Modern environment for z/OS development

Rosalind Radcliffe

Venkat Balabhadrapatruni

Feb 4, 2013

Session: 12503

Four key barriers preventing optimal return on IT investments

Decades of application investments

"We don't understand the effort, risk and impact of modernizing our legacy applications."

Islands of skills, languages and platforms

"Our skills gap keeps growing. How do we stay current with all the language and technology changes?"

Poorly integrated teams

"We need to enable our teams to collaborate across platforms, languages, and environments."

Infrastructure inefficiency

"We need a cost effective way to improve our infrastructure efficiency and free up capacity to handle more workload."

SHARE • in San Francisco
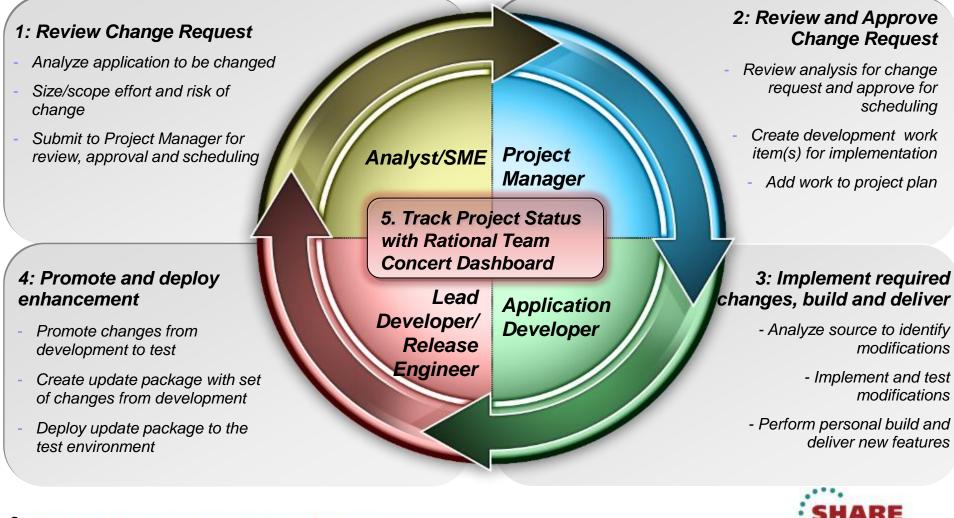2013

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*

Analyst/SME

Project Manager

5. Track Project Status with Rational Team Concert Dashboard

Lead Developer/ Release Engineer

Application Developer

Complete your sessions evaluation online at SHARE.org/SFEval

SHARE in San Francisco
2013

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

Unify

Project/Planning	**Requirements**	**Development**	**Testing**

z/OS AIX i for Business Linux

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

SODIFRANCE

Rational Team Concert – A single tool, many capabilities

- **Work Items**

- **Planning**

- **Source Control**

- **Builds – Continuous**

- **Dashboards & Reporting**

- **Method Enforcement and Automation**
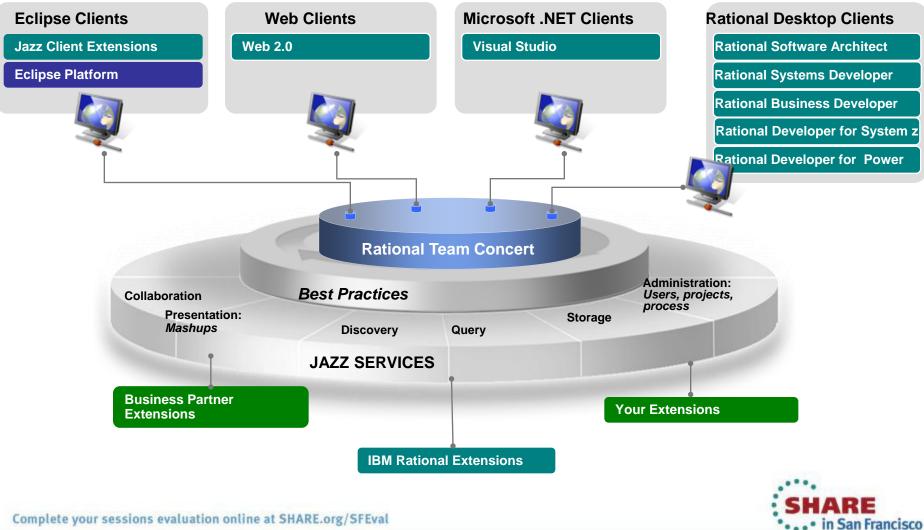
Rational Team Concert: Built on an open, Web 2.0 platform
Supporting a broad range of desktop clients, IDE's and languages

