IBM Enterprise COBOL for z/OS, Version 5 Release 1 Modification 1



Enable your COBOL applications to exploit the latest z/Architecture

Enterprise COBOL is a premier enterprise class COBOL compiler for the $z/OS^{\$}$ system. It delivers innovation for modernizing business-critical applications, programming features to increase programmer productivity, and bolsters the overall benefits of transactional and data systems such as CICS $^{\$}$, IMS $^{\mathsf{TM}}$, and DB2 $^{\$}$.

Enterprise COBOL for z/OS, V5.1.1 delivers advanced compiler support to allow you to fully benefit from hardware advancements. The Enterprise COBOL for z/OS compiler is now capable of unleashing the full power of IBM® processors delivered in the various models of System z® hardware. Developers only need to focus on the logic of the applications and let the compiler determine the best way to transform and optimize the code generation for the System z hardware on which the application will run.

With its enhanced capabilities, simplified programming, and increased programmer productivity, you can continue to use Enterprise COBOL for z/OS to modernize existing business-critical applications. You can deliver new enhancements quicker and with less cost and lower risk. You can add modern graphical user interfaces to business-critical COBOL applications or extend them to work with web, cloud, or mobile infrastructures. With the investment in new compiler technology and the continued delivery of new features, Enterprise COBOL for z/OS, V5.1.1 reaffirms IBM's commitment to COBOL on z/OS. You gain the benefit of new investment combined with more than 40 years of IBM experience in compiler development.

Highlights

Enterprise COBOL for z/OS, V5.1.1 is not a refresh of the product that can be ordered. You must order Enterprise COBOL for z/OS, V5.1.0 and then obtain and apply PTFs for APAR PM93583. The PTFs will update the compiler and run time for the features listed here and change the modification level of the compiler from 0 to 1 to make Enterprise COBOL for z/OS, V5.1.1.

- Supports AMODE 24 execution of COBOL programs
- Offers a new IMS SQL coprocessor to handle source programs that contain embedded SQLIMS statements
- Exploits the latest z/Architecture[®] and performance optimization
- Delivers XML processing enhancements for easier web interoperability
- Improves capability for programming with UTF-8 Unicode for better national language support in your applications

- Increases compiler limits to help you handle larger data items and larger groups of data and to improve application exploitation of system resources
- Supports unbounded tables and groups to improve usability in defining variable length tables and groups
- Introduces the floating comment indicator to create a comment anywhere in the program-text area
- Provides a new level of z/OS System
 Management Facilities (SMF) tracking support to reduce your administrative reporting overhead
- Supports Java[™] 7 to help you incorporate new, web-based applications as part of your business processes
- Exploits a new interface called Common Debug Architecture (CDA) to provide a consistent format for accessing information that can be used by debuggers and program analysis tools

Supports AMODE 24 execution of COBOL programs

AMODE 24 execution of COBOL programs is supported with APAR PM93583. Programs compiled by Enterprise COBOL V5.1.1 will execute in AMODE 31 or AMODE 24. However, there are some cases where AMODE 24 execution is not supported. These limitations are now common between Enterprise COBOL V4 and Enterprise COBOL V5.1.1.

Offers a new IMS SQL coprocessor to handle source programs that contain embedded SQLIMS statements

Use the SQLIMS compiler option to enable the IMS SQL coprocessor and to specify Information Management System (IMS) suboptions. You must specify the SQLIMS option if a COBOL source program contains SQLIMS statements (EXEC SQLIMS statements).

Exploits the latest z/Architecture and performance optimization

With IBM Enterprise COBOL for z/OS V5.1.1, you can maximize the delivery of z/Architecture exploitation and performance optimization within your applications. Based on new code generation and optimization technology, Enterprise COBOL for z/OS V5.1 introduces the new ARCH and enhanced OPTIMIZE compiler option:

- The ARCH(10) option produces code that uses instructions available on the IBM zEnterprise[®] BC12 and zEnterprise EC12 servers. Specifically, these ARCH(10) machines and their follow-ons add instructions supported by the following facilities:
 - Execution-hint facility
 - Load-and-trap facility
 - Miscellaneous-instructions-extension facility
 - Transactional-execution facility
- Using the OPTIMIZE(0), OPTIMIZE(1), and OPTIMIZE(2) options, you can obtain multiple levels of increasing optimization that run from comprehensive low-level optimizations to more extensive optimizations that can improve the performance of your COBOL applications. In addition, when you specify the TEST option,

you can enable debugging capabilities with minimal impact to application runtime performance.

