

IBM XL C/C++ for Linux on z Systems, V1.2



Installation Guide

Version 1.2

IBM XL C/C++ for Linux on z Systems, V1.2



Installation Guide

Version 1.2

Note

Before using this information and the product it supports, read the information in "Notices" on page 31.

First edition

This edition applies to IBM XL C/C++ for Linux on z Systems, V1.2 (Program 5725-N01) 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 2015.

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

Contents

About this information	v	Testing a PTF before you install it	18
Who should read this document.	v	Updating a basic installation.	18
How to use this document.	v	Updating multiple versions of PTFs installed under separate locations	19
How this document is organized	vi	Chapter 5. Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users)	21
Conventions	vi	Running the <code>x1c_configure</code> utility directly	21
Related information	x	<code>x1c_configure</code> options.	22
IBM XL C/C++ information	x	Chapter 6. After installing IBM XL C/C++ for Linux on z Systems, V1.2	23
Standards and specifications	xi	Testing the installation.	23
Other information	xi	Enabling the manual pages	24
Technical support	xi	Enabling the error messages.	25
How to send your comments	xii	Setting up the environment for the invocation commands.	25
Chapter 1. Before installing IBM XL C/C++ for Linux on z Systems, V1.2.	1	Setting the <code>PATH</code> environment variable to include the path to the compiler invocations	25
The installation image and packages	1	Creating symbolic links to the compiler invocations	26
System prerequisites.	3	Accessing the local documentation	26
Verifying the amount of hard disk space available	6	Viewing the manual pages	26
Verifying that the required GNU and Perl packages are installed	6	Querying for installed packages	27
Previewing the license agreements	7	Chapter 7. Uninstalling IBM XL C/C++ for Linux on z Systems, V1.2	29
Chapter 2. Basic installation	9	Example: Uninstalling IBM XL C/C++ for Linux on z Systems, V1.2	29
Tasks for basic installation.	9	Notices	31
Running the install utility for a new installation	10	Trademarks	33
Chapter 3. Advanced installation	13	Index	35
Tasks for advanced installation	13		
Installing XL C/C++ to a nondefault location (SLES and RHEL)	14		
Installing XL C/C++ to a nondefault location (Ubuntu)	14		
Querying for RPM package summaries (SLES and RHEL)	15		
Querying for dpkg package summaries (Ubuntu)	15		
Chapter 4. Installing an update	17		
Tasks for update installation.	17		

About this information

This guide provides detailed installation instructions for IBM® XL C/C++ for Linux on z Systems™, V1.2. It includes multiple installation procedures that you can follow based on your installation requirements and scenarios. Specific instructions are given in cases requiring atypical installation. It also shows you how to test the installation, launch remotely-accessible HTML help, and view different types of documentation. Read it carefully before installing the compiler. Also read the README file in the root directory of your installation media, which contains current information about the compiler.

Who should read this document

This document is intended for anyone responsible for installing IBM XL C/C++ for Linux on z Systems, V1.2.

The majority of users can follow the basic installation method detailed in this document. Basic examples are tailored to reflect, as much as possible, the steps required to perform a basic installation.

The advanced installation method addresses the needs of users who require a customized installation for various purposes, such as maintaining more than one version of XL C/C++ on a single system. These are users who are familiar with compiler installations and with the file structures of the versions of the compiler products installed on the system. In this document, you are referred to as an *advanced user*.

How to use this document

This document provides procedures for three main installation scenarios:

Basic installation

This scenario allows you to install a single version of XL C/C++ to a default location. This scenario is applicable to the majority of users, and is the recommended method of installation.

For an overview of the steps that you need to follow to perform a basic installation, see Chapter 2, “Basic installation,” on page 9.

Advanced installation

This scenario allows you to maintain multiple versions of XL C/C++ on a single system, or to install the compiler to a nondefault location. This scenario is only applicable to advanced users who have specialized needs, and is not recommended for the majority of users.

For an overview of the steps that you need to follow to perform an advanced installation, see Chapter 3, “Advanced installation,” on page 13.

Update installation

This scenario applies to users who have obtained a Program Temporary Fix (PTF) package for an existing XL C/C++ V1.2 installation.

For an overview of the steps required to install a PTF, see Chapter 4, “Installing an update,” on page 17.

For highly specialized installation scenarios that are outside the scope of this document, see the Technotes at <http://www.ibm.com/software/products/en/xlcpp-loz>.

How this document is organized

This book is organized to reflect the pre-installation, installation, and post-installation phases of an XL C/C++ installation.

Table 1. Phases of an XL C/C++ installation

Phase	Chapters	User segment
Pre-installation	Chapter 1, "Before installing IBM XL C/C++ for Linux on z Systems, V1.2," on page 1	All users
Installation	Chapter 2, "Basic installation," on page 9	<ul style="list-style-type: none"> • Users who want to use the simplest, most direct installation process • Users who do not have any special requirements, such as the use of multiple versions of the compiler.
	Chapter 3, "Advanced installation," on page 13	<ul style="list-style-type: none"> • Users who want to install the compiler in a nondefault location • Users who want to have multiple versions of the compiler on the same system
Post-installation	Chapter 4, "Installing an update," on page 17	Users who want to update XL C/C++ V1.2 to the next fix level
	Chapter 5, "Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users)," on page 21	<ul style="list-style-type: none"> • Users who installed the compiler in a nondefault location • Users who have multiple versions of the XL C/C++ compiler installed on the same system
	Chapter 6, "After installing IBM XL C/C++ for Linux on z Systems, V1.2," on page 23	All users
Product removal	Chapter 7, "Uninstalling IBM XL C/C++ for Linux on z Systems, V1.2," on page 29	Any user who needs to remove an XL C/C++ compiler from the system

Conventions

Typographical conventions

The following table shows the typographical conventions used in the IBM XL C/C++ for Linux on z Systems, V1.2 information.

Table 2. *Typographical conventions*

Typeface	Indicates	Example
bold	Lowercase commands, executable names, compiler options, and directives.	The compiler provides basic invocation commands, <code>xlc</code> and <code>xlc</code> (<code>xlc++</code>), along with several other compiler invocation commands to support various C/C++ language levels and compilation environments.
<i>italics</i>	Parameters or variables whose actual names or values are to be supplied by the user. Italics are also used to introduce new terms.	Make sure that you update the <i>size</i> parameter if you return more than the <i>size</i> requested.
<u>underlining</u>	The default setting of a parameter of a compiler option or directive.	<code>nomaf</code> <u><code>maf</code></u>
monospace	Programming keywords and library functions, compiler builtins, examples of program code, command strings, or user-defined names.	To compile and optimize <code>myprogram.c</code> , enter: <code>xlc myprogram.c -O3</code> .

