Program Directory for
IBM Open XL C/C++ 2.2 for z/OS

Program Number 5655-ZOS

FMIDs HLB77C4

for Use with
z/OS 3.1
z/OS 3.2

Document Date: October 2025

GI13-5701-03

Note

Before using this information and the product it supports, be sure to read the general information under 7.0, “Notices” on page 28.

Contents

Figures

1.0 Introduction

This program directory 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 the IBM Open XL C/C++ 2.2 for z/OS. This publication refers to IBM Open XL C/C++ 2.2 for z/OS as Open XL C/C++ 2.2.

The Program Directory contains the following sections:

- 2.0, “Program Materials” on page 2 identifies the basic program materials and documentation for Open XL C/C++ 2.2.
- 3.0, “Program Support” on page 4 describes the IBM support available for Open XL C/C++ 2.2.
- 4.0, “Program and Service Level Information” on page 5 lists the APARs (program level) and PTFs (service level) that have been incorporated into Open XL C/C++ 2.2.
- 5.0, “Installation Requirements and Considerations” on page 6 identifies the resources and considerations that are required for installing and using Open XL C/C++ 2.2.
- 6.0, “Installation Instructions” on page 17 provides detailed installation instructions for Open XL C/C++ 2.2. It also describes the procedures for activating the functions of Open XL C/C++ 2.2, or refers to appropriate publications.

1.1 Open XL C/C++ 2.2 Description

IBM Open XL C/C++ 2.2 for z/OS combines the benefits and innovations from the LLVM community with IBM XL C/C++ compiler technology to deliver leading-edge application performance for the latest z17 architecture. Open XL C/C++ 2.2 for z/OS introduces support for integrated CICS translator and IMS subsystems for clients to protect and leverage investments on IBM Z and reduce business and IT risks. Open XL C/C++ 2.2 also adds support for the C++20 language standard to improve compatibility of C++ applications across platforms, and is designed for easy migration of C and C++ applications from distributed platforms to z/OS. Usability has been enhanced with the addition of debug information storing in side-file as well as the initial rollout of listing file support to Open XL C/C++. Open XL C/C++ 2.2 for z/OS is available at no additional charge for clients that have enabled the optionally priced XL C/C++ compiler on z/OS. It is available as a web deliverable from the IBM C/C++ for z/OS website.

1.2 Open XL C/C++ 2.2 FMIDs

Open XL C/C++ 2.2 consists of the following FMIDs:

HLB77C4 (OPEN XL C/C++)

2.0 Program Materials

An IBM program is identified by a program number. The program number for Open XL C/C++ 2.2 is 5655-ZOS on z/OS 3.1 and 3.2. There is no feature number for Open XL C/C++ 2.2 as it is being delivered through the web download site.

Basic Machine-Readable Materials are materials that are supplied under the base license and are required for the use of the product.

2.1 Basic Machine-Readable Material

The distribution medium for this program is web delivery. The code for this deliverable can be downloaded from z/OS Downloads at https://www-40.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosDownloads/.

The deliverable contains all the programs and data needed for installation. This program is in SMP/E RELFILE format and is installed by using SMP/E. See 6.0, “Installation Instructions” on page 17 for more information about how to install the program.

2.2 Optional Machine-Readable Material

No optional machine-readable materials are provided for Open XL C/C++ 2.2.

2.3 Program Publications

This section identifies the basic and optional publications for Open XL C/C++ 2.2.

2.3.1 Basic Program Publications

Figure 1 identifies the basic unlicensed publications for Open XL C/C++ 2.2.

Figure 1 (Page 1 of 2). Basic Material: Unlicensed

Publication Title	Form Number	Media Format
What's new for IBM Open XL C/C++ 2.2 for z/OS	SC28-4206-00	
Migration Guide for IBM Open XL C/C++ 2.2 for z/OS	SC28-4205-00	
Compiler Reference for IBM Open XL C/C++ 2.2 for z/OS	SC28-4204-00	
z/OS C/C++ Runtime Library Reference	SC14-7314-60	
Common Debug Architecture Library Reference	SC14-7311-40	

Figure 1 (Page 2 of 2). Basic Material: Unlicensed		
Publication Title	**Form Number**	**Media Format**
DWARF/ELF Extensions Library Reference	SC14-7312-40	

Note: You can obtain softcopy formats from the following sources:

- IBM Resource Link at http://www.ibm.com/servers/resourcelink
- IBM Publications Center website at https://www.ibm.com/resources/publications
- IBM Open XL C/C++ and XL C/C++ for z/OS documentation library at https://www.ibm.com/support/pages/node/602707

Regardless of the method you use to obtain the publications, make sure you have the correct level of the documentation (as shown in the final two digits of the order number in the above table).

2.3.2 Optional Program Publications

No optional publications are provided for Open XL C/C++ 2.2.

2.4 Program Source Materials

No program source materials or viewable program listings are provided for Open XL C/C++ 2.2.

2.5 Publications Useful During Installation

You might want to use the publications listed in Figure 2 during the installation of Open XL C/C++ 2.2.

Figure 2. Publications Useful During Installation	
Publication Title	**Form Number**
IBM SMP/E for z/OS User's Guide	SA23-2277
IBM SMP/E for z/OS Commands	SA23-2275
IBM SMP/E for z/OS Reference	SA23-2276
IBM SMP/E for z/OS Messages, Codes, and Diagnosis	GA32-0883

3.0 Program Support

This section describes the IBM support available for Open XL C/C++ 2.2.

3.1 Program Services

Contact your IBM representative for specific information about available program services.

