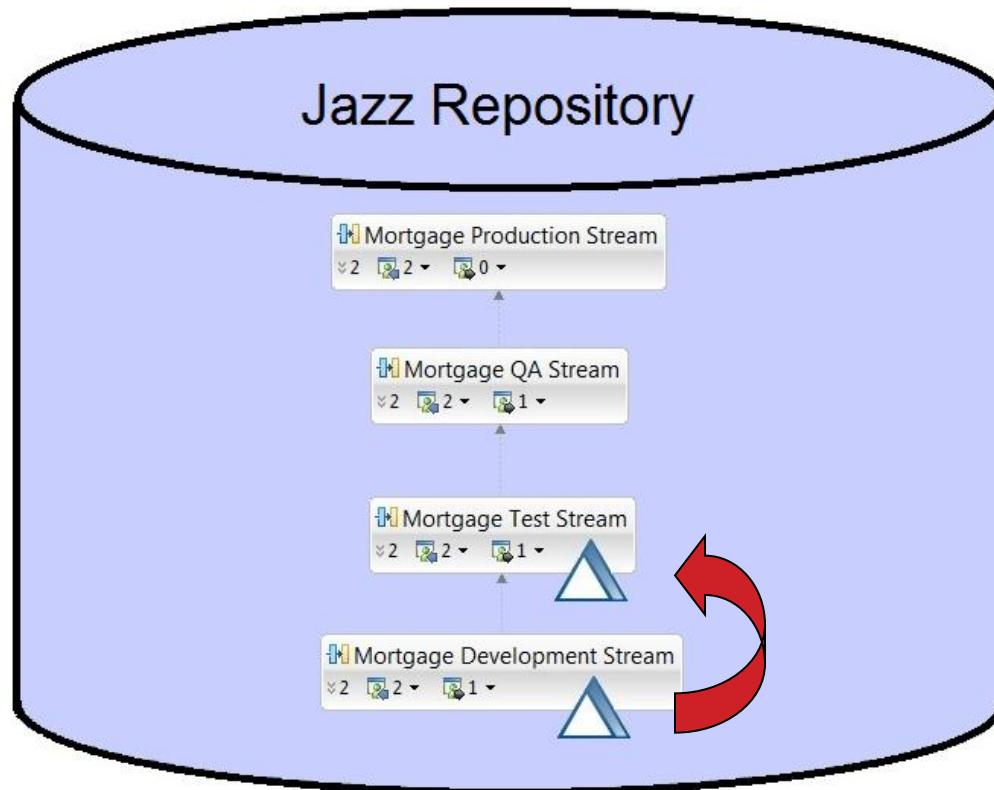
**Components**  
 Shows the components in this snapshot.

<a href="#">Liam Test RWS (4: MVS Dependency Build Dev_20111108-0955580222)</a>	<a href="#">Show Repository Files</a>
<a href="#">Mortgage Component (69: MVS Dependency Build Dev KA_20120619-0853550520)</a>	
<a href="#">plx test (2: MVS Dependency Build Dev_20110328-0857420358)</a>	

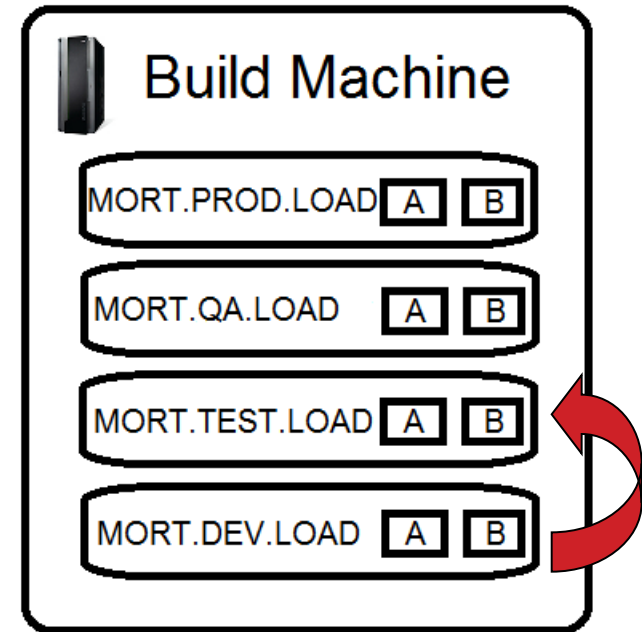
Complete your session

# Promotion

- Flow source code changes and build outputs through the development hierarchy



Source



Outputs