IBM XL C/C++ for Linux, V12.1

IBM

Installation Guide

Version 12.1

IBM XL C/C++ for Linux, V12.1

Installation Guide

Version 12.1

First edition

This edition applies to IBM XL C/C++ for Linux, V12.1 (Program 5765-J03; 5725-C73) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Contents

About this information

This guide provides detailed installation instructions for IBM® XL C/C++ for Linux, V12.1. It guides you through multiple ways to perform tasks and directs you to reference information in cases requiring atypical installations. It also shows you how to test the installation, launch remotely-accessible HTML help, and enable and view different types of documentation. Please read it carefully before installing. Please also read the README file in the root directory of your installation media, which contains current information about the compiler.

Who should read this document

This document is intended for anyone responsible for installing IBM XL C/C++ for Linux, V12.1.

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

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

How to use this document

This document provides procedures for three main installation scenarios:

"Basic" installation

> This scenario allows you to install a single version of XL C/C++ to a default location. This scenario is applicable to the majority of users, and is the recommended method of installation.
>
> For an overview of the steps that you need to follow to perform a basic installation, refer to Chapter 2, "Basic installation," on page 9.

"Advanced" installation

> This scenario allows you to maintain multiple versions of XL C/C++ on a single system, or to install the compiler to a non-default location. This scenario is only applicable to advanced users who have specialized needs, and is not recommended for the majority of users.
>
> For an overview of the steps that you need to follow to perform an advanced installation, refer to Chapter 3, "Advanced installation," on page 13.

"Update" installation

> This scenario applies to users who have obtained a Program Temporary Fix (PTF) package for an existing XL C/C++ V12.1 installation.
>
> For an overview of the steps required to install a PTF, refer to Chapter 4, "Installing an update," on page 21.

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

How this document is organized

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

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

Phase	Chapters	User segment
Pre-installation	Chapter 1, "Before installing IBM XL C/C++ for Linux, V12.1," on page 1	All users
Installation	Chapter 2, "Basic installation," on page 9	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 compiler.
	Chapter 3, "Advanced installation," on page 13	Users who: • Want to install the compiler in a non-default location • Want to have multiple versions of the compiler on the same system
Post-installation	Chapter 4, "Installing an update," on page 21	Users who want to update XL C/C++ V12.1 to the next fix level
	Chapter 6, "After installing IBM XL C/C++ for Linux, V12.1," on page 29	All users
Product removal	Chapter 7, "Uninstalling IBM XL C/C++ for Linux, V12.1," on page 37	Any user who needs to remove an XL C/C++ compiler from the system

Conventions

Typographical conventions

The following table explains the typographical conventions used in the IBM XL C/C++ for Linux, V12.1 information.

Table 2. Typographical conventions

Typeface	Indicates	Example
bold	Lowercase commands, executable names, compiler options, and directives.	The compiler provides basic invocation commands, **xlc** and **xlC** (**xlc++**), along with several other compiler invocation commands to support various C/C++ language levels and compilation environments.

Table 2. Typographical conventions (continued)

Typeface	Indicates	Example
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.
<u>underlining</u>	The default setting of a parameter of a compiler option or directive.	nomaf | <u>maf</u>
monospace	Programming keywords and library functions, compiler builtins, examples of program code, command strings, or user-defined names.	To compile and optimize myprogram.c, enter: `xlc myprogram.c -03`.

Qualifying elements (icons)

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

Table 3. Qualifying elements

Qualifier/Icon	Meaning
C only, or C only begins ▶ C C ◀ C only ends	The text describes a feature that is supported in the C language only; or describes behavior that is specific to the C language.
C++ only, or C++ only begins ▶ C++ C++ ◀ C++ only ends	The text describes a feature that is supported in the C++ language only; or describes behavior that is specific to the C++ language.
IBM extension begins ▶ IBM IBM ◀ IBM extension ends	The text describes a feature that is an IBM extension to the standard language specifications.
C1X, or C1X begins ▶ C1X C1X ◀ C1X ends	The text describes a feature that is introduced into standard C as part of C1X.
C++0x, or C++0x begins ▶ C++0x C++0x ◀ C++0x ends	The text describes a feature that is introduced into standard C++ as part of C++0x.

Syntax diagrams

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

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

    The ►►─── symbol indicates the beginning of a command, directive, or statement.

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

    The ►─── symbol indicates that a command, directive, or statement is continued from the previous line.

    The ──→◄ symbol indicates the end of a command, directive, or statement.

    Fragments, which are diagrams of syntactical units other than complete commands, directives, or statements, start with the |─── symbol and end with the ───| symbol.

- Required items are shown on the horizontal line (the main path):

    ►►──keyword──*required_argument*────────────────────────────────────►◄

- Optional items are shown below the main path:

    ►►──keyword──►◄
            └─*optional_argument*─┘
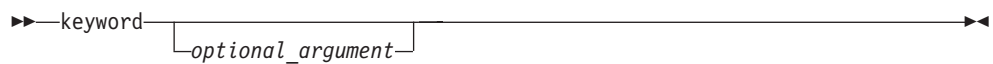
- If you can choose from two or more items, they are shown vertically, in a stack.

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

    ►►──keyword──┬─*required_argument1*─┬────────────────────►◄
              └─*required_argument2*─┘
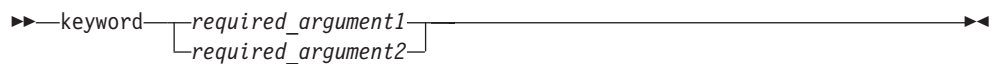
    If choosing one of the items is optional, the entire stack is shown below the main path.

    ►►──keyword──►◄
            ├─*optional_argument1*─┤
            └─*optional_argument2*─┘

- 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:

              ┌─,─────┐
    ►►──keyword──▼─*repeatable_argument*─┴────────────────►◄

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

             ┌─*default_argument*───┐
   ►►──keyword──┴─*alternate_argument*─┴────────────────►◄

- Keywords are shown in nonitalic letters and should be entered exactly as shown.

- Variables are shown in italicized lowercase letters. They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.

Sample syntax diagram

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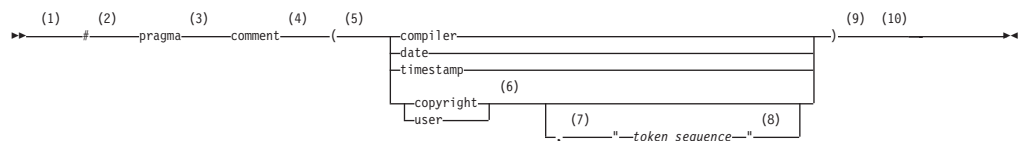
Notes:

1   This is the start of the syntax diagram.

2   The symbol # must appear first.

3   The keyword `pragma` must appear following the # symbol.

4   The name of the pragma `comment` must appear following the keyword `pragma`.

5   An opening parenthesis must be present.

6   The comment type must be entered only as one of the types indicated: `compiler`, `date`, `timestamp`, `copyright`, or `user`.

7   A comma must appear between the comment type `copyright` or `user`, and an optional character string.

8   A character string must follow the comma. The character string must be enclosed in double quotation marks.

9   A closing parenthesis is required.

10   This is the end of the syntax diagram.

The following examples of the **#pragma comment** directive are syntactically correct according to the diagram shown above:

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

Examples in this information

The examples in this information, except where otherwise noted, are coded in a simple style that does not try to conserve storage, check for errors, achieve fast performance, or demonstrate all possible methods to achieve a specific result.

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

Related information

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

IBM XL C/C++ information

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

- README files

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

- Installable man pages

  Man pages are provided for the compiler invocations and all command-line utilities provided with the product. Instructions for installing and accessing the man pages are provided in the *IBM XL C/C++ for Linux, V12.1 Installation Guide*.

- Information center

  The information center of searchable HTML files can be launched on a network and accessed remotely or locally. Instructions for installing and accessing the online information center are provided in the *IBM XL C/C++ for Linux, V12.1 Installation Guide*.

  The information center is viewable on the web at http://publib.boulder.ibm.com/infocenter/lnxpcomp/v121v141/index.jsp.

- PDF documents

  PDF documents are located by default in the /opt/ibmcmp/vacpp/12.1/doc/*LANG*/pdf/ directory, where *LANG* is one of en_US, zh_CN, or ja_JP. The PDF files are also available on the web at http://www.ibm.com/software/awdtools/xlcpp/linux/library/.