Delivers XML processing enhancements for easier web interoperability

In Enterprise COBOL for z/OS, V5.1, XML parsing and XML generation have been enhanced:

- The new XML-INFORMATION special register determines whether an XML EVENT and its associated XML-INFORMATION value are complete.
- The XML GENERATE statement is extended with the following phrases:
 - The NAME phrase allows user-supplied element and attribute names.
 - The SUPPRESS phrase allows suppression of empty attributes and elements.
 - The TYPE phrase gives the user control of attribute and element generation.

Improves capability for programming with UTF-8 Unicode for better national language support in your applications

In Enterprise COBOL for z/OS, V5.1, new intrinsic functions provide additional Unicode capability while improving usability for programming with UTF-8 Unicode and complementing the support for UTF-16 Unicode:

- The ULENGTH function returns an integer equal to the length, in UTF-8 characters, of a character string argument that is encoded in UTF-8.
- The USUBSTR function returns a substring of a character string argument that is encoded in UTF-8.
- The USUPPLEMENTARY function returns an integer equal to the index of the first Unicode supplementary character in a character string argument that is encoded in UTF-8 or UTF-16.
- The UVALID function returns an integer which has the value zero if a character string contains valid Unicode UTF-8 or UTF-16 data, and which has the index of the first invalid element if the character string does not contain valid Unicode data.

• The UWIDTH function returns an integer equal to the width in bytes of the nth UTF-8 character in a character string argument that is encoded in UTF-8.

Increases compiler limits to help you handle larger data items and larger groups of data and to improve application exploitation of system resources

In Enterprise COBOL for z/OS V5.1, a number of limits imposed by the compiler have been raised, to improve usability and enable COBOL applications to further exploit available processor storage resources.

In previous versions, the limit for each data division section and the limit for an individual data item is 128 MB. In Enterprise COBOL V5.1, there is no aggregate data division limit and the limit for an individual data item is 999,999,999 bytes. With the increase of limits on data item, table and table element, linkage section and data division sizes, you can use Enterprise COBOL for z/OS, V5.1 to work with even larger volumes of data while employing business analytics to uncover opportunities, build efficiencies, and make more informed decisions.

Supports unbounded tables and groups to improve usability in defining variable length tables and groups

In Enterprise COBOL for z/OS, V5.1, a new keyword UNBOUNDED is added to the OCCURS DEPENDING ON clause which enables you to define unbounded tables and groups.

With this new keyword, you no longer need to specify the maximum size for a table and the containing group. Usability in defining variable length tables in COBOL is improved particularly when you define COBOL group structures that correspond to XML documents.

Introduces the floating comment indicator to create a comment anywhere in the program-text area

Enterprise COBOL for z/OS, V5.1 introduces the floating comment indicator ('*>'). You can specify it anywhere in the program-text area to indicate that the ensuing text on a line is a comment line or an inline comment.

A floating comment indicator indicates a comment line if it is the first character string in the program-text area (Area A plus Area B, columns 8 - 72), or indicates an inline comment if it is after one or more character strings in the program-text area.

Provides a new level of z/OS System Management Facilities (SMF) tracking support to reduce your administrative reporting overhead

Enterprise COBOL V5.1 can now automatically be tracked by SMF89 records and will be supported by Subcapacity Reporting Tool (SCRT) V21.2.0. With this new level of SMF tracking support, you can reduce administrative reporting overhead if you are implementing subcapacity tracking while gaining the benefits of implementing subcapacity for Enterprise COBOL for z/OS, V5.1.

In conjunction with the SMF record support, system administrators can also define a disablement policy through the SYSx.PARMLIB(IFAPRDxx) parameter library. You can use this feature to disable the use of the Enterprise COBOL for z/OS, V5.1 compiler within a specific z/OS system.

