

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



Installation Guide

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



Installation Guide

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 37.

Second Edition

This edition applies to IBM XL C/C++ Advanced Edition V8.0 for Linux (Program number 5724-M16) 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, 2006. All rights reserved.

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

Contents

About this document	v
Who should read this document.	v
How to use this document.	v
How this document is organized	vi
Conventions used in this document	vi
Related information.	ix
Technical support.	x
Future updates and documentation errata	x
How to send your comments	xi

Chapter 1. Before installing XL C/C++ . . . 1

The installation image and packages	1
Installation packages.	1
National language support	2
Determining the tasks you need to perform	2
Tasks for basic installation.	2
Tasks for advanced installation	3
System prerequisites.	5
Verifying the amount of hard disk space available	7
Verifying that the required GNU, Perl, and Java packages are installed	7

Chapter 2. Basic installation 9

Running the <code>xl_install</code> utility for a new installation	9
<code>xl_install</code> options	11

Chapter 3. Advanced installation 13

Installing XL C/C++ V8.0 to co-reside with earlier versions	13
Installing XL C/C++ V8.0 to co-reside with earlier versions in the same location	13
Installing XL C/C++ to a non-default location.	14
Installing all packages to a single non-default location.	15
Installing packages to multiple, non-default locations	15

Chapter 4. Installing an update 17

Running the <code>xl_install</code> utility to update a basic installation	17
--	----

Chapter 5. Configuring XL C/C++ (for advanced users) 21

Running the <code>new_install</code> utility.	21
Running the <code>vac_configure</code> utility directly	22
<code>vac_configure</code> options.	22

Chapter 6. After installing XL C/C++ . . 25

Querying for installed packages	25
Testing the installation.	25
Basic example: Creating and running “Hello World”	25
Enabling the man pages	26
Enabling the error messages.	27
Setting up the environment for the invocation commands.	28
Setting the PATH environment variable to include the path to the compiler invocations	28
Creating symbolic links to the compiler invocations	28

Chapter 7. Uninstalling XL C/C++ . . . 31

Example (SLES10, Y-HPC, YDL): Uninstalling XL C/C++ V8.0 for Linux.	31
---	----

Chapter 8. Troubleshooting the installation and configuration 33

Error messages and recommended actions	33
The specified directory <code>rpmlocation_path</code> does not exist.	33
<code>rpmlocation_path</code> does not contain . . .	33
Could not determine location of 32-bit or 64-bit GCC (RHEL4U2, Y-HPC, YDL)	33

Notices 37

Programming interface information	38
Trademarks and service marks	39
Industry standards	39

Index 41

About this document

This document contains essential information about installing IBM® 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 *installation_path*/vac/8.0/ directory, where *installation_path* is the location of the compiler on your system. If you install the compiler in the default location, the *installation_path* is /opt/ibmcmp/.

Who should read this document

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

This document addresses the needs of the majority of users who will use the basic installation method, which provides guidance during the installation process. *Basic examples* are tailored to reflect, as much as possible, the procedures for a basic installation.

This document also addresses the needs of users who want to perform 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 all versions of all compiler products installed on the system. In this document, these users are referred to as an *advanced user*. The additional information that you will need is labeled “for advanced users”.

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. It is applicable to the majority of users, and is the recommended method of installing the product. For an overview of the steps that you need to follow to perform a basic installation, refer to “Tasks for basic installation” on page 2, which provides links to the relevant procedures in this guide.

“Advanced” installation

This scenario allows you to maintain multiple versions of XL C/C++ on a single system, or to install the product to a non-default location. This scenario is applicable only to advanced users, who have specialized needs; it 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 “Tasks for advanced installation” on page 3, which provides links to the relevant procedures in this guide.

“Update” installation

This scenario applies to users who have obtained an update package for an existing XL C/C++ V8.0 basic or advanced installation. For an overview of the steps that you need to follow to perform an update, refer to “Tasks for

basic installation” on page 2 or “Tasks for advanced installation” on page 3, depending on the type of installation you need to update.

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

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 XL C/C++,” on page 1	All users
Installation	Chapter 2, “Basic installation,” on page 9	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 compilers
	Chapter 3, “Advanced installation,” on page 13	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
	Chapter 4, “Installing an update,” on page 17	Users who want to update XL C/C++ V8.0 to the next fix level
Post-installation	Chapter 5, “Configuring XL C/C++ (for advanced users),” on page 21	Users who: <ul style="list-style-type: none">• Are using the advanced, non-default method to install or update the compiler• Need to update components previously installed to a non-default location
	Chapter 6, “After installing XL C/C++,” on page 25	All users
Product removal	Chapter 7, “Uninstalling XL C/C++,” on page 31	Any user who needs to remove an XL C/C++ compiler from the system
Troubleshooting	Chapter 8, “Troubleshooting the installation and configuration,” on page 33	Any user who needs to know how to respond to an error message or unexpected results during the installation or configuration of XL C/C++

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, executable names, compiler options and pragma directives	If you specify -O3 , the compiler assumes -qhot=level=0 . To prevent all HOT optimizations with -O3 , you must specify -qnohot .
<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.
monospace	Programming keywords and library functions, compiler built-in functions, examples of program code, command strings, or user-defined names	If one or two cases of a switch statement are typically executed much more frequently than other cases, break out those cases by handling them separately before the switch statement.