Eclipse Clients

Jazz Client Extensions

Eclipse Platform

Web Clients

Web 2.0

Microsoft .NET Clients

Visual Studio

Rational Desktop Clients

Rational Software Architect

Rational Systems Developer

Rational Business Developer

Rational Developer for System z

Rational Developer for Power

Rational Team Concert

Collaboration

Presentation: *Mashups*

Best Practices

Discovery

Query

Storage

Administration: *Users, projects, process*

JAZZ SERVICES

Business Partner Extensions

IBM Rational Extensions

Your Extensions

SHARE
• in San Francisco
2013

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

Integration with Debug Tool for Development and Test

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 Fault Analyzer for Dump Analysis

Integration with File Manager for file and test data handling

Integration with RD&T for flexible access to System z environment
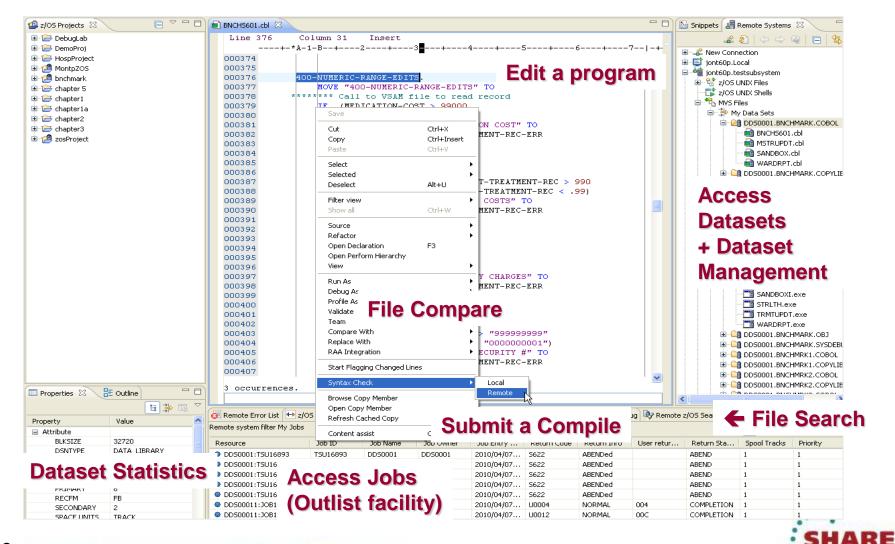
8

The Benefits of a RDz's IDE

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

Access Datasets + Dataset Management

File Compare

File Search

Submit a Compile

Dataset Statistics

Access Jobs (Outlist facility)

Complete your sessions evaluation online at SHARE.org/SFEval

Traditional development and Enterprise web services

```
Line 49        Column 1        Insert
■---+-*A-1-B--+----2----+----3----+-
*
000027     **************************
000028     *
000029     IDENTIFICATION DIVISION.
000030     PROGRAM-ID. GAMOVDB.
000031
000032     DATA DIVISION.
000033
000034     WORKING-STORAGE SECTION.
000035
000036     COPY GAMOBMD.
000037     COPY GAMOBDD.
000038     COPY GAMOBED.
000039     COPY GAMOBCD.
000040     COPY GAMOBPD.
000041     01 D
000042
000043
000044
000045
000046
000047
000048
12
000050
000051
000052
```

Supports traditional development/maintenance
- Cobol, PL/I, Assembler, JCL

Supports modern architecture development

Enterprise Service Tools

Single service projects

Top down, bottom up, and meet in the middle web service enablement for CICS, IMS, and Batch/TSO environments.
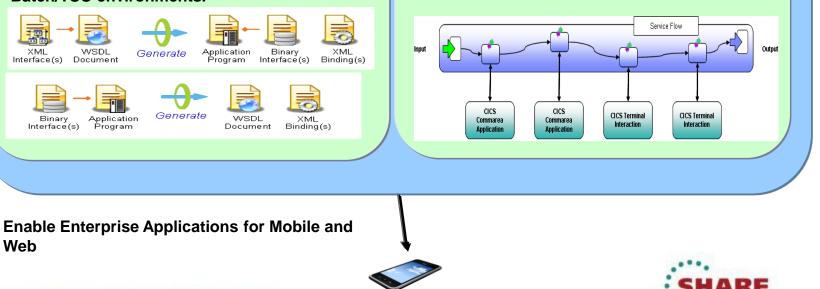
XML Interface(s) → WSDL Document → *Generate* → Application Program → Binary Interface(s) → XML Binding(s)

Binary Interface(s) → Application Program → *Generate* → WSDL Document → XML Binding(s)
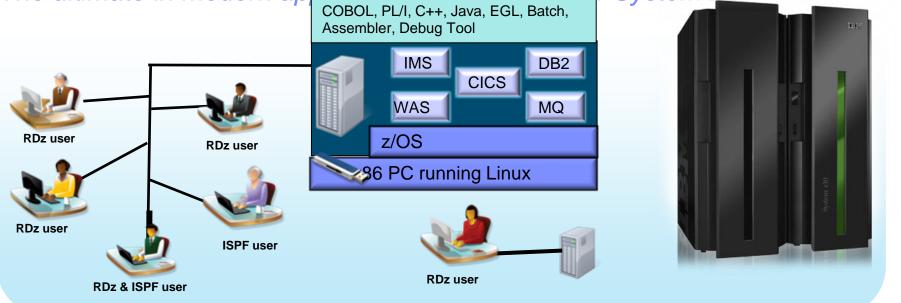
Service flow projects

Graphical composition of CICS applications chained together to form a new business service.

