IBM XL C/C++ for Linux, V16.1

Installation Guide for Little Endian Distributions

Version 16.1

GC27-8039-00

IBM XL C/C++ for Linux, V16.1

Installation Guide for Little Endian Distributions

Version 16.1

GC27-8039-00

Note

Before using this information and the product it supports, read the information in “Notices” on page 35.

First edition

This edition applies to IBM XL C/C++ for Linux, V16.1 (Program 5765-J08; 5725-C73) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

© Copyright IBM Corporation 2003, 2018.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 2003, 2018

About this document

This guide provides detailed installation instructions for IBM® XL C/C++ for Linux, V16.1 for little endian distributions. 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 files in the directory tree of your installation media, which contain the current information about the compilers.

Who should read this document

This document is intended for anyone who is responsible for installing, upgrading and uninstalling IBM XL C/C++ for Linux, V16.1 for little endian distributions.

How to use this document

This document provides installation, upgrading and uninstallation instructions for XL C/C++ for Linux for little endian distributions.

How this document is organized

The document is organized to reflect the installation, upgrading, and product removal tasks for IBM XL C/C++ for Linux for little endian distributions.

Table 1. Tasks of IBM XL C/C++ for Linux (for little endian)

Task	Chapters
Installation	Chapter 1, "Installing the compiler," on page 1
Upgrading to the latest release	Chapter 2, "Upgrading to the latest release," on page 25
Upgrading to the latest fix pack	Chapter 3, "Updating to the latest fix pack," on page 27
Upgrading from Community Edition to full version	Chapter 4, "Updating from Community Edition to full version," on page 31
Uninstallation	Chapter 5, "Uninstalling the compiler," on page 33

Conventions

Typographical conventions

The following table shows the typographical conventions used in the IBM XL C/C++ for Linux, V16.1 information.

Table 2. Typographical conventions

Typeface	Indicates	Example
bold	Lowercase commands, executable names, compiler options, and directives.	The compiler provides basic invocation commands, **xlc** and **xlC** (**xlc++**), along with several other compiler invocation commands to support various C/C++ language levels and compilation environments.

© Copyright IBM Corp. 2003, 2018

Table 2. Typographical conventions (continued)

Typeface	Indicates	Example
italics	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 *size* parameter if you return more than the *size* requested.
underlining	The default setting of a parameter of a compiler option or directive.	nomaf | maf
`monospace`	Programming keywords and library functions, compiler builtins, examples of program code, command strings, or user-defined names.	To compile and optimize myprogram.c, enter: `xlc myprogram.c -O3`.

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

Icon	Short description	Meaning
C C	C only begins / C only ends	The text describes a feature that is supported in the C language only; or describes behavior that is specific to the C language.
C++ C++	C++ only begins / C++ only ends	The text describes a feature that is supported in the C++ language only; or describes behavior that is specific to the C++ language.
C11 C11	C11 begins / C11 ends	The text describes a feature that is introduced into standard C as part of C11.
C++11 C++11	C++11 begins / C++11 ends	The text describes a feature that is introduced into standard C++ as part of C++11.
C++14 C++14	C++14 begins / C++14 ends	The text describes a feature that is introduced into standard C++ as part of C++14.
IBM IBM	IBM extension begins / IBM extension ends	The text describes a feature that is an IBM extension to the standard language specifications.

Table 3. Qualifying elements (continued)

Icon	Short description	Meaning
GPU ▶ ◀ GPU	GPU begins / GPU ends	The text describes the information that is relevant to offloading computations to the NVIDIA GPUs.

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):

```
►►—keyword—required_argument——————————————————————————————————►◄
```

- Optional items are shown below the main path:

```
►►—keyword—┬—————————————————┬————————————————————————————————►◄
           └optional_argument┘
```

- If you can choose from two or more items, they are shown vertically, in a stack.

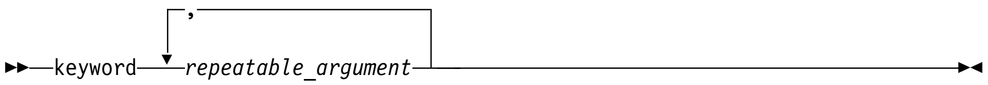
  If you *must* choose one of the items, one item of the stack is shown on the main path.

```
►►—keyword—┬required_argument1┬———————————————————————————————►◄
           └required_argument2┘
```

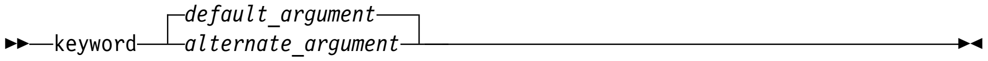
  If choosing one of the items is optional, the entire stack is shown below the main path.

```
►►—keyword—┬——————————————————┬———————————————————————————————►◄
           ├optional_argument1┤
           └optional_argument2┘
```

- 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 default installation; these need little or no modification.

Related information

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

Available help information

IBM XL C/C++ information

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

- Quick Start Guide

The Quick Start Guide (`quickstart.pdf`) is intended to get you started with IBM XL C/C++ for Linux, V16.1. 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, V16.1 Installation Guide*.

- Online product documentation

  The fully searchable HTML-based documentation is viewable in IBM Knowledge Center at http://www.ibm.com/support/knowledgecenter/SSXVZZ_16.1.0/com.ibm.compilers.linux.doc/welcome.html.

- PDF documents

  PDF documents are available on the web at https://www.ibm.com/support/knowledgecenter/SSXVZZ_16.1.0/com.ibm.compilers.linux.doc/download_pdf.html.