Syntax diagrams

Throughout this document, 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):

►—keyword—required_argument—►◄

- Optional items are shown below the main path:

►—keyword—
|optional_argument|—►◄

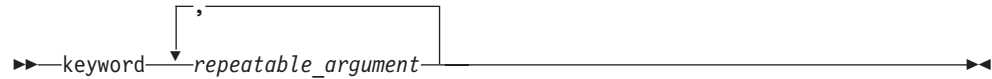
- If you can choose from two or more items, they are shown vertically, in a stack. If you *must* choose one of the items, one item of the stack is shown on the main path.

►—keyword—
|required_argument1|
|required_argument2|—►◄

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



- 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:



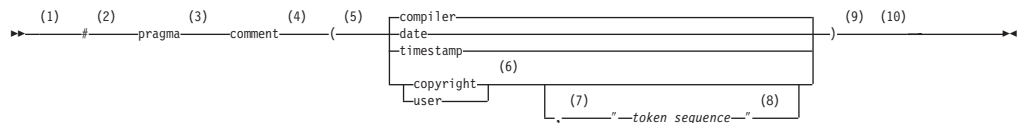
- The item that is the default is shown above the main path.



- 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 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++ 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 document.
- Information center
An information center of XL C/C++ HTML documentation is shipped with the product. HTML documents are located by default in the `/opt/ibmcmp/vacpp/8.0/doc/language/html/` directory, where *language* is one of `en_US`, `zh_CN`, or `ja_JP`. An information center of *searchable* HTML is viewable on the Web at:
<http://publib.boulder.ibm.com/infocenter/lxpcmp/v8v101/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://www.ibm.com/software/awdtools/xlcpp/library>
In addition to this document, the following files comprise the full set of XL C/C++ product manuals:

Table 3. XL C/C++ PDF files

Document title	PDF file name	Description
<i>Getting Started with IBM XL C/C++ Advanced Edition V8.0 for Linux</i> , SC09-8015-00	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.
<i>IBM XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference</i> , 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.

Table 3. XL C/C++ PDF files (continued)

Document title	PDF file name	Description
<i>IBM XL C/C++ Advanced Edition V8.0 for Linux Language Reference, SC09-8016-00</i>	language.pdf	Contains information about the C and C++ programming languages, as supported by IBM, including language extensions for portability and conformance to non-proprietary standards.
<i>IBM XL C/C++ Advanced Edition V8.0 for Linux Programming Guide, SC09-8014-00</i>	proguide.pdf	Contains information on advanced programming topics, such as application porting, interlanguage calls with Fortran code, library development, application optimization and parallelization, and the XL C/C++ 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>.

More documentation 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/library>

Other IBM publications

- Engineering and Scientific Subroutine Library (ESSL) for Linux on POWER documentation is available at <http://publib.boulder.ibm.com/infocenter/clresctr/index.jsp>

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, Technotes, and other support documents. You can find the XL C/C++ Support page on the Web at:

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

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

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>

Future updates and documentation errata

Additional information and documentation errata for all future updates to this product can be found at the following URL:

- <http://www.ibm.com/support/docview.wss?uid=swg21220307>

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 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).

Note: 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.

Chapter 1. Before installing XL C/C++

Before you install IBM® XL C/C++ Advanced Edition V8.0 for Linux®:

- 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.
- Determine the tasks you need to perform, depending on your installation requirements.
- Become either the root user or a user with administrator privileges.
- Ensure that system prerequisites are met and that all required software packages are installed.

The installation image and packages

The IBM XL C/C++ Advanced Edition V8.0 for Linux installation image 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, and documentation
- A set of RPM packages. See “Installation packages.”
- An installation tool, **xlc_install**, to install and configure the compiler for a basic installation. See “Tasks for basic installation” on page 2.
- Message catalogs for each supported national language. See “National language support” on page 2.

Installation packages

Table 4 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 Table 14 on page 15).

You can use the Linux **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 4. XL C/C++ for Linux packages and default installation locations

Package Name	Package Description	Default installation location
xlsmp.msg.rte	XL SMP runtime messages	/opt/ibmcmp/msg/
xlsmp.rte	XL SMP runtime dynamic libraries	/opt/ibmcmp/lib/ /opt/ibmcmp/lib64/
xlsmp.lib	XL SMP runtime static libraries	/opt/ibmcmp/xlsmp/1.6
xlmass.lib	IBM Mathematical Acceleration Subsystem (MASS) libraries	/opt/ibmcmp/xlmass/4.3

Table 4. XL C/C++ for Linux packages and default installation locations (continued)

Package Name	Package Description	Default installation location
vacpp.rte	XL C/C++ runtime environment	/opt/ibmcmp/lib/ /opt/ibmcmp/lib64/
vacpp.rte.lnk	XL C/C++ runtime environment links	/opt/ibmcmp/vacpp/8.0/
vac.lic	XL C/C++ license	/opt/ibmcmp/vac/8.0/
vac.lib	XL C compiler libraries	/opt/ibmcmp/vac/8.0/
vac.cmp	XL C compiler files	/opt/ibmcmp/vac/8.0/
vacpp.lib	XL C++ compiler libraries	/opt/ibmcmp/vacpp/8.0/
vacpp.cmp	XL C++ compiler files	/opt/ibmcmp/vacpp/8.0/
vacpp.samples	XL C/C++ samples	/opt/ibmcmp/vacpp/8.0/samples/
vacpp.help	XL C/C++ documentation	/opt/ibmcmp/vacpp/8.0/doc/ /opt/ibmcmp/vacpp/8.0/man/

National language support

The XL C/C++ 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 (en_US) is the default national language. Following installation, you can set the NLSPATH so that messages are displayed in a different language. See “Enabling the error messages” on page 27.

