

Enterprise COBOL for z/OS
6.2

What's New



Note

Before using this information and the product it supports, be sure to read the general information under [“Notices” on page 9](#).

First edition (31 December 2021 update)

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.

It is our intention to update the product documentation for this release periodically, without updating the order number. If you need to uniquely refer to the version of your product documentation, refer to the order number with the date of update.

© **Copyright International Business Machines Corporation 2021.**

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

Contents

About this information.....	v
How to send your comments.....	vii
Chapter 1. Video: What is new in Enterprise COBOL for z/OS 6.2.....	1
Chapter 2. What is new in Enterprise COBOL for z/OS 6.2 and COBOL 6.2 with PTFs installed.....	3
Notices.....	9
Trademarks.....	11
Enterprise COBOL for z/OS publications.....	13
Index.....	15

About this information

This information provides an executive overview of new and improved features in Enterprise COBOL for z/OS 6.2 and Enterprise COBOL for z/OS 6.2 with PTFs installed.

Throughout this information, "COBOL" or "Enterprise COBOL" means "IBM Enterprise COBOL for z/OS".

How to send your comments

Your feedback is important in helping us to provide accurate, high-quality information. If you have comments about this information or any other Enterprise COBOL documentation, send your comments to: compinfo@cn.ibm.com.

Be sure to include the name of the document, the publication number, the version of Enterprise COBOL, and, if applicable, the specific location (for example, the page number or section heading) of the text that you are commenting on.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way that IBM believes appropriate without incurring any obligation to you.

Chapter 1. Video: What is new in Enterprise COBOL for z/OS 6.2

Watch the [video](#) to get a quick overview of the major new features and enhancements in Enterprise COBOL for z/OS 6.2.

To discover all the new features and enhancements, refer to the complete list in [Chapter 2, “What is new in Enterprise COBOL for z/OS 6.2 and COBOL 6.2 with PTFs installed,”](#) on page 3.

To watch more Enterprise COBOL for z/OS videos, visit the [COBOL video gallery](#).

Chapter 2. What is new in Enterprise COBOL for z/OS 6.2 and COBOL 6.2 with PTFs installed

This section lists new and improved features in Enterprise COBOL for z/OS 6.2 and Enterprise COBOL for z/OS 6.2 with PTFs installed.

Enterprise COBOL for z/OS supports the continuous delivery (CD) model so that you can receive new features and enhanced capabilities as soon as the code is ready without waiting for the next release. The below feature descriptions that begin with an APAR number are introduced in the PTFs that are shipped with Enterprise COBOL for z/OS V6.2. Install the latest PTFs using the FIXCAT feature of SMP/E to take advantage of these features. For more information on SMP/E FIXCATs, see [Determining service required](#). For more information on PTFs and APARs for Enterprise COBOL for z/OS, see the [COBOL for z/OS support page](#).

The enhancements that are described in this section also have an associated cross-reference to the COBOL publications for your convenience.

The changes mainly fall into the following categories:

- [Support of 2002 and 2014 COBOL standards](#)
- [Support of IBM extensions](#)
- [JSON enhancements](#)
- [Other language element changes](#)
- [Compiler option changes](#)
- [Usability enhancements](#)
- [Listing changes](#)
- [IBM-supplied CICS® reserved-word table changes](#)
- [Installation customization changes](#)
- [“COBOL/JNI interface enhancement” on page 8](#)
- [Compiler message changes](#)

Support of 2002 and 2014 COBOL standards

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

- The DEFINE directive defines or undefines a compilation variable. ([DEFINE](#))
- The EVALUATE directive provides a multi-branch method of choosing the source lines to include in a compilation group. ([EVALUATE](#))
- The IF directive provides for a one-way or two-way conditional compilation. ([IF](#))
- The new INLINE directive allows the compiler to decide whether to inline procedures referenced by PERFORM statements in the source program. ([INLINE](#))
- 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. ([Format 4: SET for condition-names](#) and [Format 2 VALUE clause: condition-name value](#))

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

PI99703:

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.

- The E function returns an approximation of e , the base of natural logarithms. ([E](#))
- The PI function returns a value that is an approximation of π , the ratio of the circumference of a circle to its diameter. ([PI](#))
- The TRIM function returns a character string that contains the characters in the argument with leading spaces, trailing spaces, or both, removed. ([TRIM](#))

PH02183:

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.

