Using the latest compiler technologies on z Systems

Capitalizing on the hardware and software that run your business

Contents

1 Executive summary
2 Introduction
2 Enterprise COBOL for z/OS
5 Enterprise PL/I for z/OS
6 z/OS XL C/C++
8 XL C/C++ for Linux on z Systems
10 IBM Rational Developer for System z
11 Summary
12 For more information

Executive summary

IBM® z Systems™ servers provide a high-performance, flexible, security-rich and reliable platform that helps you meet the IT requirements of today's demanding business environments. Serving as the bridge between user applications, middleware and the increasingly complex hardware architectures that drive business systems, compiler technology is integral to harness advances in z Systems hardware and middleware. By improving the performance of your applications and the productivity of your developers, you can reduce cost, optimize system capacity and improve the quality of deliverables—helping to improve the return on your IT investment.

The value proposition of state-of-the-art compilers is twofold. First, they can help you maximize return on investment for the hardware on which they run. Secondly, they can aid in diagnosing problems, increasing programmer productivity, strengthening support for middleware, and lowering development and maintenance costs. Compilers are often an inexpensive layer in the software stack, yet they are also powerful levers to supercharge the hardware and software on which you run your business.

Although IBM's flexible compiler products for z Systems servers focus on providing performance, they also focus on portability, reliability, and provide a rich set of capabilities to help modernize your business-critical applications and improve programmer productivity. This paper describes the features and benefits of IBM's strategic compilers specifically designed for z Systems: IBM Enterprise COBOL for z/OS®, IBM Enterprise PL/I for z/OS, IBM z/OS XL C/C++, and IBM XL C/C++ for Linux on z Systems. It also highlights the advantages of staying current with the latest compiler technology.

Introduction

IBM offers the following compiler products on z Systems servers:

- Enterprise COBOL for z/OS
- Enterprise PL/I for z/OS
- z/OS XL C/C++
- XL C/C++ for Linux on z Systems

IBM compilers are used to develop and maintain business-critical applications on z Systems servers. State-of-the-art compilation technology enables programmers to take full advantage of new advancements in hardware architecture. The latest Enterprise COBOL for z/OS, Enterprise PL/I for z/OS, z/OS XL C/C++, and XL C/C++ for Linux on z Systems compilers have the ability to exploit the latest IBM z13™ hardware architecture—enabling improved application performance without changing the source code.

Compilers are often an inexpensive layer in the software stack, yet they are also powerful levers to supercharge the hardware and software on which you run your business while reducing your application development and maintenance costs.

The emergence of service-oriented architecture (SOA) enables developers to modernize existing applications and integrate them with new applications running on a web-based infrastructure. Enterprise COBOL and Enterprise PL/I compilers enable developers to extend their business-critical applications to interoperate with Java, process inbound and outbound XML documents and work with web services. In addition, for Enterprise COBOL and Enterprise PL/I compilers, the z/OS Client Web Enablement Toolkit provides the ability to parse, generate, and validate JavaScript Object Notation (JSON) text.

COBOL, PL/I, and C/C++ compilers for z/OS include features to support IBM middleware, IBM CICS®, IBM DB2® and IBM IMS™ software. These compilers provide integrated coprocessor or pre-processor support for embedded CICS and DB2 statements. In addition, COBOL compilers provide support for an integrated IMS SQL coprocessor. With this support, embedded CICS, DB2, and IMS statements can be compiled in a single step. Compilers are also enhanced to support new middleware features to improve programmability, problem determination, and developer productivity.

z/OS XL C/C++ provides an option, METAL C, to support system program development. The METAL C option enables organizations with limited high level assembler (HLASM) skills to develop optimized system programs on z Systems architectures using high level C language syntax. Programmers can also quickly enable system programs written in METAL C to run on new z Systems architectures by recompiling them to optimize for the target system.

XL C/C++ for Linux on z Systems provides a high level of source compatibility with GNU Compiler Collection (GCC) while providing binary coexistence. With XL C/C++ for Linux on z Systems, you can create and port applications for execution on the next generation of IBM systems, supporting selected Linux distributions built on IBM z/Architecture® technology.

