

IBM XL C/C++ for Multicore Acceleration for Linux,
V10.1



Installation Guide

Version 10.1

IBM XL C/C++ for Multicore Acceleration for Linux,
V10.1



Installation Guide

Version 10.1

Note

Before using this information and the product it supports, read the information in "Notices" on page 33.

First edition

This edition applies to IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1 and IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1 (Programs 5724-T42 & 5724-T43), 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 International Business Machines Corporation 2003, 2008.

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

Contents

About this information	v
Who should read this document.	v
How to use this document.	v
How this document is organized	vi
Conventions	vi
Related information	x
IBM XL C/C++ 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 Multicore Acceleration for Linux, V10.1	1
The installation image and packages	1
System prerequisites.	3
Verifying the amount of hard disk space available	4
Verifying that the required GNU and Perl packages are installed	4
Chapter 2. Basic installation	5
Tasks for basic installation.	5
Running the <code>xl_install</code> utility for a new installation	6
<code>xl_install</code> options	8
Chapter 3. Advanced installation	9
Tasks for advanced installation	9
Installing XL C/C++ to co-reside with earlier versions	10
Installing XL C/C++ to the same location as an earlier version	11
Installing XL C/C++ to a location separate from earlier versions	11
Querying for RPM package summaries	14
Chapter 4. Installing an update	15
Tasks for update installation.	15
Testing a PTF before you install it	16
Updating a basic installation using the <code>xl_install</code> utility	16
Updating an advanced installation.	18
Chapter 5. Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 (for advanced users)	19
Running the <code>new_install</code> utility	20

Running the <code>xl_configure</code> utility directly	20
<code>xl_configure</code> options	20

Chapter 6. After installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 **23**

Testing the installation.	23
Basic example: Creating and running “Hello World”	23
Enabling the man pages	24
Enabling the error messages.	25
Setting up the environment for the invocation commands.	25
Setting the PATH environment variable to include the path to the compiler invocations	25
Creating symbolic links to the compiler invocations	26
Enabling IBM Tivoli License Compliance Manager	27
Accessing the local documentation	27
Viewing the PDF documentation	27
Querying for installed packages	28

Chapter 7. Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 **29**

Example: Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1	29
--	----

Chapter 8. Troubleshooting the installation and configuration **31**

The specified directory <code>rpmlocation_path</code> does not exist.	31
<code>rpmlocation_path</code> does not contain	31
Could not determine location of 32-bit or 64-bit SDK GCC (RHEL 5.2).	32

Notices **33**

Trademarks and service marks	35
--	----

Index **37**

About this information

This guide provides detailed installation instructions for IBM® XL C/C++ for Multicore Acceleration for Linux®, V10.1. It guides you through multiple ways to perform tasks and directs you to reference information in cases requiring atypical installations. It also shows you how to test the installation, launch remotely-accessible HTML help, and enable and view different types of documentation. Please read it carefully before installing. Please also read the README file in the root directory of your installation media, which contains current information about the compiler.

Who should read this document

This document is intended for anyone responsible for installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.

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

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

How to use this document

This document provides procedures for three main installation scenarios:

"Basic" installation

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

For an overview of the steps that you need to follow to perform a basic installation, refer to Chapter 2, "Basic installation," on page 5.

"Advanced" installation

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

For an overview of the steps that you need to follow to perform an advanced installation, refer to Chapter 3, "Advanced installation," on page 9.

"Update" installation

This scenario applies to users who have obtained a Program Temporary Fix (PTF) package for an existing XL C/C++ V10.1 installation.

For an overview of the steps required to install a PTF, refer to Chapter 4, “Installing an update,” on page 15.

For highly specialized installation scenarios that are outside the scope of this document, refer to the Technotes at <http://www.ibm.com/software/awdtools/xlcpp>.

How this document is organized

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

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

Phase	Chapters	User segment
Pre-installation	Chapter 1, “Before installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1,” on page 1	All users
Installation	Chapter 2, “Basic installation,” on page 5	Users who: <ul style="list-style-type: none">• Want to use the simplest, most direct installation process• Do not have any special requirements, such as the use of multiple versions of the compiler.
	Chapter 3, “Advanced installation,” on page 9	Users who: <ul style="list-style-type: none">• Want to install the compiler in a non-default location• Want to have multiple versions of the compiler on the same system
Post-installation	Chapter 4, “Installing an update,” on page 15	Users who want to update XL C/C++ V10.1 to the next fix level
	Chapter 6, “After installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1,” on page 23	All users
Product removal	Chapter 7, “Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1,” on page 29	Any user who needs to remove an XL C/C++ compiler from the system

Conventions

Typographical conventions

The following table explains the typographical conventions used in the IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 information.

Table 2. *Typographical conventions*

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

Other conventions

In addition to typographical conventions, the following conventions are used:

- (*SPU only*) indicates functionality that only applies to code targeting the Synergistic Processor Unit (SPU), whether it is compiled using an `spu` or `cbe` prefixed compiler invocation command.
- (*PPU only*) indicates functionality that only applies to code targeting the Power Processor Unit (PPU), whether it is compiled using a `ppu` or `cbe` prefixed compiler invocation command.

Qualifying elements (icons)

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

Table 3. *Qualifying elements*

Qualifier/Icon	Meaning
C only, or C only begins  	The text describes a feature that is supported in the C language only; or describes behavior that is specific to the C language.
C only ends	

Table 3. Qualifying elements (continued)

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

Syntax diagrams

Throughout this information, diagrams illustrate XL C/C++ syntax. This section will help 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):



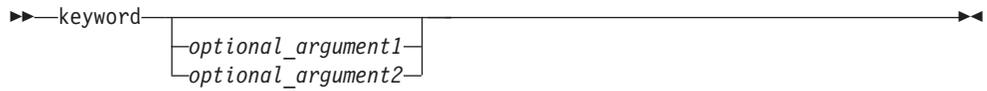
- Optional items are shown below the main path:



- 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:



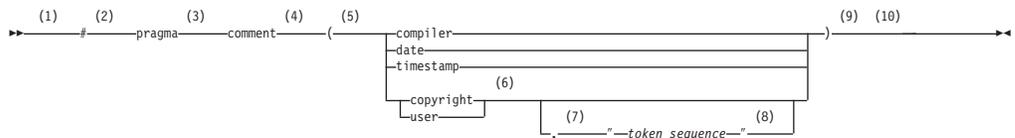
- The item that is the default is shown above the main path.



- 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.

Sample syntax diagram

The following syntax diagram example shows the syntax for the **#pragma comment** directive.



Notes:

- 1 This is the start of the syntax diagram.
 - 2 The symbol # must appear first.
 - 3 The keyword pragma must appear following the # symbol.
 - 4 The name of the pragma comment must appear following the keyword pragma.
 - 5 An opening parenthesis must be present.
 - 6 The comment type must be entered only as one of the types indicated: compiler, date, timestamp, copyright, or user.
 - 7 A comma must appear between the comment type copyright or user, and an optional character string.
 - 8 A character string must follow the comma. The character string must be enclosed in double quotation marks.
 - 9 A closing parenthesis is required.
 - 10 This is the end of the syntax diagram.
- The following examples of the **#pragma comment** directive are syntactically correct according to the diagram shown above:

```
#pragma comment(date)
#pragma comment(user)
#pragma comment(copyright,"This text will appear in the module")
```

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. Also, examples may use different compiler invocation commands interchangeably or simply indicate *invocation*. For detailed information on the commands available to invoke the compiler see "Invoking the compiler" in the *XL C/C++ Compiler Reference*.

