IBM XL C/C++ Advanced Edition V8.0 for Linux

Installation Guide

GC09-8017-00

IBM XL C/C++ Advanced Edition V8.0 for Linux

Installation Guide

GC09-8017-00

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

First Edition (November 2005)

This edition applies to IBM XL C/C++ Advanced Edition V8.0 for Linux (5724-M16) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send them to compinfo@ca.ibm.com. Be sure to include your e-mail address if you want a reply. Include the title and order number of this book, and the page number or topic related to your comment.

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

Contents

About this document

This document contains essential information about installing XL C/C++ Advanced Edition V8.0 for Linux. Please read it carefully before installing this product. Be sure to read the README file on the CD, which contains the most current information about the product. After you install the product, you can find the README file in the *xlcmp_path*/vac/8.0/ directory.

Note: *xlcmp_path* is the location of the compiler on your system. If you have installed the compiler in the default location, *xlcmp_path* is /opt/ibmcmp/.

Who should read this document

This document is intended for anyone with responsibility for installing XL C/C++ Advanced Edition V8.0 for Linux.

Most customers will use the basic installation method, which provides guidance during the installation process. Most chapters in this document are directed toward these customers. *Basic examples* are tailored to reflect, as much as possible, the procedures used by users as they perform a basic installation.

This document also addresses the needs of customers who want to use multiple versions of XL C/C++ Advanced Edition for Linux on a single system. If you are one of those customers, you should be very familiar with customized software installation processes and with the system on which you are installing the product. In this document, you are referred to as an *advanced user*. The additional information that you will need is labeled “for advanced users”.

How to use this document

Use this document to help you with typical installation requirements, processes, and messages.

For highly specialized installation scenarios that are outside the scope of this document, refer to Technotes at
http://publib.boulder.ibm.com/infocenter/lnxpcomp/index.jsp

How this document is organized

This book is organized to reflect the pre-installation, installation, post-installation, and troubleshooting phases of an installation of XL C/C++ Advanced Edition V8.0 for Linux.

Table 1. Phases of a basic XL C/C++ V8.0 installation

Phase	Chapters	Customer segment
Pre-installation	Chapter 1, “Before installing XL C/C++ V8.0,” on page 1	All customers

Table 1. Phases of a basic XL C/C++ V8.0 installation (continued)

Phase	Chapters	Customer segment
Installation	Chapter 2, "Basic installation," on page 7	Customers with the following needs: • Want to use the simplest, most direct installation process • Do not have any special requirements, such as use of multiple versions of XL C/C++ compilers
	Chapter 3, "Updating XL C/C++ V8.0," on page 11	Customers who want to update XL C/C++ V8.0 to the next fix level
	• Chapter 6, "Installing XL C/C++ in a non-default location (for advanced users)," on page 21 • Chapter 7, "Installing multiple versions of XL C/C++ on the same system (for advanced users)," on page 25	Use the information in these chapters if: • You want to install the compiler in a non-default location. • You want to have multiple XL C/C++ versions on the same system. .
Post-installation	Chapter 4, "After installing XL C/C++ V8.0," on page 13	All customers
Product removal	Chapter 5, "Uninstalling XL C/C++," on page 19	Any customer who needs to remove an XL C/C++ compiler from the computer
Troubleshooting	Chapter 9, "Troubleshooting the installation and configuration," on page 31	Any customer who needs to know how to respond to an error message or unexpected results during the installation or configuration of XL C/C++ V8.0

Conventions used in this document

Typographical conventions

The following table explains the typographical conventions used in this document.

Table 2. Typographical conventions

Typeface	Indicates	Example
bold	Commands, utility names and other executable names that are embedded in a paragraph	The **xlc_install** installation utility will uninstall any previously installed compilers before installing XL C/C++ V8.0.

Table 2. Typographical conventions (continued)

Typeface	Indicates	Example
italics	• For parameters or variables whose actual names or values either depend on the situation or are to be supplied by the user • To introduce new terms or to emphasize the text • To identify the title of another document	Example of programming variable: *xlcmp_path* is the location of the compiler on your system. If the compiler is installed in the default location, *xlcmp_path* is /opt/ibmcmp/. Example of emphasized text: Use the **xlc_install** utility to install the compiler *only if* you are: • Installing the compiler to the default location • Agreeing to remove any previously installed XL C/C++ Advanced Edition for Linux compiler
`monospace`	Examples of program code or commands.	For example, to see whether gcc-c++ is installed, query for the gcc-c++ package as follows: `rpm -qa | grep gcc-c++`

How to read syntax diagrams

Throughout this document, diagrams illustrate XL C/C++ syntax. You can use this section to help you to interpret the syntax 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.

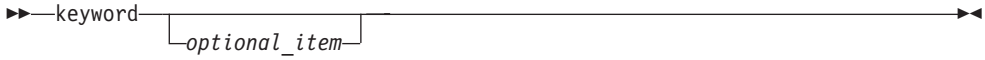
  The —▸ symbol indicates that the command, directive, or statement syntax is continued on the next line.

  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.

  Diagrams of syntactical units other than complete commands, directives, or statements start with the ▸— symbol and end with the —▸ symbol.
- Required items appear 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.

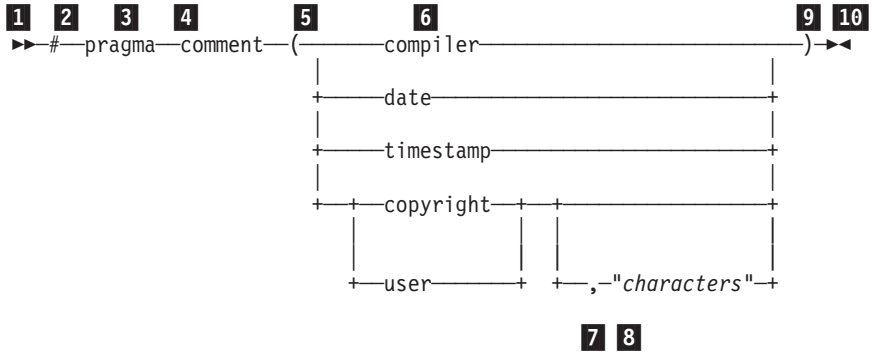
The item that is the default is shown above the main path.

- An arrow returning to the left above the main line indicates an item that can be repeated.

  A repeat arrow above a stack indicates that you can make more than one choice from the stacked items, or repeat a single choice.
- Keywords are shown in non-italic letters and should be entered exactly as shown (for example, `extern`).

  Variables are shown in italicized lowercase letters (for example, *identifier*). 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.

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



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.

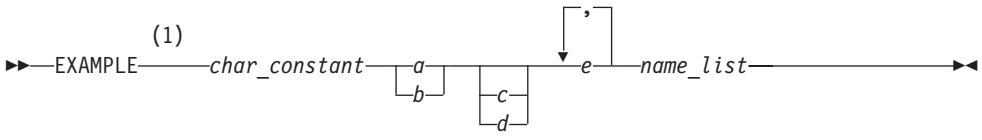
10 This is the end of the syntax diagram.

