

2015 DevOps Customer Call: Using New IBM Compilers to Reduce Operating Costs for Business-Critical Applications on z13

Roland Koo - Product Manager, Compilers

Tom Ross - Captain COBOL: compiler and run time development

Peter Elderon - z System PL/I Compilers and Architecture

Catherine Lung - C/C++ for Linux on z Systems - Optimization Technology lead

Visda Vokshoori - z/OS C/C++ - Optimization and Code Generation lead

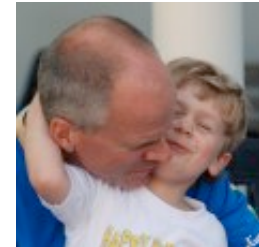


Speakers



Roland Koo - Product Manager,
Compilers

Tom Ross - Captain COBOL: compiler
and run time development



Peter Elderon - z System PL/I
Compilers and Architecture

Catherine Lung - C/C++ for Linux on z
Systems - Optimization Technology
lead



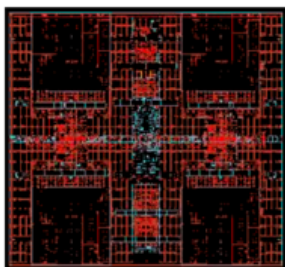
Visda Vokhshoori - z/OS C/C++ -
Optimization and Code Generation lead

Agenda

- Why stay current on compilation technology ?
- Enterprise COBOL V5.2
- Enterprise PL/I V4.5
- XL C/C++ for Linux on z Systems
- z/OS XL C/C++ V2R1M1

z Systems - Processor Roadmap

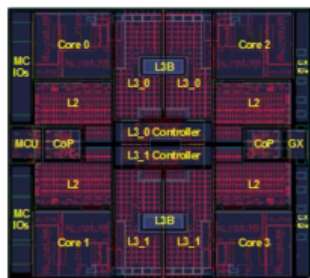
z10
2/2008



Workload Consolidation and Integration Engine for CPU Intensive Workloads

- Decimal FP
- Infiniband
- 64-CP Image
- Large Pages
- Shared Memory

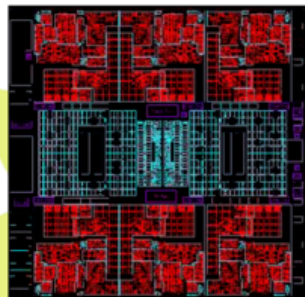
z196
9/2010



Top Tier Single Thread Performance, System Capacity

- Accelerator Integration
- Out of Order Execution
- Water Cooling
- PCIe I/O Fabric
- RAIM
- Enhanced Energy Management

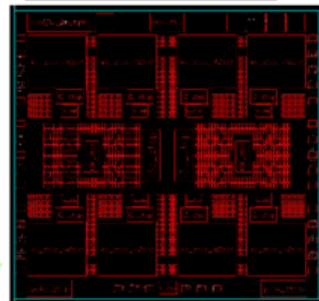
zEC12
8/2012



Leadership Single Thread, Enhanced Throughput

- Improved out-of-order
- Transactional Memory
- Dynamic Optimization
- 2 GB page support
- Step Function in System Capacity

z13
1/2015

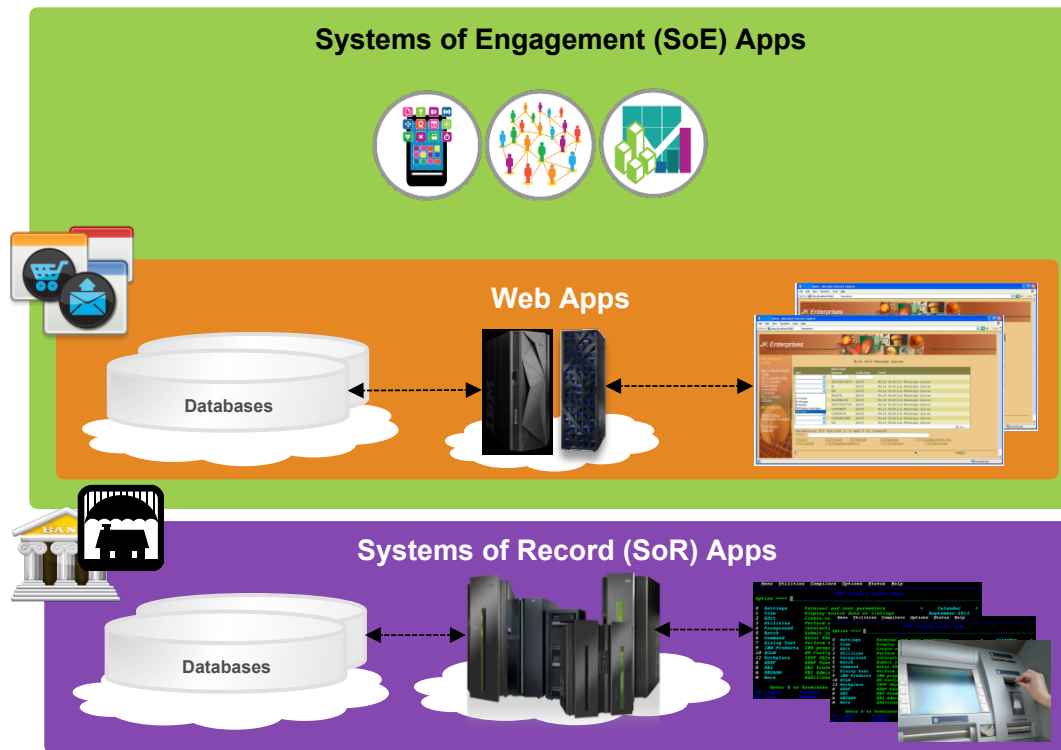


Leadership System Capacity and Performance

- Modularity & Scalability
- Dynamic SMT
- Supports two instruction threads
- SIMD
- PCIe attached accelerators (XML)
- Business Analytics Optimized

