

IBM XL C/C++ for Linux, V13.1.6



What's New for XL C/C++ for Little Endian Distributions

Version 13.1.6

IBM XL C/C++ for Linux, V13.1.6



What's New for XL C/C++ for Little Endian Distributions

Version 13.1.6

Note

Before using this information and the product it supports, read the information in "Notices" on page 27.

First edition

This edition applies to IBM XL C/C++ for Linux, V13.1.6 (Program 5765-J08; 5725-C73) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

© Copyright IBM Corporation 2017.

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

Contents

About this document	v	Chapter 7. Built-in functions	15
Who should read this document.	v	Chapter 8. Compiler options and pragma directives	19
How to send your comments.	v	Chapter 9. Other enhancements	21
Chapter 1. OpenMP API V4.5 support . . .	1	Chapter 10. Available help information	23
Chapter 2. OpenMP Technical Reports support	5	Appendix. Accessibility features for IBM XL C/C++ for Linux	25
Chapter 3. Offloading computations to the NVIDIA GPUs.	7	Notices	27
Chapter 4. OpenMP interoperability with CUDA C/C++ and CUDA Fortran	9	Trademarks	29
Chapter 5. POWER9 technology exploitation.	11	Index	31
Chapter 6. Debugging support	13		

About this document

This information provides an executive overview of new functions in the IBM® XL C/C++ for Linux, V13.1.6 compiler. The topics in this information provide a framework for describing new function in the IBM XL C/C++ for Linux, V13.1.6 compiler. New functions are categorized according to user benefits, such as OpenMP support and offloading computations to the NVIDIA GPUs.

Who should read this document

This information is written primarily for people who are evaluating and planning for IBM XL C/C++ for Linux, V13.1.6.

How to send your comments

Your feedback is important in helping us to provide accurate and high-quality information. If you have any comments about this information or any other XL C/C++ information, send your comments to compinfo@cn.ibm.com.

Be sure to include the name of the manual, the part number of the manual, the version of XL C/C++, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Chapter 1. OpenMP API V4.5 support

IBM XL C/C++ for Linux, V13.1.6 partially supports the OpenMP Application Program Interface Version 4.5 specification. The XL C/C++ implementation is based on IBM's interpretation of the OpenMP Application Program Interface 4.5.

The new OpenMP Application Program Interface Version 4.5 support falls into the following categories:

- “New directives”
- “Expanded functionality for existing directives” on page 2
- “Enhanced built-in functions” on page 3
- “Other enhancements” on page 3

For more information, see the *XL C/C++ Compiler Reference*.

New directives

In addition to the existing OpenMP directives, IBM XL C/C++ for Linux, V13.1.6 adds support for the following directives and their clauses.

omp simd

The **omp simd** directive is applied to a loop to indicate that multiple iterations of the loop can be executed concurrently by using SIMD instructions.

omp for simd

The **omp for simd** directive distributes the iterations of one or more associated loops across the threads that already exist in the team and indicates that the iterations executed by each thread can be executed concurrently using SIMD instructions.

omp distribute simd

The **omp distribute simd** directive distributes loop iterations to each master thread and then executes each set of distributed iterations concurrently by using SIMD instructions.

omp distribute parallel for simd

The **omp distribute parallel for simd** directive distributes loop iterations to each master thread, further redistributes those iterations among the threads of each team, and then applies SIMD vectorization to each iteration.

The following OpenMP combined constructs are also supported.

- **omp parallel for simd**
- **omp target simd**
- **omp target parallel for simd**
- **omp target teams distribute simd**
- **omp target teams distribute parallel for simd**
- **omp teams distribute simd**
- **omp teams distribute parallel for simd**

Expanded functionality for existing directives

The following directives have been updated with functionality for additional clauses in IBM XL C/C++ for Linux, V13.1.6.

omp ordered

The **depend** clause is now supported on the **omp ordered** directive. You can use the **omp ordered depend(source)** or **omp ordered depend(sink : vec)** to specify the order in which threads in a team execute **ordered** regions.

omp target

The **depend**, **is_device_ptr**, and **nowait** clauses are now supported on the **omp target** directive.

