IBM XL C/C++ for Multicore Acceleration for Linux, V9.0

Installation Guide

GC23-8520-00

IBM XL C/C++ for Multicore Acceleration for Linux,
V9.0

Installation Guide

GC23-8520-00

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 33.

First Edition

This edition applies to IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0 and IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0. (Programs 5724-T42 & 5724-T43)

Contents

About this document

This document contains essential information about installing IBM® XL C/C++ for Multicore Acceleration for Linux®, V9.0. Read the documentation carefully before installing this product. Be sure to read the README.FIRST file, which contains the installation image layout on the CD and 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*/xlc/cbe/9.0 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++ for Multicore Acceleration for Linux, V9.0.

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 IBM XL C/C++ for Multicore Acceleration for Linux on a single system. These are users who are familiar with IBM XL 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 advanced users. 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 IBM XL C/C++ for Multicore Acceleration for Linux 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 3.

"Advanced" installation
This scenario allows you to maintain multiple versions of IBM XL C/C++ for Multicore Acceleration for Linux 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 4.

How this document is organized

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

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

Phase	Chapters	User segment
Pre-installation	Chapter 1, "Before installing IBM XL C/C++ for Multicore Acceleration for Linux, V9.0," on page 1	All users
Installation	Chapter 2, "Basic installation," on page 7	Users who: • 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 11	Users who: • 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 15	Users who want to update IBM XL C/C++ V9.0 to the next fix level
Post-installation	Chapter 5, "Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 (for advanced users)," on page 19	Users who: • 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 IBM XL C/C++ for Multicore Acceleration for Linux, V9.0," on page 23	All users
Product removal	Chapter 7, "Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V9.0," on page 29	Any user who needs to remove an IBM XL C/C++ compiler from the system
Troubleshooting	Chapter 8, "Troubleshooting the installation and configuration," on page 31	Any user who needs to know how to respond to an error message or unexpected results during the installation or configuration of IBM 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	Lowercase 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**.
italics	Parameters or variables whose actual names or values are to be supplied by the user. Italics are also used to introduce new terms	Make sure that you update the *size* parameter if you return more than the *size* requested.
`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 IBM 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:

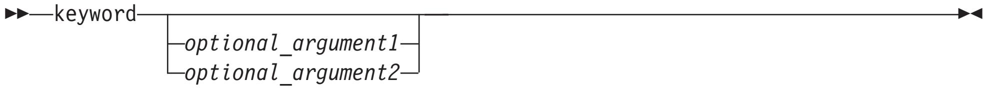
```
►►──keyword──┬────────────────────┬─────────────────────────────────────►◄
             └─optional_argument──┘
```

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

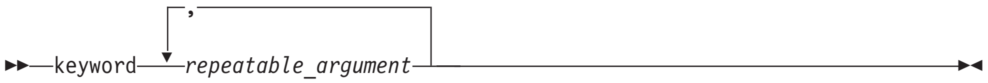
  If you *must* choose one of the items, one item of the stack is shown on the main path.

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

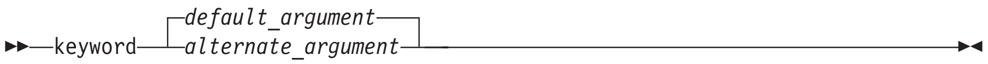
  If choosing one of the items is optional, the entire stack is shown below the main path.



- 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:



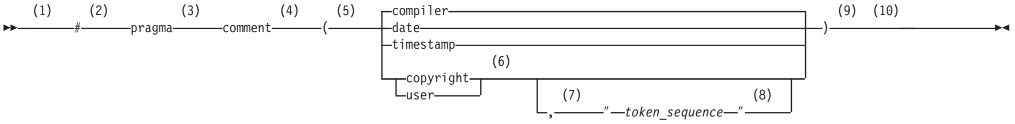
- The item that is the default is shown above the main path.



- 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

IBM 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 *installation_path*/xlc/cbe/9.0 directory.
- 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 IBM XL C/C++ HTML documentation is viewable on the Web at:

  http://publib.boulder.ibm.com/infocenter/cellcomp/v9v111/index.jsp
- PDF documents

  PDF documents are located by default in the /opt/ibmcmp/xlc/cbe/9.0/doc/*language*/pdf/ directory, where *language* is en_US . The PDFs are also available on the Web at:

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

  The following files comprise the full set of IBM XL C/C++ product manuals:

Table 3. IBM XL C/C++ PDF files

Document title	**PDF file name**	**Description**
IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 Installation Guide	install.pdf	Contains information for installing IBM XL C/C++ and configuring your environment for basic compilation and program execution.
Getting Started with IBM XL C/C++ for Multicore Acceleration for Linux, V9.0	getstart.pdf	Contains an introduction to the IBM XL C/C++ product, with information on setting up and configuring your environment, compiling and linking programs, and troubleshooting compilation errors.
IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 Compiler Reference	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. IBM XL C/C++ PDF files (continued)

Document title	PDF file name	Description
IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 Language Reference	langref.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.
IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 Programming Guide	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 IBM 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 IBM 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

Technical support

Additional technical support is available from the IBM XL C/C++ Support page. This page provides a portal with search capabilities to a large selection of Technotes, and other support documents. You can find the IBM 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 IBM XL C/C++, visit the product information site at:

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

How to send your comments

Your feedback is important in helping to provide accurate and high-quality information. If you have any comments about this document or any other IBM 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 IBM 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 IBM XL C/C++ for Multicore Acceleration for Linux, V9.0

This installation guide covers both IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0 and IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0. The term IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 will refer to both x86 and System p. x86 and System p will be explicitly specified when there is a difference between the two products.

Before you install IBM XL C/C++ for Multicore Acceleration for Linux, V9.0:

- 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++ for Multicore Acceleration for Linux, V9.0 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, notices, 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 3.

Note: IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0 and IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0 have the same installation path. This is due to the fact that the two compilers are each only supported on their corresponding pair of hardware architectures and Linux distributions:

- **IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0** is supported on Intel x86 systems running Red Hat Enterprise Linux 5 Update 1 (RHEL5.1)
- **IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0** is supported on IBM POWER™ systems running Red Hat Enterprise Linux 5 Update 1 (RHEL5.1)

Installation packages

Table 4 on page 2 lists the packages that are supplied with the installation image, and the locations to which they are installed by default during a basic installation. (For the rules on installing packages to custom, non-default locations, see Table 9 on page 12.)

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

```
rpm -qpil package_name
```

Table 4. IBM XL C/C++ for Multicore Acceleration for Linux packages and default installation locations

Package Name	Package Description	Default installation location
ppu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (PPU only)	/opt/ibmcmp/xlmass/cbe/4.5
spu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (SPU only)	/opt/ibmcmp/xlmass/cbe/4.5
cell-xlc-rte	IBM XL C/C++ runtime package	/opt/ibmcmp/lib/cbe/ /opt/ibmcmp/lib64/cbe/
cell-xlc-rte-lnk	IBM XL C/C++ runtime links package	/opt/ibmcmp/xlc/cbe/9.0/
cell-xlc-lic	IBM XL C/C++ license package	/opt/ibmcmp/xlc/cbe/9.0/
cell-xlc-lib	IBM XL C++ libraries package	/opt/ibmcmp/xlc/cbe/9.0/
cell-xlc-cmp	IBM XL C++ compiler package	/opt/ibmcmp/xlc/cbe/9.0/
cell-xlc-help	IBM XL C/C++ help documentation package	/opt/ibmcmp/xlc/cbe/9.0/doc/
cell-xlc-man	IBM XL C/C++ man pages package	/opt/ibmcmp/xlc/cbe/9.0/man/

National language support

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 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

English is the default national language and en_US is the default locale. Following installation, you can set the NLSPATH so that messages are displayed in a different language. See “Enabling the IBM XL C/C++ for Multicore Acceleration for Linux error messages” on page 25.

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.

  Note: IBM XL C/C++ Advanced Edition for Linux is not considered to be the same product as IBM XL C/C++ for Multicore Acceleration for Linux. Therefore, IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 can co-reside with compiler products for Linux with no additional installation requirements. However, IBM XL C/C++ Advanced Edition for Linux runtime libraries and IBM XL C/C++ for Multicore Acceleration for Linux runtime libraries share a common name. Setting the ***LD_LIBRARY_PATH*** environment variable may lead to the incorrect runtime library being called. Setting the ***LD_LIBRARY_PATH*** in this scenario is unsupported.
- 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 any previously installed IBM XL C/C++ for Multicore Acceleration for Linux compiler, install the latest version, and configure the compiler, all through the use of a single installation tool.

If you are installing IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 for the first time, perform the steps listed in Table 5.

If you are installing an update to IBM XL C/C++ for Multicore Acceleration for Linux, V9.0, perform the steps listed in Table 6 on page 4.

