A decorative graphic in the top left corner of the slide. It features several overlapping circles of various colors, including yellow, orange, red, purple, and blue. Each circle is divided into smaller segments, creating a complex, multi-colored pattern.

Smarter Development and Debugging for IBM System z

Venkat Balabhadrapatruni

Chief Architect, Rational Developer for System z

Steve Cooper

Architect, Parallel Debugger

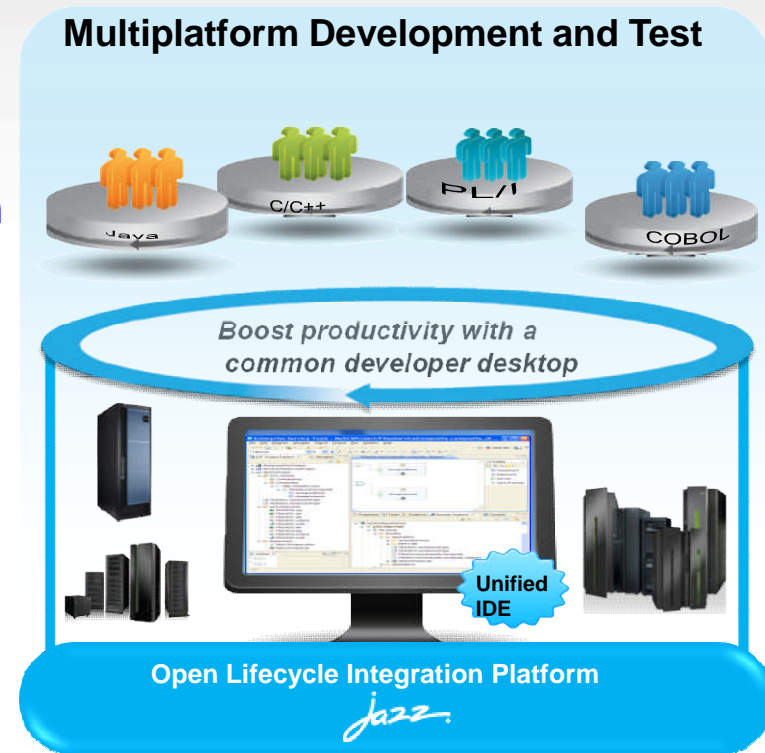


IBM Rational Developer for System z



An integrated development environment enabling enterprise modernization:

- ✓ An Eclipse-based IDE designed to improve the productivity of mainframe developers regardless of their technical background
- ✓ Tooling to create, modernize and maintain COBOL, PL/I, C/C++, Java, or Assembler transactional applications for CICS, IMS, and IBM DB2
- ✓ Support for SOA and Web services creation
- ✓ Integration with the Rational product suite in support of mobile development and DevOps
- ✓ Cross-platform support (z/OS, AIX, Linux on Power, Linux x86 and IBM i) provided by RD for the Enterprise



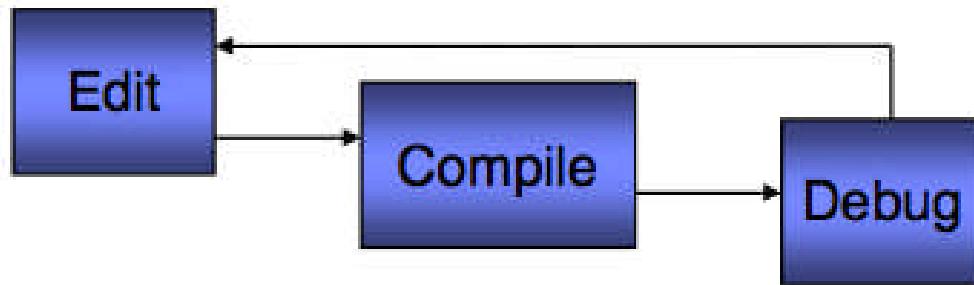


IBM Rational Developer for System z



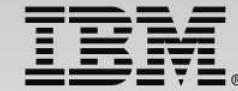
An integrated development environment to boost your E/C/D productivity:

- ✓ Rich source editing and navigation
- ✓ Code review for code governance
- ✓ Integration with the platform's latest compiler technology (including the recently released Enterprise COBOL for z/OS, V5.1)
- ✓ ... and a new RDz Integrated Debugger to provide a seamlessly integrated & complete Edit/Compile/Debug solution out-of-the-box





The Benefits of an IDE



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

The screenshot displays the IBM RDz IDE interface with several key components and annotations:

- Left Panel:** A project browser showing a tree structure of folders like 'DebugLab', 'DemoProj', 'HospProj', 'MontpZOS', 'bnchmark', and 'zosProject'.
- Central Editor:** A code editor window titled 'BNCHS601.cbl' showing COBOL code. A context menu is open over the code, with 'Syntax Check' selected and a sub-menu showing 'Local' and 'Remote' options. A red arrow points to the 'Remote' option with the label 'Submit a Compile'.
- Right Panel:** A 'Snippets' and 'Remote Systems' panel showing a tree of remote systems and datasets, including 'z/OS UNIX Files', 'MVS Files', and 'My Data Sets'. A red arrow points to this panel with the label 'File Search'.
- Bottom Left:** A 'Properties' panel showing dataset attributes like 'BLKSIZE', 'RECFM', and 'SECONDARY'.
- Bottom Center:** A 'Remote Error List' and a table of job resources.
- Annotations:**
 - 'Edit a program' is written in red above the code editor.
 - 'File Compare' is written in red next to the context menu.
 - 'Access Datasets + Dataset Management' is written in red on the right side.
 - 'Dataset Statistics' is written in red above the Properties panel.
 - 'Access Jobs (Outlist facility)' is written in red above the job resource table.

Resource	Job ID	Job Name	Job Owner	Job Entry ...	Return Code	Return Info	User retur...	Return Sta...	Spool Tracks	Priority
DD50001:TSU16893	TSU16893	DD50001	DD50001	2010/04/07...	5622	ABENDed		ABEND	1	1
DD50001:TSU16				2010/04/07...	5622	ABENDed		ABEND	1	1
DD50001:TSU16				2010/04/07...	5622	ABENDed		ABEND	1	1
DD50001:TSU16				2010/04/07...	5622	ABENDed		ABEND	1	1
DD50001:TSU16				2010/04/07...	5622	ABENDed		ABEND	1	1
DD500011:JOB1				2010/04/07...	U0004	NORMAL	004	COMPLETION	1	1
DD500011:JOB1				2010/04/07...	U0012	NORMAL	00C	COMPLETION	1	1



Rich Edit capabilities



```
-----+*A-1-B-----2-----3-----4-----5-----6-----7-----8
000010 IDENTIFICATION DIVISION.
000020*
000030 PROGRAM-ID.        PARTSUPP.
000040*
000050 DATE-WRITTEN.      OCTOBER 1ST 1979.
000060*
000070 DATE-COMPILED.
000080*
000110*****
000140*
000150*      PROGRAM NUMBER
000170*      PROGRAM NAME
000180*
000190*****
000200*
000210*
000220*
```

600

- PROGRAM: PARTSUPP
 - PROCEDURE DIVISION
 - 600-SETUP-NEW-POS SECTION.
 - 6000-SUPPLR-TOTALS SECTION.
 - 7600-LADING-COLR SECTION.
 - 8600-REJECT-REPORT-RECORD SECTION.
 - 9600-DATE-RETURN-CODE SECTION.

```
PROCEDURE DIVISION.

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

Change to "MOVE"