3.2 Statement of Support Procedures

Report any problems which you feel might be an error in the product materials to your IBM Support Center. You may be asked to gather and submit additional diagnostics to assist the IBM Support Center in their analysis.

Figure 3 lists the component IDs (COMPID) for Open XL C/C++ 2.2.

Figure 3. Component IDs

FMID	COMPID	Component Name	RETAIN Release
HLB77C4	56551210A	OPEN XL C/C++	7C4

4.0 Program and Service Level Information

This section identifies the program and relevant service levels of Open XL C/C++ 2.2. The program level refers to the APAR fixes that have been incorporated into the program. The service level refers to the PTFs that have been incorporated into the program.

4.1 Program Level Information

The following APAR fixes against previous releases of Open XL C/C++ 2.2 have been incorporated into this release. They are listed by FMID.

- HLB77C3

  PH61561 PH62174 PH63025 PH63130 PH63131 PH63132 PH63133 PH63142 PH63143 PH63144 PH63182 PH65242 PH65909 PH66037 PH66038 PH66039 PH66040 PH66041 PH66096 PH66927 PH66959

4.2 Service Level Information

No PTFs against this release of Open XL C/C++ 2.2 have been incorporated into the product package.

4.3 Understanding Open XL C/C++ 2.2 Support Service

You did not receive any service with your deliverable.

5.0 Installation Requirements and Considerations

The following sections identify the system requirements for installing and activating Open XL C/C++ 2.2. The following terminology is used:

- *Driving system*: the system on which SMP/E is executed to install the program.

  The program might have specific operating system or product level requirements for using processes, such as binder or assembly utilities during the installation.

- *Target system*: the system on which the program is configured and run.

  The program might have specific product level requirements, such as needing access to the library of another product for link-edits. These requirements, either mandatory or optional, might directly affect the element during the installation or in its basic or enhanced operation.

In many cases, you can use a system as both a driving system and a target system. However, you can make a separate IPL-able clone of the running system to use as a target system. The clone must include copies of all system libraries that SMP/E updates, copies of the SMP/E CSI data sets that describe the system libraries, and your PARMLIB and PROCLIB.

Use separate driving and target systems in the following situations:

- When you install a new level of a product that is already installed, the new level of the product will replace the old one. By installing the new level onto a separate target system, you can test the new level and keep the old one in production at the same time.
- When you install a product that shares libraries or load modules with other products, the installation can disrupt the other products. By installing the product onto a separate target system, you can assess these impacts without disrupting your production system.

5.1 Driving System Requirements

This section describes the environment of the driving system required to install Open XL C/C++ 2.2.

5.1.1 Machine Requirements

The driving system can run in any hardware environment that supports the required software.

5.1.2 Programming Requirements

Figure 4. Driving System Software Requirements

Program Number	Product Name	Minimum VRM	Minimum Service Level will satisfy these APARs	Included in the shipped product?
Any **one** of the following:				
5655-ZOS	z/OS	3.1.0	N/A	No
5655-ZOS	z/OS	3.2.0	N/A	No

Note: Installation might require migration to new z/OS releases to be service supported. See z/OS Support Lifecycle at https://www.ibm.com/support/pages/lifecycle/search/?q=5655-zos for z/OS 3.1 and z/OS 3.2.

Open XL C/C++ 2.2 is installed into a file system, either HFS or zFS. Before installing Open XL C/C++ 2.2, you must ensure that the target system file system data sets are available for processing on the driving system. OMVS must be active on the driving system and the target system file data sets must be mounted on the driving system.

If you plan to install Open XL C/C++ 2.2 in a zFS file system, this requires that zFS must be active on the driving system. Information on activating and using zFS can be found in z/OS Distributed File Service zSeries File System Administration, SC24-5989.

5.2 Target System Requirements

This section describes the environment of the target system required to install and use Open XL C/C++ 2.2.

Open XL C/C++ 2.2 installs in the z/OS (Z038) SREL.

5.2.1 Machine Requirements

The target system can run in any hardware environment that supports the required software.

5.2.2 Programming Requirements

5.2.2.1 Installation Requisites: Installation requisites identify products that are required and *must* be present on the system or products that are not required but *should* be present on the system for the successful installation of this product.

Mandatory installation requisites identify products that are required on the system for the successful installation of this product. These products are specified as PREs or REQs.

Figure 5. Target System Mandatory Installation Requisites

Program Number	Product Name	Minimum VRM	Minimum Service Level will satisfy these APARs	Included in the shipped product?
5655-ZOS	z/OS	3.1.0	N/A	No
5655-ZOS	z/OS	3.2.0	N/A	No

Note: Installation might require migration to new z/OS releases to be service supported.

To download IBM Open XL C/C++ 2.2 for z/OS see https://www.ibm.com/products/xl-cpp-compiler-zos/.

Conditional installation requisites identify products that are *not* required for successful installation of this product but can resolve such things as certain warning messages at installation time.

Note: The Open XL C/C++ 2.2 installation verification program (IVP) job uses the temporary directory /tmp as work directory for processing inputs and outputs. Ensure that the file system (either HFS or ZFS) is available on the target system before you run the job. Refer 6.1.11, "Run the Installation Verification Programs" on page 25 and the sample jobs, CNWJIVP1 and CNWJIVP2 for additional information.

5.2.2.2 Operational Requisites:

Operational requisites are products that are required and *must* be present on the system or products that are not required but *should* be present on the system for this product to operate all or part of its functions.

Mandatory operational requisites identify products that are required for this product to operate its basic functions.

