Enterprise COBOL for z/OS
6.3

Data Sheet

IBM

May 2021

This edition applies to Version 6 Release 3 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 3 was released in September 2019.

May 2021

- PH35643: New suboptions `DEC | HEX` are added to the `SOURCE` option to control whether the generated sequence numbers for the listing of the source are in decimal or hexadecimal format. (SOURCE)
- 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)
- PH37328: 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)

March 2021

- PH34804: A new `TUNE` option is added that allows you to specify the architecture for which the executable program will be optimized. (TUNE)
- PH35652: 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)

January 2021

- PH33122: New suboptions `LAXREDEF | NOLAXREDEF` are added to the `RULES` option to inform users of redefined items with mismatched lengths. (RULES)

November 2020

- PH30975: 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.

- PH31047: New date and time intrinsic functions are introduced as part of the 2002 and 2014 COBOL Standard:
  - COMBINED-DATETIME (COMBINED-DATETIME)
  - FORMATTED-CURRENT-DATE (FORMATTED-CURRENT-DATE)
  - FORMATTED-DATE (FORMATTED-DATE)
  - FORMATTED-DATETIME (FORMATTED-DATETIME)
  - FORMATTED-TIME (FORMATTED-TIME)
  - INTEGER-OF-FORMATTED-DATE (INTEGER-OF-FORMATTED-DATE)
  - SECONDS-FROM-FORMATTED-TIME (SECONDS-FROM-FORMATTED-TIME)
  - SECONDS-PAST-MIDNIGHT (SECONDS-PAST-MIDNIGHT)
  - TEST-DATE-YYYYMMDD (TEST-DATE-YYYYMMDD)
  - TEST-DAY-YYYYDDD (TEST-DAY-YYYYDDD)
  - TEST-FORMATTED-DATETIME (TEST-FORMATTED-DATETIME)

Note: COBOL Runtime LE APAR PH31133 must also be applied on all systems where programs that make use of these new date and time intrinsic functions are linked or run.

- Runtime APARs PH29755(V2R3/V2R4) and PH30338(V2R3/V2R4 AMODE 64): New support is added for LLA/VLF managed programs where DWARF diagnostic information is included. (TEST)

September 2020

- PH29542: 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 UI71591 (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)

July 2020

- PH26789: 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.
- PH27536: 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)

May 2020

- PH22581: 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)

January 2020

- PH20724: The use of passing a *file-name* to a subprogram with the USING phrase of the CALL statement is restored. (CALL statement)
- PH20997: 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.

November 2019

- PH18638: You can compile programs with the LP(64) option when the program contains UTF-8 data items. (UTF-8 literals and Defining UTF-8 data items)
- PH18640: You can compile programs with the LP(64) option when the program contains dynamic-length elementary items. (Dynamic-length items)
- PH18641: 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)

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, V6.3 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 that are 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, V6.3 reaffirms IBM's commitment to COBOL on z/OS. You gain the benefit of new investments that are combined with more than 50 years of IBM experience in compiler innovation and development.

Chapter 3. Highlights

Enterprise COBOL for z/OS, V6.3 delivers the following new and improved features:

- Compiler and runtime support for the new IBM z15™ hardware and IBM z/OS V2.4 operating system so that applications can take advantage of the latest IBM Z hardware instructions and operating system features and capabilities.
- Improved efficiency in processing UTF-8 data by supporting the UTF-8 data type. This enhancement improves efficiency for native language support.
- Support for creating AMODE 64 (64-bit) batch applications. The AMODE 64 (64-bit) support in this compiler enables users to process large data tables that require up to 2 GB of addressing space.
- From the COBOL 2002 and 2014 programming standards, the addition of Dynamic Length elementary items.
- From the COBOL 2002 and 2014 programming standards, the support for the FUNCTION keyword specifier in the REPOSITORY section to make the use of the keyword FUNCTION optional.

Compiler and runtime support for the new IBM z15 hardware and IBM z/OS V2.4 operating system

Enterprise COBOL for z/OS, V6.3 adds support for building and running COBOL applications for the z15 hardware and z/OS V2.4 operating system

With the new ARCH(13) compiler option, the compiler generates application code that exploits instructions available with the latest z15 server. Specifying ARCH(13) instructs the compiler to include exploitation of the Vector Packed Decimal Enhancement Facility, which translates into improvements in COBOL computational performance. You can recompile with ARCH(13) to target z15 without any source changes to take advantage of this new facility.