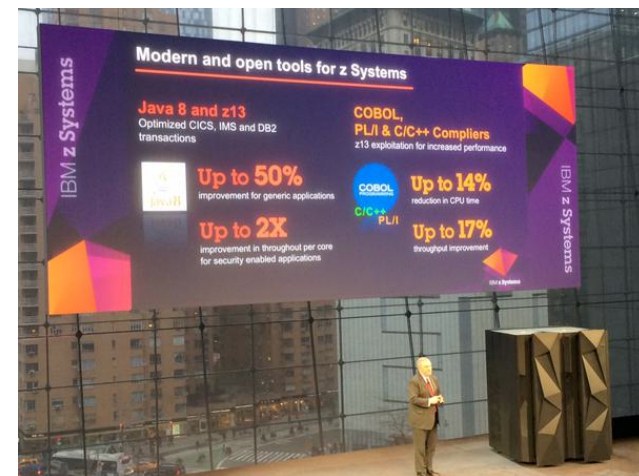
Developing next-generation Applications

Extend Business Critical (SoR) Applications to support new System of Engagement (SoE) Applications; Increase performance and increase efficiency and quality of software delivery with new compiler technologies



New Compilers launched with z13

- Enterprise COBOL for z/OS v5.2
- Enterprise PL/I for z/OS v4.5
- XL C/C++ Compiler for Linux on z Systems
- z/OS XL C/C++ V2R1M1

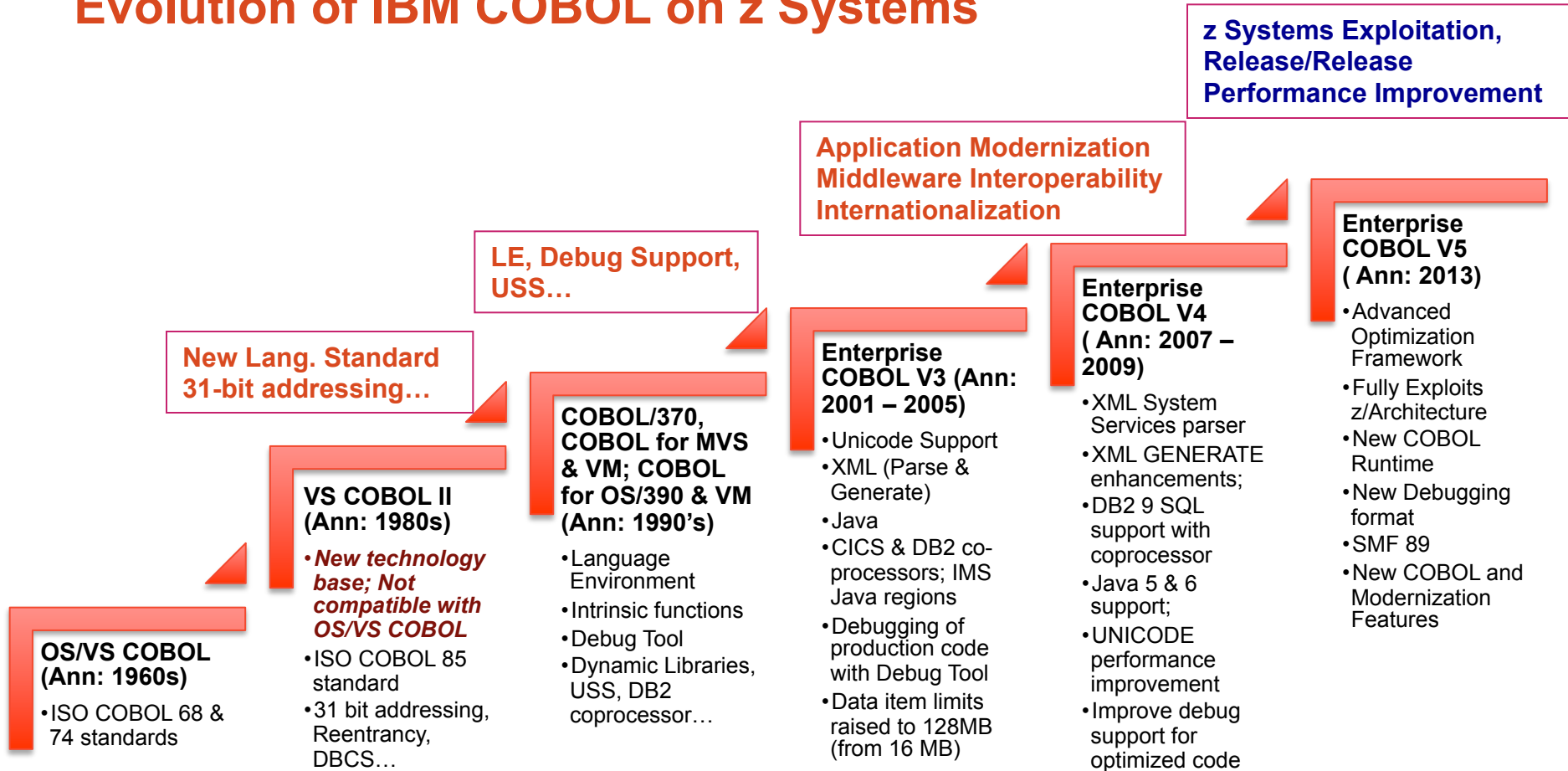


Enterprise COBOL for z/OS v5.2

Tom Ross - Captain COBOL: compiler and run time development



Evolution of IBM COBOL on z Systems



Enterprise COBOL for z/OS V5.1

GA: Jun 21, 2013

- New optimization framework
 - Delivered greater than 10% performance improvement over Enterprise COBOL v4 for well structured, CPU-intensive batch applications¹
 - Lays solid groundwork for delivering release-to-release performance improvement roadmap for COBOL on z Systems
- New COBOL Runtime
- New Object and Debugging formats

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

Enterprise COBOL for z/OS V5.1

GA: Jun 21, 2013

- Maintains source and binary compatibility
 - Correct COBOL programs will compile and execute without changes and produce the same results
 - “Old” and “new” code can be mixed within an application
 - Removed some old language extensions and options
- Supports the ecosystem of programming tools supplied by IBM and ISVs.



