

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



Installation Guide

Version 13.1.6

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



Installation Guide

Version 13.1.6

Note

Before using this information and the product it supports, read the information in "Notices" on page 67.

First edition

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

© Copyright IBM Corporation 2003, 2017.

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

Contents

About this document v

| | |
|--|-----|
| Who should read this document. | v |
| How to use this document. | v |
| How this document is organized | v |
| Conventions | vi |
| Related information | x |
| Available help information. | x |
| Standards and specifications | xi |
| Other IBM information | xii |
| Other information | xii |
| Technical support | xii |
| How to send your comments | xii |

Chapter 1. Before installing IBM XL C/C++ for Linux 1

Chapter 2. Installing IBM XL C/C++ for Linux, V13.1.6 for little endian distributions 3

| | |
|--|----|
| Installing the compiler | 4 |
| Prerequisites | 5 |
| Determining the installation procedure | 11 |
| Default installation | 11 |
| Nondefault installation | 14 |
| Configuring IBM XL C/C++ for Linux, V13.1.6 | 16 |
| Testing the installation. | 18 |
| Setting up IBM XL C/C++ for Linux, V13.1.6 | 20 |
| Upgrading to the latest release | 25 |
| Updating to the latest fix pack | 25 |
| Testing a fix pack before you install it | 27 |
| Updating a default installation | 27 |
| Updating multiple versions of fix packs installed under separate locations | 27 |
| Updating from Community Edition to full version | 28 |
| Uninstalling the compiler. | 28 |

Chapter 3. Installing IBM XL C/C++ for Linux, V13.1 for big endian distributions 31

| | |
|--|----|
| Preparing for installing IBM XL C/C++ for Linux, V13.1 | 31 |
| The installation packages | 32 |
| System prerequisites | 33 |
| Previewing the license agreements. | 36 |
| Basic installation. | 37 |

| | |
|--|----|
| Tasks for basic installation | 37 |
| Running the <code>install</code> utility for a new installation | 38 |
| Running the <code>xlc_install</code> utility for a new installation | 40 |
| Advanced installation | 42 |
| Tasks for advanced installation | 43 |
| Installing XL C/C++ to co-reside with earlier versions | 44 |
| Running the IBM License Acceptance Process (LAP) tool | 48 |
| Querying for RPM package summaries | 49 |
| Installing an update | 49 |
| Tasks for update installation. | 50 |
| Testing a PTF before you install it | 50 |
| Updating a basic installation using the <code>xlc_install</code> utility | 51 |
| Updating an advanced installation. | 52 |
| Configuring IBM XL C/C++ for Linux, V13.1 (for advanced users) | 53 |
| Running the <code>new_install</code> utility | 53 |
| Running the <code>xlc_configure</code> utility directly | 54 |
| After installing IBM XL C/C++ for Linux, V13.1 | 57 |
| Testing the installation. | 57 |
| Enabling utilization tracking. | 58 |
| Enabling the man pages | 58 |
| Enabling the error messages. | 59 |
| Setting up the environment for the invocation commands. | 60 |
| Enabling IBM License Metric Tool (ILMT) and Tivoli Asset Discovery for Distributed (TADd). | 61 |
| Accessing the local documentation | 62 |
| Querying for installed packages | 62 |
| Uninstalling IBM XL C/C++ for Linux, V13.1 | 63 |
| Example: Uninstalling IBM XL C/C++ for Linux, V13.1 | 63 |
| Troubleshooting the installation and configuration | 64 |
| The specified directory <code>rpmlocation_path</code> does not exist. | 64 |
| <code>rpmlocation_path</code> does not contain all of the RPM packages for the XL compiler | 64 |

Notices 67

| | |
|----------------------|----|
| Trademarks | 69 |
|----------------------|----|

Index 71

About this document

This guide provides detailed installation instructions for IBM® XL C/C++ for Linux, V13.1.6 for both big endian and little endian distributions. For each compiler, 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, V13.1.6 for either big endian or little endian distribution.

How to use this document

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

For instructions for the little endian compiler, see Chapter 2, “Installing IBM XL C/C++ for Linux, V13.1.6 for little endian distributions,” on page 3.

For instructions for the big endian compiler, see Chapter 3, “Installing IBM XL C/C++ for Linux, V13.1 for big endian distributions,” on page 31.

How this document is organized

This book contains information for installing IBM XL C/C++ for Linux for Linux for both little endian and big endian distributions.

Table 1. Book structure

| Chapters | User segment |
|--|---------------------------|
| Chapter 1, “Before installing IBM XL C/C++ for Linux,” on page 1 | All users |
| Chapter 2, “Installing IBM XL C/C++ for Linux, V13.1.6 for little endian distributions,” on page 3 | Linux little endian users |
| Chapter 3, “Installing IBM XL C/C++ for Linux, V13.1 for big endian distributions,” on page 31 | Linux big endian users |

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

Table 2. Tasks of IBM XL C/C++ for Linux (for little endian)

| Task | Chapters |
|--------------|-------------------------------------|
| Installation | “Installing the compiler” on page 4 |

Table 2. Tasks of IBM XL C/C++ for Linux (for little endian) (continued)

| Task | Chapters |
|--|--|
| Upgrading to the latest release | “Upgrading to the latest release” on page 25 |
| Upgrading to the latest fix pack | “Updating to the latest fix pack” on page 25 |
| Upgrading from Community Edition to full version | “Updating from Community Edition to full version” on page 28 |
| Uninstallation | “Uninstalling the compiler” on page 28 |

For compiler for big endian distributions, information is organized to reflect the pre-installation, installation, post-installation, and product removal phases of an IBM XL C/C++ for Linux installation.

Table 3. Phases of an IBM XL C/C++ for Linux installation (for big endian)

| Phase | Chapters | User segment |
|-------------------|---|--|
| Pre-installation | “Preparing for installing IBM XL C/C++ for Linux, V13.1” on page 31 | All Linux big endian users |
| Installation | “Basic installation” on page 37 | <ul style="list-style-type: none"> Linux big endian users who want to use the simplest, most direct installation process Linux big endian users who do not have any special requirements, such as the use of multiple versions of the compiler |
| | “Advanced installation” on page 42 | <ul style="list-style-type: none"> Linux big endian users who want to install the compiler in a nondefault location Linux big endian users who want to have multiple versions of the compiler on the same system |
| Post-installation | “Installing an update” on page 49 | Linux big endian users who want to update IBM XL C/C++ for Linux, V13.1.6 to the next fix level |
| | “Configuring IBM XL C/C++ for Linux, V13.1 (for advanced users)” on page 53 | <ul style="list-style-type: none"> Linux big endian users who installed the compiler in a nondefault location Linux big endian users who have multiple versions of the compiler installed on the same system |
| | “After installing IBM XL C/C++ for Linux, V13.1” on page 57 | All Linux big endian users |
| Product removal | “Uninstalling IBM XL C/C++ for Linux, V13.1” on page 63 | Any Linux big endian user who needs to remove IBM XL C/C++ for Linux from the system |

Conventions

Typographical conventions

The following table shows the typographical conventions used in the IBM XL C/C++ for Linux, V13.1.6 information.

Table 4. *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 <u>maf</u></code> |
| 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 5. *Qualifying elements*





















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

Table 5. Qualifying elements (continued)

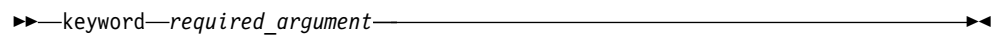
| Qualifier/Icon | Meaning |
|--|--|
| C++11 begins   C++11 ends | The text describes a feature that is introduced into standard C++ as part of C++11. |
| C++14 begins   C++14 ends | The text describes a feature that is introduced into standard C++ as part of C++14. |
| IBM extension begins   IBM extension ends | The text describes a feature that is an IBM extension to the standard language specifications. |
| GPU begins   GPU ends | The text describes the information that is relevant to offloading computations to the NVIDIA GPUs. |

Syntax diagrams

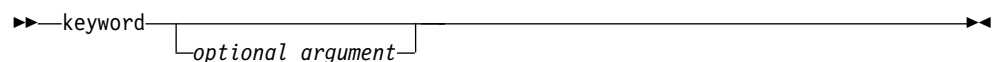
Throughout this information, diagrams illustrate XL C/C++ syntax. This section helps you to interpret and use those diagrams.

- 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):



- Optional items are shown below the main path:

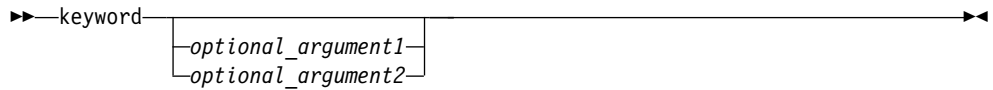


- If you can choose from two or more items, they are shown vertically, in a stack.

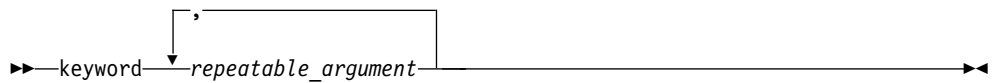
If you *must* choose one of the items, one item of the stack is shown on the main path.



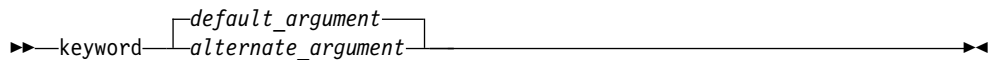
If choosing one of the items is optional, the entire stack is shown below the main path.



- 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 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, V13.1.6. It is located by default in the XL C/C++ directory and in the `\quickstart` directory of the installation DVD.

- README files

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

- Installable man pages

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

- Online product documentation

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

- PDF documents

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

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

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

Table 6. XL C/C++ PDF files

| Document title | PDF file name | Description |
|---|----------------------------|---|
| <i>What's New for IBM XL C/C++ for Linux, V13.1.6, GC27-8035-00</i> | <code>whats_new.pdf</code> | Provides an executive overview of new functions in the IBM XL C/C++ for Linux, V13.1.6 compiler, with new functions categorized according to user benefits. |

Table 6. XL C/C++ PDF files (continued)

| Document title | PDF file name | Description |
|---|---------------|---|
| <i>Getting Started with IBM XL C/C++ for Linux, V13.1.6, GI13-2875-05</i> | 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. |
| <i>IBM XL C/C++ for Linux, V13.1.6 Installation Guide, GC27-6540-05</i> | install.pdf | Contains information for installing XL C/C++ and configuring your environment for basic compilation and program execution. |
| <i>IBM XL C/C++ for Linux, V13.1.6 Migration Guide, GC27-8036-00</i> | migrate.pdf | Contains migration considerations for using XL C/C++ to compile programs that were previously compiled on different platforms, by previous releases of XL C/C++, or by other compilers. |
| <i>IBM XL C/C++ for Linux, V13.1.6 Compiler Reference, SC27-6570-05</i> | compiler.pdf | Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions. |
| <i>IBM XL C/C++ for Linux, V13.1.6 Language Reference, SC27-6550-05</i> | langref.pdf | Contains information about language extensions for portability and conformance to nonproprietary standards. |
| <i>IBM XL C/C++ for Linux, V13.1.6 Optimization and Programming Guide, SC27-6560-05</i> | proguide.pdf | Contains information about advanced programming topics, such as application porting, interlanguage calls with Fortran code, library development, application optimization, and the XL C/C++ high-performance libraries. |

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

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

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

Other IBM information

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

Other information

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

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. Before installing IBM XL C/C++ for Linux

IBM XL C/C++ for Linux, V13.1.6 supports both big endian and little endian Linux.