Table 5. Steps for basic installation

Task	**For more information, see . . .**
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system.
Ensure that all system prerequisites are satisfied.	“System prerequisites” on page 5
Install the runtime package on the target system.	“Installing cell-xlc-rte on the target system” on page 7
Uninstall any previously installed packages of the same product type.	Chapter 2, “Basic installation,” on page 7
Use the xlc_install tool to accept or decline the license.	Chapter 2, “Basic installation,” on page 7
Use the xlc_install tool to install and configure the compiler, using the default paths.	Chapter 2, “Basic installation,” on page 7
Confirm that the compiler packages were successfully installed, and test the installation.	• “Querying for installed packages” on page 23 • “Testing the installation” on page 23

Table 5. Steps for basic installation (continued)

Task	For more information, see . . .
Enable the compiler man pages.	"Enabling the man pages" on page 24
(Optional) If you did not choose to create symbolic links to the compiler invocation commands during the installation process, set up the environment to locate the invocation commands without the full path. Otherwise, you can skip this step.	"Setting up the environment for the invocation commands" on page 26

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.
Install the runtime package on the target system.	"Installing cell-xlc-rte on the target system" on page 7
Use the xlc_install tool to install the update packages.	"Running the xlc_install utility to update a basic installation" on page 15
Confirm that the compiler packages were successfully installed, and test the installation.	• "Querying for installed packages" on page 23 • "Testing the installation" on page 23
(Optional) If you did not choose to create symbolic links to the compiler invocation commands during the update process, set up the environment to locate the invocation commands without the full path. Otherwise, you can skip this step.	"Setting up the environment for the invocation commands" on page 26

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.

When installing IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 for the first time, perform the steps listed in Table 7.

Table 7. Steps for advanced 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 IBM XL C/C++ for Multicore Acceleration for Linux.	Chapter 7, "Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V9.0," on page 29
Ensure that all system prerequisites are satisfied.	"System prerequisites" on page 5

Table 7. Steps for advanced installation (continued)

Task	For more information, see . . .
Install the runtime package on the target system.	"Installing cell-xlc-rte on the target system" on page 7
Use one of the advanced installation methods to install the compiler.	Chapter 3, "Advanced installation," on page 11
Use the **new_install** or **xlc_configure** tool to configure the compiler.	Chapter 5, "Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 (for advanced users)," on page 19
Confirm that the compiler packages were successfully installed, and test the installation.	• "Querying for installed packages" on page 23 • "Testing the installation" on page 23
Enable the compiler man pages.	"Enabling the man pages" on page 24
(Optional) Set up the environment to locate the invocation commands without the full path.	"Setting up the environment for the invocation commands" on page 26

System prerequisites

The following are the requirements for installing IBM XL C/C++ for Multicore Acceleration for Linux, V9.0:

- **Operating system:**
  - Red Hat Enterprise Linux 5.1 (RHEL5.1)
- **Hardware for IBM POWER technology-based systems:**
  - IBM System p™ technology-based system: 64-bit PPC
  - Approximately 200 MB (minimum) of hard drive space
  - 256 MB (minimum) of RAM

  Note: High levels of optimization can require more space for paging and temporary files.
- **Hardware for Intel® x86 Systems:**
  - Either x86 or x86-64 2-GHz Pentium® 4 processor (minimum)
  - Approximately 200 MB (minimum) of hard drive space
  - 256 MB (minimum) of RAM

  Note: High levels of optimization can require more space for paging and temporary files.
- **Storage:** To verify that you have enough hard disk space available, see the procedure in "Verifying the amount of hard disk space available" on page 6.
- **Software:**

Table 8. Required GNU and Perl packages for the RHEL5.1 operating system

Package name	Version requirements
gcc	4.1.1
gcc-c++	4.1.1
glibc	2.5
libgcc	4.1.1
libstdc++	4.1.1

Table 8. Required GNU and Perl packages for the RHEL5.1 operating system (continued)

IBM Software Development Kit (SDK) for Multicore Acceleration Version 3.0	3.0
Perl	5.0 or greater **Note:** Perl V5.8 is shipped and automatically installed with the RHEL5 operating system.

Verifying the amount of hard disk space available

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 requires about 300 MB of hard disk storage space. This amount accommodates the optional samples and documentation that are shipped with the product.