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")
```

Sample syntax diagram

The following is an example of a syntax diagram with an interpretation:



Notes:

1 IBM Extension

Interpret the diagram as follows:

- Enter the keyword EXAMPLE.
- EXAMPLE is an IBM® extension.
- Enter a value for *char_constant*.
- Enter a value for *a* or *b*, but not for both.
- Optionally, enter a value for *c* or *d*.
- Enter at least one value for *e*. If you enter more than one value, you must put a comma between each.
- Enter the value of at least one *name* for *name_list*. If you enter more than one value, you must put a comma between each. (The *_list* syntax is equivalent to the previous syntax for *e*.)

Examples and basic examples

The examples in this document are labelled as either “Example” or “Basic example”. *Basic examples* are intended to document a procedure as it would be performed during a basic installation, with little or no modification.

Related information

IBM XL C/C++ Advanced Edition V8.0 for Linux publications

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

- README files

README files contain late-breaking information, including changes and corrections to the product documentation. README files are located by default in the /opt/ibmcmp/vacpp/8.0/ 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 this guide.

- Information center

  The information center of searchable HTML files can be launched on a network and accessed remotely or locally. Instructions for installing and accessing the information center are provided in this guide. The information center is also viewable on the web at:

  http://publib.boulder.ibm.com/infocenter/lnxpcomp/index.jsp.

- PDF documents

  PDF documents are located by default in the /opt/ibmcmp/vacpp/8.0/doc/*language*/pdf/ directory, where *language* is one of en_US, zh_CN, or ja_JP. The PDFs are also available on the web at:

  `http://publib.boulder.ibm.com/infocenter/lnxpcomp/index.jsp`

  In addition to this document, the following files comprise the full set of XL C/C++ Advanced Edition V8.0 for Linux product manuals:

Table 3. XL C/C++ Advanced Edition V8.0 for Linux PDF files

Document title	PDF file name	Description
Getting Started with XL C/C++ Advanced Edition for Linux, SC09-8016-00	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.
XL C/C++ Advanced Edition for Linux Compiler Reference, SC09-8013-00	compiler.pdf	Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions, including those used for parallel processing.
XL C/C++ Advanced Edition for Linux Language Reference, SC09-8016-00	language.pdf	Contains information about the programming languages, as supported by IBM, including language extensions for portability and conformance to non-proprietary standards.
XL C/C++ Advanced Edition for Linux Programming Guide, SC09-8014-00	proguide.pdf	Contains information on advanced programming topics, such as application porting, interlanguage calls with Fortran, library development, application optimization and parallelization, and the XL C/C++ Advanced Edition for Linux high-performance libraries.

These PDF files are viewable and printable from Adobe Reader. If you do not have the Adobe Reader installed, you can download it from http://www.adobe.com.

Additional documentation

More documentation related to XL C/C++ Advanced Edition for Linux, including redbooks, whitepapers, tutorials, and other articles, is available on the web at:

`http://www.ibm.com/software/awdtools/xlcpp/library`

Technical support

Additional technical support is available from the XL C/C++ Support page. This page provides a portal with search capabilities to a large selection of technical support FAQs and other support documents. You can find the XL C/C++ Advanced Edition for Linux Support page on the web at

`http://www.ibm.com/software/awdtools/xlcpp/`

If you cannot find what you need, you can e-mail:

`compinfo@ca.ibm.com`

For the latest information about XL C/C++ Advanced Edition for Linux, visit the product information site at
http://publib.boulder.ibm.com/infocenter/lnxpcomp/index.jsp

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 document or any other XL C/C++ documentation, send your comments by e-mail to:

`compinfo@ca.ibm.com`

Be sure to include the name of the document, the part number of the document, the version and release of XL C/C++ Advanced Edition for Linux, 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 XL C/C++ V8.0

Before you install any XL compiler:

- Familiarize yourself with the installation medium and packages. (See “The installation image.”)
- Determine the tasks that you need to perform. (See “Determining the tasks you need to perform” on page 2.)

The installation image

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

The image includes:

- A set of RPM packages for the XL C/C++ Advanced Edition V8.0 for Linux. See “The installation packages.”
- An installation tool to install and configure XL C/C++ Advanced Edition V8.0 for Linux for the basic installation scenario. See Chapter 2, “Basic installation,” on page 7.
- Message catalogs for each supported national language. See “National language support” on page 2.

The installation packages

Table 4 describes the packages that are supplied with the installation image. You can use the **rpm** utility to review packages.

Note: If you are performing a basic installation, you can disregard the rules for installation to a non-default location. For information about using these rules, see “Installing XL C/C++ V8.0 to multiple non-default locations” on page 22.

Table 4. XL C/C++ packages

Package Name	Package Description	Rules for installation to a non-default location
xlsmp.msg.rte	XL SMP runtime messages	All SMP packages must be installed in the same location.
xlsmp.rte	XL SMP runtime dynamic libraries	
xlsmp.lib	XL SMP runtime static libraries	
xlmass.lib	IBM Mathematical Acceleration Subsystem (MASS) libraries	Any location
vacpp.rte	XL C/C++ runtime environment	All XL C/C++ runtime packages must be installed in the same location.
vacpp.rte.lnk	XL C/C++ runtime environment links	
vac.lic	XL C/C++ license	Any location
vac.lib	XL C compiler libraries	All XL C/C++ compiler and library packages must be installed in the same location.
vacpp.lib	XL C++ compiler libraries	
vac.cmp	XL C compiler files	
vacpp.cmp	XL C/C++ compiler files	
vacpp.samples	XL C/C++ samples	Any location (optional).
vacpp.help	XL C/C++ documentation	Any location (optional).

National language support

English is the default national language. Following installation, you can set the NLSPATH so that messages are displayed in a different language. See “Enabling the XL C/C++ V8.0 error messages” on page 15.

The XL C/C++ V8.0 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

The XL C/C++ V8.0 man pages support the following locales:

- en_US
- en_US.utf8
- ja_JP
- ja_JP.eucjp

Determining the tasks you need to perform

You can use the tables in this section to help you find the information you need as you install and configure the product.

Basic pre-installation checklist

Table 5 lists tasks that you will, or might, need to perform *before* you actually install the product. Use the checklist presented in Table 5 if you are:

- Maintaining a single version of XL C/C++ on a computer.
- Either upgrading an existing XL C/C++ compiler installed in the default location or installing a new XL C/C++ compiler to the default location.

Table 5. Basic pre-installation checklist

Check off	Task	For more information, see . . .
	Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system.
	Ensure that all required and desired packages are installed in the default location.	“Verifying that the required GNU, Perl, Java, and browser packages are installed” on page 4
	Ensure that all system prerequisites are satisfied.	• “System prerequisites” on page 3 • “Verifying that there is enough space” on page 5

Basic post-installation checklist

Table 6 on page 3 lists tasks that you will, or might, need to perform *after* a basic installation.

Table 6. Basic post-installation checklist

Check off	Task	For more information, see . . .
	Confirm that packages were successfully installed.	"Querying for installed packages" on page 13
	Enable the short compiler invocation commands (optional).	"Setting up the environment for the invocation commands" on page 17
	Enable the man pages.	"Enabling the XL C/C++ V8.0 man pages" on page 14
	Set the correct NLSPATH (if not done during installation).	"Enabling the XL C/C++ V8.0 error messages" on page 15
	Configure the compiler (if not done during installation).	Chapter 8, "Configuring XL C/C++ V8.0," on page 27
	Test the installation.	"Testing the installation" on page 13

System prerequisites

Before you install XL C/C++ V8.0, ensure that the following requirements are met:

- **Operating system:** A supported Linux® distribution:
  - Red Hat Enterprise Linux AS 4 Update 1 (RHEL4) for IBM POWER
  - SUSE LINUX Enterprise Server 9 Service Pack 1 (SLES9) for IBM POWER
- **Hardware:** a system that is supported by your Linux distribution:
  - If your operating system is RHEL4 or SLES9, you can use any of the following hardware platforms:
    - IBM eServer™ OpenPower™ system
    - IBM eServer PowerPC® system
    - IBM eServer BladeCenter™ JS20 system
    - IBM eServer p5 system
    - IBM System p5
    - IBM eServer pSeries® system
    - IBM eServer i5 system
    - IBM eServer iSeries™ system
- **Required hard disk space:**
  - Approximately 200 MB for product packages
  - At least 2 GB for paging
  - At least 512 MB for temporary files

  Note: High levels of optimization can require more space for paging and temporary files.
- **Optional software:**
  - Frames-capable HTML browser (to access help and other web pages)
  - PDF viewer (to access documentation)

Prerequisite tasks

You must have root user access to install any release of XL C/C++.

Prerequisite tasks include:

Verifying that the required GNU, Perl, Java, and browser packages are installed

Before you can install XL C/C++ V8.0, you should verify that the required versions of GNU, Perl, and Java™ packages were installed with the operating system. For lists of the required packages for each supported operating system, see the tables in “Required GNU, Perl, and Java packages for supported operating systems.”

To enable you to configure the compiler following the installation process, or to use the documentation supplied with the compiler, you might also want to confirm that you have the packages that support those tasks. See “Packages that support specific tasks” on page 5.

Required GNU, Perl, and Java packages for supported operating systems

The GNU, Java, and operating system packages that must be installed before you install the compiler depend on which Linux operating system you are using.

Table 7. Required GNU, Perl, and Java packages for the RHEL4 operating system

Package name	Version requirements
gcc	3.4.3
gcc-c++	3.4.3
glibc	2.3.4
glibc-devel	2.3.4
libgcc	3.4.3
libstdc++	3.4.3
libstdc++-devel	3.4.3
libgcj	3.4.3
Perl	5.0 or greater **Note:** Perl V5.0 is shipped and automatically installed with the SLES9 and RHEL4 operating systems.

Table 8. Required GNU, Perl, and Java packages for the SLES9 operating system

Package name	Version requirements
gcc	3.3.3
gcc-c++	3.3.3
gcc-64bit	9
glibc	2.3.3
glibc-64bit	9
glibc-devel-64bit	9
libgcc	3.3.3
libgcc-64bit	9
libstdc++	3.3.3

Table 8. Required GNU, Perl, and Java packages for the SLES9 operating system (continued)

libstdc++-64bit	9
libstdc++-devel-64bit	9
Perl	5.0 or greater **Note:** Perl V5.0 is shipped and automatically installed with the SLES9 and RHEL4 operating systems.
java2	1.3.1
java2–jre	1.3.1

Packages that support specific tasks

If you are installing the documentation that is packaged with XL C/C++ V8.0, you will need the graphical desktop environments (such as K Desktop Environment or Gnome) that support web browsers and PDF viewers.

Command to verify installation of packages

You can use the following command to verify that the correct versions of the required packages are installed:

```
rpm -qa | grep package_name
```

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 3.3.3 is installed, you will get a result similar to the following output:

```
gcc-c++-3.3.3-43.24
```

Note: On RHEL4, both 32-bit and 64-bit `glibc-devel` and `libstdc++-devel` packages are required. To ensure that these packages are available before you install the compiler, follow the instruction in “Could not determine location of 32-bit or 64-bit GCC (RHEL4 only)” on page 31. Do not use the example in this section to verify that these packages are installed because the names of the packages on RHEL4 are structured so that they have the same name regardless whether they are 32-bit or 64-bit packages. As a result, the output does not indicate whether 32-bit, 64-bit or both packages are installed.

Verifying that there is enough space

XL C/C++ V8.0 requires about 200 MB of hard disk storage space. This amount accommodates the optional samples and documentation that are shipped with the product.

Note: Compiling at higher levels of optimization can require large amounts of temporary disk space.

Command to determine the amount of available space

You can use the following command to determine the amount of space that is available in any path:

```
df -h installation_path
```

Basic installation example: Determining the amount of space in the default installation path

You can use the following command to determine the amount of space available in the default installation location:

```
df -h /opt
```

Chapter 2. Basic installation

The XL C/C++ installation uses the Red Hat Package Manager (RPM), which is the standard utility for installing Linux packages. The **rpm** utility is shipped with the operating system.

XL C/C++ provides an interactive utility that walks you through a basic installation. The **xlc_install** utility 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 **xlc_install** utility. See "Verifying that the required GNU, Perl, Java, and browser packages are installed" on page 4.

Use the **xlc_install** utility to install the compiler *only* if you are:

- Installing the compiler to the default location, which is /opt/ibmcmp/.
- Agreeing to remove any previously installed XL C/C++ compiler.

Following employment of the **xlc_install** utility, an installation log is generated. Its location is /tmp/.

If all packages are successfully installed:

- The configuration file is generated. Its location is /etc/opt/ibmcmp/vac/8.0/vac.cfg. Any previously generated configuration file is saved in the /etc/opt/ibmcmp/vac/8.0/ directory.
- The installation log is moved to its permanent location: /opt/ibmcmp/vac/8.0/xlc_install.log.

Installation utility 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 XL C/C++ V8.0 packages are located. The default *rpmlocation_path* is *./hostOS*/rpms, which is relative to the path of the installation tool. Therefore, the default *rpmlocation_path* is one of the following:

- ./SLES9/rpms (if you are installing to SLES9)
- ./RHEL4/rpms (if you are installing to RHEL4)

Note: For most customers, the **-rpmloc** *rpmlocation_path* option is not required in the installation invocation. If the utility is used directly from the CD or electronic image, it will automatically determine the source location of the packages.

-U Updates XL C/C++ to the *v.r.m-f* level that the installation utility version supports. For example, using the **-U** option in the **xlc_install**, utility that is provided in the 8.0.0-1 update, will update the compiler to XL C/C++ V8.0.0-1, XL SMP V1.6.0-1, and XL MASS V4.3.0-1 when you already have XL C/C++ V8.0.0-0, XL SMP V1.6.0-0, and XL MASS V4.3.0-0 installed.

-v Displays debugging information.

-vv Displays more detailed debugging information.

Installation utility constraints

Do not use the **xlc_install** utility if you want to do either of the following:

- Install XL C/C++ V8.0 in a non-default location.
- Install XL C/C++ V8.0 on a computer on which you want it to co-reside with another version of XL C/C++.

There is no stand-alone uninstallation utility. The **xlc_install** installation utility will uninstall any previously installed compilers before installing XL C/C++ V8.0. If you choose not to uninstall the previously installed compiler components, the installation process will terminate.

Installing XL C/C++ V8.0 on a clean system

It is highly recommended that you uninstall any previous versions of XL C/C++ before you install XL C/C++ V8.0. If you use the **xlc_install** utility to install the new version to the default location (/opt/ibmcmp/), it will remove the old files for you.

xlc_install will detect the presence of a previous version and will require that you uninstall it before it proceeds with the installation.

Before you use the **xlc_install** utility to install and configure XL C/C++ V8.0, use the pre-installation checklist in Table 5 on page 2 to ensure that all prerequisite conditions are satisfied.

Basic installation steps

1. Assuming that the XL C/C++ Advanced Edition V8.0 for Linux CD is mounted at the /cdrom location in the system, issue the following commands:

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

2. Complete the following steps:
   a. Confirm that you want to proceed with uninstalling the existing version.
   b. Accept the licensing agreement and licensing information.
   c. Optional: Create symbolic links for the compiler invocations of XL C/C++ V8.0 in the /usr/bin/ directory.

      Note: An alternative to this step is to add the path that contains the compiler invocations to the PATH environment variable. See “Setting the PATH environment variable to include the path to the XL C/C++ V8.0 invocations” on page 17.

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

   - **gxlc**
   - **gxlc++**
   - **gxlC**
   - **xlc**
   - **xlc++**
   - **xlC**
   - **xlc_r**
   - **xlc++_r**
   - **xlC_r**

Note: Some command links are not created in /usr/bin/, either because they might delete user-defined GCC-related invocations or because they are not compiler invocation commands. These include:

- **c89**, **c89_r**, **c99**, **c99_r**, **cc**, and **cc_r**
- **mergepdf**, **new_install**, **resetpdf**, **showpdf**, **vac_configure**

If all packages are successfully installed:

- The configuration file is generated. Its location is /etc/opt/ibmcmp/vac/8.0/vac.cfg. Any previously generated configuration file is saved in the /etc/opt/ibmcmp/vac/8.0/ directory.
- The installation log is moved to its permanent location: /opt/ibmcmp/vac/8.0/xlc_install.log.

Installing XL C/C++ V8.0 to co-reside with XL Fortran 10.1

Follow the instructions in this section in the following cases:

- You have already installed XL Fortran Advanced Edition V10.1 for Linux in the default location (/opt/ibmcmp/).
- You want both the XL Fortran compiler and XL C/C++ V8.0 to use the XL SMP and MASS libraries provided with XL C/C++ V8.0.

Note: For information about installing XL Fortran V10.1, refer to the *XL Fortran Advanced Edition V10.1 for Linux Installation Guide*.

Before you use the **xlc_install** utility to install and configure XL C/C++ V8.0, use the pre-installation checklist in Table 5 on page 2 to ensure that all prerequisite conditions are satisfied.

Installation steps

After you have installed, configured, and tested XL Fortran:

1. Assuming that the XL C/C++ Advanced Edition V8.0 for Linux CD is mounted at the /cdrom location in the system, issue the following commands:

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

2. Complete the following steps:
   a. Confirm that you want to uninstall any previously installed XL SMP and MASS libraries. The installation utility will then install the XL SMP and MASS libraries provided with XL C/C++ V8.0.
   b. Accept the licensing agreement and licensing information.
   c. Optional: Create symbolic links for the compiler invocations of XL C/C++ V8.0 in the /usr/bin/ directory.

      Note: An alternative to this step is to add the path that contains the compiler invocations to the PATH environment variable. See “Setting the PATH environment variable to include the path to the XL C/C++ V8.0 invocations” on page 17.

   The following links are created in the /usr/bin/ directory:

   - **gxlc**
   - **gxlc++**
   - **gxlC**
   - **xlc**
   - **xlc++**

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

Note: Some command links are not created in /usr/bin/, either because it might delete user-defined GCC-related invocations or because they are not compiler invocation commands. These include:

- **c89**, **c89_r**, **c99**, **c99_r**, **cc**, and **cc_r**
- **mergepdf**, **new_install**, **resetpdf**, **showpdf**, **vac_configure**

If all packages are successfully installed:

- The configuration file is generated. Its location is /etc/opt/ibmcmp/vac/8.0/vac.cfg. Any previously generated configuration file is saved in the /etc/opt/ibmcmp/vac/8.0/ directory.
- The installation log is moved to its permanent location: /opt/ibmcmp/vac/8.0/xlc_install.log.

Chapter 3. Updating XL C/C++ V8.0

An update of XL C/C++ V8.0 provides a fix or multiple fixes to the product.

Every update package comes in a tar (or compressed) format and includes a version of the **xlc_install** utility that is customized to install only the upgrade that accompanies it. That is why the most recent version of the product must be installed on the computer before you run the **xlc_install** utility that comes with the update package. You cannot apply an update unless a version of XL C/C++ V8.0 is already successfully installed on the computer.

If you have any version (including an earlier update) of XL C/C++ V8.0, you can apply the latest update. For instructions, see “Steps to update a basic installation of XL C/C++ V8.0” or “Steps to update an installation of XL C/C++ V8.0 in a non-default location (for advanced users)” on page 12.

If you want to try out a new update of the compiler before you remove the existing one from the system, you must install the new update to a non-default location. After you have verified that you want to use the new update, you can use the **xlc_install** utility that comes with the update package to perform the following tasks:

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

   Note: Do not uninstall the vac.lic package because this package is required when you perform the next step.
3. Install the new update to the default location.

For instructions to use the **xlc_install** utility to install the new update, see “Steps to update a basic installation of XL C/C++ V8.0.”

Steps to update a basic installation of XL C/C++ V8.0

You can use the **xlc_install** utility to update XL C/C++ V8.0 when all the following conditions have been met:

- The base version of XL C/C++ V8.0 is already successfully installed in the /opt/ibmcmp/ directory.
- All prerequisite conditions described in Table 5 on page 2 have been met.
- The update package (in tar.gz format) has been uncompressed and unpacked in the /home/root/ directory of the system. You can download updates from the support web site:

  `http://www.ibm.com/software/awdtools/xlcpp/`

1. Issue the following command:

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

   where *mmmYYYY* is the month and year of the PTF shipment date. (For example, `jun2005` indicates a shipment date in June of 2005.)
2. Issue the following command:

   ```
   # ./xlc_install -U
   ```
3. Complete the following steps from the **xlc_install** utility:
   a. Confirm that you want to proceed with uninstalling the outdated packages.
   b. Accept the licensing agreement and licensing information.

c. Optional: Create symbolic links for the compiler invocations of XL C/C++ V8.0 in the /usr/bin/ directory.

Note: An alternative to this step is to add the path that contains the compiler invocations to the PATH environment variable. See “Setting the PATH environment variable to include the path to the XL C/C++ V8.0 invocations” on page 17.

When the update has been applied successfully, the **xlc_install** utility should generate the following message:

```
IBM XL C/C++ Advanced Edition for Linux has been successfully installed.
To learn about setting up your environment with XL C/C++,
refer to the XL C/C++ Advanced Edition V8.0 for Linux Installation Guide.
```

This message indicates that the following has taken place:

- The configuration file is generated. Its location is /etc/opt/ibmcmp/vac/8.0/vac.cfg. Any previously generated configuration file is saved in the /etc/opt/ibmcmp/vac/8.0/ directory.
- The installation log is moved to its permanent location: /opt/ibmcmp/vac/8.0/xlc_install.log.

4. If you customized the previously generated configuration file, manually edit /etc/opt/ibmcmp/vac/8.0/vac.cfg to replicate those changes in the newly generated configuration file.

Steps to update an installation of XL C/C++ V8.0 in a non-default location (for advanced users)

If you want to update an installation of XL C/C++ V8.0 that is not a basic installation, follow the procedure in either “Installing XL C/C++ V8.0 to a single non-default location” on page 21. or “Installing XL C/C++ V8.0 to multiple non-default locations” on page 22.

Chapter 4. After installing XL C/C++ V8.0

After installing XL C/C++ V8.0, there are verification and setup procedures that you will need, or might need, to follow.

For post-installation procedures, see:

- "Querying for installed packages."
- "Testing the installation."
- "Enabling the XL C/C++ V8.0 man pages" on page 14.
- "Enabling the XL C/C++ V8.0 error messages" on page 15.
- "Setting up the environment for the invocation commands" on page 17.

Querying for installed packages

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

```
rpm -q vac.cmp
```

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

The result should be:

```
vac.cmp-V.R.M-F
```

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

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

```
rpm -qa | grep -e vac -e xlsmp -e xlmass
```

If none of the packages listed in Table 4 on page 1 were properly installed, the output of the command will be null. The result should be a list that contains all of the packages listed in Table 4 on page 1.

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"

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:

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

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

```
/opt/ibmcmp/vacpp/8.0/bin/xlc hello.c -o hello
```

3. Run the program 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 <iostream>
   int main()
   {
      std::cout << "Hello World!" << std::endl;
      return 0;
   }
   ```
6. Use the **xlC** command to compile the test program.