To determine the endianness of your distribution, specify the following command:

```
uname -i
```

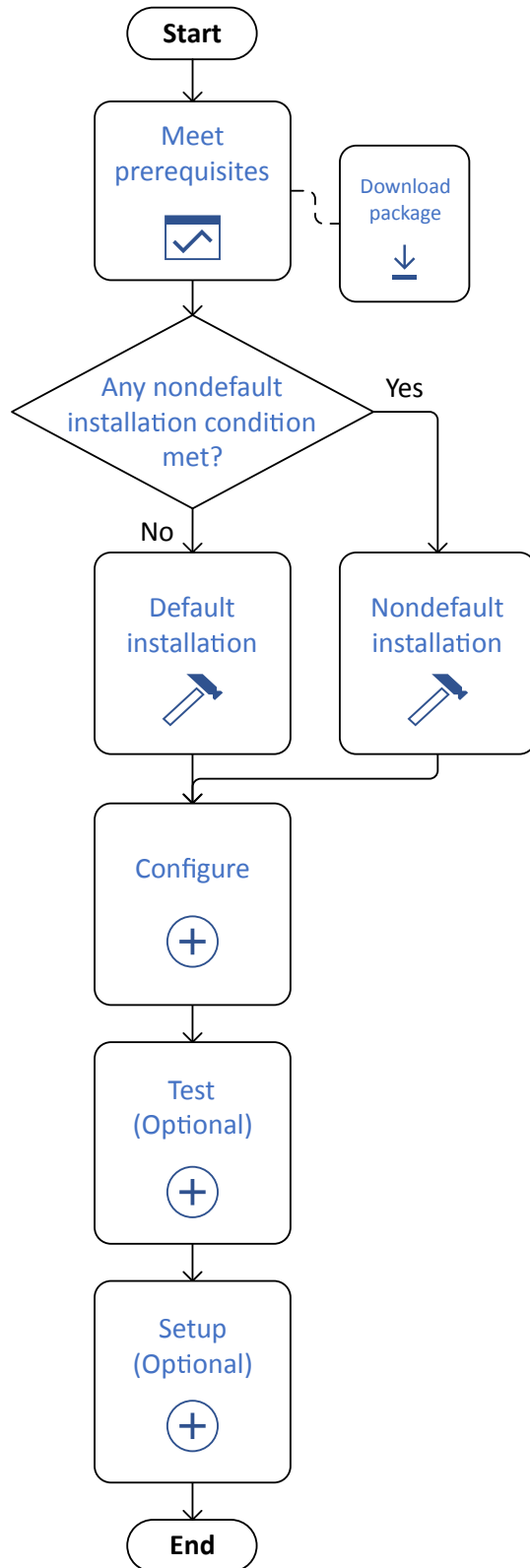
If the result is `ppc64le`, your distribution is little endian. For installation instructions, see Chapter 2, “Installing IBM XL C/C++ for Linux, V13.1.6 for little endian distributions,” on page 3.

If the result is `ppc64`, your distribution is big endian. For installation instructions, see Chapter 3, “Installing IBM XL C/C++ for Linux, V13.1 for big endian distributions,” on page 31.

Chapter 2. Installing IBM XL C/C++ for Linux, V13.1.6 for little endian distributions

Installing the compiler

This section outlines the steps required to install IBM XL C/C++ for Linux, V13.1.6.



Prerequisites

Before installing IBM XL C/C++ for Linux, V13.1.6, you must ensure all the prerequisites are met.

To prepare for installing IBM XL C/C++ for Linux, V13.1.6, do the following tasks:

- Consult the product README files for any last-minute updates, which is included in the installation image.
- 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, V13.1.6 before installation.

Download the electronic distribution package

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

The package for the community edition is available for download at: IBM developerWorks®.

The installation image

The image includes:

- README files and license agreement files
- An installation tool, `install`, to install and configure the compiler based on endianness and distribution.
- A subdirectory that contains files for the big endian compiler, which includes the following files:
 - README files
 - A set of RPM packages
 - An installation tool, `xlc_install`, to install and configure the compiler for a default installation
 - Utilization reporting tool package
- 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 7 on page 6 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 -qip package_name
dpkg -f package_name
```

Table 7. IBM XL C/C++ for Linux, V13.1.6 packages and default installation locations

| Package name | Package description | Default installation location |
|-----------------------|--|--|
| libxlsmp | IBM SMP runtime package | /opt/ibm/lib/ |
| libxlsmp-devel.4.1.6 | IBM SMP library package | /opt/ibm/xlsmp/4.1.6 |
| libxlmass-devel.8.1.6 | IBM Mathematical Acceleration Subsystem (MASS) package (Not redistributable) | /opt/ibm/xlmass/8.1.6/lib/ /opt/ibm/xlmass/8.1.6/include/ |
| libxlc | IBM XL C/C++ runtime package | /opt/ibm/lib/ |
| xlc-license.13.1.6 | IBM XL C/C++ license package for IBM XL C/C++ for Linux, V13.1.6 (Not redistributable) | /opt/ibm/xlc/13.1.6/lib/ |
| libxlc-devel.13.1.6 | IBM XL C/C++ libraries package | /opt/ibm/xlc/13.1.6/lib/ |
| xlc.13.1.6 | IBM XL C/C++ compiler package (Not redistributable) | /opt/ibm/xlc/13.1.6/ |

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

National language support

IBM XL C/C++ for Linux, V13.1.6 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, V13.1.6 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 14.04

- Ubuntu Server 14.10
- Ubuntu Server 16.04
- 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)
- Community Enterprise Operating System 7 (CentOS 7)

Note: To compile programs that contain code to be offloaded to the NVIDIA GPUs, you must use either of the following operating systems:

- Ubuntu Server 16.04.3
- Red Hat Enterprise Linux 7.3 (RHEL 7.3) or above

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 8.

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 9.

Table 8. Required minimum versions of GNU and Perl packages for the Ubuntu 14.04, Ubuntu 14.10, and Ubuntu 16.04 operating systems

| 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 9. 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 10. Required minimum versions of GNU and Perl packages for the RHEL 7.3, RHEL 7.4, 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, V13.1.6, 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, V13.1.6 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 14.04, Ubuntu 14.10, and Ubuntu 16.04 operating systems, see Table 8 on page 7.
- For the SLES 12 and SLES 12 SP3 operating systems, see Table 9 on page 8.
- For the RHEL 7.3, RHEL 7.4, and CentOS 7 operating systems, see Table 10 on page 8.

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, V13.1.6, 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 can use the following little endian operating systems supported by the IBM Power Systems servers and NVIDIA CUDA Toolkit 9.0 or 9.1:

- Red Hat Enterprise Linux 7.3 (RHEL 7.3)
- Ubuntu 16.04.1

Software requirements

- NVIDIA CUDA Toolkit 9.0 or 9.1, 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 \(http://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html\)](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, V13.1.6, 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/13.1.6/` 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, V13.1.6 to the default directory, `/opt/ibm/xlC/13.1.6/`.

It is highly recommended that you install IBM XL C/C++ for Linux, V13.1.6 to the default location by following the procedure provided in “Default installation.”

Nondefault installation

Installs IBM XL C/C++ for Linux, V13.1.6 to a nondefault location.

You must use the nondefault installation procedures provided in “Nondefault installation” on page 14 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, V13.1.6 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 14.

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 endianness and 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 architecture (big endian or little endian)
- 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 22.

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, V13.1.6 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/13.1.6/etc/xlc.cfg.$OS.$OSVersion.gcc.$gccVersion`. For example, `/opt/ibm/xlC/13.1.6/etc/xlc.cfg.sles.12.gcc.4.8.3` or `/opt/ibm/xlC/13.1.6/etc/xlc.cfg.ubuntu.14.04.gcc.4.8.2`.

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, V13.1.6” on page 16 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, V13.1.6 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/13.1.6/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.13.1.6-13.1.6.0-$B.ppc64le.rpm` RPM file that has not yet been installed, issue the following command:

```
rpm --qf="%{summary}\n" -qp libxlc-devel.13.1.6-13.1.6.0-$B.ppc64le.rpm
```

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

```
IBM XL C/C++ for Linux, V13.1.6 (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.13.1.6_13.1.6.0-$B_ppc64el.deb` file that has not yet been installed, issue the following command:

```
dpkg -f xlc.13.1.6_13.1.6.0-$B_ppc64el.deb
```

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

```
Package: xlc.13.1.6
Version: 13.1.6.0-$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 (>= 13.1.6), libxlc-devel.13.1.6, libxlmass-devel.8.1.6, xlc_license_for_13.1.6
Provides: xlc
Description: IBM XL C/C++ for Linux, V13.1.6 (5725-C73, 5765-J08) - compiler
```

Licensed Materials - Property of IBM.
IBM XL C/C++ for Linux, V13.1.6 (5725-C73, 5765-J08)
Copyright IBM Corp. 1991, 2017.
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, V13.1.6

Configure the compiler based on your conditions.

Before you can run IBM XL C/C++ for Linux, V13.1.6, 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/13.1.6/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/13.1.6/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/13.1.6/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.

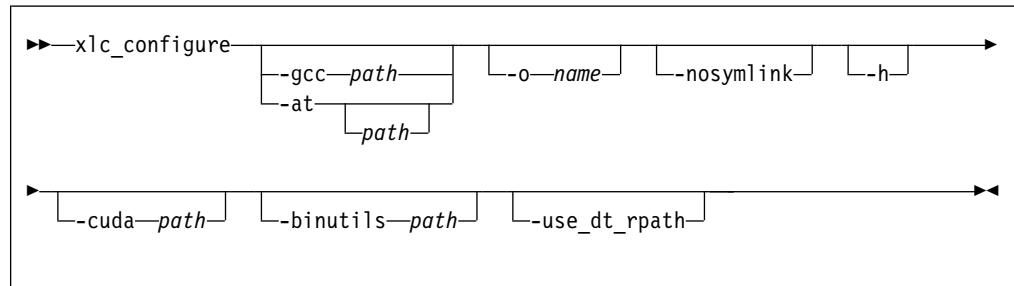
Related information:

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 x1*_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/13.1.6/etc/xlc.cfg.ubuntu.14.04.gcc.4.8.2.

-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.13.1.6
```

The result is:

```
xlc.13.1.6-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.13.1.6 -e libxlmass-devel.8.1.6
```

The result is a list containing all of the packages as shown in Table 7 on page 6. 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.13.1.6
```

The result is:

```
Package xlc.13.1.6
```

```
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.13.1.6 -e libxlmass-devel.8.1.6
```

The result is a list containing all of the packages as shown in Table 7 on page 6. 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/13.1.6/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/13.1.6/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, V13.1.6

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, V13.1.6 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/x1c/13.1.6/man/LANG:$MANPATH
```

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

```
setenv MANPATH installation_path/x1c/13.1.6/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 x1c
```

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

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

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, V13.1.6 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.”
- Create symbolic links to the compiler invocation commands, as shown in “Creating symbolic links to the compiler invocations” on page 23.

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, V13.1.6 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/13.1.6/bin/`

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

```
setenv PATH $PATH:installation_path/x1C/13.1.6/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*/x1C/13.1.6/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++
- x1C
- xlc_r
- xlc++_r
- x1C_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++ x1C xlc_r xlc++_r x1C_r;
do update-alternatives \
--install /usr/bin/$exec $exec installation_path/x1C/13.1.6/bin/$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/13.1.6/swidtag/` directory. If IBM XL C/C++ for Linux is installed in a nondefault location, the signature files are in the `installation_path/xlC/13.1.6/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

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.

