

IBM XL C/C++ Advanced Edition V7.0 for Linux



# Installation Guide



IBM XL C/C++ Advanced Edition V7.0 for Linux



# Installation Guide

**Note!**

Before using this information and the product it supports, read the information in “Notices” on page 25.

**Second Edition (March 2005)**

This edition applies to IBM XL C/C++ Advanced Edition V7.0.1 for Linux (5724-K77) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

© **Copyright International Business Machines Corporation 2003, 2005. All rights reserved.**

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

---

# Contents

<b>Chapter 1. Installing the product</b>	1
XL C/C++ V7.0 packaging	1
Installing IBM XL C/C++ Advanced Edition V7.0 for Linux.	2
System prerequisites	2
Prerequisite tasks or conditions	3
Installation	4
Product migration installation on RHEL3 U3.	8
Querying for installed packages	9
Enabling the XL C/C++ man pages	9
Setting the correct NLSPATH.	10
Configuring the compiler	10
Viewing the documentation	13
Setting up short compiler invocation commands.	14
Creating a symbolic link for a compiler in the /usr/bin subdirectory	14
Modifying the PATH environment variable	14
Testing the installation	15
Uninstalling XL C/C++ V7.0	15
 <b>Chapter 2. Procedures for advanced users</b>	17
Installing multiple compiler versions on the same RHEL3 U3 system	17
Using the latest run-time environment with multiple C/C++ compiler versions (RHEL3 U3 only)	18
Trying out a PTF update to the compiler	19
 <b>Chapter 3. Troubleshooting the configuration</b>	21
Error: Could not determine location of 32-bit or 64-bit GCC (RHEL3 U3, RHEL4)	21
Error: Could not run the license acceptance tool (RHEL4, SLES9, Y-HPC)	22
Problem: There is not enough debugging information.	22
 <b>Notices</b>	25
Trademarks and service marks	26



---

## Chapter 1. Installing the product

This document contains essential information about installing IBM XL C/C++ Advanced Edition V7.0 for Linux. Please read it carefully before installing this product. Be sure to read the **README** file on the CD-ROM, which contains the most current information about the product. After you install the product, you can find the **README** file in the `vacpp_path/vacpp/7.0` directory.

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

---

### XL C/C++ V7.0 packaging

The following packages are supplied on the CD. Use `rpm` to review them:

*Table 1. . Required XL C/C++ V7.0 packages*

Package Name	Package description
xlsmg.msg.rte	SMP run-time messages
xlsmg.rte	SMP run-time dynamic libraries
xlsmg.lib	SMP run-time static libraries
xlmass.lib	IBM® Mathematical Acceleration Subsystem (MASS) libraries
vac.lic	XL C/C++ V7.0 license
vacpp.rte	C++ run-time environment
vacpp.rte.lnk	C++ run-time environment links
vac.lib	C compiler libraries
vac.cmp	C compiler
vacpp.lib	C++ compiler libraries
vacpp.cmp	C++ compiler
vacpp.samples	XL C/C++ V7.0 samples
vacpp.help	XL C/C++ V7.0 documentation

The supported locales are as follows:

- 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

---

## Installing IBM XL C/C++ Advanced Edition V7.0 for Linux

The IBM XL C/C++ Advanced Edition V7.0 for Linux installation uses the Red Hat Package Manager (RPM), which is the standard tool for installing Linux™ packages. RPM is shipped with the operating system.

The high-level steps are:

1. Become the root user, or a user with administrator privileges.
2. Install the packages.
3. Enable the compiler man pages.
4. Set the correct NLSPATH.
5. Configure the compiler.
6. Set up the environment for the invocation commands.
7. Test the installation.

### System prerequisites

- **Operating system:** A supported Linux distribution:
  - Red Hat Enterprise Linux AS 3 Update 3 (RHEL3), IBM eServer iSeries and IBM eServer pSeries Edition
  - Red Hat Enterprise Linux AS 4 (RHEL4), IBM eServer iSeries and IBM eServer pSeries Edition
  - SUSE LINUX Enterprise Server 9 (SLES9) for IBM POWER
  - Terra Soft's Y-HPC v20050115 (Y-HPC), or equivalent
- **Hardware:** a system that is supported by your Linux distribution, as shown in the following table:

Table 2. . Hardware systems for supported Linux distributions

Linux distribution	Hardware system
RHEL3 U3, RHEL4, SLES9	One of the following: <ul style="list-style-type: none"><li>• IBM eServer™ OpenPower™ system</li><li>• IBM eServer PowerPC® system</li><li>• IBM eServer BladeCenter™ JS20 system</li><li>• IBM eServer p5 system</li><li>• IBM eServer pSeries® system</li><li>• IBM eServer i5 system</li><li>• IBM eServer iSeries™ system</li></ul>
Y-HPC	One of the following: <ul style="list-style-type: none"><li>• Apple Power Mac G5 system</li><li>• Apple Xserver G5 system</li></ul>

- **Required hard disk space:**
  - Approximately 100 MB for product packages
  - At least 2 GB for paging
  - At least 512 MB for temporary files

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

- **Optional software:**
  - Frames-capable HTML browser (to access help files)
  - PDF viewer (to access documentation)



## Prerequisite tasks or conditions

You must have root user access to install XL C/C++ V7.0.