Enterprise COBOL for z/OS

In the past decades, COBOL has become a dominant programming language for processing critical business transactions for organizations around the world. COBOL programs are simple, readable, and very maintainable. Today, many business transactions are still processed with COBOL on z Systems servers.

COBOL, PL/I, and C/C++ compilers for z/OS, and C/C++ compiler for Linux on z Systems are enhanced to support z Systems hardware, IBM middleware and application modernization. In addition, the compilers contain rich functionalities to help improve developers' productivity, including tight integration with Rational Developer for System z.

Enterprise COBOL is a premier enterprise class COBOL compiler for IBM z/OS. It delivers innovation for modernizing business-critical applications and programming features to increase programmer productivity, and bolsters the overall benefits of transactional and data systems such as IBM CICS, IMS, and DB2. With advanced compiler support, you can fully benefit from hardware advancements, modernize existing business-critical applications, add modern graphical user interfaces to COBOL applications or extend them to work with web, cloud, or mobile infrastructures. IBM has been delivering COBOL compilers for over 40 years, and uses this experience to consistently deliver compiler innovation and development.

Highlights of Enterprise COBOL for z/OS V5.2 include:

Support for the new IBM z13 hardware

- Delivers z/Architecture exploitation that includes features for the new IBM z13 hardware, taking advantage of the new Vector Facility for certain COBOL statements, and expanding the use of Decimal Floating Point Facility for Packed Decimal calculations.
- Introduces a new ARCH level. With the new ARCH (11) compiler option, the compiler will generate application code that exploits the instructions available with the latest z13 servers.
- Provides enhancements to the OPTIMIZE compiler option, which allows you to choose from no optimization at all, through to comprehensive low-level optimizations and beyond—improving the performance of your COBOL applications.

New, restored, and enhanced compiler options for ease of migration and programmer productivity

- Introduces the RULES option to help identify code that you might want to change to improve your programs, such as opportunities for improved runtime performance, improved use of memory, and conformity to standard COBOL coding practices.
- Introduces the QUALIFY option that allows users to control the behavior of name resolution and enable references to data items that the COBOL Standard would describe as ambiguous and not allow.
- Introduces the VLR option that allows COBOL programmers to indicate the desired behavior for 'wrong length read' conditions, either File Status=4 or File Status=0. File Status=0 is compatible with previous IBM COBOL compilers for ease of migration.
- Restores the XMLPARSE option that allows COBOL programmers to indicate which XML PARSE is used. XMLPARSE(COMPAT) provides compatibility with the Enterprise COBOL V3 parser for ease of migration.
- Introduces new suboptions to the MAP compiler option to allow users to choose hexadecimal or decimal offsets for MAP output in compiler listing. MAP(HEX) provides compatibility with earlier COBOL compilers for ease of migration.
- Introduces the COPYRIGHT option that places a string in the object module for copyright purposes; and introduces the SERVICE option that places a string in the object module for serviceability and debugging purposes.
- Introduces the ZONEDATA option that allows programmers to indicate whether the generated code should tolerate invalid zone bits in zoned decimal (numeric USAGE DISPLAY) data items in comparisons.

New features added from the ISO 2002 COBOL Standard

- Introduces a new format of SORT statement to sort tables in memory directly.
- Provides enhancements to the EXIT statement to provide a structured way to exit without using a GO TO statement.

- Supports the LEADING/TRAILING keywords of the REPLACING phrase of the COPY statement and the REPLACE statement, which allows users to do partial-word replacement on copy files or source code.

New IBM extensions to COBOL

- Supports the >>CALLINTERFACE directive.
- Provides enhancements to the XML GENERATE statement.
- Supports the nested COPY REPLACING statement.
- Supports the VOLATILE clause in a data description entry.

Product-related enhancements

- For COBOL applications in mobile and web solutions, the z/OS Client Web Enablement Toolkit provides JSON document parsing and generation.
- Provides enhancements to Debug Tool, which include showing new compiler options in DESCRIBE CU output and supporting new QUALIFY(EXTEND) name lookup rules in all commands that reference data items such as LIST. The solution also supports the nested COPY REPLACING statement.

Although binary compatibility is the hallmark of z Systems hardware and the IBM z/OS operating environment, if you are using an old compiler with new middleware (CICS, DB2, IMS) or on new z Systems hardware, you are not able to fully benefit from the latest technological advancements in hardware and middleware. Also, your applications and developers are still being limited to features of a COBOL compiler shipped a long time ago. Applications are not able to take advantage of new performance, modernization, debugging, and usability features. Therefore, it is important to stay current with the latest compiler technology even though your COBOL applications still work with the latest middleware and z Systems hardware.

Good reasons to upgrade to Enterprise COBOL for z/OS V5.2:

- Provides significant performance improvements over the GA release of Enterprise COBOL V5.1. Well-structured, compute intensive batch applications running on z13 (compiled with Enterprise COBOL V5.2) have shown CPU time reduction of up to 14 percent over the same applications running on zEC12 (compiled with the GA release of Enterprise COBOL V5.1).
- Improved support for CICS V5 and DB2 V11 software.
- Support for Java interoperability to Java 8 to help you incorporate new, web-based applications as part of your business processes.
- Enhanced usability features to improve developer productivity.
- Ease of migration. Provides a high level of source compatibility, object compatibility, and run time environment compatibility with prior versions of IBM COBOL.
- Improved application development. Provides a set of intrinsic functions including string handling, financial capabilities, statistical functions, and mathematical formulas.
- Support for modern development tools. IBM Rational® Developer for System z supports Enterprise COBOL and helps improve the productivity of COBOL developers.

IBM has been delivering COBOL compilers for over 40 years. Enterprise COBOL for z/OS continues to deliver innovations that help developers improve productivity and organizations modernize their mission critical applications. Enterprise COBOL for z/OS exploits the capabilities of the z/Architecture technology, including the latest IBM z13 hardware.

Enterprise PL/I for z/OS

PL/I is a general-purpose programming language first introduced in the 1960s. It is used significantly in scientific, engineering and business applications. PL/I is an important language for running many business-critical applications in the world today. PL/I is a strategic programming language and IBM will continue to deliver innovations and value to our Enterprise PL/I for z/OS customers.

IBM has been delivering PL/I compilers for more than 45 years. Each release contains significant improvements in optimization technology and exploitation of z Systems architectures. Each release also contains improvements in programmability, new customer requirements, and new application modernization features. Enterprise PL/I for z/OS has consistently delivered innovations to help developers maintain and create mission-critical, line-of-business PL/I applications optimized for deployment on z Systems servers. Like the COBOL compilers, PL/I compilers also focus on delivering new features designed to improve developer productivity and modernization of proven, business critical applications.

In February 2015, IBM introduced a new release of Enterprise PL/I for z/OS. Enterprise PL/I for z/OS V4.5 exploits the z Systems architecture, including the latest z13 hardware. With this latest release, you can take advantage of advancements in optimization technology to reduce MIPS cost for executing your applications and expand your current IT infrastructure by integrating mission-critical PL/I applications with web-based business processes. This new release also provides improved support for IBM middleware subsystems, DB2, CICS and IMS software.

Highlights of Enterprise PL/I for z/OS V4.5 include:

Modernization

- A series of new built-in functions provide the ability to parse, generate, and validate JSON text.

Performance improvements

- The ARCH option now accepts 11 as its maximum value, and when ARCH(11) is specified, the compiler generates code that exploits the new hardware instructions on the supported IBM z Systems servers. It uses the new Vector Facility to improve code for SEARCH and VERIFY functions and expands the use of Decimal Floating Point Facility for Packed Decimal calculations.
- Faster code is generated for MOD and REM of FIXED DEC with precision greater than 15 places.
- Enterprise PL/I for z/OS V4.5 provides significant performance improvements over Enterprise PL/I V4.4. Compute intensive applications running on z13 (compiled with Enterprise PL/I V4.5) have shown CPU time reduction of up to 17 percent over the same applications running on zEC12 (compiled with Enterprise PL/I V4.4)

Improved middleware support

- The code that is generated for each EXEC CICS statement now executes faster because one Move Character (MVC) instruction is eliminated from the code that is generated for each statement.
- SQL enhancements
  - The validation of an EXEC SQL statement will not stop when the first invalid host variable is found, but will instead check all host variable references.
  - A PL/I variable with the VALUE attribute can now be used as a host variable if SQL allows a constant in that setting.
  - The new SQL preprocessor option (NO)CODEPAGE determines how the compiler CODEPAGE option is honored by the SQL preprocessor.
  - The new SQL preprocessor option (NO)WARNDECP allows you to reduce the noise produced by the SQL preprocessor.
  - Structures can now be used as indicator variables for host structure variables.

Productivity and usability improvements

New compiler options make it easier to improve code quality and analyze runtime problems.

- The LIMITS option supports a new suboption that specifies the maximum length allowed when declaring a string variable and allows support for strings greater than 32 KB characters. The maximum length of the string can now be up to 128 MB.
- The new BETWEEN built-in function makes it easier to write code that tests if a variable is in between two specified values. This built-in function also makes it easier for the compiler to generate better code for such tests.
- The new INLIST built-in function makes it easier to write code that tests if a variable has a value in a specified list of values. This built-in function also makes it easier for the compiler to generate better code for such tests.
- The new MAXBRANCH compiler option can help you find excessively complex code.
- The new REINIT statement allows variables to be reset with their INITIAL values.
- Apostrophes are now accepted as insertion characters in picture strings in the same way that the comma, point, and slash have been.
- The new NULLENTRY built-in function makes it easy to assign a null value to an entry variable and to test if an entry variable is null. Under the options INITAUTO, INITBASED, etc., entry variables will now be initialized as well.
- The new PLISTCK, PLISTCKE, and PLISTCKF built-in subroutines generate the corresponding store clock hardware instruction to help you time sections of code more easily. Also, with these new built-in subroutines you can get more precise clock values than those provided by the built-in date-time functions.

PL/I is a strategic programming language for z Systems servers. IBM is committed to PL/I, and each release contains significant improvements in optimization technology to fully exploit z Systems architectures.

It is important to stay current with the latest PL/I compiler technology. Although your PL/I applications continue to work with the latest middleware (CICS, DB2, IMS) and z Systems hardware, you are not maximizing your return on z Systems hardware and middleware investment if you are using an old PL/I compiler. Your applications are not able to fully benefit from the latest technology advancements. Your developers are not able to benefit from new features that would help them improve productivity and deliver innovation because they are still using out-of-date features. Moreover, they are not able to take advantage of new modernization, debugging, optimization, or usability features provided by the new PL/I compiler. And crucially, your applications are not optimized for the hardware they are running on and you are not able to effectively reduce MIPS cost.

z/OS XL C/C++

z/OS XL C/C++ enables you to develop high-performing business applications and system programs on z/OS. It has the capability to fully exploit z Systems architectures, including the latest z13 hardware. It supports both 31-bit and 64-bit application development. It also works in concert with z/OS problem determination tools such as Debug Tool and Fault Analyzer. z/OS XL C/C++ includes an advanced optimizer designed for maximizing performance of high performance computing. This optimizer enables z/OS XL C/C++ to perform advanced loop optimizations, whole program optimizations, profile-based optimizations, and memory hierarchy optimizations.

z/OS XL C/C++ helps you create and maintain critical business applications written in C or C++, maximize application performance, and improve developer productivity. z/OS XL C/C++ can transform C or C++ source code to fully exploit your existing IBM z Systems servers and optimize workloads through smarter computing capabilities with the new IBM z13 hardware. Built-in functions, performance-tuned libraries, and language constructs are some of the features that simplify programming and boost application runtime performance.

z/OS XL C/C++ also enables straightforward porting of C and C++ applications to z Systems servers. It supports international C and C++ programming language standards and extensions (such as GCC extensions). Recently, a number of language features documented in the new C11 and C++11 standards have been delivered in z/OS XL C/C++ V2R1. This allows you to consolidate your C and C++ applications to z Systems servers to take advantage of the improved performance, flexibility, security, and reliability features offered by the platform.

z/OS XL C/C++ supports the DSECT Conversion Utility for converting descriptive data produced by HLASM into C/C++ data structures to enable C/C++ programs to interface with assembler programs. It also supports generation of dynamic link libraries (DLLs) in a way that is similar to shared library support on UNIX and other workstation platforms.

z/OS XL C/C++ includes new options and built-in functions to fully exploit z Systems architectures, including the latest IBM z13 servers.