Figure 6 (Page 1 of 2). Target System Mandatory Operational Requisites

Program Number	Product Name and Minimum VRM/Service Level	Function
5655-ZOS	Language Environment for z/OS 3.1.0 or 3.2.0	
5655-ZOS	z/OS UNIX System Service for z/OS 3.1.0 or 3.2.0	
5650-ZOS	Runtime Library Extensions for z/OS 2.4.0 with APARs(PTFs) PH44479(UI80106) applied.	OpenBLAS support for IBM Open XL C/C++ 2.2 for z/OS

Figure 6 (Page 2 of 2). Target System Mandatory Operational Requisites

Program Number	Product Name and Minimum VRM/Service Level	Function
5655-ZOS	Language Environment for z/OS 3.1.0 with these APARs(PTFs) applied: • PH53938(UI94523) • PH60056(UI95833) • PH62468(UO04851) • PH68179(UO04934)	• FASTLINK support for IBM Open XL C/C++ 2.2 for z/OS • LE TLS support for IBM Open XL C/C++ 2.2 for z/OS • IBM Open XL C/C++ 2.2 for z/OS C++ runtime support
5655-ZOS	Language Environment for z/OS 3.2.0 with these APARs(PTFs) applied: • PH62468(UO04849) • PH68180(UO04938)	• LE TLS support for IBM Open XL C/C++ 2.2 for z/OS • IBM Open XL C/C++ 2.2 for z/OS C++ runtime support
5655-ZOS	Binder for z/OS 3.1.0 (HPM77E0) with this APAR(PTF) applied: • OA68247(UJ98024)	• Binder support for IBM Open XL C/C++ 2.2 for z/OS
5655-ZOS	Binder for z/OS 3.2.0 (HPM77F0) with this APAR(PTF) applied: • OA68247(UJ98023)	• Binder support for IBM Open XL C/C++ 2.2 for z/OS

Note: Note on targeting zOSV2R5

When using the compiler on z/OS 3.1 and z/OS 3.2 with the **TARGET(zOSV2R5)** option, the generated application is intended to run on z/OS 2.5.

To successfully run the generated application on z/OS 2.5, the target system must have APAR/PTF **PH68112/UO04933** installed.

This requirement applies to the runtime environment of your end users. System administrators responsible for z/OS 2.5 environments should ensure that this APAR/PTF is applied. Please note that the compiler itself is **only supported** for use on z/OS 3.1 and z/OS 3.2; this prerequisite applies only to the runtime execution of applications built with the **TARGET(zOSV2R5)** option.

Conditional operational requisites identify products that are *not* required for this product to operate its basic functions but are required at run time for this product to operate specific functions.

There are no conditional operational requisites for this product.

5.2.2.3 Toleration/Coexistence Requisites:

Toleration/coexistence requisites identify products that must be present on sharing systems. These systems can be other systems in a multisystem environment (not necessarily sysplex), a shared DASD environment (such as test and production), or systems that reuse the same DASD environment at different time intervals.

5.2.2.4 Incompatibility (Negative) Requisites:

Negative requisites identify products that must not be installed on the same system as this product.

Open XL C/C++ 2.2 has no negative requisites.

5.2.3 DASD Storage Requirements

Open XL C/C++ 2.2 libraries can reside on all supported DASD types.

Figure 7 lists the total space that is required for each type of library.

Figure 7. Total DASD Space Required by Open XL C/C++ 2.2

Library Type	Total Space Required in 3390 Trks
Target	57000
Distribution	34000

Notes:

1. For non-RECFM U data sets, IBM recommends using system-determined block sizes for efficient DASD utilization. For RECFM U data sets, IBM recommends using a block size of 32760, which is most efficient from the performance and DASD utilization perspective.
2. Abbreviations used for data set types are shown as follows.

   U Unique data set, allocated by this product and used by only this product. This table provides all the required information to determine the correct storage for this data set. You do not need to refer to other tables or program directories for the data set size.

   S Shared data set, allocated by this product and used by this product and other products. To determine the correct storage needed for this data set, add the storage size given in this table to those given in other tables (perhaps in other program directories). If the data set already exists, it must have enough free space to accommodate the storage size given in this table.

   E Existing shared data set, used by this product and other products. This data set is *not* allocated by this product. To determine the correct storage for this data set, add the storage size given in this table to those given in other tables (perhaps in other program directories). If the data set already exists, it must have enough free space to accommodate the storage size given in this table.

   For more information about the names and sizes of the required data sets, see 6.1.5, “Allocate SMP/E Target and Distribution Libraries” on page 19.

3. Abbreviations used for the file system path type are as follows.

   N New path, created by this product.
   X Path created by this product, but might already exist from a previous release.
   P Previously existing path, created by another product.

4. All target and distribution libraries listed have the following attributes:

   - The default name of the data set can be changed.
   - The default block size of the data set can be changed.
   - The data set can be merged with another data set that has equivalent characteristics.
   - The data set can be either a PDS or a PDSE.

5. All target libraries listed have the following attributes:

   - These data sets can be SMS-managed, but they are not required to be SMS-managed.
   - These data sets are not required to reside on the IPL volume.
   - The values in the "Member Type" column are not necessarily the actual SMP/E element types that are identified in the SMPMCS.

6. All target libraries that are listed and contain load modules have the following attributes:

   - These data sets can be in the LPA, but they are not required to be in the LPA.
   - These data sets can be in the LNKLST.
   - These data sets are not required to be APF-authorized.
   - Open XL C/C++ 2.2 requires that the SMPLTS data set must be a PDSE. If your existing SMPLTS is a PDS, you will need to allocate a new PDSE and copy your existing SMPLTS into it and then change the SMPLTS DDDEF entry to indicate the new PDSE data set.