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

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

Table 4. XL C/C++ PDF files

Document title	PDF file name	Description
What's New for IBM XL C/C++ for Linux, V16.1, GC27-8041-00	`whats_new.pdf`	Provides an executive overview of new functions in the IBM XL C/C++ for Linux, V16.1 compiler, with new functions categorized according to user benefits.
Getting Started with IBM XL C/C++ for Linux, V16.1, GI13-3564-00	`getstart.pdf`	Contains an introduction to XL C/C++, with information about setting up and configuring your environment, compiling and linking programs, and troubleshooting compilation errors.
IBM XL C/C++ for Linux, V16.1 Installation Guide, GC27-8039-00	`install.pdf`	Contains information for installing XL C/C++ and configuring your environment for basic compilation and program execution.
IBM XL C/C++ for Linux, V16.1 Migration Guide, GC27-8042-00	`migrate.pdf`	Contains migration considerations for using XL C/C++ to compile programs that were previously compiled on different platforms, by previous releases of XL C/C++, or by other compilers.
IBM XL C/C++ for Linux, V16.1 Compiler Reference, SC27-8047-00	`compiler.pdf`	Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions.

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

Document title	PDF file name	Description
IBM XL C/C++ for Linux, V16.1 Language Reference, SC27-8045-00	`langref.pdf`	Contains information about language extensions for portability and conformance to nonproprietary standards.
IBM XL C/C++ for Linux, V16.1 Optimization and Programming Guide, SC27-8046-00	`proguide.pdf`	Contains information about advanced programming topics, such as application porting, interlanguage calls with Fortran code, library development, application optimization, and the XL C/C++ high-performance libraries.

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

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

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

Other IBM information

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

Other information

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

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*.
- *Information Technology - Programming languages - C++, ISO/IEC 14882:2014*, also known as *C++14* (Partial support).
- *AltiVec Technology Programming Interface Manual*, Motorola Inc. This specification for vector data types, to support vector processing technology, is available at http://www.freescale.com/files/32bit/doc/ref_manual/ALTIVECPIM.pdf.
- *ANSI/IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985.*

- *OpenMP Application Program Interface Version 3.1* (full support), *OpenMP Application Program Interface Version 4.0 (partial support)*, and *OpenMP Application Program Interface Version 4.5 (partial support)*, available at http://www.openmp.org

Other IBM information

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

Other information

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

Technical support

Additional technical support is available from the XL C/C++ Support page at https://www.ibm.com/support/home/product/Q833644Y89702U61/XL_C/C++_for_Linux. 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@cn.ibm.com.

For the latest information about XL C/C++, visit the product information site at https://www.ibm.com/us-en/marketplace/xl-cpp-linux-compiler-power.

How to send your comments

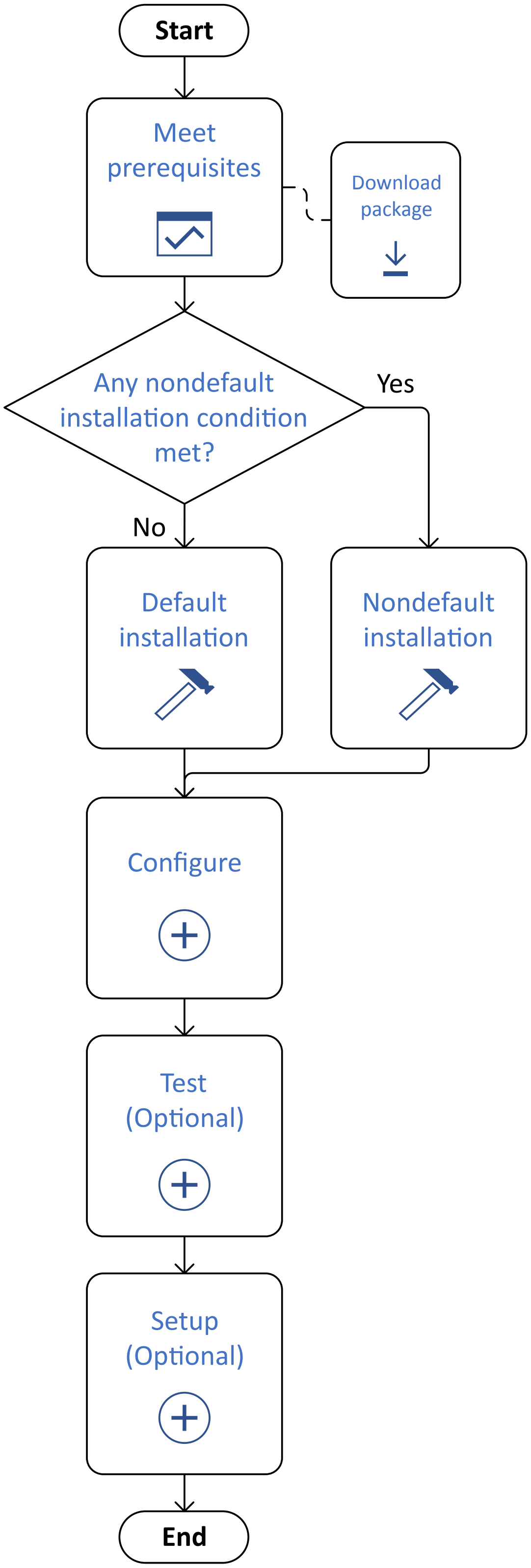
Your feedback is important in helping us to provide accurate and high-quality information. If you have any comments about this information or any other XL C/C++ information, send your comments to compinfo@cn.ibm.com.

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

Chapter 1. Installing the compiler

This section outlines the steps required to install IBM XL C/C++ for Linux, V16.1.

© Copyright IBM Corp. 2003, 2018


Start
Meet prerequisites
Download package
Any nondefault installation condition met?
Yes
No
Default installation
Nondefault installation
Configure
Test (Optional)
Setup (Optional)
End

Prerequisites

Before installing IBM XL C/C++ for Linux, V16.1, you must ensure all the prerequisites are met.

To prepare for installing IBM XL C/C++ for Linux, V16.1, do the following tasks:

- 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.
- Optional: Preview the license agreements.