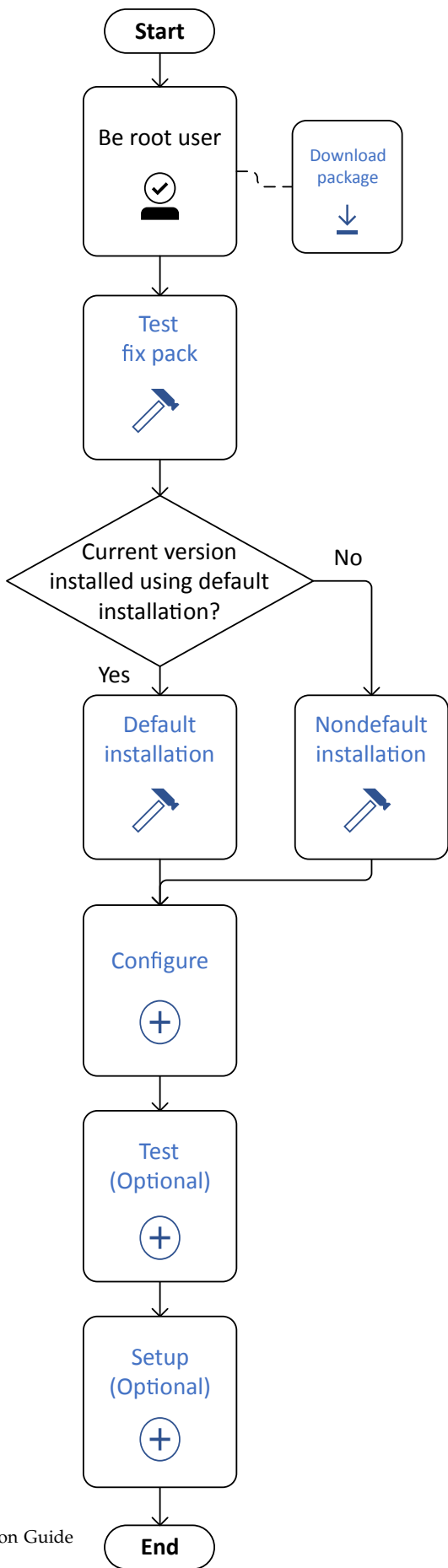
To upgrade the XL C/C++ compiler to the latest release, use the `install` utility as described in the latest release of “Installing the compiler” on page 4.

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.



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 14.

To configure the compiler, use the procedure described in “Configuring IBM XL C/C++ for Linux, V13.1.6” on page 16.

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 14.

Updating from Community Edition to full version

This section outlines the steps required to update IBM XL C/C++ for Linux, V13.1.6 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 11.
- Alternatively, upgrade your Community Edition as follows:
 1. Install the license packages using the following command:

```
rpm -Uvh xlc-license.13.1.6-*.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 11.
- Alternatively, upgrade your Community Edition as follows:
 1. Install the license packages using the following command:

```
dpkg -iG xlc-license.13.1.6-*_ppc64el.deb
```
 2. Run the `xlc_configure` utility to accept the license if it is not accepted, and reconfigure the compiler.

Uninstalling the compiler

You must use the Linux `rpm` or `dpkg` utility to uninstall IBM XL C/C++ for Linux, V13.1.6.

IBM XL C/C++ for Linux, V13.1.6 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 18.
- 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.8.1.6` is a shared component if IBM XL Fortran for Linux, V15.1.6 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, V13.1.6

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 13.1.6.
- The IBM MASS library package has a *V.R.M* of 8.1.6.

On SLES, RHEL, and CentOS

To uninstall IBM XL C/C++ for Linux, V13.1.6, take the following steps:

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

```
rpm -e xlc.13.1.6 libxlc-devel.13.1.6 \
xlc-license.13.1.6

xlc-license-community.13.1.6
```

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, V15.1.6" 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.8.1.6 libxlsmp-devel.4.1.6
```

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, V13.1.6 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.13.1.6 libxlc-devel.13.1.6 \  
xlc-license.13.1.6
```

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, V15.1.6" 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.8.1.6 libxlsmp-devel.4.1.6
```

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.

Chapter 3. Installing IBM XL C/C++ for Linux, V13.1 for big endian distributions

If you are installing IBM XL C/C++ for Linux, V13.1 for big endian distributions, follow the instructions in this section.

Preparing for installing IBM XL C/C++ for Linux, V13.1

To prepare for installing IBM XL C/C++ for Linux, V13.1, do the following tasks:

- Consult the product README file 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.

Coexistence with previous version compilers

Installing this version of the compiler does not remove or change previous versions of the compiler, only the C++ runtime library and the SMP runtime library are migrated to the new versions. This was achieved by differentiating the installation fileset names from those of the previous versions, including the version, release and modification level (V.R.M) in the name, and installing to new locations that include V.R.M in the installation paths.

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

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

- Use `xlc_install` to install the full product version of the XL C/C++ compiler as described in “Basic installation” on page 37. It uninstalls the evaluation copy and then installs the full product version.
- Alternatively, you can do the following tasks:

1. Install the new license packages using the following command:

```
rpm -Uvh new_license_package [--prefix installation_path]
```

where *new_license_package* is the license package included in the package you are installing, and *installation_path* is the target directory for installation that is specified by the `--prefix` option of the nondefault installation (NDI) command.

2. Accept the new license using one of the following utilities:
 - The `new_install` utility with `-prefix installation_path` when applicable
 - The LAP tool
3. Run the `xlc_configure` utility to reconfigure the compiler. If you used the `new_install` utility in step 2, skip this step.

If your compiler is installed in an NDI location, you must use the `--prefix` option to specify the installation path of the license RPM packages (`*-eval-license` and `*-license` packages) the same as that of the main compiler

packages (x1* packages). Furthermore, when you run the new_install utility, you must use the -prefix option to specify the same path for the x1* packages.

The installation packages

Table 11 lists the packages that are supplied with the installation image, and the default locations to which they are installed during a basic installation. You can use the rpm utility to review the packages. For example, to view package information and the package file list, enter the following rpm command:

```
rpm -qip package_name
```

For the rules on installing packages to custom, nondefault locations, see “Installing packages to multiple, nondefault locations” on page 47.

Table 11. IBM XL C/C++ for Linux, V13.1 packages and default installation locations

| Package name | Package description | Default installation location |
|---|---|--|
| xlsmp.msg.rte | IBM SMP messages package | /opt/ibmcmp/msg/ |
| xlsmp.rte | IBM SMP runtime package | /opt/ibmcmp/lib/ /opt/ibmcmp/lib64/ |
| libxlsmp-devel | IBM SMP static library package | /opt/ibm/xlsmp/4.1.0/ |
| libxlmass-devel | IBM Mathematical Acceleration Subsystem (MASS) package (Not redistributable) | /opt/ibm/xlmass/8.1.0/ |
| vacpp.rte | IBM XL C/C++ runtime package | /opt/ibmcmp/lib/ /opt/ibmcmp/lib64/ /opt/ibmcmp/vacpp/13.1/lib |
| xlc-eval-license ¹ | IBM XL C/C++ license package for the evaluation copy (Not redistributable) | /opt/ibm/xlC/13.1.0/lib/ |
| xlc-license ² | IBM XL C/C++ license package for XL C/C++ for Linux, V13.1(Not redistributable) | |
| libxlc-devel | IBM XL C/C++ libraries package | /opt/ibm/xlC/13.1.0/lib/ /opt/ibm/xlC/13.1.0/lib64/ |
| xlc | IBM XL C/C++ compiler package (Not redistributable) | /opt/ibm/xlC/13.1.0/ /opt/ibm/xlC/13.1.0/samples/ /opt/ibm/xlC/13.1.0/doc/ /opt/ibm/xlC/13.1.0/man/ |
| Notes: | | |
| 1. Only included in the evaluation copy of the XL C/C++ for Linux compiler. | | |
| 2. Only included in XL C/C++ for Linux, V13.1. | | |

Note: /opt/ibm/ is the default prefix for installation of images. The C++ runtime package and SMP runtime package will still be installed in /opt/ibmcmp/ as previous versions for backward compatibility.

Table 12. Utilization reporting tool package 1.2

| Package name | Package description | Default installation location |
|--------------|------------------------------------|---|
| urt | Utilization reporting tool package | /opt/ibmurt /opt/ibmurt/1.2/ /opt/ibmurt/1.2/bin/ /opt/ibmurt/1.2/config/ /opt/ibmurt/1.2/msg |

Note: The utilization reporting tool can be installed on the same machine as the compiler or on a different machine. You must use rpm utilities to install this package. It typically needs to be installed only once, no matter how many compiler installations you have. This tool is useful for those who are responsible for monitoring the overall compiler usage within an organization. Individual developers and build teams should not need to use this tool.

To install the utilization reporting tool, see the instructions provided in README.urt in the root directory of the installation image.

National language support

IBM XL C/C++ for Linux, V13.1 messages support the following language locales:

- en_US
- en_US.utf8
- en_US.UTF-8
- ja_JP
- ja_JP.eucjp
- ja_JP.utf8
- ja_JP.UTF-8
- zh_CN
- zh_CN.gb18030
- zh_CN.gb2312
- zh_CN.gbk
- zh_CN.utf8
- zh_CN.UTF-8

English is the default national language and en_US is the default locale. After the installation, you can set the NLSPATH so that messages are displayed in a different language. See “Enabling the error messages” on page 59.

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, V13.1:

- **Operating system:** You can use one of the following operating systems supported by the IBM Power Systems servers:
 - Red Hat Enterprise Linux 6.4 (RHEL 6.4)
 - Red Hat Enterprise Linux 6.5 (RHEL 6.5)
 - Red Hat Enterprise Linux 6.6 (RHEL 6.6)

- Red Hat Enterprise Linux 7.0 (RHEL 7.0)
- SUSE Linux Enterprise Server 11 Service Pack 2 (SLES 11 SP2)
- SUSE Linux Enterprise Server 11 Service Pack 3 (SLES 11 SP3)

- **Hardware:**

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/>.

- **Storage:**

- Approximately 200 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 35.

- **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 36.

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

Table 13. Required GNU and Perl packages for the RHEL 6.4, RHEL 6.5, and RHEL 6.6 operating systems

| 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 |
| compat-libstdc++-33 | 3.2.3 |
| perl | 5.0 or later Note: Perl V5.8 is shipped and automatically installed with the RHEL 6.4, RHEL 6.5, and RHEL 6.6 operating systems. |

Note: The 32-bit libraries are not installed by default on RHEL 6.4, RHEL 6.5, or RHEL 6.6; they must be installed from the RHEL 6.4, RHEL 6.5, or RHEL 6.6 operating system image.

