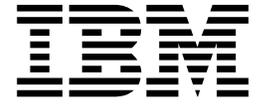


Enterprise COBOL for z/OS



Data Sheet

Version 6.2

Enterprise COBOL for z/OS



Data Sheet

Version 6.2

March 2019

This edition applies to Version 6 Release 2 of IBM Enterprise COBOL for z/OS (program number 5655-EC6) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure that you are using the correct edition for the level of the product.

You can view or download softcopy publications free of charge in the Enterprise COBOL for z/OS library. Because Enterprise COBOL for z/OS supports the continuous delivery (CD) model and publications are updated to document the features delivered under the CD model, it is a good idea to check for updates once every two months.

© **Copyright IBM Corporation 1991, 2019.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Summary of changes

This section lists the key changes that have been made to this document since Enterprise COBOL for z/OS® Version 6 Release 2 was released in September 2017.

March 2019

- PH05855: Add a new INITIAL compiler option that allows you to get a program that has initial values in data items each time the program is called, without having to add the IS INITIAL clause to the PROGRAM-ID paragraph, and without having to use dynamic CALL and CANCEL statements. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)
- PH08642: Redundant checks previously added by the NUMCHECK option have been removed, improving performance, and some checks can be done at compile time. Specifying NUMCHECK may also cause the compiler to produce some messages at compile time instead of at runtime. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)
- PH09225: The INITCHECK compiler option can now be specified with OPTIMIZE(0). (Compiler options changes in Enterprise COBOL V6.2 via the service stream)

November 2018

- PH04369: RULES(NOEVENTPACK) will not issue messages for even-digit PACKED-DECIMAL data items whose names start with DFH, DSN, EYU or SQL, that is, data items generated for/by CICS® and Db2®. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)
- PH04485: New suboptions DSNAME | NODSNAME are added to the TEST | NOTEST(SEPARATE) option to control whether the external file name, which is the SYSDEBUG dataset name used during compilation, will or will not be stored in the object program. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)

September 2018

- PH02183: Add new intrinsic functions of ABS, BIT-TO-CHAR, BYTE-LENGTH, EXP, EXP10, HEX-TO-CHAR, NUMVAL-F, SIGN, TEST-NUMVAL, TEST-NUMVAL-C, and TEST-NUMVAL-F. (“Improved application development” on page 7)
- PH02251: Add support for the NAME IS OMITTED phrase of the JSON PARSE statement. (“New IBM extension features” on page 4)

July 2018

- PI98480: Add two new suboptions ALPHNUM | NOALPHNUM to the NUMCHECK(ZON) option. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)
- PI99703: Add new intrinsic functions of BIT-OF, E, HEX-OF, PI, and TRIM. (“Improved application development” on page 7)

May 2018

- PI95081: Add a new LOC(24 | 31) phrase to the ALLOCATE statement. (“New IBM extension features” on page 4)

- PI96135: Enhance NUMCHECK(PAC) to check for left-most nibble not zero in even length packed decimal field. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)
- PI97160: Add support for the WHEN SET TO FALSE phrase of the VALUE clause and the SET TO FALSE statement. (“New features added from the COBOL 2002/2014 standards” on page 3)
- PI97434: Add support for processing national data items with the following intrinsic functions: REVERSE, ULENGTH, UPOS, USUBSTR, and UWIDTH. (“Improved application development” on page 7)
- PI97621: Add new installation options of ALLOWCOPYLOC and ALLOWDEFINE. (Compiler options changes in Enterprise COBOL V6.2 via the service stream)

January 2018

- PI90571: Enhance the ZONEDATA compiler option to support additional COBOL V4 compatibility for invalid data.
- PI91584: Add a new COPYLOC compiler option.
- PI91585: Add new suboptions OMITODOMIN | NOOMITODOMIN to the RULES option.
- PI91586: Add new suboptions UNREF | NOUNREFALL | NOUNREFSOURCE to the RULES option.