### Pre-installation required packages

The tables in this section list the GCC and operating system packages that must be installed before you install the compiler.

Table 3. . Required GCC and RHEL3 U3 operating system packages

Package name	Version requirements
gcc	3.2.3
gcc-c++	3.2.3
glibc-devel	2.3.2
libstdc++-devel	3.2.3

Table 4. . Required GCC and RHEL4 operating system packages

Package name	Version requirements
gcc	3.4.3
gcc-c++	3.4.3
glibc-devel	2.3.4
libstdc++-devel	3.4.3
libgcj	3.4.3

Table 5. . Required GCC and SLES9 operating system packages

Package name	Version requirements
gcc	3.3.3
gcc-c++	3.3.3
gcc-64bit	9
glibc-devel-64bit	9
libstdc++-devel-64bit	9
java2	1.3.1
java2-jre	1.3.1

Table 6. . Required GCC and Y-HPC operating system packages

Package name	Version requirements
gcc	3.3.3
gcc-ppc32	3.3.3
gcc-c++	3.3.3
libgcj	3.3.3

### Verifying that the required packages are on the system

You can use the following command to verify that you have the required packages installed:

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

For example, to see whether gcc-c++ is installed, query for the gcc-c++ package as follows:

```
rpm -qa | grep gcc-c++
```

If gcc-c++ version 3.3.3 is installed, you will get a result similar to the following output:

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

**Exception:** On RHEL3 U3 and RHEL4, both 32-bit and 64-bit glibc-devel and libstdc++-devel packages are required. To ensure that you have these packages installed before you install the compiler, follow the instruction in “Error: Could not determine location of 32-bit or 64-bit GCC (RHEL3 U3, RHEL4)” on page 21.

**Note:** You should not use the grep command to check whether these packages are installed because the names of the RPM packages on RHEL3 U3 and RHEL4 are structured so that they have the same name regardless whether they are 32-bit or 64-bit RPM package. As a result, the output does not indicate whether 32-bit, 64-bit or both RPM packages are installed.

## Packages that support specific tasks

Table 7 lists programs and packages that are not required for installation of the product but are required to support specific tasks.

Table 7. . Packages that support specific tasks

Task	Prerequisite RPM package	Query
<i>Required:</i> Configuring the compiler using <b>vac_configure</b>	perl	<b>rpm -qf /usr/bin/perl</b>
<i>Optional:</i> Using the documentation that is packaged with the compiler	RPM packages for graphical desktop environments (such as K Desktop Environment or Gnome) that support web browsers and PDF viewers.	<b>rpm -q mozilla</b> <b>rpm -q kdebase3</b> <b>rpm -q xpdf</b>

## Verifying that there is enough space

XL C/C++ V7.0 packages require about 100 MB of hard disk space. This amount includes the optional samples and documentation.

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

Use the following command to determine the amount of space available for the compiler installation in the default installation location:

```
df -h /opt
```

## Installation

This section describes different ways to install the XL C/C++ V7.0 compiler:

- Installing all the XL C/C++ V7.0 RPM packages with a single command
- Installing each XL C/C++ V7.0 RPM package individually

After installing the RPM packages, you must configure the compiler by running either **new\_install** or **vac\_configure**. See “Configuring the compiler” on page 10.

Table 8 on page 5 lists the packages and their prerequisites for all supported Linux distributions.

Table 8. . XL C/C++ V7.0 packages and their prerequisites

Package name	Prerequisites	Description	Relocation
xlsmp.msg.rte	None	SMP run-time messages	All SMP packages must be installed in the same location.
xlsmp.rte	xlsmp.msg.rte	SMP run-time dynamic libraries	
xlsmp.lib	xlsmp.msg.rte xlsmp.rte	SMP run-time static libraries	
xlmass.lib	None	IBM Mathematical Acceleration Subsystem (MASS) libraries	Any location
vac.lic	None	XL C/C++ V7.0 license	Any location
vacpp.rte	None	C/C++ run-time environment	All C/C++ run-time-related packages must be installed in the same location.
vacpp.rte.lnk	vacpp.rte	C/C++ run-time links	
vac.cmp	vac.lib vac.lic xlmass.lib xlsmp.lib xlsmp.msg.rte xlsmp.rte	C compiler	All C/C++ compiler and library packages must be installed in the same location.
vac.lib	None	C compiler libraries	
vacpp.cmp	vac.cmp vac.lib vac.lic vacpp.lib vacpp.rte vacpp.rte.lnk xlmass.lib xlsmp.lib xlsmp.msg.rte xlsmp.rte	C++ compiler	
vacpp.lib	None	C++ compiler libraries	
vacpp.samples	None.	Example programs	Any location (optional).
vacpp.help	None.	Man pages and compiler documentation in HTML and PDF formats	Any location (optional).

## Installing all the XL C/C++ RPM packages with a single command

If your current working directory contains all of the RPM packages for XL C/C++ V7.0 and no other RPM packages, you can install XL C/C++ V7.0 using a single command.