- **omp target depend**(*dependence-type:list*) establishes scheduling dependences between the target task and sibling tasks that share list items.
- **omp target is_device_ptr**(*list*) indicates that the data variables in *list* are device pointers that exist within the device data environment.
- **omp target nowait** eliminates the implicit barrier so the parent task can make progress even if the target task is not yet completed.

The **reduction** clause is supported by the combined constructs that consist of **omp target**, such as **omp target parallel for** or **omp target teams distribute parallel for**.

- **reduction**(*reduction-identifier:list*) specifies that for each data variable in *list*, a private copy is created and initialized based on the *reduction-identifier*. At the end of the region, the original data variable is updated with the values of the private copies using a combiner based on the *reduction-identifier*.

omp target data

The **use_device_ptr** clause is now supported on the **omp target data** directive. You can use **omp target data use_device_ptr(list)** to update the pointer variables in *list* with the address of their corresponding variables in the target device data environment.

omp target enter data

The **depend** and **nowait** clauses are now supported on the **omp target enter data** directive.

- **omp target enter data depend**(*dependence-type:list*) establishes scheduling dependences between the target task and sibling tasks that share list items.
- **omp target enter data nowait** enables the target enter data construct to perform asynchronously with respect to the encountering thread.

omp target exit data

The **depend** and **nowait** clauses are now supported on the **omp target exit data** directive.

- **omp target exit data depend**(*dependence-type:list*) establishes scheduling dependences between the target task and sibling tasks that share list items.
- **omp target exit data nowait** enables the target exit data construct to perform asynchronously with respect to the encountering thread.

omp target update

The **depend** and **nowait** clauses are now supported on the **omp target update** directive.

- **omp target update depend**(*dependence-type:list*) establishes scheduling dependences between the target task and sibling tasks that share list items.
- **omp target update nowait** enables the target update construct to execute asynchronously with respect to the encountering thread.

omp task

The **depend** clause is now supported on the **omp task** directive. You can use the **omp task depend**(*dependence-type:list*) to establish scheduling dependences between sibling tasks that share list items.

Updated clauses

ordered_clause

The *n* parameter allows you to specify the number of the loops that are associated with the loop construct. *ordered_clause* is supported on the **omp for** and **omp for simd** directives.

Enhanced built-in functions

omp_target_alloc

The device address returned by **omp_target_alloc** can be used in a **is_device_ptr** clause on the **omp target** directive.

Other enhancements

You can map a lambda functor in OpenMP target regions with IBM XL C/C++ for Linux, V13.1.6.

Chapter 2. OpenMP Technical Reports support

IBM XL C/C++ for Linux, V13.1.6 partially supports the OpenMP Technical Reports augmenting the OpenMP Application Program Interface Version 4.5 specification.

Implicit declare target

If any function is called and defined in the same translation unit, you do not need to explicitly declare it with the **omp declare target** directive.

Pointer mapping for aggregate members

Mapping of a pointer is extended to support mapping a pointer that is a member of a struct and the storage that the pointer points to.

For more information, see Extensions from OpenMP Technical Reports in the *XL C/C++ Optimization and Programming Guide*.

Chapter 3. Offloading computations to the NVIDIA GPUs

IBM XL C/C++ for Linux, V13.1.6 provides the following enhancements that you can benefit from when offloading computations to the NVIDIA GPUs.

Support for CUDA Toolkit 9.0 and 9.1

IBM XL C/C++ for Linux, V13.1.6 requires CUDA Toolkit 9.0 and 9.1. The sm_70 and compute_70 GPU architectures are supported as defined by the CUDA Toolkit.

Specification of GPU architectures for the generated code

You can use the **-qgtarch** option to specify the real or virtual GPU architectures where the code can run, overriding the default GPU architecture. The default GPU architecture is detected at compiler configuration time and encoded into the compiler configuration file.

This allows the compiler to take maximum advantage of the capabilities and machine instructions which are specific to a GPU architecture, or common to a virtual architecture.

Support for the cuda-memcheck tool

You can use the **XLSMPOPTS=cudamemcheckfriendly={off | on}** environment variable to control whether to disable the check for pinned memory in the runtime and allow the program to be executed under the cuda-memcheck tool from the NVIDIA CUDA Toolkit.