Service Flow

Input → → → → → Output

CICS Commarea Application | CICS Commarea Application | CICS Terminal Interaction | CICS Terminal Interaction

Enable Enterprise Applications for Mobile and Web

Rational Development and Test Environment for System z

The ultimate in modern application development for System z

COBOL, PL/I, C++, Java, EGL, Batch, Assembler, Debug Tool

IMS · DB2 · CICS · WAS · MQ

z/OS

86 PC running Linux

RDz user

RDz user

RDz user

ISPF user

RDz & ISPF user

RDz user

- 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.

SHARE
• in San Francisco
2013

Centralized administration
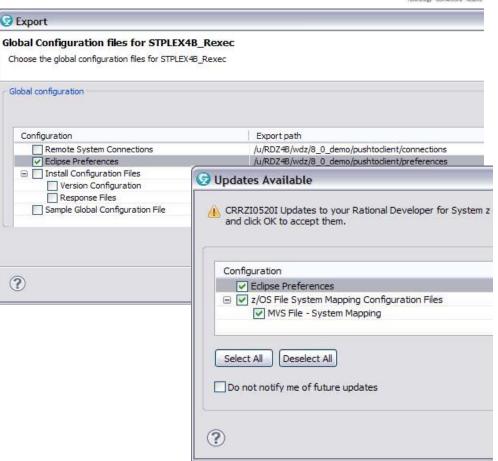
Push to Client - Configuration Files management and product updates

Desktop configuration setup

- upon connecting to the host, a user is notified if they have incoming changes to their configuration files.

Product updates

- triggered from central place to all the users connecting to the host

An Export Wizard – to help an Admin configure various settings on RDz client and upload it to a central location.

Settings are grouped by function into configuration files such as remote connections or system mappings.

Complete your sessions evaluation online at SHARE.org/SFEval

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

SHARE in San Francisco 2013

Any process: Executable and repeatable
Use ONE tool to support both agile and non-agile

Process Architect

Project Manager

Variant #1

Scrum

Variant #2

Iterative

Variant #3

Waterfall

Project A

Agile

Product Owner   **Scrum Master**   **Team Member**

Project B

Iterative

Analyst   **Developer**   **Quality Professional**   **Release Engineer**

Project C

Waterfall

Analyst   **Developer**   **Quality Professional**   **Release Engineer**

Work Items ⌄   Plans ⌄   Source Control ⌄   Build

⭐ Welcome to Work Items

Queries
🔍 My Queries
🔍 Shared Queries

Create Query
🔍 New Query

Create Work Item
✱ Defect
📋 Task
⭐ Story
↻ Epic
🔧 Track Build Item
⊠ Impediment
📘 Adoption Item
📋 Retrospective

Create Work Item Set
📋 Create from Template

Scrum Template

Work Items ⌄   Plans ⌄   Source Control ⌄   Builds

Welcome to Work Items

Queries
🔍 My Queries
🔍 Shared Queries

Create Query
🔍 Create Query

Create Work Item
✱ Defect
📋 Task
Ⓐ Project Change Request
📋 Issue
↻ Business Need
❗ Risk
❗ Risk Action
◆ Milestone

Create Work Item Set
📋 Create From Template...

Formal Project Mgt Template

Multiple plan views facilitate continuous planning

Complete your sessions evaluation online at SHARE.org/SFEval

Progress Tracking - Everyone can see live project status
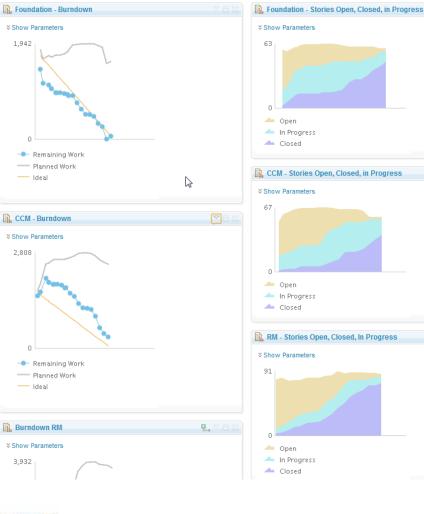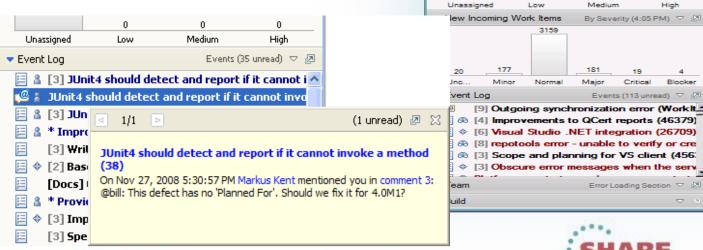
Complete your sessions evaluation online at SHARE.org/SFEval