```
000067 *
000068 0150-SECOND-PART.
000069 MOVE 2 TO BRANCHFLAG.
000070 move
000071
000072
000073
000074
000075 0200
000076
000077 * If
000078
000079
```

- CUST-ADDR1
- CUST-CITY
- CUST-CTRY
- CUST-FN
- CUST-LN
- CUST-NO
- CUST-ST
- POTVSAM-RECORD-REC

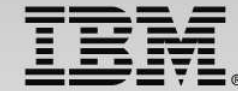
POTVSAM-FILE
01 POTVSAM-RECORD-REC.
...
03 CUST-ADDR1 PIC X(20).

force a divide by ZERO

D-FROM-CALLED



Enterprise web services



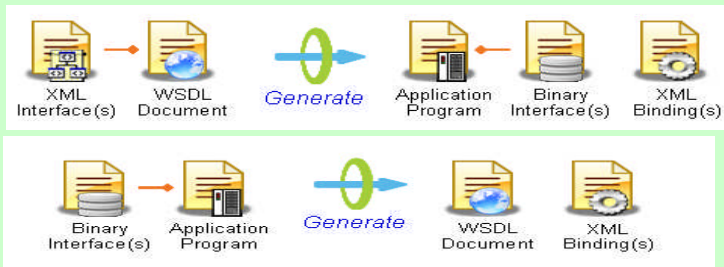
Supports modern architecture development

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

Enterprise Service Tools

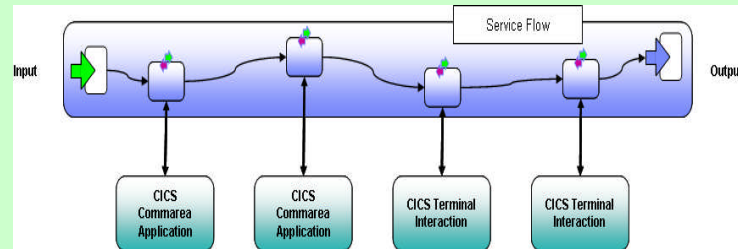
Single service projects

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



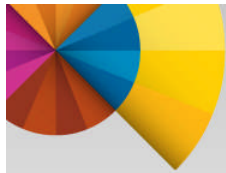
Service flow projects

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

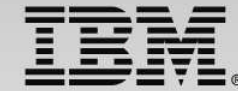


Enable Enterprise Applications for Mobile and Web

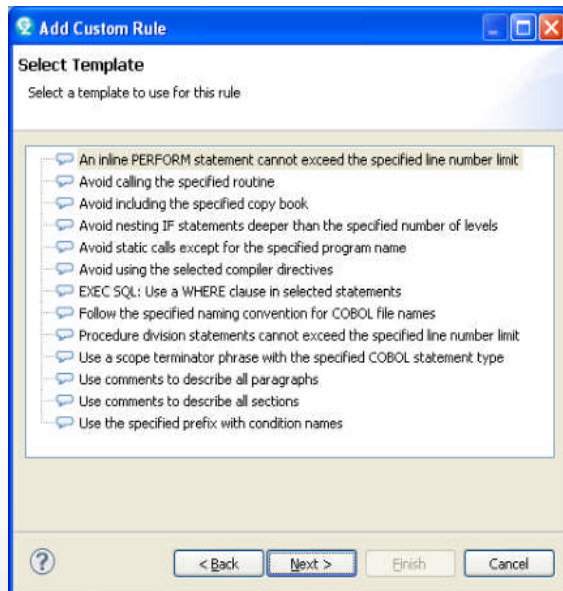




Code review & analysis

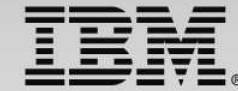


- Code Review/Governance - provides predefined rules and templates for COBOL and PL/I applications
 - Ensure adherence to corporate standards

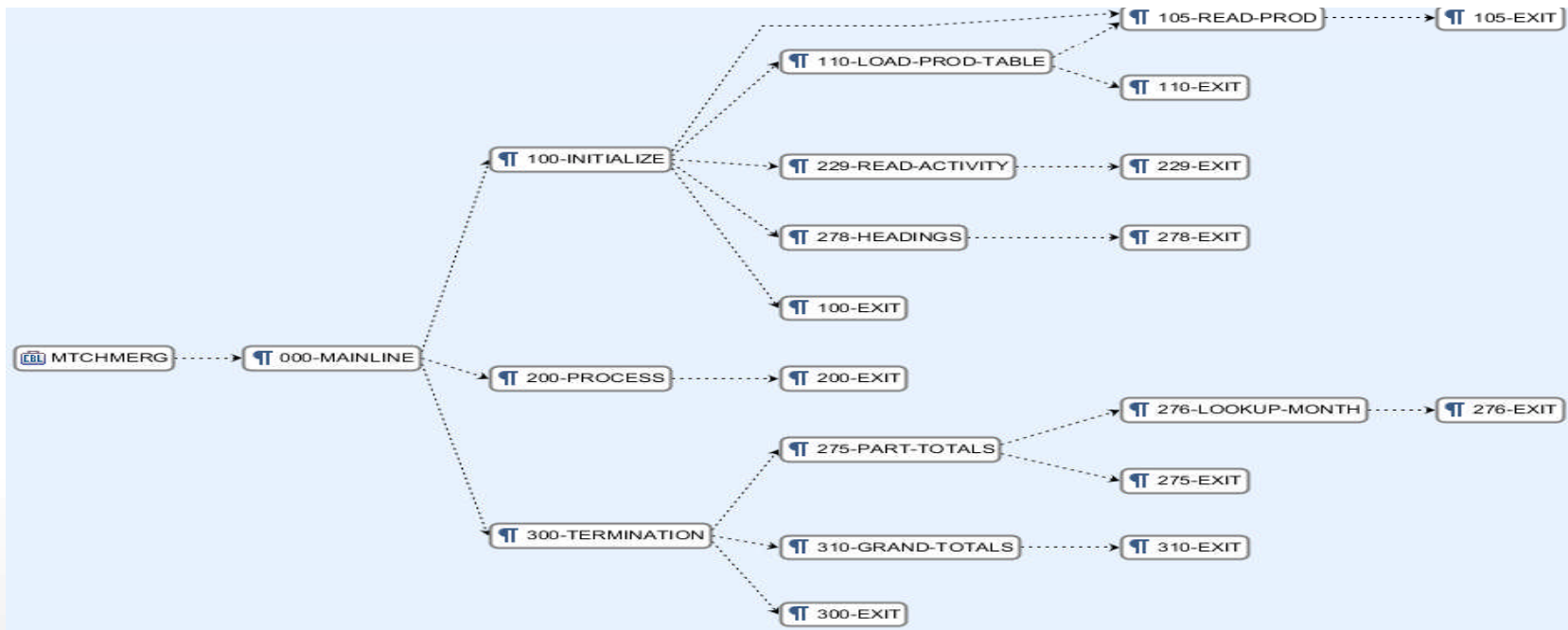


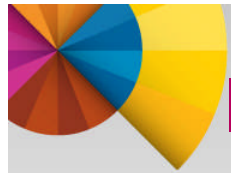


Enhanced Application Understanding



- 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





Integrated Debugger...

IBM®



A GUI-based, powerful, multi-platform, multi-language debugger:

✓ **Platform exploitation:**

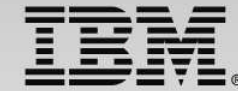
- ✓ Language and subsystem support in RDz v9.0.1
 - ✓ COBOL V5.1, V4, V3.4
 - ✓ Batch, Batch IMS, Batch DB2, CICS 5.1, 4.2, 4.1

- ✓ Full asynchronous mode:
 - ✓ Thread-level control of multithreaded applications

- ✓ Automonitor support



Integrated Debugger...



A GUI-based, powerful, multi-platform, multi-language debugger:

- ✓ **Interactive Code coverage:**
 - ✓ Measures and reports on the test coverage of an application
- ✓ **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



Integrated Debugger...



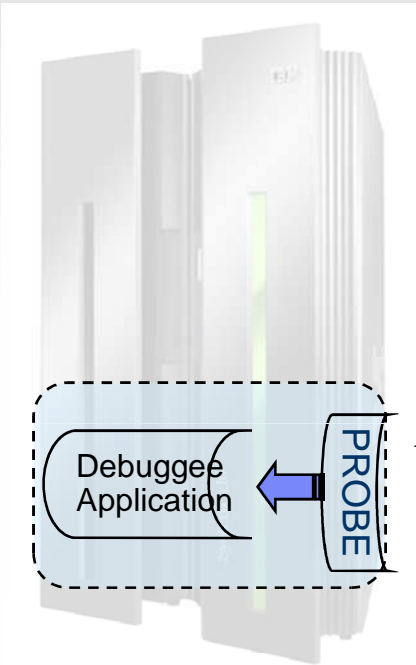
- ✓ **Leverages Debug Manager to:**
 - ✓ Accept incoming connections from the client engine
 - ✓ Authentication of the client