Table 14. Required GNU and Perl packages for the RHEL 7.0 operating system

| Package name | Version requirements |
|--------------|----------------------|
| gcc | 4.8.2 |
| gcc-c++ | 4.8.2 |
| glibc* | 2.17 |

Table 14. Required GNU and Perl packages for the RHEL 7.0 operating system (continued)

| Package name | Version requirements |
|---------------------|---|
| glibc-devel* | 2.17 |
| libgcc* | 4.8.2 |
| libstdc++* | 4.8.2 |
| libstdc++-devel* | 4.8.2 |
| compat-libstdc++-33 | 3.2.3 |
| perl | 5.0 or later Note: Perl V5.8 is shipped and automatically installed with the RHEL 7.0 operating system. |

Note: The 32-bit libraries are not installed by default on RHEL 7.0; they must be installed from the RHEL 7.0 operating system image.

Table 15. Required GNU and Perl packages for the SLES 11 SP2 and SLES 11 SP3 operating systems

| Package name | Version requirements |
|-------------------------|--|
| gcc-32bit | 4.3.4 |
| gcc | 4.3.4 |
| gcc-c++ | 4.3.4 |
| glibc-devel-32bit | 2.11.3 |
| glibc-devel | 2.11.3 |
| libgcc46-32bit | 4.6.1 |
| libgcc46 | 4.6.1 |
| libstdc++46-32bit | 4.6.1 |
| libstdc++46 | 4.6.1 |
| libstdc++43-devel-32bit | 4.3.4 |
| libstdc++43-devel | 4.3.4 |
| perl | 5.0 or later Note: Perl V5.10 is shipped and automatically installed with the SLES 11 SP2 and SLES 11 SP3 operating systems. |

To view the documentation that is packaged with IBM XL C/C++ for Linux, V13.1, you will need:

- 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, V13.1 requires about 200 MB of hard disk storage space. This amount of space accommodates the optional samples and documentation that are shipped with the product.

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.

Verifying that the required GNU and Perl packages are installed

Before you install IBM XL C/C++ for Linux, V13.1, you should verify that the required GNU and Perl packages were installed with the operating system.

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

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

- For the RHEL 6.4 operating system, see Table 13 on page 34.
- For the RHEL 7.0 operating system, see Table 14 on page 34.
- For the SLES 11 SP2 operating system, see Table 15 on page 35.

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

```
rpm -q 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 required for that package.

Example: Determining the installed version of gcc-c++:

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

```
rpm -q gcc-c++
```

If gcc-c++ version 4.3 is installed, you will get a result similar to the following output:

```
gcc-c++-4.3-62.198
```

Previewing the license agreements

Before installing IBM XL C/C++ for Linux, V13.1, 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 basic installation, the license files can be found in the `/opt/ibm/xlC/13.1.0/` directory with the same names as above.

Note: You can use the utilization tracking and reporting feature to ensure the compiler usage is compliant with the number of Concurrent User licenses you have purchased. For more information, see “Tracking and reporting compiler usage” in the *XL C/C++ Compiler Reference* (for big endian).

Basic installation

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

- Install IBM XL C/C++ for Linux, V13.1 on a system where no IBM XL compiler is currently installed.
- Install IBM XL C/C++ for Linux, V13.1 on a system and update the XL C++ runtime library and SMP runtime library where an older version of IBM XL compiler is currently installed.
- Install IBM XL C/C++ for Linux, V13.1 on a system where IBM XL Fortran for Linux, V15.1 has already been installed.

In this case, both compilers will use the IBM MASS library provided with IBM XL C/C++ for Linux, V13.1.

For information about installing IBM XL Fortran for Linux, V15.1, refer to the *IBM XL Fortran for Linux, V15.1 Installation Guide*.

- Install an update on a system where IBM XL C/C++ for Linux, V13.1 has already been installed.

By running the `xlc_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 exits without installing the compiler and no files will be written to your system.

Alternatively, you can use the `install` utility from the root directory of the installation DVD to perform a basic installation.

You should consider a basic installation of IBM XL C/C++ for Linux, V13.1 with the `xlc_install` or `install` utility on a system as long as *both* of the following conditions apply:

- You are installing the compiler to the default location, `/opt/ibm/`.
- You are maintaining a single version of the product on your system.

If either of these conditions does not apply, use the procedures in “Advanced installation” on page 42.

Tasks for basic installation

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

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

If you are using the basic installation to install IBM XL C/C++ for Linux, V13.1, the previous version of the XL compiler is not uninstalled, only the C++ runtime libraries and SMP runtime library are migrated to the new versions.

Table 16. Steps for basic installation

| Task | References |
|--|---|
| 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 33 |
| Uninstall any previously installed compiler runtime package of the same product type and SMP runtime package. | <ul style="list-style-type: none"> • “Running the install utility for a new installation” • “Running the xlc_install utility for a new installation” on page 40 |
| Use the install or xlc_install utility to accept or decline the license. | <ul style="list-style-type: none"> • “Running the install utility for a new installation” • “Running the xlc_install utility for a new installation” on page 40 |
| Use the install or xlc_install utility to install and configure the compiler, using the default paths. | <ul style="list-style-type: none"> • “Running the install utility for a new installation” • “Running the xlc_install utility for a new installation” on page 40 |
| Confirm that the compiler packages were successfully installed, and test the installation. | <ul style="list-style-type: none"> • “Querying for installed packages” on page 62 • “Testing the installation” on page 57 |
| Enable the compiler man pages. | “Enabling the man pages” on page 58 |
| If your system locale, 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 59 |
| Optional: If you did not choose to create symbolic links to the compiler invocation commands during the installation process, set up the environment to locate the invocation commands without the full path. Otherwise, you can skip this step. | “Setting up the environment for the invocation commands” on page 60 |

Running the install utility for a new installation

The install utility is the recommended tool for installing XL C/C++. The utility is located in the root directory of the installation image.

Before you begin

Ensure that Perl is installed on your system before you run the installation. See “Verifying that the required GNU and Perl packages are installed” on page 9.

About this task

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

- Detects the current architecture (big endian or little endian)
- Automatically invokes the xlc_install utility to install all compiler packages into the default location, /opt/ibm/, and configure the compiler

To run the install utility to install IBM XL C/C++ for Linux, V13.1.6:

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

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

- If another instance of IBM XL C/C++ for Linux is detected on your system in the same installation directory, you are prompted to uninstall the C++ runtime package and SMP runtime package. Confirm that you want to proceed with the uninstallation. If you choose not to uninstall the existing C++ runtime package and SMP runtime package, the installation process will end.
 - If other versions of the IBM SMP and MASS packages are detected on your system either alone or as part of an installation of IBM XL Fortran for Linux, V15.1, you are prompted to uninstall them. Confirm that you want to proceed with uninstalling the existing IBM SMP and MASS packages. If you choose not to uninstall the previously installed components, the installation process will end.
2. Read the license agreement and licensing information. If you agree to the licensing terms, accept the license agreement to continue installation.
You are prompted to create symbolic links for the compiler invocations in the `/usr/bin/` directory.
 3. Optional: Create the symbolic links.

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 60.

If you chose to create symbolic links, the following links are created in the `/usr/bin/` subdirectory:

- `gxc`
- `gxc++`
- `gxc`
- `xc`
- `xc++`
- `xc`
- `xc_r`
- `xc++_r`
- `xc_r`

For more information about other specialized invocations that might be available, refer to *IBM XL C/C++ for Linux, V13.1 Compiler Reference, “Invoking the compiler”*.

Note: Symbolic links of these 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`, `new_install`, `showpdf`, `xc_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
```

Note: NLS is not supported for messages that are issued by the `install` utility.

- The configuration file is generated. Its location is `/opt/ibm/xlC/13.1.0/etc/xlc.cfg.$OSRelease.gcc$gccVersion`. For example, `/opt/ibm/xlC/13.1.0/etc/xlc.cfg.sles11.gcc432` or `/opt/ibm/xlC/13.1.0/etc/xlc.cfg.rhel6.2.gcc446`.

Running the `xlc_install` utility for a new installation

If you used the `install` utility to install the compiler, `xlc_install` is already run by that utility. If you did not use the `install` utility, you can find the `xlc_install` utility under the `/images/bigEndian/` directory of the installation image.

Before you begin

Because the `xlc_install` utility is written in the Perl language, you must ensure that Perl is installed on your system before you run the utility. See “Verifying that the required GNU and Perl packages are installed” on page 36.

About this task

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

- Checks for all prerequisite software packages
- Uninstalls any previously installed XL C/C++ runtime package and SMP runtime package
- Installs all compiler packages into the default location, `/opt/ibm/`, except for the C++ runtime package and SMP runtime package, which are installed into the `/opt/ibmcmp/` directory
- Automatically invokes the `new_install` utility, which installs the license file and generates the default configuration file
- Optionally creates symbolic links in `/usr/bin/` to the compiler invocation commands
- Generates an installation log in the `/tmp/` directory

To run the `xlc_install` utility to install IBM XL C/C++ for Linux, V13.1:

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

```
cd /cdrom
./xlc_install
```

For additional arguments that you can specify for `xlc_install`, see “`xlc_install` options” on page 41.

- If another instance of IBM XL C/C++ for Linux is detected on your system in the same installation directory, you are prompted to uninstall the C++ runtime package and SMP runtime package. Confirm that you want to proceed with the uninstallation. If you choose not to uninstall the existing C++ runtime package and SMP runtime package, the installation process will end.
 - If other versions of the IBM SMP and MASS packages are detected on your system either alone or as part of an installation of IBM XL Fortran for Linux, V15.1, you are prompted to uninstall them. Confirm that you want to proceed with uninstalling the existing IBM SMP and MASS packages. If you choose not to uninstall the previously installed components, the installation process will end.
2. Read the license agreement and licensing information. If you agree to the licensing terms, accept the license agreement to continue installation.

You are prompted to create symbolic links for the compiler invocations in the `/usr/bin/` directory.

3. Optional: Create the symbolic links.

Note: 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 60.

If you chose to create symbolic links, the following links are created in the `/usr/bin/` subdirectory:

- `gxc`
- `gxc++`
- `gxC`
- `xc`
- `xc++`
- `xC`
- `xc_r`
- `xc++_r`
- `xC_r`

For more information about other specialized invocations that may be available, see *IBM XL C/C++ for Linux, V13.1 Compiler Reference, “Invoking the compiler”*.

Tip: The following command links 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`, `new_install`, `showpdf`, `xc_configure`

If all packages are successfully installed:

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

```
"IBM XL C/C++ for Linux, V13.1" Version 13.1.0 has been
successfully installed and configured under prefix "/opt/ibm".
"IBM XL C/C++ for Linux, V13.1" Version 13.1.0.0 has been
successfully installed.
```

Note: NLS is not supported for messages that are issued by the `xc_install` utility.

- The configuration file is generated. Its location is `/opt/ibm/xlC/13.1.0/etc/xlc.cfg.$OSRelease.gcc$gccVersion`. For example, `/opt/ibm/xlC/13.1.0/etc/xlc.cfg.sles11.gcc432` or `/opt/ibm/xlC/13.1.0/etc/xlc.cfg.rhel6.2.gcc446`.
- The installation log is moved to its permanent location: `xlC/13.1.0/xlc_install.log`.

xc_install options

The `xc_install` utility provides the following options:

-h Displays the installation utility help page.