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
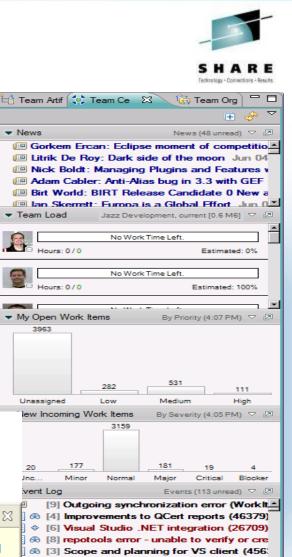
Complete your sessions evaluation online at SHARE.org/SFEval

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
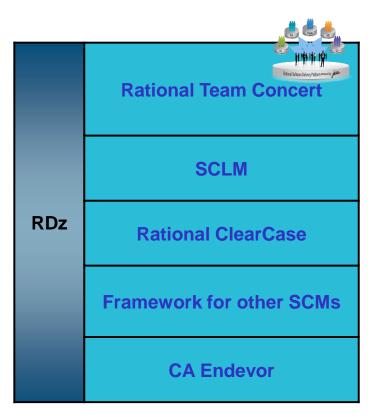
CLM	RDz RTC	RDz RD&T RTC	RTC RDz

RDz Source Code Integration
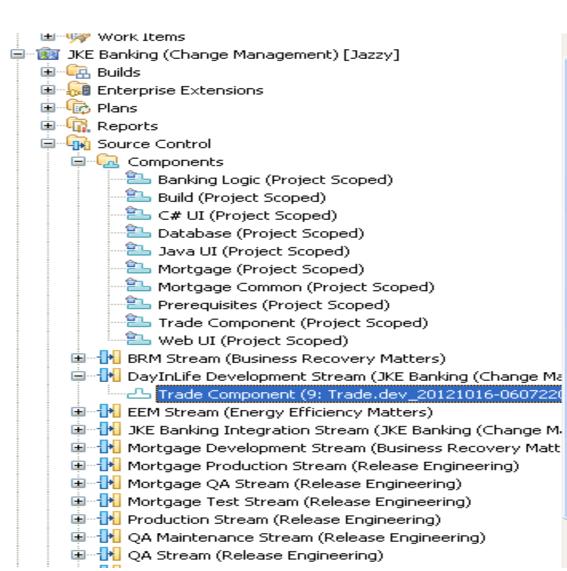
- Rational's Strategic Source Code tooling is RTC and RDz provides tight integration

- 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

RDz	Rational Team Concert
	SCLM
	Rational ClearCase
	Framework for other SCMs
	CA Endevor
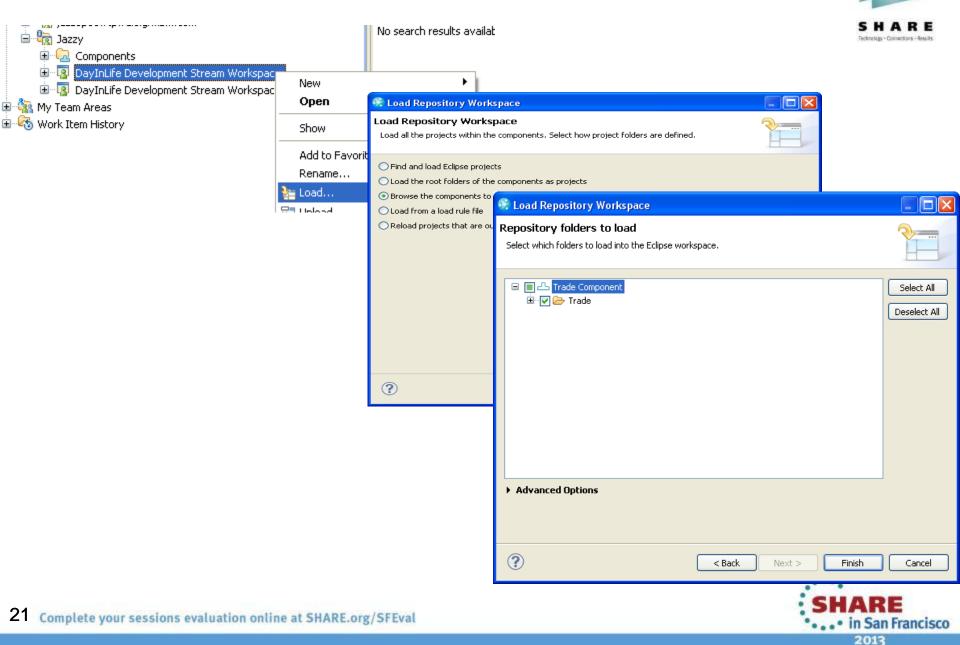
Source Control Management