**Note:** By default, the compiler packages are installed in the `/opt/ibmcmp` directory.

**To install all packages to the default location, use the following command:**

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

**To install all packages to a single non-default location, use the following command:**

```
rpm -ivh *.rpm --prefix relocation_path
```

where `relocation_path` is the location of the product files.

## Installing each XL C/C++ RPM package individually

If your current working directory contains RPM packages in addition to the XL C/C++ V7.0 RPM packages, you must install each XL C/C++ V7.0 RPM package individually. In order to avoid dependency errors when you install packages individually, you must follow the installation order described in “Examples of default installations” or “Examples of non-default installations” on page 7.

**Note:** The commands in this section use the following variables:

- *package\_name* represents any of the packages listed in “Verifying that the required packages are on the system” on page 3.
- *V.R.M-F* represents the Version.Release.Modification-Fix level of the package:
  - Package **xlsmplib** has V.R.M-F **1.5.1-0**
  - Package **xlmasslib** has V.R.M-F **4.1.1-0**
  - Package **vac** has V.R.M-F **7.0.1-0**
  - Package **vacpp** has V.R.M-F **7.0.1-0**
- *arch* represents the hardware platform required by the operating system:
  - Operating system **RHEL3 U3** requires architecture type **ppc64pseries**
  - Operating system **RHEL4** requires architecture type **ppc64pseries**
  - Operating system **SLES9** requires architecture type **ppc64**
  - Operating system **Y-HPC** requires architecture type **ppc64**
- *relocation\_path* is the installation location that you specify.

**To install a package to the default location, use the following command:**

```
rpm -ivh package_name-V.R.M-F.arch.rpm
```

**To install a package to the non-default location, use the following command:**

```
rpm -ivh package_name-V.R.M-F.arch.rpm --prefix relocation_path
```

## Examples of default installations

**Default installation on RHEL3 U3 or RHEL4:** Issue the following commands to install XL C/C++ V7.0 to the default location, **/opt/ibmcomp**. In order to avoid dependency errors during installation of XL C/C++ V7.0 on RHEL3 U3 or RHEL4, issue the following commands in the order given:

```
rpm -ivh xlsmplib.msg.rte-1.5.1-0.ppc64pseries.rpm
rpm -ivh xlsmplib.rte-1.5.1-0.ppc64pseries.rpm
rpm -ivh xlsmplib.lib-1.5.1-0.ppc64pseries.rpm
rpm -ivh xlmasslib.lib-4.1.1-0.ppc64pseries.rpm
rpm -ivh vac.lib-7.0.1-0.ppc64pseries.rpm
rpm -ivh vac.lib-7.0.1-0.ppc64pseries.rpm
rpm -ivh vac.comp-7.0.1-0.ppc64pseries.rpm
rpm -ivh vacpp.rte-7.0.1-0.ppc64pseries.rpm
rpm -ivh vacpp.rte.lnk-7.0.1-0.ppc64pseries.rpm
rpm -ivh vacpp.lib-7.0.1-0.ppc64pseries.rpm
rpm -ivh vacpp.comp-7.0.1-0.ppc64pseries.rpm
```

**Note:** If you have already installed the XL Fortran compiler, **xlsmplib.msg.rte**, **xlsmplib.rte**, **xlsmplib.lib** and **xlmasslib** will already be installed. RPM prevents you from installing these components a second time.

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

```
rpm -ivh vacpp.help-7.0.1-0.ppc64pseries.rpm
rpm -ivh vacpp.samples-7.0.1-0.ppc64pseries.rpm
```

**Default installation on SLES9 or Y-HPC:** Issue the following commands to install XL C/C++ V7.0 to the default location, `/opt/ibmcomp`. In order to avoid dependency errors during installation of XL C/C++ V7.0 on SLES9 or Y-HPC, issue the following commands in the order given:

```
rpm -ivh xlsmp.msg.rte-1.5.1-0.ppc64.rpm
rpm -ivh xlsmp.rte-1.5.1-0.ppc64.rpm
rpm -ivh xlsmp.lib-1.5.1-0.ppc64.rpm
rpm -ivh xlmass.lib-4.1.1-0.ppc64.rpm
rpm -ivh vac.lic-7.0.1-0.ppc64.rpm
rpm -ivh vac.lib-7.0.1-0.ppc64.rpm
rpm -ivh vac.cmp-7.0.1-0.ppc64.rpm
rpm -ivh vacpp.rte-7.0.1-0.ppc64.rpm
rpm -ivh vacpp.rte.lnk-7.0.1-0.ppc64.rpm
rpm -ivh vacpp.lib-7.0.1-0.ppc64.rpm
rpm -ivh vacpp.cmp-7.0.1-0.ppc64.rpm
```

**Note:** If you have already installed the XL Fortran compiler, `xlsmp.msg.rte`, `xlsmp.rte` and `xlsmp.lib` will already be installed. RPM prevents you from installing these components a second time.