   Compile the program:

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

     ```
     xlC hello.cpp -o hello
     ```
   - If short invocation commands have not been set up, enter the following command:

     ```
     /opt/ibmcmp/vacpp/8.0/bin/xlC hello.cpp -o hello
     ```
7. Run the program:

   ```
   ./hello
   ```

   The result should be ″Hello World!″.
8. Check the exit code of the program:

   ```
   echo $?
   ```

   The result should be “0”.

Enabling the XL C/C++ V8.0 man pages

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

Notes:

1. To invoke a man page, enter **man** followed by the command that you want explained. For example:

   ```
   man xlC
   ```
2. To leave the man page, type: **q**.

Before you can read the compiler-supplied man pages, you must add the full directory path to the MANPATH environment variable. The command depends on the Linux shell you are using. See “Basic example: Bourne or Korn shell command to enable the XL C/C++ V8.0 man pages” on page 15 or “Basic example: C shell command to enable the XL C/C++ V8.0 man pages” on page 15. To determine the shell you are using issue the following command:

```
echo $SHELL
```

To determine the current setting of the national language on your system, use the following echo command:

```
echo $LANG
```

The XL C/C++ V8.0 man pages support the following locales:

- en_US
- en_US.utf8
- ja_JP
- ja_JP.eucjp

Basic example: Bourne or Korn shell command to enable the XL C/C++ V8.0 man pages

This example assumes that:

- Your national language is English.
- The XL C/C++ V8.0 man pages do not appear properly when you display them.

Follow these steps:

1. Open a terminal window.
2. To change the locale enter one of the following commands:

   ```
   export LANG=en_US
   ```

   ```
   export LANG=en_US.utf8
   ```
3. Add the full subdirectory path (/opt/ibmcmp/vacpp/8.0/man/) to the beginning of the MANPATH environment variable:

   ```
   export MANPATH=/opt/ibmcmp/vacpp/8.0/man:$MANPATH
   ```

Basic example: C shell command to enable the XL C/C++ V8.0 man pages

This example assumes that:

- Your national language is English.
- The XL C/C++ V8.0 man pages do not appear properly when you display them.

Follow these steps:

1. Open a terminal window.
2. To change the locale, enter one of the following commands:

   ```
   setenv LANG en_US
   ```

   ```
   setenv LANG en_US.utf8
   ```
3. Add the full subdirectory path (/opt/ibmcmp/vacpp/8.0/man/)to the beginning of the MANPATH environment variable:

   ```
   setenv MANPATH  /opt/ibmcmp/vacpp/8.0/man:$MANPATH
   ```

Enabling the XL C/C++ V8.0 error messages

You do not need the information in this section if you have used the **xlc_install** utility to install XL C/C++ V8.0 and your national language is English. See Chapter 2, “Basic installation,” on page 7.

However, you must set the NLSPATH environment variable so that the XL C/C++ V8.0 compiler and runtime functions can find the appropriate message catalogs following installation *only if at least one of the following is true:*

- The national language is *not* English.
- The runtime packages are installed in one or more non-default locations.

Setting the NLSPATH environment variable to include the paths to the XL C/C++ V8.0 message catalogs

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

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

```
export NLSPATH=$NLSPATH:
  xlsmprt_path/msg/%L/%N:
  xlrte_path/msg/%L/%N:
  xlcmp_path/vacpp/8.0/msg/%L/%N
```

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

```
setenv NLSPATH $NLSPATH:
  xlsmprt_path/msg/%L/%N:
  xlrte_path/msg/%L/%N:
  xlcmp_path/vacpp/8.0/msg/%L/%N
```

where:

- *xlsmprt_path* is the installation location of the XL SMP packages.
- *xlrte_path* is the installation location of the XL C/C++ runtime package.
- *xlcmp_path* is the installation location of the XL C/C++ compiler packages.

Note: If XL SMP, XL C/C++ runtime, and XL C/C++ compiler files are all installed in the default location, *xlsmprt_path*, *xlrte_path*, and *xlcmp_path* will all be /opt/ibmcmp/.

Basic example: Setting the NLSPATH environment variable to include the paths to the XL C/C++ V8.0 message catalogs

If you are using Bourne or Korn shell:

```
export NLSPATH=$NLSPATH:
/opt/ibmcmp/msg/%L/%N:
/opt/ibmcmp/vacpp/8.0/msg/%L/%N:
```

If you are using C shell:

```
setenv NLSPATH $NLSPATH:
/opt/ibmcmp/msg/%L/%N:
/opt/ibmcmp/vacpp/8.0/msg/%L/%N:
```

Example: Setting the NLSPATH environment variable to Include the paths to the XL C/C++ V8.0 message catalogs

This example assumes that all XL SMP, XL C/C++ runtime, and XL C/C++ compiler packages are installed in different non-default locations:

- The XL SMP packages are installed in /usr/local/*xlsmprt_path*/.
- The XL C/C++ runtime package is installed in /usr/local/*xlrte_path*/.
- The XL C/C++ compiler packages are installed in /usr/local/*xlcmp_path*/.

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

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

```
export NLSPATH=$NLSPATH:
  /usr/local/xlsmprt_path/msg/%L/%N:
  /usr/local/xlrte_path/msg/%L/%N:
  /usr/local/xlcmp_path/vacpp/8.0/msg/%L/%N
```

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

```
setenv NLSPATH $NLSPATH:
  /usr/local/xlsmprt_path/msg/%L/%N:
  /usr/local/xlrte_path/msg/%L/%N:
  /usr/local/xlcmp_path/vacpp/8.0/msg/%L/%N
```

Setting up the environment for the invocation commands

If you used the **xlc_install** utility to install XL C/C++ V8.0 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 XL C/C++ V8.0 and want to be able 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 XL C/C++ V8.0 invocations."
- Create symbolic links to the compiler invocation commands, as shown in "Creating symbolic links to the XL C/C++ V8.0 compiler invocations."

Setting the PATH environment variable to include the path to the XL C/C++ V8.0 invocations

To use XL C/C++ V8.0 commands without typing the complete path, you can add the location of the compiler invocations to the PATH environment variable.

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

```
export PATH=$PATH:xlcmp_path/vacpp/8.0/bin/
```

where *xlcmp_path* is the location of all XL C/C++ V8.0 compiler files.

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

```
setenv PATH $PATH:xlcmp_path/vacpp/8.0/bin
```

where *xlcmp_path* is the location of all XL C/C++ V8.0 compiler files.

Basic example: Setting the PATH environment variable to include the path to the XL C/C++ V8.0 invocations

If you are using Bourne or Korn shell:

```
export PATH=$PATH:/opt/ibmcmp/vacpp/8.0/bin/
```

If you are using C shell:

```
setenv PATH $PATH:/opt/ibmcmp/vacpp/8.0/bin/
```

Creating symbolic links to the XL C/C++ V8.0 compiler invocations

To use XL C/C++ V8.0 without typing the complete path, you can create symbolic links in the /usr/bin/ directory for the specific invocations contained in the *xlcmp_path* /vacpp/8.0 /bin/ directory.