Pass LLVM IR bitcode libraries to llvm2ptx

You can specify LLVM IR bitcode libraries, which have a suffix of .bc, on the command line, to pass the LLVM IR bitcode libraries to llvm2ptx, the NVVM-IR to PTX translator.

GPU runtime inlining support for inlining calls made to the OpenMP GPU runtime libraries

This enhancement reduces overhead and significantly improves performance of OpenMP target regions that are offloaded to the accelerator.

Improved GPU code generation

This enhancement applies to several OpenMP directives when contained in an OpenMP target region, most notably parallel loops and reductions.

Pay attention to the following change when you migrate to IBM XL C/C++ for Linux, V13.1.6 from earlier releases.

Changed environment variables

XLSMPOPTS=target=optional is renamed to **XLSMPOPTS=target=default** with the identical functionality.

XLSMPOPTS=target=disable is renamed to **XLSMPOPTS=target=disabled** with the identical functionality.

Related information


- **-qgtarch** in the *XL C/C++ Compiler Reference*
- **XLSMPOPTS** in the *XL C/C++ Compiler Reference*
- Offloading computations to the NVIDIA GPUs in the *XL C/C++ Optimization and Programming Guide*


Chapter 4. OpenMP interoperability with CUDA C/C++ and CUDA Fortran

IBM XL C/C++ for Linux, V13.1.6 provides support for OpenMP interoperability with CUDA C/C++ and CUDA Fortran.

- You can call kernels written in CUDA C/C++ or CUDA Fortran in your OpenMP programs from the host.
- You can use the OpenMP **use_device_ptr** clause to pass OpenMP mapped variables to CUDA kernels that are launched from the host.
- You can use the OpenMP **is_device_ptr** clause to access CUDA device attribute variables or to pass device addresses directly to target regions.

Related information in the *XL C/C++ Compiler Reference*

 **use_device_ptr**

 **is_device_ptr**

Chapter 5. POWER9 technology exploitation

Use these enhancements to exploit the POWER9 technology with IBM XL C/C++ for Linux, V13.1.6.

MASS libraries

The newly added vector library `libmassvp9.a` and the SIMD library `libmass_simdp9.a` contain functions that are tuned for the POWER9 architecture.

For more information, see *Using the Mathematical Acceleration Subsystem (MASS) libraries in the XL C/C++ Optimization and Programming Guide*.

New built-in functions

For built-in functions that are new or changed for IBM XL C/C++ for Linux, V13.1.6, see Chapter 7, “Built-in functions,” on page 15.

Chapter 6. Debugging support

IBM XL C/C++ for Linux, V13.1.6 provides these debugging enhancements.

DWARF debugging information in separate .dwo files

You can use the **-gsplit-dwarf** option to enable the DWARF debugging information to be generated in one or more separate .dwo files. For more information, see **-gsplit-dwarf** in the *XL C/C++ Compiler Reference*.

Support for debug at opt

You can use **-g8** or **-g9** to get general debug at optimization support with optimization level **-O2**, which provide the same functionality as provided in IBM XL C/C++ for Linux for big endian distributions. When the **-O2** optimization level is in effect, you can use **-g8** to generate read-only debugging information about line numbers, source file names, and variables. Beyond that, if you want to modify the value of the variables in the debugger, use **-O2 -g9**.

For more information, see **-g** in the *XL C/C++ Compiler Reference*.

Support for inlined functions debugging

The support for debugging of inlined functions includes, for example, stopping, showing call stack, and showing parameters and local variables.

- When no optimization is enabled (**-qnoopt**), functions that are specified with attribute `__attribute__((__always_inline__))` can still be debugged, although they are inlined.
- You can compile your program with **-O2 -g8** or **-O2 -g9** to get debug at optimization support for inlined functions.

Chapter 7. Built-in functions

This section describes built-in functions that are new or changed for IBM XL C/C++ for Linux, V13.1.6.

For more information about each built-in function, see the *XL C/C++ Compiler Reference*.

New built-in functions

Note: The following new built-in functions are valid only when `-qarch(-mcpu)` is set to utilize the POWER9 technology.

`__builtin_addex`

Add Extended using Overflow Bit.

