

Enterprise COBOL for z/OS  
6.2

*Data Sheet*



**October 2022**

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 International Business Machines Corporation 1991, 2022.**

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

---

# Contents

**Chapter 1. Summary of changes..... 1**

**Chapter 2. Enable your COBOL applications to exploit the latest z/Architecture..... 5**

**Chapter 3. Highlights..... 7**

**Chapter 4. Other Enterprise COBOL for z/OS features..... 11**

**Chapter 5. System requirements..... 15**

**Chapter 6. Upgrade to Enterprise COBOL for z/OS V6.2..... 17**

**Chapter 7. For more information..... 19**

**Chapter 8. Notices..... 21**

    Trademarks..... 21



---

# Chapter 1. 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 July 2017.

## October 2022

- PH50301: The NUMCHECK compiler option is updated to avoid generating runtime checking code of zoned decimal senders in MOVE statements when the receiver is an alphanumeric data item and NUMCHECK(ZON(LAX)) is in effect. ([NUMCHECK](#))

## December 2021

- PH42575: The following messages are updated to report additional information and return codes:
  - IGYSI5305-U ([IGYSI5305-U](#))
  - IGYSI5306-U ([IGYSI5306-U](#))

## October 2021

- PH41362: LAXREDEF | STRICTREDEF option is deprecated but is tolerated for compatibility, and it is replaced by the LAX | STRICT option. ([NUMCHECK](#))

## August 2021

- Runtime APAR PH34885 (V2R3/V2R4): The UUID4 randomness and UUID4 intrinsic function requires significant CPU usage. ([UUID4](#))
- Runtime APAR PH37101(V2R3/V2R4): An enhancement is made to assist COBOL programs running in AMODE 31 to interact with Java™ programs in AMODE 64. ([COBOL programs in AMODE 31 interacting with Java programs in AMODE 64](#))

**Note:** IBM SDK, Java Technology Edition V8.0.6.35 (JVM) is needed for this enhancement.

## June 2021

- PH36777: Adds support for diagnosing miscoded options or options coded as OPTION() instead of OPTION= in the COBOL customization macro. ([Changing the defaults for compiler options](#))

## April 2021

- PH35100: The new INVDATA compiler option replaces the deprecated ZONEDATA compiler option and provides users fine-grained control over how the compiler generates code to handle USAGE DISPLAY and USAGE PACKED-DECIMAL data items that contain invalid data. ([INVDATA](#))
- PH36688: New suboptions LAXREDEF | NOLAXREDEF are added to the RULES option to inform users of redefined items with mismatched lengths. ([RULES](#))
- PH36690: The OFFSET option behavior is changed. If there are multiple blocks of instructions for a single line of COBOL code, multiple entries will be generated for those instructions in the OFFSET table. ([OFFSET](#))

## December 2020

- PH32356: New when-phrase and generic-suppression-phrase are added to the JSON GENERATE statement so that you can conditionally suppress data items during JSON GENERATE. ([JSON GENERATE statement](#))

**Note:** COBOL Runtime LE APAR PH31172 must also be applied on all systems where programs that make use of this new feature are linked or run.

## October 2020

- Runtime APAR PH20569(V2R2/V2R3/V2R4): The included DWARF diagnostic information when TEST(NOSEPARATE) is in effect can be extracted from the LLA/VLF managed programs. ([TEST](#))

## August 2020

- PH28544: New suboptions TRUNCBIN | NOTRUNCBIN are added to the NUMCHECK(BIN) option to control whether the compiler will generate the checking code for binary data items. ([NUMCHECK](#))
- PTF UI71111 (No APAR number): New functionality is added to NUMCHECK to check alphanumeric senders whose contents are being moved to a numeric receiver. For alphanumeric senders whose contents are being moved to a numeric receiver, the compiler treats the sender as a numeric integer so NUMCHECK generates an implicit numeric class test for each alphanumeric sender. ([NUMCHECK](#))
- PH28546: A new CONVERTING phrase is added to the JSON GENERATE and JSON PARSE statements so that you can generate and parse JSON boolean values. ([JSON GENERATE statement](#) and [JSON PARSE statement](#))

**Note:** COBOL Runtime LE APAR PH26698 must also be applied on all systems where programs that make use of this new feature are linked or run.

## June 2020

- PH24340: New suboptions LAXREDEF | STRICTREDEF are added to the NUMCHECK(ZON) option to control whether the compiler will check and issue warning messages for redefined items. ([NUMCHECK](#))

## April 2020

- PH24413: New suboptions LAX | STRICT are added to the INITCHECK option to control whether the compiler will issue warning messages for data items unless they are initialized on at least one, or on all, logical paths to a statement. ([INITCHECK](#))

## December 2019

- PH19715: A new UUID4 intrinsic function is introduced, which returns a 36-character alphanumeric string that is a version 4 universally unique identifier. ([UUID4](#))

**Note:** COBOL Runtime LE PTF UI66560(V2R2)/UI66555(V2R3)/UI66557(V2R4) must also be applied on all systems where programs that make use of this new feature are linked or run.

- PH20084: A new NAME is OMITTED phrase is added to the JSON GENERATE statement to allow generation of an anonymous JSON object, whose top-level parent name is not generated. ([JSON GENERATE statement](#))

**Note:** COBOL Runtime LE PTF UI66560(V2R2)/UI66555(V2R3)/UI66557(V2R4) must also be applied on all systems where programs that make use of this new feature are linked or run.

## March 2019

- PH05855: The new INITIAL compiler option 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. ([INITIAL](#))
- 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 run time. ([NUMCHECK](#))
- PH09225: The INITCHECK option can be specified with OPTIMIZE(0). ([INITCHECK](#))

- PH11667: NUMCHECK(BIN) will check for binary data items (COMP, COMP-4, and USAGE BINARY) even when TRUNC(BIN) is in effect. ([NUMCHECK](#))

## 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®. ([RULES](#))
- 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. ([TEST](#))

## September 2018

- PH02183:
  - The following intrinsic functions are added as IBM extensions:
    - BIT-TO-CHAR ([BIT-TO-CHAR](#))
    - HEX-TO-CHAR ([HEX-TO-CHAR](#))
  - The following intrinsic functions are added as part of the 2014 COBOL Standard:
    - ABS ([ABS](#))
    - BYTE-LENGTH ([BYTE-LENGTH](#))
    - EXP ([EXP](#))
    - EXP10 ([EXP10](#))
    - NUMVAL-F ([NUMVAL-F](#))
    - SIGN ([SIGN](#))
    - TEST-NUMVAL ([TEST-NUMVAL](#))
    - TEST-NUMVAL-C ([TEST-NUMVAL-C](#))
    - TEST-NUMVAL-F ([TEST-NUMVAL-F](#))

**Note:** COBOL Runtime LE PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied on all systems where these programs are linked or run.

- PH02251: As the new keyword OMITTED is introduced, the JSON PARSE statement is updated. ([JSON PARSE statement](#))

**Note:** COBOL Runtime LE PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied on all systems where these programs are linked or run.

## July 2018

- PI98480: 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. ([NUMCHECK](#))
- PI99703:
  - The following intrinsic functions are added as IBM extensions:
    - BIT-OF ([BIT-OF](#))
    - HEX-OF ([HEX-OF](#))
  - The following intrinsic functions are added as part of the 2014 COBOL Standard:
    - E ([E](#))
    - PI ([PI](#))

- TRIM ([TRIM](#))

**Note:** COBOL Runtime LE PTF UI57304(V2R1)/UI57303(V2R2)/UI57302(V2R3) must also be applied on all systems where these programs are linked or run.

## May 2018

- PI96135: For packed decimal (COMP - 3) data items that have an even number of digits, the unused bits are checked for zeros. ([NUMCHECK](#))
- PI97160: Add 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. This is part of the 2002 COBOL Standard. ([Format 4: SET for condition-names](#) and [Format 2 VALUE clause: condition-name value](#))
- PI97434: Add support for processing national data items including surrogate-pair awareness to the following intrinsic functions: REVERSE, ULENGTH, UPOS, USUBSTR, and UWIDTH. ([Intrinsic functions and national data](#))
- PI97621: Add new installation options of ALLOWCOPYLOC and ALLOWDEFINE. ([ALLOWCOPYLOC](#) and [ALLOWDEFINE](#))

## March 2018

- PI95081: A new LOC(24|31) phrase is added to the ALLOCATE statement to control the location of dynamic storage that is acquired, which overrides the influence of the DATA compiler option. ([Storage and its addressability](#))

## January 2018

- PI91584: The new COPYLOC compiler option can be used 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. ([COPYLOC](#))
- PI91585: 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). ([RULES](#))
- PI91586: 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). ([RULES](#))
- PI91589: New COBOL words are added to the IBM-supplied CICS reserved-word table. ([CICS reserved-word table](#))

## November 2017

- PI90571: The ZONEDATA option is updated to affect the behaviour 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. ([ZONEDATA](#))



---

## Chapter 2. 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 3. Highlights

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

- Support for the new IBM z14<sup>®</sup> 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<sup>®</sup> 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<sup>®</sup> 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.

In Enterprise COBOL V6.2 with PTF for APAR PI97160 installed, support for the T0 FALSE phrase of the SET statement is added, as well as the corresponding WHEN SET T0 FALSE clause, which allow you to use the SET statement to set condition names to values that will test false in conditions. This is part of the 2002 COBOL Standard.

### **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 which parse the JSON text and process the transaction.

In Enterprise COBOL V6.2 with PTF for APAR PH02251 installed, a new keyword OMITTED is introduced to the JSON PARSE statement.

**Note:** COBOL Runtime LE PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied on all systems where these programs are linked or run.

In Enterprise COBOL V6.2 with PTF for APAR PH20084 installed, a new NAME is OMITTED phrase is added to the JSON GENERATE statement to allow generation of an anonymous JSON object, whose top-level parent name is not generated.

**Note:** COBOL Runtime LE PTF UI66560(V2R2)/UI66555(V2R3)/UI66557(V2R4) must also be applied on all systems where programs that make use of this new feature are linked or run.

In Enterprise COBOL V6.2 with PTF for APAR PH28546 installed, a new CONVERTING phrase is added to the JSON GENERATE and JSON PARSE statements so that you can generate and parse JSON boolean values.

**Note:** COBOL Runtime LE PTF PH26698 must also be applied on all systems where programs that make use of this new feature are linked or run.

In Enterprise COBOL V6.2 with PTF for APAR PH32356 installed, new when-phrase and generic-suppression-phrase are added to the JSON GENERATE statement so that you can conditionally suppress data items during JSON GENERATE.

**Note:** COBOL Runtime LE APAR PH31172 must also be applied on all systems where programs that make use of this new feature 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 times. 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 continue processing for additional 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 updates are introduced in Enterprise COBOL V6.2 via the service stream:
  - PI90571: The ZONEDATA option is updated to affect the behaviour 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.
  - PI91584: The new COPYLOC compiler option can be used 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.
  - PI91585: 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).
  - PI91586: 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).
  - PI96135: For packed decimal (COMP-3) data items that have an even number of digits, the unused bits are checked for zeros.
  - PI97621: New installation options of ALLOWCOPYLOC and ALLOWDEFINE are added.
  - PI98480: 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.
  - 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.
  - 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.
  - PH05855: The new INITIAL compiler option 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.
  - 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 run time.
  - PH09225: The INITCHECK option can be specified with OPTIMIZE(0).
  - PH11667: NUMCHECK(BIN) will check for binary data items (COMP, COMP-4, and USAGE BINARY) even when TRUNC(BIN) is in effect.
  - PH24340: New suboptions LAXREDEF | STRICTREDEF are added to the NUMCHECK(ZON) option to control whether the compiler will check and issue warning messages for redefined items.
  - PH24413: New suboptions LAX | STRICT are added to the INITCHECK option to control whether the compiler will issue warning messages for data items unless they are initialized on at least one, or on all, logical paths to a statement.
  - PH28544: New suboptions TRUNCBIN | NOTRUNCBIN are added to the NUMCHECK(BIN) option to control whether the compiler will generate the checking code for binary data items.
  - PTF UI71111 (No APAR number): New functionality is added to NUMCHECK to check alphanumeric senders whose contents are being moved to a numeric receiver. For alphanumeric senders whose contents are being moved to a numeric receiver, the compiler treats the sender as a numeric integer so NUMCHECK generates an implicit numeric class test for each alphanumeric sender.

- PH31500: The new INVDATA compiler option replaces the deprecated ZONEDATA compiler option and provides users fine-grained control over how the compiler generates code to handle USAGE DISPLAY and USAGE PACKED-DECIMAL data items that contain invalid data.
- PH36688: New suboptions LAXREDEF | NOLAXREDEF are added to the RULES option to inform users of redefined items with mismatched lengths.
- PH36690: The OFFSET option behavior is changed. If there are multiple blocks of instructions for a single line of COBOL code, multiple entries will be generated for those instructions in the OFFSET table.
- PH36777: Adds support for diagnosing miscoded options or options coded as OPTION() instead of OPTION= in the COBOL customization macro.

## **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 4. 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 (a base element of z/OS), also supports PL/I, C/C++, and Fortran programs.

The following intrinsic functions are introduced in Enterprise COBOL V6.2 via the service stream:

In Enterprise COBOL V6.2 with PTF for APAR PH02183 installed:

- The following intrinsic functions are added as IBM extensions:
  - BIT-TO-CHAR
  - HEX-TO-CHAR
- The following intrinsic functions are added as part of the 2014 COBOL Standard:
  - ABS
  - BYTE-LENGTH
  - EXP
  - EXP10
  - NUMVAL-F
  - SIGN
  - TEST-NUMVAL
  - TEST-NUMVAL-C
  - TEST-NUMVAL-F

**Note:** COBOL Runtime LE PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied on all systems where these programs are linked or run.

In Enterprise COBOL V6.2 with PTF for APAR PH19715 installed, a new UUID4 intrinsic function is introduced, which returns a 36-character alphanumeric string that is a version 4 universally unique identifier(UUID).

**Note:** COBOL Runtime LE PTF UI66560(V2R2)/UI66555(V2R3)/UI66557(V2R4) must also be applied on all systems where programs that make use of this new feature are linked or run.

In Enterprise COBOL V6.2 with PTF for APAR PI97434 installed, support for processing national data items including surrogate-pair awareness to the following intrinsic functions is added: REVERSE, ULENGTH, UPOS, USUBSTR, and UWIDTH.

In Enterprise COBOL V6.2 with PTF for APAR PI99703 installed:

- – The following intrinsic functions are added as IBM extensions:
  - BIT-OF
  - HEX-OF
- The following intrinsic functions are added as part of the 2014 COBOL Standard:

- E
- PI
- TRIM

**Note:** COBOL Runtime LE PTF UI57304(V2R1)/UI57303(V2R2)/UI57302(V2R3) must also be applied on all systems where these programs are linked or run.

In Enterprise COBOL V6.2 with PTF Runtime APAR PH34885 (V2R3/V2R4) installed, the UUID4 randomness and UUID4 intrinsic function requires significant CPU usage.

Runtime APAR PH37101(V2R3/V2R4): An enhancement is made to assist COBOL programs running in AMODE 31 to interact with Java programs in AMODE 64.

**Note:** IBM SDK, Java Technology Edition V8.0.6.35 (JVM) is needed for this enhancement.

In Enterprise COBOL V6.2 with PTF for APAR PH42575 installed, the following messages are updated to report additional information and return codes:

- IGYSI5305-U
- IGYSI5306-U

## Enhanced support for z/OS UNIX

With the new COPYLOC compiler option, you can specify a mix of z/OS UNIX directories and partitioned data sets to search for COBOL copybooks within a single compile job, regardless of how the compiler is invoked.

In addition, if you want to compile a COBOL application containing embedded SQL statements using the Db2 coprocessor and invoke the compiler from z/OS UNIX using the cob2 command, you can specify the DBRMLIB location.

Those enhancements for z/OS UNIX make it more efficient and flexible to develop and build COBOL applications leveraging DevOps tools such as IBM Developer for z/OS (IDz) or IBM Dependency Based Build (DBB).

## IBM-supplied CICS reserved-word table changes

In Enterprise COBOL V6.2 with PTF for APAR PI91589 installed, new COBOL words are added to the IBM-supplied CICS reserved-word table.

## Statement changes

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

## 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 using Enterprise COBOL for z/OS. Use the COBOL conversion tool (CCCA) included in Debug Tool for this purpose. Debug Tool also includes a load module analyzer that can help identify which of your programs were compiled with the OS/VS 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/OS (formerly IBM Developer for z Systems® and Rational® Developer for z Systems) supports Enterprise COBOL and helps improve the productivity of COBOL developers. IBM Developer for z/OS provides an interactive, workstation-based environment to help you create, maintain, and reuse applications. IBM Developer for z/OS 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/OS provides a workstation interface to IBM Debug Tool, 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/OS workbench, and gives you the ability to browse and update data sets. By integrating with Fault Analyzer, IBM Developer for z/OS 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 5. System requirements

The following table presents the 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"><li>• z/OS V2.1 (5650-ZOS), or later is required.</li><li>• For installation on z/OS, z/OS SMP/E is required.</li><li>• For customization during or after installation, z/OS High Level Assembler is required.</li><li>• 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.</li></ul>	<p>Enterprise COBOL for z/OS, V6.2 runs on the following IBM servers:</p> <ul style="list-style-type: none"><li>• z14</li><li>• z13 or z13s<sup>®</sup></li><li>• zEnterprise<sup>®</sup> EC12 and zEnterprise BC12</li><li>• zEnterprise 196 or zEnterprise 114</li></ul>

### Optional licensed programs

Depending on the functions used, you might require other software products such as CICS, Db2, or IMS. For a list of compatible software, see the [Software Product Compatibility Reports \(SPCR\)](#) site.

From the SPCR web page, in the **In-depth reports** section, under **Detailed system requirements**, click **Create a report**. Search for Enterprise COBOL for z/OS, choose **Version 6.2** and then click **Submit**.



---

## Chapter 6. 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 7. 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 the [Enterprise COBOL for z/OS product page](#).





---

## Chapter 8. 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](http://ibm.com)<sup>®</sup> 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](http://www.ibm.com/legal/copytrade.shtml).







Product Number: 5655-EC6