Supports Java 7 to help you incorporate new, web-based applications as part of your business processes

In Enterprise COBOL for z/OS, Version 5.1, Enterprise COBOL applications using object-oriented syntax for Java interoperability can now run with Java 7. Java SDK 1.4.2 continues to be supported. Enterprise COBOL supports object-oriented syntax to facilitate the

interoperability of COBOL and Java. This support is based on the facilities of the Java Native Interface (JNI), the primary means that Java provides for interoperability with non-Java programs. Object-oriented COBOL is designed to enable you to:

- Define classes, with methods and data implemented in COBOL
- Define classes that inherit from Java classes or from other COBOL classes
- Create object instances of Java or COBOL classes
- Invoke methods on Java or COBOL objects
- · Define and invoke overloaded methods

Exploits a new interface called Common Debug Architecture (CDA) to provide a consistent format for accessing information that can be used by debuggers and program analysis tools

In Enterprise COBOL for z/OS, V5.1, a new DWARF/Common Debug Architecture interface is provided. CDA works towards a common debug information format across various programming

languages. The debugging information format is open-ended, allowing for the addition of debugging information that accommodates new languages and debugger capabilities while facilitating the porting of debug and analysis tools to z/OS from other DWARF-compliant platforms.

To enable the CDA support, when you compile your application with the TEST option, Enterprise COBOL V5.1 creates and stores the debug information in the program object. This information is placed in the object as NOLOAD class segments, so that the debugging information does not take memory for typical executions. This means that you can compile with TEST, you will always have the correct debugging information available, and yet you can still run these applications in production because the LOAD footprint size is not increased!

Other Enterprise COBOL for z/OS features

Improved application development

Enterprise COBOL for z/OS provides a set of intrinsic functions including string handling, financial capabilities, statistical functions, and mathematical formulas. You can also use the COBOL CALL statement to take advantage of Language Environment® services for everything from storage management to condition handling. The condition handling support enables you to write programs in which exception handling is done in a separate routine that is loaded only when needed. Using Language Environment condition handling, you do not have to write the exception-handling routines in assembler — you can write them in COBOL! Enterprise COBOL for z/OS offers support for recursive calls, structured programming, improved interoperability with other languages, and dynamic link library (DLL) support. The Enterprise COBOL for z/OS runtime library, Language Environment, also supports PL/I, C/C++, and Fortran programs.

Ease into migration

Enterprise COBOL for z/OS gives you a migration path from OS/VS COBOL, VS COBOL II, IBM COBOL for $MVS^{\mathbb{M}}$ & VM, and IBM COBOL for OS/390 $^{\mathbb{G}}$ & VM. With the exception of OS/VS COBOL programs and VS COBOL II NORES programs and any programs that were previously compiled with the CMPR2 compiler option, your current programs can continue to compile and run without modification, while you selectively update existing applications to take advantage of new functions.

You can convert OS/VS COBOL programs and programs compiled with the CMPR2 compiler option into 1985 Standard programs, which can then be compiled using Enterprise COBOL for z/OS. Use the COBOL conversion tool (CCCA) included in Debug Tool Utilities and Advanced Functions for this purpose. Debug Tool Utilities and Advanced Functions also includes a load module analyzer that can help identify which of your programs were compiled with the OS/VS compiler.

Support for modern development tools

Rational[®] Developer for System z supports Enterprise COBOL and helps improve the productivity of COBOL developers. Rational Developer for System z provides an interactive, workstation-based environment to help you create, maintain, and reuse applications. Rational Developer for System z includes support for traditional development using COBOL, but also has the ability to generate Web services interfaces from COBOL constructs to ease creation of Web services from existing COBOL applications.

Rational Developer for System z provides a workstation interface to Debug Tool, and is also integrated with IBM File Manager and Fault Analyzer. File Manager integration enables you to access Keyed Sequence Data Set (KSDS) files from the Rational Developer for System z workbench, and gives you the ability to browse and update data sets. By integrating with Fault Analyzer, Rational Developer for System z enables you to browse Fault Analyzer ABEND reports on CICS, IMS, batch, Java, WebSphere®, and other run times.

COBOL across platforms

Enterprise COBOL for z/OS is part of a family of compatible compilers, application development tools, and maintenance tools.

Software requirements

The following table presents the system requirements for Enterprise COBOL for z/OS V5.1.