If you have not already done so when you ran **xlc_install**, you can create the symbolic links for the following XL C/C++ V8.0 invocations:

- **gxlc**
- **gxlc++**
- **gxlC**
- **xlc**
- **xlc++**
- **xlC**
- **xlc_r**
- **xlc++_r**
- **xlC_r**

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

- **c89**, **c89_r**, **c99**, **c99_r**, **cc**, and **cc_r**
- **mergepdf**, **new_install**, **resetpdf**, **showpdf**, **vac_configure**

Use the following command to create a symbolic link:

```
ln –s xlcmp_path/vacpp/8.0/bin/invocation /usr/bin/invocation
```

where:

- *xlcmp_path* is the location where the XL C/C++ V8.0 compiler packages were installed.
- /usr/bin/ is the directory that contains commands that can be used without specifying the full path
- *invocation* is one of the compiler invocations (such as **xlC**) in *xlcmp_path*/vacpp/8.0/bin/.

Basic example: Command to create a symbolic link to the XL C/C++ V8.0 xlC invocation

This example assumes that the entire XL C/C++ V8.0 is installed in the default location /opt/ibmcmp/.

```
ln –s /opt/ibmcmp/vacpp/8.0/bin/xlC /usr/bin/xlC
```

Chapter 5. Uninstalling XL C/C++

Notes:

1. You must have root user access to uninstall this product.
2. Whenever you uninstall a package, specify the *V.R.M-F* (Version.Release.Modification-Fix level) of the package.
3. 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 sample programs and product documentation do not have any package dependencies. You can remove them in any order.
4. You cannot uninstall packages that are required by other packages. For example, xlsmp.rte is a shared component if IBM XL Fortran compiler is also installed on the same system.
5. The uninstallation commands will not remove any configuration files that were generated by **new_install** or **vac_configure**.

Example: Uninstalling XL C/C++ V8.0

In this example:

- The XL C/C++ packages have a *V.R.M-F* of 8.0.0-0
- The XL MASS library package has a *V.R.M-F* of 4.3.0–0.
- The XL SMP library packages have a *V.R.M-F* of 1.6.0–0.

```
rpm -e vacpp.cmp-8.0.0-0
rpm -e vacpp.lib-8.0.0-0
rpm -e vacpp.rte.lnk-8.0.0-0
rpm -e vacpp.rte-8.0.0-0
rpm -e vac.cmp-8.0.0-0
rpm -e vac.lib-8.0.0-0
rpm -e vac.lic-8.0.0-0
rpm -e xlmass.lib-4.3.0-0
rpm -e xlsmp.lib-1.6.0-0
rpm -e xlsmp.rte-1.6.0-0
rpm -e xlsmp.msg.rte-1.6.0-0
```

You can issue the following commands in any order:

```
rpm -e vacpp.samples-8.0.0-0
rpm -e vacpp.help-8.0.0-0
```

Chapter 6. Installing XL C/C++ in a non-default location (for advanced users)

It is highly recommended that you uninstall any previous versions of XL C/C++ before you install XL C/C++ V8.0. If you use the **xlc_install** utility to install the new version to the default location (/opt/ibmcmp/), it will remove the old files for you.

The procedures in this chapter are intended for users who are:

- Experienced with compiler installations.
- Familiar with the file structures of all versions of all compiler products installed on the system.

Determining the tasks you need (for advanced users)

You can use the tables in this section to help you find the information you need as you install and configure the product if you have any of the following responsibilities:

- Maintaining more than one version of XL C/C++ on a computer.
- Upgrading an existing version of XL C/C++ that is installed in the non-default location.
- Installing XL C/C++ V8.0 to the non-default location.

Pre-installation checklist for an installation to a non-default location

Table 9 lists tasks that you will, or might, need to perform *before* you actually install the product.

Table 9. Pre-installation checklist for an installation to a non-default location

Check off	Task	For more information, see . . .
	Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system.
	Ensure that required and desired packages are installed in the default location.	"Verifying that the required GNU, Perl, Java, and browser packages are installed" on page 4
	Ensure that all system prerequisites are satisfied.	• "System prerequisites" on page 3 • "Verifying that there is enough space" on page 5
	Remove any older version of the compiler.	Chapter 5, "Uninstalling XL C/C++," on page 19

Post-installation checklist for an installation to a non-default location

Table 6 on page 3 lists tasks that you will, or might, need to perform *after* you actually install the product.

Installing XL C/C++ V8.0 to a single non-default location

If you want to try out a new update of the compiler before you remove the existing one from the system, you must install the new update to a non-default location. After you have verified that you want to use the new update, you will probably want to install it in the default location by performing the following tasks:

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

3. Install the new update to the default location.

Note: If you use the **xlc_install** utility to install the new update to the default location, it will remove the previous update for you.

You cannot us the **xlc_install** utility if you want to install XL C/C++ V8.0 in a non-default location. The **xlc_install** utility installs XL C/C++ V8.0 in the /opt/ibmcmp/ (default) directory.

You must install XL C/C++ V8.0 in a non-default location in order to meet both of the following criteria:

- You want to keep using an XL C/C++ version that is already installed in *xlcmp_path*.
- You want each version to use the runtime packages that were shipped with it

If your current working directory contains all of the packages for XL C/C++ V8.0 and no other RPM packages, you can use the following command to install all XL C/C++ V8.0 packages to a single non-default directory:

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

Note: When configuring the relocated compiler, use the **vac_configure** utility to configure the compiler. Do not use the **new_install** utility.

Installing XL C/C++ V8.0 to multiple non-default locations

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

Table 4 on page 1 provides information about which packages must be installed together in the same directory and which can be installed in any directory.

To install a relocatable RPM package to any location other than the default location, use the following command:

```
rpm -ivh package --prefix  installation_path
```

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

Note: When configuring the relocated compiler, use the **vac_configure** utility to configure the compiler. Do not use the **new_install** utility.

Example: Installation to multiple non-default directories

Note: In the example in this section, XL C/C++ V8.0 packages are installed in groups of packages, in different directories. Table 4 on page 1 lists the package groups that must be installed together. The variables that represent each of these directories are:

In order to avoid dependency errors during installation of XL C/C++ V8.0, issue the following commands in the order given:

```
rpm -ivh xlsmp.msg.rte-1.6.1-0.ppc64.rpm --prefix xlsmprt_path
rpm -ivh xlsmp.rte-1.6.1-0.ppc64.rpm --prefix xlsmprt_path
rpm -ivh xlsmp.lib-1.6.1-0.ppc64.rpm --prefix xlsmprt_path
rpm -ivh xlmass.lib-4.3.1-0.ppc64.rpm --prefix xlmass_path
rpm -ivh vac.lic-8.0.0-0.ppc64.rpm --prefix lic_path
rpm -ivh vac.lib-8.0.0-0.ppc64.rpm --prefix xlcmp_path
rpm -ivh vac.cmp-8.0.0-0.ppc64.rpm --prefix xlcmp_path
```

```
rpm -ivh vacpp.rte-8.0.0-0.ppc64.rpm --prefix xlrte_path
rpm -ivh vacpp.rte.lnk-8.0.0-0.ppc64.rpm --prefix xlrte_path
rpm -ivh vacpp.lib-8.0.0-0.ppc64.rpm --prefix xlcmp_path
rpm -ivh vacpp.cmp-8.0.0-0.ppc64.rpm --prefix xlcmp_path
```

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

```
rpm -ivh vacpp.help-8.0.0-0.ppc64.rpm --prefix doc_path
rpm -ivh vacpp.samples-8.0.0-0.ppc64.rpm --prefix smpls_path
```

Note: When configuring the relocated compiler, use the **vac_configure** utility to configure the compiler. Do not use the **new_install** utility.

Chapter 7. Installing multiple versions of XL C/C++ on the same system (for advanced users)

It is highly recommended that you uninstall any previous versions of XL C/C++ before you install XL C/C++ V8.0. If you use the **xlc_install** utility to install the new version to the default location (/opt/ibmcmp/), it will remove the old files for you.