 - ✓ Eliminate dependency on client IP/hostname to start debug sessions
 - ✓ Encrypt communications
 - ✓ SSL/TES encryption



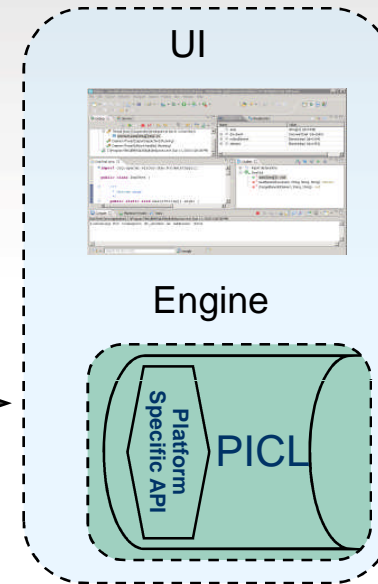
Integrated Debugger Architecture



System z (z/OS)



RDz Client



TCP/IP





Leveraging the Debug Perspective...



The screenshot displays the IBM Rational Developer for System z interface in the Debug perspective. The main window shows the source code of a COBOL program named SAM1.list.cob. The code includes a PROCEDURE DIVISION and a main routine (000-MAIN) that accepts current date and time from the system and displays them. The current line of execution is highlighted in green.

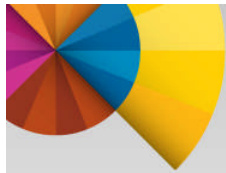
The Variables window on the right shows the following data:

Name	Value
CURRENT-DATE	{ 14, 01, 29 }
CURRENT-DAY	29
CURRENT-MONTH	01
CURRENT-TIME	{ .., .., .. }
CURRENT-HOUR	..
CURRENT-MINUTE	..

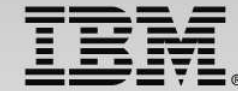
The Outline window on the bottom right shows the program structure:

- PROCEDURE DIVISION.
 - 000-MAIN.
 - 100-PROCESS-TRANSACTIONS.
 - 200-PROCESS-UPDATE-TRAN.
 - 210-PROCESS-ADD-TRAN.
 - 220-PROCESS-DELETE-TRAN.
 - 299-REPORT-BAD-TRAN.

The Console window at the bottom is currently empty, and the status bar indicates "Not connected" and "<No Current Work>".



Interactive Code Coverage



You can generate code coverage statistics from within the workbench:

- ✓ RDz interactive code coverage determines the extent and effectiveness of test coverage
 - ✓ If you can debug it... you can analyze code coverage
 - Does not require instrumentation of the executable
- ✓ Measures and captures code coverage statistics for an application
 - ✓ Acceptance criteria are customizable
- ✓ Reports on line-level or function-level coverage
 - ✓ Workbench and HTML report formats are supported
 - Supports comparisons to previous results (delta markers indicate differences)
 - ✓ Code coverage is also conveyed through customizable decorations in the editor

Element	Coverage	Covered Lines	Total Lines
source.a	100%	55/55	164
main.c	100%	55/55	164
main.o	100%	41/41	96
main.o	100%	3	3
main.o	100%	1/1	4
main.o	100%	0/0	5
main.o	100%	0/0	5
main.o	100%	37/37	69
main.o	100%	63	63
main.o	100%	13	13
main.o	100%	1	1

```
// Use the functions in cout to set the decimal places
// in the output
cout.setf(ios::fixed);
Lines 105 to 106 Covered (2)

// Use the inline function title
title();

// Use the payout function
payout(managers_pay);

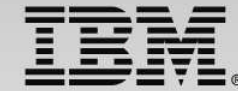
// Use the overloaded payout function
payout(reg_emp_pay, reg_emp_hrs);

// Use another overloaded payout function
payout(monthly_salary, commission, units);

// A place to put in a security feature
```



RDz - IBM PD Tools integration



- Leverage the problem diagnosis capabilities of PD Tools to have a comprehensive and state of the art z/OS application development and problem diagnosis environment

The screenshot displays the IBM PD Tools interface with several panels:

- Control flow diagram:** A flowchart showing program execution paths with nodes like '000-STARTER', '000-PROGRAM', and '000-TERMINATOR'.
- Variables:** A table listing variables and their values:

Name	Value
LOOP-DONE	0
PROGRAM-FLAGS	
PROGRAM-OTHER-FIELDS	
INPUT-NAME	"Bill Hudak"
CHAR-COUNT	00
PROGRAM-PASS-FIELDS	
SQLA-PROGRAM-ID	
- Source code (StartApp.cbl):** A code editor showing COBOL code for input processing:

```
Line 58      Column 1      Insert      Browse
-----*A-1-B-----2-----3-----4-----5-----6-----7-----
05 Input-name      Pic x(30).
  Move Spaces to Input-name
  Accept Input-name
  IF Input-name = Spaces
    Move "Q" to Input-name
  End-IF

  Move 1 to Char-count
  Inspect Input-name Tallying Char-count For Leading Spac
  Move Input-name(Char-count: 30 - Char-count) to Temp-na

  If function upper-case (Temp-name) = "Q"
    or Temp-name = Spaces
```
- HEALTH_PLAN Table:** A data table with columns PLAN_ID, GROUP_ID, and PROVIDER. A context menu is open over the row 'GBINS-2D5-T00IX81-01'.

PLAN_ID [CHAR(20)]	GROUP_ID [CHAR(10)]	PROVIDER [CH
GBINS-2FD-T00IX81-00	GRP-000D-5	INS-0001
GBINS-2FD-T-0IX81-A0	GRP-000D-1	INS-0002
FBINE-3D5-I-09323E32	GRP-000D05	INS-0004
FBINE-3D5-I-09323E3D	GRP-003-HU	INS-0005
GBINS-2WK-T00IX81-00	GRP-00WK-1	INS-WK01
GBINS-2D5-T00IX81-01		INS-006
<new row>		
- Monitors:** A panel showing variable values: CHAR-COUNT = 00, INPUT-NAME = "Bill Hudak", TEMP-NAME = "", PROGRAM-OTHER-FIELDS, INPUT-NAME = "Bill Hudak", CHAR-COUNT = 00.
- Data Source Explorer:** Shows a tree view with Tables: CUST, DIAG_CODES.



DEMO



Questions



Thank You

© Copyright IBM Corporation 2014. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.