Improved processing of UTF-8 data with the introduction of the UTF-8 data type

- Enterprise COBOL for z/OS, V6.3 provides increased efficiency and support for Unicode data encoded in UTF-8 format. You can now store UTF-8 data in data items that natively understand UTF-8. Correct UTF-8 padding and truncation are performed on these data items during MOVE and other operations.
- The native support for UTF-8 data items is provided by a new USAGE clause of UTF-8, plus a picture symbol 'U', which together define a new class, category, and USAGE of data in Enterprise COBOL.
- UTF-8 data items can be declared as having either a fixed character length where a UTF-8 character corresponds to one Unicode code point or a fixed byte length, with the latter being provided for ease of interoperability with Db2 CHAR columns in Unicode tables.
- USAGE UTF-8 data items can be optionally declared to have a dynamic length so that the actual length of the UTF-8 data is carried around automatically with the item, and memory for the item is acquired dynamically.
- COBOL statements that directly support UTF-8 include ALLOCATE, EVALUATE, FREE, IF, INITIALIZE, MOVE, MERGE, and SORT.
- UTF-8 data items are supported in conditions such as those found in IF, WHEN, and SEARCH statements.
- UTF-8 data can be a new class and category of argument for intrinsic functions. UTF-8 data can also be a return type for intrinsic functions. Supported functions in this release are BIT-OF, BYTE-LENGTH, DISPLAY-OF, HEX-OF, LENGTH, LOWER-CASE, NATIONAL-OF, TRIM, ULENGTH, UPPER-CASE, UPOS, USUBSTR, USUPPLEMENTARY, and UVALID.

The introduction of native support for UTF-8 data items means that you can now work directly with UTF-8 data without having to waste CPU resources converting from UTF-8 to UTF-16 and back again. This results in more maintainable programs and is especially useful when you modernize your COBOL applications to work with web services or to interoperate with Db2 databases if you choose to store data in UTF-8 format. In cases where it is more efficient to perform processing in UTF-16, the conversion is handled automatically when necessary.

AMODE 64 support

With this release, Enterprise COBOL for z/OS, V6.3 provides support for creating AMODE 64 COBOL batch applications. Application developers do not need to do any redesign or source code changes on their applications to obtain AMODE 64 support.

AMODE 64 COBOL applications can now access data items greater than the existing AMODE 31 data size limits without changes to the program logic.

AMODE 64 COBOL application features are listed as follows:

- A new LP(32|64) compiler option indicates whether an AMODE 31 (31-bit) or AMODE 64 (64-bit) program should be generated with related language features enabled.
- Files that are created by AMODE 31 programs can be accessed (read, write, and rewrite) by AMODE 64 programs and vice-versa. Data files are compatible between AMODE 64 and AMODE 31 programs.
- AMODE 64 COBOL programs can call other AMODE 64 Language Environment® (LE) conforming programs by using either static or DLL calls.
- AMODE 64 COBOL programs cannot be called by non-LE conforming programs. Assembler programs that use LOAD and then branch to the entry point of the subprogram cannot work. Instead, the LE macro CEEFETCH should be used to fetch and call AMODE 64 COBOL programs.
- The linkage convention is XPLINK and the XPLINK(ON) runtime option is required for AMODE 64 COBOL programs.
- All AMODE 64 COBOL programs are reentrant, that is, LP(64) implies RENT.
- Enterprise COBOL V6.3 with PTF for APAR PH18638 installed adds support for compiling programs with the LP(64) option when the program contains UTF-8 data items.
- Enterprise COBOL V6.3 with PTF for APAR PH18640 installed adds support for compiling programs with the LP(64) option when the program contains dynamic-length elementary items.
- The following features are not available:
  - No support for building AMODE 64 applications to run under CICS or IMS.
  - No support for building AMODE 64 applications that contain OO COBOL, XML, or JSON statements or use the THREAD compiler option.
  - No support for mixing AMODE 64 and AMODE 31 object files in the same COBOL program, or mixing AMODE 64 and AMODE 31 in the same COBOL run unit; if one COBOL source file is compiled with LP(64), all COBOL source files that are compiled and linked into the same program must be compiled with LP(64).

COBOL 2002 and 2014 standards

The support of programming language standards provides you with additional functionality so that you can modernize your applications. It also allows for maximum portability of your source code among a variety of compiler implementations.