You can use the following command to determine the amount of space available in the default installation location (/opt/ibmcmp/xlc/cbe/):

```
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 and Perl packages are installed

Before you can install IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 you should verify that the required versions of GNU and Perl packages were installed with the operating system.

For a list of the required packages:

- Table 8 on page 5: Required GNU and Perl packages for the RHEL5.1 operating system

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

```
gcc-c++-4.1.1-43.24
```

Chapter 2. Basic installation

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

- Accept or decline the license agreement. If you accept the agreement, the license files will be output to .txt files for your future reference. If you decline the agreement, the installation process will exit without installing the compiler and no files will have been written to your machine.

You can use xlc_install to do the following:

- Install IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 on a system with no IBM XL compiler currently installed.

You should use the xlc_install utility to install IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 as long as *both* 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 IBM 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 11.

Installing cell-xlc-rte on the target system

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is a cross-compiler. Therefore, you will need to install the runtime library package, cell-xlc-rte, on the target system.

Issue the following command:

IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0

```
rpm -ivh cell-xlc-rte-9.0.0-0.i386.rpm --prefix $RTEpath
```

IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0

```
rpm -ivh cell-xlc-rte-9.0.0-0.ppc64.rpm --prefix $RTEpath
```

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 and Perl packages are installed" on page 6.

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 IBM XL C/C++ components
- Installs all compiler packages into the default location
- Automatically invokes the **new_install** utility, which installs the license file and generates the default configuration file

- Optionally creates symbolic links in /usr/bin/ to the compiler invocation commands
- Generates an installation log in the /tmp/ directory

To run the xlc_install utility to install IBM XL C/C++ for Multicore Acceleration for Linux, V9.0:

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 9.

   - If another instance of IBM XL C/C++ for Multicore Acceleration for Linux is detected on your system, you are prompted to uninstall it. Confirm that you want to proceed with the uninstallation. If you choose not to uninstall the existing instance of the compiler, the installation process will end.
   - If other versions of the IBM MASS packages are detected on your system you are prompted to uninstall them. Confirm that you want to proceed with uninstalling the existing IBM MASS packages. If you choose not to uninstall the previously installed components, the installation process will terminate.
2. You are presented with the licensing agreement and licensing information. Read the licensing agreement and licensing information. If you agree to the licensing terms, accept the license agreement and licensing information to continue installation.

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

   Note: An alternative to this step is to add the path that contains the compiler invocations to the PATH environment variable. See "Setting the PATH environment variable to include the path to the compiler invocations" on page 26.

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

   - ppuxlc++
   - ppuxlC
   - ppuxlC_r
   - ppuxlc++_r
   - spuxlc++
   - spuxlC
   - ppuxlc
   - ppuxlc_r
   - spuxlc
   - ppuc89
   - ppuc89_r
   - ppuc99
   - ppuc99_r
   - ppucc
   - ppucc_r

- ppugxlc
- spugxlc
- ppugxlc++
- spugxlc++
- ppugxlC
- spugxlC

If all packages are successfully installed:

- A message is displayed confirming the successful installation.
- The configuration file is generated. Its location is /opt/ibmcmp/xlc/cbe/9.0/etc/vac.cfg. Any previously generated configuration file is renamed and saved in the same directory.
- The installation log is moved to its permanent location: /opt/ibmcmp/xlc/cbe/9.0/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.

- **IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0**./RHEL5-PPC/rpms/ (RHEL5.1)
- **IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0**./RHEL5-X86/rpms/ (RHEL5.1)

-U Updates the compiler to the *Version.Release.Modification-Fix* (V.R.M-F) level that the installation utility version supports.

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

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

Chapter 3. Advanced installation

It is highly recommended that you install IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 to the default location and use the procedure provided in Chapter 2, "Basic installation," on page 7. However, you will need to use alternate procedures for customized scenarios, including the following:

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

In all of these scenarios, you must use the **rpm** utility to install the compiler; you cannot use the xlc_install utility to do so. Once you have successfully installed the compiler to a non-default location, you will need to manually configure the compiler environment using the **new_install** or **xlc_configure** utilities; see Chapter 5, "Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 (for advanced users)," on page 19 for procedures.

Installing cell-xlc-rte on the target system

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is a cross-compiler. Therefore, you will need to install the runtime library package, cell-xlc-rte, on the target system.

Issue the following command:

IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0

```
rpm -ivh cell-xlc-rte-9.0.0-0.i386.rpm --prefix $RTEpath
```

IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0