Complete your sessions evaluation online at SHARE.org/SFEval
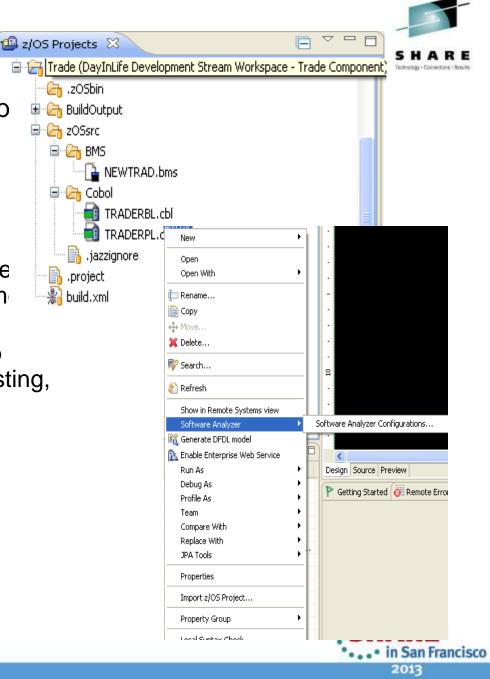
Load the source artifacts

Complete your sessions evaluation online at SHARE.org/SFEval

RDz and RTC together

- Once the project is loaded, it will app[ear] in the RDz z/OS projects view

- RDz augments the development productivity & experience
  - Appropriate editors (COBOL, maps, e[tc.]) and functions (content assist, real tim[e] 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
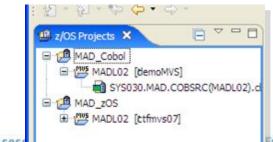
Complete your sessions evaluation online at SHARE.org/SFEval

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

Complete your sessions evaluation online at SHARE.org/SFEval

Edit capabilities in RDz
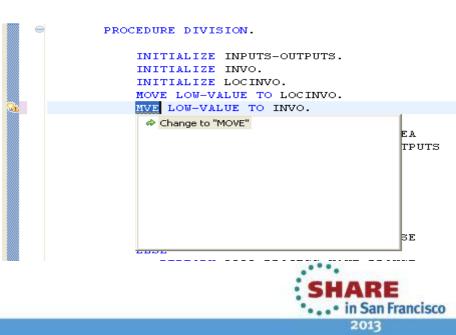
- RDz at a high level has different types of editors

  - 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 and PLI 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.

SHARE in San Francisco
2013

Edit capabilities in RDz

- RDz at a high level has different types of editors
  - WYSYWIG editors
    - Creation, edit of BMS and MFS maps
    - Has the source and design view – allows drag and drop of fields in the design view which generates the appropriate source

  - Data Editors
    - Edit QSAM data

Complete your sessions evaluation online at SHARE.org/SFEval
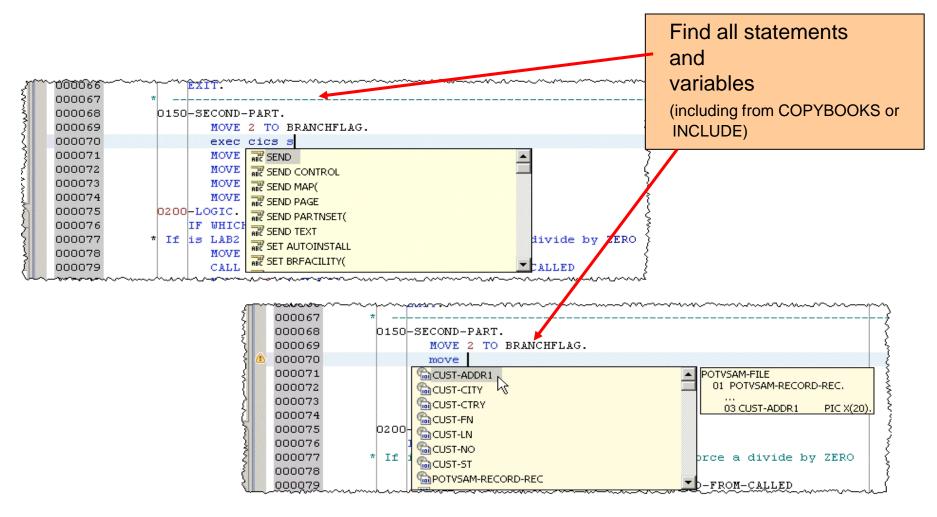
Editor Productivity features - Develop code more efficiently

Find all statements and variables
(including from COPYBOOKS or INCLUDE)