Determining the tasks you need to perform

You can use the tables provided in the following sections to help you find the information you need as you install and configure the product.

Tasks for basic installation

It is highly recommended that you use the “basic”, or default, method of installation, as long as all of the following are true:

- You are maintaining a single version of the product on your system, with or without XL Fortran Advanced Edition V10.1 for Linux.
- You are installing the product to the default location, /opt/ibmcmp/.

If these conditions match your needs, the basic installation is the easiest and fastest method, as it allows you to automatically uninstall previous versions of the compiler, install the latest version, and configure the compiler, all through the use of a single installation tool.

If you are installing XL C/C++ V8.0 for the first time, perform the steps listed in Table 5. If you are installing an update to XL C/C++ V8.0, perform the steps listed in Table 6.

Table 5. Steps for basic installation: new or upgrade installation

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 system prerequisites are satisfied.	"System prerequisites" on page 5
Use the xlc_install tool to install and configure the compiler, using the default paths.	Chapter 2, "Basic installation," on page 9
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • "Querying for installed packages" on page 25 • "Testing the installation" on page 25
Enable the compiler man pages.	"Enabling the man pages" on page 26
If your system locale and/or encoding are <i>not</i> English (en_US), enable the compiler error messages. Otherwise, you can skip this step.	"Enabling the error messages" on page 27
(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 28

Table 6. 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.
Ensure that all system prerequisites are satisfied.	"System prerequisites" on page 5
Use the xlc_install tool to install the update packages.	"Running the xlc_install utility to update a basic installation" on page 17
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • "Querying for installed packages" on page 25 • "Testing the installation" on page 25
(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 28

Tasks for advanced installation

You will need to use the "advanced" method of installation in the following cases:

- You are maintaining multiple versions of the same product on a single system.
- You are installing the product to a non-default location.

If any of these conditions is true, you need to follow an "advanced" installation method which requires that you separately install and configure the compiler. You may also need to manually uninstall previous versions of the compiler from your system.

If you are installing XL C/C++ V8.0 for the first time, perform the steps listed in Table 7. If you are installing an update to XL C/C++ V8.0, perform the steps listed in Table 8.

Table 7. Steps for advanced installation: new or upgrade installation

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 XL C/C++ for Linux.	Chapter 7, "Uninstalling XL C/C++," on page 31
Ensure that all system prerequisites are satisfied.	"System prerequisites" on page 5
Use one of the advanced installation methods to install the compiler.	Chapter 3, "Advanced installation," on page 13
Use the new_install or vac_configure tool to configure the compiler.	Chapter 5, "Configuring XL C/C++ (for advanced users)," on page 21
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • "Querying for installed packages" on page 25 • "Testing the installation" on page 25
Enable the compiler man pages.	"Enabling the man pages" on page 26
If your system locale and/or encoding are <i>not</i> English (en_US), enable the compiler error messages. Otherwise, you can skip this step.	"Enabling the error messages" on page 27
(Optional) Set up the environment to locate the invocation commands without the full path.	"Setting up the environment for the invocation commands" on page 28

Table 8. 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.
Ensure that all system prerequisites are satisfied.	"System prerequisites" on page 5
Install the update packages to the non-default location.	"Installing XL C/C++ to a non-default location" on page 14
Configure the compiler.	Chapter 5, "Configuring XL C/C++ (for advanced users)," on page 21
Confirm that the compiler packages were successfully installed, and test the installation.	<ul style="list-style-type: none"> • "Querying for installed packages" on page 25 • "Testing the installation" on page 25
(Optional) Set up the environment to locate the invocation commands without the full path.	"Setting up the environment for the invocation commands" on page 28

System prerequisites

The following are the requirements for installing IBM XL C/C++ Advanced Edition V8.0 for Linux:

- **Operating system:** A supported Linux distribution:
 - Red Hat Enterprise Linux AS 4 Update 2 (RHEL4U2) for IBM POWER™
 - SUSE Linux Enterprise Server 9 Service Pack 2 (SLES9 SP2) for IBM POWER
 - SUSE Linux Enterprise Server 10 (SLES10) for IBM POWER
 - Terra Soft's Y-HPC v1.1 (Y-HPC) or Yellow Dog Linux v4.1 (YDL)
- **Hardware:** You can use any of the following hardware platforms (as long as your operating system distribution supports it):
 - IBM POWER Technology-based system
 - IBM BladeCenter® JS20
 - IBM System i™
 - IBM System p™
 - Apple Power Mac G5, Apple Xserver G5
- **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.

To verify that you have enough hard disk space available, see the procedure in “Verifying the amount of hard disk space available” on page 7.

- **Required software:**
 - GNU, Perl, and Java™ packages, as listed in the following tables. To verify that the required packages are installed, see the procedure in “Verifying that the required GNU, Perl, and Java packages are installed” on page 7.