- The ABS function returns the absolute value of the argument. ([ABS](#))
- The BYTE-LENGTH function returns an integer equal to the length of the argument in bytes. ([BYTE-LENGTH](#))
- The EXP function returns an approximation of the value of e raised to the power of the argument. ([EXP](#))
- The EXP10 function returns an approximation of the value of 10 raised to the power of the argument. ([EXP10](#))
- The NUMVAL-F function returns 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. ([NUMVAL-F](#))
- The SIGN function returns +1, 0, or -1 depending on the sign of the argument. ([SIGN](#))
- The TEST-NUMVAL function verifies that the contents of argument-1 conform to the specification for argument-1 of the NUMVAL function. ([TEST-NUMVAL](#))
- The TEST-NUMVAL-C function verifies that the contents of argument-1 conform to the specification for argument-1 of the NUMVAL-C function. ([TEST-NUMVAL-C](#))
- The TEST-NUMVAL-F function verifies that the contents of argument-1 conform to the specification for argument-1 of the NUMVAL-F function. ([TEST-NUMVAL-F](#))

Support of IBM extensions

The following intrinsic functions are added as IBM® extensions:

- PI99703:
 - 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. ([BIT-OF](#))
 - The HEX-OF function returns an alphanumeric character string consisting of the bytes of the input argument converted to a hexadecimal representation. ([HEX-OF](#))

You can use the HEX-OF or BIT-OF intrinsic functions to convert data of any type to hexadecimal or binary digits. ([Converting to hexadecimal or bit data \(HEX-OF, BIT-OF\)](#))

- PH02183:
 - 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. ([BIT-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. ([HEX-TO-CHAR](#))

You can use the HEX-TO-CHAR or BIT-TO-CHAR intrinsic functions to convert hexadecimal character strings to alphanumeric data items. ([Converting from hexadecimal or bit data \(HEX-TO-CHAR, BIT-TO-CHAR\)](#))

JSON enhancements

You can watch this [video](#) to get an overview of the JSON support in Enterprise COBOL for z/OS V6.

- JSON GENERATE:

- 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.

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.

- PH28546: A new "CONVERTING" phrase is added to the JSON GENERATE statement so that you can generate 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.

- 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.

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.

([JSON GENERATE statement](#))

- JSON PARSE:

- The new JSON-STATUS special register is used to indicate either that a JSON PARSE statement executed successfully or that a nonexception condition occurred. ([JSON-STATUS](#))

- The JSON-CODE special register is also used to indicate either that a JSON PARSE statement executed successfully or that an exception condition occurred. ([JSON-CODE](#))

- PH02251: As the new keyword OMITTED is introduced, the JSON PARSE statement is updated.

Note: COBOL Runtime LE PTF UI58596(V2R1)/UI58595(V2R2)/UI58603(V2R3) must also be applied on all systems where programs that make use of this new feature are linked or run.

- PH28546: A new "CONVERTING" phrase is added to the JSON PARSE statement so that you can 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.

([JSON PARSE statement](#))

Other language element changes

- 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.

- 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](#))
- 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](#))

Compiler option changes

The following compiler options are new:

- PI97621: ALLOWCOPYLOC: ALLOWCOPYLOC affects whether COPYLOC options can be used when compiling COBOL programs. ([ALLOWCOPYLOC](#))
- PI97621: ALLOWDEFINE: ALLOWDEFINE affects whether DEFINE options can be used when compiling COBOL programs. ([ALLOWDEFINE](#))

- **PI91584: COPYLOC:** 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](#))
- **DEFINE:** The DEFINE compiler option provides a way for you to assign values to compilation variables from outside the program source. If that is not needed, it is sufficient to use the DEFINE directive within program source to define compilation variables. ([DEFINE](#))
- **PH05855: INITIAL:** 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](#))
- **INITCHECK:**
 - Use the INITCHECK option to have the compiler check for uninitialized data items and issue warning messages when they are used without being initialized.
 - **PH09225:** The INITCHECK option can be specified with OPTIMIZE(0).
 - **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](#))
- **INLINE:** The INLINE compiler option controls whether the inlining of procedures (paragraphs or sections) referenced by PERFORM statements in the source program is allowed. ([INLINE](#))
- **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. ([INVDATA](#))
- **NUMCHECK:**
 - The NUMCHECK compiler 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 see whether the data item has more digits than its PICTURE clause allows.
 - **PI96135:** For packed decimal (COMP-3) data items that have an even number of digits, the unused bits are checked for zeros.
 - **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.
 - **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.
 - **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.
 - **PH28544:** New suboptions TRUNCBIN | NOTTRUNCBIN 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.
 - **PH41362:** When a zoned decimal data item is moved to another zoned decimal data item, NUMCHECK will not check the sender of the move. However, if the sender is subsequently used in a numeric context, it will be checked.