The following figures describe the target and distribution libraries and file system paths required to install Open XL C/C++ 2.2. The storage requirements of Open XL C/C++ 2.2 must be added to the storage required by other programs that have data in the same library or path.

Note: Use the data in these tables to determine which libraries can be merged into common data sets. In addition, since some ALIAS names may not be unique, ensure that no naming conflicts will be introduced before merging libraries.

Figure 8 (Page 1 of 5). Storage Requirements for Open XL C/C++ 2.2 Target Libraries

Library DDNAME	Member Type	Target Volume	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SCNWCMP	PROGRAM	T2	U	LIBRARY	U	0	15000	4
SCNWHCH	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWHCZ	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWJCL	SAMP	T2	U	PDS	FB	80	15	2
SCNWOBJ	DATA	T2	U	LIBRARY	FB	80	300	20

Figure 8 (Page 2 of 5). Storage Requirements for Open XL C/C++ 2.2 Target Libraries

Library DDNAME	Member Type	Target Volume	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SCNWOBJN	DATA	T2	U	LIBRARY	FB	80	300	20
SCNWPRC	PROC	T2	U	LIBRARY	FB	80	150	20
SCNWSAM	SAMP	T2	U	LIBRARY	FB	80	150	20
SCNWH100	DATA	T2	U	LIBRARY	FB	250	750	20
SCNWH101	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH102	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH103	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH104	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH105	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH106	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH107	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH108	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH109	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH110	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH111	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH112	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH113	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH114	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH115	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH116	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH117	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH118	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH119	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH120	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH121	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH122	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH123	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH124	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH125	DATA	T2	U	LIBRARY	FB	250	150	20

Figure 8 (Page 3 of 5). Storage Requirements for Open XL C/C++ 2.2 Target Libraries

Library DDNAME	Member Type	Target Volume	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SCNWH126	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH127	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH128	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH129	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH130	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH131	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH132	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH133	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH134	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH135	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH136	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH137	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH138	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH139	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH140	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH141	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH142	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH143	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH144	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH145	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH146	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH147	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH148	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH149	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH150	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH151	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH152	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH153	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH154	DATA	T2	U	LIBRARY	FB	250	150	20

Figure 8 (Page 4 of 5). Storage Requirements for Open XL C/C++ 2.2 Target Libraries

Library DDNAME	Member Type	Target Volume	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SCNWH155	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH156	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH157	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH158	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH159	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH160	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH161	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH162	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH163	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH164	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH165	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH166	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH167	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH168	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH169	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH170	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH171	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH172	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH173	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH174	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH175	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH176	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH177	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH178	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH179	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH180	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH181	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH182	DATA	T2	U	LIBRARY	FB	250	150	20
SCNWH183	DATA	T2	U	LIBRARY	FB	250	150	20

Figure 8 (Page 5 of 5). Storage Requirements for Open XL C/C++ 2.2 Target Libraries

Library DDNAME	Member Type	Target Volume	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
SCNWH184	DATA	T2	U	LIBRARY	FB	250	150	20

Figure 9. Open XL C/C++ 2.2 File System Paths

DDNAME	TYPE	Path Name
SCNWZ22	N	/usr/lpp/IBM/cnw/v2r2/IBM/

Figure 10. Storage Requirements for Open XL C/C++ 2.2 Distribution Libraries

Library DDNAME	TYPE	ORG	RECFM	LRECL	No. of 3390 Trks	No. of DIR Blks
ACNWSR1	U	PDS	FB	80	15	2
ACNWSR2	U	PDS	VB	255	4500	2
ACNWSR3	U	LIBRARY	U	0	15000	4
ACNWSR4	U	LIBRARY	FB	80	150	20
ACNWSR5	U	LIBRARY	FB	80	150	20
ACNWSR6	U	LIBRARY	FB	80	150	20
ACNWSR7	U	LIBRARY	FB	250	750	20
ACNWSR8	U	LIBRARY	FB	250	300	20
ACNWSR9	U	LIBRARY	FB	250	300	20
ACNWSR10	U	LIBRARY	FB	250	150	20
ACNWSR11	U	LIBRARY	FB	250	150	20
ACNWSR12	U	LIBRARY	FB	250	150	20
ACNWSR13	U	LIBRARY	FB	250	150	20
ACNWSR14	U	LIBRARY	FB	250	150	20
ACNWSR15	U	LIBRARY	FB	250	150	20

5.3 FMIDs Deleted

Installing Open XL C/C++ 2.2 might result in the deletion of other FMIDs. To see which FMIDs will be deleted, examine the ++VER statement in the SMPMCS of the product.

If you do not want to delete these FMIDs at this time, install Open XL C/C++ 2.2 into separate SMP/E target and distribution zones.

Note: These FMIDs are not automatically deleted from the Global Zone. If you want to delete these FMIDs from the Global Zone, use the SMP/E REJECT NOFMID DELETEFMID command. See the SMP/E Commands book for details.

5.4 Special Considerations

Open XL C/C++ 2.2 and Open XL C/C++ 2.1 Co-existence:

Open XL C/C++ 2.2 cannot coexist with Open XL C/C++ 2.1 in the same CSI. The two releases cannot share target or distribution data set names. Due to packaging changes in Open XL C/C++ 2.2, data sets from Open XL C/C++ 2.1 are not compatible. Uninstall Open XL C/C++ 2.1 before installing Open XL C/C++ 2.2.