Qualifying elements (icons)

Most features described in this information apply to both C and C++ languages. In descriptions of language elements where a feature is exclusive to one language, or where functionality differs between languages, this information uses icons to delineate segments of text as follows:

Table 3. *Qualifying elements*

Qualifier/Icon	Meaning
C only begins	The text describes a feature that is supported in the C language only; or describes behavior that is specific to the C language.
C only ends	
C++ only begins	The text describes a feature that is supported in the C++ language only; or describes behavior that is specific to the C++ language.
C++ only ends	
IBM extension begins	The text describes a feature that is an IBM extension to the standard language specifications.
IBM extension ends	

Table 3. Qualifying elements (continued)

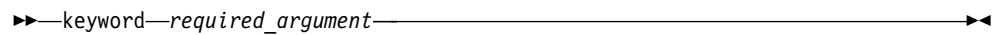
Qualifier/Icon	Meaning
C11 begins	The text describes a feature that is introduced into standard C as part of C11.
C11 ends	
C++11 begins	The text describes a feature that is introduced into standard C++ as part of C++11.
C++11 ends	
C++14 begins	The text describes a feature that is introduced into standard C++ as part of C++14.
C++14 ends	

Syntax diagrams

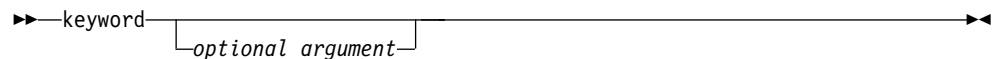
Throughout this information, diagrams illustrate XL C/C++ syntax. This section helps you to interpret and use those diagrams.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.
 - The **▶—** symbol indicates the beginning of a command, directive, or statement.
 - The **—▶** symbol indicates that the command, directive, or statement syntax is continued on the next line.
 - The **▶—** symbol indicates that a command, directive, or statement is continued from the previous line.
 - The **—▶** symbol indicates the end of a command, directive, or statement.
- Fragments, which are diagrams of syntactical units other than complete commands, directives, or statements, start with the **|—** symbol and end with the **—|** symbol.

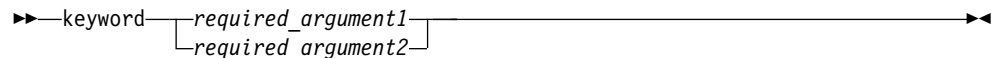
- Required items are shown on the horizontal line (the main path):



- Optional items are shown below the main path:



- If you can choose from two or more items, they are shown vertically, in a stack. If you *must* choose one of the items, one item of the stack is shown on the main path.



If choosing one of the items is optional, the entire stack is shown below the main path.



- An arrow returning to the left above the main line (a repeat arrow) indicates that you can make more than one choice from the stacked items or repeat an item. The separator character, if it is other than a blank, is also indicated:



- The item that is the default is shown above the main path.



- Keywords are shown in nonitalic letters and should be entered exactly as shown.
- Variables are shown in italicized lowercase letters. They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

Example of a syntax statement

EXAMPLE *char_constant* {*a|b*}[*c|d*]*e*[,*e*]*... name_list*{*name_list*}*...*

The following list explains the syntax statement:

- Enter the keyword EXAMPLE.
- Enter a value for *char_constant*.
- Enter a value for *a* or *b*, but not for both.
- Optionally, enter a value for *c* or *d*.
- Enter at least one value for *e*. If you enter more than one value, you must put a comma between each.
- Optionally, enter the value of at least one *name* for *name_list*. If you enter more than one value, you must put a comma between each *name*.

Note: The same example is used in both the syntax-statement and syntax-diagram representations.

Examples in this information

The examples in this information, except where otherwise noted, are coded in a simple style that does not try to conserve storage, check for errors, achieve fast performance, or demonstrate all possible methods to achieve a specific result.

The examples for installation information are labelled as either *Example* or *Basic example*. *Basic examples* are intended to document a procedure as it would be performed during a basic, or default, installation; these need little or no modification.

Related information

The following sections provide related information for XL C/C++:

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 on z Systems, V1.2. 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 on z Systems, V1.2 Installation Guide*.

- Online product documentation

The fully searchable HTML-based documentation is viewable in IBM Knowledge Center at http://www.ibm.com/support/knowledgecenter/SSVUN6_1.2.0/com.ibm.compilers.loz.doc/welcome.html.

- PDF documents

PDF documents are available on the web at <http://www.ibm.com/support/docview.wss?uid=swg27044043>.

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

Table 4. XL C/C++ PDF files

Document title	PDF file name	Description
<i>IBM XL C/C++ for Linux on z Systems, V1.2 Installation Guide, GC27-5995-01</i>	<code>install.pdf</code>	Contains information for installing XL C/C++ and configuring your environment for basic compilation and program execution.
<i>Getting Started with IBM XL C/C++ for Linux on z Systems, V1.2, GI13-2865-01</i>	<code>getstart.pdf</code>	Contains an introduction to the XL C/C++ product, with information about setting up and configuring your environment, compiling and linking programs, and troubleshooting compilation errors.
<i>IBM XL C/C++ for Linux on z Systems, V1.2 Compiler Reference, SC27-5998-01</i>	<code>compiler.pdf</code>	Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions.
<i>IBM XL C/C++ for Linux on z Systems, V1.2 Language Reference, SC27-5996-01</i>	<code>langref.pdf</code>	Contains information about language extensions for portability and conformance to nonproprietary standards.

Table 4. XL C/C++ PDF files (continued)

Document title	PDF file name	Description
<i>IBM XL C/C++ for Linux on z Systems, V1.2 Optimization and Programming Guide, SC27-5997-01</i>	proguide.pdf	Contains information about advanced programming topics, such as application porting, 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=swg27044043>.

For more information about C/C++, see the C/C++ café at <https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=5894415f-be62-4bc0-81c5-3956e82276f3>.

Standards and specifications

XL C/C++ is designed to support the following standards and specifications. You can refer to these standards and specifications for precise definitions of some of the features found in this information.

- *Information Technology - Programming languages - C, ISO/IEC 9899:1990*, also known as C89.
- *Information Technology - Programming languages - C, ISO/IEC 9899:1999*, also known as C99.
- *Information Technology - Programming languages - C, ISO/IEC 9899:2011*, also known as C11.
- *Information Technology - Programming languages - C++, ISO/IEC 14882:1998*, also known as C++98.
- *Information Technology - Programming languages - C++, ISO/IEC 14882:2003*, also known as C++03.
- *Information Technology - Programming languages - C++, ISO/IEC 14882:2011*, also known as C++11 (Partial support).
- *Information Technology - Programming languages - C++, ISO/IEC 14882:2014*, also known as C++14 (Partial support).
- *ANSI/IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985*.

Other information

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

Technical support

Additional technical support is available from the XL C/C++ Support page at http://www.ibm.com/support/entry/portal/product/rational/xl_c/c++_for_linux_on_z_systems. This page provides a portal with search capabilities to a large selection of Technotes and other support information.

If you cannot find what you need, you can send an email to compinfo@ca.ibm.com.

For the latest information about XL C/C++, visit the product information site at <http://www.ibm.com/software/products/en/xlcpp-loz>.

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@ca.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. Before installing IBM XL C/C++ for Linux on z Systems, V1.2

Before you install IBM XL C/C++ for Linux on z Systems, V1.2, do the following tasks:

- Consult the product README files for any last-minute updates.
- Familiarize yourself with the installation image, which contains the installable compiler packages.
- Ensure that system prerequisites are met and that all required software packages are installed.
- Become either the root user or a user with administrator privileges.

Upgrading the evaluation copy of the XL C/C++ compiler

On SLES and RHEL

If you are using the evaluation copy of the XL C/C++ compiler on SLES or RHEL, use one of the following options to upgrade it to a full product version:

- Use `install` to install the full product version of the XL C/C++ compiler as described in Chapter 2, “Basic installation,” on page 9.
- Alternatively, upgrade your evaluation copy as follows:
 1. Install the new license packages using the following command:

```
rpm -Uvh xlc-license.1.2.0-*.s390x.rpm
```
 2. Run the `xlc_configure` utility to accept the new license if it is not accepted, and reconfigure the compiler.