-prefix *installation_path*

Explicitly specifies the path to install all the compiler packages.

Note: If this option is not used, the default installation location is `/opt/ibm/`. The C++ runtime package and SMP runtime package will still be installed in `/opt/ibmcmp/` as previous versions for backward compatibility.

-rpmloc *rpmlocation_path*

Explicitly specifies the path where all compiler packages are located. The default *rpmlocation_path* is `./images-x86/rpms` or `./images/rpms`, which is relative to the path of the installation tool.

Note: For most users, the `-rpmloc rpmlocation_path` option is not required in the installation invocation. If you use the utility directly from the DVD or the electronic image, it will automatically determine the source location of the packages.

-U Updates the compiler to the *Version.Release.Modification.Fix-Build* (V.R.M.F-B) level that the installation utility version supports. For more information, see “Updating a basic installation using the `xlc_install` utility” on page 51.

Note: The `-U` option cannot update the compiler to a different *Version.Release.Modification* (V.R.M) level.

-v Displays debugging information generated during the installation of the compiler.

-vv Displays extra debugging information generated during the installation of the compiler.

Advanced installation

It is highly recommended that you install IBM XL C/C++ for Linux, V13.1 to the default location and use the procedure provided in “Basic installation” on page 37. However, you must use alternative procedures if any of the following scenarios apply to you:

- You want to maintain more than one version of IBM XL C/C++ for Linux in a nondefault location on the same system.
- You want to update an existing version of IBM XL C/C++ for Linux that is installed in a nondefault location.
- You want to try out a new update of the compiler before removing an existing installation from the default location.

In all of these scenarios, you can use the `rpm` utility to install the compiler packages; you can also use the `xlc_install` 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 `new_install` or `xlc_configure` utility. If you use the `xlc_configure` utility instead of the `new_install` utility to configure an installation that is not PTF updates, before you run the `xlc_configure` utility, you must manually run the IBM License Acceptance Process (LAP) tool to review and accept the license. For the instructions about using the LAP tool, see “Running the IBM License Acceptance Process (LAP) tool” on page 48. For configuration procedures, see “Configuring IBM XL C/C++ for Linux, V13.1 (for advanced users)” on page 53.

Tasks for advanced installation

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

If you want to maintain multiple versions of XL C/C++ in the same location on the same system, follow the steps in this table.

Table 17. Steps for installing to the same location as an earlier version

| Task | References |
|--|---|
| Become either the root user or a user with administrator privileges. | Documentation supplied with the operating system |
| If you do not need to maintain multiple versions of the product on your system, remove any existing versions of IBM XL C/C++ for Linux . | "Uninstalling IBM XL C/C++ for Linux, V13.1" on page 63 |
| Ensure that all system prerequisites are satisfied. | "System prerequisites" on page 33 |
| Install IBM XL C/C++ for Linux, V13.1. | "Installing XL C/C++ to the same location as an earlier version" on page 44 |
| Run the IBM License Acceptance Process (LAP) tool to review and accept the license. | "Running the IBM License Acceptance Process (LAP) tool" on page 48 |
| Use the <code>xlc_configure</code> utility to configure the compiler. | "Running the <code>xlc_configure</code> utility directly" on page 54 |
| Confirm that the compiler packages were successfully installed, and test the installation. | <ul style="list-style-type: none"> • "Querying for installed packages" on page 62 • "Testing the installation" on page 57 |
| Enable the compiler man pages. | "Enabling the man pages" on page 58 |
| If your system locale, your encoding, or both 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 59 |
| Optional: Set up the environment to locate the invocation commands without the full path. | "Setting up the environment for the invocation commands" on page 60 |

If you want to install IBM XL C/C++ for Linux, V13.1 to a nondefault location, separate from any other versions of XL C/C++ on the same system, follow the steps in this table.

Table 18. Steps for installing to a location separate from earlier versions

| Task | References |
|---|---|
| Become either the root user or a user with administrator privileges. | Documentation supplied with the operating system |
| If you do not need to maintain multiple versions of the product on your system, remove any existing versions of IBM XL C/C++ for Linux. | "Uninstalling IBM XL C/C++ for Linux, V13.1" on page 63 |
| Ensure that all system prerequisites are satisfied. | "System prerequisites" on page 33 |
| Install IBM XL C/C++ for Linux, V13.1. | "Installing XL C/C++ to a location separate from earlier versions" on page 45 |

Table 18. Steps for installing to a location separate from earlier versions (continued)

| Task | References |
|--|---|
| If you will use the <code>xlc_configure</code> utility to configure the compiler, run the IBM License Acceptance Process (LAP) tool to review and accept the license. | "Running the IBM License Acceptance Process (LAP) tool" on page 48 |
| Use the <code>new_install</code> or <code>xlc_configure</code> utility to configure the compiler. | "Configuring IBM XL C/C++ for Linux, V13.1 (for advanced users)" on page 53 |
| Confirm that the compiler packages were successfully installed, and test the installation. | <ul style="list-style-type: none"> • "Querying for installed packages" on page 62 • "Testing the installation" on page 57 |
| Enable the compiler man pages. | "Enabling the man pages" on page 58 |
| If your system locale, your encoding, or both 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 59 |
| Optional: Set up the environment to locate the invocation commands without the full path. | "Setting up the environment for the invocation commands" on page 60 |

Installing XL C/C++ to co-reside with earlier versions

If you want to maintain more than one version of XL C/C++ on your system, you can use the `rpm` utility or `xlc_install` utility to perform an advanced installation.

To install IBM XL C/C++ for Linux, V13.1 to co-reside with earlier versions of XL C/C++ on the same system, you have the following options:

- You can install multiple versions in the same location, while forcing all versions to use the most recent runtime environment. This option is recommended if you want to continue using the different versions of IBM XL C/C++ for Linux over the long term.

Multiple versions of IBM XL C/C++ for Linux can co-reside in the same location as long as the most recently released version does not attempt to use runtime packages that pre-date it. If your existing version is installed in the default location, do not try to use `xlc_install` to install the additional version because it will remove the existing version. Instead, use the procedure in "Installing XL C/C++ to the same location as an earlier version"

- You can allow each version to use the runtime environment that was shipped with it. This option is recommended if you want to phase a migration to the later version. In this case, you will need to install IBM XL C/C++ for Linux, V13.1 in a separate location from all other versions of IBM XL C/C++ for Linux. To do so, follow one of the procedures in "Installing XL C/C++ to a location separate from earlier versions" on page 45.

Installing XL C/C++ to the same location as an earlier version About this task

The following procedure describes how to install IBM XL C/C++ for Linux, V13.1 and update the runtime packages used by an existing version of IBM XL C/C++ for Linux, V12.1 in the same location.

This procedure assumes the following:

- You are installing IBM XL C/C++ for Linux, V13.1 to a location where IBM XL C/C++ for Linux, V12.1 has already been installed (by default, V12.1 is installed to the /opt/ibmcmp/ directory, V13.1 is installed to the /opt/ibm/ directory).
- Your current working directory contains all IBM XL C/C++ for Linux, V13.1 packages, and no other RPM packages.
- The existing runtime component versions listed in Table 19 are installed in the same installation location (by default, /opt/ibmcmp/).

Table 19. XL C/C++ SMP and runtime packages

| Existing XL C/C++ V12.1 runtime packages | New XL C/C++ V13.1 runtime packages |
|--|-------------------------------------|
| xlsmp.msg.rte-3.1.0.\$F-\$B | xlsmp.msg.rte-4.1.0.\$F-\$B |
| xlsmp.rte-3.1.0.\$F-\$B | xlsmp.rte-4.1.0.\$F-\$B |
| vacpp.rte-12.1.0.\$F-\$B | vacpp.rte-13.1.0.0.\$F-\$B |
| Note: \$F is the PTF number of the package currently installed on your system. \$B is the build number of the package currently installed on your system. | |

To install IBM XL C/C++ for Linux, V13.1, and update the IBM XL C/C++ for Linux, V12.1 XL SMP and runtime packages:

1. Remove the existing runtime packages by issuing the following commands to avoid future dependency errors:

```
rpm -e vacpp.rte-12.1.0.0-$B --nodeps
rpm -e xlsmp.rte-3.1.0.0-$B --nodeps
rpm -e xlsmp.msg.rte-3.1.0.0-$B --nodeps
```

Note: This example shows package names with no applied updates.

2. To replace the deleted runtime packages and to install IBM XL C/C++ for Linux, V13.1, issue the following command:

```
rpm -ivh *.rpm
```

3. As a precaution, make a backup copy of any existing configuration file.
4. Generate the default IBM XL C/C++ for Linux, V13.1 configuration file:

```
installation_path/xlc/13.1.0/bin/xlc_configure -gcc /usr -gcc64 /usr
-ibmcmp installation_path/ -ibmrt installation_runtime_path/
installation_path/xlc/13.1.0/etc/xlc.base.cfg
-o installation_path/xlc/13.1.0/etc/xlc.cfg.$OSRelease.gcc$gccVersion
```

where:

- *installation_path* is the location in which all IBM XL C/C++ for Linux, V13.1 packages are installed (by default, /opt/ibm/).
- *installation_runtime_path* is the location in which all IBM XL C/C++ for Linux, V13.1 runtime packages are installed (by default, /opt/ibmcmp/).
- *\$OSRelease* is the level of the Operating System.
- *\$gccVersion* is the GCC version installed on the system.

For example, xlc.cfg.sles11.gcc432, or xlc.cfg.rhel6.2.gcc446.

Installing XL C/C++ to a location separate from earlier versions

About this task

This section provides procedures for two possible installation scenarios:

- You can 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/. The procedure for doing so is provided in “Installing all packages to a single, nondefault location.”

- For highly specialized situations only, you can install groups of compiler packages to multiple, different locations. For example, you could install all the compiler library packages to one directory, the runtime environment packages to another directory, and so on. However, certain packages must be installed together in the same directory. Rules for installing packages into multiple nondefault directories, as well as procedures for doing so, are provided in “Installing packages to multiple, nondefault locations” on page 47.

Installing all packages to a single, nondefault location:

About this task

To install all compiler packages to a single nondefault directory, you have the following options:

- Use the `xlc_install` utility. This option is recommended if you want to install all compiler packages to a single, nondefault directory.

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

```
cd /cdrom
./xlc_install -prefix installation_path
```

For additional arguments that you can specify for `xlc_install`, see “`xlc_install` options” on page 41.

- Use the `rpm` utility. Ensure that your current working directory contains all of the packages for IBM XL C/C++ for Linux, V13.1 and no other RPM packages. From your current working directory, use the following command:

```
rpm -ivh *.rpm --prefix installation_path
```

In the preceding commands, *installation_path* is a directory that is not `/opt/ibm/`.

Updating multiple versions of PTFs installed under separate locations:

About this task

Updating multiple versions of PTFs that are earlier than XL C/C++ for Linux V12.1

To update multiple versions of PTFs that are earlier than XL C/C++ for Linux V12.1, take the following steps:

1. Uninstall all previous packages except the `vac.lic` RPM package using the `rpm -e` command.

If a package is the prerequisite of a different compiler, take the following steps:

- a. Use the `--nodeps` option to uninstall the prerequisite package.
- b. Reconfigure the compiler that has its prerequisite updated in Step 2 below. You can skip this step if both compilers are updated.