Table 9. Required GNU, Perl, and Java packages for the RHEL4U2 operating system

Package name	Version requirements
gcc	3.4.4
gcc-c++	3.4.4
glibc	2.3.4
glibc-devel	2.3.4
libgcc	3.4.4
libstdc++	3.4.4
libstdc++-devel	3.4.4
Perl	5.0 or greater Note: Perl V5.0 is shipped and automatically installed with the RHEL4U2 operating system.
libgcj	3.4.4

Table 10. Required GNU, Perl, and Java packages for the SLES9 SP2 operating system

Package name	Version requirements
gcc	3.3.3

Table 10. Required GNU, Perl, and Java packages for the SLES9 SP2 operating system (continued)

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
libstdc++-64bit	9
libstdc++-devel-64bit	9
Perl	5.0 or greater Note: Perl V5.0 is shipped and automatically installed with the SLES9 SP2 operating system.
java2	1.3.1
java2-jre	1.3.1

Table 11. Required GNU, Perl, and Java packages for the SLES10 operating system

Package name	Version requirements
gcc	4.1.0
gcc-c++	4.1.0
glibc	2.4
glibc-64bit	2.4
glibc-devel-64bit	2.4
libgcc	4.1.0
libgcc-64bit	4.1.0
libstdc++	4.1.0
libstdc++-64bit	4.1.0
libstdc++-devel-64bit	4.1.0
Perl	5.0 or greater Note: Perl V5.0 is shipped and automatically installed with the SLES10 operating system.
libgcj	4.1.0

Table 12. Required GNU, Perl, and Java packages for the Y-HPC and YDL operating systems

Package name	Version requirements
gcc	3.4.4
gcc-ppc32	3.4.4
gcc-c++	3.4.4
glibc*	2.3.4

Table 12. Required GNU, Perl, and Java packages for the Y-HPC and YDL operating systems (continued)

glibc-devel*	2.3.4
libgcc*	3.4.4
libstdc++*	3.4.4
libstdc++-devel*	3.4.4
Perl	5.0 or greater Note: Perl V5.0 is shipped and automatically installed with the Y-HPC or YDL operating system.
libgcj	3.4.4

Note: * Both 32-bit and 64-bit versions are required.

- **Optional software:**

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

Verifying the amount of hard disk space available

XL C/C++ requires about 200 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/):

```
df -h /opt
```

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

```
df -h installation_path
```

where *installation_path* represents the non-default location.

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

Before you can install XL C/C++ for Linux, 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 Linux distribution, see one of the following:

- Table 9 on page 5 Required GNU, Perl, and Java packages for the RHEL4U2 operating system
- Table 10 on page 5 Required GNU, Perl, and Java packages for the SLES9 SP2 operating system
- Table 11 on page 6 Required GNU, Perl, and Java packages for the SLES10 operating system
- Table 12 on page 6 Required GNU, Perl, and Java packages for the Y-HPC and YDL operating systems

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

```
gcc-c++-4.1.0-43.24
```

Note: On RHEL4U2, YDL, and Y-HPC, 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 (RHEL4U2, Y-HPC, YDL)” on page 33. Do not use the example in this section to verify that these packages are installed because the names of the packages on RHEL4U2, Y-HPC, and YDL 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.

Chapter 2. Basic installation

XL C/C++ provides an interactive utility, **xlc_install**, that walks you through a basic installation. You can use **xlc_install** to do any of the following:

- Install XL C/C++ V8.0 for Linux on a clean system
- Install XL C/C++ V8.0 for Linux on a system where XL Fortran Advanced Edition V10.1 for Linux is already installed.

In this case, both compilers will use the XL SMP and IBM MASS libraries provided with XL C/C++ for Linux. (For information about installing XL Fortran V10.1 for Linux, refer to the *XL Fortran Advanced Edition V10.1 for Linux Installation Guide*.)

- Install an update on a system where XL C/C++ V8.0 for Linux has already been installed.

In this case, use the procedure in “Running the **xlc_install** utility to update a basic installation” on page 17.

You should use the **xlc_install** utility to install XL C/C++ as long as *all* of the following conditions apply:

- You are installing the compiler to the default location, which is `/opt/ibmcmp/`.
- You agree to remove any previously installed XL C/C++ components.

If any of these conditions does not apply, do *not* use the **xlc_install** utility. Instead, see the procedures in Chapter 3, “Advanced installation,” on page 13.

Running the **xlc_install** utility for a new 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 utility. See “Verifying that the required GNU, Perl, and Java packages are installed” on page 7.

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++ Advanced Edition V8.0 for Linux:

1. Assuming that 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 11.

If another instance of XL C/C++ for Linux is detected on your system, you are prompted to uninstall it.

2. 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 XL SMP and IBM MASS packages are detected on your system, either alone or as part of an XL Fortran Advanced Edition V10.1 for Linux installation, you are prompted to uninstall them.
3. Confirm that you want to proceed with uninstalling the existing XL SMP and IBM MASS packages. If you choose not to uninstall the previously installed components, the installation process will terminate.

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 XL Fortran for Linux installation, you will need to run `xlf_configure` to reconfigure the XL Fortran compiler to point to the default location for these packages. For procedures, see “Running the `xlf_configure` utility directly” in the *XL Fortran Advanced Edition V10.1 for Linux Installation Guide*.

You are presented with the licensing agreement and licensing information.

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

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

5. 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 28.