6.0 Installation Instructions

This chapter describes the installation method and the step-by-step procedures to install and to activate the functions of Open XL C/C++ 2.2.

Please note the following points:

- If you want to install Open XL C/C++ 2.2 into its own SMP/E environment, consult the SMP/E manuals for instructions on creating and initializing the SMPCSI and the SMP/E control data sets.
- You can use the sample jobs that are provided to perform part or all of the installation tasks. The SMP/E jobs assume that all DDDEF entries that are required for SMP/E execution have been defined in appropriate zones.
- You can use the SMP/E dialogs instead of the sample jobs to accomplish the SMP/E installation steps.

6.1 Installing Open XL C/C++ 2.2

This product is available as a web download package. If you are downloading this product from the web, follow the instructions below.

6.1.1 SMP/E considerations for using the SMP/E web download

The SMP/E web download package for Open XL C/C++ 2.2 is packaged using the SMP/E GIMZIP function, which was introduced in SMP/E for z/OS 3.1. Although GIMZIP and GIMUNZIP are used for the packaging, the full SMP/E RECEIVE FROMNETWORK function is not available with this package. The SMP/E GIMUNZIP function is required to process the downloaded package. See the z/OS SMP/E Reference for information about using GIMZIP and GIMUNZIP. Ensure that you meet the driving system requirements as documented in Driving System Requirements.

Perform the following tasks:

1. Ensure that the configuration requirements for using GIMUNZIP are completed. For a description of the GIMZIP and GIMUNZIP functions, see the SMP/E web page at http://www.ibm.com/servers/eserver/zseries/zos/smpe/.
2. Allocate a R/W file system directory on the z/OS system where the package will be staged. The R/W file system directory is the repository for the download package.
3. Download the web package from https://www.ibm.com/products/xl-cpp-compiler-zos/. The package contains two parts, OPNXLC2.README.txt and OPNXLC2.pax.Z. The file OPNXLC2.README.txt contains a sample job that performs the following tasks. You must update this file to reflect your environment. Make sure this file is transferred from the download site as a text file.
    - Executes the z/OS UNIX System Services pax command to extract the GIMZIP archives from the download package.

- Executes the GIMUNZIP program to expand the GIMZIP archives and places their contents in data sets that can be processed by SMP/E.
- Executes the SMP/E RECEIVE from DASD function to receive the FMIDs.

The file OPNXLC2.pax.Z contains the SMP/E MCS and the associated RELFILEs. You must download this pax archive file to a node that has connectivity to the target z/OS system. Transfer this file to the host using binary format.

4. Run the sample job in OPNXLC2.README.txt. This job will perform the required tasks up to and including the SMP/E RECEIVE from DASD step.

   Expected Return Codes: RC=0.

5. Upgrade your target system (z/OS 3.1 or 3.2) with current service.
6. Complete the installation by using the instructions in this program directory.

6.1.2 SMP/E Options Subentry Values

The recommended values for certain SMP/E CSI subentries are shown in Figure 11. Using values lower than the recommended values can result in failures in the installation. DSSPACE is a subentry in the GLOBAL options entry. PEMAX is a subentry of the GENERAL entry in the GLOBAL options entry. See the SMP/E manuals for instructions on updating the global zone.

Figure 11. SMP/E Options Subentry Values

Subentry	Value	Comment
DSSPACE	Existing target CSI value	IBM suggests using your existing target system CSI's DSSPACE value.
PEMAX	SMP/E Default	IBM suggests using the SMP/E default for PEMAX.

6.1.3 SMP/E CALLLIBS Processing

Open XL C/C++ 2.2 uses the CALLLIBS function that is provided in SMP/E to resolve external references during installation. When Open XL C/C++ 2.2 is installed, ensure that DDDEFs exist for the following libraries:

- SCEEBND2
- SCEELIB

Note: CALLLIBS uses the previous DDDEFs only to resolve the link-edit for Open XL C/C++ 2.2 compiler. These data sets are not updated during the installation of Open XL C/C++ 2.2, except for SCEELIB, which is a Language Environment data set and is shared with Runtime Library Extensions.

6.1.4 Sample Jobs

The following sample installation jobs are provided as part of the product to help you install Open XL C/C++ 2.2:

Figure 12. Sample Installation Jobs

Job Name	Job Type	Description	RELFILE
CNWWALOC	ALLOCATE	Sample job to allocate target and distribution libraries for Open XL C/C++ 2.2	IBM.HLB77C4.F1
CNWWZFS	ALLOMZFS	Sample job to allocate, create mountpoint, & mount zFS data sets **(Optional)** Open XL C/C++ 2.2	IBM.HLB77C4.F1
CNWWDDEF	DDDEF	Sample job to define SMP/E DDDEFs for Open XL C/C++ 2.2	IBM.HLB77C4.F1
CNWISMKD	MKDIR	Sample job to invoke the supplied CNWMKDIR EXEC to allocate file system paths	IBM.HLB77C4.F1
CNWJIVP1	IVP	Sample job to verify installation has been successful for Open XL C/C++ 2.2	IBM.HLB77C4.F1
CNWJIVP2	IVP	Sample job to verify installation has been successful for Open XL C/C++ 2.2	IBM.HLB77C4.F1

You can access the sample installation jobs by performing an SMP/E RECEIVE and then copying the jobs from the relfiles to a work data set for editing and submission. See Figure 12 on page 18 to find the appropriate relfile data set.

6.1.5 Allocate SMP/E Target and Distribution Libraries

Edit and submit sample job CNWWALOC to allocate the SMP/E target and distribution libraries for Open XL C/C++ 2.2. Consult the instructions in the sample job for more information.

