

z/OS
Version 2 Release 3

*Getting Started
with XL C/C++ V2.3.1 for z/OS V2.3*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 9.](#)

This edition applies to Version 2 Release 3 Modification 1 of IBM® z/OS® XL C/C++ (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2019-03-24

© **Copyright International Business Machines Corporation 2019.**

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

Contents

- Chapter 1. About this document..... 1**
- Chapter 2. Overview of IBM XL C/C++ V2.3.1 for z/OS V2.3..... 3**
- Chapter 3. Downloading and installing IBM XL C/C++ V2.3.1 for z/OS V2.3..... 5**
- Chapter 4. Compiling using IBM XL C/C++ V2.3.1 for z/OS V2.3..... 7**
- Notices..... 9**
 - Trademarks..... 9
 - Standards..... 10

Chapter 1. About this document

This information supports IBM z/OS XL C/C++ (5650-ZOS) and contains information about IBM XL C/C++ V2.3.1 for z/OS V2.3.

This information is primarily intended as a quick start guide for the IBM XL C/C++ V2.3.1 for z/OS V2.3 compiler. It provides information about the compiler overview and how to compile applications written in C and C++.

Softcopy documents

The XL C/C++ V2.3.1 for z/OS V2.3 publications are supplied in PDF format and available for download from the [z/OS XL C/C++ documentation library](#).

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

Technical support

Additional technical support is available at the [z/OS XL C/C++ Support page \(www.ibm.com/support/home/product/K673111E24114C80/z_OS_XL_C/C++\)](http://www.ibm.com/support/home/product/K673111E24114C80/z_OS_XL_C/C++). This page provides a portal with search capabilities to a large selection of technical support FAQs and other support documents.

For the latest information about z/OS XL C/C++, visit [Marketplace page for z/OS XL C/C++ \(www.ibm.com/us-en/marketplace/xl-cpp-compiler-zos\)](http://www.ibm.com/us-en/marketplace/xl-cpp-compiler-zos)

For information about boosting performance, productivity and portability, visit [C/C++ Compilers for IBM Z - Community & Forum](#).

If you cannot find what you need, you can e-mail: compinfo@cn.ibm.com

Chapter 2. Overview of IBM XL C/C++ V2.3.1 for z/OS V2.3

IBM XL C/C++ V2.3.1 for z/OS V2.3 provides C and C++ compilers that adopt the Clang infrastructure from the LLVM open source community for a portion of the compilers. These compilers add support for the core C11 standard and most of the C++11 standard alongside the language standards supported in the z/OS XL C/C++ V2.3 compiler for easier application migration to IBM Z.

XL C/C++ V2.3.1 for z/OS V2.3 is designed to aid in porting code from other platforms and to give a more familiar view to those who are accustomed to a UNIX environment. As such, the compilers are supported on z/OS UNIX Systems Services only. They support EBCDIC and ASCII execution character sets and generate AMODE 64 code. They can be invoked in a z/OS UNIX shell session or OMVS session. The invocation utility, configuration file, and system header files are all located in z/OS UNIX. For more information about utilities and binder, refer to *z/OS XL C/C++ User's Guide, V2.3* (https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.cbcux01/abstract.htm).

The code generated by XL C/C++ V2.3.1 for z/OS V2.3 is Language Environment® based, using 64-bit addressing and XPLINK linkage. This allows mixing C and C++ code easily while providing the performance benefits of XPLINK.

XL C/C++ V2.3.1 for z/OS V2.3 and the new C++ library provides support for the core C11 standard and most of the C++11 standard along with a number of commonly used extensions. You can modernize your C and C++ development on z/OS for improved platform performance with C11 and C++11 language standard features.

Note: The C++ language standard (in 2011) broke binary compatibility in the C++ library with previous C++ language standards, and so objects compiled with z/OS XL C/C++ V2.3 or previous releases will not be compatible with objects generated by XL C/C++ V2.3.1 for z/OS V2.3. Although this breakage requires additional migration efforts, the XL C/C++ V2.3.1 for z/OS V2.3 compiler also takes this breakage as an opportunity to modernize other aspects of the C++ implementation.