If you chose to create symbolic links, 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 or GCC-related invocations, or because they are not compiler invocation commands. These include:

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

If all packages are successfully installed:

- A message is displayed confirming the successful installation.
- The configuration file is generated. Its location is `/etc/opt/ibmcmp/vac/8.0/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/vac/8.0/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 `./platform/rpms`, which is relative to the path of the installation tool. Therefore, the default *rpmlocation_path* is one of the following:

- `./RHEL4/rpms` (if you are installing on RHEL4U2)
- `./SLES9/rpms` (if you are installing on SLES9 SP2)
- `./SLES10/rpms` (if you are installing on SLES10)
- `./Y-HPC/rpms` (if you are installing on Y-HPC or YDL)

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

-U Updates the compiler to the *Version.Release.Modification-Fix* (V.R.M-F) level that the installation utility version supports. For more information, see “Running the **xlc_install** utility to update a basic installation” on page 17.

-v Displays debugging information.

-vv Displays more detailed debugging information.

Chapter 3. Advanced installation

It is highly recommended that you install XL C/C++ for Linux to the default location and use the procedure provided in Chapter 2, “Basic installation,” on page 9. However, you will need to use alternate procedures for customized scenarios, including the following:

- You want to maintain more than one version of XL C/C++ for Linux on the same system. For more information on possible scenarios as well as installation procedures, see “Installing XL C/C++ V8.0 to co-reside with earlier versions.”
- You want to upgrade or update an existing version of XL C/C++ that is installed in a non-default location. For installation procedures, see “Installing XL C/C++ to a non-default location” on page 14.
- You want to try out a new update of the compiler before removing an existing installation from the default location. In this case, you will need to install the update to a non-default location; for procedures, see “Installing XL C/C++ to a non-default location” on page 14.

In all of these scenarios, you must use the **rpm** utility to install the compiler; you cannot use the **xl_c_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 **vac_configure** utilities; see Chapter 5, “Configuring XL C/C++ (for advanced users),” on page 21 for procedures.

Installing XL C/C++ V8.0 to co-reside with earlier versions

To install XL C/C++ V8.0 to co-reside with earlier versions of XL C/C++ on the same system, you have the following options:

- You can install multiple versions in the same location, while forcing all versions to use the most recent runtime environment. This option is recommended if you want to continue using the different versions of XL C/C++ over the long term. Multiple versions of XL C/C++ can co-reside in the same location 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 **xl_c_install** to install the additional version; it will remove the old files for you. Instead, use the procedure in “Installing XL C/C++ V8.0 to co-reside with earlier versions in the same location”
- You can allow each version to use the runtime environment that was shipped with it. This option is recommended if you want to phase a migration to the later version. In this case, you will need to install XL C/C++ V8.0 in a separate location from all other versions of XL C/C++. To do so, follow one of the procedures in “Installing XL C/C++ to a non-default location” on page 14.

Installing XL C/C++ V8.0 to co-reside with earlier versions in the same location

The following procedure describes how to install XL C/C++ V8.0 and update the runtime packages used by an existing version of XL C/C++ in the same location. This procedure assumes the following:

- You are installing XL C/C++ V8.0 to the same location where XL C/C++ V 7.0 is already installed (by default, `/opt/ibmcmp/`).

- Your current working directory contains all XL C/C++ V8.0 packages, and no other RPM packages.
- The existing runtime component versions listed in Table 13 are installed in the same installation location (by default, /opt/ibmcmp/).

Table 13. XL C/C++ SMP and runtime packages (for RHEL4U2 or SLES9 SP2 installations)

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

To install XL C/C++ V8.0 and update the XL C/C++ V7.0 XL SMP and runtime packages:

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
```

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 XL C/C++ V8.0, issue the following command:

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

3. As a precaution, make a backup copy of any existing configuration file.
4. 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/installation_path/vac/7.0/vac.cfg >
    /etc/installation_path/vac/7.0/vac.cfg.new
# mv /etc/installation_path/vac/7.0/vac.cfg.new
    /etc/installation_path/vac/7.0/vac.cfg
```

where *installation_path* is the location in which all XL C/C++ packages are installed (by default, /opt/ibmcmp/).

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

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

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

where *installation_path* is the location in which all XL C/C++ packages are installed (by default, /opt/ibmcmp/).

Installing XL C/C++ to a non-default location

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.”

Note that in both of these scenarios, it is highly recommended that you uninstall any previous versions of the compiler from your system before installing any new versions; for uninstallation procedures, see Chapter 7, “Uninstalling XL C/C++,” on page 31.

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 XL C/C++ V8.0 for Linux 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 14 provides information about which packages must be installed together in the same directory and which can be installed in any directory.

Table 14. Rules for installing packages to multiple, non-default locations

Package Name	Package Description	Rules for installation to a non-default location
xlsmp.msg.rte	XL SMP runtime messages	All XL SMP packages must be installed in the same location. For the remainder of this document, the name <i>xlsmp_rte_path</i> is used to refer to this 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. For the remainder of this document, the name <i>xlmass_path</i> is used to refer to this location.
vacpp.rte	XL C/C++ runtime environment	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.
vacpp.rte.lnk	XL C/C++ runtime environment links	

Table 14. Rules for installing packages to multiple, non-default locations (continued)

Package Name	Package Description	Rules for installation to a non-default location
vac.lic	XL C/C++ license	Any location. For the remainder of this document, the name <i>lic_path</i> is used to refer to this location.
vac.lib	XL C compiler libraries	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.
vac.cmp	XL C compiler files	
vacpp.lib	XL C++ compiler libraries	Must be installed in the same location as the other compiler and library packages above.
vacpp.cmp	XL C++ compiler files	
vacpp.help	XL C/C++ documentation	Any location (optional). For the remainder of this document, the name <i>doc_path</i> is used to refer to this location.
vacpp.samples	XL C/C++ samples	Any location (optional). For the remainder of this document, the name <i>smpls_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 14 on page 15.