Things you should know when upgrading to V5

- PDSE load libraries
- Dataset and Memory Requirements & SMF
- Old “OS/VS COBOL” and “VS COBOL II NORES” Code

Migrating to COBOL V5 requires advanced planning and more testing.



Enterprise COBOL for z/OS V5.1 – Since GA

Continuously Delivered Improvements based on customers' feedback via PTF stream

- COBOL V3/V4 features to improve migration
 - AMODE 24, XMLPARSE(COMPAT), VLR(COMPAT), MAP(HEX), ZONEDATA(MIG)
- New features
 - Support for IMS V13 EXEC SQLIMS
- Performance Improvements
 - Execution Performance: Working Storage, Procedure Pointer calls Unstring...
 - Compile time Performance

Complete Fix list for Enterprise COBOL for z/OS

<http://www-01.ibm.com/support/docview.wss?uid=swg27041164>

Enterprise COBOL for z/OS V5.2

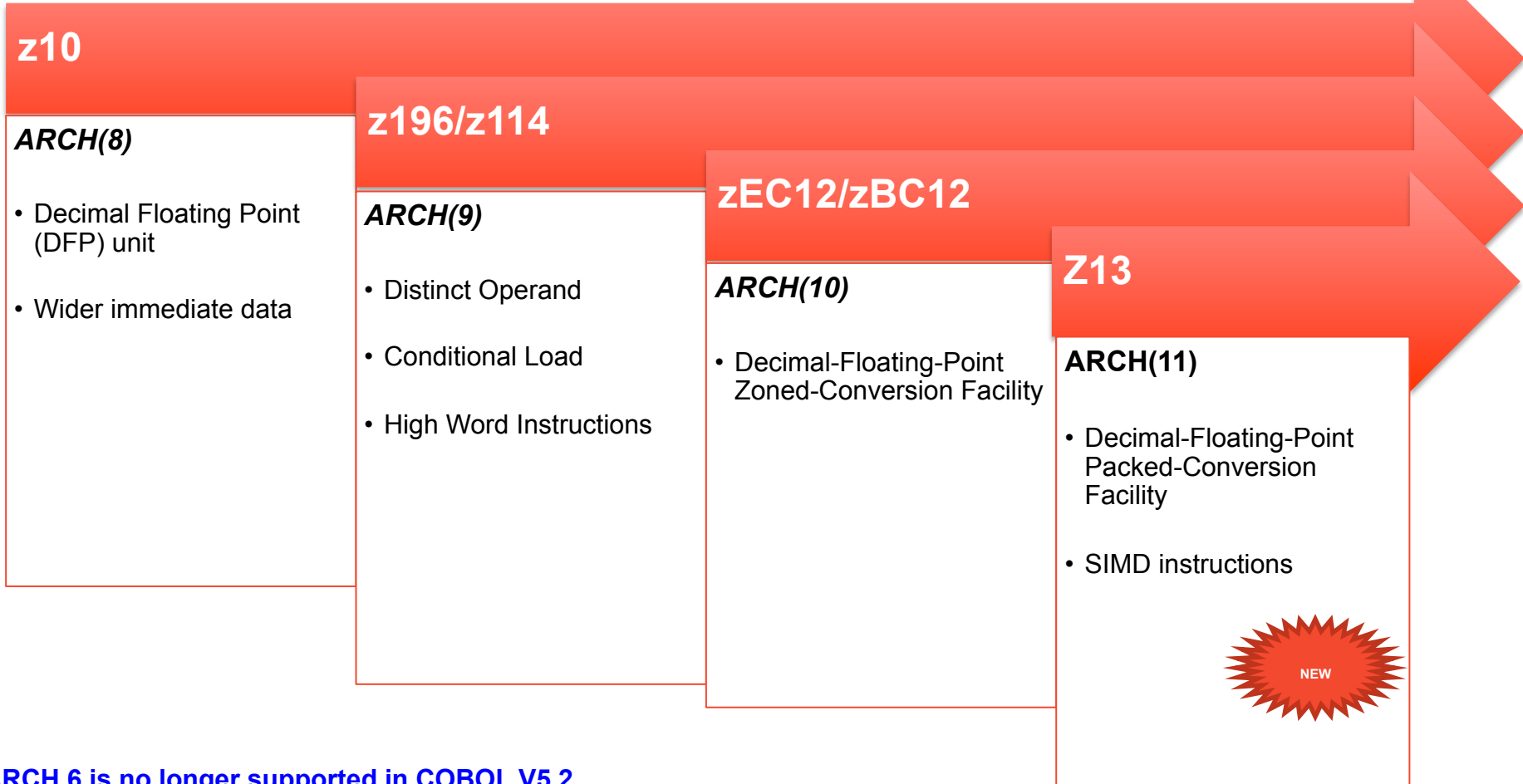
Ann: Jan 14, 2015; GA Feb 27, 2015

2nd release of the new COBOL roadmap

- Provides easy migration from COBOL V5.1
- Includes all enhancements delivered in COBOL V5.1 PTFs
 - Restored migration features, performance features, new programming features
- Provides Day 1 support for z13 processor
 - Instruction scheduler tuned to new micro architecture at ARCH 11
 - Expanded use of Decimal Floating Point for PACKED-DECIMAL data
 - Uses new SIMD instructions for INSPECT TALLYING or REPLACING statements
- Improves Application Performance
 - Well-structured, compute intensive batch applications running on z13 (compiled with Enterprise COBOL V5.2) have shown CPU time reduction of up to 14% over the same applications running on zEC12 (compiled with the GA release of Enterprise COBOL V5.1) ¹

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

Exploiting z/Architecture with ARCH option



ARCH 6 is no longer supported in COBOL V5.2

Default ARCH level is now ARCH 7 - IBM System z9EC (2094-xxx models) IBM System z9 BC (2096-xxx models)

Exploiting z13 – Example

PACKED-DECIMAL (COMP-3) faster with Decimal Floating Point

WORKING-STORAGE SECTION.