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

```
rpm -ivh vacpp.help-7.0.1-0.ppc64.rpm
rpm -ivh vacpp.samples-7.0.1-0.ppc64.rpm
```

## Examples of non-default installations

In the examples in this section, there are different directories for each group of packages that must be installed together. Table 8 on page 5 lists these package groups. The variables that represent each of these directories are:

- `smprrt_path` represents the directory that contains all SMP packages.
- `xlmass_path` represents the directory that contains the MASS library package.
- `vaclic_path` represents the directory that contains the XL C/C++ product license.
- `vacpp_path` represents the directory that contains the XL C/C++ compiler and library packages.
- `vacpprrt_path` represents the directory that contains the XL C/C++ run-time-related packages.
- `vacppdoc_path` represents the directory that contains the XL C/C++ guides and online help.
- `vacppsmpls_path` represents the directory that contains the XL C/C++ program samples that are referenced in the documentation.

**Non-default installation on RHEL3 U3 or RHEL4:** In order to avoid dependency errors during installation of XL C/C++ V7.0 on RHEL3 U3 or RHEL4, issue the following commands in the order given:

```
rpm -ivh xlsmp.msg.rte-1.5.1-0.ppc64pseries.rpm --prefix smprrt_path
rpm -ivh xlsmp.rte-1.5.1-0.ppc64pseries.rpm --prefix smprrt_path
rpm -ivh xlsmp.lib-1.5.1-0.ppc64pseries.rpm --prefix smprrt_path
rpm -ivh xlmass.lib-4.1.1-0.ppc64pseries.rpm --prefix xlmass_path
rpm -ivh vac.lic-7.0.1-0.ppc64pseries.rpm --prefix vaclic_path
rpm -ivh vac.lib-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path
rpm -ivh vac.cmp-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path
rpm -ivh vacpp.rte-7.0.1-0.ppc64pseries.rpm --prefix vacpprrt_path
```

```
rpm -ivh vacpp.rte.lnk-7.0.1-0.ppc64pseries.rpm --prefix vacpprt_path
rpm -ivh vacpp.lib-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path
rpm -ivh vacpp.cmp-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path
```

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

```
rpm -ivh vacpp.help-7.0.1-0.ppc64pseries.rpm --prefix vacppdoc_path
rpm -ivh vacpp.samples-7.0.1-0.ppc64pseries.rpm --prefix vacppsmppls_path
```

**Non-default installation on SLES9 or Y-HPC:** In order to avoid dependency errors during installation of XL C/C++ V7.0 on SLES9 or Y-HPC, issue the following commands in the order given:

```
rpm -ivh xlsmp.msg.rte-1.5.1-0.ppc64.rpm --prefix smprt_path
rpm -ivh xlsmp.rte-1.5.1-0.ppc64.rpm --prefix smprt_path
rpm -ivh xlsmp.lib-1.5.1-0.ppc64.rpm --prefix smprt_path
rpm -ivh xlmass.lib-4.1.1-0.ppc64.rpm --prefix xlmass_path
rpm -ivh vac.lic-7.0.1-0.ppc64.rpm --prefix vaclic_path
rpm -ivh vac.lib-7.0.1-0.ppc64.rpm --prefix vacpp_path
rpm -ivh vac.cmp-7.0.1-0.ppc64.rpm --prefix vacpp_path
rpm -ivh vacpp.rte-7.0.1-0.ppc64.rpm --prefix vacpprt_path
rpm -ivh vacpp.rte.lnk-7.0.1-0.ppc64.rpm --prefix vacpprt_path
rpm -ivh vacpp.lib-7.0.1-0.ppc64.rpm --prefix vacpp_path
rpm -ivh vacpp.cmp-7.0.1-0.ppc64.rpm --prefix vacpp_path
```

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

```
rpm -ivh vacpp.help-7.0.1-0.ppc64.rpm --prefix vacppdoc_path
rpm -ivh vacpp.samples-7.0.1-0.ppc64.rpm --prefix vacppsmppls_path
```

## Product migration installation on RHEL3 U3

Product migration installation does not apply to RHEL4, SLES9 or Y-HPC because V7.0 is the first version of XL C/C++ supported on those operating systems.

**Note:** If you have a previous version of the compiler installed on the system, you must use the **vac\_configure** utility to configure the compiler. Do not use **new\_install** if you have more than one version of the product installed.

### Version upgrades

If you have the base version of the compiler installed, it is highly recommended that you uninstall it before you install a new version of the product.

If you want to have more than one version of the compiler on the system, refer to instructions in Chapter 2, "Procedures for advanced users," on page 17.

### PTF updates

Before applying PTF updates, it is highly recommended that you uninstall any previous compiler packages.

#### Notes:

1. With the PTF (Program Temporary Fix) updates, all packages except **vac.lic** are shipped. You must use the **vac.lic** you have from your base version.
2. When applying PTF updates, you can test the PTF update before uninstalling the previous Fix level. In this case, you must install the PTF to a different location. For instructions, see Chapter 2, "Procedures for advanced users," on page 17.

## Querying for installed packages

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

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

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

If the package has been installed properly, the result should be:

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

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

To query all compiler packages, issue the following command:

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

If none of the packages are installed, the output of the command will be null.