The procedures in this chapter are intended for users who are both experienced with compiler installations and familiar with the file structures of all versions of all compiler products installed on the system.

Before you install multiple versions of XL C/C++ on the same system, consider the following options:

- 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.
- You can force both versions to use the more recent runtime environment. This option is recommended if you want to continue using the different versions of XL C/C++.

If you are phasing a migration to the later version of XL C/C++, you must install each version in a different installation location. You will use the information in “Installing XL C/C++ V8.0 to a single non-default location” on page 21.

If you are maintaining different versions of XL C/C++ for the long term, you should install both versions in the same installation location. Multiple versions of XL C/C++ can co-reside in the same location path as long as the later 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; it will remove the old files for you. Instead, use the following procedure: “Updating XL SMP and XL C/C++ runtime files and installing XL C/C++ V8.0”

Updating XL SMP and XL C/C++ runtime files and installing XL C/C++ V8.0

It is highly recommended that you uninstall any previous versions of XL C/C++ before you install XL C/C++ V8.0. If you use the **xlc_install** utility to install the new version to the default location (/opt/ibmcmp/), it will remove the old files for you.

However, advanced users might want to keep different versions of XL C/C++ in /opt/ibmcmp/.

For example, you might want to meet all of the following criteria:

- Continue to use XL C/C++ V7.0, which is already installed in /opt/ibmcmp/.
- Install XL C/C++ V8.0 in /opt/ibmcmp/.
- Both versions of XL C/C++ use the same runtime environment.

Example: Updating the XL SMP and XL C/C++ V7.0 runtime packages and installing XL C/C++ V8.0

This example is based on the following assumptions:

- XL C/C++ V7.0 is already installed in `/opt/ibmcmp/`.

- You need to install XL C/C++ V8.0 in /opt/ibmcmp/.
- Your current working directory contains all XL C/C++ V8.0 packages, and no other RPM packages.
- You have completed the preinstallation checklist in Table 5 on page 2.
- The existing runtime component versions listed in Table 10 are installed in /opt/ibmcmp/.

Table 10. Runtime packages that must be compatible with XL C/C++

Existing XL C/C++ V7.0 runtime packages	New XL C/C++ V8.0 runtime packages
xlsmp.msg.rte-1.5.0-0	xlsmp.msg.rte-1.6.0-0
xlsmp.rte-1.5.0-0	xlsmp.rte-1.6.0-0
xlsmp.lib-1.5.0-0	xlsmp.lib-1.6.0-0
vacpp.rte-7.0.0–0	vacpp.rte-8.0.0-0

For this scenario, you use the following steps:

1. To avoid future dependency errors, remove the existing runtime packages by issuing the following commands:

   ```
   rpm -e vacpp.rte-7.0.0-0  --nodeps
   rpm -e xlsmp.lib-1.5.0-0  --nodeps
   rpm -e xlsmp.rte-1.5.0-0  --nodeps
   rpm -e xlsmp.msg.rte-1.5.0-0  --nodeps
   ```

2. To replace the deleted runtime packages and to install XL C/C++ V8.0, issue the following command:

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

3. Modify any existing XL C/C++ V7.0 configuration file to use the new configuration path:

   ```
   # sed -e "s/xlsmp\/1\.5/xlsmp\/1\.6/g" < /etc/opt/ibmcmp/vac/7.0/vac.cfg >
     /etc/opt/ibmcmp/vac/7.0/vac.cfg.new
   # mv /etc/opt/ibmcmp/vac/7.0/vac.cfg.new
     /etc/opt/ibmcmp/vac/7.0/vac.cfg
   ```

   Note: The XL C/C++ V7.0 path does not change.

4. Generate the default XL C/C++ V8.0 configuration file:

   ```
   # /opt/ibmcmp/vac/8.0/bin/vac_configure -gcc /usr -gcc64 /usr
     -ibmcmp /opt/ibmcmp/ /opt/ibmcmp/vac/8.0/etc/vac.base.cfg
     -o  /etc/opt/ibmcmp/vac/8.0/vac.cfg
   ```

Chapter 8. Configuring XL C/C++ V8.0

After installing and trying out XL C/C++ V8.0, you might need to re-configure the compiler.

If only one version of the compiler is installed on your system, you should use the **new_install** utility to configure the compiler. The **new_install** utility automatically backs up any existing configuration file and invokes the **vac_configure** utility. Before you can run **new_install**, you must have root or administrator privileges.

For instructions, see "Running the new_install utility."

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

- You have multiple versions of the compiler installed on your system.
- You receive an error from the **new_install** command. (See Chapter 9, "Troubleshooting the installation and configuration," on page 31.)
- You want the generated configuration file be placed in a non-default location.

  Note: The default location is /etc/opt/ibmcmp/vac/8.0/
- You have multiple versions of GCC installed on your system and you need to specify which GCC version you would like to reference in the configuration file.

For instructions, see "Running the vac_configure utility directly" on page 28.

Note: If you configure the compiler using **vac_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.

Running the new_install utility

You can use the **new_install** utility to configure the compiler if *all* of the following conditions are met:

- Only one version of XL C/C++ is installed in your system.
- Only one version of GCC is installed in your system and it can be found in the PATH environment variable.
- You have write privilege in /etc/ directory.
- You want to generate the default configuration file, /etc/opt/ibmcmp/vac/8.0/vac.cfg.

.

The **new_install** utility performs the following tasks:

- Backs up any existing configuration file.
- Queries the RPM database for the paths to the XL C/C++ V8.0 packages and the path to the 32-bit GCC (*gcc32path*) and the 64-bit GCC (*gcc64path*) in the PATH environment variable, and then uses the obtained values to execute the **vac_configure** utility.
- Generates the configuration file in /etc/opt/ibmcmp/vac/8.0/vac.cfg.

You will be required to accept the license agreement and install the license file before the configuration file is generated.

To run the new_install utility:

1. Change to the directory that contains the XL C/C++ V8.0 executables:

   ```
   cd  xlcmp_path/vacpp/8.0/bin/
   ```

   where *xlcmp_path* is the installation location of the XL C/C++ compiler packages.

   Note: If XL C/C++ V8.0 is installed in the default location, *xlcmp_path* is /opt/ibmcmp/.

2. Run the following command:

   ```
   ./new_install
   ```

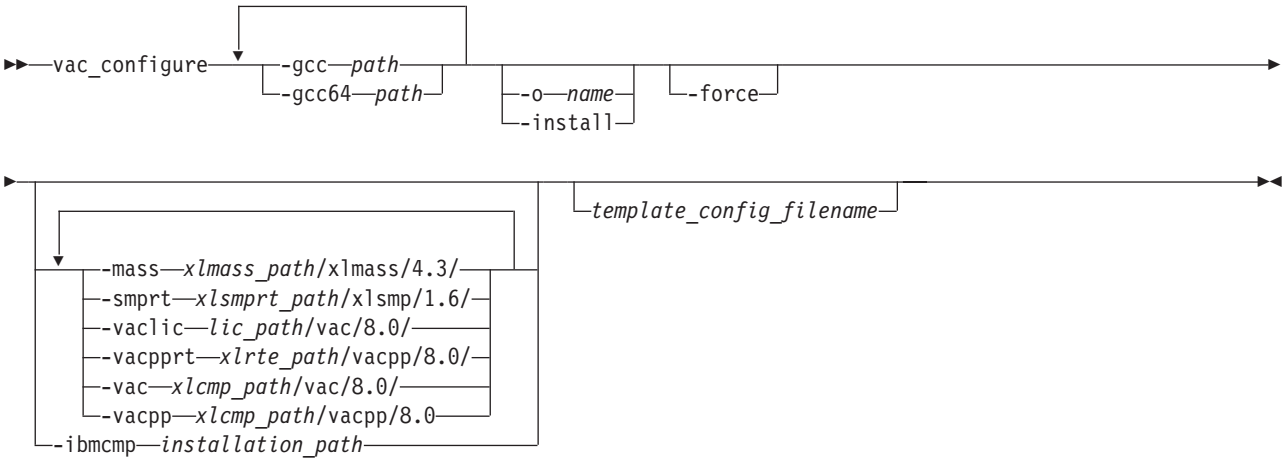
   The **new_install** command executes the following command:

   ```
   installation_path/vac/8.0/bin/vac_configure
   -gcc gcc32path
   -gcc64 gcc64path
   -install
   -mass xlmass_path/xlmass/4.3/
   -smprt xlsmprt_path/xlsmp/1.6/
   -vaclic lic_path/vac/8.0
   -vacpprt xlrte_path/vacpp/8.0
   -vac xlcmp_path/vac/8.0/
   -vacpp xlcmp_path/vacpp/8.0/
  xlcmp_path/vac/8.0/etc/vac.base.cfg
   ```

Running the vac_configure utility directly

You can use the **vac_configure** utility to generate the configuration file as long as the compiler has been successfully installed.

The **vac_configure** command has the following syntax:



where:

-gcc *path*

Specifies the path where the GCC bin/ directory is installed. In the configuration file, the *gcc_path* attribute is set equal to this path.

For example, if the GCC command is **/usr/bin/gcc**, you would specify

```
-gcc /usr
```

-gcc64 *path*
Specifies the path where the 64-bit GCC bin/ directory is installed. In the configuration file, the *gcc_path_64* attribute is set equal to this path. For example, if the 64-bit GCC command is /usr/bin/gcc -m64, you would specify

```
-gcc64 /usr
```

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

-install
Generates a default configuration file /etc/opt/ibmcmp/vac/8.0/vac.cfg.

-force Forces the **vac_configure** utility to overwrite any existing output file with the specified name and path. By default, if you do not use **force**, **vac_configure** issues an error message and stops if the specified file already exists.

-mass *xlmass_path***/xlmass/4.3/**
Specifies the path for the xlmass.lib file. By default the path is /opt/ibmcmp/xlmass/4.3/.

-smprt *xlsmprt_path***/xlsmp/1.6/**
Specifies the path for the xlsmp.msg.rte, xlsmp.rte, and xlsmp.lib files. By default the path is /opt/ibmcmp/xlsmp/1.6/.

-vaclic *lic_path***/vac/8.0**
Specifies the path for the vac.lic file. By default, this is /opt/ibmcmp/vac/8.0/.

-vacpprt *xlrte_path***/vacpp/8.0**
Specifies the path for the vacpp.rte and vacpp.rte.lnk files. By default, this is /opt/ibmcmp/vacpp/8.0/.

-vac *xlcmp_path***/vac/8.0**
Specifies the path for the vac.cmp and vac.lib files. By default, this is /opt/ibmcmp/vac/8.0/.

-vacpp *xlcmp_path***/vacpp/8.0**
Specifies the path for the vacpp.cmp and vacpp.lib files. By default, this is /opt/ibmcmp/vacpp/8.0/.

-ibmcmp *installation_path*
Alternatively specifies the path where all of the XL C/C++ packages are to be installed. For a list of these packages, see Table 4 on page 1.

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

template_config_filename
The input file that is used to construct the configuration file. By default, this is /opt/ibmcmp/vac/8.0/etc/vac.base.cfg. If you relocated the vac.cmp package to *xlcmp_path* but want to use the default template, specify:

```
xlcmp_path/vac/8.0/etc/vac.base.cfg.
```

Chapter 9. Troubleshooting the installation and configuration

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 chapter to help you respond to any problems you can encounter when you install and configure XL C/C++ V8.0.

Error messages and recommended actions

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

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 “Installation utility options” on page 7.

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 4 on page 1, 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 “Installation utility options” on page 7.

Could not determine location of 32-bit or 64-bit GCC (RHEL4 only)

Scenario

You are running either the **new_install** or the **vac_configure** utility to configure the compiler on a computer running RHEL4 when you get at least one of the following error messages:

```
ERROR: Could not determine location of 32-bit GCC. Suggestion: Ensure 32-bit "glibc-devel", 32-bit
"libstdc++-devel" are installed.  These packages can be obtained from your operating system install media.
ERROR: Could not determine location of 64-bit GCC.  Suggestion: Ensure 64-bit "glibc-devel", 64-bit
"libstdc++-devel" are installed.  These packages can be obtained from your operating system install media.
ERROR: Please ensure all relevant 32 and 64-bit GCC packages are installed before running "new_install"
again. If they are installed but cannot be detected by "new_install", please run "vac_configure" manually.
```

Explanation

Either or both of the following packages are not installed in the appropriate directory:

- `glibc-devel`
- `libstdc++-devel`

Note: On RHEL4, you cannot tell whether the 32-bit or the 64-bit version of GCC is installed by querying the packages because both the 32-bit and 64-bit packages have exactly the same name.

Action

Verify that both of the 32-bit and 64-bit packages of `glibc-devel` and `libstdc++-devel` are installed on the system by compiling test cases in both 32-bit and 64-bit modes. If the test programs compile successfully without any error message, it indicates that you have the package installed. If you get an error message, it means that you need to install the package.

Example

This example uses instances of the classic “Hello World” test case.

1. To test for 32-bit mode version of GCC, run the following commands:

   ```
   gcc helloWorld.c
   g++ helloWorld.cpp
   ```

2. To test for 64-bit mode version of GCC:

   ```
   gcc -m64 helloWorld.c
   g++ -m64 helloWorld.cpp
   ```

3. If the programs cannot be compiled successfully, it means that you need to install the necessary packages and configure the compiler again:

   a. If you get an error message from 32-bit mode testing, install the required packages, using the following commands:

      ```
      rpm -ivh glibc-devel-V.R.M-F.ppc.rpm
      rpm -ivh libstdc++-devel-V.R.M-F.ppc.rpm
      ```

      where *V.R.M-F* is the Version.Release.Modification-Fix level of the package.

      Note: The 32-bit `glibc-devel` and `libstdc++-devel` packages are available from the installation media provided with the operating system. The package file name indicates whether the package is for 32-bit mode or 64-bit mode. The 32-bit mode package file names are *.ppc.rpm.

   b. If you get an error message from 64-bit mode testing, install the required packages, using the following commands:

      ```
      rpm -ivh glibc-devel-V.R.M-F.ppc64.rpm
      rpm -ivh libstdc++-devel-V.R.M-F.ppc64.rpm
      ```

      where *V.R.M-F* is the Version.Release.Modification-Fix level of the package.

Note: The 64-bit `glibc-devel` and `libstdc++-devel` packages are available from the installation media provided with the operating system. The package file name indicates whether the package is for 32-bit mode or 64-bit mode. The 64–bit mode package file names are *.ppc64.rpm.

c. Run **new_install** or **vac_configure** again.

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
B3/KB7/8200/MKM
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.

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.

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.

Trademarks and service marks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

- eServer
- IBM
- iSeries
- POWER™
- IBM POWER

- pSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Index

Special characters

A

B

C

D

E

F

G

H

I

J

L

N

P

R

S

T

U

V

X

Program Number: 5724-M16

GC09-8017-00