01 VARS.

02 A PIC S9(25) COMP-3 VALUE +1234567890123456789012345.

02 B PIC S9(25) COMP-3 VALUE +2468097531246809753124680.

02 C PIC S9(25) COMP-3 VALUE 0.

PROCEDURE DIVISION.

PERFORM 100000000 TIMES

DIVIDE A BY B GIVING C

END-PERFORM

V5.1 – ARCH10

```

XGR      R0,R0
ICMH     R0,X'1',152(,R8)      #  A
L        R0,153(,R8)          #  A
LG       R1,157(,R8)          #  A
CXSTR    FP0,R0
XGR      R0,R0
ICMH     R0,X'1',165(,R8)      #  B
L        R0,166(,R8)          #  B
LG       R1,170(,R8)          #  B
CXSTR    FP1,R0
DXTR     FP4:FP6,FP0:FP2,FP1:FP3
FIXTR    FP0:FP2,9,FP4:FP6
CSXTR    R0:R1,0,FP0:FP2
STCMH    R0,X'1',178(,R8)      #  C
ST       R0,179(,R8)          #  C
STG      R1,183(,R8)          #  C
ZAP      178(13,R8),178(13,R8) #  C
AHI      R2,0xffff
CIJ      R2,L0034,0,HT(mask=0x2),
    
```

V5.2 – ARCH11

```

CXPT     FP0:FP2,152(13,R8),0x8
CXPT     FP1:FP3,165(13,R8),0x8
DXTR     FP4:FP6,FP0:FP2,FP1:FP3
FIXTR    FP0:FP2,9,FP4:FP6
CPXT     FP0:FP2,178(13,R8),0x9
AHI      R2,0xffff
CIJ      R2,L0034,0,HT(mask=0x2),
    
```

Performance Comparison

Timing – (100 Million times in a loop)

V5.1 : 2.53 cpu seconds

V5.2 : 1.64 cpu seconds **(36% faster)**

Exploiting z13 – Example

INSPECT ... TALLYING faster with SIMD

```

WORKING-STORAGE SECTION.
01 VARS.
  02 STR PIC X(255).
  02 C PIC 9(5) COMP-5 VALUE 0.
PROCEDURE DIVISION.
  MOVE ALL 'abc def ghi jkl ' TO STR
  PERFORM 100000000 TIMES
    INSPECT STR TALLYING C FOR ALL ''
  END-PERFORM
  STOP RUN.

```

V5.1 – ARCH10

```

      LHI      R0,0xff
      XR       R1,R1
      LA       R12,152(,R8)      # STR
L0064: EQU     *
      CLI     0(,R12),X'40'      #
      JNOP    L0066
      LA      R1,1(,R1)          #
L0066: EQU     *
      LA      R12,1(,R12)        #
      BRCT   R0,L0064
      A       R1,407(,R8)        # C
      ST      R1,407(,R8)        # C

```

V5.2 – ARCH11

```

      LHI      R0,0xfe
      XR       R1,R1
      LA       R12,152(,R8)      # STR
      VREPIB   VRF27,0x40
      VGBM     VRF25,0x0
L0066: EQU     *
      VLL      VRF24,R0,0(,R12)  #
      AHI      R12,0x10
      VCEQB    VRF24,VRF24,VRF27
      AHI      R0,0xffff0
      VLCB     VRF24,VRF24
      VAB      VRF25,VRF25,VRF24
      JNL      L0066
      VGBM     VRF26,0x0
      VSUMB    VRF25,VRF25,VRF26
      VSUMQF   VRF25,VRF25,VRF26
      VLGVG    R1,VRF25,1(,R1)   #
      A        R1,407(,R8)        # C
      ST       R1,407(,R8)        # C

```

Performance Comparison

Timing – (100 Million times in a loop)

V5.1 : 46.63 cpu seconds

V5.2 : 1.54 cpu seconds (**30X faster !**)

Advanced Optimizations

- Provide multiple levels of optimization
- Debugging of optimized code is supported with OPT + TEST options

OPT(0)

- Minimum Optimization

OPT(1)

- Increased Optimization
 - e.g. Inline PERFORM statement
 - Commoning sub-expressions in a block
 - Sequential constant store simplification...

OPT(2)

- Maximum Optimization
 - e.g. Eliminating a stored value that is never re-used anywhere in the program
 - Global view of register assignment...
 - Instruction scheduling to exploit micro-architecture...

Longer compile time
Reduced debugging
Faster Executing Code

Enterprise COBOL for z/OS V5.2 - New features

- Access to z/OS JSON services * from COBOL.
 - Provide mobile (i.e. system of engagement) applications easy access to data and the processing they need from business critical enterprise (i.e. system of record) applications written in COBOL
- Improved XML GENERATE
 - More powerful SUPPRESS capabilities
- RULES option to help programmers write better code
- COPYRIGHT and SERVICE options to better manage applications
- New VOLATILE attribute and SERVICE LABEL functionality
 - Enable User-Written condition handlers to be compiled with OPT(1|2)
- Accessing VSAM data sets with extended addressability attribute is now supported
- Some New COBOL 2002 features