## Enabling the XL C/C++ man pages

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

Before you can read the man pages, you must add the full path to the XL C/C++ MANPATH environment variable, as follows:

1. Open a terminal window.
2. Depending on the shell you are using, enter either of the following commands:
  - If you are using the bash or ksh shell:

```
export LANG=locale
```
  - If you are using the csh shell:

```
setenv LANG locale
```

The supported locales for manual pages are as follows:

- en\_US
- en\_US.utf8
- ja\_JP
- ja\_JP.eucjp

For example, to view man pages in English, enter either of the following commands:

- If you are using the bash shell:

```
export LANG=en_US
```
  - If you are using the csh shell:

```
setenv LANG en_US
```
3. Add **/opt/ibmcmp/vacpp/7.0/man** to the beginning of the MANPATH environment variable, as shown in the following table.

Table 9. . How to update the MANPATH environment variable

Shell	Command
bash or ksh	<b>export MANPATH=/opt/ibmcmp/vacpp/7.0/man:\$MANPATH</b>
csh	<b>setenv MANPATH /opt/ibmcmp/vacpp/7.0/man:\$MANPATH</b>

To invoke a man page, enter **man** followed by the command you want to see information about; for example:

**man xlc.**

To leave the man page, type: **q**.

## Setting the correct NLSPATH

After installation, you must set the NLSPATH environment variable so that the run-time environment finds the appropriate message catalogs.

Issue the following command:

```
export NLSPATH=$NLSPATH:  
    smprt_path/msg/%L/%N:  
    vacpprt_path/msg/%L/%N:  
    vacpp_path/vacpp/7.0/msg/%L/%N
```

where:

- *smprt\_path* is the installation location of the SMP run-time
- *vacpprt\_path* is the installation location of the C++ run-time environment
- *vacpp\_path* is the installation location of the C and C++ compilers

**Note:** If the default installation location is used, *smprt\_path*, *vacpprt\_path*, and *vacpp\_path* will all be **/opt/ibmcomp**.

## Configuring the compiler

If only one version of the compiler is installed on your system, you should use the **new\_install** utility to configure the compiler. The **new\_install** utility automatically backs up any existing configuration file and invokes the **vac\_configure** utility. In order to run **new\_install**, you must have root or administrator privileges.

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

- You receive an error from the **new\_install** command. (See Chapter 3, “Troubleshooting the configuration,” on page 21.)
- You want to change the generated (default) configuration file (vac.cfg).
- You want to have multiple configuration files.
- You have multiple versions of GCC are installed on your system and you need to use **vac\_configure** to specify which GCC version you would like to reference in the configuration file.

For instructions, see “Running the vac\_configure utility directly” on page 12.

**Note:** If you configure the compiler using **vac\_configure**, your output configuration file, vac.cfg, can be installed in a location where you have write permission. You would not need root or administrative privileges.

The following table describes some of the attributes in the generated configuration file.



Table 10. . Linux-specific configuration attributes

Attribute	Contents	Additional information
gcc_path gcc_path_64	The location of the GCC executable, in 32-bit mode or 64-bit mode	The <b>gcc</b> command must be located in the <b>bin</b> directory under the specified path.
gcc_libs gcc_libs_64	A comma-separated list of GCC libraries, in 32-bit mode or 64-bit mode	The <b>gcc</b> returns this list to the <b>vac_configure</b> utility.
gcc_libdirs gcc_libdirs_64	A comma-separated list of directories that contain GCC libraries, in 32-bit mode or 64-bit mode	The <b>gcc</b> returns this list to the <b>vac_configure</b> utility.

## Running the new\_install utility

If only one version of the compiler is installed on your system, you should use the **new\_install** utility to install and configure the compiler.

The **new\_install** utility:

- Backs up any existing configuration file.
- Queries the RPM database for the paths to the XL C/C++ packages and the path to the 32-bit GCC (*gcc32path*) and the 64-bit GCC (*gcc64path*) in the PATH environment variable, and then uses the obtained values to execute the **vac\_configure** utility.
- Generates the **/etc/opt/ibmcmp/vac/7.0/vac.cfg** configuration file.

When running **new\_install** on RHEL4, SLES9 or Y-HP, you will be required to accept the license agreement before the configuration file is generated. In order to run the license acceptance tool, you must have installed the Java™ Runtime Environment, as specified in Table 7 on page 4

**Note:** This step is not required for RHEL3 U3.

### To run the new\_install utility:

1. Change to the directory that contains the **new\_install** and **vac\_configure** executables:  

```
cd /opt/ibmcmp/vacpp/7.0/bin
```

or, if you installed XL C/C++ in a non-default location:  