Example (SLES10, Y-HPC, YDL): Installing XL C/C++ V8.0 for Linux to multiple non-default directories

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

```
rpm -ivh xlsmp.msg.rte-1.6.1-0.ppc64.rpm --prefix xlsmp_rte_path
rpm -ivh xlsmp.rte-1.6.1-0.ppc64.rpm --prefix xlsmp_rte_path
rpm -ivh xlsmp.lib-1.6.1-0.ppc64.rpm --prefix xlsmp_rte_path
rpm -ivh xlmass.lib-4.3.1-0.ppc64.rpm --prefix xlmass_path

rpm -ivh vacpp.rte-8.0.1-0.ppc64.rpm --prefix xlrte_path
rpm -ivh vacpp.rte.lnk-8.0.1-0.ppc64.rpm --prefix xlrte_path
rpm -ivh vac.lic-8.0.1-0.ppc64.rpm --prefix lic_path
rpm -ivh vac.lib-8.0.1-0.ppc64.rpm --prefix xlcmp_path
rpm -ivh vac.cmp-8.0.1-0.ppc64.rpm --prefix xlcmp_path
rpm -ivh vacpp.lib-8.0.1-0.ppc64.rpm --prefix xlcmp_path
rpm -ivh vacpp.cmp-8.0.1-0.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-8.0.1-0.ppc64.rpm --prefix doc_path
rpm -ivh vacpp.samples-8.0.1-0.ppc64.rpm --prefix smpls_path
```

Chapter 4. Installing an update

An update of XL C/C++ V8.0 for Linux 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 update package comes in tar.gz (or 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 XL C/C++ V8.0 for Linux installed on your system, you can apply the latest update. If you are applying the update to an installation in the default location, follow the procedure in “Running the **xlc_install** utility to update a basic installation.” If you are applying the update to an installation in a non-default location, you must use one of the procedures in “Installing XL C/C++ to a non-default location” on page 14.

If you want to try out a new update of the compiler before you remove the existing version from the system, you must install the new update to a non-default location. After you have verified that you want to replace the older version with the new update, you can run the **xlc_install** utility that comes with the update package, and it will do all of the following:

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

Note: Do not uninstall the **vac.lic** package because this package is required for the next step.

3. Re-install the new update to the default location

For instructions to use the **xlc_install** utility to install the new update, see “Running the **xlc_install** utility to update a basic installation.”

Running the **xlc_install** utility to update a basic installation

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

- The base version of XL C/C++ V8.0 for Linux is already successfully installed in the `/opt/ibmcmp/` directory.
- The update package (in tar.gz format) 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 XL C/C++ V8.0 for Linux 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++ Advanced Edition V8.0 for Linux:

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, jun2006 indicates a shipment date in June of 2006.)

2. Issue the following command:

```
# ./xlc_install -U
```

For additional arguments that you can specify for **xlc_install**, see “**xlc_install** options” on page 11.

You are prompted to uninstall any previously installed XL C/C++ V8.0 for Linux packages.

3. Confirm that you want to proceed with uninstalling the outdated packages.

You are prompted to uninstall any XL SMP and IBM MASS packages that were previously installed with XL C/C++ V8.0 for Linux.

4. Confirm that you want to proceed with uninstalling the existing XL SMP and 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 XL Fortran for Linux installation, you will need to run **xlfc_configure** to reconfigure the XL Fortran compiler to point to the default location for these packages. For procedures, see “Running the **xlfc_configure** utility directly” in the *XL Fortran Advanced Edition V10.1 for Linux 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 28.

If you chose to create symbolic links, 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 or GCC-related invocations, or because they are not compiler invocation commands. These include:

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

7. 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.

Chapter 5. Configuring XL C/C++ (for advanced users)

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

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

Two configuration tools are provided with the compiler: **new_install** and **vac_configure**, both located in the *installation_path*/vacpp/8.0/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 XL C/C++ for Linux is installed on your system.
- Only one version of GCC is installed in your system and it can be found in the PATH environment variable.
- You have root or administrator privileges.
- You want to generate the configuration file in the default directory /etc/opt/ibmcmp/vac/8.0/.

For instructions, see “Running the new_install utility.”

You should invoke the **vac_configure** utility directly *only* when any of the following is true:

- You have multiple versions of XL C/C++ 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 33.)
- You want the generated configuration file to be placed in a non-default location.
- You have multiple versions of GCC installed on your system and you need to specify which GCC version you would like to reference in the configuration file.

Note: If you configure the compiler using **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.

For instructions, see “Running the **vac_configure** utility directly” on page 22.

Running the new_install utility

The **new_install** utility does the following:

- Backs up any existing configuration file
- Queries the RPM database for the paths to the compiler 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
- Installs the license file
- Generates the configuration file in the default location /etc/opt/ibmcmp/vac/8.0/vac.cfg

To run the **new_install** utility:

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

```
cd installation_path/vacpp/8.0/bin/
```

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

2. Run the following command:

```
./new_install
```

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

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.