Complete your sessions evaluation online at SHARE.org/SFEval

Editor Productivity Features – real time syntax checking
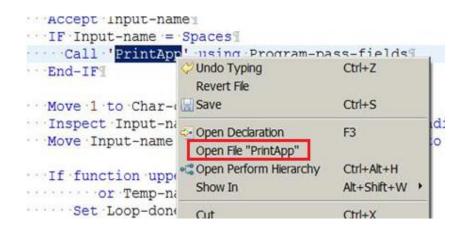
Real-time syntax check without requiring code compile or save
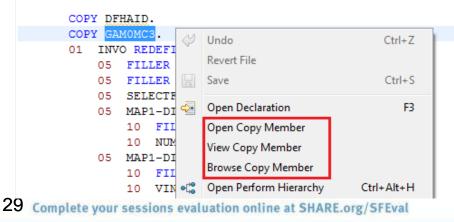
Editor Productivity Features

- Provide "**Open Called Program**" action
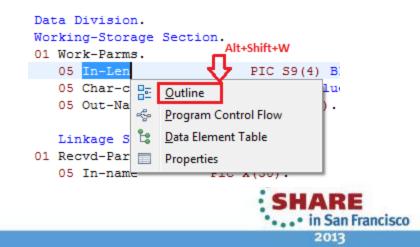
- Hyper linking support for Open/ Browse/ View copybooks/include files

. Show In > Outline action to COBOL and PL/I Editor

29

Editor Productivity Features

● Mark "Write occurrences" capability to the supported EXEC statements

● Occurrences within EXEC statements known to be "writes" are highlighted with a BROWN background

• All "read" statements will continue to be highlighted with a GREY background

```
* PHRASE and NEWPHRASE are read-only data areas;
* ESMREASON and ESMRESP are write
    EXEC CICS
    CHANGE PHRASE(data-area) PHRASELEN(data-value)
           NEWPHRASE(data-area) NEWPHRASELEN(data-value)
           USERID(data-value)
           ESMREASON(data-area) ESMRESP(data-area)
    END-EXEC.
```

```
/* INTO is write, LENGTH is read */
EXEC DLI STATISTICS
 USING PCB(expression)
  INTO(area1)
  LENGTH(area1)
  VSAM
 FORMATTED
;
```

```
* INTO :hv1:ind1, :hv2:ind2 are WRITE,  :hv3 is READ
    EXEC SQL
        FETCH ABSOLUTE :hv3 CURSOR1 INTO :hv1:ind1, :hv2:ind2
    END-EXEC.
```

```
* all READ
    EXEC SQL
    FREE LOCATOR :hv1, :HV2, :HV3
    END-EXEC.
```

Search for Occurrences Action

➢ Once a variable is selected the user triggers the "Find Occurrences" action using the Menu under search or keyboard shortcut "Ctrl+Shift+U"

➢ The occurrences are shown in the "Search results" page

Complete your sessions evaluation online at SHARE.org/SFEval

•••• in San Francisco

2013

Copy book and Include file resolution

- Hover over a COPY book name or a INCLUDE file to see the contents
  - Pressing F2 when hovering will "pin" the hover as shown
  - The window can then be dragged to expand, the Pencil icon shown below can be used to edit the copy book

JCL Template Support

➤ Templates are provided for standard JCL statements, and users can create their own Templates
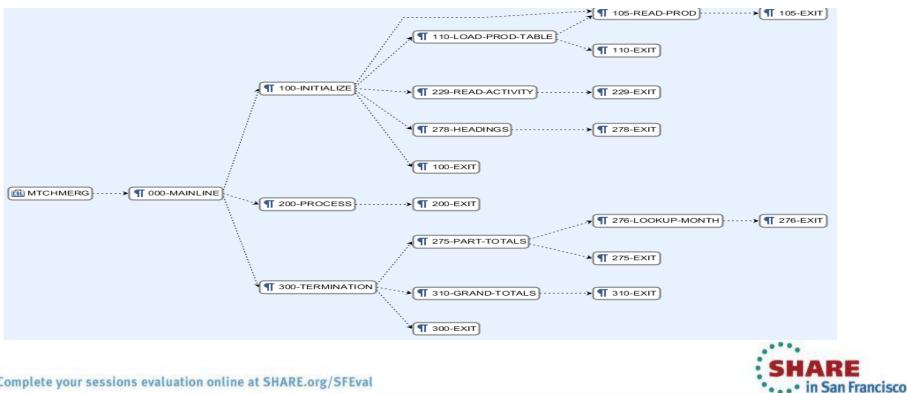
➤ When editing .jcl file using "Ctrl+Space" in the editor will trigger a pop list allowing the user to select the template to insert into the editor contents

Complete your sessions evaluation online at SHARE.org/SFEval

Enhanced Application Quality & Structure Analysis

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

Complete your sessions evaluation online at SHARE.org/SFEval

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

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".