`__builtin_bcdcopysign`

Returns the decimal value of the first parameter combined with the sign code of the second parameter.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_bcdsetsign`

Set the sign code of the input parameter in packed decimal format.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_bcdshift`

Decimal shift.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_bcdshiftround`

Decimal shift and round.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_bcdtruncate`

Decimal truncate.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_bcdunsignedshift`

Decimal unsigned shift.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_bcdunsignedtruncate`

Decimal unsigned truncate.

This built-in function is valid only when you include the `altivec.h` file.

`__builtin_cmpeqb`

Compares the corresponding bytes of the given parameters and returns the result.

`__builtin_cmprb`

Compares the ranged byte.

`__builtin_compare_exp_uo`, `__builtin_compare_exp_lt`, `__builtin_compare_exp_eq`, `__builtin_compare_exp_gt`

Compares the exponents of two parameters.

__builtin_darn, __builtin_darn_32, __builtin_darn_raw
 Delivers a random number.

__builtin_extract_exp
 Returns the exponent of the given parameter.

__builtin_extract_sig
 Returns the significand of the given parameter.

__builtin_insert_exp
 Replaces the exponent of the first parameter with the second parameter and returns the result.

__builtin_maddhd
 Fixed-point Multiply-Add signed high doubleword.

__builtin_maddhdu
 Fixed-point Multiply-Add high doubleword unsigned.

__builtin_maddld
 Fixed-point Multiply-Add low doubleword.

__builtin_national2packed
 Transforms the national decimal value to packed decimal format.
 This built-in function is valid only when you include the `altivec.h` file.

__builtin_packed2national
 Transforms the packed decimal value to national decimal format.
 This built-in function is valid only when you include the `altivec.h` file.

__builtin_packed2zoned
 Transforms the packed decimal value to zoned decimal format.
 This built-in function is valid only when you include the `altivec.h` file.

__builtin_setb
 Sets boolean extension.

__builtin_zoned2packed
 Transforms the zoned decimal value to packed decimal format.
 This built-in function is valid only when you include the `altivec.h` file.

vec_extsbd
 Sign-extends the rightmost byte of each doubleword element of the given vector.

vec_extsbw
 Sign-extends the rightmost byte of each word element of the given vector.

vec_extshd
 Sign-extends the rightmost halfword of each doubleword element of the given vector.

vec_extshw
 Sign-extends the rightmost halfword of each word element of the given vector.

vec_extswd
 Sign-extends the rightmost word of each doubleword element of the given vector.

vec_load_splats

Loads a 4-byte element from the memory address specified by the displacement *a* and the pointer *b* and then splats it onto a vector.

vec_slv

Left-shifts the elements of a given vector by a given number of bits.

vec_srv

Right-shifts the elements of a given vector by a given number of bits.

vec_xl_len_r

Loads a string of bytes into vector register, right-justified. Sets the leftmost elements (*16-cnt*) to 0.

vec_xst_len_r

Stores a right-justified string of bytes.

Changed built-in functions

You must include `altivec.h` to use the following built-in functions.

- BCD add and subtract functions
- BCD comparison functions
- BCD load and store functions
- BCD test add and subtract for overflow functions
- Vector built-in functions

Chapter 8. Compiler options and pragma directives

This topic describes new or changed compiler options and pragma directives.

You can specify compiler options on the command line. You can also modify compiler behavior through pragma directives embedded in your application source files. For detailed descriptions and usage information for XL C/C++ compiler options and pragma directives, see the *XL C/C++ Compiler Reference*.

New compiler options

-g Two suboptions **-g8** and **-g9** are added for debug at opt support. When the **-O2** optimization level is in effect, you can use **-g8** to generate read-only debugging information about line numbers, source file names, and variables. Beyond that, if you want to modify the value of the variables in the debugger, use **-O2 -g9**.

-gsplit-dwarf

The GCC option **-gsplit-dwarf** option is supported to enable the DWARF debugging information to be generated in one or more separate .dwo files. For a complete list of the GCC options that are supported by IBM XL C/C++ for Linux, see Supported GCC options.