To run the **vac_configure** utility:

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

```
cd installation_path/vacpp/8.0/bin/
```

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

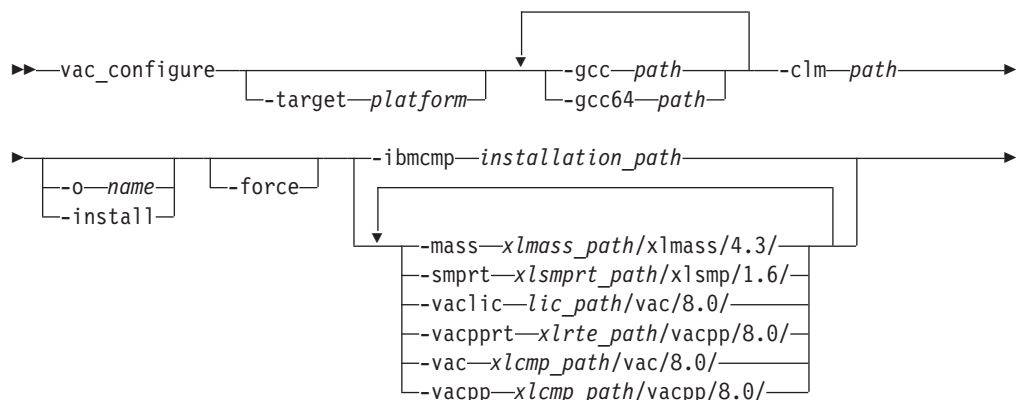
2. Run the following command:

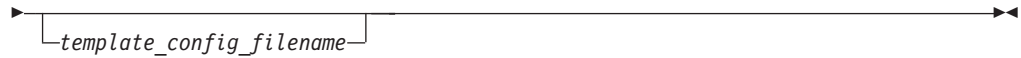
```
./vac_configure options
```

See the following section for required arguments to the **vac_configure** command.

vac_configure options

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





where:

-target *platform*

Specifies the operating system platform. Valid values are:

- sles
- rhel
- yhpc

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

-gcc *path*

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

-gcc /usr

-gcc64 *path*

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

-gcc64 /usr

-clm *path*

Specifies the path where the Common License Manager's license.dat file is located. By default, this is **/opt/clm_ibm**

-o *file_name*

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

-install

Generates the configuration file as **/etc/opt/ibmcmp/vac/8.0/vac.cfg**. By default, output is written to the display only.

-force

Forces the **vac_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**, **vac_configure** issues an error message and stops if the specified file already exists.

-ibmcmp *installation_path*

Specifies the path where all of the XL C/C++ packages are installed (if all packages are installed in the same path). By default, the path is **/opt/ibmcmp/**.

-mass *xlmass_path/xlmass/4.3/*

Specifies the path where the xlmass.lib package is installed. By default, the full path is **/opt/ibmcmp/xlmass/4.3**.

-smprt *xlsmprt_path/xlsmp/1.6/*

Specifies the path where the xlsmp.msg.rte, xlsmp.rte, and xlsmp.lib packages are installed. By default, the full path is **/opt/ibmcmp/xlsmp/1.6**.

-vaclic *lic_path/vac/8.0/*

Specifies the path where the vac.lic package is installed. By default, the full path is /opt/ibmcmp/vac/8.0/.

-vacpprt *xlrte_path/vacpp/8.0/*

Specifies the path where the vacpp.rte and vacpp.rte.lnk packages are installed. By default, the full path is /opt/ibmcmp/vacpp/8.0/.

-vac *xlcmp_path/vac/8.0/*

Specifies the path where the vac.cmp and vac.lib packages are installed. By default, the full path is /opt/ibmcmp/vac/8.0/.

-vacpp *xlcmp_path/vacpp/8.0/*

Specifies the path where the vacpp.cmp and vacpp.lib packages are installed. By default, the full path is /opt/ibmcmp/vacpp/8.0/.

template_config_file_name

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 6. After installing XL C/C++

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:

- “Querying for installed packages”: This section applies to all users.
- “Testing the installation”: This section applies to all users.
- “Enabling the man pages” on page 26: This section applies to all users.
- “Enabling the error messages” on page 27: 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” on page 28: This section applies only to users who did not use `xl_c_install` to install or update the product, or who did not create symbolic links during the installation process with `xl_c_install`.

Querying for installed packages

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

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

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.

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 vac -e xlsmp -e xlmass
```

The result should be a list that contains all of the packages listed in Table 4 on page 1. If none of the packages listed in the table was properly installed, there will be no output from the command.

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/ibmcomp/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. 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/ibmcomp/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 man pages

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

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

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

However, 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.

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

```
export MANPATH=installation_path/vacpp/8.0/man:$MANPATH
```


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

```
setenv MANPATH installation_path/vacpp/8.0/man:$MANPATH
```

where *installation_path* is the location where you have installed the XL C/C++ 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. 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, 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. If your system uses the en_US locale but the runtime is installed in a non-default location, you must set the NLSPATH environment variable.

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:  
  xlsmp_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:  
  xlsmp_path/msg/%L/%N:  
  xlrte_path/msg/%L/%N:  
  xlcmp_path/vacpp/8.0/msg/%L/%N
```

where:

- *xlsmp*_path is the installation location of the XL SMP packages. By default this is /opt/ibmcmp/.
- *xlrte*_path is the installation location of the XL C/C++ runtime packages. By default this is /opt/ibmcmp/.
- *xlcmp*_path is the installation location of the XL C/C++ 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. 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 `xl_c_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.”