(NUMCHECK)

- PARMCHECK: The PARMCHECK option tells the compiler to generate an extra data item following the last item in WORKING-STORAGE. This buffer data item is then used at run time to check whether a called subprogram corrupted data beyond the end of WORKING-STORAGE. (PARMCHECK)

The following compiler options are modified:

- AFP: The default value is changed to AFP(NOVOLATILE). (AFP)
- ARCH: A new higher level of ARCH(12) is accepted. ARCH(7) is still the default. (ARCH)
- MAXPCF: The default value is changed to MAXPCF(100000) to reflect the increased capacity of the V6 compiler. (MAXPCF)
- PH36690: OFFSET: 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)
- NOSTGOPT: In earlier versions, data items can get optimized with OPT(2) even when NOSTGOPT was in effect. NOSTGOPT was changed in this version so that no optimization of storage or data items occurs even with OPT(2). This is especially helpful for WORKING-STORAGE eye-catchers. (STGOPT)
- RULES:
 - PH04369: RULES(NOEVENPACK) 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®.
 - 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).
 - PH36688: New suboptions LAXREDEF | NOLAXREDEF are added to the RULES option to inform users of redefined items with mismatched lengths.

(RULES)

- SSRANGE: New suboptions MSG and ABD are added to the SSRANGE compiler option to control how the compiler checks reference modification lengths. (SSRANGE)
- TEST:
 - New suboptions SEPARATE and NOSEPARATE are added to the TEST compiler option to control program object size on disk while retaining debugging capability. In addition, new combinations of suboptions are supported in both the TEST and NOTEST compiler options, including TEST(NODWARF), TEST(SEPARATE), and NOTEST(DWARF,SOURCE).
 - 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.
 - 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)

- PI90571: ZONEDATA: 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)

The following compiler options are deprecated:

- PH31500: ZONEDATA: This compiler option is deprecated but tolerated and is automatically mapped to an equivalent form of the new INVDATA compiler option. (ZONEDATA)
- PH41362: NUMCHECK(ZON) : LAXREDEF | STRICTREDEF option is deprecated but is tolerated for compatibility, and it is replaced by the LAX | STRICT option. (NUMCHECK)

Usability enhancements

Improves usability of the compiler in the z/OS UNIX System Services environment with addition of help information for the cob2 compiler invocation command. ([cob2 syntax and options](#))

Listing changes

- Compiler diagnostic messages now appear at the end of the listing, as was the case in COBOL compilers before Enterprise COBOL V5.
- Addition of MD5 signature to program objects and debug data to allow matching of debug data with executables even if a program is recompiled. ([Example: MD5 signature](#))
- Three new fields are added 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.([Example: Program prolog areas](#))

IBM-supplied CICS reserved-word table changes

PI91589: New COBOL words are added to the IBM-supplied CICS reserved-word table. ([CICS reserved-word table](#))

Installation customization changes

- 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](#))

COBOL/JNI interface enhancement

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.

Compiler message changes

- PH08642: Specifying NUMCHECK may cause the compiler to produce the following messages at compile time:
 - IGYCB7313-E ([IGYCB7313-E](#))
 - IGYCB7314-E ([IGYCB7314-E](#))
 - IGYCB7315-W ([IGYCB7315-W](#))
 - IGYCB7316-W ([IGYCB7316-W](#))
- PH42575: The following messages are updated to report additional information and return codes:
 - IGYSI5305-U ([IGYSI5305-U](#))
 - IGYSI5306-U ([IGYSI5306-U](#))

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1991, 2020.

PRIVACY POLICY CONSIDERATIONS:

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, or to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies,"

and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.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.

Other company, product, or service names may be trademarks or service marks of others.

Enterprise COBOL for z/OS publications

COBOL for z/OS publications

You can find the following publications in the [Enterprise COBOL for z/OS library](#):

- *What's new*
- *Customization Guide*, SC27-8712-01
- *Language Reference*, SC27-8713-01
- *Programming Guide*, SC27-8714-01
- *Migration Guide*, GC27-8715-01
- *Performance Tuning Guide*, SC27-9202-00
- *Messages and Codes*, SC27-4648-00
- *Program Directory*, GI13-4526-01
- *Licensed Program Specifications*, GI13-4532-01

Softcopy publications

The following collection kits contain Enterprise COBOL and other product publications. You can find them at <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.

- *z/OS Software Products Collection*
- *z/OS and Software Products DVD Collection*

Support

If you have a problem using Enterprise COBOL for z/OS, see the following site that provides up-to-date support information: https://www.ibm.com/support/home/product/B984385H82239E03/Enterprise_COBOL_for_z/OS.

Index

A

about this information [v](#)

B

Bibliography [13](#)

C

comments
 [sending \[vii\]\(#\)](#)
compiler options
 [OFFSET \[7\]\(#\)](#)
customer support [13](#)

E

enhancements [3](#)

F

feedback
 [sending \[vii\]\(#\)](#)

L

List of resources [13](#)

N

new features [3](#)
Notices [9](#)

P

product support [13](#)

R

RCFs
 [sending \[vii\]\(#\)](#)
reader comments
 [sending \[vii\]\(#\)](#)

S

support [13](#)



Product Number: 5655-EC6