* z/OS Client Web Enablement Toolkit

New features from COBOL 2002 Standard

- Format 2 of SORT – the table SORT statement
 - Arranges table elements in a user-specified sequence
 - No file I/O necessary! Supported in CICS!
- New formats of EXIT statements
 - EXIT SECTION
 - EXIT PARAGRAPH
 - EXIT PERFORM
 - EXIT PERFORM CYCLE
- Improved COPY REPLACING statement
 - LEADING and TRAILING phrases (better partial word replacement)
 - Nested COPY REPLACING now supported
- Improved REPLACE statement
 - LEADING and TRAILING phrases (better partial word replacement)

For more information, visit <http://www-01.ibm.com/software/awdtools/cobol/zos/>

Pre-requisites for COBOL V5.2

- Hardware
 - Minimum hardware requirement: (IBM System z9 EC or BC)
 - COBOL V5.2 programs will not run on z990/z890 and earlier machines
- Software
 - z/OS V1R13 or V2
 - CICS Transaction Server, V3 or later
 - IBM DB2 V9 or later
 - IBM IMS V11 or later
 - PD tools V12 or later (Debug Tool, Fault Analyzer) and Application Performance Analyzer (V13))
 - Rational Developer for System z V9
- Required PTFs
 - Use **SMP/E MISSINGFIX** command w/keyword **Enterprise-COBOL.V5R2** to identify required PTFs
 - Required PTFs are not listed in COBOL for z/OS V5 Migration Guide or PSP bucket

Developer Trial

- Zero cost evaluation license for 90 days
 - Does not initiate Single Version Charging (SVC)
- Assess the value that could be gained from upgrading to Enterprise COBOL V5.2
- Offer same functionalities as Enterprise COBOL for z/OS V5.2
 - Code compiled with Enterprise COBOL Developer Trial cannot be used for production
- Available as standard offering through ShopzSeries on **March 27, 2015**
 - Contact your IBM representative for ordering assistance

For more information, visit

<http://www-03.ibm.com/software/products/ph/en/enterprise-cobol-developer-trial-for-zos>



Enterprise PL/I for z/OS v4.5

Peter Elderon - z System PL/I Compilers and Architecture



Enterprise PL/I

- Strategic Programming Language
 - Significant use in business applications but also in some scientific and engineering application
- Advanced optimization technology
 - Shares optimizing back-end technology with z/OS XL C/C++
 - Timely delivery of leading edge optimization and hardware exploitation to PL/I customers
- Time proven
 - First Enterprise PL/I product released in 2001
 - (Enterprise PL/I for z/OS and OS/390 3.1)
 - Latest release of Enterprise PL/I for z/OS (4.5) is based on same architecture
 - Provides easy migration
- Shipped 14 new releases in the last 15 years
 - Addressed many customer requirements
 - Improved optimization technology, z/Architecture exploitation, usability, middleware support, and application modernization features.



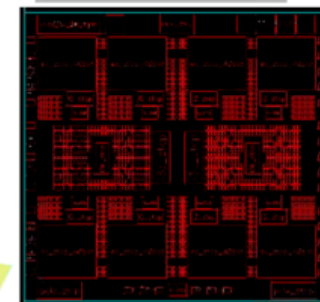
Enterprise PL/I

- The PL/I compilers from before 1999 use the set of machine instructions available as of 1985
 - They make no use of the many new instructions introduced in the many hardware releases since then
 - But you have paid for and installed that new hardware
 - Enterprise PL/I maximizes your return on that investment
- Enterprise PL/I also gives you options to enforce your coding standards
 - This insures your code is more transparent and more maintainable
 - It helps increase your code quality and reduces the number of runtime problems
 - By the use of the new compiler, a large Swiss insurance company saw that the number of programs failures that they needed to analyze went down from several a day to only a few per month

z Systems - Processor Roadmap

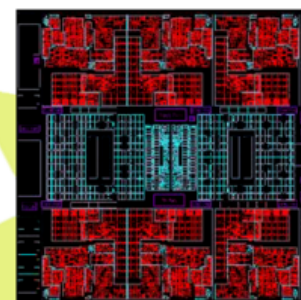
PL/I 4.5

z13
1/2015



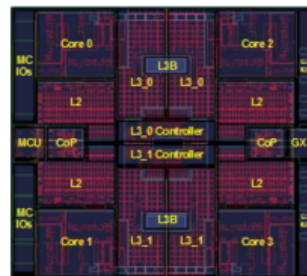
PL/I 4.3, 4.4

zEC12
8/2012



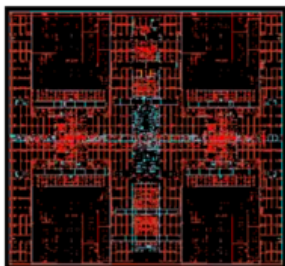
PL/I 4.1, 4.2

z196
9/2010



PL/I 3.9

z10
2/2008



Workload Consolidation and Integration Engine for CPU Intensive Workloads

- Decimal FP
- Infiniband
- 64-CP Image
- Large Pages
- Shared Memory

Top Tier Single Thread Performance, System Capacity

- Accelerator Integration
- Out of Order Execution
- Water Cooling
- PCIe I/O Fabric
- RAIM
- Enhanced Energy Management

Leadership Single Thread, Enhanced Throughput

- Improved out-of-order
- Transactional Memory
- Dynamic Optimization
- 2 GB page support
- Step Function in System Capacity

Leadership System Capacity and Performance

- Modularity & Scalability
- Dynamic SMT
- Supports two instruction threads
- SIMD
- PCIe attached accelerators (XML)
- Business Analytics Optimized

Enterprise PL/I for z/OS 4.5

- Advanced optimization and z13 exploitation
 - New optimizations to improve performance for packed decimal
 - Use of SIMD instructions to improve code for SEARCH and VERIFY
 - Inlining of MOD and REM of large packed decimal and multiply
 - Much better code for SELECT of CHAR(2) and CHAR(4)
- Provides significant performance improvements over Enterprise PL/I 4.4
 - Compute intensive applications running on z13 (compiled with Enterprise PL/I 4.5) have shown CPU time reduction of up to 17% over the same applications running on zEC12 (compiled with Enterprise PL/I 4.4)¹
- Addressed 28 RFEs