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

To use XL C/C++ for Linux 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/vacpp/8.0/bin/
```

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

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

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

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. 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*/vacpp/8.0/bin/ directory.

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

- `gxc`
- `gxc++`
- `gxC`
- `xc`
- `xc++`

- 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
- cleanpdf, mergepdf, new_install, resetpdf, showpdf, vac_configure

Use the following command to create a symbolic link:

```
ln -s installation_path/vacpp/8.0/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 xlc++) in *installation_path*/vacpp/8.0/bin/.

Basic example: Creating a symbolic link to the xlc compiler invocation

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

```
ln -s /opt/ibmcmp/vacpp/8.0/bin/xlc /usr/bin/xlc
```

Chapter 7. Uninstalling XL C/C++

XL C/C++ does not provide a standalone uninstallation tool. You must use the Linux **rpm** utility to uninstall XL C/C++.

Note:

1. You must have root user access to uninstall the compiler.
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 XL Fortran Advanced Edition V10.1 for Linux 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 (SLES10, Y-HPC, YDL): Uninstalling XL C/C++ V8.0 for Linux

In this example:

- The compiler packages have a *V.R.M-F* of 8.0.1-0
- The IBM MASS library package has a *V.R.M-F* of 4.3.1-0.
- The XL SMP library packages have a *V.R.M-F* of 1.6.1-0.

To uninstall XL C/C++ V8.0 for Linux, issue the following commands, in the same order:

```
rpm -e vacpp.cmp-8.0.1-0
rpm -e vacpp.lib-8.0.1-0
rpm -e vac.cmp-8.0.1-0
rpm -e vac.lib-8.0.1-0
rpm -e vac.lic-8.0.1-0
rpm -e vacpp.rte.lnk-8.0.1-0
rpm -e vacpp.rte-8.0.1-0

rpm -e xlmass.lib-4.3.1-0
rpm -e xlsmp.lib-1.6.1-0
rpm -e xlsmp.rte-1.6.1-0
rpm -e xlsmp.msg.rte-1.6.1-0
```

You can issue the following commands in any order:

```
rpm -e vacpp.samples-8.0.1-0
rpm -e vacpp.help-8.0.1-0
```

Chapter 8. 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 section to help you respond to any problems you may encounter when you install and configure XL C/C++.

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 "xlc_install options" on page 11.

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 "xlc_install options" on page 11.

Could not determine location of 32-bit or 64-bit GCC (RHEL4U2, Y-HPC, YDL)

Scenario

You are running either the **new_install** or the **vac_configure** utility to

configure the compiler on a computer running RHEL4U2, Y-HPC, YDL or 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 RHEL4U2, Y-HPC, or YDL 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

Note to U.S. Government Users Restricted Rights -- use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

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.

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.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow the customer to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification, and tuning information is provided to help you debug your application software.

Note: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

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

BladeCenter
IBM
IBM (logo)
POWER
System i
System p

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

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

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

Industry standards

The following standards are supported:

- The C language is consistent with the International Standard for Information Systems-Programming Language C (ISO/IEC 9899-1999 (E)).
- The C++ language is consistent with the International Standard for Information Systems-Programming Language C++ (ISO/IEC 14882:1998).
- The C++ language is also consistent with the International Standard for Information Systems-Programming Language C++ (ISO/IEC 14882:2003 (E)).
- The C and C++ languages are consistent with the OpenMP C and C++ Application Programming Interface Version 2.5.

Index

A

- advanced installation, defined 3
- advanced user, described v
- advanced users, procedures for 13
- available space, determining 7

B

- basic example, described ix
- basic installation, defined 2
- basic user, described v

C

- caches
 - required hard disk space 5
- co-residency
 - compilers 13
- configuration files
 - backup 21
 - customizing vi, 21, 22
 - default 23
 - editing 21, 22
 - generation 11, 23
 - modifying vi, 21, 22
 - multiple vi, 21
 - overwriting 23
 - renaming 23
 - security 31

D

- debugger support 11
- debugging 33

G

- gcc-c++, determining installed version 8
- GNU 5, 7

H

- hard drive space, available 7
- help
 - displaying 11

I

- installation
 - multiple versions 13
 - options 11
 - package location 15
 - specialized 15
- installation CD 1
- installation image 1
- installation log 33
- installation utility
 - options 11

- installation utility (*continued*)
 - using 17

J

- Java 5, 7

L

- locales
 - supported 2

N

- national language support 2

P

- packages
 - not found 33
 - required 5
- packages, installation 1
- Perl 5, 7
- pre-installation
 - planning 2
- preinstallation 1

R

- Red Hat Package Manager (RPM)
 - installation image 1
 - utility 1

T

- temporary files
 - high optimization levels 5
 - installation log 33
 - required hard disk space 5

U

- updates
 - applying 17
 - installation utility option 11
 - packages 17
 - prerequisites 17
 - steps 17
 - to a non-default location 17
 - trying out 15, 17

V

- verification of installation 7

X

- XL compilers
 - co-residency 13



Program Number: 5724-M16

GC09-8017-01