Expected Return Codes and Messages: RC=0

6.1.6 Allocate, create and mount ZFS Files (Optional)

This job allocates, creates a mountpoint, and mounts zFS data sets.

If you plan to install Open XL C/C++ 2.2 into a new z/OS UNIX file system, you can edit and submit the optional CNWWZFS job to perform the following tasks:

- Create the z/OS UNIX file system
- Create a mount point
- Mount the z/OS UNIX file system on the mountpoint

Consult the instructions in the sample job for more information.

The recommended z/OS UNIX file system type is *zFS*. The recommended mount point is */usr/lpp/IBM/cnw/v2r2*.

Before running the sample job to create the z/OS UNIX file system, you must ensure that OMVS is active on the driving system. zFS must be active on the driving system if you are installing Open XL C/C++ 2.2 into a file system that is zFS.

If you create a new file system for this product, consider updating the BPXPRMxx PARMLIB member to mount the new file system at IPL time. This action can be helpful if an IPL occurs before the installation is completed.

```
MOUNT FILESYSTEM('#dsn')
 MOUNTPOINT('/usr/lpp/IBM/cnw/v2r2')
 MODE(RDWR)         /* can be MODE(READ) */
 TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
```

See the following information to update the statements in the previous sample:

#dsn is the name of the data set holding the z/OS UNIX file system.
/usr/lpp/IBM/cnw/v2r2 is the name of the mount point where the z/OS UNIX file system will be mounted.

Expected Return Codes and Messages: You will receive a return code of 0 if this job runs correctly.

6.1.7 Allocate File System Paths

The target system HFS or zFS data set must be mounted on the driving system when running the sample CNWISMKD job since the job will create paths in the HFS or zFS.

Before running the sample job to create the paths in the file system, you must ensure that OMVS is active on the driving system and that the target system's HFS or zFS file system is mounted to the driving system. zFS must be active on the driving system if you are installing Open XL C/C++ 2.2 into a file system that is zFS.

If you plan to install Open XL C/C++ 2.2 into a new HFS or zFS file system, you must create the mountpoint and mount the new file system to the driving system for Open XL C/C++ 2.2.

The recommended mountpoint is */usr/lpp/IBM/cnw/v2r2*.

Edit and submit sample job CNWISMKD to allocate the HFS or zFS paths for Open XL C/C++ 2.2. Consult the instructions in the sample job for more information.

If you create a new file system for this product, consider updating the BPXPRMxx PARMLIB member to mount the new file system at IPL time. This action can be helpful if an IPL occurs before the installation is completed.

Expected Return Codes and Messages: You will receive a return code of 0 if this job runs correctly.

6.1.8 Create DDDEF Entries

Edit and submit sample job CNWWDDEF to create DDDEF entries for the SMP/E target and distribution libraries for Open XL C/C++ 2.2. Consult the instructions in the sample jobs for more information.

Expected Return Codes and Messages: RC=0

6.1.9 Perform SMP/E APPLY

1. Ensure that you have the latest HOLDDATA; then edit and submit sample job, as shown below, to perform an SMP/E APPLY CHECK for Open XL C/C++ 2.2.

   HOLDDATA introduces ERROR HOLDs against FMIDs for HIPER APARs. Before the installation, ensure that you have the latest Enhanced HOLDDATA, which is available through several different portals, including https://www.ibm.com/support/pages/enhanced-holddata-zos. Install the FMIDs regardless of the status of unresolved HIPERs. However, don't deploy the software until the unresolved HIPERs are analyzed to determine applicability.

   To receive the full benefit of the SMP/E Causer SYSMOD Summary Report, do not bypass the PRE, ID, REQ, and IFREQ on the APPLY CHECK. This is because the SMP/E root cause analysis identifies the cause of only errors but not warnings (SMP/E treats bypassed PRE, ID, REQ, and IFREQ conditions as warnings rather than errors).

   Here are two methods to install FMIDs when ++HOLDs for HIPERs exist for the FMIDs that you install:

   a. To ensure that all recommended and critical service is installed with the FMIDs, if you are using SMP/E 3.5 or higher and have received the latest HOLDDATA, add the FIXCAT operand to the APPLY command as shown below. If you are using a prior release of SMP/E, add the SOURCEID(HIPER,RSU*) operand to the APPLY command.

      ```
      If using SMP/E V3.5 or higher:
      APPLY S(HLB77C4) CHECK
      FORFMID(HLB77C4)
      SOURCEID(RSU*)
      FIXCAT(IBM.ProductInstall-RequiredService)
      GROUPEXTEND .

      If using SMP/E V3.4 or prior:
      APPLY S(HLB77C4) CHECK
      FORFMID(HLB77C4)
      SOURCEID(HIPER,RSU*,...)
      GROUPEXTEND .
      ```

      Some HIPER APARs might not have PTFs available yet. You should analyze the symptom flags for to determine if you want to bypass the specific ERROR HOLDs and continue the installation of the FMIDs.

This method requires more initial research, but can provide resolution for all HIPERs that have fixing PTFs available and are not in a PE chain. Unresolved PEs or HIPERs might still exist and require the use of BYPASS.

b. To install the FMIDs without regard for the HIPERs, you can add a BYPASS(HOLDCLASS(HIPER)) operand to the APPLY command. In this way, you can install FMIDs even though HIPER ERROR HOLDs against them still exist. Only the HIPER ERROR HOLDs are bypassed. After the FMIDs are installed, run the SMP/E REPORT ERRSYSMODS command to identify missing HIPER maintenance.

```
APPLY S(HLB77C4) CHECK
FORFMID(HLB77C4)
SOURCEID(RSU*)
FIXCAT(IBM.ProductInstall-RequiredService)
GROUPEXTEND
BYPASS(HOLDCLASS(HIPER)) .
other parameters documented in the program directory
```

This method is the quicker of the two, but requires subsequent review of the REPORT ERRSYSMODS to investigate any HIPERs. If you are running SMP/E V3.5 or higher and have received the latest HOLDDATA, you can also choose to run REPORT MISSINGFIX for Fix Category IBM.ProductInstall-RequiredService to investigate missing recommended service.