(Compiler options changes in Enterprise COBOL V6.2 via the service stream)

Contents

Summary of changes	iii
Chapter 1. Enable your COBOL applications to exploit the latest z/Architecture	1
Chapter 2. Highlights	3
Chapter 3. Other Enterprise COBOL for z/OS features	7
Chapter 4. System requirements.	11
Chapter 5. Upgrade to Enterprise COBOL for z/OS V6.2	13
Chapter 6. For more information.	15
Chapter 7. Notices.	17
Trademarks	17

Chapter 1. Enable your COBOL applications to exploit the latest z/Architecture

Enterprise COBOL is a premier enterprise class COBOL compiler for IBM® z/OS. 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 IBM CICS, IMS, and Db2®.

Enterprise COBOL for z/OS, V6.2 delivers advanced compiler support to allow you to fully benefit from hardware advancements. The Enterprise COBOL for z/OS compiler is capable of unleashing the full power of IBM processors delivered in the various models of IBM 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 IBM Z hardware on which the application will run.

With its enhanced capabilities, simplified programming, and increased programmer productivity features, you can use Enterprise COBOL for z/OS to modernize existing business-critical applications. You can deliver new enhancements quicker, with less cost and with lower risks. 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, V6.2 reaffirms IBM's commitment to COBOL on z/OS. You gain the benefit of new investments combined with more than 50 years of IBM experience in compiler innovation and development.

Chapter 2. Highlights

Enterprise COBOL for z/OS, V6.2 delivers the following new and improved features:

- Support for the new IBM z14™ hardware and IBM z/OS V2.3 operating system
- New features added from the COBOL 2002/2014 standards
- New IBM extension features
- New, replaced, and enhanced compiler options for ease of migration and programmer productivity
- Usability enhancements in the z/OS UNIX System Services environment
- Improved interfaces to other licensed programs and tools
- Compile-time and runtime performance improvements

Support for the new IBM z14 hardware and IBM z/OS V2.3 operating system

IBM Enterprise COBOL for z/OS, V6.2 delivers z/Architecture® exploitation, including a new ARCH(12) level. You can now build and run COBOL applications on the latest IBM z14 hardware and IBM z/OS V2.3 operating system.

With the new ARCH(12) compiler option, the compiler generates application code that exploits the instructions available with the latest z14 server. Specifying ARCH(12) instructs the compiler to include exploitation of the new Vector Packed Decimal Facility, which allows the dominant COBOL data types, packed and zoned decimal, to be handled in wide 16-byte vector registers instead of in memory. This translates into improvements in COBOL computational performance. You can recompile with ARCH(12) to target z14 without any source changes to take advantage of this new facility.

Decimal and Floating Point computationally intensive COBOL programs, which are optimized with Enterprise COBOL V6.2 and that target z14 ARCH(12), can deliver CPU time reduction on the z14 server over the same applications built with COBOL V6.1 on the z13® server.

New features added from the COBOL 2002/2014 standards

Enterprise COBOL V6.2 adds support for the COBOL 2002/2014 Conditional Compilation language feature with the introduction of the DEFINE compiler option that allows you to define or set conditional compilation constants at compile time, and also the >>DEFINE, >>IF, and >>EVALUATE compiler directives that are used for conditional compilation. Conditional compilation provides a way of including or omitting selected lines or source code (or copybooks) depending on the values of literals or arithmetic expressions that are specified by the DEFINE directive. In this way, you can create multiple variants of the same program without the need to maintain separate source streams.

Enterprise COBOL V6.2 with PTF for APAR PI97160 installed adds support for the TO FALSE phrase of the SET statement, as well as the corresponding WHEN SET TO FALSE clause, which allow you to use the SET statement to set condition names to values that will test false in conditions.

New IBM extension features

Enterprise COBOL V6.2 adds support for the JSON PARSE statement. This completes the support for JSON (Java™ Script Object Notation) that was started in Enterprise COBOL V6.1 with the addition of JSON GENERATE statement. The JSON PARSE statement enables capability to parse JSON documents directly from COBOL by converting JSON text to COBOL data formats, and extending business critical (System of Record) applications that are written in COBOL to support mobile (Systems of Engagement) applications where the preference is REST/JSON. This support also complements z/OS Connect by supporting the 'pass-through service' where JSON texts are passed directly to COBOL applications that parse the JSON text and process the transaction.

Enterprise COBOL V6.2 with PTF for APAR PI95081 installed adds a new LOC(24|31) phrase to the ALLOCATE statement to control the location of dynamic storage that is acquired, which overrides the influence of the DATA compiler option.

Enterprise COBOL V6.2 with PTF for APAR PH02251 installed adds support for the NAME IS OMITTED phrase of the JSON PARSE statement. In this way, you can parse an anonymous JSON object, whose top parent name is not specified.

Note: If the new JSON PARSE keyword OMITTED in the September compiler PTFs (UI58632, UI58633, UI58634) is used, then the September Runtime PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied on all systems where these programs are linked or run.

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

- The IBM-supplied default for the AFP compiler option is changed from VOLATILE to NOVOLATILE, so that the compiler can generate more efficient code sequences for programs with floating point operations.
- The new DEFINE option allows you to define or set conditional compilation constants at compile time.
- The new INITCHECK option tells the compiler to perform a static analysis of the program, and to emit a warning message for data items that are used before they are initialized.
- Two new features are added to control procedure inlining behaviors at OPTIMIZE(1) or OPTIMIZE(2):
 - The new INLINE option controls whether inlining of procedures (paragraphs or sections) that are referenced by PERFORM statements in a program is to be allowed or disallowed.
 - The >>INLINE ON and >>INLINE OFF compiler directives can be used to disable specific paragraphs within the source from being inlined.
- The new NUMCHECK option tells the compiler whether to generate extra code to validate data items when they are used as sending data items. For zoned decimal (numeric USAGE DISPLAY) and packed decimal (COMP-3) data items, the compiler generates implicit numeric class tests for each sending field. For binary data items, the compiler generates SIZE ERROR checking to determine whether the data item has more digits than its PICTURE clause allows.

Note: The ZONECHECK option is deprecated but is tolerated for compatibility, and it is replaced by NUMCHECK(ZON).

- The new PARMCHECK option detects if a subprogram writes beyond the end of the WORKING-STORAGE section. This option tells the compiler to generate an extra data item following the last item in WORKING-STORAGE that is then used at run time to check whether a called subprogram corrupted data beyond the end of WORKING-STORAGE.
- New SSRANGE suboptions MSG|ABD and ZLEN|NOZLEN allow, respectively:
 - A message instead of an abend and continued processing for additional reporting of out-of-range conditions in a single run.
 - A reference modification of zero length to proceed without a message or abend.
- New combinations of suboptions are supported in both the TEST and NOTEST compiler options, including TEST(NODWARF), TEST (SEPARATE), and NOTEST(DWARF, SOURCE).
- The following compiler options or suboptions are added or updated in Enterprise COBOL V6.2 via the service stream:
 - COPYLOC (available with PTF for APAR PI91584 installed). The new COPYLOC compiler option allows you to add either a PDSE (or PDS) dataset or z/OS UNIX directory as an additional location to be searched for copy members during the library phase.
 - RULES:
 - Available with PTF for APAR PI91585 installed, new suboptions OMITODOMIN|NOOMITODOMIN are added to the RULES option to control whether the compiler will issue warning messages for any OCCURS DEPENDING ON clauses that are specified without integer-1 (the minimum number of occurrences).
 - Available with PTF for APAR PI91586 installed, new suboptions UNREF|NOUNREFALL|NOUNREFSOURCE are added to the RULES option to control whether the compiler will report unreferenced data items, and to control whether the reporting is done only for data items not declared in a copy member (NOUNREFSOURCE) or all data items (NOUNREFALL).
 - ZONEDATA. With PTF for APAR PI90571 installed, the ZONEDATA option is updated to affect the behavior of MOVE statements, comparisons, and computations for USAGE DISPLAY or PACKED-DECIMAL data items that could contain invalid digits, an invalid sign code, or invalid zone bits.
 - ALLOWCOPYLOC and ALLOWDEFINE (available with PTF for APAR PI97621 installed). New installation options of ALLOWCOPYLOC and ALLOWDEFINE allow installation administrators to decide whether their compiler users are permitted to use the COPYLOC and DEFINE compiler options respectively.
 - NUMCHECK(PAC). With PTF for APAR PI96135 installed, the NUMCHECK(PAC) option behavior is updated. For packed decimal (COMP-3) data items that have an even number of digits, the unused bits are checked for zeros.
 - NUMCHECK(ZON). With PTF for APAR PI98480 installed, new suboptions ALPHNUM|NOALPHNUM are added to the NUMCHECK(ZON) option to control whether the compiler will generate code for an implicit numeric class test for zoned decimal data items that are being compared with an alphanumeric data item, alphanumeric literal or alphanumeric figurative constant.
 - RULES(NOEVENTPACK). With PTF for APAR PH04369 installed, RULES(NOEVENTPACK) will not issue messages for even-digit

PACKED-DECIMAL data items whose names start with DFH, DSN, EYU or SQL, that is, data items generated for/by CICS and Db2.

- TEST|NOTEST. With PTF for APAR PH04485 installed, new suboptions DSNAME|NODSNAME are added to the TEST|NOTEST(SEPARATE) option to control whether the external file name, which is the SYSDEBUG dataset name used during compilation, will or will not be stored in the object program.
- INITIAL (available with PTF for APAR PH05855 installed). The new INITIAL compiler option causes a program and all of its nested programs to behave as if the IS INITIAL clause was specified on the PROGRAM-ID paragraph.
- NUMCHECK. With PTF for APAR PH08642 installed, redundant checks previously added by the NUMCHECK option have been removed, improving performance, and some checks can be done at compile time. Specifying NUMCHECK may also cause the compiler to produce some messages at compile time instead of at runtime.
- INITCHECK. With PTF for APAR PH09225 installed, the INITCHECK option can now be specified with OPTIMIZE(0).

Usability enhancements in the z/OS UNIX System Services environment

Enterprise COBOL V6.2 improves usability of the compiler in the z/OS UNIX System Services environment with addition of help information for the cob2 compiler invocation command.

Improved interfaces to other licensed programs and tools

- Addition of MD5 signature to program objects and debug data to allow matching of debug data with executables even if a program is recompiled.
- Three new fields at the end of PPA4:
 - Offset of the first user-defined data item in WORKING-STORAGE
 - Total length of user-defined data items in WORKING-STORAGE
 - Bit to indicate whether there are EXTERNAL data items

Compile-time and runtime performance improvements

- General compile-time performance improvements (with OPTIMIZE(1) and OPTIMIZE(2))
- General batch runtime performance improvements
- General online transaction runtime performance improvements

Chapter 3. 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.

Enterprise COBOL V6.2 with PTF for APAR PI97434 installed adds support for processing national data items with the following intrinsic functions:

- REVERSE. The REVERSE function returns a character value of the same length as the argument, whose characters are the same as those specified in the argument except that they are in reverse order.
- ULENGTH. The ULENGTH function returns an integer value that is equal to the number of UTF-8 or UTF-16 characters in a character data item argument that contains UTF-8 or UTF-16 data.
- UPOS. The UPOS function returns an integer value that is equal to the index of the *n*th UTF-8 or UTF-16 character in a character data item argument that contains UTF-8 or UTF-16.
- USUBSTR. The USUBSTR function returns a substring of the data in a character data item argument that contains UTF-8 or UTF-16 data.
- UWIDTH. The UWIDTH function returns an integer value that is equal to the width in bytes of the *n*th UTF-8 or UTF-16 character in a character data item argument that is encoded in UTF-8 or UTF-16.

Note: If the updated intrinsic functions (REVERSE, ULENGTH, UPOS, USUBSTR, UWIDTH) in the May compiler PTFs (UI56120, UI56121, UI56122) are used, then the May Runtime PTF UI56043(V2R1)/UI56042(V2R2)/UI55861(V2R3) must also be applied to Language Environment[®] on all systems where these programs are linked or run.

Enterprise COBOL V6.2 with PTF for APAR PI99703 installed adds support for the following intrinsic functions:

- BIT-OF. The BIT-OF function returns an alphanumeric character string consisting of characters "1" and "0" that correspond to the binary value of each byte in the input argument.
- E. The E function returns an approximation of *e*, the base of natural logarithms.
- HEX-OF. The HEX-OF function returns an alphanumeric character string consisting of the bytes of the input argument converted to a hexadecimal representation.
- PI. The PI function returns a value that is an approximation of *pi*, the ratio of the circumference of a circle to its diameter.
- TRIM. The TRIM function returns a character string that contains the characters in the argument with leading spaces, trailing spaces, or both, removed.

Note: If the new intrinsic functions (BIT-OF, E, HEX-OF, PI, TRIM) in the July compiler PTFs (UI57342, UI57343, UI57344, UI57345) are used, then the July Runtime PTF UI57304(V2R1)/UI57303(V2R2)/UI57302(V2R3) must also be applied to Language Environment on all systems where these programs are linked or run.

Enterprise COBOL V6.2 with PTF for APAR PH02183 installed adds support for the following intrinsic functions:

- ABS. The ABS function returns the absolute value of the argument.
- BIT-TO-CHAR. The BIT-TO-CHAR function returns a character string consisting of bytes that correspond to the bit pattern indicated by the sequence of "0" and "1" characters in the input argument.
- BYTE-LENGTH. The BYTE-LENGTH function returns an integer equal to the length of the argument in bytes.
- EXP. The EXP function returns an approximation of the value of e raised to the power of the argument.
- EXP10. The EXP10 function returns an approximation of the value of 10 raised to the power of the argument.
- HEX-TO-CHAR. The HEX-TO-CHAR function returns a character string consisting of bytes that correspond to the hexadecimal digit characters in the input argument.
- NUMVAL-F. The NUMVAL-F function returns the numeric value or an approximation of the numeric value represented by the alphanumeric character string or national character string specified as the argument. The function removes any leading or trailing spaces in the string to produce a numeric value.
- SIGN. The SIGN function returns +1, 0, or -1 depending on the sign of the argument.
- TEST-NUMVAL. The TEST-NUMVAL function verifies that the contents of the argument conform to the specification for the argument of the NUMVAL function.
- TEST-NUMVAL-C. The TEST-NUMVAL-C function verifies that the contents of the argument conform to the specification for the argument of the NUMVAL-C function.
- TEST-NUMVAL-F. The TEST-NUMVAL-F function verifies that the contents of the argument conform to the specification for the argument of the NUMVAL-F function.

Note: If the new intrinsic functions (BIT-TO-CHAR, HEX-TO-CHAR, NUMVAL-F, TEST-NUMVAL, TEST-NUMVAL-C, TEST-NUMVAL-F) in the September compiler PTFs (UI58632, UI58633, UI58634) are used, then the September Runtime PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied to Language Environment on all systems where these programs are linked or run.

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 (a base element of z/OS), 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[™] & VM, and IBM COBOL for OS/390[®] & VM.

With the exception of OS/VS COBOL programs, 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 COBOL Standard programs, which can then be compiled by using Enterprise COBOL for z/OS. Use the COBOL conversion tool (CCCA) included in IBM Debug for z Systems® (formerly IBM Debug Tool for z/OS) for this purpose. IBM Debug for z Systems® also includes a load module analyzer that can help identify which of your programs were compiled with the OS/VS COBOL compiler.

You can use the COBOL Migration Assistant to navigate through the compiler migration process from Enterprise COBOL V4 or earlier versions to Enterprise COBOL V5 or V6.

Support for modern development tools

IBM Developer for z Systems (formerly IBM Rational® Developer for z Systems) supports Enterprise COBOL and helps improve the productivity of COBOL developers. IBM Developer for z Systems provides an interactive, workstation-based environment to help you create, maintain, and reuse applications. IBM Developer for z Systems 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.

IBM Developer for z Systems provides a workstation interface to IBM Debug for z Systems, and is also integrated with IBM File Manager for z/OS and IBM Fault Analyzer for z/OS. File Manager integration enables you to access Keyed Sequence Data Set (KSDS) files from the IBM Developer for z Systems workbench, and gives you the ability to browse and update data sets. By integrating with Fault Analyzer, IBM Developer for z Systems 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.

Chapter 4. System requirements

The following table presents the system requirements for Enterprise COBOL for z/OS V6.2.

Table 1. System requirements for Enterprise COBOL for z/OS V6.2

Software	Hardware
<p>Enterprise COBOL for z/OS, V6.2 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 on the following programs listed that require program temporary fixes (PTFs), refer to the Program Directory and the preventive service planning (PSP) bucket.</p> <ul style="list-style-type: none"> • z/OS V2.1 (5650-ZOS), or later is required. • For installation on z/OS, z/OS SMP/E is required. • For customization during or after installation, z/OS High Level Assembler is required. • Enterprise COBOL XML PARSE statements in programs, which are compiled with the XMLPARSE(XMLSS) compiler option, require z/OS XML System Services V2.1 (5650-ZOS), or later. 	<p>Enterprise COBOL for z/OS, V6.2 will run on the following IBM Z servers:</p> <ul style="list-style-type: none"> • z14 • z13[®] or z13s[®] • zEnterprise[®] EC12 or zEnterprise BC12 • zEnterprise 196 or zEnterprise 114 • z10[™] Enterprise Class or z10 Business Class • z9[®] Enterprise Class or z9 Business Class

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

- IBM CICS Transaction Server (CICS TS) for z/OS, V5.x (5655-Y04)
- IBM CICS Transaction Server for z/OS Value Unit Edition, V5.x (5722-DFJ)
- IBM CICS Transaction Server for z/OS, V4.2 (5655-S97)
- IBM Db2 12 for z/OS (5650-Db2)
- IBM Db2 12 for z/OS Value Unit Edition (5770-AF3)
- IBM Db2 11 for z/OS (5615-Db2)
- IBM Db2 11 for z/OS Value Unit Edition (5697-P43)
- IBM IMS, V14 (5635-A05)
- IBM IMS Transaction Manager Value Unit Edition, V14 (5655-TM3)
- IBM IMS Database Value Unit Edition, V14 (5655-DSE)
- IBM IMS, V13 (5635-A04)
- IBM IMS Transaction Manager Value Unit Edition, V13 (5655-TM2)
- IBM IMS Database Value Unit Edition, V13 (5655-DSM)
- IBM DFSORT element of z/OS, V2 (5650-ZOS)
- IBM High Level Assembler/MVS and VM and VSE, V1.6.x (5696-234)
- IBM 31-bit SDK for z/OS, Java Technology Edition, V8.0 (5655-DGG)
- 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 z/OS V2.1 Client Web Enablement Toolkit (5650-ZOS - APAR Number OA46575)
- IBM Debug for z Systems, V14.x (5655-Q50)
- IBM Debug Tool for z/OS, V13.1 (5655-Q10)
- IBM Fault Analyzer for z/OS, V14.1 (5655-Q41)
- IBM Fault Analyzer for z/OS, V13.1 (5655-Q11)
- IBM File Manager for z/OS, V14.1 (5655-Q42)
- IBM File Manager for z/OS, V13.1 (5655-Q12)
- IBM Application Performance Analyzer for z/OS, V14.1 (5655-Q49)
- IBM Application Performance Analyzer for z/OS, V13.1 (5655-Q09)
- IBM Developer for z Systems, V14.x (5724-T07) (formerly IBM Rational Developer for z Systems, V9.x)
- IBM Application Delivery Foundation for z Systems, V3.1 (5655-AC6)
- IBM Application Delivery Intelligence, V5.x (5725-Y87)
- IBM Application Discovery and Delivery Intelligence, V5.x (5737-B66)
- IBM COBOL Report Writer Release 4 (5798-DYR, 5798-DZX)
- IBM Automatic Binary Optimizer for z/OS, V1.3 (5697-AB1) (for automatically optimizing pre-V5 COBOL program modules)
- IBM Enterprise COBOL for z/OS, V5.x (5655-W32)
- IBM Enterprise COBOL for z/OS, V4.2 (5655-S71)
- IBM Enterprise PL/I for z/OS, V5.x (5655-PL5)
- IBM Enterprise PL/I for z/OS, V4.5 (5655-W67)
- XL C/C++ with Enterprise COBOL (You must use the XL C/C++ feature of z/OS V2 (5650-ZOS), or later.)
- IBM VS FORTRAN, V2 (5668-806, 5688-087)

Chapter 5. Upgrade to Enterprise COBOL for z/OS V6.2

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

Chapter 6. For more information

To learn more about IBM Enterprise COBOL for z/OS V6.2, contact your IBM representative or IBM Business Partner, or visit Enterprise COBOL for z/OS.

Chapter 7. Notices

References in this document to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. 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.

UNIX is a registered trademark of The Open Group in the United States and other countries.



Product Number: 5655-EC6

Printed in USA