The METAL C option is designed to simplify system program development on z/OS. Developers can now use C syntax in place of HLASM to develop system programs. This can make the code easier to maintain and evolve to match the changing requirements of your development organization. The METAL C option also helps users to develop freestanding C programs that do not require the IBM z/OS Language Environment®. METAL C programs work with HLASM, support MVS™ (Multiple Virtual Storage) system linkage conventions, and can obtain system services by calling assembler services directly. You can also quickly enable programs written in METAL C to new z Systems architectures. You just need to recompile the source and leverage advanced optimization technology in z/OS XL C/C++ to optimize the code for the target system. This can significantly reduce development cost and delivery time.

z/OS XL C/C++ is a leading-edge compiler that maximizes middleware by providing interoperability with IBM DB2, CICS, and IMS systems. This enables CICS and SQL statements in C and C++ programs to be passed through the compiler without the need for a preprocessing step. This can simplify the operation of C and C++ within the CICS and DB2 environments, improving programming efficiency and problem determination capabilities.

Highlights of z/OS XL C/C++ V2R1M1 web deliverable include:

Exploits IBM z13 hardware through new options and built-in functions

- Introduces the ARCH(11) and Tune(11) options to enable applications to fully exploit the IBM z13 processor.
- 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 a 17 percent throughput improvement over the same applications running on zEC12 (compiled with z/OS XL C/C++ V2R1).

- New vector types have been added to provide easy algorithmic access to the single instruction, multiple data (SIMD) registers and instructions to allow applications to take advantage of the new Vector Facility on IBM z13 processors.
- C and C++ operators have been enhanced to provide natural operations on vector types.
- New built-in functions have been added to allow use of specialized vector instructions for tuned performance.

Improved source compatibility

- Provides runtime architecture blocks to enable a single source file with sections designed to take advantage of various hardware architecture levels.
- Enables inline assembler statements in C and C++ code without compiling with METAL C.

Supports MASS and ATLAS libraries

- Ships with the MASS library specifically tuned for z13 to speed up execution of elementary math functions and as a higher performance alternative to the standard math library.
- Ships with the ATLAS library specifically tuned for z13 to provide high performance linear algebra function support for BLAS and LAPACK functions routinely used in Business Analytics and Optimization solutions.
- Provides a powerful framework for development of new Business Analytics workloads, porting math intensive workloads from other platforms, and accelerating Business Analytics workloads on IBM z13 hardware.

Integrated development and collaboration environment

- Enables easy edit, compile, and debug of z/OS XL C/C++ applications on IBM Rational Developer for System z.
- Enables easy management of distributed software projects and teams.

Improved industry language standards compliance

- Facilitates porting from other platforms to z/OS.
- Provides compiler diagnostics to help achieve level of conformance to a particular programming language standard.
- Supports commonly used IBM and non-IBM language extensions.

It is important to stay current with the latest C/C++ compiler technology and upgrade to the latest release when you upgrade z Systems servers. Although your C and C++ applications continue to work on new z Systems hardware, you will not be able to take full advantage of the latest hardware advancements with a dated level of z/OS and XL C/C++ compiler. As a result, you are not able to optimize your return on your hardware investment. In addition, if you are challenged in system programming resources, the METAL C option might be your solution.

Good reasons to upgrade to z/OS XL C/C++ V2R1M1:

- Improved execution performance of C and C++ code on z Systems servers including the latest z13 hardware.
- Improved industry language standards compliance.
- Capability to support vector processing technologies through language extensions.
- Capability to support high-performance mathematical computing through MASS and ATLAS libraries.
- Improved debugging and programmability features to boost programmer productivity.

XL C/C++ for Linux on z Systems

IBM XL C/C++ for Linux on z Systems, V1.1, which is shipped in February 2015, is a new XL C/C++ compiler for application development that takes advantage of the latest IBM z Systems servers that run on selected Linux distributions. It is the newest member of the IBM XL compiler family and is built on the performance gains from many years of IBM compiler optimization experience with existing XL C/C++ compilers that are available for IBM z/OS, IBM z/VM®, IBM AIX® and Linux on IBM Power Systems™. This XL C/C++ compilation technology for Linux on z Systems strengthens the platform, exploits the z Systems environment, and provides superior performance.

Highlights of XL C/C++ for Linux on z Systems V1.1 include:

Generation of highly optimized code

- Supports generation of highly optimized code exploiting the z Systems servers, adheres to the latest programming standards partially, and provides GNU Compiler Collection (GCC) compatibility that allows you to easily port your applications to selected Linux distributions running on z Systems servers.
- Provides significant performance advantage over GNU C/C++. CPU intensive applications compiled with XL C/C++ for Linux on z Systems V1.1 have shown up to 10 percent performance improvement over the same applications compiled with GNU C/C++ V4.4 and 7 percent over GNU C/C++ V4.7.

For application development, IBM is delivering its first XL C/C++ compiler for Linux on z Systems in 2015. The complier takes advantage of the latest IBM z Systems servers and runs on selected Linux distributions.

The Clang infrastructure

- Leverages the Clang infrastructure from the open source community for a portion of its compiler front end. Clang is a component of the LLVM open source compiler and toolchain project, and provides the C and C++ language family front end for LLVM. XL C/C++ for Linux on z Systems combines the Clang front-end infrastructure with the advanced optimization technology in the IBM compiler back end.

New architecture and tune compiler options for the z Systems technology

- Supports the new generation of z Systems hardware running SUSE Linux Enterprise Server 11 (SLES 11), SLES 12, Red Hat Enterprise Linux 6 (RHEL 6), and RHEL 7.
- The -qarch compiler option specifies the processor architecture for which code is generated. -qarch=arch11 instructs the compiler to produce code that can exploit the new instructions in the z13 processors.
- The -qtune compiler option tunes instruction selection, scheduling, and other architecture-dependent performance enhancements to run best on a specific hardware architecture. -qtune=arch11 enables optimizations specifically for the z13 processors.

Optimization capabilities

- Offers the benefit of optimization technology that decreases execution time, makes your applications run faster, and produces code that is highly tuned for execution on z Systems servers.
- Includes three base optimization levels, which allow you to choose from minimal optimization to intense program analysis that range from local basic block to subprogram unit scopes, file-level, and whole-program analysis. Levels of optimization include:
  - -O0: Provides minimal optimization, which is best for debugging.
  - -O2: Provides strong low-level optimization and benefits most programs.
  - -O3: Provides intense low-level optimization analysis and base-level loop analysis.
- Supports sophisticated optimization techniques such as interprocedural analysis. -qipa can be used for aggressive optimization of the whole program, including aggressive data flow analysis and loop transformations.

Programming language standards

- Supports C and C++ programming language standards, including partial support for the latest C11 and C ++11 standards.

GNU compatibility

- Provides a greater level of GNU source compatibility that helps to port applications that are originally developed with GCC and g++ compilers, while keeping binary compatibility with GNU-built objects, archives, and shared objects.

Automatically Tuned Linear Algebra Software (ATLAS)

- Includes the ATLAS libraries that consist of numerical linear algebra operations. The ATLAS libraries are specifically tuned for optimum performance on the z Systems architecture.

IBM Mathematical Acceleration Subsystem (MASS)

- Includes the IBM MASS libraries that consist of mathematical intrinsic functions. The MASS libraries are specifically tuned for optimum performance on the z Systems architecture.

Basic Linear Algebra Subprograms (BLAS)

- Includes BLAS that provides high-performance, algebraic functions that are used to perform combined matrix multiplication and addition on general matrices or their transposes, as well as to compute the matrix-vector product for a general matrix or its transpose.

Support for z/OS Language Environment

The z/OS Language Environment provides a common foundation to run programs written using different programming languages. It was architected to facilitate interoperation between different programming languages. It provides a consistent means for developing quality applications with multiple languages while maintaining existing applications. Programs developed using Enterprise COBOL for z/OS, Enterprise PL/I for z/OS, and z/OS XL C/C++ compilers can interoperate with one another as well as with programs written in HLASM. This capability enables you to take advantage of the strengths of different programming languages within a single application.

Applications developed with Enterprise COBOL, Enterprise PL/I, and z/OS XL C/C++ compilers and IBM High Level Assembler (HLASM) are fully interoperable. This capability offers users the flexibility to take advantage of the strengths of different programming languages within a single application.

IBM Rational Developer for System z

IBM Rational Developer for System z software consists of a common development workbench and an integrated set of tools that support model-based development, run-time testing, and rapid deployment of applications. It offers an integrated development environment with advanced, easy-to-use tools to help accelerate development of your web applications, traditional COBOL and PL/I applications, web services and XML-based interfaces for z Systems servers. Rational Developer for System z software is designed to work with Enterprise COBOL for z/OS, Enterprise PL/I for z/OS, and z/OS XL C/C++ compilers. It can improve productivity of experienced developers on z Systems servers, and it can also help attract new hires to work on your business critical applications and lower training costs.

Highlights of IBM Rational Developer for System z include:

Modern, simple-to-use application development tools

- Supports the design, creation, and maintenance of mainframe applications.
- Protects investment in mainframe applications by making them accessible as web services in a service-oriented architecture (SOA) environment.

Fully integrated source level debugger

- Provides debug support for COBOL, PL/1, C/C++, batch IMS and IMS TM, CICS, batch DB2, and DB2 SP. Uses IBM cross-language and cross-platform debug architecture.
- Enables users to step through the z/OS application, monitor and modify variables, set breakpoints during or before a debug session, and inspect program memory.

Optimized tooling support

- Offers CICS, IMS, and DB2 development tools for CICS Transaction Server, IBM CICS Explorer®, IMS Transaction Manager and Database, DB2 databases and stored procedures.

Source control and collaborative lifecycle management systems integration

- Allows z/OS developers to work with Rational Team Concert™ on z Systems servers for an enhanced development environment with collaborative capabilities.

Enhanced application structural and quality analysis tools

- Provides application structural analysis, code analysis, and code metrics to help deliver COBOL and PL/I applications that conform to in-house programming guidelines.
- Provides an expanded set of COBOL and PL/I code quality rules and enablement of custom rules.

Summary

Organizations often skip upgrading their compilers when they upgrade their z Systems hardware and middleware stack. As a result, they are running new high-powered z Systems servers with outdated compilers and the latest middleware, which severely limits the servers' potential and can translate to greater power consumption, slower response times, administrative challenges, and higher MIPS costs.

IBM compilers on z Systems servers are designed to enable applications to take full advantage of the hardware features provided by IBM z/Architecture technology and the latest middleware subsystems. By using optimization technologies and modernization features in IBM compilers, you can improve your return on investment from capital investments in hardware and increase programmer productivity. MIPS used by compilers to optimize code can easily be offset by the savings from running more efficient applications on z Systems servers.

Long-running applications are often the main consumers of hardware resources, and improving their efficiency will result in a significant reduction of the overall operation cost. Also, upgrading compilers when you upgrade middleware or hardware reduces the need for multiple migration and quality assurance efforts. You can take advantage of new capabilities to improve application performance, and you can also improve middleware programmability and programmer productivity. This is perhaps the best way to optimize your return on investment.

IBM has a long history of delivering innovative, high-quality compiler products on z Systems servers. Combined with premium services and application development tools, IBM provides a total solution to enable you to develop, modernize, and maintain your business-critical applications.

For more information

To learn more about how IBM can help you increase your system performance with upgraded compilers, contact your IBM sales representative or IBM Business Partner.

For more information about the IBM family of COBOL compilers, visit: **ibm.com**/software/products/en/entecoboforzos

For information about the IBM family of PL/I compilers, visit: **ibm.com**/software/products/en/plizos

For more information about the IBM family of z/OS XL C/C++ compilers, visit: **ibm.com**/software/products/en/czos

For more information about the XL C/C++ for Linux on z Systems compiler, visit: **ibm.com**/software/products/en/czlinux

For more information about Rational Developer for System z, visit: **ibm.com**/software/products/en/developerforsystemz

For more information about the Rational software communities, visit: **ibm.com**/developerworks/rational/community.html

Additionally, IBM Global Financing can help you acquire the IT solutions that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize an IT financing solution to suit your business goals, enable effective cash management, and improve your total cost of ownership. IBM Global Financing is your smartest choice to fund critical IT investments and propel your business forward. For more information, visit: **ibm.com**/financing

Route 100
Somers, NY 10589

Produced in the United States of America
March 2015

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

♻ Please Recycle