If you bypass HOLDs during the installation of the FMIDs because fixing PTFs are not yet available, you can be notified when the fixing PTFs are available by using the APAR Status Tracking (AST) function of ServiceLink or the APAR Tracking function of ResourceLink.

2. After you take actions that are indicated by the APPLY CHECK, remove the CHECK operand and run the job again to perform the APPLY.

A sample APPLY job is shown below.

```
//APPLY JOB <job parameters>
//STEP1 EXEC PGM=GIMSMP,REGION=0M,TIME=NOLIMIT
//SMPCSI DD DSN=#globalcsi,DISP=SHR
//SMPCNTL DD *
  SET BOUNDARY(#tzone) .
  APPLY CHECK
  FORFMID(HLB77C4)
  SELECT(HLB77C4)
  GROUPEXTEND(NOAPARS,NOUSERMODS)
  BYPASS(HOLDSYSTEM,
  HOLDUSER,HOLDCLASS(UCLREL,ERREL,HIPER)) .
/*
```

Required Updates:

1. Update the job parameters.
2. Replace the #globalcsi on the SMPCSI DD statement with your CSI name.
3. Update #tzone to your target zone name.

Note: The GROUPEXTEND operand indicates that SMP/E applies all requisite SYSMODs. The requisite SYSMODS might be applicable to other functions.

Expected Return Codes and Messages from APPLY CHECK: You will receive a return code of 0 if this job runs correctly.

Expected Return Codes and Messages from APPLY: You will receive a return code of 0 if this job runs correctly.

6.1.10 Customize Open XL C/C++ 2.2

Once you have successfully performed SMP/E APPLY with the Open XL C/C++ 2.2, you must customize the product. Customization may be run from your driving system unless otherwise specified.

6.1.10.1 PARMLIB member considerations

You must tailor the system for Open XL C/C++ 2.2 to meet the needs of your installation.

Figure 13. PARMLIB members and description

PARMLIB Member	Description
IFAPRDxx	Defines the enablement policy for XL C/C++ See 6.1.10.2, “IFAPRDxx considerations.”

6.1.10.2 IFAPRDxx considerations

With z/OS, products can use registration services to determine if they are enabled to run on a particular system. This requires the product be defined appropriately in the enablement policy for the system using the IFAPRDxx PARMLIB member.

You must ensure that the policy in IFAPRDxx enables only what are licensed. Use of (and enablement of) z/OS features is subject to the z/OS license terms and conditions and must be done with the knowledge of your asset manager according to the terms and conditions for z/OS. See the Usage Restriction section of the *z/OS Licensed Program Specifications* for additional license terms and conditions.

For IBM Open XL C/C++ 2.2 for z/OS, its FEATURENAME and ID specified in IFAPRDxx are the same as the XL C/C++ compiler, which are:

```
ID(5655-ZOS)
FEATURENAME('C/C++')
```

Here is an example:

```
PRODUCT OWNER('IBM CORP')
        NAME('z/OS')
        ID(5655-ZOS)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME('C/C++')
        STATE(ENABLED)
```

For the full list of z/OS priced features which includes XL C/C++ compiler, please refer to:
https://www.ibm.com/docs/en/zos/3.1.0?topic=parmlib-zos-priced-features-tcpip

See *z/OS Planning for Installation* for additional information on enabling z/OS features.

6.1.10.3 Customization file

A customization file clang.cfg on z/OS UNIX must be created for the IBM Open XL C/C++ 2.2 for z/OS compiler if the High Level Qualifier (HLQ) for Language Environment (LE) datasets or CSSLIB datasets deviate from the default values. The customization file can be created using the following instructions:

- The file must be created as a text file under the USS path '#PathPrefix/usr/lpp/IBM/cnw/v2r2/openxl/etc/clang.cfg', where '#PathPrefix' will be default to root '/' directory if it's not changed in DDDEF entry. Note that the 'etc' directory must be created in the same parent directory as the 'bin' directory with UNIX permission 755. The clang.cfg file can also be created in another directory (with the same permission requirements) and then add

  ```
  export CLANG_CONFIG_PATH=<path>/clang.cfg
  ```

  to the user profile.
- If the HLQ for Language Environment (LE) datasets is not the default value CEE. A new line

  ```
  -mzos-hlq-le=<hlq>
  ```

  where <hlq> is the actual HLQ for Language Environment datasets must be added to the file.
- If the HLQ for CSSLIB datasets is not the default value SYS1. A new line

  ```
  -mzos-hlq-csslib=<hlq>
  ```

  where <hlq> is the actual HLQ for CSSLIB datasets must be added to the file.
- if the HLQ for the CICS datasets is not the same as the CSSLIB datasets. A new line

  ```
  -mzos-hlq-cics=<hlq>
  ```

  where <hls> is the actual HLQ for the CICS datasets must be added to the file.
- Any lines in the files starting with the '#' character are treated as comments.