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

Table 4. XL C/C++ PDF files

Document title	PDF file name	Description
IBM XL C/C++ for Linux, V12.1 Installation Guide, GC14-7340-00	install.pdf	Contains information for installing XL C/C++ and configuring your environment for basic compilation and program execution.
Getting Started with IBM XL C/C++ for Linux, V12.1, SC14-7339-00	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.
IBM XL C/C++ for Linux, V12.1 Compiler Reference, SC14-7341-00	compiler.pdf	Contains information about the various compiler options, pragmas, macros, environment variables, and built-in functions, including those used for parallel processing.
IBM XL C/C++ for Linux, V12.1 Language Reference, SC14-7342-00	langref.pdf	Contains information about the C and C++ programming languages, as supported by IBM, including language extensions for portability and conformance to nonproprietary standards.
IBM XL C/C++ for Linux, V12.1 Optimization and Programming Guide, SC14-7343-00	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.

To read a PDF file, use the Adobe Reader. If you do not have the Adobe Reader,

you can download it (subject to license terms) from the Adobe website at
http://www.adobe.com.

More information related to XL C/C++ including IBM Redbooks® publications,
white papers, tutorials, and other articles, is available on the web at:

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

For more information about boosting performance, productivity, and portability,
see the C/C++ café at http://www.ibm.com/software/rational/cafe/community/
ccpp.

Standards and specifications

XL C/C++ is designed to support the following standards and specifications. You
can refer to these standards for precise definitions of some of the features found in
this information.

- *Information Technology - Programming languages - C, ISO/IEC 9899:1990*, also
  known as *C89*.
- *Information Technology - Programming languages - C, ISO/IEC 9899:1999*, also
  known as *C99*.
- *Information Technology - Programming languages - C++, ISO/IEC 14882:1998*, also
  known as *C++98*.
- *Information Technology - Programming languages - C++, ISO/IEC 14882:2003(E)*, also
  known as *Standard C++*.
- *Information Technology - Programming languages - Extensions for the programming
  language C to support new character data types, ISO/IEC DTR 19769*. This draft
  technical report has been accepted by the C standards committee, and is
  available at http://www.open-std.org/JTC1/SC22/WG14/www/docs/
  n1040.pdf.
- *Draft Technical Report on C++ Library Extensions, ISO/IEC DTR 19768*. This draft
  technical report has been submitted to the C++ standards committee, and is
  available at http://www.open-std.org/JTC1/SC22/WG21/docs/papers/2005/
  n1836.pdf.
- *AltiVec Technology Programming Interface Manual*, Motorola Inc. This specification
  for vector data types, to support vector processing technology, is available at
  http://www.freescale.com/files/32bit/doc/ref_manual/ALTIVECPIM.pdf.
- *ANSI/IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985*.
- *OpenMP Application Program Interface Version 3.1*, available at
  http://www.openmp.org

Other IBM information

- *ESSL for AIX® V5.1/ESSL for Linux on POWER® V5.1 Guide and Reference* available
  at the Engineering and Scientific Subroutine Library (ESSL) and Parallel ESSL
  web page.

Other information

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

Technical support

Additional technical support is available from the XL C/C++ Support page at http://www.ibm.com/software/awdtools/xlcpp/linux/support/. This page provides a portal with search capabilities to a large selection of Technotes and other support information.

If you cannot find what you need, you can send email to compinfo@ca.ibm.com.

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

How to send your comments

Your feedback is important in helping to provide accurate and high-quality information. If you have any comments about this information or any other XL C/C++ information, send your comments by email to compinfo@ca.ibm.com.

Be sure to include the name of the information, the part number of the information, the version of XL C/C++, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Chapter 1. Before installing IBM XL C/C++ for Linux, V12.1

Before you install IBM XL C/C++ for Linux, V12.1, perform the following tasks:

- Consult the product README file for any last-minute updates.
- Familiarize yourself with the installation image, which contains the installable compiler packages.
- Ensure that system prerequisites are met and that all required software packages are installed.
- Become either the root user or a user with administrator privileges.

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

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

- Use `xlc_install` to install the full product version of the XL C/C++ compiler as described in Chapter 2, "Basic installation," on page 9. It uninstalls the evaluation copy and then installs the full product version.
- Install only the `vac.lic` RPM packages for the full product version of the XL C/C++ compiler and run `vac_configure` to re-configure the compiler.

The installation image and packages

The installation image for IBM XL C/C++ for Linux, V12.1 is available on an installation CD or for download from an IBM website to a local drive.

The image includes:

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

Installation packages

Table 5 lists the packages that are supplied with the installation image, and the default locations to which they are installed during a basic installation. You can use the `rpm` utility to review the packages. For example, to view package information and its file list, enter the following `rpm` query command:

```
rpm -qpil package_name
```

For the rules on installing packages to custom, nondefault locations, see "Installing packages to multiple, non-default locations" on page 17.

Table 5. IBM XL C/C++ for Linux, V12.1 packages and default installation locations

Package name	Package description	Default installation location
xlsmp.msg.rte	IBM SMP messages package	/opt/ibmcmp/msg/

Table 5. IBM XL C/C++ for Linux, V12.1 packages and default installation locations (continued)

Package name	Package description	Default installation location
xlsmp.rte	IBM SMP runtime package	/opt/ibmcmp/lib/ /opt/ibmcmp/lib64/
xlsmp.lib	IBM SMP static library package	/opt/ibmcmp/xlsmp/3.1/
xlmass.lib	IBM Mathematical Acceleration Subsystem (MASS) package (Not redistributable)	/opt/ibmcmp/xlmass/7.1/
vacpp.rte	IBM XL C/C++ runtime package	/opt/ibmcmp/lib/ /opt/ibmcmp/lib64/
vacpp.rte.lnk	IBM XL C/C++ runtime links package (Not redistributable)	/opt/ibmcmp/vacpp/12.1/
vac.lic	IBM XL C/C++ license package (Not redistributable)	/opt/ibmcmp/vac/12.1/
vac.lib	IBM XL C libraries package	/opt/ibmcmp/vac/12.1/lib /opt/ibmcmp/vac/12.1/lib64
vac.cmp	IBM XL C compiler package (Not redistributable)	/opt/ibmcmp/vac/12.1/
vacpp.lib	IBM XL C++ libraries package (Not redistributable)	/opt/ibmcmp/vacpp/12.1/lib /opt/ibmcmp/vacpp/12.1/lib64
vacpp.cmp	IBM XL C++ compiler package (Not redistributable)	/opt/ibmcmp/vacpp/12.1/
vacpp.samples	IBM XL C/C++ ANSI class library samples package (Not redistributable)	/opt/ibmcmp/vacpp/12.1/samples/
vacpp.help.pdf	IBM XL C/C++ Help PDF documentation package (Not redistributable)	/opt/ibmcmp/vacpp/12.1/doc/
vacpp.man	IBM XL C/C++ man pages package (Not redistributable)	/opt/ibmcmp/vacpp/12.1/man/

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

Table 6. Utilization reporting tool package 1.1

Package name	Package description	Default installation location
urt	Utilization reporting tool package	/opt/ibmurt /opt/ibmurt/1.1/ /opt/ibmurt/1.1/bin/ /opt/ibmurt/1.1/config/ /opt/ibmurt/1.1/msg

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

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

National language support

IBM XL C/C++ for Linux, V12.1 messages support the following language locales:
- en_US
- en_US.utf8
- ja_JP
- ja_JP.eucjp
- ja_JP.utf8
- zh_CN
- zh_CN.gb18030
- zh_CN.gb2312
- zh_CN.gbk
- zh_CN.utf8

English is the default national language and en_US is the default locale. Following installation, you can set the NLSPATH so that messages are displayed in a different language. See "Enabling the error messages" on page 31.

System prerequisites

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

The following are the requirements for installing IBM XL C/C++ for Linux, V12.1:
- **Operating system:** You can use either of the following operating systems supported by the IBM Power Systems™ servers:
  - SUSE Linux Enterprise Server 10 Service Pack 4 (SLES 10 SP4)
  - SUSE Linux Enterprise Server 11 Service Pack 2 (SLES 11 SP2)
  - Red Hat Enterprise Linux 5.7 (RHEL 5.7)
  - Red Hat Enterprise Linux 6.2 (RHEL 6.2)
- **Hardware:**

You can use any of the IBM Power Systems servers as long as it is supported by your operating system distribution. For a complete list of the new IBM Power Systems servers, see http://www.ibm.com/systems/power/hardware/.