Operating system	Software	Hardware
z/OS	Enterprise COBOL for z/OS, V5.1 runs under the control of, or in conjunction with, the currently supported releases of the following programs and their subsequent releases or their equivalents. For more information about the following programs that require program temporary fixes (PTFs), instead of reviewing the PSP Bucket, it is recommended you use the SMP/E V3R6 or SMP/E V3R5 product support for FIXCAT HOLDDATA to verify programmatic target system PTFs. In Enhanced HOLDDATA, these PTFs are identified with a FIXCAT called IBM.TargetSystem-RequiredService.Enterprise-COBOL.V5R1. A HOLDDATA type FIXCAT (fix category) is used to associate an APAR to a particular category of fix for necessary target system PTFs. To help identify the uninstalled PTFs on your current system for your upgrade to IBM Enterprise COBOL for z/OS, V5.1.0, use the SMP/E REPORT MISSINGFIX command. Use the FIXCAT(IBM.TargetSystem-RequiredService.Enterprise-COBOL.V5R1) operand on the APPLY CHECK command. Required licensed programs z/OS V1.13 (5694-A01), or later	Enterprise COBOL for z/OS, V5.1 will run on the following IBM servers: • zEnterprise BC12 or zEnterprise EC12 • zEnterprise 196 • zEnterprise 114 • z10™ Enterprise Class • IBM System z9® Enterprise Class or z9 Business Class • zSeries z990 • zSeries z890

Depending on the functions used, one or more of the following programs might be required:

- CICS Transaction Server for z/OS, V5 (5655-Y04)
- CICS Transaction Server for z/OS, V4 (5655-S97)
- CICS Transaction Server for z/OS, V3 (5655-M15)
- IBM DB2 10 for z/OS (5605-DB2)
- IBM DB2 10 for z/OS Value Unit Edition (5697-P31)
- IBM DB2 V9 for z/OS (5635-DB2)
- IBM DB2 V9 for z/OS Value Unit Edition (5697-P12)
- IBM DFSORT element of z/OS (5694-A01)
- IBM High Level Assembler/MVS and VM and VSE (5696-234)
- IBM IMS V13 (5635-A04)
- IBM IMS V12 (5635-A03)
- IBM IMS V11 (5635-A02)
- IBM 31-bit SDK for z/OS, Java Technology Edition, V7.0 (5655-W43)
- IBM 31-bit SDK for z/OS, Java Technology Edition, V6.0 (5655-R31)
- IBM 31-bit SDK for z/OS, Java 2 Technology Edition, V5.0 (5655-N98)

- IBM Debug Tool for z/OS, V12.1 (5655-W70)
- IBM Fault Analyzer for z/OS V12.1 (5655-W69)
- IBM File Manager for z/OS V12.1 (5655-W68)
- IBM Application Performance Analyzer for z/OS V12.1 (5655-W71)
- IBM Rational® Developer for System z, V9 (5724-T07)
- COBOL Report Writer Release 4 (5798-DYR, 5798-DZX)
- Enterprise COBOL for z/OS, V4 (5655-S71)
- Enterprise COBOL for z/OS and OS/390, V3 (5655-G53)
- Enterprise PL/I for z/OS, V4 (5655-W67)
- Enterprise PL/I for z/OS, V3 (5655-H31)
- For XL C/C++ with Enterprise COBOL -- You must use the XL C/C++ feature of z/OS V1.13 (5694-A01), or later
- IBM VS FORTRAN V2 (5668-806, 5688-087)

Upgrade to Enterprise COBOL for z/OS V5.1.1

Upgrade to the latest Enterprise COBOL compiler and get more out of your zEnterprise investment and stay ahead of competitors on the technology curve.

For more information

To learn more about IBM Enterprise COBOL for z/OS V5.1.1, contact your IBM representative or IBM Business Partner, or visit: Enterprise COBOL for z/OS at www.ibm.com/software/products/us/en/entecoboforzos/

© Copyright IBM Corporation 2014.

IBM Corporation Software Group Route 100 Somers, NY 10589 U.S.A.

Produced in the United States of America March 2014 All Rights Reserved

IBM, the IBM logo, ibm.com[®], CICS, DB2, IMS, Rational, System z, z/Architecture, z/OS, and zEnterprise are trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ([®] or [™]), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

References in this document to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information provided in this document is distributed "as is" without any warranty, either express or implied. IBM expressly disclaims any warranties of merchantability, fitness for a particular purpose or non-infringement. IBM products are warranted according to the terms and conditions of the agreements (e.g. IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided.