Enterprise COBOL for z/OS 6.1

Data Sheet

August 2021

This edition applies to Version 6 Release 1 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.

Contents

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 1 was released in April 2016.

August 2021

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.

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)

April 2020

- PH23265: 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)
- PH24414: 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

- PH20081: A new UUID4 intrinsic function is introduced, which returns a 36-character alphanumeric string that is a version 4 universally unique identifier (UUID). (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.

September 2019

- PH13943: NUMCHECK(BIN) is enhanced to check for binary data items (COMP, COMP-4, and USAGE BINARY) even when TRUNC(BIN) is in effect. (NUMCHECK)

August 2018

- PH01251: 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)

June 2018

- PI98996: NUMCHECK(PAC) is enhanced to check for zeros in the unused bits for packed decimal (COMP-3) data items that have an even number of digits. (NUMCHECK)

April 2018

- PI96231: A new COPYLOC 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. The

location specified by the COPYLOC option is added to the end of the order of locations to search for copy members. (COPYLOC)

February 2018

- PI92944: 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. (ALLOCATE statement)

June 2017

- PI81838: All data items with a VALUE clause are initialized when the NOSTGOPT compiler option is in effect, regardless of the optimization level. (STGOPT)

April 2017

- PI77981: A new INLINE option (and a new INLINE compiler directive) can be used to guide the inlining of performed sections and paragraphs at OPTIMIZE(1) or OPTIMIZE(2). (INLINE)
- PI78089: A new PARMCHECK option can be used to discover parameter mismatches, that is, if programs pass arguments to subprograms that are then misused as parameters. It 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 the called subroutine corrupted data beyond the end of WORKING-STORAGE. (PARMCHECK)

February 2017

- PI71625: A new NUMCHECK option tells the compiler whether to generate extra code to validate data items when they are used as sending data items. (ZONECHECK)

  Note: The ZONECHECK option is deprecated but is tolerated for compatibility, and it is replaced by NUMCHECK(ZON).
- PI74933: New suboptions MSG and ABD are added to the SSRANGE compiler option to control how the compiler checks reference modification lengths. (SSRANGE)

December 2016

- PI68023: When NOSTGOPT and OPT(1) compiler options are in effect, all data items with VALUE clauses are initialized, even when they are never referenced. (STGOPT)

September 2016

- PI68226: A new INITCHECK option tells the compiler to check for uninitialized data items and issue warning messages when they are used without being initialized. (INITCHECK)

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, V6.1 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, V6.1 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, V6.1 delivers the following new and improved features:

- Increased compiler capacity
- New features added from the ISO 2002 COBOL Standard
- New and enhanced compiler options for ease of migration and programmer productivity
- Runtime and product-related enhancements

Increased compiler capacity

The capacity of the compiler has been expanded to allow for the compilation and optimization of large programs. With Enterprise COBOL for z/OS, V6.1, you can now compile much larger programs including those COBOL programs that are created by code generators.

New features added from the ISO 2002 COBOL Standard

In Enterprise COBOL V6.1, the following features are added for ISO 2002 COBOL Standard conformance:

The ALLOCATE statement
The ALLOCATE statement obtains dynamic storage.

In Enterprise COBOL V6.1 with PTF for APAR PI92944 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.

The FREE statement
The FREE statement releases dynamic storage previously obtained with an ALLOCATE statement.

The INITIALIZE statement
The INITIALIZE statement sets selected categories of data fields to predetermined values. The INITIALIZE statement is functionally equivalent to one or more MOVE statements. The following new clauses from the 2002 and 2014 COBOL Standards are added in V6.1:

- WITH FILLER
- TO VALUE
- THEN TO DEFAULT

New IBM extension features

The JSON GENERATE statement
The new JSON GENERATE statement converts data to JSON format.

In Enterprise COBOL V6.1 with PTF for APAR PH23265 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.

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

- The new VSAMOPENFS(COMPAT|SUCC) option allows you to change File Status=97 into File Status=00 for certain VSAM OPEN statements.
- The new SUPPRESS|NOSUPRESS option enables or disables the SUPPRESS phrase of COPY statements.
- The new SSRANGE(ZLEN|NOZLEN) suboption allows a 0-length reference modification.

- The diagnostic message for the ZONECHECK(MSG) compiler option (and also the newer NUMCHECK(MSG) compiler option) is improved by adding the data item contents for the offending data item and also adding the program name of the program that contained the offending data item.
- The LVLINFO installation option has been removed and replaced by a 7-character build-level identifier, of the format PYYMMDD, that is added to the compiler listing header. The build-level identifier is placed in locations that previously held the following LVLINFO data:
  - Listing header
  - Signature information bytes
  - ADATA field called PTF Level
- The following updates are introduced in Enterprise COBOL V6.1 via the service stream:
  - PI68023: When NOSTGOPT and OPT(1) compiler options are in effect, all data items with VALUE clauses are initialized, even when they are never referenced.
  - PI68226: A new INITCHECK option tells the compiler to check for uninitialized data items and issue warning messages when they are used without being initialized.
  - PI71625: A new NUMCHECK option tells the compiler whether to generate extra code to validate data items when they are used as sending data items.

    Note: The ZONECHECK option is deprecated but is tolerated for compatibility, and it is replaced by NUMCHECK(ZON).
  - PI74933: New suboptions MSG and ABD are added to the SSRANGE compiler option to control how the compiler checks reference modification lengths.
  - PI77981: A new INLINE option (and a new INLINE compiler directive via PI77981) can be used to guide the inlining of performed sections and paragraphs at OPTIMIZE(1) or OPTIMIZE(2).
  - PI78089: A new PARMCHECK option can be used to discover parameter mismatches, that is, if programs pass arguments to subprograms that are then misused as parameters. It 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 the called subroutine corrupted data beyond the end of WORKING-STORAGE.
  - PI81838: All data items with a VALUE clause are initialized when the NOSTGOPT compiler option is in effect, regardless of the optimization level.
  - PI96231: A new COPYLOC 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. The location specified by the COPYLOC option is added to the end of the order of locations to search for copy members.
  - PI98996: NUMCHECK(PAC) is enhanced to check for zeros in the unused bits for packed decimal (COMP-3) data items that have an even number of digits.
  - PH01251: 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.
  - PH13943: NUMCHECK(BIN) is enhanced to check for binary data items (COMP, COMP-4, and USAGE BINARY) even when TRUNC(BIN) is in effect.
  - PH24414: 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.

Product-related enhancements

Enterprise COBOL for z/OS, V6.1 delivers the following runtime and performance-related enhancements:

- The WORKING-STORAGE section will be acquired from HEAP storage, and is managed more efficiently by initializing it when the corresponding COBOL program is being called. The STORAGE runtime option will affect the WORKING-STORAGE section. In previous versions, the WORKING-STORAGE section was initialized when the program object is being loaded, regardless of whether the COBOL program will

actually be called later. Also, sometimes WORKING-STORAGE was not acquired from HEAP, and in those cases the STORAGE option did not affect WORKING-STORAGE.

- Reduced storage requirements and performance tuning is implemented in Table SORT. Performance improvements are implemented in INSPECT, UNSTRING, and SEARCH ALL.

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.

In Enterprise COBOL V6.1 with PTF for APAR PH20081 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.

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.

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 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 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 V6.1.

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

Software	Hardware
Enterprise COBOL for z/OS, V6.1 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 on the following programs listed that require program temporary fixes (PTFs), refer to the Program Directory and the preventive service planning (PSP) bucket. • z/OS V2.1 (5650-ZOS), or later	Enterprise COBOL for z/OS, V6.1 will run on the following IBM servers: • z13 or z13s • zEnterprise® EC12 and zEnterprise BC12 • zEnterprise 196 or zEnterprise 114

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) website.

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 6.1** and then click **Submit**.

Chapter 6. Upgrade to Enterprise COBOL for z/OS V6.1

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

Chapter 7. For more information

To learn more about IBM Enterprise COBOL for z/OS V6.1, 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® 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