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

Related information

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

IBM XL C/C++ information

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

- 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 of the installation CD.
- 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 Multicore Acceleration for Linux, V10.1 Installation Guide*.
- Information center
 The information center of searchable HTML files is viewable on the Web at <http://publib.boulder.ibm.com/infocenter/cellcomp/v101v121/index.jsp>.
- PDF documents
 PDF documents are located by default in the `/opt/ibmcmp/xlc/cbe/10.1/doc/en_US/pdf/` directory. The PDF files are also available on the Web at <http://www.ibm.com/software/awdtools/xlcpp/multicore/library/>.
 The following files comprise the full set of XL C/C++ product information:

Table 4. XL C/C++ PDF files

Document title	PDF file name	Description
<i>IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 Installation Guide, GC23-8574-00</i>	install.pdf	Contains information for installing XL C/C++ and configuring your environment for basic compilation and program execution.

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

Document title	PDF file name	Description
<i>Getting Started with IBM XL C/C++ for Multicore Acceleration for Linux, V10.1, GC23-8572-00</i>	getstart.pdf	Contains an introduction to the XL C/C++ product, with information on setting up and configuring your environment, compiling and linking programs, and troubleshooting compilation errors.
<i>IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 Compiler Reference, SC23-8570-00</i>	compiler.pdf	Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions.
<i>IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 Language Reference, SC23-8573-00</i>	langref.pdf	Contains information about the C and C++ programming languages, as supported by IBM, including language extensions for portability and conformance to nonproprietary standards.
<i>IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 Optimization and Programming Guide, SC23-8571-00</i>	proguide.pdf	Contains information on advanced programming topics, such as application porting, library development, application optimization, and the XL C/C++ high-performance libraries.

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

More information related to XL C/C++ including redbooks, white papers, tutorials, and other articles, is available on the Web at:

<http://www.ibm.com/software/awdtools/xlcpp/multicore/library/>

Standards and specifications

XL C/C++ is designed to support the following standards and specifications. You can refer to these standards 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 14882:1998*, also known as C++98.
- *Information Technology – Programming languages – C++, ISO/IEC 14882:2003(E)*, also known as *Standard C++*.
- *Information Technology – Programming languages – Extensions for the programming language C to support new character data types, ISO/IEC DTR 19769*. This draft technical report has been accepted by the C standards committee, and is available at <http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1040.pdf>.
- *Draft Technical Report on C++ Library Extensions, ISO/IEC DTR 19768*. This draft technical report has been submitted to the C++ standards committee, and is available at <http://www.open-std.org/JTC1/SC22/WG21/docs/papers/2005/n1836.pdf>.

- *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.

Other IBM information

- *IBM C/C++ Language Extensions for Cell Broadband Engine Architecture*, available at <http://publib.boulder.ibm.com/infocenter/systems/topic/eiccg/eiccgkickoff.htm>
- Specifications, white papers, and other technical information for the Cell Broadband Engine™ architecture are available at http://www.ibm.com/chips/techlib/techlib.nsf/products/Cell_Broadband_Engine.
- The Cell Broadband Engine resource center, at <http://www.ibm.com/developerworks/power/cell/>, is the central repository for technical information, including articles, tutorials, programming guides, and educational resources.
- The IBM Systems Information Center, at <http://publib.boulder.ibm.com/infocenter/systems/index.jsp> is a resource for technical information about IBM systems, including the Cell Broadband Engine solution.

Other information

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

Technical support

Additional technical support is available from the XL C/C++ Support page at <http://www.ibm.com/software/awdtools/xlcpp/support>. 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 e-mail to compinfo@ca.ibm.com.

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

How to send your comments

Your feedback is important in helping 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 by e-mail to compinfo@ca.ibm.com.

Be sure to include the name of the information, the part number of the information, 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 Multicore Acceleration for Linux, V10.1

This installation guide covers IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1 and IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1. The term IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 will refer to both x86 and Power Systems. x86 and Power Systems will be explicitly specified when there is a difference between the two products.

Before you install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1:

- Consult the product README file for any last minute updates you may need to be aware of.
- Familiarize yourself with the installation image, which contains the installable compiler packages, and a utility program for installation.
- 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.

The installation image and packages

The installation image for IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 is available on an installation CD or for download from an IBM web site to a local drive.

The image includes:

- READMEs, license agreement files, notices, and documentation.
- A set of RPM packages.
- An installation tool, `xlc_install`, to install and configure the compiler for a basic installation.

Note: XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1 and XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1 have the same installation path. This is due to the fact that the two compilers are each only supported on their corresponding pair of hardware architectures and Linux distributions:

- **IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1** is supported on Intel® x86 systems running Red Hat Enterprise Linux 5 Update 2(RHEL 5.2)
- **IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1** is supported on IBM POWER™ systems running Red Hat Enterprise Linux 5 Update 2(RHEL 5.2)

Installation Packages

Table 5 on page 2 lists the packages that are supplied with the installation image, and the locations to which they are installed by default during a basic installation. (For the rules on installing packages to custom, non-default locations, see “Installing packages to multiple, non-default locations” on page 12.)

You can use the rpm utility to review the packages. For example, to view package information and its file list, issue the following rpm query command:

```
rpm -qip package_name
```

Table 5. IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 packages and default installation locations

Package Name	Package Description	Default installation location
ppu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (PPU only)	/opt/ibmcmp/xlmass/cbe/5.0/
spu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (SPU only)	/opt/ibmcmp/xlmass/cbe/5.0/
cell-xlc-rte	IBM XL C/C++ runtime package	/opt/ibmcmp/lib/cbe/ /opt/ibmcmp/lib64/cbe/
cell-xlc-rte-lnk	IBM XL C/C++ runtime links package	/opt/ibmcmp/xlc/cbe/10.1/
cell-xlc-lic	IBM XL C/C++ license package	/opt/ibmcmp/xlc/cbe/10.1/
cell-xlc-lib	IBM XL C/C++ libraries package	/opt/ibmcmp/xlc/cbe/10.1/
cell-xlc-cmp	IBM XL C/C++ compiler package	/opt/ibmcmp/xlc/cbe/10.1/
cell-xlc-omp	IBM XL C/C++ OpenMP package	/opt/ibmcmp/xlc/cbe/10.1/
cell-xlc-help	IBM XL C/C++ help documentation package	/opt/ibmcmp/xlc/cbe/10.1/doc/
cell-xlc-samples	IBM XL C/C++ samples package	/opt/ibmcmp/xlc/cbe/10.1/
cell-xlc-man	IBM XL C/C++ man pages package	/opt/ibmcmp/xlc/cbe/10.1/man/

National language support

IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 messages support the following language locales:

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

- zh_CN.utf8

English is the default national language and en_US is the default locale. Following installation, you can set the NLSPATH so that messages are displayed in a different language. See “Enabling the error messages” on page 25.

System prerequisites

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

The following are the requirements for installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1:

- **Operating system:**
 - Red Hat Enterprise Linux 5.2 (RHEL 5.2) for IBM Power
 - Red Hat Enterprise Linux 5.2 (RHEL 5.2) for Intel x86
- **Hardware:**
 - IBM Power Systems technology-based system: 64-bit PPC
 - Intel x86 systems: Either x86 or x86-64 2-GHz Pentium 4 processor (minimum)
- **Storage:**
 - Approximately 300 MB for product packages
 - Minimum of 256 MB of RAM

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 the procedure in “Verifying the amount of hard disk space available” on page 4.

- **Required software:**

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

Table 6. Required GNU and Perl packages for the RHEL 5.2 operating system

Package name	Version requirements
gcc	4.1.2
gcc-c++	4.1.2
glibc*	2.5
glibc-devel*	2.5
libgcc*	4.1.2
libstdc++*	4.1.2
libstdc++-devel*	4.1.2
IBM Software Development Kit (SDK) for Multicore Acceleration V3.1	3.1
perl	5.0 or greater Note: Perl V5.8 is shipped and automatically installed with the RHEL 5.2 operating system.

To view the documentation that is packaged with IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 you will need:

- A graphical desktop environment (such as K Desktop Environment or GNOME) that supports web browsers and PDF viewers
- PDF viewer (to access PDF documentation)

Verifying the amount of hard disk space available

IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 requires about 300 MB of hard disk storage space. This amount 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/ibmcmp/ppuxlc/cbe/`):

```
df -h /opt
```

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

```
df -h installation_path
```

where *installation_path* represents the non-default location.

Verifying that the required GNU and Perl packages are installed

Before you install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1, 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's installation media. For a list of the required packages, see "System prerequisites" on page 3.

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 -qa | grep package_name
```

If the package has been installed, this command should return the package's name and version number. The version number must be the same or higher than the version requirement 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 -qa | grep gcc-c++
```

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

```
gcc-c++-4.1.2-42.e15
```

Chapter 2. Basic installation

IBM XL C/C++ for Multicore Acceleration for Linux provides an interactive utility, `xlc_install`, that walks you through a basic installation. You can use `xlc_install` to:

- Install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 on a system where no IBM XL compiler is currently installed.
- Replace an older version of XL C/C++ with IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.
- Install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 on a system where IBM XL Fortran for Multicore Acceleration for Linux, V12.1 is already installed.

In this case, both compilers will use the IBM MASS library provided with IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.

For information about installing IBM XL Fortran for Multicore Acceleration for Linux, V12.1 refer to the *IBM XL Fortran for Multicore Acceleration for Linux, V12.1 Installation Guide*.

- Install an update on a system where IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 has already been installed.

In this case, use the procedure in “Updating a basic installation using the `xlc_install` utility” on page 16.

The `xlc_install` utility will ask you 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 without installing the compiler and no files will be written to your machine.

You should use the `xlc_install` utility to install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 as long as *both* of the following conditions apply:

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

If either of these conditions does not apply, do *not* use the `xlc_install` utility. Instead, use the procedures in Chapter 3, “Advanced installation,” on page 9.

Tasks for basic installation

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

If the conditions listed in Chapter 2, “Basic installation” match your needs, then a basic installation is the easiest and fastest method you can use to install XL C/C++, as it allows you to automatically uninstall any previously installed IBM XL C/C++ for Multicore Acceleration for Linux compiler, install the latest version, and configure the compiler, all through the use of a single installation tool.

Table 7. Steps for basic installation

Task	For more information, see . . .
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system.

Table 7. Steps for basic installation (continued)

Task	For more information, see . . .
Ensure that all system prerequisites are satisfied.	“System prerequisites” on page 3
Uninstall any previously installed packages of the same product type.	“Running the xlc_install utility for a new installation”
Use the xlc_install tool to accept or decline the license.	“Running the xlc_install utility for a new installation”
Use the xlc_install tool to install and configure the compiler, using the default paths.	“Running the xlc_install utility for a new installation”
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • “Querying for installed packages” on page 28 • “Testing the installation” on page 23
Enable the compiler man pages.	“Enabling the man pages” on page 24
(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 25

Running the xlc_install utility for a new installation

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

Note: Because the xlc_install utility is written in Perl, 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 4.

When you run the xlc_install utility for a new installation, it does the following:

- Checks for all prerequisite software packages
- Uninstalls any previously installed XL C/C++ components
- Installs all compiler packages into the default location
- 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 Multicore Acceleration for Linux, V10.1:

1. If the product CD is mounted at the /cdrom location 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 8.

- If another instance of IBM XL C/C++ for Multicore Acceleration for Linux is detected on your system, you are prompted to uninstall it. Confirm that you

want to proceed with the uninstallation. If you choose not to uninstall the existing instance of the compiler, the installation process will end.

- If other versions of the IBM MASS packages are detected on your system either alone or as part of an IBM XL Fortran for Multicore Acceleration for Linux, V12.1 installation, you are prompted to uninstall them. Confirm that you want to proceed with uninstalling the existing IBM MASS packages. If you choose not to uninstall the previously installed components, the installation process will terminate.
2. You are presented with the licensing agreement and licensing information. Read the licensing agreement and licensing information. If you agree to the licensing terms, accept the license agreement and licensing information to continue installation.

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

3. Optionally, 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 25.

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

- `cbxlc`
- `cbxlc++`
- `cbxlC`
- `ppuxlc++`
- `ppuxlC`
- `ppuxlC_r`
- `ppuxlc++_r`
- `spuxlc++`
- `spuxlC`
- `ppuxlc`
- `ppuxlc_r`
- `spuxlc`
- `ppuc89`
- `ppuc89_r`
- `ppuc99`
- `ppuc99_r`
- `ppucc`
- `ppucc_r`
- `ppugxlc`
- `spugxlc`
- `ppugxlc++`
- `spugxlc++`
- `ppugxlC`
- `spugxlC`

If all packages are successfully installed:

- A message is displayed confirming the successful installation.
- The configuration file is generated. Its location is `/opt/ibmcmp/xlc/cbe/10.1/etc/vac.cfg`. Any previously generated configuration file is renamed and saved in the same directory.
- The installation log is moved to its permanent location: `/opt/ibmcmp/xlc/cbe/10.1/xlc_install.log`.

xlc_install options

The `xlc_install` utility provides the following options:

- h** Displays the installation utility help page.
- rpmloc** *rpmlocation_path*
Explicitly specifies the path where all compiler packages are located. The default *rpmlocation_path* is `./images-x86/rpms` or `./images-ppc/rpms`, which is relative to the path of the installation tool.
- U** Updates the compiler to the *Version.Release.Modification-Fix* (V.R.M-F) level that the installation utility version supports.
- v** Displays debugging information generated during the installation of the compiler.
- vv** Displays extra debugging information generated during the installation of the compiler.

Chapter 3. Advanced installation

It is highly recommended that you install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 to the default location and use the procedure provided in Chapter 2, “Basic installation,” on page 5. However, you will need to use alternate procedures for customized scenarios, including the following:

- You want to maintain more than one version of IBM XL C/C++ for Multicore Acceleration for Linux on the same system.
- 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 must use the rpm utility to install the compiler; you cannot use the xlc_install utility to do so. Once you have successfully installed the compiler to a non-default location, you will need to manually configure the compiler environment using the new_install or xlc_configure utilities; see Chapter 5, “Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 (for advanced users),” on page 19 for procedures.

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, you should follow the steps outlined in the following table.

Table 8. Steps for installing to the same location as an earlier version

Task	For more information, see . . .
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 Multicore Acceleration for Linux .	Chapter 7, “Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1,” on page 29
Ensure that all system prerequisites are satisfied.	“System prerequisites” on page 3
Install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.	“Installing XL C/C++ to the same location as an earlier version” on page 11
Use the xlc_configure tool to configure the compiler.	“Running the xlc_configure utility directly” on page 20
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none">• “Querying for installed packages” on page 28• “Testing the installation” on page 23
Enable the compiler man pages.	“Enabling the man pages” on page 24
(Optional) Set up the environment to locate the invocation commands without the full path.	“Setting up the environment for the invocation commands” on page 25

If you want to install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 to a non-default location, separate from any other versions of XL C/C++ on the same system, you should follow the steps outlined in the following table.

Table 9. Steps for installing to a location separate from earlier versions

Task	For more information, see . . .
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 Multicore Acceleration for Linux .	Chapter 7, “Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1,” on page 29
Ensure that all system prerequisites are satisfied.	“System prerequisites” on page 3
Install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.	“Installing XL C/C++ to a location separate from earlier versions” on page 11
Use the <code>new_install</code> or <code>xlc_configure</code> tool to configure the compiler.	Chapter 5, “Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 (for advanced users),” on page 19
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • “Querying for installed packages” on page 28 • “Testing the installation” on page 23
Enable the compiler man pages.	“Enabling the man pages” on page 24
(Optional) Set up the environment to locate the invocation commands without the full path.	“Setting up the environment for the invocation commands” on page 25

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

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

To install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 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. This option is recommended if you want to continue using the different versions of IBM XL C/C++ for Multicore Acceleration for Linux over the long term.
Multiple versions of IBM XL C/C++ for Multicore Acceleration 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 as it will remove the existing 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. To do so, follow one of the procedures in “Installing XL C/C++ to a location separate from earlier versions” on page 11.

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

The following procedure describes how to install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 and update the runtime packages used by an existing version of IBM for Multicore Acceleration for Linux, V9.0 in the same location.

This procedure assumes the following:

- You are installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 to a location where IBM for Multicore Acceleration for Linux, V9.0 has already been installed (by default, /opt/ibmcmp/).
- Your current working directory contains all IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 packages, and no other RPM packages.
- The existing runtime component versions listed in Table 10 are installed in the same installation location (by default, /opt/ibmcmp/).

Table 10. XL C/C++ runtime packages

Existing XL C/C++ V9.0 runtime package	New XL C/C++V10.1 runtime package
cell-xlc-rte-9.0.0-\$F	cell-xlc-rte-10.1.0-0

Note: \$F is the PTF number of the package currently installed on your system.

To install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1, and update the IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 runtime packages:

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

```
rpm -e cell-xlc-rte-9.0.0-0 --nodeps
```

Note: This example shows package names with no applied updates. The -0 fix level will be different if you have applied any updates.

2. To replace the deleted runtime packages and to install IBM XL C/C++ for Multicore Acceleration for Linux, V10.1, 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 Multicore Acceleration for Linux, V10.1 configuration file:

```
/installation_path/xlc/cbe/10.1/bin/xlc_configure  
-ppugcc32 toolchain_path  
-ppugcc64 toolchain_path  
-spugcc32 toolchain_path  
-ibmcmp /installation_path/  
-o /installation_path/xlc/cbe/10.1/etc/vac.cfg  
/installation_path/xlc/cbe/10.1/etc/vac.cfg
```

where *installation_path* is the location in which all IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 packages are installed (by default, /opt/ibmcmp/) and *toolchain_path* is /opt/cell/toolchain for IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1 and /usr for IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1 .

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

This section provides procedures for two possible installation scenarios:

- You can install all compiler packages to a single non-default location. For example, rather than installing all packages into the default directory `/opt/ibmcmp/`, you could install them to a directory of your choice, such as `/home/mydirectory/`. The procedure for doing so is provided in “Installing all packages to a single, non-default 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. Note, however, that certain packages must be installed together in the same directory; rules for installing packages into multiple non-default directories, as well as procedures for doing so, are provided in “Installing packages to multiple, non-default locations.”

Installing all packages to a single, non-default location

To install all compiler packages to a single non-default directory, ensure that your current working directory contains all of the packages for IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 and no other RPM packages. From your current working directory, use the following command:

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

where *installation_path* is a directory that is not `/opt/ibmcmp/`.

Installing packages to multiple, non-default locations

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/ibmcmp/` directory.

Table 11 provides information about which packages must be installed together in the same directory and which can be installed in any directory.

Table 11. Rules for installing packages to multiple, non-default locations

Package Name	Package Description	Rules for installation to a non-default location
ppu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (PPU)	Any location. For the remainder of this document, the name <i>xlmass_path</i> is used to refer to this location.
spu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (SPU)	Any location. For the remainder of this document, the name <i>xlmass_path</i> is used to refer to this location.
cell-xlc-rte	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 <i>xlrte_path</i> is used to refer to this location.
cell-xlc-rte-lnk	IBM XL C/C++ runtime links package	
cell-xlc-lic	IBM XL C/C++ license package	Any location. For the remainder of this document, the name <i>lic_path</i> is used to refer to this location.

Table 11. Rules for installing packages to multiple, non-default locations (continued)

Package Name	Package Description	Rules for installation to a non-default location
cell-xlc-lib	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.
cell-xlc-cmp	IBM XL C/C++ compiler package	
cell-xlc-omp	IBM XL C/C++ OpenMP package	
cell-xlc-help	IBM XL C/C++ Help html documentation package	Any location (optional). For the remainder of this document, the name <i>doc_path</i> is used to refer to this location.
cell-xlc-man	IBM XL C/C++ compiler man pages	Any location (optional). For the remainder of this document, the name <i>manpag_path</i> is used to refer to this location.
cell-xlc-samples	IBM XL C/C++ samples package	Any location (optional). For the remainder of this document, the name <i>smppls_path</i> is used to refer to this location.

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 non-default directory:

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

where *package_installation_path* is a directory other than `/opt/ibmcmp/` and corresponds to one of the appropriate paths listed in Table 11 on page 12.

Example: Installing XL C/C++ to multiple non-default directories:

In order to avoid dependency errors during installation of IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 issue the following commands in the order given:

```
rpm -ivh ppu-xlmass.lib-5.0.0-0.rpm --prefix $MASS_path
rpm -ivh spu-xlmass.lib-5.0.0-0.rpm --prefix $MASS_path
```

IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1

```
rpm -ivh cell-xlc-rte-10.1.0-0.i386.rpm --prefix $RTEpath
rpm -ivh cell-xlc-rte-lnk-10.1.0-0.i386.rpm --prefix $RTEpath
```

```
rpm -ivh cell-xlc-lic-10.1.0-0.i386.rpm --prefix $LICpath
```

```
rpm -ivh cell-xlc-lib-10.1.0-0.i386.rpm --prefix $CMPpath
rpm -ivh cell-xlc-cmp-10.1.0-0.i386.rpm --prefix $CMPpath
rpm -ivh cell-xlc-omp-10.1.0-0.i386.rpm --prefix $CMPpath
```

Note: IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1 is a cross-compiler. Therefore, you will need to also install the runtime library package, `cell-ppuxlc-rte`, on the target system.

IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1

```
rpm -ivh cell-xlc-rte-10.1.0-0.ppc64.rpm --prefix $RTEpath
rpm -ivh cell-xlc-rte-lnk-10.1.0-0.ppc64.rpm --prefix $RTEpath

rpm -ivh cell-xlc-lic-10.1.0-0.ppc64.rpm --prefix $LICpath

rpm -ivh cell-xlc-lib-10.1.0-0.ppc64.rpm --prefix $CMPpath
rpm -ivh cell-xlc-cmp-10.1.0-0.ppc64.rpm --prefix $CMPpath
rpm -ivh cell-xlc-omp-10.1.0-0.ppc64.rpm --prefix $CMPpath
```

Note: IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1 is a cross-compiler. Therefore, you will need to also install the runtime library package, `cell-ppuxlc-rte`, on the target system.

The help and man product documentation packages have no dependency on other RPM packages and can be installed in any order using the following commands:

IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V10.1

```
rpm -ivh cell-xlc-samples-10.1.0-0.i386.rpm --prefix $SAMPpath
rpm -ivh cell-xlc-man-10.1.0-0.i386.rpm --prefix $MANPAGpath
rpm -ivh cell-xlc-help-10.1.0-0.i386.rpm --prefix $DOCpath
```

IBM XL C/C++ for Multicore Acceleration for Linux on Power Systems, V10.1

```
rpm -ivh cell-xlc-samples-10.1.0-0.ppc64.rpm --prefix $SAMPpath
rpm -ivh cell-xlc-man-10.1.0-0.ppc64.rpm --prefix $MANPAGpath
rpm -ivh cell-xlc-help-10.1.0-0.ppc64.rpm --prefix $DOCpath
```

Querying for RPM package summaries

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

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 a `cell-xlc-lib-10.1.0-0.ppc64.rpm` RPM file that has not yet been installed, issue the following command:

```
$ rpm --qf="%{summary}\n" -qp cell-xlc-lib-10.1.0-0.ppc64.rpm
```

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

```
IBM XL C/C++ Compiler Libraries Package (RHEL5-CBE)
```

Chapter 4. Installing an update

An update, or program temporary fix (PTF), of IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 provides a fix or multiple fixes to the product. You can download updates from the support web site: <http://www.ibm.com/software/awdtools/xlcpp/support>.

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 Multicore Acceleration for Linux, V10.1 installed on your system, you can apply the latest update.

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/ibmcmp/`, you should follow the steps outlined in the following table.

Table 12. Steps for basic installation: update installation

Task	For more information, see . . .
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 16
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none">• "Querying for installed packages" on page 28• "Testing the installation" on page 23
(Optional) If you did 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 25

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

Table 13. Steps for advanced installation: update installation

Task	For more information, see . . .
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>cell-xlc-lic</code> .	"Updating an advanced installation" on page 18
Install update packages to the non-default location.	"Installing XL C/C++ to a location separate from earlier versions" on page 11

Table 13. Steps for advanced installation: update installation (continued)

Task	For more information, see . . .
Configure the compiler.	Chapter 5, “Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 (for advanced users),” on page 19
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • “Querying for installed packages” on page 28 • “Testing the installation” on page 23
(Optional) Set up the environment to locate the invocation commands without the full path.	“Setting up the environment for the invocation commands” on page 25

Testing a PTF before you install it

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

To install a PTF to a non-default location, use the procedure described in “Installing XL C/C++ to a location separate from earlier versions” on page 11.

After you have finished testing the new PTF and have verified that you want to replace the older version with the new update, you can install the PTF to the default location by doing the following:

1. Remove the new update from the non-default location.
2. Remove the older version from the default location.

Important: Do not uninstall the cell-xlc-lic package. This package is required for the next step.

3. Re-install the new update to the default location using the `xlc_install` utility that comes with the update package.

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.”

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 Multicore Acceleration for Linux, V10.1. It is located in the root directory of the installation image.

You can use the `xlc_install` utility to update IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 if all the following conditions have been met:

- The base version of IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 is already successfully installed in the `/opt/ibmcmp/` directory.
- The update package has been uncompressed and unpacked in the `/home/root/` directory of the system.

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

- Checks for all prerequisite software packages
- Uninstalls IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 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 Multicore Acceleration for Linux, V10.1:

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

```
cd /home/root/update/xlc/mmmYYYY
```

where `mmmYYYY` is the month and year of the update shipment date. (For example, `dec2008` indicates a shipment date in December of 2008.)

2. Issue the following command:

```
./xlc_install -U
```

For additional arguments that you can specify for `xlc_install`, see “`xlc_install` options” on page 8.

You are prompted to uninstall any previously installed IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 packages.

3. Confirm that you want to proceed with uninstalling the outdated packages. You are prompted to uninstall any IBM MASS packages that were previously installed with IBM XL C/C++ for Multicore Acceleration for Linux.
4. Confirm that you want to proceed with uninstalling the existing IBM MASS packages.

Note: `xlc_install` will uninstall these packages from their existing location and reinstall them to the default location, `/opt/ibmcmp/`. Therefore, if they were previously installed in a non-default location as part of an IBM XL Fortran for Multicore Acceleration for Linux installation, you will need to run `xlf_configure` to reconfigure the IBM XL Fortran for Multicore Acceleration for Linux 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 Multicore Acceleration for Linux, V12.1 Installation Guide*.

You are presented with the licensing agreement and licensing information.

5. Accept the licensing agreement and licensing information. You are prompted to create symbolic links for the compiler invocations in the `/usr/bin/` directory.
6. Optionally, 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 25.

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

- `cbexlc`
- `cbexlc++`
- `cbexlC`
- `ppuxlc++`
- `ppuxlC`

- ppuxlC_r
 - ppuxlc++_r
 - spuxlc++
 - spuxlC
 - ppuxlc
 - ppuxlc_r
 - spuxlc
 - ppuc89
 - ppuc89_r
 - ppuc99
 - ppuc99_r
 - ppucc
 - ppucc_r
 - ppugxc
 - spugxc
 - ppugxc++
 - spugxc++
 - ppugxlC
 - spugxlC
7. If you customized the previously generated configuration file, manually edit `/opt/ibmcmp/xlc/cbe/10.1/etc/vac.cfg` to replicate those changes in the newly generated configuration file.

Updating an advanced installation

To update to an advanced installation of IBM XL C/C++ for Multicore Acceleration for Linux, V10.1, it is recommended that you first uninstall all IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 packages except the license package, `cell-xlc-lic`.

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 location conflict: two files that belong to two different PTF levels cannot be installed to the same location at the same time. See Chapter 7, “Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1,” on page 29 for instructions on how to uninstall XL C/C++.

After the old packages have been uninstalled, follow the instructions in Chapter 3, “Advanced installation,” on page 9 to install the PTF packages to the location of your choice.

Chapter 5. Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 (for advanced users)

Before you can run IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 you must configure (or re-configure) the compiler if either of the following conditions apply:

- You did not use `xlc_install` to install the compiler.
- The compiler is installed in a non-default location, or compiler components were relocated after installation.

Two configuration tools are provided with the compiler: `new_install` and `xlc_configure`, both located in the `installation_path/xlc/cbe/10.1/bin/` directory after installation.

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

- Only one version of IBM XL C/C++ for Multicore Acceleration for Linux is installed on your system.
- Only one version of the SDK GCC is installed in your system and it is installed in its default location.
- You have root or administrator privileges.
- You want to generate the configuration file in the default directory `/opt/ibmcmp/xlc/cbe/10.1/etc/`

For instructions, see “Running the `new_install` utility” on page 20.

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

- You have multiple versions of IBM XL C/C++ for Multicore Acceleration for Linux installed on your system.
- You receive an error from the `new_install` command. (See Chapter 8, “Troubleshooting the installation and configuration,” on page 31.)
- You want the generated configuration file to be placed in a non-default location.

Note: The default location is

`/opt/ibmcmp/xlc/cbe/10.1/etc/`

- You have multiple versions of SDK GCC installed on your system and you need to specify which SDK GCC version you would like to reference in the configuration file.

Note: If you configure the compiler using `xlc_configure`, your output configuration file, `vac.cfg`, 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 20.

Running the new_install utility

Use the `new_install` utility to configure an advanced installation if only one version of XL C/C++ and one version of the SDK GCC are installed on your system.

The `new_install` utility does the following:

- Backs up any existing configuration file.
- Installs the license file.
- Generates the configuration file in the default location `/opt/ibmcmp/xlc/cbe/10.1/etc/vac.cfg`.

To run the `new_install` utility:

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

```
cd installation_path/xlc/cbe/10.1/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/ibmcmp/`. If the compiler has been installed to multiple, non-default locations, the `new_install` utility can be found in `$CMPpath/xlc/cbe/10.1/bin/`

2. Run the following command:

```
./new_install
```

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 the SDK GCC are installed on your system.

To run the `xlc_configure` utility:

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

```
cd installation_path/xlc/cbe/10.1/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/ibmcmp/`.

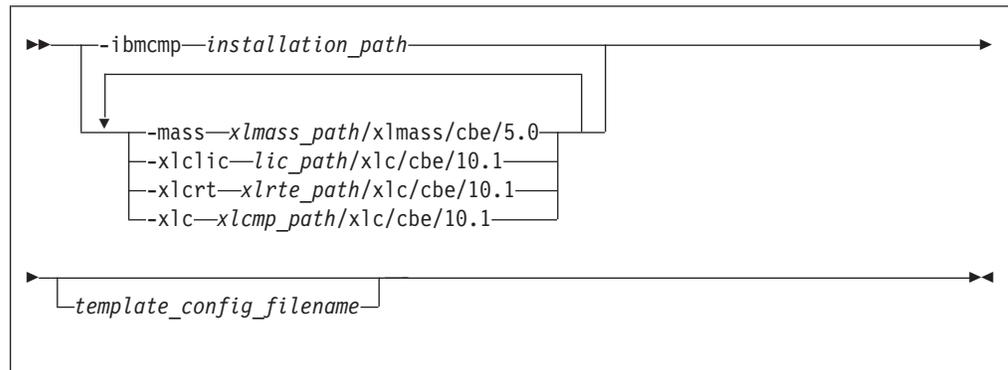
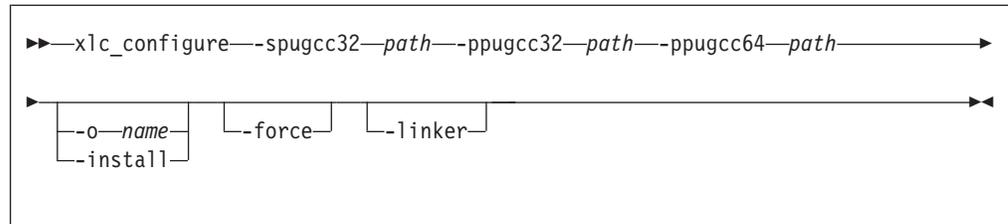
2. Run the following command:

```
./xlc_configure options
```

See the following section for required arguments to the `xlc_configure` command.

xlc_configure options

The `xlc_configure` command has the following syntax:



where:

-h Displays the xlc_configure options help page.

-ppugcc32 path

Path where the PPU toolchain GCC is installed. In the configuration file, the gcc_path attribute is set to equal this path. For example, if the PPU-GCC command is `/usr/bin/ppu32-gcc`, you would specify:

```
-ppugcc32 /usr
```

-ppugcc64 path

Specifies the path where the 64-bit PPU toolchain GCC is installed. In the configuration file, the gcc_path_64 attribute is set to equal this path. For example, if the 64-bit PPU-GCC command is `/usr/bin/ppu-gcc`, you would specify:

```
-ppugcc64 /usr
```

-spugcc32 path

Specifies the path where the SPU toolchain GCC is installed. In the configuration file, the SPU default gcc_path attribute is set to equal this path. For example, if the SPU-GCC command is `/usr/bin/spu-gcc`, you would specify:

```
-spugcc32 /usr
```

-linker path

Specifies the path where the Software Development Kit linker script, `elf32_spu.xc`, is located. By default, the path is `[-spugcc32 <path>]/spu/lib/ldscripts/`

-o file_name

Specifies the name of the configuration file to generate. By default, output is written to the display only.

-install

Generates the configuration file as `/opt/ibmcmp/xlc/cbe/10.1/etc/vac.cfg`. By default, output is written to the display only.

-force Forces the `xlc_configure` utility to overwrite any existing output file with the same name and path as that 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 Multicore Acceleration for Linux packages are installed (if all packages are installed in the same path). By default, the path is `/opt/ibmcmp/`.

-mass *xlmass_path/xlmass/cbe/5.0/*

Specifies the path where the `xlmass` package is installed. By default, the full path is `/opt/ibmcmp/xlmass/cbe/5.0/`.

-xlcllc *lic_path/xlc/cbe/10.1/*

Specifies the path where the `cell-xlc-lic` package is installed. By default, the full path is `/opt/ibmcmp/xlc/cbe/10.1/`.

-xlcrte *xlrte_path/xlc/cbe/10.1/*

Specifies the path where the `cell-xlc-rte` and `cell-xlc-rte-lnk` packages are installed. By default, the full path is `/opt/ibmcmp/xlc/cbe/10.1/`.

-xlc *xlcmp_path/xlc/cbe/10.1/*

Specifies the path where the `cell-xlc-cmp`, `cell-xlc-omp` and `cell-xlc-lib` packages are installed. By default, the full path is `/opt/ibmcmp/xlc/cbe/10.1/`.

template_config_file_name

The input file that is used to construct the configuration file. By default, this is `/opt/ibmcmp/xlc/cbe/10.1/etc/vac.base.cfg`. If you relocated the `cell-xlc-cmp` package to *xlcmp_path* but want to use the default template, specify:

`xlcmp_path/xlc/cbe/10.1/etc/vac.base.cfg`

Chapter 6. After installing IBM XL C/C++ for Multicore Acceleration for Linux, V10.1

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

- Testing the installation: This section applies to all users.
- Enabling the man pages: This section applies to all users.
- Enabling the error messages: This section applies only to users whose system uses a locale or language encoding other than en_US.
- Setting up the environment for the invocation commands: This section applies only to users who did not use `xlc_install` to install or update the product, or who did not create symbolic links during the installation process with `xlc_install`.
- Querying for installed packages: This section applies to all users.

Testing the installation

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

Basic example: Creating and running “Hello World”

Either prefix **PPU** or **SPU** may be used. **PPU** will be used for this example.

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

```
ppuxlc hello.c -o hello
```

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

```
/opt/ibmcmp/xlc/cbe/10.1/bin/ppuxlc hello.c -o hello
```

3. Run the program on the target system by entering the following command:

```
./hello
```

The result should be "Hello World!".

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

```
echo $?
```

The result should be 0.

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

```

#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}

```

6. Compile the program:

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

```
ppuxlc++ hello.cpp -o hello
```

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

```
/opt/ibmcmp/xlc/cbe/10.1/bin/ppuxlc++ hello.cpp -o hello
```

7. Run the program on the target system:

```
./hello
```

The result should be "Hello World!".

8. Check the exit code of the program:

```
echo $?
```

The result should be "0".

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 Multicore Acceleration for Linux, V10.1 man pages support the following locales:

- en_US
- en_US.utf8

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 you are using.

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

```
export MANPATH=installation_path/xlc/cbe/10.1/man/LANG:$MANPATH
```

where *LANG* is either of the language locales listed above.

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

```
setenv MANPATH installation_path/xlc/cbe/10.1/man/LANG:$MANPATH
```

where *installation_path* is the location where you have installed the XL C/C++ packages (by default, this is /opt/ibmcmp/) and where *LANG* is either of the language locales listed above.

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

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 2), you must set the NLSPATH environment variable so that the compiler and runtime functions can find the appropriate message catalogs following 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:xlcmp_path/x1c/cbe/10.1/msg/%L/%N
```

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

```
setenv NLSPATH $NLSPATH:xlcmp_path/x1c/cbe/10.1/msg/%L/%N
```

where:

- *xlcmp_path* is the installation location of the IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 compiler packages. By default this is `/opt/ibmcmp/`.

Note: To set this variable in the Bourne, Korn, or BASH shell so that it applies to all users, add the commands to the file `/etc/profile`. To set it for a specific user only, add the commands to the file `.profile` in the user’s home directory. To set this variable in the C shell, add the commands to the file `/etc/csh.cshrc`. To set it for a specific user only, add the commands 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 `x1c_install` utility to install the compiler and you elected 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 elect 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” on page 26.

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

To use IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 commands without typing the complete path, you can add the location of the compiler invocations to the PATH environment variable.

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

```
export PATH=$PATH:installation_path/x1c/cbe/10.1/bin/
```

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

```
setenv PATH $PATH:installation_path/xlc/cbe/10.1/bin/
```

where *installation_path* is the location where you have installed the compiler packages (by default, this is `/opt/ibmcmp/`).

Note: To set this variable in the Bourne, Korn, or BASH shell so that it applies to all users, add the commands to the file `/etc/profile`. To set it for a specific user only, add the commands to the file `.profile` in the user's home directory. To set this variable in the C shell, add the commands to the file `/etc/csh.cshrc`. To set it for a specific user only, add the commands 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 contained in the `installation_path/xlc/cbe/10.1/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:

- `cbexlc`
- `cbexlc++`
- `cbexlC`
- `ppuxlc++`
- `ppuxlC`
- `ppuxlC_r`
- `ppuxlc++_r`
- `spuxlc++`
- `spuxlC`
- `ppuxlc`
- `ppuxlc_r`
- `spuxlc`
- `ppuc89`
- `ppuc89_r`
- `ppuc99`
- `ppuc99_r`
- `ppucc`
- `ppucc_r`
- `ppugxc`
- `spugxc`
- `ppugxc++`
- `spugxc++`
- `ppugxlC`
- `spugxlC`

Use the following command to create a symbolic link:

```
ln -s installation_path/xlc/cbe/10.1/bin/invocation /usr/bin/invocation
```

where:

- *installation_path* is the location where you have installed the compiler packages (by default, this is `/opt/ibmcmp/`).
- *invocation* is one of the compiler invocations (such as `ppuxlc`) in *installation_path*/`xl/cbe/10.1/bin/`.

Basic example: Creating a symbolic link to a compiler invocation

This example assumes that all of IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 is installed in the default location `/opt/ibmcmp/`.

```
ln -s /opt/ibmcmp/xlc/cbe/10.1/bin/ppuxlc /usr/bin/ppuxlc
```

Enabling IBM Tivoli License Compliance Manager

IBM Tivoli® License Compliance Manager (ITLCM) is a Web-based solution that can help you manage software usage metering and license allocation services on supported systems. In general, ITLCM recognizes and monitors the products that are installed and in use on your system.

Note: ITLCM is not part of the IBM XL C/C++ for Multicore Acceleration for Linux offering, and must be ordered and installed separately.

Once installed and activated, ITLCM scans your system for product inventory signatures that indicate whether a given product is installed on your system. ITLCM also identifies the version, release, and modification levels of the product. Inventory signature files are not updated after a PTF update package is installed.

ITLCM agent, once deployed on a computer, collects information about the level and duration of IBM XL C/C++ for Multicore Acceleration for Linux compiler use on that computer and can generate reports based on the information it collects.

If IBM XL C/C++ for Multicore Acceleration for Linux is installed in the default location, the signature files are in the `/opt/ibmcmp/xlc/cbe/10.1/` directory. Otherwise, if IBM XL C/C++ for Multicore Acceleration for Linux is installed in a non-default location, the signature files are in the `$target_dir/xlc/10.1/` directory, where *\$target_dir* is the target directory for installation specified by `--prefix` option of the non-default installation command.

For more information about IBM Tivoli License Compliance Manager see: <http://www.ibm.com/software/tivoli/products/license-mgr/>.

Accessing the local documentation

Help for IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 is available in PDF format only. Manual pages for the compiler invocation commands and other command utilities are also included.

Viewing the PDF documentation

PDF versions of the IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 product manuals are available in the `/doc/en_US/pdf/` directory of the installation media (either product CD or electronic package).

After default installation, the PDF documentation can be found in the `/opt/ibmcmp/xlc/cbe/10.1/doc/en_US/pdf/` directory. For non-default installations, the PDF documentation is located in the `$installation_path/xlc/cbe/10.1/doc/en_US/pdf/` directory.

Querying for installed packages

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

To query for an individual package, issue a command such as the following:

```
rpm -q cell-xlc-cmp
```

The result should be:

```
cell-xlc-cmp-V.R.M-F
```

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

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

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

```
rpm -qa | grep -e cell-xlc -e xlmass
```

The result should be a list that contains all of the packages listed in Table 5 on page 2. If none of the packages listed in the table was properly installed, there will be no output from the command.

Chapter 7. Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1

IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 does not provide a standalone uninstallation tool. You must use the Linux rpm utility to uninstall IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.

Note:

- You must have root user access to uninstall the compiler.
- Whenever you uninstall a package, specify the *V.R.M-F* (Version.Release.Modification-Fix level) of the package. For information on how to determine the package's *V.R.M-F*, see "Querying for installed packages" on page 28.
- Always uninstall packages in the reverse order of that in which they were installed. In other words, the last package installed is the first package that you remove. *Exception:* The help files and man documentation do not have any package dependencies. You can remove them in any order.
- You cannot uninstall packages that are required by other packages.
- The uninstallation commands will not remove any configuration files that were generated by `new_install` or `xlc_configure`.

Example: Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V10.1

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

In this example:

- The compiler packages have a *V.R.M-F* of 10.1.0-0
- The IBM MASS library package has a *V.R.M-F* of 5.0.0-0.

Note:

IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 is a cross-compiler and you will have needed to install the `cell-xlc-rte` rpm package on the target system. Therefore, you will also uninstall the runtime library package, `cell-xlc-rte`, from the target system.

To uninstall IBM XL C/C++ for Multicore Acceleration for Linux, V10.1 issue the following commands, in the same order:

```
rpm -e cell-xlc-cmp-10.1.0-0
rpm -e cell-xlc-omp-10.1.0-0
rpm -e cell-xlc-lib-10.1.0-0

rpm -e cell-xlc-lic-10.1.0-0

rpm -e cell-xlc-rte-lnk-10.1.0-0
rpm -e cell-xlc-rte-10.1.0-0

rpm -e spu-xlmass.lib-5.0.0-0
rpm -e ppu-xlmass.lib-5.0.0-0
```

If installed, the following packages can be uninstalled in any order. Uninstalling them will not affect the performance of XL Fortran for Multicore Acceleration for Linux:

```
rpm -e cell-xlc-man-10.1.0-0  
rpm -e cell-xlc-help-10.1.0-0  
rpm -e cell-xlc-samples-10.1.0-0
```

Chapter 8. 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 will stay 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 may encounter when you install and configure IBM XL C/C++ for Multicore Acceleration for Linux, V10.1.

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 8.

`rpmlocation_path` does not contain . . .

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

Ensure you have all of the packages listed in Table 5 on page 2 in the path before running the `xlc_install` utility again. 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 8.

Could not determine location of 32-bit or 64-bit SDK GCC (RHEL 5.2)

Scenario

You are running either the `new_install` or the `xlC_configure` utility to configure the compiler on a computer running RHEL 5.2 when you get at least one of the following error messages:

ERROR: File "<path>/ppu32-gcc" not found

ERROR: File "<path>/ppu32-g++" not found

ERROR: File "<path>/ppu-gcc" not found

ERROR: File "<path>/ppu-g++" not found

ERROR: File "<path>/spu-gcc" not found

ERROR: File "<path>/spu-g++" not found

Note: <path> is the location specified by the `-ppugcc32`, `-ppugcc64`, and `-spugcc32` when invoking `xlC_configure`.

Explanation

There are four SDK GCC rpms required. At least one is not installed in the corresponding directory specified by the `-ppugcc32`, `-ppugcc64`, and `-spugcc32`:

- `ppu-gcc`
- `ppu-gcc-c++`
- `spu-gcc`
- `spu-gcc-c++`

To check if the RPMs are already installed:

```
rpm -qa | grep <rpm name>
```

Action

- If the RPMs are installed, please make sure the installation path of the RPMs are correctly specified with `-ppugcc32`, `-ppugcc64`, and `-spugcc32`.
- If the RPMs are not installed, please install them. They are available with the IBM Software Development Kit (SDK) for Multicore Acceleration.

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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 wish 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:

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario L6G 1C7
Canada

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. 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 IBM's application programming interfaces.

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, 2008. All rights reserved.

Trademarks and service marks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

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

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

Index

A

- advanced installation
 - example 13
 - overview 9
 - procedures 10
- advanced user, described v
- available space, determining 4

B

- basic example, described x
- basic installation
 - overview 5
 - procedure 6
- basic user, described v

C

- co-residency
 - compilers 10
- configuration files
 - backup 19
 - customizing 19, 20
 - default 22
 - editing 20
 - generation 8, 21
 - modifying 19, 20
 - multiple 19
 - overwriting 22
 - renaming 22
 - security 29

D

- debugging 31
- default installation 5

E

- environment variable 25
- error messages 25

G

- GNU packages
 - required 3
 - verifying 4

H

- hard disk space
 - required 3
 - verifying 4
- hardware requirements 3
- help 27
 - displaying 8
 - man pages 24
 - pdf 27

I

- installation
 - multiple versions 9, 10
 - options 8
 - package location 12
 - specialized 12
 - testing 23
- installation CD 1
- installation image 1
- installation log 31
- installation utility
 - options 8
 - using 15
- invocation commands 25
- ITLCM 27

L

- languages 2
- License Compliance Manager, IBM
 - Tivoli 27
- locales 2

M

- man pages 24

N

- national language support 2
- non-default installation 9

O

- operating systems, supported 3

P

- packages
 - installation 1
 - not found 31
 - querying 28
- PATH environment variable 25
- Perl packages
 - required 3
 - verifying 4
- preinstallation 1
- PTF 15

Q

- querying
 - installed packages 28
 - RPM file summaries 14

R

- Red Hat Package Manager (RPM) 1

S

- software requirements 3
- symbolic links 6
 - creating 26

T

- temporary files
 - high optimization levels 3
 - installation log 31
- testing
 - installation 23
 - PTF 16
- Tivoli License Compliance Manager 27

U

- uninstallation
 - example 29
 - overview 29
- updates
 - applying 16
 - installation utility option 8
 - overview 15
 - prerequisites 16
 - steps 16
 - to a non-default location 16
 - trying out 12, 16

V

- verifying
 - GNU and Perl packages 4
 - hard disk space 4



Program Number: 5724-T42 & 5724-T43

Printed in USA

GC23-8574-00