The installation packages

Download and view the installation image for IBM XL C/C++ for Linux, V16.1 before installation.

Download the electronic distribution package

The package for the licensed version is available for download at the IBM Support website.

The package for the Community Edition is available for download at the IBM Marketplace website.

The installation image

The image includes the following files:

- README files and license agreement files
- An installation tool, `install`, to install and configure the compiler based on distribution.
- A subdirectory that contains files for the little endian compiler, which includes the following files:
  - A README file
  - A set of RPM packages
  - A set of dpkg packages

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

```
rpm -qpil package_name
dpkg -f package_name
```

Table 5. IBM XL C/C++ for Linux, V16.1 packages and default installation locations

Package name	Package description	Default installation location
libxlsmp	IBM SMP runtime package	/opt/ibm/lib/
libxlsmp-devel.5.1.0	IBM SMP library package	/opt/ibm/xlsmp/5.1.0

Table 5. IBM XL C/C++ for Linux, V16.1 packages and default installation locations (continued)

Package name	Package description	Default installation location
libxlmass-devel.9.1.0	IBM Mathematical Acceleration Subsystem (MASS) package (Not redistributable)	/opt/ibm/xlmass/9.1.0/lib/ /opt/ibm/xlmass/9.1.0/include/
libxlc	IBM XL C/C++ runtime package	/opt/ibm/lib/
xlc-license.16.1.0	IBM XL C/C++ license package for IBM XL C/C++ for Linux, V16.1 (Not redistributable)	/opt/ibm/xlC/16.1.0/lib/
libxlc-devel.16.1.0	IBM XL C/C++ libraries package	/opt/ibm/xlC/16.1.0/lib/
xlc.16.1.0	IBM XL C/C++ compiler package (Not redistributable)	/opt/ibm/xlC/16.1.0/

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

National language support

IBM XL C/C++ for Linux, V16.1 messages support the following language locales:

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

en_US is the default locale.

Related information:

Installation workflow diagram

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 requirements for installing IBM XL C/C++ for Linux, V16.1 are listed below:

Supported platforms

You can use any of the following little endian operating systems supported by the IBM Power Systems™ servers:

- Ubuntu Server 16.04
- Ubuntu Server 18.04
- SUSE Linux Enterprise Server 12 (SLES 12)
- SUSE Linux Enterprise Server 12 Service Pack 3 (SLES 12 SP3)

- Red Hat Enterprise Linux 7.3 (RHEL 7.3)
- Red Hat Enterprise Linux 7.4 (RHEL 7.4)
- Red Hat Enterprise Linux 7.4 for Power Little Endian (POWER9)
- Red Hat Enterprise Linux 7.5 (RHEL 7.5)
- Red Hat Enterprise Linux 7.5 for Power Little Endian (POWER9)
- Community Enterprise Operating System 7 (CentOS 7)

Note: To compile programs that contain code used to offload computation to the NVIDIA GPUs, you must use a system that satisfies the installation requirements of the CUDA Toolkit. See the NVIDIA CUDA Toolkit website for more information.

NVIDIA CUDA Toolkit 9.2 with support for IBM Power Little Endian (POWER9) is required for the compilation and linking process for programs that use OpenMP 4.5 to offload computation to the NVIDIA GPUs. CUDA Toolkit 9.2 with support for IBM Power Little Endian (POWER9) is available from NVIDIA.

More operation systems will be supported in later releases or fix packs. For more information, check Fix list for XL C/C++ for Linux.

Hardware requirements

You can use any IBM Power Systems server as long as it is supported by your operating system distribution. For a complete list of the IBM Power Systems servers, see http://www.ibm.com/systems/power/hardware/.

- Approximately 153 MB for product packages
- Minimum of 2 GB hard drive space for paging
- Minimum of 512 MB for temporary files
- 2 GB RAM minimum; 4 GB or more RAM recommended

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

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

Software requirements

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 7.

Table 6. Required minimum versions of GNU and Perl packages for the Ubuntu 16.04 and Ubuntu 18.04 operating systems

Package name	Version requirements
gcc	7.3.0
g++	7.3.0
libc6	2.27
libstdc++6	7.3.0
libgcc1	8
libc6-dev	2.27
libstdc++-dev	7.3.0

Table 6. Required minimum versions of GNU and Perl packages for the Ubuntu 16.04 and Ubuntu 18.04 operating systems (continued)

Package name	Version requirements
perl	5.26.1

Table 7. Required minimum versions of GNU and Perl packages for the SLES 12 and SLES 12 SP3 operating systems

Package name	Version requirements
gcc	4.8
gcc-c++	4.8
glibc	2.19
libgcc_s1	4.8.3
libstdc++6	4.8.3
glibc-devel	2.19
libstdc++-devel	4.8.3
perl	5.18.2

Table 8. Required minimum versions of GNU and Perl packages for the RHEL 7.3, RHEL 7.4, RHEL 7.4 for Power Little Endian (POWER9), RHEL 7.5, RHEL 7.5 for Power Little Endian (POWER9), and CentOS 7 operating systems

Package name	Version requirements
gcc	4.8.3
gcc-c++	4.8.3
glibc	2.17
libgcc	4.8.3
libstdc++	4.8.3
glibc-devel	2.17
libstdc++-devel	4.8.3
perl	5.16.3

To view the online documentation for IBM XL C/C++ for Linux, V16.1, 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)

Related information:

Installation workflow diagram

Verifying the amount of hard disk space available

IBM XL C/C++ for Linux, V16.1 requires about 153 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
```

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