GPU -qgtarch

The **-qgtarch** option is added to embed the PTX code for virtual GPU architectures and the compiled code images for real GPU architectures into the object files or executables. **GPU**

Changed compiler options

-qaltivec

Starting from IBM XL C/C++ for Linux V13.1.6, the **-qaltivec** option takes effect only when you include the `altivec.h` file.

-qx1compatmacros

The default value is changed from **-qx1compatmacros** to **-qnox1compatmacros** in order to facilitate building popular open source packages.

To define the these legacy macros: `__xlc__`, `__xlc_ver__`,

C++ `__IBMCPP__` **C++**, **C** `__IBMC__`, and `__xlc__` **C**, you must use **-qx1compatmacros**.

New pragma directives

IBM XL C/C++ for Linux, V13.1.6 adds support for the pragma directives from the OpenMP Application Program Interface Version 4.5 specification. For more information, see Chapter 1, "OpenMP API V4.5 support," on page 1.

Chapter 9. Other enhancements

This section describes other features and enhancements in IBM XL C/C++ for Linux, V13.1.6.

IBM Advance Toolchain 11.0 support

IBM XL C/C++ for Linux, V13.1.6 now fully supports IBM Advance Toolchain 11.0, which is a set of open source development tools and runtime libraries. With IBM Advance Toolchain, you can take advantage of the latest POWER[®] hardware features on Linux, especially the tuned libraries. To use the Advance Toolchain, see Using IBM XL C/C++ for Linux, V13.1.6 with Advance Toolchain in the *XL C/C++ Compiler Reference*.

New xlc_configure options

-binutils *path*

Specifies the path where the binary utilities (binutils) are installed.

-use_dt_rpath

Determines whether the compiler uses the DT_RPATH or DT_RUNPATH property to encode shared library load paths into your program.

For more information about the options, see xlc_configure options in the *XL C/C++ Installation Guide*.

Chapter 10. Available help information

IBM XL C/C++ information

XL C/C++ provides product information in the following formats:

- Quick Start Guide

The Quick Start Guide (`quickstart.pdf`) is intended to get you started with IBM XL C/C++ for Linux, V13.1.6. It is located by default in the XL C/C++ directory and in the `\quickstart` directory of the installation DVD.

- README files

README files contain late-breaking information, including changes and corrections to the product information. README files are located by default in the XL C/C++ directory, and in the root directory and subdirectories of the installation DVD.

- Installable man pages

Man pages are provided for the compiler invocations and all command-line utilities provided with the product. Instructions for installing and accessing the man pages are provided in the *IBM XL C/C++ for Linux, V13.1.6 Installation Guide*.

- Online product documentation

The fully searchable HTML-based documentation is viewable in IBM Knowledge Center at http://www.ibm.com/support/knowledgecenter/SSXVZZ_13.1.6/com.ibm.compilers.linux.doc/welcome.html.

- PDF documents

PDF documents are available on the web at https://www.ibm.com/support/knowledgecenter/SSXVZZ_13.1.6/com.ibm.compilers.linux.doc/download_pdf.html.

The following files comprise the full set of XL C/C++ product information.

Note: To ensure that you can access cross-reference links to other XL C/C++ PDF documents, download and unzip the .zip file that contains all the product documentation files, or you can download each document into the same directory on your local machine.

Table 1. XL C/C++ PDF files

Document title	PDF file name	Description
<i>What's New for IBM XL C/C++ for Linux, V13.1.6, GC27-8035-00</i>	<code>whats_new.pdf</code>	Provides an executive overview of new functions in the IBM XL C/C++ for Linux, V13.1.6 compiler, with new functions categorized according to user benefits.
<i>Getting Started with IBM XL C/C++ for Linux, V13.1.6, GI13-2875-05</i>	<code>getstart.pdf</code>	Contains an introduction to XL C/C++, with information about setting up and configuring your environment, compiling and linking programs, and troubleshooting compilation errors.
<i>IBM XL C/C++ for Linux, V13.1.6 Installation Guide, GC27-6540-05</i>	<code>install.pdf</code>	Contains information for installing XL C/C++ and configuring your environment for basic compilation and program execution.