2. Install the PTF packages using the `rpm` utility with the following options if applicable:

- If a previous version of compiler or PTFs are installed under prefix `$CMPpath`, use the `--prefix $CMPpath` option.
- If a later version of compiler is installed on the system, use the `--oldpackage` option.

3. Reconfigure the compiler by running the `vac_configure` utility. For instructions, see “Running the `xlc_configure` utility directly” on page 54. In XL C/C++ for Linux, V13.1, `vac_configure` is renamed to `xlc_configure`.
4. Test the updated compiler. For instructions, see “Testing the installation” on page 57.

Updating multiple versions of PTFs that are later than XL C/C++ for Linux V12.1

To update multiple versions of PTF that are later than XL C/C++ for Linux V12.1, take the following steps:

1. Run the `xlc_install` utility to install the PTF packages with the `-U` option. If a previous version of compiler or PTFs are installed under prefix `$CMPpath`, use the `--prefix $CMPpath` option. When you are prompted with the "uninstall all previous installations" question, answer **yes**.
2. Test the updated compiler. For instructions, see “Testing the installation” on page 57.

Installing packages to multiple, nondefault locations:

About this task

For highly specialized situations only, you might need to install different packages to different locations.

Note: If you want to install packages in different subdirectories, do not install any packages in the `/opt/ibm/` or `/opt/ibmcomp/` directory.

Table 20 provides information about which packages must be installed together in the same directory and which can be installed in any directory.

Table 20. Rules for installing packages to multiple, nondefault locations

| Package Name | Package Description | Rules for installation to a nondefault location |
|---|--|---|
| <code>xlsmp.msg.rte</code> | IBM SMP messages package | All XL SMP packages must be installed in the same location. For the remainder of this document, the name <code>xlsmpprt_path</code> is used to refer to this location. |
| <code>xlsmp.rte</code> | IBM SMP runtime package | |
| <code>libxlsmp-devel</code> | IBM SMP static library package | |
| <code>libxlmass-devel</code> | IBM Mathematical Acceleration Subsystem (MASS) package | Any location. For the remainder of this document, the name <code>xlmass_path</code> is used to refer to this location. |
| <code>vacpp.rte</code> | IBM XL C/C++ runtime package | All XL C/C++ runtime packages must be installed in the same location. For the remainder of this document, the name <code>xlrte_path</code> is used to refer to this location. |
| <code>xlc-eval-license¹</code> | IBM XL C/C++ license package for the evaluation copy | Any location. For the remainder of this document, the name <code>lic_path</code> is used to refer to this location. |
| <code>xlc-license²</code> | IBM XL C/C++ license package for XL C/C++ for Linux, V13.1 | |

Table 20. Rules for installing packages to multiple, nondefault locations (continued)

| Package Name | Package Description | Rules for installation to a nondefault location |
|--|---|--|
| libxlc-devel | IBM XL C/C++ compiler libraries package | All XL C/C++ compiler and library packages must be installed in the same location. For the remainder of this document, the name <i>xlcmp_path</i> is used to refer to this location. |
| xlc | IBM XL C/C++ compiler package | |
| Notes: <ol style="list-style-type: none"> 1. Only included in the evaluation copy of the XL C/C++ for Linux compiler. 2. Only included in XL C/C++ for Linux, V13.1. If you are installing a PTF, ignore the xlc-license package. | | |

To install a relocatable RPM package to any location other than the default location, issue the following command for each group of packages you want to install to a nondefault directory:

```
rpm -ivh package --prefix package_installation_path
```

where *package_installation_path* is a directory other than */opt/ibm/* or */opt/ibmcmp/*, and corresponds to one of the appropriate paths listed in Table 20 on page 47.

Example: Installing XL C/C++ to multiple nondefault directories:

About this task

To avoid dependency errors during the installation of IBM XL C/C++ for Linux, V13.1, issue the following commands in the order given:

```
rpm -ivh xlsmp.msg.rte-4.1.0.0-$B.ppc64.rpm --prefix $XLRTEpath
rpm -ivh xlsmp.rte-4.1.0.0-$B.ppc64.rpm --prefix $XLRTEpath
rpm -ivh libxlsmp-devel.4.1.0-4.1.0.0-$B.ppc64.rpm --prefix $SMPPpath
rpm -ivh libxlmass-devel.8.1.0-8.1.0.0-$B.ppc64.rpm --prefix $MASSpath

rpm -ivh vacpp.rte-13.1.0.0-$B.ppc64.rpm --prefix $RTEpath
rpm -ivh xlc-license.13.1.0-13.1.0.0-$B.ppc64.rpm --prefix $LICpath
rpm -ivh libxlc-devel.13.1.0-13.1.0.0-$B.ppc64.rpm --prefix $CMPpath
rpm -ivh xlc.13.1.0-13.1.0.0-$B.ppc64.rpm --prefix $CMPpath
```

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

Running the IBM License Acceptance Process (LAP) tool

When you install a compiler that is not a PTF package, before you run the *xlc_configure* utility, you must manually run the IBM License Acceptance Process (LAP) tool to review and accept the license.

About this task

Note: If you are already using the *new_install* or *xlc_install* utility, skip this step because the LAP tool is always run when you are running these utilities.

The instruction assumes the compiler main package is installed under prefix *\$CMPpath*, which is the location that *xlc* is installed.

- If the compiler is installed in the default location, the installation path is */opt/ibm/*.

- If the compiler is installed in a nondefault location, see “Installing XL C/C++ to a location separate from earlier versions” on page 45 for information about the nondefault location prefix.

Take the following steps to run the LAP tool:

Procedure

1. Change to the installation directory:
`cd $CMPpath/xlc/13.1.0`
2. Run the LAP tool by specifying the following command:
`lap/lapc -l $CMPpath/xlc/13.1.0/lap -s $CMPpath/xlc/13.1.0/lap`
3. Read the license agreement. During the running of the LAP tool, if your language preference is not available, exit the application and view the PDF version of the license (`license.pdf` under `$CMPpath/xlc/13.1.0` directory) before proceeding.
4. If you agree to the licensing terms, accept the license agreement. The successful return code of the LAP tool is 9.

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 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.13.1.0-13.1.0.0-$B.ppc64.rpm` RPM file that has not yet been installed, issue the following command:

```
rpm --qf="%{summary}\n" -qp libxlc-devel.13.1.0-13.1.0.0-130701.ppc64.rpm
```

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

```
IBM XL C/C++ for Linux, V13.1 - C/C++ Compiler Libraries Package (SLES11/RHEL6/RHEL7)
```

Installing an update

An update, or program temporary fix (PTF), of IBM XL C/C++ for Linux, V13.1 provides a fix or multiple fixes to the product. You can download updates from the support website: https://www.ibm.com/support/home/product/Q833644Y89702U61/XL_C/C++_for_Linux.

Every PTF update package comes in `tar.gz` (compressed) format and includes a version of the `xlc_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, V13.1 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 using the `xlc_install` utility” on page 51. If you are applying the update to an installation in a nondefault location, you must use one of the procedures in “Updating an advanced installation” on page 52. If you want to try out a PTF before installing it, use the procedure in “Testing a PTF before you install it” on page 50.

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 21. Steps for basic installation: update installation

| Task | References |
|---|--|
| Become either the root user or a user with administrator privileges. | Documentation supplied with the operating system |
| Use the <code>xlc_install</code> tool to install the update packages. | "Updating a basic installation using the <code>xlc_install</code> utility" on page 51 |
| 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 62• "Testing the installation" on page 57 |
| Optional: If you do not choose to create symbolic links to the compiler invocation commands during the update process, set up the environment to locate the invocation commands without the full path. Otherwise, you can skip this step. | "Setting up the environment for the invocation commands" on page 60 |

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

Table 22. Steps for advanced installation: update installation

| Task | References |
|---|--|
| Become either the root user or a user with administrator privileges. | Documentation supplied with the operating system |
| Uninstall all XL C/C++ packages except <code>xlc-license</code> . | "Updating an advanced installation" on page 52 |
| Install update packages to the nondefault location. | "Updating an advanced installation" on page 52 |
| Configure the compiler. | "Configuring IBM XL C/C++ for Linux, V13.1 (for advanced users)" on page 53 |
| 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 62• "Testing the installation" on page 57 |
| Optional: Set up the environment to locate the invocation commands without the full path. | "Setting up the environment for the invocation commands" on page 60 |

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.

About this task

To install a PTF to a nondefault location, use the procedure described in "Installing packages to multiple, nondefault locations" on page 47, using the `rpm` utility.

Note: If you are installing a different version of compiler but not a PTF to a separate location, use the procedure described in “Installing XL C/C++ to a location separate from earlier versions” on page 45.

To configure the compiler, use the procedure described in “Running the `xlc_configure` utility directly” on page 54.

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 `xlc_install` utility to install the new update, see “Updating a basic installation using the `xlc_install` utility.”

If the existing version on the system was installed using an advanced installation, see “Updating an advanced installation” on page 52.

Updating a basic installation using the `xlc_install` utility

The `xlc_install` utility is the recommended tool for updating a basic installation of IBM XL C/C++ for Linux, V13.1. It is located in the root directory of the installation image.

About this task

You can use the `xlc_install` utility to update IBM XL C/C++ for Linux, V13.1 if all the following conditions have been met:

- The base version of IBM XL C/C++ for Linux, V13.1 is already successfully installed in the `/opt/ibm/` directory.
- The update package has been uncompressed and unpacked on the system.

When you run the `xlc_install` utility to apply an update, it performs the following tasks:

- Checks for all prerequisite software packages
- Uninstalls IBM XL C/C++ for Linux, V13.1 packages
- Installs updated compiler packages into the default location
- Automatically invokes the `new_install` utility, which installs the license file, renames the old configuration file, and generates a new configuration file
- Optionally creates symbolic links in `/usr/bin/` to the compiler invocation commands
- Generates an installation log in the `/tmp/` directory

To run the `xlc_install` utility to apply an update for IBM XL C/C++ for Linux, V13.1:

1. Change to the directory in which you have unpacked the update package:

```
cd /home/root/update/x1C/mmmYYYY
```

where `mmmYYYY` is the month and year of the update shipment date. (For example, `dec2014` indicates a shipment date in December of 2014.)

2. Install the update by issuing the following command:

```
./xlc_install -U
```

For additional arguments that you can specify for `xlc_install`, see “`xlc_install` options” on page 41.

3. Confirm that you want to uninstall any previously installed IBM XL C/C++ for Linux, V13.1 packages.
4. Confirm that you want to uninstall any IBM SMP and MASS packages that were previously installed with IBM XL C/C++ for Linux.

Note: `xlc_install` uninstalls these packages from their existing location and reinstalls them to the default location, `/opt/ibmcmp/`. Therefore, if they were previously installed in a nondefault location as part of an IBM XL Fortran installation, you must run `xlf_configure` to reconfigure the IBM XL Fortran compiler to point to the default location for these packages. For procedures, see “Running the `xlf_configure` utility directly” in the *IBM XL Fortran for Linux, V15.1 Installation Guide*.

5. Accept the license agreement and licensing information.
6. Optionally, create the symbolic links for the compiler invocations in the `/usr/bin/` directory.

Note: 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 60.