- **Storage:**
  - Approximately 200 MB for product packages
  - Minimum of 2 GB hard drive space for paging
  - Minimum of 512 MB for temporary files

  Note: High levels of optimization can require more space for paging and temporary files.

  To verify that you have enough hard disk space available, see "Verifying the amount of hard disk space available" on page 6.

- **Required software:**

  To determine which GNU and Perl packages are required to run the compiler on your operating system, consult the relevant table below. To verify that the required packages are installed, see "Verifying that the required GNU and Perl packages are installed" on page 6

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

Table 7. Required GNU and Perl packages for the RHEL 5.7 operating system

Package name	Version requirements
gcc	4.1.2
gcc-c++	4.1.2
glibc*	2.5
glibc-devel*	2.5
libgcc*	4.1.2
libstdc++*	4.1.2
libstdc++-devel*	4.1.2
compat-libstdc++-33	3.2.3
perl	5.0 or greater **Note:** Perl V5.8 is shipped and automatically installed with the RHEL 5.7 operating system.

Table 8. Required GNU and Perl packages for the RHEL 6.2 operating system

Package name	Version requirements
gcc	4.4.6
gcc-c++	4.4.6
glibc*	2.12
glibc-devel*	2.12
libgcc*	4.4.6
libstdc++*	4.4.6
libstdc++-devel*	4.4.6
compat-libstdc++-33	3.2.3

Table 8. Required GNU and Perl packages for the RHEL 6.2 operating system (continued)

Package name	Version requirements
perl	5.0 or greater **Note:** Perl V5.8 is shipped and automatically installed with the RHEL 6.2 operating system.

Note: The 32-bit libraries are not installed by default on RHEL 6.2, they must be installed from the RHEL 6.2 operating system image.

Table 9. Required GNU and Perl packages for the SLES 10 SP4 operating system

Package name	Version requirements
gcc	4.1.2
gcc-c++	4.1.2
glibc	2.4
glibc-64bit	2.4
glibc-devel	2.4
glibc-devel-64bit	2.4
libgcc	4.1.2
libgcc-64bit	4.1.2
libstdc++	4.1.2
libstdc++-devel	4.1.2
libstdc++-64bit	4.1.2
libstdc++-devel-64bit	4.1.2
perl	5.0 or greater **Note:** Perl V5.8 is shipped and automatically installed with the SLES 10 SP4 operating system.

Table 10. Required GNU and Perl packages for the SLES 11 SP2 operating system

Package name	Version requirements
gcc43-32bit	4.3.4
gcc43	4.3.4
gcc43-c++	4.3.4
glibc-32bit	2.11.3
glibc	2.11.3
glibc-devel-32bit	2.11.3
glibc-devel	2.11.3
libgcc46-32bit	4.6.1
libgcc46	4.6.1
libstdc++46-32bit	4.6.1
libstdc++46	4.6.1
libstdc++43-devel-32bit	4.3.4
libstdc++43-devel	4.3.4

Table 10. Required GNU and Perl packages for the SLES 11 SP2 operating system (continued)

perl	5.0 or greater **Note:** Perl V5.10 is shipped and automatically installed with the SLES 11 SP2 operating system.

To view the documentation that is packaged with IBM XL C/C++ for Linux, V12.1, you will need:

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

Verifying the amount of hard disk space available

IBM XL C/C++ for Linux, V12.1 requires about 200 MB of hard disk storage space. This amount of space accommodates the optional samples and documentation that are shipped with the product.

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

```
df -h /opt
```

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

```
df -h installation_path
```

where *installation_path* represents the non-default location.

Verifying that the required GNU and Perl packages are installed

Before you install IBM XL C/C++ for Linux, V12.1, you should verify that the required GNU and Perl packages were installed with the operating system.

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

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

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

```
rpm -q package_name
```

If the package has been installed, this command should return the name and version number of the package. The version number must be the same as or higher than the version requirement for that package.

Note: If you are installing the compiler onto a system running RHEL 5.7, follow the action described in "Could not determine location of 32-bit or 64-bit GCC (RHEL 5.7)" on page 39 to verify that the `glibc`, `glibc-devel`, `libgcc`, `libstdc++` and `libstdc++-devel` packages are installed.

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

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

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

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

```
gcc-c++-4.1.2-42.el5
```

Previewing the license agreements

Before installing IBM XL C/C++ for Linux, V12.1, you can preview the license agreements.

The terms and conditions for using XL C/C++ are specified in the following PDF files, which are located in the root directory of the distribution package:
* LicenseAgreement.pdf
* license.pdf

After basic installation, the license files can be found in the /opt/ibmcmp/vacpp/ 12.1/ directory with the same names as above.

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

Chapter 2. Basic installation

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

- Install IBM XL C/C++ for Linux, V12.1 on a system where no IBM XL compiler is currently installed.
- Replace an older version of XL C/C++ with IBM XL C/C++ for Linux, V12.1.
- Install IBM XL C/C++ for Linux, V12.1 on a system where IBM XL Fortran for Linux, V14.1 has already been installed.

  In this case, both compilers will use the IBM MASS library provided with IBM XL C/C++ for Linux, V12.1.

  For information about installing IBM XL Fortran for Linux, V14.1, refer to the *IBM XL Fortran for Linux, V14.1 Installation Guide*.

- Install an update on a system where IBM XL C/C++ for Linux, V12.1 has already been installed.

By running the `xlc_install` utility, you are prompted to accept or decline the license agreement. If you accept the agreement, the license files will be output to .txt files for your future reference. If you decline the agreement, the installation process exits without installing the compiler and no files will be written to your system.

You should consider a basic installation of IBM XL C/C++ for Linux, V12.1 with `xlc_install` utility on a system as long as *both* of the following conditions apply:

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

If either of these conditions does not apply, use the procedures in Chapter 3, "Advanced installation," on page 13.

Tasks for basic installation

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

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

Table 11. 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 3
Uninstall any previously installed packages of the same product type.	"Running the xlc_install utility for a new installation" on page 10

Table 11. Steps for basic installation (continued)

Task	For more information, see . . .
Use the `xlc_install` tool to accept or decline the license.	"Running the xlc_install utility for a new installation"
Use the `xlc_install` tool to install and configure the compiler, using the default paths.	"Running the xlc_install utility for a new installation"
Confirm that the compiler packages were successfully installed, and test the installation.	• "Querying for installed packages" on page 35 • "Testing the installation" on page 29
Enable the compiler man pages.	"Enabling the man pages" on page 30
If your system locale and/or encoding are *not* en_US, enable the compile-time error messages for your locale. Otherwise, you can skip this step.	"Enabling the error messages" on page 31
(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 32

Running the xlc_install utility for a new installation

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

About this task

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 performs the following tasks:

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

To run the `xlc_install` utility to install IBM XL C/C++ for Linux, V12.1:

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

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

   For additional arguments that you can specify for `xlc_install`, see "xlc_install options" on page 12.

- If another instance of IBM XL C/C++ 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 SMP and MASS packages are detected on your system either alone or as part of an installation of IBM XL Fortran for Linux, V14.1, you are prompted to uninstall them. Confirm that you want to proceed with uninstalling the existing IBM SMP and MASS packages. If you choose not to uninstall the previously installed components, the installation process will end.

2. You are presented with the license agreement and licensing information. Read the license agreement and licensing information. If you agree to the licensing terms, accept the license agreement 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 32.

   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

   For more information on other specialized invocations that may be available, please refer to *IBM XL C/C++ for Linux, V12.1 Compiler Reference, "Invoking the compiler"*.

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

If all packages are successfully installed:
- A message is displayed confirming the successful installation.

  Note: NLS is not supported for messages that are issued by the xlc_install utility.
- The configuration file is generated. Its location is /opt/ibmcmp/vac/12.1/etc/ vac.cfg.*$OSRelease.gcc$gccVersion*. For example, /opt/ibmcmp/vac/12.1/etc/ vac.cfg.sles11.gcc432 or /opt/ibmcmp/vac/12.1/etc/vac.cfg.rhel6.2.gcc446.

- The installation log is moved to its permanent location: /opt/ibmcmp/vacpp/ 12.1/xlc_install.log.

xlc_install options

The xlc_install utility provides the following options:

-h   Displays the installation utility help page.

-prefix *installation_path*
Explicitly specifies the path to install all the compiler packages.

Note: If this option is not used, the default installation location is /opt/ibmcmp/.

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

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

-U   Updates the compiler to the *Version.Release.Modification.Fix-Build* (V.R.M.F-B) level that the installation utility version supports. For more information, see "Updating a basic installation using the xlc_install utility" on page 22

-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 Linux, V12.1 to the default location and use the procedure provided in Chapter 2, "Basic installation," on page 9. However, you must use alternative procedures if any of the following scenarios apply to you:

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

In all of these scenarios, you can use the `rpm` utility to install the compiler packages; you can also use the `xlc_install` utility to install the compiler packages to a single, nondefault location. After you have successfully installed the compiler packages to a nondefault location using the `rpm` utility, you need to manually configure the compiler environment using the `new_install` or `vac_configure` utility. For configuration procedures, see Chapter 5, "Configuring IBM XL C/C++ for Linux, V12.1 (for advanced users)," on page 25.

Tasks for advanced installation

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

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

Table 12. Steps for installing to the same location as an earlier version

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
If you do not need to maintain multiple versions of the product on your system, remove any existing versions of IBM XL C/C++ for Linux .	Chapter 7, "Uninstalling IBM XL C/C++ for Linux, V12.1," on page 37
Ensure that all system prerequisites are satisfied.	"System prerequisites" on page 3
Install IBM XL C/C++ for Linux, V12.1.	"Installing XL C/C++ to the same location as an earlier version" on page 15
Use the `vac_configure` tool to configure the compiler.	"Running the `vac_configure` utility directly" on page 26
Confirm that the compiler packages were successfully installed, and test the installation.	• "Querying for installed packages" on page 35 • "Testing the installation" on page 29
Enable the compiler man pages.	"Enabling the man pages" on page 30

Table 12. Steps for installing to the same location as an earlier version (continued)

Task	Further information
If your system locale, your encoding, or both, are *not* en_US, enable the compile-time error messages for your locale. Otherwise, you can skip this step.	"Enabling the error messages" on page 31
Optional: Set up the environment to locate the invocation commands without the full path.	"Setting up the environment for the invocation commands" on page 32

If you want to install IBM XL C/C++ for Linux, V12.1 to a nondefault location, separate from any other versions of XL C/C++ on the same system, follow the steps in this table.

Table 13. Steps for installing to a location separate from earlier versions

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
If you do not need to maintain multiple versions of the product on your system, remove any existing versions of IBM XL C/C++ for Linux.	Chapter 7, "Uninstalling IBM XL C/C++ for Linux, V12.1," on page 37
Ensure that all system prerequisites are satisfied.	"System prerequisites" on page 3
Install IBM XL C/C++ for Linux, V12.1.	"Installing XL C/C++ to a location separate from earlier versions" on page 16
Use the new_install or vac_configure tool to configure the compiler.	Chapter 5, "Configuring IBM XL C/C++ for Linux, V12.1 (for advanced users)," on page 25
Confirm that the compiler packages were successfully installed, and test the installation.	• "Querying for installed packages" on page 35 • "Testing the installation" on page 29
Enable the compiler man pages.	"Enabling the man pages" on page 30
If your system locale, your encoding, or both, are *not* en_US, enable the compile-time error messages for your locale. Otherwise, you can skip this step.	"Enabling the error messages" on page 31
Optional: Set up the environment to locate the invocation commands without the full path.	"Setting up the environment for the invocation commands" on page 32

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

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

To install IBM XL C/C++ for Linux, V12.1 to co-reside with earlier versions of XL C/C++ on the same system, you have the following options:

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

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

- You can allow each version to use the runtime environment that was shipped with it. This option is recommended if you want to phase a migration to the later version. In this case, you will need to install IBM XL C/C++ for Linux, V12.1 in a separate location from all other versions of IBM XL C/C++ for Linux. To do so, follow one of the procedures in "Installing XL C/C++ to a location separate from earlier versions" on page 16.

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

The following procedure describes how to install IBM XL C/C++ for Linux, V12.1 and update the runtime packages used by an existing version of IBM XL C/C++ for Linux, V11.1 in the same location.

This procedure assumes the following:
- You are installing IBM XL C/C++ for Linux, V12.1 to a location where IBM XL C/C++ for Linux, V11.1 has already been installed (by default, /opt/ibmcmp/).
- Your current working directory contains all IBM XL C/C++ for Linux, V12.1 packages, and no other RPM packages.
- The existing runtime component versions listed in Table 14 are installed in the same installation location (by default, /opt/ibmcmp/).

Table 14. XL C/C++ SMP and runtime packages

Existing XL C/C++ V11.1 runtime packages	New XL C/C++ V12.1 runtime packages
xlsmp.msg.rte-2.1.0.$F-$B	xlsmp.msg.rte-3.1.0.0-$B
xlsmp.rte-2.1.0.$F-$B	xlsmp.rte-3.1.0.0-$B
xlsmp.lib-2.1.0.$F-$B	xlsmp.lib-3.1.0.0-$B
vacpp.rte-11.1.0.$F-$B	vacpp.rte-12.1.0.0-$B

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

To install IBM XL C/C++ for Linux, V12.1, and update the IBM XL C/C++ for Linux, V11.1 XL SMP and runtime packages:

1. Remove the existing runtime packages by issuing the following commands to avoid future dependency errors:
   ```
   rpm -e vacpp.rte-11.1.0.0-$B  --nodeps
   rpm -e xlsmp.lib-2.1.0.0-$B  --nodeps
   rpm -e xlsmp.rte-2.1.0.0-$B  --nodeps
   rpm -e xlsmp.msg.rte-2.1.0.0-$B  --nodeps
   ```

   Note: This example shows package names with no applied updates.
2. To replace the deleted runtime packages and to install IBM XL C/C++ for Linux, V12.1, issue the following command:
   ```
   rpm -ivh *.rpm
   ```
3. As a precaution, make a backup copy of any existing configuration file.

4. Modify any existing IBM XL C/C++ for Linux, V11.1 configuration file to use the new configuration path:

```
sed -e "s/xlsmp\/2\.1/xlsmp\/3\.1/g"
< /etc/opt/ibmcmp/vac/11.1/vac.cfg >
/etc/opt/ibmcmp/vac/11.1/vac.cfg.new
mv /etc/opt/ibmcmp/vac/11.1/vac.cfg.new
/etc/opt/ibmcmp/vac/11.1/vac.cfg
```

   Note: The IBM XL C/C++ for Linux, V11.1 path does not change.

5. Generate the default IBM XL C/C++ for Linux, V12.1 configuration file:

```
 installation_path/vacpp/12.1/bin/vac_configure -gcc /usr -gcc64 /usr
  -ibmcmp installation_path/ installation_path/vac/12.1/etc/vac.base.cfg
  -o  installation_path/vac/12.1/etc/vac.cfg.$OSRelease.gcc$gccVersion
```

   where *installation_path* is the location in which all IBM XL C/C++ for Linux, V12.1 packages are installed (by default, /opt/ibmcmp/). *$OSRelease* is the level of the Operating System, and *$gccVersion* is the GCC version installed on the system. For example, vac.cfg.sles11.gcc432, or vac.cfg.rhel6.2.gcc446.

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

About this task

This section provides procedures for two possible installation scenarios:

- You can install all compiler packages to a single nondefault location. For example, rather than installing all packages into the default directory /opt/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 nondefault directories, as well as procedures for doing so, are provided in "Installing packages to multiple, non-default locations" on page 17.

Installing all packages to a single, non-default location
About this task

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

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

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

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

   For additional arguments that you can specify for xlc_install, see "xlc_install options" on page 12.

- Use the rpm utility. Ensure that your current working directory contains all of the packages for IBM XL C/C++ for Linux, V12.1 and no other RPM packages. From your current working directory, use the following command:

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

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

Installing packages to multiple, non-default locations
About this task

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

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

Table 15 provides information about which packages must be installed together in
the same directory and which can be installed in any directory.

Table 15. Rules for installing packages to multiple, non-default locations

Package Name	Package Description	Rules for installation to a non-default location
xlsmp.msg.rte	IBM SMP messages package	All XL SMP packages must be installed in the same location. For the remainder of this document, the name *xlsmprt_path* is used to refer to this location.
xlsmp.rte	IBM SMP runtime package	
xlsmp.lib	IBM SMP static library package	
xlmass.lib	IBM Mathematical Acceleration Subsystem (MASS) package	Any location. For the remainder of this document, the name *xlmass_path* is used to refer to this location.
vacpp.rte	IBM XL C/C++ runtime package	All XL C/C++ runtime packages must be installed in the same location. For the remainder of this document, the name *xlrte_path* is used to refer to this location.
vacpp.rte.lnk	IBM XL C/C++ runtime links package	
vac.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.
vac.lib	IBM XL C/C++ compiler libraries package	All XL C/C++ compiler and library packages must be installed in the same location. For the remainder of this document, the name *xlcmp_path* is used to refer to this location.
vac.cmp	IBM XL C/C++ compiler package	
vacpp.omp	IBM XL C/C++ OpenMP package	
vacpp.lib	IBM XL C/C++ compiler libraries package	Must be installed in the same location as the other compiler and library packages above.
vacpp.cmp	IBM XL C/C++ compiler package	
vacpp.help.pdf	IBM XL C/C++ Help pdf documentation package	Any location (optional). For the remainder of this document, the name *doc_path* is used to refer to this location.

Table 15. Rules for installing packages to multiple, non-default locations (continued)

Package Name	Package Description	Rules for installation to a non-default location
vacpp.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.
vacpp.samples	IBM XL C/C++ samples package	Any location (optional). For the remainder of this document, the name *smpls_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 15 on page 17.

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

In order to avoid dependency errors during installation of IBM XL C/C++ for Linux, V12.1 issue the following commands in the order given:

```
rpm -ivh xlsmp.msg.rte-3.1.0.0-$B.ppc64.rpm --prefix $SMPpath
rpm -ivh xlsmp.rte-3.1.0.0-$B.ppc64.rpm --prefix $SMPpath
rpm -ivh xlsmp.lib-3.1.0.0-$B.ppc64.rpm --prefix $SMPpath
rpm -ivh xlmass.lib-7.1.0.0-$B.ppc64.rpm --prefix $MASSpath

rpm -ivh vacpp.rte-12.1.0.0-$B.ppc64.rpm --prefix $RTEpath
rpm -ivh vacpp.rte.lnk-12.1.0.0-$B.ppc64.rpm --prefix $RTEpath
rpm -ivh vac.lic-12.1.0.0-$B.ppc64.rpm --prefix $LICpath
rpm -ivh vac.lib-12.1.0.0-$B.ppc64.rpm --prefix $CMPpath
rpm -ivh vac.cmp-12.1.0.0-$B.ppc64.rpm --prefix $CMPpath
rpm -ivh vacpp.lib-12.1.0.0-$B.ppc64.rpm --prefix $CMPpath
rpm -ivh vacpp.cmp-12.1.0.0-$B.ppc64.rpm --prefix $CMPpath
```

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

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.man-12.1.0.0-$B.ppc64.rpm --prefix $MANPAGpath
rpm -ivh vacpp.samples-12.1.0.0-$B.ppc64.rpm --prefix $SAMPpath
 rpm -ivh vacpp.help.pdf-12.1.0.0-$B.ppc64.rpm --prefix $DOCpath
```

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

Querying for RPM package summaries

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

To query for the summary of an RPM package that is not installed, issue the command:

```
$ rpm --qf="%{summary}\n" -qp rpm_file_name
```

For instance, to query for the summary of the vac.lib-12.1.0.0-*$B*.ppc64.rpm RPM file that has not yet been installed, issue the following command:

```
$ rpm --qf="%{summary}\n" -qp vac.lib-12.1.0.0-100701.ppc64.rpm
```

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

```
IBM XL C/C++ for Linux, V12.1 - C/C++ Compiler Libraries Package (SLES10/RHEL5)
```

Chapter 4. Installing an update

An update, or program temporary fix (PTF), of IBM XL C/C++ for Linux, V12.1 provides a fix or multiple fixes to the product. You can download updates from the support website: http://www.ibm.com/software/awdtools/xlcpp/linux/support/.

Every PTF update package comes in tar.gz (compressed) format and includes a version of the `xlc_install` utility that is customized to install only the update that accompanies it. If you have any version (including an earlier update) of IBM XL C/C++ for Linux, V12.1 installed on your system, you can apply the latest update. If you are applying the update to an installation in the default location, follow the procedure in "Updating a basic installation using the xlc_install utility" on page 22. If you are applying the update to an installation in a non-default location, you must use one of the procedures in "Updating an advanced installation" on page 24. If you want to try out a PTF before installing it, use the procedure in "Testing a PTF before you install it" on page 22.

Tasks for update installation

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

If the compiler has been installed to the default directory, /opt/ibmcmp/, you must follow the steps outlined in the following table.

Table 16. Steps for basic installation: update installation

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
Use the `xlc_install` tool to install the update packages.	"Updating a basic installation using the xlc_install utility" on page 22
Confirm that the compiler packages have been successfully installed, and test the installation.	• "Querying for installed packages" on page 35 • "Testing the installation" on page 29
Optional: If you do not choose to create symbolic links to the compiler invocation commands during the update process, set up the environment to locate the invocation commands without the full path. Otherwise, you can skip this step.	"Setting up the environment for the invocation commands" on page 32

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

Table 17. Steps for advanced installation: update installation

Task	Further information
Become either the root user or a user with administrator privileges.	Documentation supplied with the operating system
Uninstall all XL C/C++ packages except vac.lic.	"Updating an advanced installation" on page 24

Table 17. Steps for advanced installation: update installation (continued)

Task	Further information
Install update packages to the nondefault location.	"Updating an advanced installation" on page 24
Configure the compiler.	Chapter 5, "Configuring IBM XL C/C++ for Linux, V12.1 (for advanced users)," on page 25
Confirm that the compiler packages have been successfully installed, and test the installation.	• "Querying for installed packages" on page 35 • "Testing the installation" on page 29
Optional: Set up the environment to locate the invocation commands without the full path.	"Setting up the environment for the invocation commands" on page 32

Testing a PTF before you install it

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

About this task

To install a PTF to a nondefault location, use the procedure described in "Installing XL C/C++ to a location separate from earlier versions" on page 16, using the **rpm** utility.

To configure the compiler, use the procedure described in "Running the `vac_configure` utility directly" on page 26.

What to do next

After you have finished testing the new PTF and have verified that you want to replace the older version with the new update, install the PTF to the default location.

For instructions on how to use the `xlc_install` utility to install the new update, see "Updating a basic installation using the xlc_install utility."

If the existing version on the system was installed using an advanced installation, see "Updating an advanced installation" on page 24.

Updating a basic installation using the xlc_install utility

The `xlc_install` utility is the recommended tool for updating a basic installation of IBM XL C/C++ for Linux, V12.1. It is located in the root directory of the installation image.

About this task

You can use the `xlc_install` utility to update IBM XL C/C++ for Linux, V12.1 if all the following conditions have been met:
• The base version of IBM XL C/C++ for Linux, V12.1 is already successfully installed in the /opt/ibmcmp/ directory.

- The update package has been uncompressed and unpacked on the system.

When you run the `xlc_install` utility to apply an update, it performs the following tasks:
- Checks for all prerequisite software packages
- Uninstalls IBM XL C/C++ for Linux, V12.1 packages
- Installs updated compiler packages into the default location
- Automatically invokes the `new_install` utility, which installs the license file, renames the old configuration file, and generates a new configuration file
- Optionally creates symbolic links in `/usr/bin/` to the compiler invocation commands
- Generates an installation log in the `/tmp/` directory

To run the `xlc_install` utility to apply an update for IBM XL C/C++ for Linux, V12.1:

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

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

   where *mmmYYYY* is the month and year of the update shipment date. (For example, `dec2012` indicates a shipment date in December of 2012.)

2. Install the update by issuing the following command:

   ```
   ./xlc_install -U
   ```

   For additional arguments that you can specify for `xlc_install`, see "xlc_install options" on page 12.

3. Confirm that you want to uninstall any previously installed IBM XL C/C++ for Linux, V12.1 packages.

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

   Note: `xlc_install` uninstalls these packages from their existing location and reinstalls them to the default location, /opt/ibmcmp/. Therefore, if they were previously installed in a nondefault location as part of an IBM XL Fortran installation, you must run `xlf_configure` to reconfigure the IBM XL Fortran compiler to point to the default location for these packages. For procedures, see "Running the `xlf_configure` utility directly" in the *IBM XL Fortran for Linux, V14.1 Installation Guide*.

5. Accept the license agreement and licensing information.

6. Optionally, create the symbolic links for the compiler invocations in the /usr/bin/ directory.

   Note: An alternative to this step is to add the path that contains the compiler invocations to the PATH environment variable. See "Setting the *PATH* environment variable to include the path to the compiler invocations" on page 32.

   If you choose 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

7. If you customized the previously generated configuration file, manually edit *installation_path*/vac/12.1/etc/vac.cfg to replicate those changes in the newly generated configuration file.

Updating an advanced installation

To update an advanced installation of IBM XL C/C++ for Linux, V12.1, it is recommended that you first uninstall all IBM XL C/C++ for Linux, V12.1 packages except the license package, vac.lic.

If you attempt to install a new PTF to a location where an older PTF is currently installed, the installation of the new PTF will fail due to a location conflict: two files that belong to two different PTF levels cannot be installed to the same location at the same time. See Chapter 7, "Uninstalling IBM XL C/C++ for Linux, V12.1," on page 37 for instructions on how to uninstall XL C/C++.

After the old packages have been uninstalled, follow the instructions in Chapter 3, "Advanced installation," on page 13 to install the PTF packages to the location of your choice.

Chapter 5. Configuring IBM XL C/C++ for Linux, V12.1 (for advanced users)

Before you can run IBM XL C/C++ for Linux, V12.1, you must configure (or re-configure) the compiler if any of the following conditions apply to you:

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

The compiler provides two configuration tools: `new_install` and `vac_configure`, both located in the *installation_path*/vacpp/12.1/bin/ directory after installation.

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

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

- All IBM XL C/C++ for Linux, V12.1 compiler packages are installed in the same location, *installation_path*.
- Only one version of GCC is installed in your system and it can be found in the PATH environment variable.
- You have root or administrator privileges.
- You want to generate the configuration file in the *installation_path*/vac/12.1/etc/ directory.

For instructions, see "Running the `new_install` utility."

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

- You have multiple versions of IBM XL C/C++ for Linux installed on your system.
- You receive an error message from the `new_install` command. (See Chapter 8, "Troubleshooting the installation and configuration," on page 39.)
- You want the generated configuration file to be placed in a location that is different from *installation_path*/vac/12.1/etc/.
- You have multiple versions of GCC installed on your system and you need to specify which GCC version you would like to reference in the configuration file.

Note: If you configure the compiler using `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 26.

Running the `new_install` utility

Use the `new_install` utility that is provided with an instance of IBM XL C/C++ for Linux, V12.1 compiler to configure an advanced installation if all compiler packages corresponding to this instance have been installed in the same location, and only one version of GCC is installed on your system.

About this task

The `new_install` utility performs the following tasks:

- Backs up any existing configuration file that is located in *installation_path*/`vac/12.1/etc/vac.cfg`.
- Queries the RPM database for the paths to the compiler packages and the paths to the 32-bit GCC (*gcc32path*) and the 64-bit GCC (*gcc64path*) in the PATH environment variable, and then uses the obtained values to run the `vac_configure` utility.
- Installs the license file.
- Generates the configuration file in the default location *installation_path*/`vac/12.1/etc/vac.cfg`.

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

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

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

       cd *installation_path*/vacpp/12.1/bin/

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

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 `vac_configure` utility directly

Use the `vac_configure` utility to configure an advanced installation if multiple versions of XL C/C++ or multiple versions of the SDK GCC are installed on your system.

About this task

Take the following steps to run the `vac_configure` utility:

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

       cd *installation_path*/vacpp/12.1/bin/

   where *installation_path* is the installation location of the compiler packages. If the compiler is installed in the default location, *installation_path* is /opt/ibmcmp/. If the compiler has been installed to multiple, non-default locations, the `new_install` utility can be found in *$CMPpath*/vacpp/12.1/bin/.

2. Run the following command:

       ./vac_configure *options*

For details about required arguments to the `vac_configure` command, see "vac_configure options" on page 27.

vac_configure **options**

The vac_configure command has the following syntax:
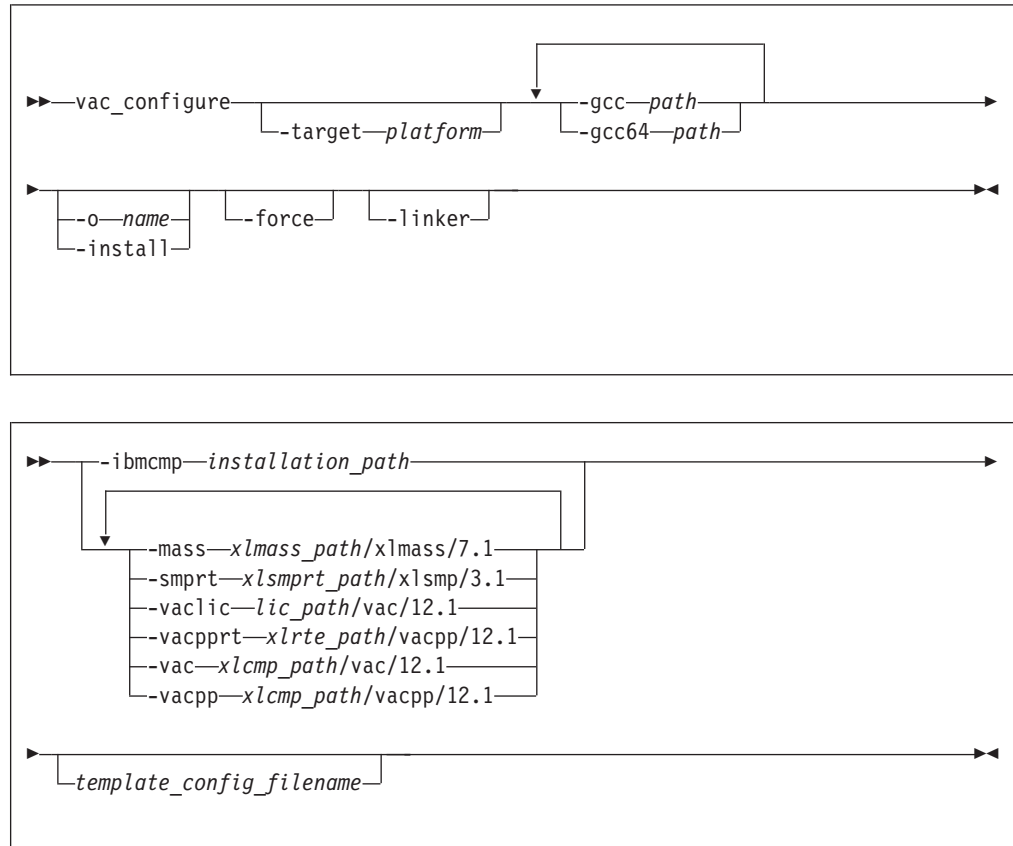
where:

-h   Displays the help page for the vac_configure options.

-target *platform*

Specifies the operating system platform. Valid names are:

- sles
- rhel

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

-gcc *path*

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

    -gcc /usr

-gcc64 *path*

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

    -gcc64 /usr

-o *file_name*

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

-install

Generates the default configuration file as *installation_path*/vac/12.1/ etc/vac.cfg.*$OSRelease.gcc$gccVersion*, where *$OSRelease* is the level of the Operating System, and *$gccVersion* is the GCC version installed on the system. For example, `vac.cfg.sles11.gcc432`, or `vac.cfg.rhel6.2.gcc446`. By default, output is written to the display only.

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

-force   Forces the `vac_configure` utility to overwrite any existing output file with the same name and path as that are 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 IBM XL C/C++ 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/7.1/**

Specifies the path where the xlmass package is installed. By default, the full path is /opt/ibmcmp/xlmass/7.1/.

-smprt *xlsmprt_path***/xlsmp/3.1/**

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/3.1/.

-vaclic *lic_path***/vac/12.1/**

Specifies the path where the vac.lic package is installed. By default, the full path is /opt/ibmcmp/vac/12.1/.

-vacpprt *xlrte_path***/vacpp/12.1/**

Specifies the path where the vacpp.rte and vacpp.rte.lnk packages are installed. By default, the full path is /opt/ibmcmp/vacpp/12.1/.

-vac *xlcmp_path***/vac/12.1/**

Specifies the path where the vac.cmp, vacpp.omp.rte and vac.lib packages are installed. By default, the full path is /opt/ibmcmp/vac/12.1/.

template_config_file_name

The input file that is used to construct the configuration file. By default, this is /opt/ibmcmp/vac/12.1/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/12.1/etc/vac.base.cfg

Chapter 6. After installing IBM XL C/C++ for Linux, V12.1

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

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

Testing the installation

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

About this task

Take the following steps to build and run a "Hello World" application:

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

   ```
   #include <stdio.h>

   int main(void)
   {
      printf("Hello World!\n");
      return 0;
   }
   ```

2. Compile the program:

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

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

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

   ```
      installation_path/vacpp/12.1/bin/xlc hello.c -o hello
   ```

   where *installation_path* is the installation location of the compiler packages. If the compiler has been installed to the default location, *installation_path* is `/opt/ibmcmp/`.

3. Run the program by entering the following command:

   ```
      ./hello
   ```

   The result is "Hello World!".

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

   ```
      echo $?
   ```

   The result is "0".

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

```
#include <iostream>
using namespace std;

int main()
{
   cout << "Hello World!\n";
   return 0;
}
```

6. Compile the program:

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

```
 xlc++ hello.cpp -o hello
```

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

```
 installation_path/vacpp/12.1/bin/xlc++ hello.cpp -o hello
```

where *installation_path* is the installation location of the compiler packages. If the compiler has been installed to the default location, *installation_path* is `/opt/ibmcmp/`.

7. Run the program :

```
  ./hello
```

The result should be "Hello World!".

8. Check the exit code of the program:

```
  echo $?
```

The result should be "0".

Enabling utilization tracking

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

Utilization tracking is disabled by default. You can enable it by modifying the entry **-qnoenabletracking** to **-qenabletracking** in the utilization tracking configuration file, urtxlc_cpp1201linux.cfg. Various other aspects of utilization tracking can also be specified in this file. In each installation of the compiler, there is a symlink named `urt_client.cfg` that points to the utilization tracking configuration file. You might need to change the symlink if you use a copy of the utilization tracking configuration file in a different location or if you use multiple copies of the file. For more information about utilization tracking, see Configuring utilization tracking in the *XL C/C++ Compiler Reference*.

Enabling the man pages

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

The IBM XL C/C++ for Linux, V12.1 man pages support the following 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

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

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

  ```
  export MANPATH=installation_path/vacpp/12.1/man/LANG:$MANPATH
  ```

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

  ```
  setenv MANPATH installation_path/vacpp/12.1/man/LANG:$MANPATH
  ```

where:

- *installation_path* is the location where you have installed the XL C/C++ packages. By default, this is /opt/ibmcmp/.
- *LANG* is any of the language locales as shown in the preceding list.

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

To test whether the manual pages have been successfully installed, open one or more of the manual pages as described in "Viewing the manual pages" on page 35.

Enabling the error messages

If your system uses the en_US locale and encoding, the compiler message catalogs are automatically configured to display correctly, regardless of whether you used the basic or advanced method of installation and configuration. However, if your system uses any other supported locale (for a list of supported language locales, see "National language support" on page 3), you must set the *NLSPATH* environment variable, so that the compiler and runtime functions can find the appropriate message catalogs after the installation.

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

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

  ```
  export NLSPATH=$NLSPATH:
    xlsmprt_path/msg/%L/%N:
    xlrte_path/msg/%L/%N:
    xlcmp_path/vacpp/12.1/msg/%L/%N
  ```

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

  ```
  setenv NLSPATH $NLSPATH:
    xlsmprt_path/msg/%L/%N:
    xlrte_path/msg/%L/%N:
    xlcmp_path/vacpp/12.1/msg/%L/%N
  ```

where:

- *xlsmprt_path* is the installation location of the SMP packages. By default, this is /opt/ibmcmp/.
- *xlrte_path* is the installation location of the IBM XL C/C++ for Linux, V12.1 runtime packages. By default, this is /opt/ibmcmp/.
- *xlcmp_path* is the installation location of the IBM XL C/C++ for Linux, V12.1 compiler packages. By default, this is /opt/ibmcmp/.

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

Setting up the environment for the invocation commands

If you used the xlc_install utility to install the compiler and you selected to create the symbolic links at that time, you have already set up the environment for the invocation commands. Do not perform the procedures in this section.

If you did not select to create the symbolic links when you installed the compiler and want to invoke the compiler without having to specify the full path, you must perform one of the following tasks:
- Set the *PATH* environment variable, as shown in "Setting the *PATH* environment variable to include the path to the compiler invocations."
- Create symbolic links to the compiler invocation commands, as shown in "Creating symbolic links to the compiler invocations" on page 33.

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

To use IBM XL C/C++ for Linux, V12.1 commands without typing the complete path, you can add the location of the compiler invocations to the *PATH* environment variable.

The command to set the *PATH* environment variable depends on the shell that you are using.
- If you are using the Bourne, Korn, or BASH shell, use the following command:

  export PATH=$PATH:*installation_path*/vacpp/12.1/bin/
- If you are using the C shell, use the following command:

  setenv PATH $PATH:*installation_path*/vacpp/12.1/bin/

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

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

Creating symbolic links to the compiler invocations

To use the compiler without typing the complete path, you can create symbolic links in the `/usr/bin/` directory for the specific invocations that are contained in the *installation_path*/vacpp/12.1/bin/ directory.

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

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

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

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

Enter the following command to create a symbolic link:

`ln –s` *installation_path*/vacpp/12.1/bin/*invocation* /usr/bin/*invocation*

where:

- *installation_path* is the location where you have installed the compiler packages. By default, this is /opt/ibmcmp/.
- *invocation* is one of the compiler invocations (such as `xlc`) in *installation_path*/vacpp/12.1/bin/.

Basic example: Creating a symbolic link to a compiler invocation

This example assumes that all of IBM XL C/C++ for Linux, V12.1 is installed in the default location /opt/ibmcmp/.

`ln –s /opt/ibmcmp/vacpp/12.1/bin/xlc /usr/bin/xlc`

Enabling IBM License Metric Tool (ILMT) and Tivoli Asset Discovery for Distributed (TADd)

IBM License Metric Tool (ILMT) and Tivoli® Asset Discovery for Distributed (TADd) can help you manage software usage metering and license allocation services on supported systems. In general, ILMT and TADd recognize and monitor the products that are installed and in use on your system.

Note: ILMT and TADd are not part of the IBM XL C/C++ for Linux offering, and must be ordered and installed separately.

Once installed and activated, ILMT and TADd scan your system for product inventory signatures that indicate whether a given product is installed on your

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

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

If IBM XL C/C++ for Linux is installed in the default location, the signature files are in the /opt/ibmcmp/vac/12.1/properties/version/ directory. Otherwise, if IBM XL C/C++ for Linux is installed in a non-default location, the signature files are in the *$installation_path*/vac/12.1/properties/version/ directory, where *$installation_path* is the target directory for installation specified by --prefix option of the non-default installation command.

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

Accessing the local documentation

Help for IBM XL C/C++ for Linux, V12.1 is available in both HTML and PDF formats. Manual pages for the compiler invocation commands and other command utilities are also included.

Viewing the HTML documentation

The product documentation is available in a fully searchable HTML-based information center.

The information center is viewable on the web at http://publib.boulder.ibm.com/infocenter/lnxpcomp/v121v141/index.jsp. You can also download a version of this information center containing the product documentation for IBM XL C/C++ for Linux, V12.1 and IBM XL Fortran for Linux, V14.1. This information center can be installed on any supported operating systems of IBM XL C/C++ for Linux, V12.1. It can be accessed with a web browser from the machine on which it is installed or from other computers on the same network.

Notes:
- You must have root access to launch and shut down the information center.
- The compressed information center package file is approximately 160 MB, and 220 MB when it is uncompressed.

Follow these steps to download the help system:
1. Go to the XL C/C++ compiler library page at http://www.ibm.com/software/awdtools/xlcpp/linux/library/. Download the information center package file XLHelp-v121-v141-LINUX-64bit.tar.Z from the download link on this page.
2. Extract the package on your machine. Run the following command to extract the files to a new directory at ./xlhelp:

   ```
   zcat XLHelp-v121-v141-LINUX64.tar.Z | tar -xf -
   ```
3. Enter the new directory.

   ```
   cd ./xlhelp
   ```
4. You can review the licence information in the following two files:
   - license.pdf
   - LicenseAgreement.pdf

5. Start the information center by running the following command:

   `./xlhelp &`

6. You can access the information center either locally or remotely. You need to have a web browser installed on your machine to view the information center.

   - To access the information center locally, use this URL:

     `http://localhost:1214/help/index.jsp`

   - To access the information center remotely, use this URL:

     `http://machine_name:1214/help/index.jsp`

     where *machine_name* is the name of the computer on which the Eclipse server has been launched.

7. To stop the information center, run this command:

   `./xlhelp_end`

8. To remove the information center, you must first stop the information center, and remove the extracted directory `.xlhelp` using the following command:

   `rm -r ./xlhelp`

Viewing the PDF documentation

PDF versions of the IBM XL C/C++ for Linux, V12.1 product manuals are available in the `/doc/$LANG/pdf/` directory of the installation media (either product CD or electronic package).

After default installations, the PDF documentation can be found in the `/opt/ibmcmp/vacpp/12.1/doc/$LANG/pdf/` directory. For nondefault installations, the PDF documentation is located in the `$installation_path/vacpp/12.1/doc/$LANG/pdf/` directory. *$LANG* can be en_US, ja_JP, or zh_CN.

Viewing the manual pages

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

For instructions about how to enable the man pages for viewing, see "Enabling the man pages" on page 30.

To invoke a manual page, enter the following command:

`man command`

Example:

`man xlc`

Querying for installed packages

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

About this task

To query for an individual package, enter the following command:

`rpm -q vac.cmp`

The result is:

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

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

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

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

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

The result is a list containing all of the packages as shown in Table 5 on page 1. If none of the packages was properly installed, there will be no output from the command.

Chapter 7. Uninstalling IBM XL C/C++ for Linux, V12.1

IBM XL C/C++ for Linux, V12.1 does not provide a stand-alone uninstallation tool. You must use the Linux `rpm` utility to uninstall IBM XL C/C++ for Linux, V12.1.

Note:

- You must have root user access to uninstall the compiler.
- Whenever you uninstall a package, specify the *V.R.M.F-B* (Version.Release.Modification.Fix-Build level) of the package. For information about how to determine the package's V.R.M.F-B, see "Querying for installed packages" on page 35.
- Always uninstall packages in the reverse order in which they have been installed, that is, 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.
- You cannot uninstall packages that are required by other packages. For example, `xlsmp.rte` is a shared component if IBM XL Fortran for Linux, V14.1 is also installed on the same system.
- The uninstallation commands do not remove any configuration files that have been generated by the `new_install` or `vac_configure` utility.

Example: Uninstalling IBM XL C/C++ for Linux, V12.1

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

In this example:
- The compiler packages have a *V.R.M.F-B* of 12.1.0.0-*$B*.
- The IBM MASS library package has a *V.R.M.F-B* of 7.1.0.0-*$B*.
- The IBM SMP library packages have a *V.R.M.F-B* of 3.1.0.0-*$B*.

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

To uninstall IBM XL C/C++ for Linux, V12.1, issue the following commands in the order given below.

The following packages are C or C++ specific, and must be uninstalled to uninstall XL C/C++:

```
rpm -e vacpp.cmp-12.1.0.0-$B
rpm -e vacpp.lib-12.1.0.0-$B
rpm -e vac.cmp-12.1.0.0-$B
rpm -e vac.lib-12.1.0.0-$B
rpm -e vac.lic-12.1.0.0-$B
rpm -e vacpp.rte.lnk-12.1.0.0-$B
rpm -e vacpp.rte-12.1.0.0-$B
```

Note: *$B* is the build number of the package that is currently installed on your system. If you want to uninstall XL Fortran at the same time as XL C/C++, uninstall all XL Fortran specific packages before proceeding. See "*Example:*

Uninstalling IBM XL Fortran for Linux, V14.1" in the XL Fortran Install Guide for a complete list of XL Fortran specific packages and the order in which they must be uninstalled.

If you have XL Fortran installed on your system, and want it to continue to function normally, do *not* uninstall the following packages; otherwise, uninstall them in the order given below:

```
rpm -e xlmass.lib-7.1.0.0-$B
rpm -e xlsmp.lib-3.1.0.0-$B
rpm -e xlsmp.rte-3.1.0.0-$B
rpm -e xlsmp.msg.rte-3.1.0.0-$B
```

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

```
rpm -e vacpp.samples-12.1.0.0-$B
rpm -e vacpp.man-12.1.0.0-$B
rpm -e vacpp.help.pdf-12.1.0.0-$B
```

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

Chapter 8. Troubleshooting the installation and configuration

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

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

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

Use the information in this section to help you respond to any problems you may encounter when you install and configure IBM XL C/C++ for Linux, V12.1.

The specified directory *rpmlocation_path* does not exist.

Scenario

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

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

Action

Ensure that you have specified the location of the existing compiler packages correctly. You might need to use the **-rpmloc** *rpmlocation_path* option if you moved the `xlc_install` utility to a different location than the one provided in the installation image. For more information, see "xlc_install options" on page 12.

rpmlocation_path does not contain . . .

Scenario

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

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

Action

Ensure you have all of the packages listed in Table 5 on page 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 12.

Could not determine location of 32-bit or 64-bit GCC (RHEL 5.7)

Scenario

You are running either the `new_install` or the `vac_configure` utility to configure the compiler on a computer running RHEL 5.7 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

At least one of the following packages is not installed in the appropriate directory:
- glibc
- glibc-devel
- libgcc
- libstdc++
- libstdc++-devel

Action

Verify that the 32-bit and 64-bit `glibc`, `glibc-devel`, `libgcc`, `libstdc++` and `libstdc++-devel` packages are installed on the system by issuing the following command:

```
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE}-%{ARCH}\n' packagename
```

where *packagename* is one of (glibc, glibc-devel, libgcc, libstdc++, libstdc++-devel)

For example, to check whether the required versions of the 32-bit and 64-bit glibc packages are installed on a RHEL 5.7 system run:

```
rpm -q --qf '%{NAME}-%{VERSION}-%{RELEASE}-%{ARCH}\n' glibc
```

If both the 32–bit and 64–bit glibc packages are installed, and the packages' version numbers are both 2.5, the expected output is:

```
glibc-2.5-12-ppc
glibc-2.5-12-ppc64
```

Note: The 64-bit `glibc`, `glibc-devel`, `libgcc`, `libstdc++` 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`.

Run `new_install` or vac_configure again.

The names of RHEL 5.7 packages are structured so that they have the same name regardless whether they are 32-bit or 64-bit packages. As a result, the `rpm` command's default output will not indicate whether the packages installed on your system are 32-bit, 64-bit or both.

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 websites are provided for
convenience only and do not in any manner serve as an endorsement of those
websites. The materials at those websites are not part of the materials for this IBM
product and use of those websites is at your own risk.

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 1998, 2012.

Trademarks and service marks

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

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

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

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

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

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

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

Index

A
advanced installation
   example   18
   overview   13
   procedures   14
advanced user, described   v
available space, determining   6

B
basic example, described   ix
basic installation
   overview   9
   procedure   10
basic user, described   v

C
co-residency
   compilers   14
configuration files
   backup   25
   customizing   25, 26
   default   28
   editing   26
   generation   11, 28
   modifying   25, 26
   multiple   25
   overwriting   28
   renaming   28
   security   37

D
debugging   39
default installation   9
documentation
   man pages   35

E
environment variable   32
error messages   31

G
GNU packages
   required   3
   verifying   6

H
hard disk space
   required   3
   verifying   6
hardware requirements   3
help   34
   displaying   12

help *(continued)*
   html   34
   man pages   30
   pdf   35

I
IBM License Metric Tool   33
ILMT   33
installation
   multiple versions   13, 14
   options   12
   package location   16, 17
   specialized   17
   testing   29
installation CD   1
installation image   1
installation log   39
installation utility
   options   12
   using   21
invocation commands   32

L
languages   3
locales   3

M
man pages   30
   viewing   35

N
national language support   3
nondefault installation   13

O
operating systems, supported   3

P
packages
   installation   1
   not found   39
   querying   35
PATH environment variable   32
Perl packages
   required   3
   verifying   6
preinstallation   1
PTF   21

Q
querying
   installed packages   35
   RPM file summaries   19

R
Red Hat Package Manager (RPM)   1

S
software requirements   3
symbolic links   10
   creating   33

T
TADd   33
temporary files
   installation log   39
testing
   installation   29
   PTF   22
Tivoli Asset Discovery for
  Distributed   33

U
uninstallation
   example   37
   overview   37
updates
   applying   22
   installation utility option   12
   overview   21
   prerequisites   23
   steps   23
   to a nondefault location   22
   trying out   16, 22

V
verifying
   GNU and Perl packages   6
   hard disk space   6

IBM ®

Product Number: 5765-J03; 5725-C73

Printed in USA