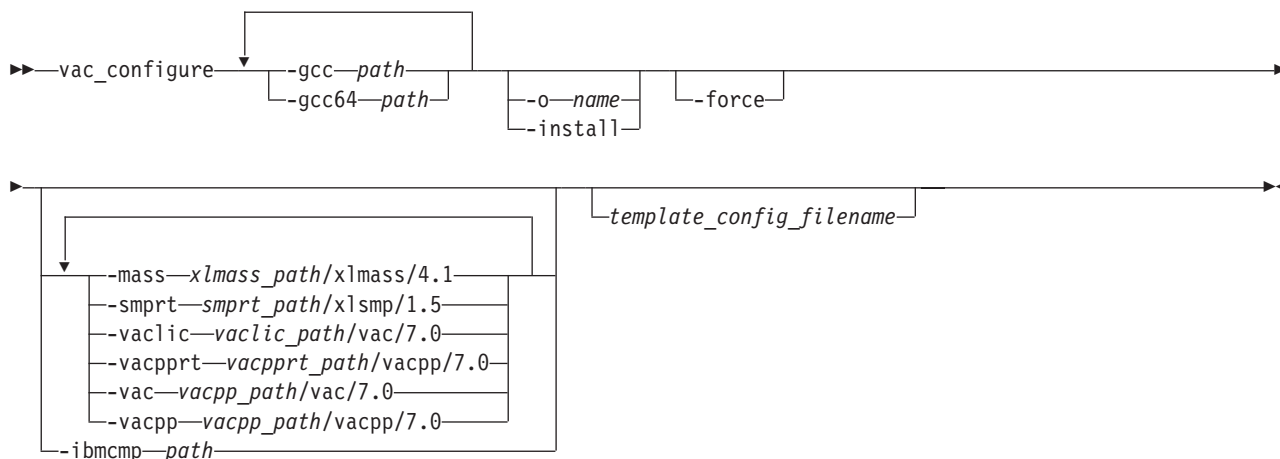
```
cd vacpp_path/vacpp/7.0/bin
```
2. Run the following command:  

```
./new_install
```

The **new\_install** command executes the following command:

```
vac_configure
-gcc gcc32path
-gcc64 gcc64path
-install
-mass xlmass_path/xlmass/4.1
-smprt smprt_path/xlsmprt/1.5
-vaclic vaclic_path/vac/7.0
-vacpprt vacpprt_path/vacpp/7.0
-vac vacpp_path/vac/7.0
-vacpp vacpp_path/vacpp/7.0
vacpp_path/vac/7.0/etc/vac.base.cfg
```

The **vac\_configure** utility is automatically invoked by the **new\_install** command.



**-gcc path**

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

**-gcc /usr**

**-gcc64 path**

Specifies the path where the 64-bit GCC bin directory is installed. In the configuration file, the *gcc path 64* attribute is set equal to this path.

**-o** *name*

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

**-install**

Generates a default compiler configuration file `vac.cfg` and places it in the default location `/etc/opt/ibmcomp/vac/7.0/vac.cfg`.

**-force** Forces the **vac\_configure** utility to overwrite any existing output file with the specified name and path. By default, if you don't use *force*, **vac\_configure** issues an error message and stops if the specified file already exists.

```
-mass xlmass_path/xlmass/4.1
```

Specifies the path for the **xlmass.lib** package. By default the path is **/opt/ibmcmp/xlmass/4.1**.

**-smprt** *smprt\_path/xlsmp/1.5*

Specifies the path for the **xlsmp.msg.rte**, **xlsmp.rte**, and **xlsmp.lib** packages. By default, the path is **/opt/ibmcmp/xlsmp/1.5**.

**-vaclic** *vaclic\_path/vac/7.0*

Specifies the path for the **vac.lic** package. By default, this is **/opt/ibmcmp/vac/7.0**.

**-vacpprt** *vacpprt\_path/vacpp/7.0*

Specifies the path for the **vacpp.rte** and **vacpp.rte.lnk** package. By default, this is **/opt/ibmcmp/vacpp/7.0**.

**-vacpp** *vacpp\_path/vacpp/7.0*

Specifies the path for the **vacpp.cmp** package. By default, this is **/opt/ibmcmp/vacpp/7.0**.

**-vac** *vacpp\_path/vac/7.0*

Specifies the path for the **vac.cmp** package. By default, this is **/opt/ibmcmp/vac/7.0**.

**-ibmcmp** *relocation\_path*

Alternatively specifies the path where all of the XL C/C++ packages (**xlsmp.msg.rte**, **xlsmp.rte**, **xlsmp.lib**, **vac.lic**, **vacpp.rte**, **vacpp.rte.lnk**, **vac.cmp**, and **vacpp.cmp** ) are installed. In this case, *relocation\_path* represents *smprrt\_path*, *vaclic\_path*, *vacpprt\_path*, and *vacpp\_path*.

**Note:** If you want to install packages in different subdirectories, you should not use **ibmcmp**.

*template\_config\_filename*

The input file that is used to construct the configuration file. By default, this is **/opt/ibmcmp/vac/7.0/etc/vac.base.cfg**. If you relocated the **vac.cmp** package to the *vacpp\_path* location but want to use the default template, specify:

*vacpp\_path/vac/7.0/etc/vac.base.cfg*.

---

## Viewing the documentation

The following documentation is provided with XL C/C++ V7.0:

**README file** A README file is located in the root directory of the installation CD. This file is installed in the **/opt/ibmcmp/vacpp/7.0** directory.