SHARE in San Francisco

2013

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

Complete your sessions evaluation online at SHARE.org/SFEval

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

Debug Multiple Runtimes – using IBM Debug tool

- Use the cross-platform debugger to debug end-to-end systems as they execute in the runtime
  - IMS
  - CICS
  - DB2
  - Batch
  - WAS
  - Java
- From the workstation:
  - View executing source code
  - Step through host code line-by-line
  - Set breakpoints
  - Alter working storage values
  - Alter register values
  - Etc…
- Debug zOS and distributed code in the same interface even stepping between runtimes and platforms!

Enhanced Application Quality – Code Coverage

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

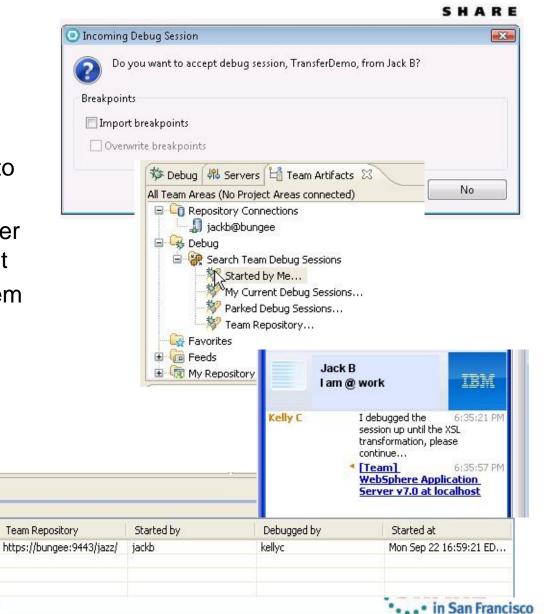Complete your sessions evaluation online at SHARE.org/SFEval

Innovative Debugging using Collaboration

- Collaborative debug with RDz and IBM Debug Tool via the Rational Team Concert Server!
  - Share breakpoints and monitors with other team members
  - Transfer debug session control to other users
  - Save debugging sessions for later retrieval in the team environment
  - Works for WAS (JEE) and System z applications

in San Francisco
2013

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
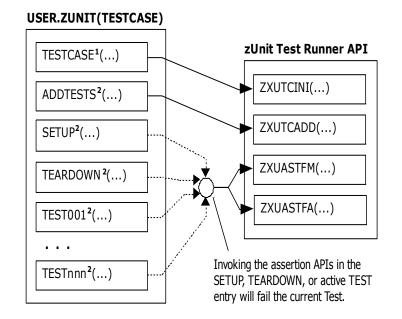
Complete your sessions evaluation online at SHARE.org/SFEval

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 *x*Unit.
- 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

USER.ZUNIT(TESTCASE)

zUnit Test Runner API

USER.ZUNIT(TESTCASE)	zUnit Test Runner API
TESTCASE[1](…)	ZXUTCINI(…)
ADDTESTS[2](…)	ZXUTCADD(…)
SETUP[2](…)	ZXUASTFM(…)
TEARDOWN[2](…)	ZXUASTFA(…)
TEST001[2](…)	
. . .	
TESTnnn[2](…)	

Invoking the assertion APIs in the SETUP, TEARDOWN, or active TEST entry will fail the current Test.

[1]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, the 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.

[2]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).
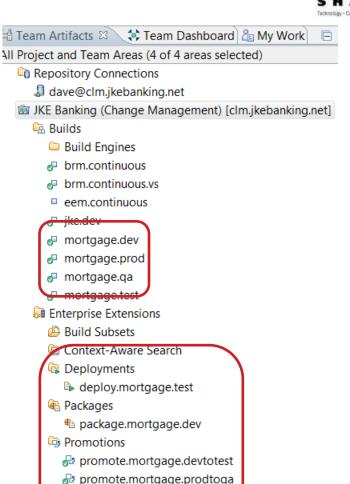
Development Life Cycle

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**

Enterprise Extensions Specific Functions

- Dependency build
  - "Smart build" of z/OS and IBM i applications, based on what has changed
- Promotion
  - Flow of source code changes and build outputs through development hierarchy
- Packaging and deployment
  - Package build outputs and deploy to another system (e.g. test environment, QA, production, etc)