If you choose to create symbolic links, the following links are created in the `/usr/bin/` subdirectory:

- `gxc`
- `gxc++`
- `gxc`
- `xlc`
- `xlc++`
- `xlc`
- `xlc_r`
- `xlc++_r`
- `xlc_r`

7. If you customized the previously generated configuration file, manually edit `installation_path/xlc/13.1.0/etc/xlc.cfg.$OSRelease.gcc$gccVersion` to replicate those changes in the newly generated configuration file.

Updating an advanced installation

To update an advanced installation of IBM XL C/C++ for Linux, V13.1, it is recommended that you first uninstall all IBM XL C/C++ for Linux, V13.1 packages except the license package, `xlc-license`.

If you attempt to install a new PTF to a location where an older PTF is currently installed, the installation of the new PTF will fail due to a location conflict: two files that belong to two different PTF levels cannot be installed to the same location at the same time. See “Uninstalling IBM XL C/C++ for Linux, V13.1” on page 63 for instructions on how to uninstall XL C/C++.

After the old packages have been uninstalled, follow the instructions in “Advanced installation” on page 42 to install the PTF packages to the location of your choice.

Configuring IBM XL C/C++ for Linux, V13.1 (for advanced users)

Before you can run IBM XL C/C++ for Linux, V13.1, you must configure (or re-configure) the compiler if any of the following conditions apply to you:

- You did not use `xlc_install` to install the compiler, or the configuration step failed with `xlc_install`.
- Compiler components were relocated after installation.

The compiler provides two configuration tools: `new_install` and `xlc_configure`, both located in the `installation_path/xlc/13.1.0/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/`.

It is recommended that you use the `new_install` utility to configure the compiler, provided that *all* of the following conditions are met:

- All IBM XL C/C++ for Linux, V13.1 compiler packages are installed in the same location, `installation_path`.
- Only one version of GCC is installed in your system and it can be found in the `PATH` environment variable.
- You have root or administrator privileges.
- You want to generate the configuration file in the `installation_path/xlc/13.1.0/etc/` directory.

For instructions, see “Running the `new_install` utility.”

You should invoke the `xlc_configure` utility directly *only* when at least one of the following conditions is true:

- You have multiple versions of IBM XL C/C++ for Linux installed on your system.
- You receive an error message from the `new_install` command. (See “Troubleshooting the installation and configuration” on page 64.)
- You want the generated configuration file to be placed in a location that is different from `installation_path/xlc/13.1.0/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.$OSRelease.gcc$gccVersion`, can be written to any location where you have write permission. You would not need root or administrative privileges.

For instructions, see “Running the `xlc_configure` utility directly” on page 54.

Running the `new_install` utility

Use the `new_install` utility that is provided with an instance of IBM XL C/C++ for Linux, V13.1 compiler to configure an advanced installation if all compiler packages corresponding to this instance have been installed in the same location, and only one version of GCC is installed on your system.

About this task

The `new_install` utility performs the following tasks:

- Backs up any existing configuration file that is located in *installation_path/xlC/13.1.0/etc/xlc.cfg.\$OSRelease.gcc\$gccVersion*.
- Queries the RPM database for the paths of the following packages in the *PATH* environment variable, and then uses the obtained values to run the *xlc_configure* utility:
 - compiler packages
 - 32-bit GCC (*gcc32path*)
 - 64-bit GCC (*gcc64path*)
- Installs the license file.
- Generates the configuration file in the default location *installation_path/xlC/13.1.0/etc/xlc.cfg.\$OSRelease.gcc\$gccVersion*.

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/*.

Take the following steps to run the *new_install* utility:

1. Change to the directory that contains the compiler executable files:

```
cd installation_path/xlC/13.1.0/bin/
```

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/*.

2. Run the following command:

```
./new_install -prefix installation_path
```

3. Read the license agreement and licensing information. If you agree to the licensing terms, accept the license agreement and licensing information.

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.

About this task

Take the following steps to run the *xlc_configure* utility:

1. Change to the directory that contains the compiler executables:

```
cd installation_path/xlC/13.1.0/bin/
```

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/*. If the compiler has been installed to multiple, nondefault locations, the *new_install* utility can be found in *\$CMPpath/xlC/13.1.0/bin/*.

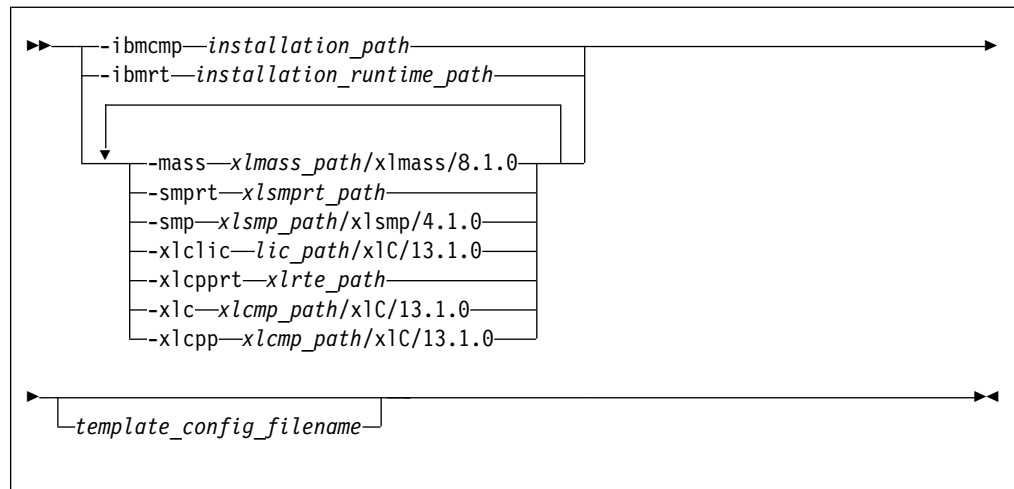
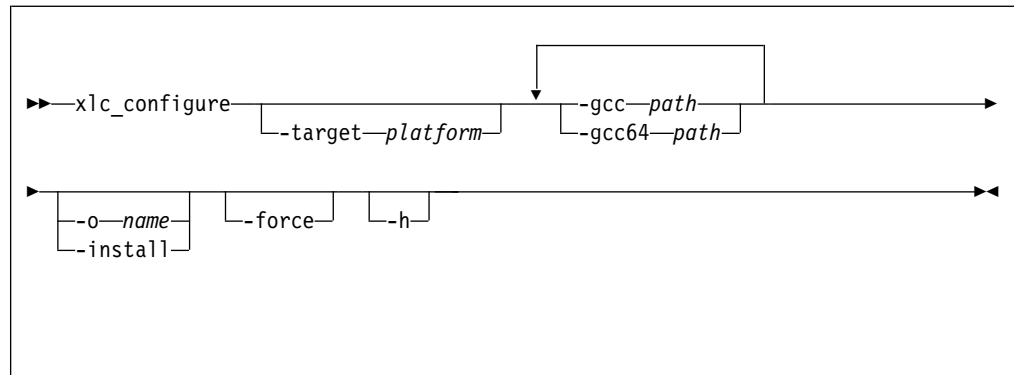
2. Run the following command:

```
./xlc_configure options
```

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:

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

-target platform

Specifies the operating system platform. Valid names are:

- sles
- rhel

If you do not specify a value, the default is the one corresponding to the host operating system.

-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`

-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`

-o file_name

Specifies the name of the configuration file to be generated. By default, output is written to the standard output only.

-install

Generates the default configuration file as *installation_path/xlC/13.1.0/etc/xlc.cfg.\$OSRelease.gcc\$gccVersion*, where *\$OSRelease* is the level of the Operating System, and *\$gccVersion* is the GCC version installed on the system. For example, *xlc.cfg.sles11.gcc432*, or *xlc.cfg.rhel6.4.gcc447*. By default, output is written to the display only.

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/*.

-force Forces the *xlc_configure* utility to overwrite any existing output file with the same name and path as those that are specified by the **-o** or **-install** options. By default, if you do not use **-force**, *xlc_configure* issues an error message and stops if the specified file already exists.

-ibmcmp *installation_path*

Specifies the path where all of the IBM XL C/C++ for Linux packages except the runtime packages are installed (if all packages are installed in the same path). By default, the path is */opt/ibm/*.

-ibmrt *installation_runtime_path*

Specifies the path where all of the IBM XL C/C++ for Linux runtime packages are installed. By default, the path is */opt/ibmcmp/*.

-mass *xlmass_path/xlmass/8.1.0/*

Specifies the path where the *xlmass* package is installed. By default, the full path is */opt/ibm/xlmass/8.1.0/*.

-smprt *xlsmprt_path*

Specifies the path where the *xlsmp.msg.rte* and *xlsmp.rte* packages are installed. By default, the full path is */opt/ibmcmp/*.

-smp *xlsmp_path/xlsmp/4.1.0/*

Specifies the path where the *libxlsmp-devel* package is installed. By default, the full path is */opt/ibm/xlsmp/4.1.0/*.

-xlic *lic_path/xlC/13.1.0/*

Specifies the path where the *xlc-license* package is installed. By default, the full path is */opt/ibm/xlC/13.1.0/lib/*.

-xlcpprt *xlrte_path*

Specifies the path where the *vacpp.rte* package is installed. By default, the full path is */opt/ibmcmp/*.

-xlc *xlcmp_path/xlC/13.1.0/*

Specifies the C path where the *xlc* and *libxlc-devel* packages are installed. By default, the full path is */opt/ibm/xlC/13.1.0/*.

-xlcpp *xlcmp_path/xlC/13.1.0/*

Specifies the C++ path where the *xlc* and *libxlc-devel* packages are installed. By default, the full path is */opt/ibm/xlC/13.1.0/*.

template_config_file_name

The input file that is used to construct the configuration file. By default, this is */opt/ibm/xlC/13.1.0/etc/xlc.base.cfg*. If you relocated the *xlc* package to *xlcmp_path* but want to use the default template, specify:

xlcmp_path/xlC/13.1.0/etc/xlc.base.cfg

After installing IBM XL C/C++ for Linux, V13.1

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 topic applies to all users.
- Enabling the man pages: This topic applies to all users.
- Enabling the error messages: This topic 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 topic only applies to users who have not used `xlc_install` to install or update the product, or who have not created symbolic links during the installation process with `xlc_install`.
- Accessing the local documentation: This topic applies to all users.
- Querying for installed packages: This topic applies to all users.

Testing the installation

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:

1. Create the following C program and name the source file as `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/13.1.0/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 as `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/13.1.0/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".

Enabling utilization tracking

Utilization tracking can be used to detect whether the compiler usage exceeds your entitlement based on the number of Concurrent User licenses you have purchased.

Utilization tracking is disabled by default. You can enable it by modifying the entry **-qnoenabletracking** to **-qenabletracking** in the utilization tracking configuration file, `urtxlc_cpp1301linux.cfg`. Various other aspects of utilization tracking can also be specified in this file. In each installation of the compiler, there is a symlink named `urt_client.cfg` that points to the utilization tracking configuration file. You might need to change the symlink if you use a copy of the utilization tracking configuration file in a different location or if you use multiple copies of the file. For more information about utilization tracking, see "Configuring utilization tracking" in the *XL C/C++ Compiler Reference*.