```
rpm -ivh cell-xlc-rte-9.0.0-0.ppc64.rpm --prefix $RTEpath
```

Installing all packages to a single non-default location

To install all compiler packages to a single non-default directory, ensure that your current working directory contains all of the packages for IBM XL C/C++ for Multicore Acceleration for Linux 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 9 provides information about which packages must be installed together in the same directory and which can be installed in any directory.

Table 9. Rules for installing packages to multiple, non-default locations

Package Name	Package Description	Rules for installation to a non-default location
ppu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (PPU)	Any location. For the remainder of this document, the name *xlmass_path* is used to refer to this location.
spu-xlmass-lib	IBM Mathematical Acceleration Subsystem (MASS) package (SPU)	Any location. For the remainder of this document, the name *xlmass_path* is used to refer to this location.
cell-xlc-rte	IBM XL C/C++ runtime package	All IBM XL C/C++ runtime packages must be installed in the same location. For the remainder of this document, the name *xlrte_path* is used to refer to this location.
cell-xlc-rte-lnk	IBM XL C/C++ runtime links package	
cell-xlc-lic	IBM XL C/C++ license package	Any location. For the remainder of this document, the name *lic_path* is used to refer to this location.
cell-xlc-lib	IBM XL C/C++ compiler libraries package	All IBM XL C/C++ compiler and library packages must be installed in the same location. For the remainder of this document, the name *xlcmp_path* is used to refer to this location.
cell-xlc-cmp	IBM XL C/C++ compiler package	
cell-xlc-help	IBM XL C/C++ help documentation package	Any location (optional). For the remainder of this document, the name *doc_path* is used to refer to this location.
cell-xlc-man	IBM XL C/C++ compiler man pages	Any location (optional). For the remainder of this document, the name *manpag_path* 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 9.

Example : Installing IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 to multiple non-default directories

In order to avoid dependency errors during installation of IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 issue the following commands in the order given:

```
rpm -ivh ppu-xlmass-lib-4.5.0-0.rpm --prefix $MASS_path
rpm -ivh spu-xlmass-lib-4.5.0-0.rpm --prefix $MASS_path
```

IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0

```
rpm -ivh cell-xlc-rte-9.0.0-0.i386.rpm --prefix $RTEpath
rpm -ivh cell-xlc-rte-lnk-9.0.0-0.i386.rpm --prefix $RTEpath
rpm -ivh cell-xlc-lic-9.0.0-0.i386.rpm --prefix $LICpath
rpm -ivh cell-xlc-lib-9.0.0-0.i386.rpm --prefix $CMPpath
rpm -ivh cell-xlc-cmp-9.0.0-0.i386.rpm --prefix $CMPpath
```

Note:

> IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is a cross-compiler. Therefore, you will need to also install the runtime library rpm, cell-xlc-rte, on the target system.

IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0

```
rpm -ivh cell-xlc-rte-9.0.0-0.ppc64.rpm --prefix $RTEpath
rpm -ivh cell-xlc-rte-lnk-9.0.0-0.ppc64.rpm --prefix $RTEpath
rpm -ivh cell-xlc-lic-9.0.0-0.ppc64.rpm --prefix $LICpath
rpm -ivh cell-xlc-lib-9.0.0-0.ppc64.rpm --prefix $CMPpath
rpm -ivh cell-xlc-cmp-9.0.0-0.ppc64.rpm --prefix $CMPpath
```

Note:

> IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is a cross-compiler. Therefore, you will need to also install the runtime library package, cell-xlc-rte, on the target system.

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

IBM XL C/C++ for Multicore Acceleration for Linux on x86 Systems, V9.0

```
rpm -ivh cell-xlc-man-9.0.0-0.i386.rpm --prefix $MANPAGpath
rpm -ivh cell-xlc-help-9.0.0-0.i386.rpm --prefix $DOCpath
```

IBM XL C/C++ for Multicore Acceleration for Linux on System p, V9.0