```
df -h installation_path
```

where *installation_path* represents the nondefault location.

Tip: For an overview of the installation process, see Installation workflow diagram.

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 apt-get (on Ubuntu), zypper (on SLES), or yum (on RHEL and CentOS) 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.

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

- For the Ubuntu 16.04 and Ubuntu 18.04 operating systems, see Table 6 on page 5.
- For the SLES 12 and SLES 12 SP3 operating systems, see Table 7 on page 6.
- For the RHEL 7.3, RHEL 7.4, RHEL 7.4 for Power Little Endian (POWER9), RHEL 7.5, RHEL 7.5 for Power Little Endian (POWER9), and CentOS 7 operating systems, see Table 8 on page 6.

All required packages can be obtained from your OS installation media.

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, RHEL, and CentOS

```
rpm -q package_name
```

On Ubuntu

```
dpkg -l package_name
```

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

Tip: For an overview of the installation process, see Installation workflow diagram.

Determining the installed version of gcc:
On SLES, RHEL, and CentOS

To see whether gcc is installed on SLES, RHEL, or CentOS, query for the gcc package as follows:

```
rpm -q gcc
```

If gcc version 4.8-5.3 is installed, you will get a result similar to the following output:

```
gcc-4.8-5.3
```

On Ubuntu

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

```
dpkg -l gcc
```

If gcc version 4.8.2-1ubuntu4 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
```

Related information:

Installation workflow diagram

System prerequisites to offload computations to the NVIDIA GPUs

To compile and link programs that contain code to be offloaded to the NVIDIA GPUs with IBM XL C/C++ for Linux, V16.1, such as using OpenMP directives, you must ensure that your hardware, operating system, and software meet these requirements.

Hardware requirements

You can use any IBM Power Systems server that has one or more NVIDIA GPUs installed and is supported by your Linux operating system distribution and CUDA Toolkit. For example, you can use IBM POWER® System S822LC for high performance computing or IBM POWER System S824L. For a complete list of the IBM Power Systems servers, see Power Servers.

Supported platforms

You must use a system that satisfies the installation requirements of the CUDA Toolkit. See the NVIDIA CUDA Toolkit website for more information.

Software requirements

- NVIDIA CUDA Toolkit 9.2, which you can download from CUDA Toolkit Download

  Note: To install the CUDA Toolkit, use the Package Manager installation. The Runfile installation is currently not supported on Power processors. For instructions about Package Manager installation, see the NVIDIA CUDA Installation Guide for Linux (http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html).

Related information:

Installation workflow diagram

Previewing the license agreements

Before installing IBM XL C/C++ for Linux, V16.1, 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 default installation, the license files can be found in the /opt/ibm/xlC/16.1.0/ directory with the same names as above.

Tip: For an overview of the installation process, see Installation workflow diagram.

Determining the installation procedure

You must choose either the default or nondefault installation procedure to install the compiler.

Default installation

Installs IBM XL C/C++ for Linux, V16.1 to the default directory, /opt/ibm/xlC/16.1.0/.

It is highly recommended that you install IBM XL C/C++ for Linux, V16.1 to the default location by following the procedure provided in "Default installation."

Nondefault installation

Installs IBM XL C/C++ for Linux, V16.1 to a nondefault location.

You must use the nondefault installation procedures provided in "Nondefault installation" on page 12 if any of the following scenarios applies to you:

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

Default installation

The default installation installs the compiler to a default location. You are recommended to use the default installation procedure.

IBM XL C/C++ for Linux provides a utility, `install`, that performs a default installation of IBM XL C/C++ for Linux, V16.1 and its dependencies. You can use `install` to install the RPM packages on SLES, RHEL, or CentOS, or install the dpkg packages on Ubuntu.

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.

Alternatively, you can use the system default package manager to install the packages. For example, you can use the `rpm` utility to install the RPM packages on SLES, RHEL, or CentOS, or you can use the `dpkg` utility to install the dpkg packages on Ubuntu.

If you are installing the compiler to a nondefault location, use the procedures in "Nondefault installation" on page 12.

Running the `install` utility

The `install` utility is the recommended tool for installing XL C/C++. It automatically installs the correct packages and all the required dependencies based on 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:

- 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 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 21.

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

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

For more information on other specialized invocations that might be available, refer to *IBM XL C/C++ for Linux, V16.1 Compiler Reference, "Invoking the compiler"*.

Note: Symbolic links of the following commands are not created in `/usr/bin/`, either because they might delete user-defined or GCC-related invocations, or because they are not compiler invocation commands:

- c89, c89_r, c99, c99_r, cc, cc_r
- cleanpdf, mergepdf, showpdf, xlc_configure

If all packages are successfully installed:

- The install script returns 0 and the following message is displayed confirming the successful installation:
  ```
  Installation and configuration successful
  ```
- The configuration file is generated. Its location is `/opt/ibm/xlC/16.1.0/etc/xlc.cfg.`*`$OS.$OSVersion`*`.gcc.`*`$gccVersion`*. For example, `/opt/ibm/xlC/16.1.0/etc/xlc.cfg.sles.12.gcc.4.8.3` or `/opt/ibm/xlC/16.1.0/etc/xlc.cfg.ubuntu.16.04.gcc.4.8.5`.

Related information:

Installation workflow diagram

Running the alternative utility

To install XL C/C++ using the default installation procedure, the dpkg utility is the alternative utility on Ubuntu and the rpm utility is the alternative utility on SLES, RHEL, or CentOS.

On Ubuntu

dpkg is the software that forms the low-level base of the Debian package management system. It is the default package manager on Ubuntu. You can use dpkg to install, configure, upgrade or remove Debian packages, and retrieve information of these Debian packages.

Note: The installation instructions provided in this topic assume that the product DVD is mounted on the /cdrom directory in the system.

Here are the steps to use the dpkg utility:

1. Install the prerequisite software packages using Advanced Package Tool (apt):

   ```
   apt-get install gcc g++ perl
   ```

   For more information about apt, see the system man page or the online manual at http://manpages.debian.org/cgi-bin/man.cgi?query=apt.
2. Install all compiler packages into the default location, /opt/ibm/:

   ```
   dpkg -iG /cdrom/images/littleEndian/ubuntu/*.deb
   ```

On SLES

The installation instructions provided in this topic assume that the product DVD is mounted on the /cdrom directory in the system.