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

Enterprise PL/I for z/OS v4.5

Ann: Jan 14, 2015; GA: Feb 27,2015

- For example, this simple code tests if a UTF-16 string is numeric

```
wnum: proc( s );
```

```
    dcl s    wchar(*) var;
```

```
    dcl n    wchar value( '0123456789' );
```

```
    dcl sx   fixed bin(31);
```

```
    sx = verify( s, n );
```

```
    if sx > 0 then ...
```

- With ARCH <= 10, it is implemented via an expensive library call
- With ARCH(11), it is inlined with vector instructions and runs 2X faster

Enterprise PL/I and z13 speed up compares

- Or consider this code that validates a 2-byte country code

checkcc:

```
proc( countryCode )  
options( nodescriptor );
```

```
dc1 countryCode char(2);
```

```
if inlist( countryCode,  
          'AT', 'DE', 'CH', 'NL', 'DK', 'FI', 'SE', 'NO' ) then;
```

```
else  
  signal error;
```

Enterprise PL/I and z13 speed up compares

- Under arch(10) and opt(3), it becomes 8 compares and branches

```
5810 1000      L      r1,_addrCOUNTRYCODE(,r1,0)
4800 1000      LH     r0,_shadow1(,r1,0)
A70E C1E3      CHI    r0,H'-15901'
A784 0026      JE     @1L13
A70E C4C5      CHI    r0,H'-15163'
A784 0022      JE     @1L13
A70E C3C8      CHI    r0,H'-15416'
A784 001E      JE     @1L13
A70E D5D3      CHI    r0,H'-10797'
A784 001A      JE     @1L13
A70E C4D2      CHI    r0,H'-15150'
A784 0016      JE     @1L13
A70E C6C9      CHI    r0,H'-14647'
A784 0012      JE     @1L13
A70E E2C5      CHI    r0,H'-7483'
A784 000E      JE     @1L13
A70E D5D6      CHI    r0,H'-10794'
A784 000A      JE     @1L13
```

Enterprise PL/I and z13 speed up compares

- But under arch(11) and opt(3), one vector-find-any-equal and one branch do it faster and more simply!

5810	1000		L	r1,_addrCOUNTRYCODE(,r1,0)
4100	0002		LA	r0,2
E700	1000	0037	VLL	v0,r0,_shadow1(r1,0)
E720	E000	0006	VL	v2,+CONSTANT_AREA(,r14,0)
E700	2000	1082	VFAE	v0,v0,v2,b'0001',b'0000'
E700	0001	2021	VLGV	r0,v0,1,2
EC08	000B	007E	CIJE	r0,H'0',@1L4

Enterprise PL/I for z/OS 4.5 New Features

- JSON support
 - Support Parse, Generate, and Validate with native PL/I language
 - Also works with z/OS JSON services*
 - Allows enterprise (i.e. system of record) applications written in PL/I to be extended to handle data access and processing requests from mobile (i.e. system of engagement) applications
- Improved middleware support
 - Faster code for CICS calls
 - Support for named constants as SQL host variables
 - Allow structures as SQL indicator variables
- New productivity features
 - Extend size of strings from 32K to 128M
 - New INLIST and BETWEEN built-in functions
 - New built-in functions to generate the store-clock hardware instructions
 - New REINIT statement
 - Added features to help enforce code quality

For more information, visit <http://www-03.ibm.com/software/products/en/plizos>

* z/OS Client Web Enablement Toolkit

XL C/C++ Compiler for Linux on z Systems

Catherine Lung - C/C++ for Linux on z Systems - Optimization Technology lead



Linux for z Systems

- Growing rapidly
 - Installed IFL MIPS increased 12% from 4Q'13 to 4Q'14.
 - As of 4Q14...
 - 27.3% of Total Installed MIPS run Linux
 - 39% of z Systems Customers have IFL's
- Only choice for C/C++ developers is GNU C/C++
- IBM middleware, Business Intelligence & Analytic, and ISV workloads are driving demand to increase performance



XL C/C++ for Linux on z Systems

Ann: Jan 14, 2015; GA: Feb 16,2015

- Based on new technologies
 - clang front end
 - C/C++ language support (Partial C11, C++11)
 - GNU C/C++ language and option compatibility
 - Allows easy migration from distributed Linux systems to Linux for z Systems
 - IBM optimization technology shipped in Enterprise COBOL and IBM Java
- Ships with high performance Math Libraries tuned for z Systems
 - MASS (Mathematical Acceleration Subsystem software) and ATLAS (Automatically Tuned Linear Algebra Software)
 - Provide elemental and basic linear algebra functions to simplify coding and improve application performance
- Runs on RHEL (6 & 7); and SLES (11 & 12)

XL C/C++ for Linux on z Systems

- Provides significant performance advantage over GNU C/C++
 - Advanced optimization and z/Architecture exploitation
 - CPU intensive applications compiled with XL C/C++ for Linux on z Systems V1.1 have shown up to 10% performance improvement over the same applications compiled with GNU C/C++ V4.4 ¹
 - 7% performance improvement over the same applications compiled with GNU C/C++ V4.7¹.
 - World class service and support by IBM