**PDF books** On the installation CD, the PDF versions of the XL C/C++ documentation are stored in the **/doc/language\_path/pdf** directories. When you install the **vacpp.help** package, the PDF files are installed in the corresponding directory on the system. For example, if you install the **vacpp.help** package to the default location:

- The English PDF is installed in the **/opt/ibmcmp/vacpp/7.0/doc/en\_US/pdf** directory.
- The Japanese PDF is installed in the **/opt/ibmcmp/vacpp/7.0/doc/ja\_JP/pdf** directory.
- The Chinese PDF is installed in the **/opt/ibmcmp/vacpp/7.0/doc/zh\_CN/pdf** directory.

**Note:** You need a PDF viewer to view these files. The **xpdf** viewer is included in each supported Linux distribution.

**HTML files** On the installation CD, the HTML version of the XL C/C++ documentation is stored in the **/doc/language\_path/html** directories. When you install the **vacpp.help** package, the HTML files are

installed to the corresponding directory on the system. For example, if you install the `vacpp.help` package to the default location:

- The English HTML files are installed in the `/opt/ibmcmp/vacpp/7.0/doc/en_US/html` directory.
- The Japanese HTML files are installed in the `/opt/ibmcmp/vacpp/7.0/doc/ja_JP/html` directory.
- The Chinese HTML files are installed in the `/opt/ibmcmp/vacpp/7.0/doc/zh_CN/html` directory.

**Note:** To view the HTML files, open the file `index.htm` in a web browser such as Mozilla or Konqueror.

#### Man pages

Man pages are provided for the compiler invocation commands (such as `xlc`) and the following additional commands: `vac_configure`, `new_install`, `gxlc`, `gxlc++`, `showpdf`, `mergepdf`, `resetpdf`, and `cleanpdf`. For example, if you install the man pages to the default location:

- English man pages are installed in `/opt/ibmcmp/vacpp/7.0/man/en_US/man1`.
- Japanese man pages are installed in `/opt/ibmcmp/vacpp/7.0/man/ja_JP/man1`.
- Chinese man pages are installed in `/opt/ibmcmp/vacpp/7.0/man/zh_CN/man1`.

---

## Setting up short compiler invocation commands

If you want to be able to invoke a compiler without having to specify the full path, you must do either of the following:

- Create a symbolic link for the compiler.

**Note:** The `/usr/bin` directory is typically used for this purpose.

- Modify the `PATH` environment variable.

### Creating a symbolic link for a compiler in the `/usr/bin` subdirectory

To create a symbolic link for one of the XL C/C++ compiler invocations (for example `xlc`) that is installed in the `vacpp_path`, use the following command:

```
ln -s vacpp_path/vacpp/7.0/bin/xlc /usr/bin/xlc
```

Example: if the compiler is in the default location, use the following command:

```
ln -s /opt/ibmcmp/vacpp/7.0/bin/xlc /usr/bin/xlc
```

### Modifying the `PATH` environment variable

To modify the `PATH` environment variable for XL C/C++, add the target path (`vacpp_path/vacpp/7.0/bin`) to the `PATH` environment variable:

**Example:** If the compiler is in the default location, use the following command to modify the `PATH` environment variable:

```
export PATH=$PATH:/opt/ibmcmp/vacpp/7.0/bin
```

---

## Testing the installation

To test the product install and the critical search paths, try building the following simple C and C++ programs.

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. Use the `xc` command to compile the test program. For example:

```
/opt/ibmcmp/vac/7.0/bin/xc hello.c -o hello
```

3. Run the program:

```
./hello
```

The expected result is that "Hello World!" is displayed on the screen.

4. Check the exit code of the program:

```
echo $?
```

The result should be 0.

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

```
#include <iostream>
int main()
{
    std::cout << "Hello World!" << std::endl;
    return 0;
}
```

6. Use the `xc` command to compile the test program. For example:

```
/opt/ibmcmp/vacpp/7.0/bin/xc hello.cpp -o hello
```

7. Run the program:

```
./hello
```

The expected result is that "Hello World!" is displayed on the screen.

8. Check the exit code of the program:

```
echo $?
```

The result should be 0.

---

## Uninstalling XL C/C++ V7.0

### Notes:

1. You must have root user access to uninstall this product.
2. Whenever you uninstall a package, specify the *V.R.M-F* (Version.Release.Modification-Fix level) of the package.
3. Always uninstall packages in the reverse of that in which they were installed (that is, LIFO).
4. You cannot uninstall packages that are required by other packages. For example, the SMP runtime (`xlsmp.rte`) might be a shared component if IBM XL Fortran is also installed on the same system.
5. The package removal commands will not remove any configuration files that were generated by **new\_install** or **vac\_configure**.