Table 1. XL C/C++ PDF files (continued)

Document title	PDF file name	Description
<i>IBM XL C/C++ for Linux, V13.1.6 Migration Guide, GC27-8036-00</i>	migrate.pdf	Contains migration considerations for using XL C/C++ to compile programs that were previously compiled on different platforms, by previous releases of XL C/C++, or by other compilers.
<i>IBM XL C/C++ for Linux, V13.1.6 Compiler Reference, SC27-6570-05</i>	compiler.pdf	Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions.
<i>IBM XL C/C++ for Linux, V13.1.6 Language Reference, SC27-6550-05</i>	langref.pdf	Contains information about language extensions for portability and conformance to nonproprietary standards.
<i>IBM XL C/C++ for Linux, V13.1.6 Optimization and Programming Guide, SC27-6560-05</i>	proguide.pdf	Contains information about advanced programming topics, such as application porting, interlanguage calls with Fortran code, library development, application optimization, and the XL C/C++ high-performance libraries.

To read a PDF file, use Adobe Reader. If you do not have Adobe Reader, you can download it (subject to license terms) from the Adobe website at <http://www.adobe.com>.

More information related to XL C/C++, including IBM Redbooks® publications, white papers, and other articles, is available on the web at <http://www.ibm.com/support/docview.wss?uid=swg27036675>.

For more information about the compiler, see the XL compiler on Power® community at <http://ibm.biz/xl-power-compilers>.

Other IBM information

- *ESSL product documentation* available at http://www.ibm.com/support/knowledgecenter/SSFHY8/essl_welcome.html?lang=en

Other information

- *Using the GNU Compiler Collection* available at <http://gcc.gnu.org/onlinedocs>

Appendix. Accessibility features for IBM XL C/C++ for Linux

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

Accessibility features

IBM XL C/C++ for Linux uses the latest W3C Standard, WAI-ARIA 1.0 (<http://www.w3.org/TR/wai-aria/>), to ensure compliance to US Section 508 (<http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards>) and Web Content Accessibility Guidelines (WCAG) 2.0 (<http://www.w3.org/TR/WCAG20/>). To take advantage of accessibility features, use the latest release of your screen reader in combination with the latest web browser that is supported by this product.

The IBM XL C/C++ for Linux online product documentation in IBM Knowledge Center is enabled for accessibility. The accessibility features of IBM Knowledge Center are described at IBM Knowledge Center help.

Keyboard navigation

This product uses standard navigation keys.

Interface information

You can use speech recognition software like a Text-to-speech (TTS) tool to view the output generated by the compiler.

The IBM XL C/C++ for Linux online product documentation is available in IBM Knowledge Center, which is viewable from a standard web browser.

PDF files have limited accessibility support. With PDF documentation, you can use optional font enlargement, high-contrast display settings, and can navigate by keyboard alone.

To enable your screen reader to accurately read syntax diagrams, source code examples, and text that contains the period or comma PICTURE symbols, you must set the screen reader to speak all punctuation.

Related accessibility information

To learn the accessibility features of the operation systems that are supported by IBM XL C/C++ for Linux, see the following information:

- Ubuntu (<https://help.ubuntu.com/community/Accessibility>)
- SUSE Linux Enterprise Server (<https://www.suse.com>)
- Red Hat Enterprise Linux (<http://www.redhat.com>)
- Community Enterprise Operating System (<https://www.centos.org/>)

In addition to standard IBM help desk and support websites, IBM has established a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

TTY service
800-IBM-3383 (800-426-3383)
(within North America)

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see IBM Accessibility (www.ibm.com/able).

Notices

Programming interfaces: Intended programming interfaces allow the customer to write programs to obtain the services of IBM XL C/C++ for Linux.

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.

Licenses 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:

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
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. 1998, 2017.

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 <http://www.ibm.com/legal/copytrade.shtml>.

Adobe and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

NVIDIA is either registered trademark or trademark of NVIDIA Corporation in the United States, other countries, or both.

Index

A

accessibility
 interfaces 25
 keyboard navigation 25

B

built-in functions
 new features
 V13.1.6 15

C

compiler options
 new or changed features
 V13.1.6 19

M

more enhancements
 new features
 V13.1.6 21



Product Number: 5765-J08; 5725-C73

Printed in USA

GC27-8035-00