For more information, visit <http://www.ibm.com/software/products/en/czlinux>

¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

XL C/C++ for Linux on z Systems

New ISA exploitation

```
xlc -march=z10 test1.c
```

```
LGR  %r0,%r7
AHI  %r0,1
LTR  %r0,%r0
BRC  0x4,.L_generateMTFValues_0167
LGFR %r0,%r7
LA   %r3,672(,%r2)
SLLG %r4,%r0,2
AGHI %r4,8
AGHI %r4,-1
BRC  0x4,.L_generateMTFValues_0192
```

```
xlc -march=z196 test1.c
```

```
AHIK  %r0,%r5,1
LTR   %r0,%r0
BRC   0x4,.L_generateMTFValues_0183
LGHI  %r13,0
RISBG %r11,%r0,30,189,2
AGHIK %r1,%r11,-15
BRC   0x4,.L_generateMTFValues_0184
```

```
xlc -march=zEC12 test1.c
```

```
AHIK  %r0,%r5,1
CIJL  %r0,0,.L_generateMTFValues_0183
LGHI  %r13,0
RISBGN %r11,%r0,30,189,2
AGHIK %r1,%r11,-15
CGIJL %r1,0,.L_generateMTFValues_0184
```

XL C/C++ for Linux on z Systems

Whole Program Optimization

a.c:

```
void starline();
```

```
int main()
```

```
{  
  starline();  
  return 0;  
}
```

b.c:

```
void starline()
```

```
{  
  for(int j=0; j<3; j++)  
    printf("***");  
}
```

```
xlc -O2 -qipa -c b1.c a1.c
```

```
xlc -O2 -qipa -S b1.o a1.o"
```

main:

.L_main:

```
    STMG  %r13,%r15,104(%r15)  
    LAY  %r15,-160(,%r15)  
    LARL  %r13,$CONSTANT_AREA  
    LGR  %r2,%r13  
    BRASL %r14,printf  
    LGR  %r2,%r13  
    BRASL %r14,printf  
    LGR  %r2,%r13  
    BRASL %r14,printf  
    LHI  %r2,0  
    LGFR %r2,%r2  
.L_main_0032:  
    LMG  %r13,%r15,264(%r15)  
    BCR  0xf,%r14
```

z/OS XL C/C++ V2R1M1

z/OS C/C++ - Optimization and Code Generation lead



IBM z/OS XL C/C++

- Optionally priced feature of z/OS
 - Enables development of high performing business applications, system programs and low level C applications
- IBM has been delivering leading edge C/C++ compilers on z/OS for over 20 years
 - Every release sets new standard for performance
 - Includes advanced optimization technology originally designed for HPC applications, and innovations to improve programmer productivity
 - Improves support for C and C++ language standards
- Provides system programming capabilities with Metal C option
 - Allows developers to use C syntax to develop system programs and low level free standing applications on z/OS without coding in HLASM
 - Significantly shortens the learning curve
 - Leverage advanced optimization technology to generate high performance optimized code



Innovation

z/OS XL C/C++ V2R1M1

Ann: Jan 14, 2015; GA: Feb 16, 2015

- z/OS Web deliverable -- for z13 customers
 - Provide Day 1 support for z13
 - Runs on z/OS V2.1 only
 - Replaces z/OS XL C/C++ V2.1 after installation
- Support for Vector/SIMD
 - Compile option, datatype, and built-in functions
- Ships with High performance Math Libraries tuned for z13
 - MASS (Mathematical Acceleration Subsystem) library providing scalar, vector, and SIMD mathematical elemental functions
 - ATLAS (Automatically Tuned Linear Algebra Software) library providing basic linear algebra functions

z/OS XL C/C++ V2R1M1 (cont'd)

- New Programming Features
 - New support for inline assembler (GNU compatibility)
 - New Debug support for Vector/SIMD data type
 - Header file cache optimization
- Provides significant performance improvements over z/OS XL C/C++ V2R1.
 - CPU intensive applications running on z13 (compiled with z/OS XL C/C++ V2R1M1) have shown up to 17% throughput improvement over the same applications running on zEC12 (compiled with z/OS XL C/C++ V2R1) ¹

For more information, visit <http://www-03.ibm.com/software/products/en/czos>

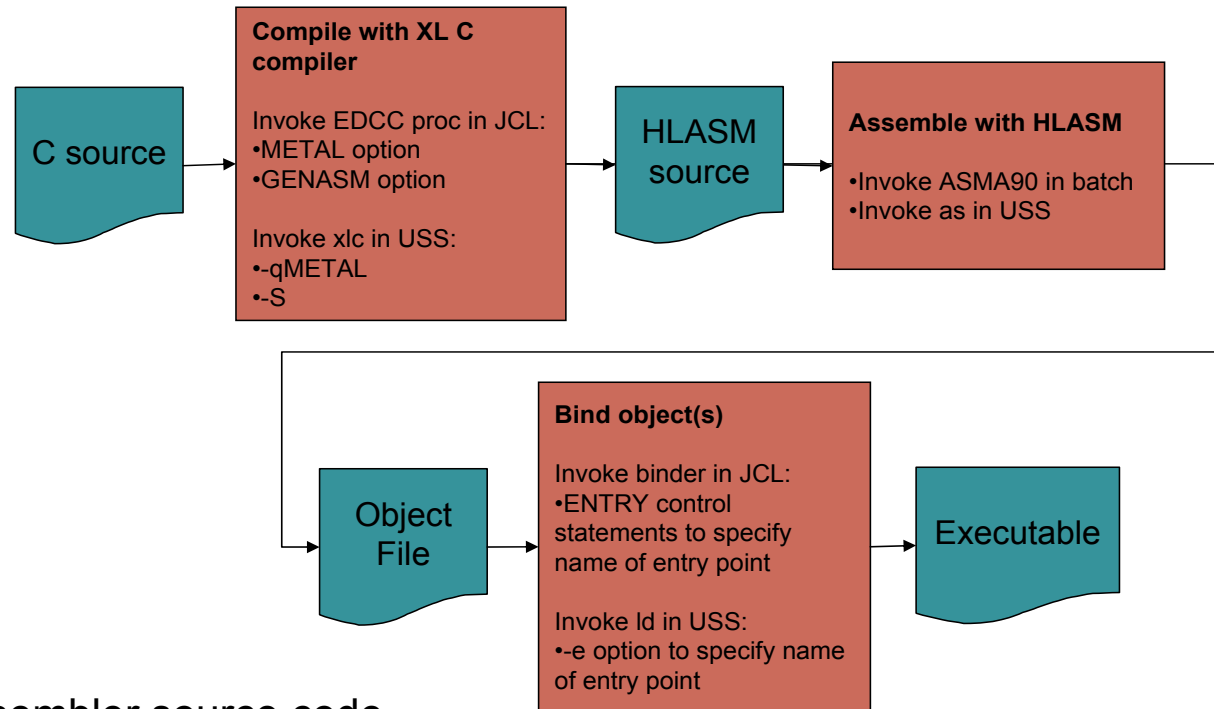
¹ Results are based on an internal compute-intensive test suite. Performance results from other applications may vary.