On Ubuntu

If you are using the evaluation copy of the XL C/C++ compiler on Ubuntu, use one of the following options to upgrade it to a full product version:

- Use `install` to install the full product version of the XL C/C++ compiler as described in Chapter 2, “Basic installation,” on page 9.
- Alternatively, upgrade your evaluation copy as follows:
 1. Install the license packages using the following command:

```
dpkg -iG xlc-license.1.2*_ppc64el.deb
```
 2. Run the `xlc_configure` utility to accept the license if it is not accepted, and reconfigure the compiler.

The installation image and packages

The installation image for IBM XL C/C++ for Linux on z Systems, V1.2 is available on an installation DVD or for download from an IBM website to a local drive.

The image includes:

- README files, and license agreement files.
- Sets of RPM packages.
- An installation tool, `install`, to install and configure the compiler for a basic installation.

Installation packages

Table 5 lists the packages that are supplied with the installation image, and the default locations to which they are installed during a basic installation. To view package information and the package file list, enter the following rpm or dpkg command:

```
rpm -qip package_name
```

```
dpkg -f package_name
```

Table 5. IBM XL C/C++ for Linux on z Systems, V1.2 packages and default installation locations

Package name	Package description	Default installation location
libxlmass-devel.1.2.0	IBM Mathematical Acceleration Subsystem (MASS) package (Not redistributable)	/opt/ibm/xlmass/1.2.0/lib/ /opt/ibm/xlmass/1.2.0/lib64/ /opt/ibm/xlmass/1.2.0/include/
libatlas-devel.1.2.0	Automatically Tuned Linear Algebra Software (ATLAS) package (Redistributable)	/opt/ibm/atlas/1.2.0/include /opt/ibm/atlas/1.2.0/lib /opt/ibm/atlas/1.2.0/lib64
libxlc	IBM XL C/C++ runtime package	/opt/ibm/lib /opt/ibm/lib64/
xlc-license-eval.1.2.0 ¹	IBM XL C/C++ license package for the evaluation copy (Not redistributable)	/opt/ibm/xlC/1.2.0/ /opt/ibm/xlC/1.2.0/lap/ /opt/ibm/xlC/1.2.0/lib/
xlc-license.1.2.0 ²	IBM XL C/C++ license package for IBM XL C/C++ for Linux on z Systems, V1.2 (Not redistributable)	
libxlc-devel.1.2.0	IBM XL C/C++ libraries package	/opt/ibm/xlC/1.2.0/lib/ /opt/ibm/xlC/1.2.0/lib64/
xlc.1.2.0	IBM XL C/C++ compiler package (Not redistributable)	/opt/ibm/xlC/1.2.0/bin/ /opt/ibm/xlC/1.2.0/etc/ /opt/ibm/xlC/1.2.0/exe/ /opt/ibm/xlC/1.2.0/include/ /opt/ibm/xlC/1.2.0/lib/ /opt/ibm/xlC/1.2.0/lib64/ /opt/ibm/xlC/1.2.0/msg/ /opt/ibm/xlC/1.2.0/man/
Notes:		
1. Only included in the evaluation copy of IBM XL C/C++ for Linux on z Systems, V1.2.		
2. Only included in the licensed copy of IBM XL C/C++ for Linux on z Systems, V1.2.		

Note: /opt/ibm/ is the default prefix for installation of images.

National language support

IBM XL C/C++ for Linux on z Systems, V1.2 messages support the following language locales:

- en_US
- en_US.utf8
- en_US.UTF-8

en_US is the default locale.

System prerequisites

Ensure that your system meets all prerequisites before installing the product. Failure to meet the prerequisites will cause the installation or configuration of the compiler to fail.

The following are the requirements for installing IBM XL C/C++ for Linux on z Systems, V1.2:

- **Operating system:** You can use any of the following operating systems supported by the IBM z Systems servers:
 - Ubuntu Server 16.04
 - Red Hat Enterprise Linux for IBM System z[®] 6.3 (RHEL 6.3)
 - Red Hat Enterprise Linux for IBM System z 7 (RHEL 7.0)
 - Red Hat Enterprise Linux for IBM System z 7.1 (RHEL 7.1)
 - Red Hat Enterprise Linux for IBM System z 7.2 (RHEL 7.2)
 - SUSE Linux Enterprise Server for System z 11 Service Pack 3 (SLES 11 SP3)
 - SUSE Linux Enterprise Server for System z 12 (SLES 12)
 - SUSE Linux Enterprise Server for System z 12 Service Pack 1 (SLES 12 SP1)
- **Hardware:**

The compiler, its libraries, and its generated object programs run on IBM z Systems, models z13[™], zBC12, zEC12, z114, z196, z10[™] BC, or z10 EC with the required software and disk space.
- **Storage:**
 - Approximately 310 MB for product packages
 - Minimum of 2 GB hard drive space for paging
 - Minimum of 512 MB for temporary files

Note: High levels of optimization can require more space for paging and temporary files.

To verify that you have enough hard disk space available, see “Verifying the amount of hard disk space available” on page 6.

- **Required software:**

To determine which GNU and Perl packages are required to run the compiler on your operating system, consult the relevant table below. To verify that the required packages are installed, see “Verifying that the required GNU and Perl packages are installed” on page 6

Note: Packages marked with an asterisk (*) must be available in both 32-bit and 64-bit versions.

Table 6. Required minimum versions of GNU and Perl packages for the Ubuntu 16.04 operating system

Package name	Version requirements
gcc	4.8.2
g++	4.8.2
libc6	2.19
libstdc++6	4.8.2
libgcc1	4.9
libc6-dev	2.19
libstdc++-dev	4.8.2
perl	5.18.2

Table 7. Required GNU and Perl packages for the RHEL 6.3 operating system

Package name	Version requirements
gcc	4.4.7
gcc-c++	4.4.7
glibc*	2.12
glibc-devel*	2.12
libgcc*	4.4.7
libstdc++*	4.4.7
libstdc++-devel*	4.4.7
zlib	1.2.3
perl	5.10.0 or greater Note: Perl V5.10.1 is shipped and automatically installed with the RHEL 6.3 operating system.

Note: The 32-bit libraries are not installed by default on RHEL 6.3, they must be installed from the RHEL 6.3 operating system image.

Table 8. Required GNU and Perl packages for the RHEL 7.0, RHEL 7.1, and RHEL 7.2 operating system

Package name	Version requirements
gcc	4.8.2
gcc-c++	4.8.2
glibc*	2.17
glibc-devel*	2.17
libgcc*	4.8.2
libstdc++*	4.8.2
libstdc++-devel*	4.8.2
zlib	1.2.7
perl	5.10.0 or greater Note: Perl V5.16.3 is shipped and automatically installed with the RHEL 7.0, RHEL 7.1, and RHEL 7.2 operating system.

Note: The 32-bit libraries are not installed by default on RHEL 7.0, RHEL 7.1, or RHEL 7.2. They must be installed from the RHEL 7.0, RHEL 7.1, and RHEL 7.2 operating system image.

Table 9. Required GNU and Perl packages for the SLES 11 SP3 and operating system

Package name	Version requirements
gcc-32bit	4.3
gcc	4.3
gcc-c++	4.3
glibc-devel-32bit	2.11.3
glibc-devel	2.11.3
glibc-32bit	2.11.3
glibc	2.11.3
libgcc_s1	4.7.2
libgcc_s1-32bit	4.7.2
libstdc++43-devel-32bit	4.3.4
libstdc++-devel	4.3
libstdc++6	4.7.2
libstdc++6-32bit	4.7.2
zlib	1.2.7
perl	5.10.0 or greater Note: Perl V5.10.0 is shipped and automatically installed with the SLES 11 SP3 operating system.

Table 10. Required GNU and Perl packages for the SLES 12 and SLES 12 SP1 operating system

Package name	Version requirements
gcc-32bit	4.8
gcc	4.8
gcc-c++	4.8
glibc-devel-32bit	2.19
glibc-devel	2.19
glibc-32bit	2.19
glibc	2.19
libgcc_s1	4.8.3
libgcc_s1-32bit	4.8.3
libstdc++6-32bit	4.8.3
libstdc++6	4.8.3
libstdc++-devel	4.8
libstdc++-devel-32bit	4.8
zlib	1.2.8

Table 10. Required GNU and Perl packages for the SLES 12 and SLES 12 SP1 operating system (continued)

perl	5.10.0 or greater Note: Perl V5.18.2 is shipped and automatically installed with the SLES 12 and SLES 12 SP1 operating system.
------	--

To view the online documentation for IBM XL C/C++ for Linux on z Systems, V1.2, you need the following environment and tools:

- A graphical desktop environment (such as K Desktop Environment or GNOME) that supports web browsers and PDF viewers
- A frames-capable HTML browser (to access help and other web pages)
- A PDF viewer (to access PDF documentation)

Verifying the amount of hard disk space available

IBM XL C/C++ for Linux on z Systems, V1.2 requires about 310 MB of hard disk storage space.

You can use the following command to determine the amount of space available in the default installation location (`/opt/ibm/`):

```
df -h /opt/ibm
```

If you plan to install the compiler to a nondefault location, you can use the following command instead:

```
df -h installation_path
```

where *installation_path* represents the nondefault location.

Verifying that the required GNU and Perl packages are installed

If you use the `install` utility to install the compiler, you do not need to manually install the dependencies, because the `install` utility will use `yum` (on RHEL) or `zypper` (on SLES) to download and install the dependencies. If you are not using the `install` utility, follow the instructions in this topic to verify the required GNU and Perl packages are installed.

All required packages can be obtained from your operating system installation media.

For lists of the required packages and version requirements for each supported Linux distribution, see the following topics:

- For the Ubuntu 16.04 operating system, see Table 6 on page 4.
- For the RHEL 6.3 operating system, see Table 7 on page 4.
- For the RHEL 7.0, RHEL 7.1, and RHEL 7.2 operating system, see Table 8 on page 4.
- For the SLES 11 SP3 operating system, see Table 9 on page 5.
- For the SLES 12 and SLES 12 SP1 operating system, see Table 10 on page 5.

To verify that the correct versions of the required packages are installed on your system, enter the following command once for each required package:

- **On SLES and RHEL**

```
| rpm -q package_name
|
| • On Ubuntu
|   dpkg -l package_name
```

If the package has been installed, this command should return the name and version number of the package. The version number must be the same as or higher than the version requirement for that package.

Example: Determining the installed version of gcc **On SLES and RHEL**

To see whether gcc is installed, query for the gcc package as follows:

```
rpm -q gcc
```

If gcc version 4.8 is installed, you will get a result similar to the following output:

```
gcc-4.8
```

On Ubuntu

To see whether gcc is installed, query for the gcc package as follows:

```
dpkg -l gcc
```

If gcc version 4.8.2 is installed, you will get a result similar to the following output:

```
Desired=Unknown/Install/Remove/Purge/Hold
Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version             Architecture        Description
+++-----+-----+-----+-----+
ii gcc             4:4.8.2-1ubuntu4    ppc64el             GNU C compiler
```

Previewing the license agreements

Before installing IBM XL C/C++ for Linux on z Systems, V1.2, you can preview the license agreements.

The terms and conditions for using XL C/C++ are specified in the following PDF files, which are located in the root directory of the distribution package:

- LicenseAgreement.pdf
- license.pdf

After basic installation, the license files can be found in the /opt/ibm/xlC/1.2.0/ directory with the same names as above.

Chapter 2. Basic installation

IBM XL C/C++ for Linux on z Systems provides an interactive utility, `install`, that walks you through a basic installation. You can use `install` to perform any of the following tasks:

- Install IBM XL C/C++ for Linux on z Systems, V1.2 on a system where no IBM XL compiler is currently installed.
- Install an update on a system where IBM XL C/C++ for Linux on z Systems, V1.2 has already been installed.

During the installation with the `install` utility, you are prompted to accept or decline the license agreement. If you accept the agreement, the license files will be output to `.txt` files for your future reference. If you decline the agreement, the installation process will exit; the compiler will remain unconfigured, and will not be usable.

You should consider a basic installation of IBM XL C/C++ for Linux on z Systems, V1.2 with the `install` utility on a system as long as you are installing the compiler to the default location, `/opt/ibm/`. If you are installing the compiler to a nondefault location, use the procedures in Chapter 3, “Advanced installation,” on page 13.

Tasks for basic installation

This section lists the procedures you must perform before, during, and after a basic installation.

If the conditions listed in Chapter 2, “Basic installation” match your needs, a basic installation is the easiest and fastest method you can use to install. A basic installation allows you to install the latest version of IBM XL C/C++ for Linux on z Systems compiler and configure the compiler, all through the use of a single installation tool.

Table 11. Steps for basic installation

Task	References
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system.
Use the <code>install</code> utility to install and configure the compiler, using the default paths.	“Running the install utility for a new installation” on page 10
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none">• “Querying for installed packages” on page 27• “Testing the installation” on page 23
Enable the compiler man pages.	“Enabling the manual pages” on page 24
If your system locale and/or encoding are <i>not</i> <code>en_US</code> , enable the compile-time error messages for your locale. Otherwise, you can skip this step.	“Enabling the error messages” on page 25

Running the install utility for a new installation

The `install` utility is the recommended tool for installing XL C/C++. It automatically installs the correct packages and their dependencies based on the installed Linux distribution. The utility is located in the root directory of the installation image.

About this task

When you run the `install` utility for a new installation, it performs the following tasks:

- Detects the current operating system
- Installs all prerequisite software packages (using `apt-get`, `zypper`, or `yum`)
- Installs all compiler packages into the default location, `/opt/ibm/`
- Automatically invokes the `xlc_configure` utility, which installs the license file and generates the default configuration file
- Creates symbolic links in `/usr/bin/` to the compiler invocation commands

Procedure

1. If your product is downloaded from an IBM website, issue the following commands:

```
tar -zxvf package_name.tar.gz
./install
```

where *package_name* is the name of the compiler package you downloaded.

If the product DVD is mounted on the `/cdrom` directory in the system, issue the following commands:

```
cd /cdrom
./install
```

2. Read the license agreement and licensing information. If you agree to the licensing terms, accept the license agreement to continue installation.

Results

The symbolic links are created automatically (using the `update-alternatives` command).

Tip: An alternative to this step is to add the path that contains the compiler invocations to the `PATH` environment variable. See “Setting the `PATH` environment variable to include the path to the compiler invocations” on page 25.

The following links are created in the `/usr/bin/` subdirectory:

- `xlc`
- `xlc++`
- `xlC`
- `xlc_r`
- `xlc++_r`
- `xlC_r`

If all packages are successfully installed:

- The install Perl script returns 0 and the following messages are displayed confirming the successful installation:

Installation and configuration successful

|
|
|
|
|
|
|

- The configuration file is generated. Its location is `/opt/ibm/xlC/1.2.0/etc/xlc.cfg.$OS.$OSver.gcc.$gccver`. Examples of the configuration file are listed below:
 - `/opt/ibm/xlC/1.2/etc/xlc.cfg.ubuntu.14.04.gcc.4.8.2`
 - `/opt/ibm/xlC/1.2/etc/xlc.cfg.sles.11.gcc.4.3`
 - `/opt/ibm/xlC/1.2/etc/xlc.cfg.rhel.6.3.gcc.4.4.7`

Chapter 3. Advanced installation

It is highly recommended that you install IBM XL C/C++ for Linux on z Systems, V1.2 to the default location and use the procedure provided in Chapter 2, “Basic installation,” on page 9. However, you must use alternative procedures if any of the following scenarios apply to you:

- You want to install IBM XL C/C++ for Linux on z Systems to a nondefault location.
- You want to try out a new update of the compiler before removing an existing installation from the default location.

Installing on SLES and RHEL

In both scenarios, you can use the rpm utility to install the compiler packages to a single, nondefault location. After you have successfully installed the compiler packages to a nondefault location using the rpm utility, you need to manually configure the compiler environment using the xlc_configure utility. For configuration procedures, see Chapter 5, “Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users),” on page 21.

Installing on Ubuntu

In this scenario, you can use the dpkg and the chroot utilities to install the compiler packages to a single, nondefault location. After you have successfully installed the compiler packages to a nondefault location using the dpkg utility, you need to manually configure the compiler environment using the xlc_configure utility. For configuration procedures, see Chapter 5, “Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users),” on page 21.

Tasks for advanced installation

This section lists the procedures you must perform before, during, and after an advanced installation.

If you want to install IBM XL C/C++ for Linux on z Systems, V1.2 to a nondefault location, follow the steps in this table.

Table 12. Steps for installing to a nondefault location

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
Ensure that all system prerequisites are satisfied.	“System prerequisites” on page 3
Install IBM XL C/C++ for Linux on z Systems, V1.2.	“Installing XL C/C++ to a nondefault location (SLES and RHEL)” on page 14
Use the xlc_configure utility to configure the compiler.	Chapter 5, “Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users),” on page 21
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none">• “Querying for installed packages” on page 27• “Testing the installation” on page 23

Table 12. Steps for installing to a nondefault location (continued)

Task	Further information
Enable the compiler man pages.	“Enabling the manual pages” on page 24
If your system locale, your encoding, or both, are <i>not</i> en_US, enable the compile-time error messages for your locale. Otherwise, you can skip this step.	“Enabling the error messages” on page 25
Optional: Set up the environment to locate the invocation commands without the full path.	“Setting up the environment for the invocation commands” on page 25

Installing XL C/C++ to a nondefault location (SLES and RHEL)

About this task

Use this procedure if you want to install all compiler packages to a single nondefault location. For example, rather than installing all packages into the default directory /opt/ibm/, you could install them to a directory such as /home/mydirectory/.

Procedure

To install all compiler packages to a single nondefault directory, use the rpm utility. Ensure that your current working directory contains all of the packages for IBM XL C/C++ for Linux on z Systems, V1.2 and no other RPM packages. From your current working directory, use the following command:

```
cd /cdrom/images/$OS/rpms
rpm -Uvh libxlc-1.2.0.0-$B.s390x.rpm xlc-license.1.2.0-1.2.0.0-$B.s390x.rpm
rpm -Uvh libxlmass-devel.1.2.0-1.2.0.0-$B.s390x.rpm \
libatlas-devel.1.2.0-1.2.0.0-$B.s390x.rpm \
libxlc-devel.1.2.0-1.2.0.0-$B.s390x.rpm \
xlc.1.2.0-1.2.0.0-$B.s390x.rpm --prefix $installation_path
$installation_path/xlc/1.2.0/bin/xlc_configure
```

where *installation_path* is a directory that is not /opt/ibm/, *\$OS* is rhel, sles11, or sles12, and *\$B* is the build number of the package.

Installing XL C/C++ to a nondefault location (Ubuntu)

About this task

Use this procedure if you want to install all compiler packages to a single nondefault location. For example, rather than installing all packages into the default directory /opt/ibm/, you could install them to a directory such as /home/mydirectory/.

Procedure

To install all compiler packages to a single nondefault directory, use the rpm utility. Ensure that your current working directory contains all of the packages for IBM XL C/C++ for Linux on z Systems, V1.2 and no other RPM packages. From your current working directory, use the following command:

```
| cd /cdrom/images/$OS/rpms
| rpm -Uvh libxlc-1.2.0.0-$B.s390x.rpm xlc-license.1.2.0-1.2.0.0-$B.s390x.rpm
| rpm -Uvh libxlmass-devel.1.2.0-1.2.0.0-$B.s390x.rpm \
| libatlas-devel.1.2.0-1.2.0.0-$B.s390x.rpm \
| libxlc-devel.1.2.0-1.2.0.0-$B.s390x.rpm \
| xlc.1.2.0-1.2.0.0-$B.s390x.rpm --prefix $installation_path
| $installation_path/xlc/1.2.0/bin/xlc_configure
```

| where *installation_path* is a directory that is not /opt/ibm/, *\$OS* is rhel, sles11, or
| sles12, and *\$B* is the build number of the package.

Querying for RPM package summaries (SLES and RHEL)

Querying for the summary of an RPM package is useful when you do not know which product the package belongs to. For instance, it might be useful if you have moved or copied packages from the XL C/C++ DVD or E-Image layout. The RPM package summary includes a short description of the queried RPM file.

To query for the summary of an RPM package that is not installed, issue the command:

```
$ rpm --qf="{summary}\n" -qp rpm_file_name
```

For instance, to query for the summary of the libxlc-devel.1.2.0-1.2.0.0-\$B.s390x.rpm RPM file that has not yet been installed, issue the following command:

```
$ rpm --qf="{summary}\n" -qp libxlc-devel.1.2.0-1.2.0.0-$B.s390x.rpm
```

where *\$B* is the build number of the RPM package that you are querying for.

The resulting output depends on the specific file queried, but the following is a typical example:

```
IBM XL C/C++ for Linux on z Systems, V1.2 - XL C/C++ Compiler Libraries
Package (SLES11/SLES12/RHEL6/RHEL7)
```

Querying for dpkg package summaries (Ubuntu)

Querying for the summary of a dpkg package is useful when you do not know which product the package belonged to. For instance, it might be useful if you have moved or copied packages from the XL C/C++ CD or E-Image layout. The dpkg package summary includes a short description of the queried dpkg file.

To query for the summary of a dpkg package that is not installed, issue the command:

```
$ dpkg -f pkg_file_name
```

where *pkg_file_name* is the full file name of the dpkg package that you want to query for a summary.

For instance, to query for the summary of the xlc.1.2.0_1.1.0.0-\$B_ppc64el.deb file that has not yet been installed, issue the following command:

```
$ dpkg -f xlc.1.2.0_1.1.0.0-$B_ppc64el.deb
```

The resulting output depends on the specific file queried, but the following is a typical example:

```
Package xlc.1.2.0
Version: 1.2.0.0-$B
```

Section: Development/Tools/Compilers

Priority: extra

Architecture: ppc64el

Maintainer: For reporting bugs, please open a PMP instead of sending an email
<noreply@ibm.com>

Depends: debconf, perl-modules, gcc, g++, libxlmass-devel.1.1.0

Description: IBM XL
C/C++ for Linux on z Systems, V1.2 (\$product_id) - C/C++ compiler package

Licensed Materials - Property of IBM

IBM XL
C/C++ for Linux on z Systems, V1.2 (\$product_id)

Copyright IBM Corp. 1991, 2016

IBM is a registered trademark of IBM Corp. in the U.S.

other countries or both.

US Government Users Restricted Rights -

Use, duplication or disclosure restricted by

GSA ADP Schedule Contract with IBM Corp.

Note: \$B is the build number of the package that is currently installed on your system.

Chapter 4. Installing an update

An update, or program temporary fix (PTF), of IBM XL C/C++ for Linux on z Systems, V1.2 provides a fix or multiple fixes to the product. You can download updates from the support website: http://www.ibm.com/support/entry/portal/product/rational/xl_c/c++_for_linux_on_z_systems.

Every PTF update package comes in tar.gz (compressed) format and includes a version of the `install` utility that is customized to install only the update that accompanies it. If you have any version (including an earlier update) of IBM XL C/C++ for Linux on z Systems, V1.2 installed on your system, you can apply the latest update. If you are applying the update to an installation in the default location, follow the procedure in “Updating a basic installation” on page 18. If you are applying the update to an installation in a nondefault location, follow the instructions in Chapter 3, “Advanced installation,” on page 13 to install the PTF packages to the location of your choice. If you want to try out a PTF before installing it, use the procedure in “Testing a PTF before you install it” on page 18.

Tasks for update installation

This section lists the procedures you must perform before, during, and after an update installation.

If the compiler has been installed to the default directory, `/opt/ibm/`, you must follow the steps outlined in the following table.

Table 13. Steps for basic installation: update installation

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
Use the <code>install</code> utility to install the update packages.	“Updating a basic installation” on page 18
Confirm that the compiler packages have been successfully installed, and test the installation.	<ul style="list-style-type: none">• “Querying for installed packages” on page 27• “Testing the installation” on page 23

If the compiler has been installed to any directory other than `/opt/ibm/`, you must follow the steps outlined in the following table.

Table 14. Steps for advanced installation: update installation

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
Install update packages to the nondefault location.	<ul style="list-style-type: none">• “Installing XL C/C++ to a nondefault location (Ubuntu)” on page 14• “Installing XL C/C++ to a nondefault location (SLES and RHEL)” on page 14
Configure the compiler.	Chapter 5, “Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users),” on page 21

Table 14. Steps for advanced installation: update installation (continued)

Task	Further information
Confirm that the compiler packages have been successfully installed, and test the installation.	<ul style="list-style-type: none">• “Querying for installed packages” on page 27• “Testing the installation” on page 23
Optional: Set up the environment to locate the invocation commands without the full path.	“Setting up the environment for the invocation commands” on page 25

Testing a PTF before you install it

If you want to try out a new update to the compiler before you remove the existing version from the system, you must install the new update to a nondefault location.

Procedure

To install a PTF to a nondefault location, use the procedure described in “Installing XL C/C++ to a nondefault location (SLES and RHEL)” on page 14 or “Installing XL C/C++ to a nondefault location (Ubuntu)” on page 14. To configure the compiler, use the procedure described in “Running the `xlc_configure` utility directly” on page 21.

What to do next

After you have finished testing the new PTF and have verified that you want to replace the older version with the new update, install the PTF to the default location.

For instructions on how to use the `install` utility to install the new update, see “Updating a basic installation.”

If the existing version on the system was installed using an advanced installation, follow the instructions in “Updating multiple versions of PTFs installed under separate locations” on page 19 to install the PTF packages to the location of your choice.

Updating a basic installation

Updates to the compiler are supplied as PTF packages. You can follow the instructions in this section to download, uncompress, and install the update packages.

Procedure

1. Download the PTF package that you want into an empty directory.
2. Restore the compressed file and extract the PTF filesets from the downloaded package. To uncompress and unpack the TAR file, use the following command:

```
tar -zxvf package_name.tar.gz
```

where *package_name* is the name of the update package you downloaded.

3. Install the update package: Run the `install` utility from the directory where you extracted the PTF package to:

```
./install
```


Alternatively, you can install the update package manually:

- **SLES:**
rpm -Uvh images/SLES/rpms/*.rpm
- **RHEL:**
rpm -Uvh images/RHEL/rpms/*.rpm
- **Ubuntu:**
dpkg -iG images/Ubuntu/rpms/*.deb

where *OS* is your current operating system.

Updating multiple versions of PTFs installed under separate locations

On SLES and RHEL

To update multiple versions of PTF, take the following steps:

1. Run the RPM utility to install the PTF packages with the `-U` option by issuing the following commands:

```
cd /cdrom/images/$OS/rpms
rpm -Uvh libxlc-1.2.0.$F-$B.s390x.rpm xlc-license.1.2.0-1.2.0.$F-$B.s390x.rpm
rpm -Uvh libxlmass-devel.1.2.0-1.2.0.$F-$B.s390x.rpm \
libatlas-devel.1.2.0-1.2.0.$F-$B.s390x.rpm \
libxlc-devel.1.2.0-1.2.0.$F-$B.s390x.rpm \
xlc.1.2.0-1.2.0.$F-$B.s390x.rpm --prefix $installation_path
$installation_path/xlc/1.2.0/bin/xlc_configure
```

where *\$OS* is your current operating system, *\$F* is the fix level of the package, *\$B* is the build number of the package, and *\$installation_path* is the target directory for installation specified by `--prefix` option of the nondefault installation command.

2. Test the updated compiler. For instructions, see “Testing the installation” on page 23.

On Ubuntu

To update multiple versions of PTF, take the following steps:

1. Run the `dpkg` utility to install the PTF packages under `chroot`. For instructions, see “Installing XL C/C++ to a nondefault location (Ubuntu)” on page 14.
2. Test the updated compiler. For instructions, see “Testing the installation” on page 23.

Chapter 5. Configuring IBM XL C/C++ for Linux on z Systems, V1.2 (for advanced users)

Before you can run IBM XL C/C++ for Linux on z Systems, V1.2, you must configure (or re-configure) the compiler if any of the following conditions apply to you:

- You did not use `install` to install the compiler, or the configuration step failed with `install`.
- Your system or its GCC configuration is changed after the compiler configuration was last run.
- Compiler components were relocated after installation.

The compiler provides a configuration tool, `xlc_configure`, located in the `installation_path/xlc/1.2.0/bin/` directory after installation.

Note: `installation_path` is the installation location of the compiler packages. If the compiler is installed in the default location, `installation_path` is `/opt/ibm/`.

You must invoke the `xlc_configure` utility directly if any of the following conditions is true:

- You did not use `install` to install the compiler.
- You have multiple versions of IBM XL C/C++ for Linux on z Systems installed on your system.
- You want the generated configuration file to be placed in a location that is different from `installation_path/xlc/1.2.0/etc/`.
- You have multiple versions of GCC installed on your system and you need to specify which GCC version you would like to reference in the configuration file.

Note: If you configure the compiler using `xlc_configure`, your output configuration file, `xlc.cfg.$OS.$OSver.gcc.$gccver`, can be written to any location where you have write permission. You need root privileges to accept the license for the first time you run the configuration. Subsequent reconfiguration does not require root privileges if the license was accepted.

For instructions, see “Running the `xlc_configure` utility directly.”

Running the `xlc_configure` utility directly

Use the `xlc_configure` utility to configure an advanced installation if multiple versions of XL C/C++ or multiple versions of GCC are installed on your system.

Procedure

Become a superuser with `su`, or `sudo`, and run the following command:

```
installation_path/xlc/1.2.0/bin/xlc_configure options
```

where `installation_path` is the installation location of the compiler packages. If the compiler is installed in the default location, `installation_path` is `/opt/ibm/`.

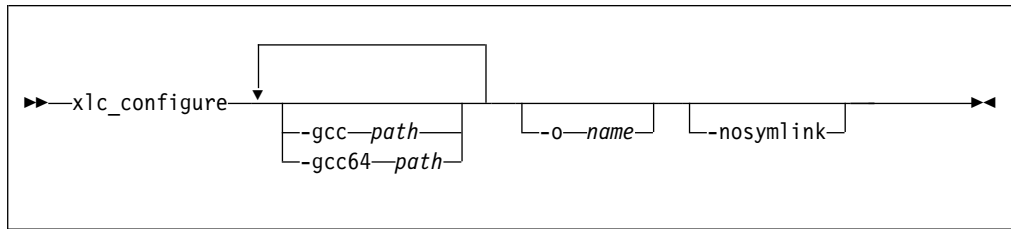
Results

You are presented with the license agreement and licensing information. Read the license agreement and licensing information. If you agree to the licensing terms, accept the license agreement to continue configuration.

For details about required arguments to the `xlc_configure` command, see “`xlc_configure` options.”

`xlc_configure` options

The `xlc_configure` command has the following syntax:



where:

-gcc path

Specifies the path where the native GCC bin/ directory is installed. For example, if the GCC command is `/usr/bin/gcc`, you can specify:

```
-gcc /usr
```

By default, *path* is `/usr`.

-gcc64 path

Specifies the path where the native 64-bit GCC bin/ directory is installed. For example, if the 64-bit GCC command is `/usr/bin/gcc -m64`, you can specify:

```
-gcc64 /usr
```

By default, *path* is `/usr`.

-o file_name

Specifies the name of the configuration file to be generated. If this option is not specified, the configuration file is written to the installation location of the compiler based on OS distribution and gcc version. For example, `/opt/ibm/xlC/1.2/etc/xlc.cfg.sles.11.gcc.4.3` or `/opt/ibm/xlC/1.2/etc/xlc.cfg.rhel.6.3.gcc.4.4.7`

-nosymlink

Specifies not to create symbolic links in `/usr/bin`. If this option is not specified, the following symbolic links are created in `/usr/bin`:

- `xlc`
- `xlc++`
- `xlC`
- `xlc_r`
- `xlc++_r`
- `xlC_r`

Chapter 6. After installing IBM XL C/C++ for Linux on z Systems, V1.2

After installing the compiler, you might need to follow the verification and setup procedures. These are documented in the following sections:

- Testing the installation: This section applies to all users.
- Enabling the man pages: This section applies to all users.
- Enabling the error messages: This section only applies to users whose system uses a locale or language encoding other than en_US.
- Setting up the environment for the invocation commands: This section only applies to users who have not used `install` to install the product, or who have not created symbolic links during the configuration process with `xlc_configure`.
- Enabling IBM License Metric Tool (ILMT) and Tivoli® Asset Discovery for Distributed (TADd): This section applies to all users.
- Accessing the local documentation: This section applies to all users.
- Querying for installed packages: This section applies to all users.

Testing the installation

To test the product installation and the critical search paths, build and run a sample application.

Procedure

1. Create the following C program and name the source file `hello.c`:

```
#include <stdio.h>

int main(void)
{
    printf("Hello World!\n");
    return 0;
}
```

2. Compile the program:

If you have set up the short invocation commands, enter the following command:

```
xlc hello.c -o hello
```

If you have not set up the short invocation commands, enter the following command:

```
installation_path/xlc/1.2.0/bin/xlc hello.c -o hello
```

where *installation_path* is the installation location of the compiler packages. If the compiler has been installed to the default location, *installation_path* is `/opt/ibm/`.

3. Run the program by entering the following command:

```
./hello
```

The result is "Hello World!".

4. Check the exit code of the program by entering the following command:

```
echo $?
```

The result is "0".

5. Create the following C++ program and name the source file `hello.cpp`:

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello World!\n";
    return 0;
}
```

6. Compile the program:

If short invocation commands have been set up, enter the following command:

```
xlc++ hello.cpp -o hello
```

If short invocation commands have not been set up, enter the following command:

```
installation_path/xlc/1.2.0/bin/xlc++ hello.cpp -o hello
```

where *installation_path* is the installation location of the compiler packages. If the compiler has been installed to the default location, *installation_path* is `/opt/ibm/`.

7. Run the program :

```
./hello
```

The result should be "Hello World!".

8. Check the exit code of the program:

```
echo $?
```

The result should be "0".

Enabling the manual pages

Manual pages are provided for the compiler invocation commands and other utilities that are supplied with the compiler.

The IBM XL C/C++ for Linux on z Systems, V1.2 manual pages support the following locales:

- `en_US`
- `en_US.utf8`
- `en_US.UTF-8`

However, before you can read the compiler-supplied manual pages, you must add the full directory path to the `MANPATH` environment variable. The command that accomplishes this depends on the Linux shell that you are using.

- To set the `MANPATH` environment variable using the Bourne, Korn, or BASH shell, use the following command:

```
export MANPATH=installation_path/xlc/1.2.0/man/LANG:$MANPATH
```

- To set the `MANPATH` environment variable using the C shell, use the following command:

```
setenv MANPATH installation_path/xlc/1.2.0/man/LANG:$MANPATH
```

where:

- *installation_path* is the location where you have installed the XL C/C++ packages. By default, this is `/opt/ibm/`.
- *LANG* is any of the language locales as shown in the preceding list.

Note: To set this variable in the Bourne, Korn, or BASH shell so that it applies to all users, add the command to the file `/etc/profile`. To set it for a specific user only, add the command to the file `.profile` in the user's home directory. To set this variable in the C shell so that it applies to all users, add the command to the file `/etc/csh.cshrc`. To set it for a specific user only, add the command to the file `.cshrc` in the user's home directory. The environment variable is set each time the user logs in.

To test whether the manual pages have been successfully installed, open one or more of the manual pages as described in “Viewing the manual pages” on page 26.

Enabling the error messages

The compiler message catalogs are automatically configured to display correctly, regardless of whether you used the basic or advanced method of installation and configuration.

Setting up the environment for the invocation commands

If you used the `install` utility to install the compiler, you have already set up the environment for the invocation commands. Do not perform the procedures in this section.

If you did not select to create the symbolic links when you configured the compiler and want to invoke the compiler without having to specify the full path, you must perform one of the following tasks:

- Set the `PATH` environment variable, as shown in “Setting the `PATH` environment variable to include the path to the compiler invocations.”
- Create symbolic links to the compiler invocation commands, as shown in “Creating symbolic links to the compiler invocations” on page 26.

Setting the `PATH` environment variable to include the path to the compiler invocations

To use IBM XL C/C++ for Linux on z Systems, V1.2 commands without typing the complete path, you can add the location of the compiler invocations to the `PATH` environment variable.

The command to set the `PATH` environment variable depends on the shell that you are using.

- If you are using the Bourne, Korn, or BASH shell, use the following command:

```
export PATH=$PATH:installation_path/x1c/1.2.0/bin/
```
- If you are using the C shell, use the following command:

```
setenv PATH $PATH:installation_path/x1c/1.2.0/bin/
```

where `installation_path` is the location where you have installed the compiler packages. By default, this is `/opt/ibm/`.

Note: To set this variable in the Bourne, Korn, or BASH shell so that it applies to all users, add the command to the file `/etc/profile`. To set it for a specific user only, add the command to the file `.profile` in the user's home directory. To set this variable in the C shell so that it applies to all users, add the command to the file `/etc/csh.cshrc`. To set it for a specific user only, add the command to the file `.cshrc` in the user's home directory. The environment variable is set each time the user logs in.

Creating symbolic links to the compiler invocations

To use the compiler without typing the complete path, you can create symbolic links in the `/usr/bin/` directory for the specific invocations that are contained in the `installation_path/xlC/1.2.0/bin/` directory.

If you have not already done so when you ran the `rpm` or `dpkg` utility, you can create the symbolic links for the following compiler invocations:

- `xlC`
- `xlC_r`
- `xlC++`
- `xlC++_r`

Links to some invocations are not recommended, either because they delete user-defined or GCC invocations, or because they are not compiler invocation commands. These include the following commands:

- `c89`, `c89_r`, `c99`, `c99_r`, `cc`, `cc_r`
- `cleanpdf`, `mergepdf`, `showpdf`, `xlC_configure`

Enter the following command to create all the symbolic links:

```
for exec in xlC xlC++ xlC xlC_r xlC++_r xlC_r;
do update-alternatives --install /usr/bin/$exec $exec installation_path/xlC/1.2.0/ \
bin/$exec 2;
done
```

where `installation_path` is the location where you have installed the compiler packages. By default, this is `/opt/ibm/`.

Accessing the local documentation

Manual pages for the compiler invocation commands and other command utilities are included.

Viewing the manual pages

Manual pages are included for all compiler invocation commands and utilities.

For instructions about how to enable the man pages for viewing, see “Enabling the manual pages” on page 24.

To invoke a manual page, enter the following command:

```
man command
```

Example:

```
man xlC
```

Querying for installed packages

To determine the Version.Release.Modification.Fix-Build level of a package, query for it using the rpm or dpkg command.

On SLES and RHEL

To query for an individual package, enter the following command:

```
rpm -q xlc
```

The result is:

```
xlc.V.R.M-V.R.M.F-B
```

where *V.R.M-V.R.M.F-B* is the Version.Release.Modification-Version.Release.Modification.Fix-Build level of the compiler that is installed on the system.

If the installation was not successful, you will get a message indicating that the package has not been installed.

To confirm the installation of all compiler packages, enter the following command:

```
rpm -qa | grep -e xlc -e libxlmass -e libatlas
```

The result is a list containing all of the packages as shown in Table 5 on page 2. If none of the packages was properly installed, there will be no output from the command.

On Ubuntu

To query for an individual package, enter the following command:

```
dpkg -p xlc
```

The result is:

```
Package xlc
```

```
Version: V.R.M.F-B
```

where *V.R.M.F-B* is the Version.Release.Modification.Fix-Build level of the compiler that is installed on the system.

If the installation is not successful, you will get a message indicating that the package has not been installed.

To confirm the installation of all compiler packages, enter the following command:

```
dpkg -l | grep -e xlc -e xlc
```

The result is a list containing all of the packages as shown in Table 5 on page 2. If none of the packages was properly installed, there will be no output from the command.

Chapter 7. Uninstalling IBM XL C/C++ for Linux on z Systems, V1.2

IBM XL C/C++ for Linux on z Systems, V1.2 does not provide a stand-alone uninstallation tool. You must use the Linux rpm or dpkg utility to uninstall IBM XL C/C++ for Linux on z Systems, V1.2.

Notes:

- You must have root access to uninstall the compiler.
- Whenever you uninstall a package that is not a PTF package, specify the package name. If you are uninstalling a PTF package, specify the *V.R.M.F* (Version.Release.Modification.Fix level) of the package. For information about how to determine the package's *V.R.M.F*, see “Querying for installed packages” on page 27.
- Always uninstall packages in the reverse order in which they have been installed, that is, the last package that has been installed is the first package that you remove.
- On Ubuntu, the dpkg utility provides the purge option, **-P**, which removes the configuration files that are under the compiler installation path and have been generated by the `xlc_configure` utility. If you want to keep the configuration files while uninstalling the compiler, use the **-r** option with the dpkg command. Any configuration files generated outside of the compiler installation path (such as home directories) will not be removed.
- On SLES and RHEL, the rpm uninstallation command is equivalent to the dpkg purge (**-P**) option, all configuration files under the compiler installation path will be removed. Any configuration files generated outside of the compiler installation path (such as home directories) will not be removed.

Example: Uninstalling IBM XL C/C++ for Linux on z Systems, V1.2

When uninstalling XL C/C++, you must remove many of the packages in a specific order to avoid dependency errors.

Example 1

To uninstall IBM XL C/C++ for Linux on z Systems, V1.2, take the following steps:

1. Issue the following commands in the order given below to uninstall those C/C++ specific packages:

```
rpm -e xlc.1.2.0
rpm -e libxlc-devel.1.2.0
rpm -e xlc-license.1.2.0
rpm -e libatlas-devel.1.2.0
rpm -e libxlmass-devel.1.2.0
```
2. Optional: If other applications are using the runtime libraries and you want them to continue to function normally, do not uninstall the runtime package. Otherwise, uninstall the following package:

```
rpm -e libxlc
```

Example 2

In this example:

- The compiler packages have a *V.R.M.F* of 1.2.0.*\$F*.
- The IBM MASS library package has a *V.R.M.F* of 1.2.0.*\$F*.
- The ATLAS library package has a *V.R.M.F* of 1.2.0.*\$F*.

Note: *\$F* is the fix level of the package that is currently installed on your system.

To uninstall a PTF that is installed in a nondefault location, take the following steps:

1. Issue the following commands in the order given below to uninstall those C/C++ specific packages:
rpm -e xlc.1.2.0-1.2.0.*\$F*
rpm -e libxlc-devel.1.2.0-1.2.0.*\$F*
rpm -e xlc-license.1.2.0-1.2.0.*\$F*
rpm -e libatlas-devel.1.2.0-1.2.0.*\$F*
rpm -e libxlmass-devel.1.2.0-1.2.0.*\$F*
2. Optional: If other applications are using the runtime libraries and you want them to continue to function normally, do not uninstall the runtime package. Otherwise, uninstall the following package:
rpm -e libxlc-1.2.0.*\$F*

Notices

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

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

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

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. 2015.

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.

Index

A

- advanced installation
 - overview 13
- advanced user, described v
- available space, determining 6

B

- basic example, described ix
- basic installation
 - overview 9
 - procedure 10
- basic user, described v

C

- configuration files
 - backup 21
 - customizing 21
 - editing 21
 - generation 11
 - modifying 21
 - multiple 21
 - security 29

D

- default installation 9
- documentation
 - manual pages 26

E

- environment variable 25
- error messages 25

G

- GNU packages
 - required 3
 - verifying 6

H

- hard disk space
 - required 3
 - verifying 6
- hardware requirements 3
- help 26
 - manual pages 24

I

- installation
 - multiple versions 13
 - multiple versions of PTFs 19
 - testing 23

- installation DVD 1
- installation image 1
- installation utility
 - using 17
- invocation commands 25

L

- languages 3
- locales 3

M

- manual pages 24
 - viewing 26

N

- national language support 3
- nondefault installation 13

O

- operating systems, supported 3

P

- packages
 - installation 2
 - querying 27
- PATH environment variable 25
- Perl packages
 - required 3
 - verifying 6
- preinstallation 1
- PTF 17

Q

- querying
 - dpkg file summaries 15
 - installed packages 27
 - RPM file summaries 15

R

- Red Hat Package Manager (RPM) 1, 2

S

- software requirements 3
- symbolic links 10
 - creating 26

T

- testing
 - installation 23
 - PTF 18

U

- uninstallation
 - example 29
 - overview 29
- updates
 - a nondefault location 18
 - applying 18
 - overview 17
 - trying out 18

V

- verifying
 - GNU and Perl packages 6
 - hard disk space 6



Product Number: 5725-N01

Printed in USA

GC27-5995-01