1. Install the prerequisite software packages:

   ```
   zypper install gcc\>=4.8 gcc-c++\>=4.8 perl\>=5.18.2
   ```
2. Install all compiler packages into the default location, /opt/ibm/:

   ```
   rpm -Uvh /cdrom/images/littleEndian/sles/*.rpm
   ```

On RHEL and CentOS

The installation instructions provided in this topic assume that the product DVD is mounted on the /cdrom directory in the system.

1. Install the prerequisite software packages:

   ```
   yum install perl gcc gcc-c++ glibc libgcc libstdc++ glibc-devel libstdc++-devel
   ```
2. Install all compiler packages into the default location, /opt/ibm/:

   ```
   rpm -Uvh /cdrom/images/littleEndian/rhel/*.rpm
   ```

After you have installed all the compiler packages, follow the steps in "Configuring IBM XL C/C++ for Linux, V16.1" on page 14 to review the license and configure the compiler.

Related information:

Installation workflow diagram

Nondefault installation

The nondefault installation installs the compiler to a nondefault location. If you do not know which product the package belongs to, you can query for the summary of an RPM package.

On SLES, RHEL, or CentOS

In these scenarios, you can use the rpm utility to install the compiler packages.

On Ubuntu

In this scenario, you can use the dpkg and the chroot utilities to install the compiler packages.

Related information:

Installation workflow diagram

Installing XL C/C++ to a nondefault location

You can install all compiler packages to a single nondefault location.

On SLES, RHEL, and CentOS

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, V16.1 and no other RPM packages. From your current working directory, issue the following command:

```
rpm -Uvh *.rpm --prefix installation_path
```

where *installation_path* is a directory that is not /opt/ibm/.

The compiler is installed in the *installation_path* directory.

On Ubuntu

The steps are as follows:

1. Create a chroot directory by issuing the following commands:

   ```
   mkdir </path/to/chroot_dir>
   cd </path/to>
   debootstrap  --arch ppc64el trusty chroot_dir \
   http://ports.ubuntu.com/ubuntu-ports
   ```

   where *</path/to>* is a directory of your choice. *</path/to/chroot_dir>* is assumed to be the root directory of chroot.

   For detailed information about chroot, see the system man page for chroot/debootstrap or the online manual at https://help.ubuntu.com/community/BasicChroot.

2. Install the prerequisites and compiler packages by issuing the following commands:

   ```
   chroot </path/to/chroot_dir> apt-get install gcc g++ perl
   <ctrl-d>
   dpkg --root </path/to/chroot_dir> -iG /cdrom/images/littleEndian/ubuntu/*.deb
   chroot </path/to/chroot_dir>
   sudo /opt/ibm/xlC/16.1.0/bin/xlc_configure
   ```

The compiler is installed in the `/opt/ibm/` directory under `chroot`. To use the compiler under `chroot`, issue the following `chroot` command to enter the root directory of `chroot`:

```
chroot </path/to/chroot_dir>
```

Tip: For an overview of the installation process, see Installation workflow diagram.

Querying for RPM package summaries

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 package summary includes a short description of the queried file.

On SLES, RHEL, and CentOS

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

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

For instance, to query for the summary of the libxlc-devel.16.1.0-16.1.0.0-*$B*.ppc64le.rpm RPM file that has not yet been installed, issue the following command:

```
rpm --qf="%{summary}\n" -qp libxlc-devel.16.1.0-16.1.0.0-$B.ppc64le.rpm
```

The resulting output depends on the specific file queried. A typical example is shown below:

```
IBM XL C/C++ for Linux, V16.1 (5725-C73, 5765-J08) - compiler libraries (SLES)
```

On Ubuntu

To query for the summary of a `dpkg` package that is not installed, issue this 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.16.1.0_16.1.0.0-`*$B*`_ppc64el.deb` file that has not yet been installed, issue the following command:

```
dpkg -f xlc.16.1.0_16.1.0.0-$B_ppc64el.deb
```

The resulting output depends on the specific file queried. A typical example is shown below:

```
Package: xlc.16.1.0
Version: 16.1.0.0-$B
Section: devel
Priority: extra
Architecture: ppc64el
Maintainer: For issues in XL compilers, open a PMR (http://ibm.biz/servicerequest).
For issues in XL Community Edition, visit the XL compiler on Power community
(http://ibm.biz/xl-power-compilers).
Depends: perl (>= 5.18.2), gcc (>= 4.8.2), g++ (>= 4.8.2), libc6 (>= 2.19),
libgcc1 (>= 4.9), libstdc++6 (>= 4.8.2), libc6-dev (>= 2.19), libstdc++-dev,
libxlc (>= 16.1.0), libxlc-devel.16.1.0, libxlmass-devel.9.1.0, xlc_license_for_16.1.0
Provides: xlc
```

```
Description: IBM XL C/C++ for Linux, V16.1.0 (5725-C73, 5765-J08) - compiler
Licensed Materials - Property of IBM.
IBM XL C/C++ for Linux, V16.1.0 (5725-C73, 5765-J08)
Copyright IBM Corp. 1991, 2018.
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 installed on your system.

Related information:

Installation workflow diagram

Configuring IBM XL C/C++ for Linux, V16.1

Configure the compiler based on your conditions.

Before you can run IBM XL C/C++ for Linux, V16.1, 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 on SLES, RHEL, or CentOS.

The compiler provides a configuration tool, `xlc_configure`, located in the *installation_path*`/xlC/16.1.0/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 installed on your system.
- You want the generated configuration file to be placed in a location that is different from *installation_path*`/xlC/16.1.0/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.`*$OS.$OSVersion*`.gcc.`*$gccVersion*, 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 has been accepted.

To run the `xlc_configure` utility to configure an installation that is not done with the `install` utility, become a superuser with `su`, or `sudo`, and run the following command:

```
installation_path/xlC/16.1.0/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/.