Enabling the man pages

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

The IBM XL C/C++ for Linux, V13.1 man pages support the following locales:

- en_US
- en_US.utf8
- en_US.UTF-8
- ja_JP
- ja_JP.eucjp
- ja_JP.utf8
- ja_JP.UTF-8
- zh_CN
- zh_CN.gb18030
- zh_CN.gb2312

- zh_CN.gbk
- zh_CN.utf8
- zh_CN.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/x1C/13.1.0/man/LANG:$MANPATH
```

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

```
setenv MANPATH installation_path/x1C/13.1.0/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.

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 62.

Enabling the error messages

If your system uses the en_US locale and encoding, the compiler message catalogs are automatically configured to display correctly, regardless of whether you used the basic or advanced method of installation and configuration. However, if your system uses any other supported locale (for a list of supported language locales, see “National language support” on page 33), you must set the *NLSPATH* environment variable, so that the compiler and runtime functions can find the appropriate message catalogs after the installation.

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:  
xlsmpert_path/msg/%L/%N:  
xlrte_path/msg/%L/%N:  
xlcmp_path/x1C/13.1.0/msg/%L/%N
```

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

```
setenv NLSPATH $NLSPATH:  
xlsmpert_path/msg/%L/%N:  
xlrte_path/msg/%L/%N:  
xlcmp_path/x1C/13.1.0/msg/%L/%N
```

where:

- *xlsmpert_path* is the installation location of the SMP packages. By default, it is */opt/ibmcmp/*.

- *xlrte_path* is the installation location of the IBM XL C/C++ for Linux, V13.1 runtime packages. By default, this is `/opt/ibmcmp/`.
- *xlcmp_path* is the installation location of the IBM XL C/C++ for Linux, V13.1 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.

Setting up the environment for the invocation commands

If you used the `xlc_install` utility to install the compiler and you selected to create the symbolic links at that time, 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 installed 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.”

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

To use IBM XL C/C++ for Linux, V13.1 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/13.1.0/bin/
```
- If you are using the C shell, use the following command:

```
setenv PATH $PATH:installation_path/xlc/13.1.0/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.

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/13.1.0/bin/` directory.

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

- `gxc`
- `gxc++`
- `gxC`
- `xc`
- `xc++`
- `xC`
- `xc_r`
- `xc++_r`
- `xC_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`, `new_install`, `showpdf`, `xlc_configure`

Enter the following command to create a symbolic link:

```
ln -s installation_path/xlc/13.1.0/bin/invocation /usr/bin/invocation
```

where:

- *installation_path* is the location where you have installed the compiler packages. By default, this is `/opt/ibm/`.
- *invocation* is one of the compiler invocations (such as `xlc`) in `installation_path/xlc/13.1.0/bin/`.

Example: Creating a symbolic link to a compiler invocation:

This example assumes that all of IBM XL C/C++ for Linux, V13.1 is installed in the default location `/opt/ibm/`.

```
ln -s /opt/ibm/xlc/13.1.0/bin/xlc /usr/bin/xlc
```

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 PTF update package is installed.

ILMT and TADd, once 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/13.1.0/properties/version/` directory. Otherwise, if IBM XL C/C++ for Linux is installed in a nondefault location, the signature files are in the `installation_path/xlC/13.1.0/properties/version/` directory, where *installation_path* is the target directory for installation specified by `--prefix` option of the nondefault installation command.

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

Accessing the local documentation

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

Viewing the manual pages

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

For instructions about how to enable the man pages for viewing, see “Enabling the man pages” on page 58.

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` command.

About this task

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.

Note: The C++ runtime package and SMP runtime package still use *V.R.M.F-B* in the package names for backward compatibility.

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 vacpp -e xlsmp -e xlmass
```

The result is a list containing all of the packages as shown in Table 11 on page 32. If none of the packages was properly installed, there will be no output from the command.

For more information about the rpm command, see the installation information in the Linux documentation at:

- Red Hat Enterprise Linux documentation page: https://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/
- SUSE documentation page: <https://www.suse.com/documentation/>

Uninstalling IBM XL C/C++ for Linux, V13.1

IBM XL C/C++ for Linux, V13.1 does not provide a stand-alone uninstallation tool. You must use the Linux rpm utility to uninstall IBM XL C/C++ for Linux, V13.1.

Notes:

- You must have root access to uninstall the compiler.
- Whenever you uninstall a package, specify the *V.R.M-V.R.M.F-B* (Version.Release.Modification-Version.Release.Modification.Fix-Build level) of the package. For information about how to determine the package's *V.R.M-V.R.M.F-B*, see “Querying for installed packages” on page 62.
- 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.

Note: The C++ runtime package and SMP runtime package still use *V.R.M.F-B* in the package names for backward compatibility.

- You cannot uninstall packages that are required by other packages. For example, `libxlmass-devel` is a shared component if IBM XL Fortran for Linux, V15.1 is also installed on the same system.
- The uninstallation commands do not remove any configuration files that have been generated by the `new_install` or `xlc_configure` utility.

Example: Uninstalling IBM XL C/C++ for Linux, V13.1

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-V.R.M.F-B* of `13.1.0-13.1.0.0-$B`.
- The IBM MASS library package has a *V.R.M-V.R.M.F-B* of `8.1.0-8.1.0.0-$B`.
- The IBM SMP library packages have a *V.R.M-V.R.M.F-B* of `4.1.0-4.1.0.0-$B`.

Notes:

- *\$B* is the build number of the package that is installed on your system.
- The C++ runtime package and SMP runtime package still use *V.R.M.F-B* in the package names.

To uninstall IBM XL C/C++ for Linux, V13.1, take the following steps:

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

```
rpm -e xlc.13.1.0-13.1.0.0-$B
rpm -e libxlc-devel.13.1.0-13.1.0.0-$B
rpm -e xlc-license.13.1.0-13.1.0.0-$B
rpm -e vacpp.rte-13.1.0.0-$B
```

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, V15.1" in the *XL Fortran Installation Guide*.

- 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 them in the order given below.

```
rpm -e libxlmass-devel.8.1.0-8.1.0.0-$B
rpm -e libxlsmp-devel.4.1.0-4.1.0.0-$B
rpm -e xlsmp.rte-4.1.0.0-$B
rpm -e xlsmp.msg.rte-4.1.0.0-$B
```

Troubleshooting the installation and configuration

The compiler generates messages to help you recognize and respond to error conditions. This section provides recommended responses.

At the beginning of the installation process, the installation utility creates a new log file in `/tmp/`. The temporary log files are uniquely named.

After the installation is completed successfully, the log file is moved to the default installation location for future reference. If the installation fails, the installation log is in the `/tmp/` directory. Regardless of whether the installation succeeds or fails, the file name of the corresponding installation log is displayed as part of the standard output.

Use the information in this section to help you respond to any problems you might encounter when you install and configure IBM XL C/C++ for Linux, V13.1.

The specified directory `rpmlocation_path` does not exist

Scenario

You are running the `xlc_install` utility to install the compiler to the default location when you get the following error message:

```
ERROR: The specified directory, "rpmlocation_path", does not exist.
```

Action

Ensure that you have specified the location of the existing compiler packages correctly. You might need to use the `-rpmloc rpmlocation_path` option if you moved the `xlc_install` utility to a different location than the one provided in the installation image. For more information, see "`xlc_install` options" on page 41.

`rpmlocation_path` does not contain all of the RPM packages for the XL compiler

Scenario

You are running the `xlc_install` utility to install the compiler to the default location when you get the following error message:

```
ERROR: rpmlocation_path does not contain all of the RPM packages for the XL compiler.
```

Action

Before running the `xlc_install` utility again, ensure you have all of the packages listed in Table 11 on page 32 in the path. You might need to use the `-rpmloc rpmlocation_path` option if you have moved the `xlc_install`

utility to a different location than the one provided in the installation image. For more information, see “xlc_install options” on page 41.

Notices

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

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

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

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

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

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

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

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

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

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

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

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

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, 2017.

PRIVACY POLICY CONSIDERATIONS:

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, or to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

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

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

A

- advanced installation
 - examples (BE) 48
 - multiple nondefault locations (BE) 47
 - multiple versions (BE) 44
 - overview 11, 14
 - overview (BE) 42
 - single nondefault locations (BE) 46
 - the new_install utility (BE) 53
- audience
 - overview v
- available space
 - determining 8
 - determining (BE) 35

B

- basic installation
 - overview 4, 25, 28, 37
 - overview (BE) 11
 - the install utility 12
 - the install utility (BE) 38
 - utilities (BE) 40
- big endian distributions
 - installation
 - overview 31

C

- compiler invocations
 - environment variables
 - PATH 22
 - PATH (BE) 60
 - symbolic links
 - creating 23
 - creating (BE) 61
- configuration
 - conditions 16
 - conditions (BE) 53
 - options 17
 - options (BE) 56
 - utilities (BE) 54
- configuration files
 - generation 13, 40, 41

D

- default installation
 - overview 4, 25, 28
- documentation
 - manual pages (BE) 62

E

- enabling
 - error messages (BE) 59
 - IBM License Metric Tool 24
 - IBM License Metric Tool (BE) 61
 - manual pages (BE) 58

- enabling (*continued*)
 - Tivoli Asset Discovery for Distributed 24
 - Tivoli Asset Discovery for Distributed (BE) 61
- error message
 - incomplete 22

G

- GNU packages
 - verifying 9
 - verifying (BE) 36

H

- hard disk space
 - verifying 8
 - verifying (BE) 35
- hardware requirements
 - offloading to GPU 10

I

- installation packages
 - dpkg packages 6
 - national language support 6
 - national language support (BE) 33
 - RPM packages 5
 - RPM packages (BE) 32
- installation utilities
 - options (BE) 41
- invocation commands
 - environments 22
 - environments (BE) 60

L

- licenses
 - the LAP tool (BE) 48
- little endian distributions
 - installation
 - overview 4
- local documentation
 - manual pages (BE) 62

M

- manual pages
 - viewing (BE) 62
- message
 - description missing 22

N

- nondefault installation
 - overview 11

O

- operating system requirements
 - offloading to GPU 10

P

- Perl packages
 - verifying 9
 - verifying (BE) 36
- post-installation
 - overview 18
- PTF installation
 - multiple versions 27
 - multiple versions (BE) 46
 - overview 25
 - overview (BE) 49
 - steps 27
 - steps (BE) 51

Q

- queries
 - installed packages 18
 - installed packages (BE) 62
 - RPM packages 15
 - RPM packages (BE) 49

R

- requirements
 - hardware 10
 - operating systems 10
 - software 10

S

- software requirements
 - offloading to GPU 10
- system prerequisites
 - requirements 6
 - requirements (BE) 33

T

- tests
 - installation 19
 - installation (BE) 57
 - PTF installation 27
 - PTF installation (BE) 50
- trademarks
 - copyright 69
- troubleshooting
 - non-existing directories (BE) 64
 - overview (BE) 64
 - RPM packages (BE) 64

U

uninstallation

- examples (BE) 63
- overview 28
- overview (BE) 63

V

verification

- GNU packages 9
- GNU packages (BE) 36
- hard disk space 8
- hard disk space (BE) 35
- Perl packages 9
- Perl packages (BE) 36



Product Number: 5765-J08; 5725-C73

Printed in USA

GC27-6540-05