```
rpm -ivh cell-xlc-man-9.0.0-0.ppc64.rpm --prefix $MANPAGpath
rpm -ivh cell-xlc-help-9.0.0-0.ppc64.rpm --prefix $DOCpath
```

Chapter 4. Installing an update

An update of IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 provides a fix or multiple fixes to the product. You can download updates from the support web site: http://www.ibm.com/software/awdtools/xlcpp/support

Every PTF update package comes in tarball format and includes a version of the xlc_install utility that is customized to install only the update that accompanies it. If you have any version (including an earlier update) of IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 installed on your system, you can apply the latest update.

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 cell-xlc-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 IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 when all the following conditions have been met:

- The base version of IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is already successfully installed in the /opt/ibmcmp/ directory.
- The update package has been uncompressed and unpacked in the /home/root/ directory of the system.

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

- Checks for all prerequisite software packages
- Uninstalls IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 packages
- Installs updated compiler packages into the default location
- Automatically invokes the **new_install** utility, which installs the license file, renames the old configuration file, and generates a new configuration file
- Optionally creates symbolic links in /usr/bin/ to the compiler invocation commands
- Generates an installation log in the /tmp/ directory

To run the xlc_install utility to apply an update for IBM XL C/C++ for Multicore Acceleration for Linux, V9.0:

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, `sep2007` indicates a shipment date in September of 2007.)

2. Issue the following command:

   ```
   ./xlc_install -U
   ```

   For additional arguments that you can specify for xlc_install, see "xlc_install options" on page 9.

   You are prompted to uninstall any previously installed IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 packages.
3. Confirm that you want to proceed with uninstalling the outdated packages.

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

   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 26.

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

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

- spugxlC

7. If you customized the previously generated configuration file, manually edit `/opt/ibmcmp/xlc/cbe/9.0/etc/vac.cfg` to replicate those changes in the newly generated configuration file.

Chapter 5. Configuring IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 (for advanced users)

Before you can run IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 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 in a non-default location, or compiler components were relocated after installation.

Two configuration tools are provided with the compiler: **new_install** and **xlc_configure**, both located in the *installation_path*/xlc/cbe/9.0/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 IBM XL C/C++ for Multicore Acceleration for Linux is installed on your system.
- Only one version of the SDK GCC is installed in your system and it is installed in its default location.
- You have root or administrator privileges.
- You want to generate the configuration file in the default directory
  ```
  /opt/ibmcmp/xlc/cbe/9.0/etc/
  ```

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

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

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

  Note: The default location is
  ```
  /opt/ibmcmp/xlc/cbe/9.0/etc/
  ```
- You have multiple versions of SDK GCC installed on your system and you need to specify which SDK GCC version you would like to reference in the configuration file.

Note: If you configure the compiler using **xlc_configure**, your output configuration file, vac.cfg, can be written to any location where you have write permission. You would not need root or administrative privileges.

For instructions, see “Running the **xlc_configure** utility directly” on page 20.

Running the new_install utility

The **new_install** utility does the following:

- Backs up any existing configuration file.
- Installs the license file.
- Generates the configuration file in the default location `/opt/ibmcmp/xlc/cbe/9.0/etc/vac.cfg`.

To run the **new_install** utility:

1. Change to the directory that contains the compiler executables:
   ```
   cd  installation_path/xlc/cbe/9.0/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 license agreement and licensing information. If you agree to the licensing terms, accept the license agreement and licensing information.

Running the xlc_configure utility directly

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

To run the **xlc_configure** utility:

1. Change to the directory that contains the compiler executables:
   ```
   cd  installation_path/xlc/cbe/9.0/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:
   ```
   ./xlc_configure options
   ```

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

xlc_configure options

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

```
►►──xlc_configure──┬─-spugcc32─┬──┬─-o──name─┬──┬────────┬──────────────────►
                   ├─-ppugcc32─┤  └─-install─┘  └─-force─┘
                   └─-ppugcc64─┘
```

```
►──┬─-ibmcmp──installation_path──────────────────┬──┬──────────────────────────┬──►◄
   │  ┌─────────────────────────────────────────┐│  └─template_config_filename─┘
   │  ▼                                          ││
   └────┬─-mass──xlmass_path/xlmass/4.5/cbe─┬───┴┘
        ├─-xlclic──lic_path/xlc/9.0/cbe─────┤
        ├─-xlcrt──xlrte_path/xlc/9.0/cbe────┤
        └─-xlc──xlcmp_path/xlc/9.0/cbe──────┘
```

where:

-h Displays the **xlc_configure** options help page.

-ppugcc32 *path*
Path where the PPU toolchain GCC is installed. In the configuration file, the `gcc_path` attribute is set to equal this path. For example, if the PPU-GCC command is **/usr/bin/ppu32-gcc**, you would specify:

```
-ppugcc32 /usr
```

-ppugcc64 *path*
Specifies the path where the 64-bit PPU toolchain GCC is installed. In the configuration file, the gcc_path_64 attribute is set to equal this path. For example, if the 64-bit PPU-GCC command is **/usr/bin/ppu-gcc** , you would specify:

```
-ppugcc64 /usr
```

-spugcc32 *path*
Specifies the path where the SPU toolchain GCC is installed. In the configuration file, the SPU default gcc_path attribute is set to equal this path. For example, if the SPU-GCC command is **/usr/bin/spu-gcc** , you would specify:

```
-spugcc32 /usr
```

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

-install
Generates the configuration file as /opt/ibmcmp/xlc/cbe/9.0/etc/**vac**.cfg. By default, output is written to the display only.

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

-ibmcmp *installation_path*
Specifies the path where all of the IBM XL C/C++ for Multicore Acceleration for Linux packages are installed (if all packages are installed in the same path). By default, the path is /opt/ibmcmp/.

-mass *xlmass_path***/xlmass/cbe/4.5/**
Specifies the path where the xlmass package is installed. By default, the full path is /opt/ibmcmp/xlmass/cbe/4.5/.

-xlclic *lic_path***/xlc/cbe/9.0/**
Specifies the path where the cell-xlc-lic package is installed. By default, the full path is /opt/ibmcmp/xlc/cbe/9.0/.

-xlcrt *xlrte_path***/xlc/cbe/9.0/**

Specifies the path where the C++ runtime is installed. By default, the full path is /opt/ibmcmp/xlc/cbe/9.0/.

-xlc *xlcmp_path***/xlc/cbe/9.0/**

Specifies the path where the C/C++ compiler is installed. By default, the full path is /opt/ibmcmp/xlc/cbe/9.0/.

template_config_file_name

The input file that is used to construct the configuration file. By default, this is /opt/ibmcmp/xlc/cbe/9.0/etc/vac.base.cfg. If you relocated the cell-xlc-cmp package to *xlcmp_path* but want to use the default template, specify:

```
xlcmp_path/xlc/cbe/9.0/etc/vac.base.cfg
```

Chapter 6. After installing IBM XL C/C++ for Multicore Acceleration for Linux, V9.0

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 24: This section applies to all users.
- "Enabling the IBM XL C/C++ for Multicore Acceleration for Linux error messages" on page 25: 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 26: This section applies only to users who did not use xlc_install to install or update the product, or who did not create symbolic links during the installation process with xlc_install.

Querying for installed packages

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

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

The result should be:

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

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

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

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

```
rpm -qa | grep -e cell-xlc  -e ppu-(spu-)xlmass
```

The result should be a list that contains all of the packages listed in Table 4 on page 2. 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"

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

1. Create the following C program and name the source file `hello.c`:

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

2. Compile the program:
   - If short invocation commands have been set up, enter the following command:

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

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

     ```
     /opt/ibmcmp/xlc/cbe/9.0/bin/ppuxlc hello.c -o hello
     ```
3. Run the program on the target system by entering the following command:

   ```
   ./hello
   ```

   The result should be ″Hello World!″.
4. Check the exit code of the program by entering the following command:

   ```
   echo $?
   ```

   The result should be 0.
5. Create the following C++ program and name the source file `hello.cpp`:

   ```
   #include <stdio.h>
   int main(void)
   {
      printf("Hello World!\n");
      return 0;
   }
   ```
6. Compile the program:
   - If short invocation commands have been set up, enter the following command:

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

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

     ```
     /opt/ibmcmp/xlc/cbe/9.0/bin/ppuxlc++ hello.cpp -o hello
     ```
7. Run the program on the target system:

   ```
   ./hello
   ```

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

   ```
   echo $?
   ```

   The result should be “0”.

Enabling the man pages

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

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 man pages support the following locales:

- en_US
- en_US.utf8

However, before you can read the compiler-supplied man pages, you must add the full directory path to the MANPATH environment variable. The command depends on the Linux shell you are using.

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

```
export MANPATH=installation_path/xlc/cbe/9.0/man/LANG:$MANPATH
```

where LANG is either of the language locales listed above.

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

```
setenv MANPATH  installation_path/xlc/cbe/9.0/man/LANG:$MANPATH
```

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

Note: To set this variable in the Bourne, Korn, or BASH shell so that it applies to all users, add the commands to the file `/etc/profile`. To set it for a specific user only, add the commands to the file `.profile` in the user's home directory. 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 IBM XL C/C++ for Multicore Acceleration for Linux 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.

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

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

```
export NLSPATH=$NLSPATH:xlcmp_path/xlc/cbe/9.0/msg/%L/%N
```

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

```
setenv NLSPATH $NLSPATH:xlcmp_path/xlc/cbe/9.0/msg/%L/%N
```

where:

- *xlcmp_path* is the installation location of the IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 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 xlc_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 IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 commands without typing the complete path, you can add the location of the compiler invocations to the PATH environment variable.

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

```
export PATH=$PATH:installation_path/xlc/cbe/9.0/bin/
```

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

```
setenv PATH $PATH:installation_path/xlc/cbe/9.0/bin/
```

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

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

Creating symbolic links to the compiler invocations

To use the compiler without typing the complete path, you can create symbolic links in the /usr/bin/ directory for the specific invocations contained in the *installation_path*/xlc/cbe/9.0/bin/ directory.

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

- ppuxlc++
- ppuxlC
- ppuxlC_r
- ppuxlc++_r

- spuxlc++
- spuxlC
- ppuxlc
- ppuxlc_r
- spuxlc
- ppuc89
- ppuc89_r
- ppuc99
- ppuc99_r
- ppucc
- ppucc_r
- ppugxlc
- spugxlc
- ppugxlc++
- spugxlc++
- ppugxlC
- spugxlC

Use the following command to create a symbolic link:

```
ln –s installation_path/xlc/cbe/9.0/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 **ppuxlc++**) in *installation_path*/xlc/cbe/9.0/bin/.