6.1.11 Run the Installation Verification Programs

The Open XL C/C++ 2.2 consists of a C/C++ compiler, which supports compilation in both z/OS UNIX environment and batch environment. Therefore, you need to run 2 IVP jobs to ensure that all parts for Open XL C/C++ compiler is properly installed. The job CNWJIVP1 tests the compilation in UNIX environment, while it uses the directory /tmp as work directory for processing inputs and outputs. Ensure that the file system (either HFS or ZFS) is available on the target system before you run the job. If you use a customized TMPDIR instead of the default /tmp, the customized TMPDIR must not include the special charater _ (underscore) in directory name or the binder will fail to find the primary input file. The job CNWJIVP2 tests the compilation in batch environment.

The following sections describe installation verification procedures for the Open XL C/C++ 2.2.

6.1.11.1 Run the Open XL C/C++ 2.2 Installation Verification Procedures:

If Open XL C/C++ 2.2 has been enabled (see 6.1.10.2, “IFAPRDxx considerations” on page 23), verify that the Open XL C/C++ 2.2 has been installed properly.

You need to run the following IVP jobs on the target system to ensure that the C/C++ compiler is properly installed. The JCL can be found in your SCNWJCL library. Refer to the JCL for instructions and expected output.

Figure 14. IVPs for Open XL C/C++ Compilers

Job Name	Job Description
CNWJIVP1	Verify the Open XL C/C++ compiler installation in UNIX environment
CNWJIVP2	Verify the Open XL C/C++ compiler installation in batch environment

Expected Return Codes and Messages: RC=0.

6.1.12 Perform SMP/E ACCEPT

Edit and submit the sample job, as shown in 6.1.9, “Perform SMP/E APPLY” on page 21, to perform an SMP/E ACCEPT CHECK for Open XL C/C++ 2.2.

To receive the full benefit of the SMP/E Causer SYSMOD Summary Report, do *not* bypass the PRE, ID, REQ, and IFREQ on the ACCEPT CHECK. This is because the SMP/E root cause analysis identifies the cause of only errors but not warnings (SMP/E treats bypassed PRE, ID, REQ, and IFREQ conditions as warnings rather than errors).

Before you use SMP/E to load new distribution libraries, it is recommended that you set the ACCJCLIN indicator in the distribution zone. In this way, you can save the entries that are produced from JCLIN in the distribution zone whenever a SYSMOD that contains inline JCLIN is accepted. For more information about the ACCJCLIN indicator, see the description of inline JCLIN in the SMP/E manuals.

After you take actions that are indicated by the ACCEPT CHECK, remove the CHECK operand and run the job again to perform the ACCEPT.

A sample APPLY job is shown below.

```
//ACCEPT JOB <job parameters>
//STEP1 EXEC PGM=GIMSMP,REGION=0M,TIME=NOLIMIT
//SMPCSI DD DSN=#globalcsi,DISP=SHR
//SMPCNTL DD *
  SET BOUNDARY(#dzone) .
  ACCEPT CHECK
  FORFMID(HLB77C4)
  SELECT(HLB77C4)
  GROUPEXTEND(NOAPARS,NOUSERMODS)
  BYPASS(HOLDSYSTEM,
  HOLDUSER,HOLDCLASS(UCLREL,ERREL,HIPER)) .
/*
```

Required Updates:

1. Update the job parameters.
2. Replace the #globalcsi on the SMPCSI DD statement with your CSI name.
3. Update #dzone to your dlib zone name.

Note: The GROUPEXTEND operand indicates that SMP/E accepts all requisite SYSMODs. The requisite SYSMODS might be applicable to other functions.

Expected Return Codes and Messages from ACCEPT CHECK: You will receive a return code of 0 if this job runs correctly.

If PTFs that contain replacement modules are accepted, SMP/E ACCEPT processing will link-edit or bind the modules into the distribution libraries. During this processing, the Linkage Editor or Binder might issue messages that indicate unresolved external references, which will result in a return code of 4 during the ACCEPT phase. You can ignore these messages, because the distribution libraries are not executable and the unresolved external references do not affect the executable system libraries.

Expected Return Codes and Messages from ACCEPT: You will receive a return code of 0 if this job runs correctly.

6.1.13 Run REPORT CROSSZONE

The SMP/E REPORT CROSSZONE command identifies requisites for products that are installed in separate zones. This command also creates APPLY and ACCEPT commands in the SMPPUNCH data set. You can use the APPLY and ACCEPT commands to install those cross-zone requisites that the SMP/E REPORT CROSSZONE command identifies.

After you install Open XL C/C++ 2.2, it is recommended that you run REPORT CROSSZONE against the new or updated target and distribution zones. REPORT CROSSZONE requires a global zone with ZONEINDEX entries that describe all the target and distribution libraries to be reported on.

For more information about REPORT CROSSZONE, see the SMP/E manuals.

7.0 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.

APAR numbers are provided in this document to assist in locating PTFs that may be required. Ongoing problem reporting may result in additional APARs being created. Therefore, the APAR lists in this document may not be complete. To obtain current service recommendations and to identify current product service requirements, always contact the IBM Customer Support Center.

IBM may have patents or pending patent applications covering subject matter 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 the

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, New York 10504-1785
USA

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

7.1 Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

Reader's Comments

Program Directory for IBM Open XL C/C++ 2.2 for z/OS, October 2025

IBM appreciates your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Send your comments by emailing IBM at ibmdocs@us.ibm.com, and include the following information:

```
Your name and address
Your email address
Your telephone or fax number
The publication title and order number
The topic and page number related to your comment
The text of your comment
```

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

IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you submit.

Thank you for your participation.

Printed in USA

GI13-5701-03