z/OS 2.2 XL C/C++

Ann: July, 2015; GA: Sept 30, 2015

- Ships with z/OS v2.2
- Runs on z/OS V2.2 only
- In addition to z/OS XL C/C++ V2R1M1
 - Improve Application Performance
 - AutoSIMD - Automatically convert serial code to take advantage of the vector facility
 - Light-weight runtime for Open MP
 - Default architecture level set to z10
 - Runtime performance improvements
 - Metal C enhancements – Larger DSAUSER space
 - Customer Requirements, Application Portability, and Programmer Productivity
 - dbgld option to capture all source in MDBG file
 - non-xplink version of CDA runtime to improve Fault Analyzer Support

Metal C Option



- Generates optimized assembler source code
 - Language Environment independent!
- Interoperates with existing HLASM programs
 - Uses MVS™ linkage conventions
- Supports AMODE 64 (allow mixing of AMODE 31 & AMODE 64 code)
- Provides a subset of C library functions.

Metal C optimizes for newer target architecture

No code modification required

`/bin/xlc -O3 -qmetal -S a.c -qarch=9`

```
@1L3      DS      0H          000010
          LR      1,2        000010
          AHI     2,1        000010
          LLC     0,1(1,15)  000010
          LTR     0,0        000010
          BRE     @1L4       000010
          LLC     0,2(1,15)  000010
          ALHSIK  2,1,2     000010
```

Optimized Assembly z196

`/bin/xlc -O3 -qmetal -S a.c -qarch=10`

```
@1L3      DS      0H          000010
          LR      1,2        000010
          AHI     2,1        000010
          LLC     0,1(1,15)  000010
          CIJE    0,0,@1L4   000010
          ALHSIK  2,1,2     000010
```

Optimized Assembly (zEC12)

Metal C whole program optimization

No code modification required

```
a.c:
int f1(void);

int total = 0;
int count = 0;

int main(int argc, char* argv[]) {
    for (int i=0; i<argc + 5; i++) {
        count++;
        total = total + f1();
    }
    return total + count;
}
```

Metal C

```
b.c:
int f1(void) {
    return 3;
}
```

Metal C

```
xlc -qmetal -qrent -qipa -qargparse -c a.c b.c
xlc -qmetal -S -qrent -qipa -qargparse a.o b.o
```

```
L      0,@4argc      000007
      AHI      0,5      000007
      LTR      0,0      000007
      BRNH     @1L9     000007
      LR       15,0     000008
      L        1,=Q(MAIN#S) 000000
      DROP     1        000000
      MHI      0,3      000009
      AL       15,0(1,14) 000008
      AL       0,4(1,14) 000009
      ST       15,0(1,14) 000008
      ST       0,4(1,14) 000009
      ALR      15,0     000011
      BRU      @1L6     000000
```

Optimized Assembly

Benefits of Metal C

- Provides high level language alternative to writing programs in assembly language.
- Allows developers to use C syntax to develop system programs
 - Significantly shortens the learning curve
 - No need to manage use of registers and developing the correct assembly instruction sequences
- Allows developers to develop low level free standing applications on z/OS
 - Supports embedding of HLASM source within C statements
 - Provide direct access to z/OS System services
- Leverage advanced optimization technology to generate high performance optimized code
- Develop the application once
 - Recompile to optimize for new z/Architecture

Summary

- IBM Compilers provides timely z/Architecture exploitation and advanced optimization to improve application performance
 - COBOL, PL/I, C/C++
- System z compilers provides modernization features to help extend business critical System of Record applications to work with new System of Engagement infrastructures
 - Java, XML, JSON
- New XL C/C++ for Linux on z Systems enables leading edge optimization and exploitation technology on to Linux environment
- Stay Current...
 - You cannot maximize your return on investment unless you stay current with compiler technology

References

Enterprise COBOL for z/OS Product Page

<http://www.ibm.com/software/products/en/entecoboforzos>

Enterprise COBOL Developer Trial

<http://www-03.ibm.com/software/products/en/enterprise-cobol-developer-trial-for-zos>

Enterprise PL/I for z/OS Product Page

<http://www.ibm.com/software/products/en/plizos>

z/OS XL C/C++ Product Page

<http://www.ibm.com/software/products/en/czos>

XL C/C++ for Linux on z Systems Product Page

<http://www.ibm.com/software/products/en/czlinux>

Link to z Systems Compilers White Paper

<http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=ZSW03065USEN&attachment=ZSW03065USEN.PDF#loaded>



QUESTIONS

The word "QUESTIONS" is written in large, white, 3D-style letters. Each letter is filled with a different portrait of a diverse group of business professionals. The portraits include a man in a light blue shirt, a man in a grey suit and tie, a woman in a white top, a man in a blue suit and tie, a woman in a white top, a man in a pink shirt and tie, a woman in a purple top, a woman in a green top, and a man in a white shirt and patterned tie. The letters have a slight shadow and are set against a white background.



www.ibm.com/software/rational

© Copyright IBM Corporation 2011. 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.