Basic example: Creating a symbolic link to a compiler invocation

This example assumes that the entire IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is installed in the default location /opt/ibmcmp/.

```
ln –s /opt/ibmcmp/xlc/cbe/9.0/bin/ppuxlc /usr/bin/ppuxlc
```

Enabling IBM Tivoli License Compliance Manager

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

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is ITLCM-enabled for inventory support only, which means that ITLCM is able to detect product installation of IBM XL C/C++ for Multicore Acceleration for Linux, but not its usage.

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

Once installed and activated, ITLCM scans your system for product inventory signatures that indicate whether a given product is installed on your system.

ITLCM also identifies the version, release, and modification levels of the product. Inventory signature files are not updated after a PTF update package is installed.

If IBM XL C/C++ for Multicore Acceleration for Linux is installed in the default location, the signature files are in the /opt/ibmcmp/xlc/cbe/9.0/ directory. For more information about IBM Tivoli License Compliance Manager see: http://www.ibm.com/software/tivoli/products/license-mgr/.

Accessing the documentation

Help using IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is available in PDF format. Manual pages for the compiler invocation commands and other command utilities are also included.

Viewing the PDF documentation

PDF versions of the IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 product manuals are available in the `/doc/`*`en_US`*`/pdf/` directory of the installation media.

After default installation, the PDF documentation can be found in the `/opt/ibmcmp/xlc/cbe/9.0/doc/en_US/pdf/` directory. For non-default installations, the PDF documentation is located in the *`$target_dir`*`/xlc/cbe/9.0/doc/en_US/pdf/` directory.

Chapter 7. Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V9.0

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 does not provide a standalone uninstallation tool. You must use the Linux **rpm** utility to uninstall IBM XL C/C++ for Multicore Acceleration for Linux, V9.0.

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 help and man documentation does not have any package dependencies. You can remove them in any order.
4. You cannot uninstall packages that are required by other packages.
5. The uninstallation commands will not remove any configuration files that were generated by **new_install** or **xlc_configure**.

Example : Uninstalling IBM XL C/C++ for Multicore Acceleration for Linux, V9.0

In this example:

- The compiler packages have a *V.R.M-F* of 9.0.0-0
- The IBM MASS library package has a *V.R.M-F* of 4.5.0–0.

Note:

IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 is a cross-compiler and you will have needed to install the cell-xlc-rte rpm package on the target system. Therefore, you will also uninstall the runtime library package, cell-xlc-rte, from the target system.

To uninstall IBM XL C/C++ for Multicore Acceleration for Linux, V9.0 issue the following commands, in the same order:

```
rpm -e cell-xlc-cmp-9.0.0-0
rpm -e cell-xlc-lib-9.0.0-0

rpm -e cell-xlc-lic-9.0.0-0
rpm -e cell-xlc-rte-lnk-9.0.0-0
rpm -e cell-xlc-rte-9.0.0-0

rpm -e spu-xlmass-lib-4.5.0-0
rpm -e ppu-xlmass-lib-4.5.0-0
```

You can issue the following commands in any order:

```
rpm -e cell-xlc-man-9.0.0-0
rpm -e cell-xlc-help-9.0.0-0
```

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 IBM XL C/C++ for Multicore Acceleration for Linux, V9.0.

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 "xlc_install options" on page 9.

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 2 in the path before running the xlc_install utility again. You might need to use the **-rpmloc** *rpmlocation_path* option if you have moved the xlc_install utility to a different location than the one provided in the installation image. For more information, see "xlc_install options" on page 9.

Could not determine location of 32-bit or 64-bit SDK GCC (RHEL5.1)

Scenario

You are running either the **new_install** or the **xlc_configure** utility to configure the compiler on a computer running RHEL5.1 when you get at least one of the following error messages:

```
ERROR: File "<path>/ppu32-gcc" not found
ERROR: File "<path>/ppu32-g++" not found
ERROR: File "<path>/ppu-gcc" not found
ERROR: File "<path>/ppu-g++" not found
ERROR: File "<path>/spu-gcc" not found
ERROR: File "<path>/spu-g++" not found
```

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

Explanation

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

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

To check if the RPMs are already installed:

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

Action

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

Notices

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

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

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

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

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

Trademarks and service marks

Company, product, or service names identified in the text may be trademarks or service marks of IBM or other companies. Information on the trademarks of International Business Machines Corporation in the United States, other countries, or both is located at http://www.ibm.com/legal/copytrade.shtml.

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

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

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

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

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

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

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 9899-1990).
- The C language is also 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)).

Index

Program Number: 5724-T42 & 5724-T43

GC23-8520-00