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.

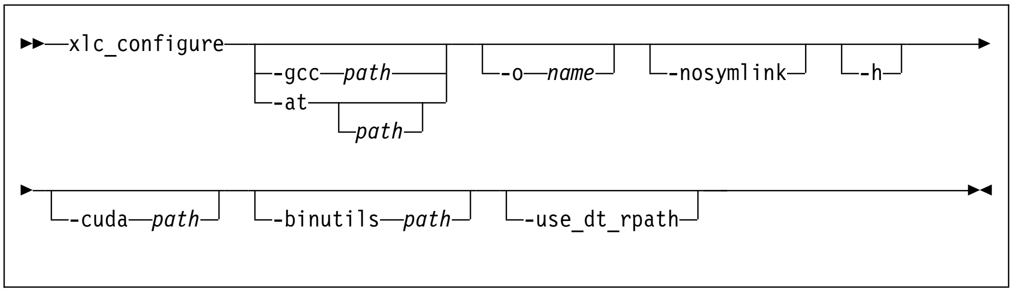
Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

xlc_configure options

The xlc_configure command has the following syntax:



where:

-gcc *path*

Specifies the path where the 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.

-at *path*

Configures the compiler for usage with the Advance Toolchain, and creates compiler invocations for xl*_at. *path* is the install location of the Advance Toolchain.

If *path* is not specified, the first path found in the following ordered list is used:

1. /opt/at11.0
2. /opt/at9.0
3. /opt/at8.0

-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/16.1.0/etc/xlc.cfg.ubuntu.16.04.gcc.4.8.5.

-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

-h Displays the help page for the `xlc_configure` options.

-cuda *path*

Specifies the path to the CUDA Toolkit. By default, *path* is `/usr/local/cuda` if it exists.

To disable the automatic detection of the CUDA Toolkit, specify the following argument:

```
-cuda null
```

-binutils *path*

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

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

-use_dt_rpath

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

- When this option is in effect, the compiler uses the DT_RPATH property. As a result, the **LD_LIBRARY_PATH** environment variable does not have no effect on your program.
- When this option is not in effect, --enable-new-dtags is passed to the linker to set DT_RUNPATH instead of DT_RPATH. The load library path search rules are modified and you can use the **LD_LIBRARY_PATH** environment variable to override the DT_RUNPATH property.

By default, this option is not enabled.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Testing the installation

After you install the compiler, you can optionally query for installed packages and test a sample application.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

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, RHEL, and CentOS

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

```
rpm -q xlc.16.1.0
```

The result is:

```
xlc.16.1.0-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:

```
rpm -qa | grep -e xlc.16.1.0 -e libxlmass-devel.9.1.0
```

The result is a list containing all of the packages as shown in Table 5 on page 3. 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.16.1.0
```

The result is:

```
Package xlc.16.1.0

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.16.1.0 -e libxlmass-devel.9.1.0
```

The result is a list containing all of the packages as shown in Table 5 on page 3. If none of the packages was properly installed, there will be no output from the command.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Testing a sample application

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

About this task

Take the following steps to build and run a “Hello World” 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/16.1.0/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 you have set up the short invocation commands, enter the following command:

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

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

   ```
    installation_path/xlC/16.1.0/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”.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Setting up IBM XL C/C++ for Linux, V16.1

After you install the compiler, you can optionally setup the compiler, such as enabling the manual pages and error messages, setting up utilization tracking, and enabling IBM ILMT and TADd.

Enabling and viewing the manual pages

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

Enabling the manual pages

The IBM XL C/C++ for Linux, V16.1 manual pages support the following locales:

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

However, before you can read the compiler-supplied man 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/16.1.0/man/LANG:$MANPATH
  ```
- To set the *MANPATH* environment variable using the C shell, use the following command:

  ```
  setenv MANPATH installation_path/xlC/16.1.0/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.

Viewing the manual pages

You can view the manual pages after they are enabled in the compiler. To invoke a manual page, enter the following command:

```
man command
```

Example:

```
man xlc
```

Tip: For an overview of the installation process, see Installation workflow diagram.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Enabling the error messages

You must set the *NLSPATH* environment variable so that the runtime functions can find the appropriate message catalogs after the installation. Otherwise, incomplete error messages might be issued. The compiler message catalogs are automatically configured to display correctly, regardless of whether you used the default or nondefault method of installation and configuration.

The command to set the *NLSPATH* 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 NLSPATH=$NLSPATH:xlrte_path/msg/%L/%N
  ```
- If you are using the C shell, use the following command:
  ```
  setenv NLSPATH $NLSPATH:xlrte_path/msg/%L/%N
  ```

where *xlrte_path* is the installation location of the IBM XL C/C++ for Linux, V16.1 runtime 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.

Tip: For an overview of the installation process, see Installation workflow diagram.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Setting up the environment for the invocation commands

If you used the `install` utility to install the compiler or if you selected to create the symbolic links during the configuration, 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" on page 21.
- Create symbolic links to the compiler invocation commands, as shown in "Creating symbolic links to the compiler invocations" on page 21.

Tips:

- For an overview of the installation process, see Installation workflow diagram.
- For an overview of the updating to the latest fix pack process, see Updating to the latest fix pack workflow diagram.