Chapter 3. Downloading and installing IBM XL C/C++ V2.3.1 for z/OS V2.3

XL C/C++ V2.3.1 for z/OS V2.3 is for clients that have enabled the z/OS XL C/C++ compiler (an optionally priced feature) on z/OS V2.3 only.

Downloading IBM XL C/C++ V2.3.1 for z/OS V2.3

XL C/C++ V2.3.1 for z/OS V2.3 is a web deliverable that can be downloaded at [z/OS Downloads website](http://www.ibm.com/systems/z/os/zos/downloads/) (<http://www.ibm.com/systems/z/os/zos/downloads/>).

Installing IBM XL C/C++ V2.3.1 for z/OS V2.3

Program Directory for z/OS V2R3M1 XL C/C++ (<http://publibfp.dhe.ibm.com/epubs/pdf/i1343900.pdf>) is intended for the system programmers who are responsible for program installation and maintenance. It contains information about the material and procedures associated with the installation of XL C/C++ V2.3.1 for z/OS V2.3.

Chapter 4. Compiling using IBM XL C/C++ V2.3.1 for z/OS V2.3

The XL C/C++ V2.3.1 for z/OS V2.3 compiler introduces two new invocation commands, `xlclang` and `xlclang++`, to invoke the new Clang-based compilers for your C and C++ source code. You can compile with any supported compiler options that are needed.

To ease your migration, most IBM z/OS XL C/C++ compiler options that are available in the z/OS XL C/C++ V2.3 compiler front end, including the `-q` options and flag options, are selectively supported by the Clang-based front end of XL C/C++ V2.3.1 for z/OS V2.3. In addition, some GCC or Clang options are supported by the Clang-based front end as well, which means fewer migration effort is needed when you migrate your GCC or Clang programs to XL C/C++ V2.3.1 for z/OS V2.3. You are encouraged to use GCC or Clang options instead of equivalent z/OS XL C/C++ V2.3 compiler options with the XL C/C++ V2.3.1 for z/OS V2.3 Clang-based front end wherever possible.

Some IBM pragmas that are supported in z/OS XL C/C++ V2.3 and previous releases are not supported in the Clang-based front end introduced in XL C/C++ V2.3.1 for z/OS V2.3. Therefore, if you want to compile with the Clang-based front end, you need to modify your programs that were previously written for and compiled by earlier releases than XL C/C++ V2.3.1 for z/OS V2.3.

Examples

The following source code examples show simple C and C++ source code files and the typical invocation commands used to compile them.

Example 1: Main program (C): `validateArguments.c`

```
#include <stdlib.h>

int lookupTable[11];

int getRC(int factor1, int factor2);

int main(int argc, char* argv[]) {
    // Validate the argument count
    if (argc < 3) {
        return 1;
    }
    if (argc > 3) {
        return 2;
    }

    // Initialize the global data
    for (int i = 0; i < 11; i++) {
        lookupTable[i] = i;
    }

    // Convert the arguments to numbers
    // Note: argv[0] is the executable name
    int factor1 = atoi(argv[1]);
    int factor2 = atoi(argv[2]);
    return getRC(factor1, factor2);
}
```

The source can be compiled as follows:

```
xlclang -c -Wall -qdebug validateArguments.c
```

The `xlclang` invocation command here is compiling the C source code. The `-Wall` argument (option) to the compiler instructs it to warn of possible code issues. Note that this option uses a different form of option specification than the z/OS XL C/C++ V2.3 compiler. The `-qdebug` option instructs the compiler to generate debugging information for the compiled module. This option uses the same form as the z/OS XL C/C++ V2.3 compiler. The XL C/C++ V2.3.1 for z/OS V2.3 compiler generally accepts options of both forms.