The following COBOL 2002 and 2014 features are provided by the V6.3 compiler:

- Dynamic-length elementary items (COBOL 2014 language feature)
  - Addition of the DYNAMIC LENGTH clause provides the ability to describe a data item of varying size.
- FUNCTION keyword optional for intrinsic functions (COBOL 2002 language feature)
  - The REPOSITORY paragraph FUNCTION specifier INTRINSIC allows for the declaration of intrinsic function names that may be used without specifying the word FUNCTION.

- PH31047: New date and time intrinsic functions are introduced that support encoding and decoding of date and time information to and from formats specified in ISO 8601, and that support encoding and decoding date and time information to and from integers that are suitable for arithmetic.

  The following intrinsic functions are added as part of the 2002 COBOL Standard:

  - TEST-DATE-YYYYMMDD
  - TEST-DAY-YYYYDDD

  The following intrinsic functions are added as part of the 2014 COBOL Standard:

  - COMBINED-DATETIME
  - FORMATTED-CURRENT-DATE
  - FORMATTED-DATE
  - FORMATTED-DATETIME
  - FORMATTED-TIME
  - INTEGER-OF-FORMATTED-DATE
  - SECONDS-FROM-FORMATTED-TIME
  - SECONDS-PAST-MIDNIGHT
  - TEST-FORMATTED-DATETIME

  Note: COBOL Runtime LE APAR PH31133 must also be applied on all systems where programs that make use of these new date and time intrinsic functions are linked or run.

Chapter 4. Other Enterprise COBOL for z/OS features

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).

Additional enhancements

- Default ARCH changed to ARCH(8) (IBM System z10® BC) and removed support for ARCH(7) (IBM System z9® BC).
- Expanded signature information area for future option byte definitions.
- Compiler listing enhancements for AMODE 64 and COBOL terminology.
- Predefined compilation variables available with IGY-prefixes to keep all variables in the IGY-xxxx namespace.
- The following enhancements are introduced in Enterprise COBOL V6.3 via the service stream:
  - PH18641: 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.
  - PH20724: The use of passing a *file-name* to a subprogram with the USING phrase of the CALL statement is restored.
  - PH20997: A new UUID4 intrinsic function is introduced, which returns a 36-character alphanumeric string that is a version 4 universally unique identifier.

    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.
  - PH22581: 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.
  - PH26789: 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 APAR PH26698 must also be applied on all systems where programs that make use of this new feature are linked or run.
  - PH27536: 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.
  - PH29542: 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 UI71591 (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.
  - PH30975: 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.

- Runtime APARs PH29755(V2R3/V2R4) and PH30338(V2R3/V2R4 AMODE 64): New support is added for LLA/VLF managed programs where DWARF diagnostic information is included.
- PH33122: New suboptions LAXREDEF | NOLAXREDEF are added to the RULES option to inform users of redefined items with mismatched lengths.
- PH34804: A new `TUNE` option is added that allows you to specify the architecture for which the executable program will be optimized.
- PH35643: New suboptions `DEC | HEX` are added to the `SOURCE` option to control whether the generated sequence numbers for the listing of the source are in decimal or hexadecimal format.
- PH35652: 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.
- PH37328: A 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.

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 V6.3.

Table 1. System requirements for Enterprise COBOL for z/OS V6.3

Software	**Hardware**
Enterprise COBOL for z/OS, V6.3 runs under the control of, or along with, the currently supported releases of the following programs and their subsequent releases or their equivalents. For more information about the following programs listed that require program temporary fixes (PTFs), refer to the Program Directory and the preventive service planning (PSP) bucket. • z/OS V2.2 (5650-ZOS), or later is required. **Note:** To run 64-bit applications, z/OS V2.3 (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 V2.2 (5650-ZOS), or later.	Enterprise COBOL for z/OS, V6.3 runs on the following IBM Z servers: • z15 • IBM z14® or IBM z14 ZR1 • IBM z13® or IBM z13s® • zEnterprise® EC12 or 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) 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 6.3** and then click **Submit**.

Chapter 6. Upgrade to Enterprise COBOL for z/OS V6.3

Upgrade to the latest Enterprise COBOL compiler and get more out of your IBM mainframe 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 V6.3, contact your IBM representative or IBM Business Partner, or visit the Enterprise COBOL for z/OS product page.

Chapter 8. Notices

Trademarks

Product Number: 5655-EC6