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

To use IBM XL C/C++ for Linux, V16.1 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/xlC/16.1.0/bin/
```

- If you are using the C shell, use the following command:

```
setenv PATH $PATH:installation_path/xlC/16.1.0/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.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

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/16.1.0/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++
- xlC
- xlc_r
- xlc++_r
- xlC_r

Links to the following invocations are not recommended, either because they delete user-defined or GCC invocations, or because they are not compiler invocation 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 $exec installation_path/xlC/16.1.0/bin/$exec 1;
done
```

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

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Setting up utilization tracking

You can use utilization tracking to detect whether the compiler usage exceeds your entitlement based on the number of concurrent user and authorized user licenses you have purchased.

IBM XL C/C++ for Linux supports IBM Software License Metric (SLM) Tags logging, which tracks compiler usage through IBM License Metric Tool (ILMT). The compiler logs compiler license usage in the SLM Tags format, to a location you can define by specifying the **slm_dir** attribute of the configuration file. The default location is `/var/opt/ibm/xl-compiler/` for a default installation, or *`$prefix`*`/var/opt/ibm/xl-compiler/` for a nondefault installation, where *`$prefix`* is the nondefault installation path. You must set up SLM Tags logging if you want to use it to track utilization. For more information, see Setting up SLM Tags logging in the *XL C/C++ Compiler Reference*.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Enabling IBM License Metric Tool (ILMT) and Tivoli Asset Discovery for Distributed (TADd)

IBM License Metric Tool (ILMT) and Tivoli® Asset Discovery for Distributed (TADd) can help you manage software usage metering and license allocation services on supported systems. In general, ILMT and TADd recognize and monitor the products that are installed and in use on your system.

Note: ILMT and TADd are not part of the IBM XL C/C++ for Linux offering, and must be ordered and installed separately.

Once installed and activated, ILMT and TADd scan your system for product inventory signatures that indicate whether a given product is installed on your system. ILMT and TADd also identify the version, release, and modification levels of the product. Inventory signature files are not updated after a fix pack is installed.

ILMT and TADd, after they are deployed on a computer, collect information about the level and duration of IBM XL C/C++ for Linux compiler use on that computer and can generate reports based on the information it collects.

If IBM XL C/C++ for Linux is installed in the default location, the signature files are in the `/opt/ibm/xlC/16.1.0/swidtag/` directory. If IBM XL C/C++ for Linux is installed in a nondefault location, the signature files are in the *`installation_path`*`/xlC/16.1.0/swidtag/` directory, where *installation_path* is the target directory for installation specified by --prefix option of the nondefault installation command on SLES, RHEL, and CentOS, or the `/opt/ibm/` directory under the root directory of `chroot` on Ubuntu.

For more information, see IBM License Metric Tool and Tivoli Asset Discovery for Distributed.

Related information:

Installation workflow diagram

Updating to latest fix pack workflow diagram

Chapter 2. Upgrading to the latest release

If you are not using the latest release of the compiler, you can upgrade the compiler to the latest release.

You can find the latest release of IBM XL C/C++ for Linux from Fix list for IBM XL C/C++ for Linux.

To upgrade the XL C/C++ compiler to the latest release, use the `install` utility as described in the latest release of Chapter 1, “Installing the compiler,” on page 1.

© Copyright IBM Corp. 2003, 2018

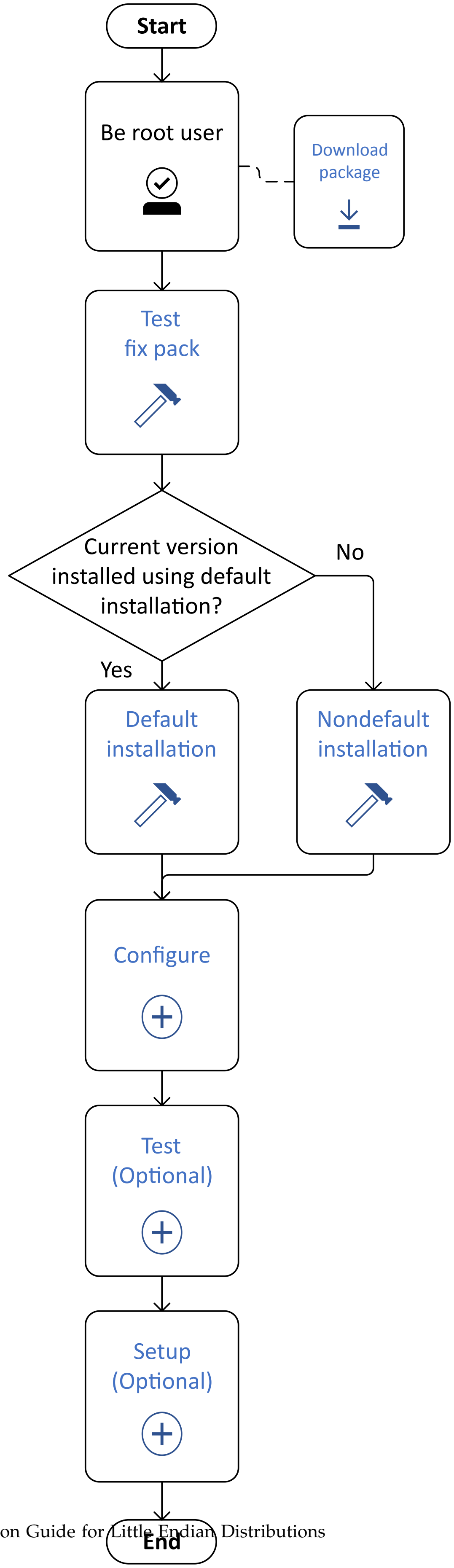
Chapter 3. Updating to the latest fix pack

A fix pack of IBM XL C/C++ for Linux provides a fix or multiple fixes to the product.

You can download updates from Fix list for IBM XL C/C++ for Linux.

The following diagram shows the procedure to apply an update to your compiler.

© Copyright IBM Corp. 2003, 2018


Start
Be root user
Download package
Test fix pack
Current version installed using default installation?
No
Yes
Default installation
Nondefault installation
Configure
Test (Optional)
Setup (Optional)
End

Every fix pack comes in the 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 of IBM XL C/C++ for Linux installed on your system, an earlier update included, you can apply the latest update.

Testing a fix pack 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 fix pack to a nondefault location, use the procedure described in "Installing XL C/C++ to a nondefault location" on page 12.
To configure the compiler, use the procedure described in "Configuring IBM XL C/C++ for Linux, V16.1" on page 14.

Updating a default installation

Updates to the compiler are supplied as fix packs. You can follow the instructions in this section to download, decompress, and install the fix packs.

Procedure

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

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

   where *package_name* is the name of the fix pack that you downloaded.
3. Install the fix pack:
   - Run the `install` utility:

     ```
     ./install
     ```
   - Alternatively, you can install the fix pack manually:
     - **SLES:**

       ```
       rpm -Uvh images/littleEndian/sles/*.rpm
       ```
     - **RHEL and CentOS:**

       ```
       rpm -Uvh images/littleEndian/rhel/*.rpm
       ```
     - **Ubuntu:**

       ```
       dpkg -iG images/littleEndian/ubuntu/*.deb
       ```

Updating multiple versions of fix packs installed under separate locations

On SLES, RHEL, and CentOS

To update multiple versions of fix pack, run the `rpm` utility to install the fix packs with the **-U** option. If a previous version of compiler or fix packs are installed under prefix *$CMPpath*, use the **--prefix** *$CMPpath* option.

On Ubuntu

To update multiple versions of fix pack, Run the dpkg utility to install the fix packs under chroot. For instructions, see "Installing XL C/C++ to a nondefault location" on page 12.

Chapter 4. Updating from Community Edition to full version

This section outlines the steps required to update IBM XL C/C++ for Linux, V16.1 from Community Edition to full version.

On SLES, RHEL, and CentOS

If you are using the Community Edition of the XL C/C++ compiler on SLES, RHEL, or CentOS, 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 "Default installation" on page 9.
- Alternatively, upgrade your Community Edition as follows:
  1. Install the license packages using the following command:
     ```
     rpm -Uvh xlc-license.16.1.0-*.ppc64le.rpm
     ```
  2. Run the `xlc_configure` utility to accept the license if it is not accepted, and reconfigure the compiler.

On Ubuntu

If you are using the Community Edition 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 "Default installation" on page 9.
- Alternatively, upgrade your Community Edition as follows:
  1. Install the license packages using the following command:
     ```
     dpkg -iG xlc-license.16.1.0_*_ppc64el.deb
     ```
  2. Run the `xlc_configure` utility to accept the license if it is not accepted, and reconfigure the compiler.

© Copyright IBM Corp. 2003, 2018

Chapter 5. Uninstalling the compiler

You must use the Linux `rpm` or `dpkg` utility to uninstall IBM XL C/C++ for Linux, V16.1.

IBM XL C/C++ for Linux, V16.1 does not provide a stand-alone uninstallation tool.

Notes:

- You must have root access to uninstall the compiler.
- Whenever you uninstall a package, specify the package name. For information about how to determine the package name, see "Querying for installed packages" on page 16.
- It is recommended to uninstall all packages in a single command. If you prefer to uninstall using multiple commands, 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.
- You cannot uninstall packages that are required by other packages. For example, `libxlmass-devel.9.1.0` is a shared component if IBM XL Fortran for Linux, V16.1 is also installed on the same system.
- 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, RHEL, and CentOS, 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, V16.1

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

In this example:

- The compiler packages have a *V.R.M* of 16.1.0.
- The IBM MASS library package has a *V.R.M* of 9.1.0.

On SLES, RHEL, and CentOS

To uninstall IBM XL C/C++ for Linux, V16.1, take the following steps:

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

   ```
   rpm -e xlc.16.1.0 libxlc-devel.16.1.0 \
   xlc-license.16.1.0

   xlc-license-community.16.1.0
   ```

   Note: To uninstall XL Fortran at the same time when uninstalling XL C/C++, uninstall all XL Fortran specific packages before proceeding. For a complete list of XL Fortran specific packages and the order in which they must be uninstalled, see "Example: Uninstalling IBM XL Fortran for Linux, V16.1" in the *XL Fortran Installation Guide*.

2. Optional: If you have XL Fortran installed on your system and want it to continue to function normally, do *not* uninstall the following packages. Otherwise, uninstall the following packages:

   ```
   rpm -e libxlmass-devel.9.1.0  libxlsmp-devel.5.1.0
   ```

3. 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  libxlsmp
   ```

On Ubuntu

To uninstall IBM XL C/C++ for Linux, V16.1 and remove the configuration files, take the following steps:

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

   ```
   dpkg -P xlc.16.1.0 libxlc-devel.16.1.0 \
   xlc-license.16.1.0
   ```

   Note: To uninstall XL Fortran at the same time when uninstalling XL C/C++, uninstall all XL Fortran specific packages before proceeding. For a complete list of XL Fortran specific packages and the order in which they must be uninstalled, see "Example: Uninstalling IBM XL Fortran for Linux, V16.1" in the *XL Fortran Installation Guide*.

2. Optional: If you have XL Fortran installed on your system and want it to continue to function normally, do *not* uninstall the following package. Otherwise, uninstall the following package:

   ```
   dpkg -P libxlmass-devel.9.1.0  libxlsmp-devel.5.1.0
   ```

3. 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:

   ```
   dpkg -P libxlc  libxlsmp
   ```

Note: If you want to keep the configuration files while uninstalling the compiler, use the **-r** option instead of the **-P** option in the above commands.

Notices

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

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

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

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

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

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

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

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

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

© Copyright IBM Corp. 2003, 2018

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. 1998, 2018.

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 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.

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

NVIDIA and CUDA are either registered trademarks or trademarks of NVIDIA Corporation in the United States, other countries, or both.

Index

© Copyright IBM Corp. 2003, 2018

Product Number: 5765-J08; 5725-C73

Printed in USA

GC27-8039-00