Example 2: Linked function (C++): intMap.C

```
extern int lookupTable[11];

extern "C" {
    int getRC(int index1, int index2);
}

int getRC(int index1, int index2) {
    for (int i : lookupTable) {
        if (i == index1) {
            return lookupTable[i] * lookupTable[index2];
        }
    }
    return -1;
}
```

The source can be compiled with the `xlcclang++` invocation command as follows:

```
xlcclang++ -c -Wall -qdebug intMap.C
```

Generally you should use the `xlcclang++` invocation command when compiling C++ code. This allows binding to the C++ library, if it was used, as specified in the invocation stanza in the compiler configuration file. If you need to use `xlcclang++` to compile source code with the `.c` suffix, you need to specify the `-+` option so that all C source files are to be recognized as C++ source files. An example is as follows:

```
xlcclang++ -c -Wall -qdebug -+ validateArguments.c
```

You can then link using `xlcclang++` as follows:

```
xlcclang++ validateArguments.o intMap.o -o myMultiply
```

The `xlcclang++` invocation command binds (or links) the two object files, `validateArguments.o` (C based) and `intMap.o` (C++ based), into a single executable file `myMultiply`, which can be invoked as follows:

```
myMultiply 9 5
```

You can use the command `echo $?` to view the return code of `myMultiply`, which is 0.

For detailed information about compiling and linking applications using the XL C/C++ V2.3.1 for z/OS V2.3 compiler, see *Compiler Reference for XL C/C++ V2.3.1 for z/OS V2.3*.

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 Corporation
J74/G4
555 Bailey Avenue
San Jose, CA 95141-1099
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.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711

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

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 \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Standards

The following standards are supported:

- The C language is consistent with *Programming languages - C (ISO/IEC 9899:1999)* and a subset of *Programming languages - C (ISO/IEC 9899:2011)*. For more information, see [International Organization for Standardization \(ISO\) \(www.iso.org\)](http://www.iso.org).
- The C++ language is consistent with *Programming languages - C++ (ISO/IEC 14882:1998)*, *Programming languages - C++ (ISO/IEC 14882:2003(E))*, and a subset of *Programming languages - C++ (ISO/IEC 14882:2011)*.

The following standards are supported in combination with the z/OS UNIX System Services element:

- A subset of *IEEE Std. 1003.1-2001 (Single UNIX Specification, Version 3)*. For more information, see [IEEE \(www.ieee.org\)](http://www.ieee.org).
- *IEEE Std 1003.1–1990, IEEE Standard Information Technology—Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API) [C language]*, copyright 1990 by the Institute of Electrical and Electronic Engineers, Inc.
- The core features of *IEEE P1003.1a Draft 6 July 1991, Draft Revision to Information Technology—Portable Operating System Interface (POSIX), Part 1: System Application Program Interface (API) [C Language]*, copyright 1992 by the Institute of Electrical and Electronic Engineers, Inc.
- *IEEE Std 1003.2–1992, IEEE Standard Information Technology—Portable Operating System Interface (POSIX)—Part 2: Shells and Utilities*, copyright 1990 by the Institute of Electrical and Electronic Engineers, Inc.
- The core features of *IEEE Std P1003.4a/D6–1992, IEEE Draft Standard Information Technology—Portable Operating System Interface (POSIX)—Part 1: System Application Program Interface (API)—Amendment 2: Threads Extension [C language]*, copyright 1990 by the Institute of Electrical and Electronic Engineers, Inc.
- The core features of *IEEE 754-1985 (R1990) IEEE Standard for Binary Floating-Point Arithmetic (ANSI)*, copyright 1985 by the Institute of Electrical and Electronic Engineers, Inc.
- *X/Open CAE Specification, System Interfaces and Headers, Issue 4 Version 2*, copyright 1994 by The Open Group
- *X/Open CAE Specification, Networking Services, Issue 4*, copyright 1994 by The Open Group
- United States Government's *Federal Information Processing Standard (FIPS) publication for the programming language C, FIPS-160*, issued by National Institute of Standards and Technology, 1991



SC27-9419-00