The big picture

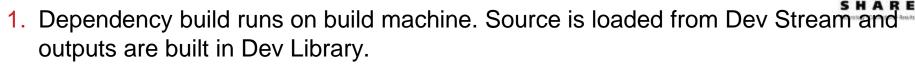1. Dependency build runs on build machine. Source is loaded from Dev Stream and outputs are built in Dev Library.
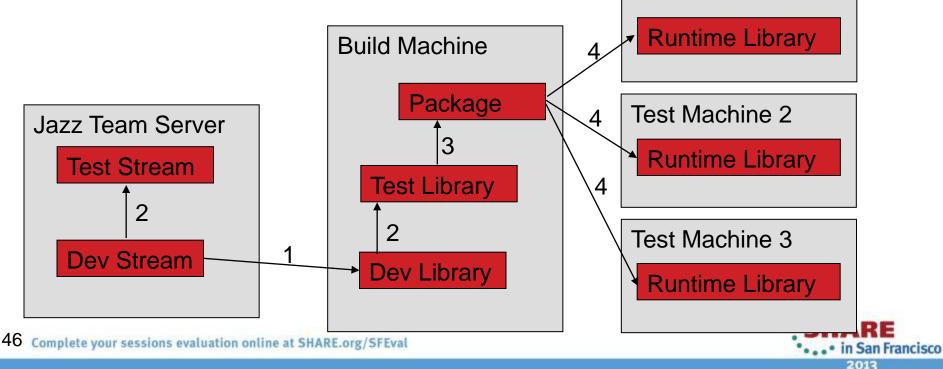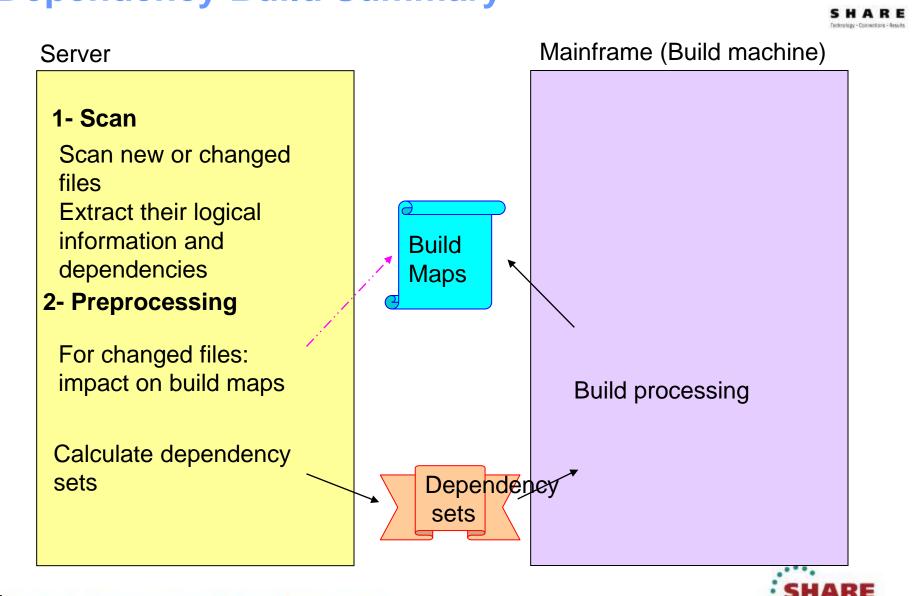2. Promotion build runs on build machine. Source is promoted from Dev Stream to Test Stream and build outputs are copied from Dev Library to Test Library.
3. Package build runs on build machine. Test Library build outputs are archived in a package.
4. Deploy build runs on various test machines. Package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

Complete your sessions evaluation online at SHARE.org/SFEval

Dependency Build Summary

Server

Mainframe (Build machine)

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

Build Maps

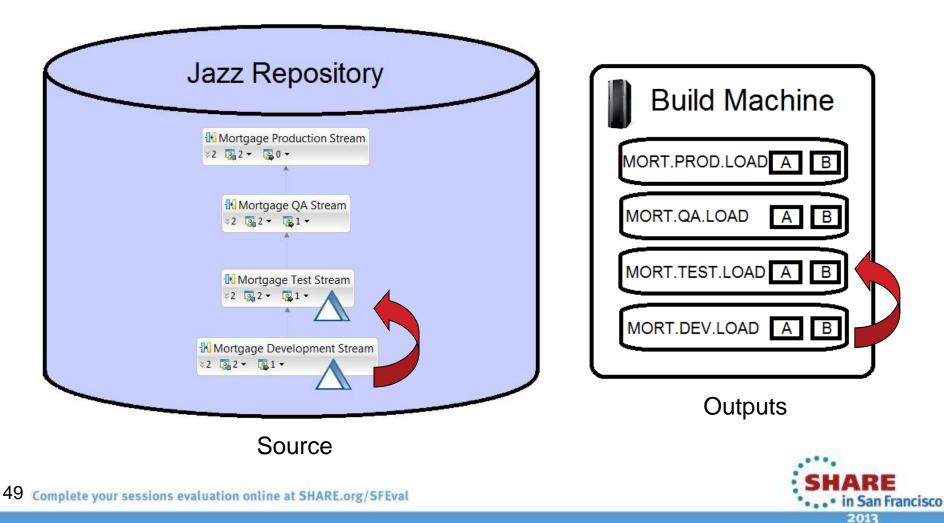Dependency sets

Build processing

Snapshots for every build

Promotion

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

Source

Outputs

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

Empower

Unify

Revitalize

Enterprise Modernization

Optimize

Enabling Product and Service Innovation

Jazz

➢ **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**

Complete your sessions evaluation online at SHARE.org/SFEval

Complete your sessions evaluation online at SHARE.org/SFEval

SHARE
• in San Francisco
2013