Uninstall the compiler packages in the following order to avoid dependency errors during uninstallation:

- **rpm -e vacpp.cmp-7.0.1-0**

- **rpm -e vacpp.lib-7.0.1-0**
- **rpm -e vacpp.rte.lnk-7.0.1-0**
- **rpm -e vacpp.rte-7.0.1-0**
- **rpm -e vac.cmp-7.0.1-0**
- **rpm -e vac.lic-7.0.1-0**
- **rpm -e vac.lib-7.0.1-0**
- **rpm -e xlmass.lib-4.1.1-0**
- **rpm -e xlsmp.lib-1.5.1-0**
- **rpm -e xlsmp.rte-1.5.1-0**
- **rpm -e xlsmp.msg.rte-1.5.1-0**

The sample programs and product documentation do not have any package dependencies. To uninstall them, you can issue the following commands in any order.

**rpm -e vacpp.samples-7.0.1-0**

**rpm -e vacpp.help-7.0.1-0**

---

## Chapter 2. Procedures for advanced users

If you have a previous version of the compiler installed, it is highly recommended that you uninstall it before you install a new version of the product. These procedures are for users who are

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

---

### Installing multiple compiler versions on the same RHEL3 U3 system

**Note:** You need a separate license for each user.

You might want to install multiple versions of the compiler on a system to satisfy the requirements of the following scenarios:

- There is already an existing version (VisualAge® C++ V6.0 for example) of the compiler installed in the default location, **/opt/ibmcmp** and:
  - You want to keep using the existing version as you migrate to the new version.
  - You want to install XL C/C++ 7.0 packages to a different location.
  - You want each version of the compiler to use the run-time environment that was created for it.

For this scenario, follow the instructions in “Non-default installation on RHEL3 U3 or RHEL4” on page 7. Your existing compiler version will use the corresponding version of the run-time environment and your new compiler version will use its corresponding run-time environment. For example, if you have VisualAge C++ V6.0 already installed, it will continue to use V6.0 of the run-time environment. When you install XL C/C++ V7.0 to another location on the system, it will use XL C/C++ V7.0 of the run-time environment.

- There is already an existing version (VisualAge C++ V6.0, for example) of the compiler installed in a non-default location and:
  - You want to keep using the existing version as you migrate to the new version.
  - You want to install XL C/C++ V7.0 packages to the default location.
  - You want each version of the compiler to use the run-time environment that was created for it.

For this scenario, follow the instructions in “Default installation on RHEL3 U3 or RHEL4” on page 6. Your existing compiler version will use the corresponding version of the run-time environment and your new compiler version will use its corresponding run-time environment. For example, if you have VisualAge C++ V6.0 already installed, it will continue to use V6.0 of the run-time environment. When you install XL C/C++ 7.0 to the default location, XL C/C++ 7.0 will use XL C/C++ V7.0 of the run-time environment.

- With two versions of the compiler installed, you want each version to use the more current run-time environment.

For this scenario, follow the instructions in “Using the latest run-time environment with multiple C/C++ compiler versions (RHEL3 U3 only)” on page 18. Both your existing compiler version and the newly installed compiler version will use the newly-installed run-time environment. For example, when you install XL C/C++

7.0 to the default location, it will use V7.0 of the run-time environment. If you have VisualAge C++ V6.0 already installed, it will also use XL C/C++ V7.0 of the run-time environment.

- You want to try out the latest PTF before committing to it.

For this scenario, follow the instructions in “Trying out a PTF update to the compiler” on page 19.

## Using the latest run-time environment with multiple C/C++ compiler versions (RHEL3 U3 only)

If you want to have both Visual Age C/C++ V6.0 and XL C/C++ V7.0 installed on the same system and you would like both to use the more current run-time environment, you must ensure that the newer run-time-related packages overwrite the existing VisualAge C++ V6.0 run-time-related packages.

**Note:** To avoid dependency errors while installing the newer XL C/C++ V7.0 run-time environment and compiler, use the steps in “Example: Steps to overwrite the existing run-time environment.”

The following table lists the run-time-related packages for both VisualAge C++ V6.0 and XL C/C++ V7.0.

Table 11. . Run-time-related packages

VisualAge C++ V6.0 run-time-related packages	XL C/C++ 7.0 run-time-related packages
xlsmp.msg.rte-1.3.M-F	xlsmp.msg.rte-1.5.M-F
xlsmp.rte-1.3.M-F	xlsmp.rte-1.5.M-F
xlsmp.lib-1.3.M-F	xlsmp.lib-1.5.M-F
vacpp.rte 6.0.m-f	vacpp.rte 7.0.m-f
	vacpp.rte.lnk 7.0.m-f

**Note:** These package names contain the following variables:

- *M* represents the Mod level of the SMP package
- *F* represents the Fix level of the SMP package
- *m* represents the Mod level of the compiler package
- *f* represents the Fix level of the compiler package

### Example: Steps to overwrite the existing run-time environment

This example uses the following scenario:

- VisualAge C++ V6.0 is installed in the default location, /opt/ibmcmp and:
  - V.R.M-F level for xlsmp.\* is 1.3.7–2.
  - V.R.M-F level for vacpp.\* is 6.0.1–1

.

- You need to update the existing SMP packages to 1.5.1-0.
- You need to update the existing VisualAge C++ V6.0 run-time packages to 7.0.1-0.

**Note:** The **--prefix path** utility for the packages that are not related to XL C/C++ V7.0 can be omitted if you also install them to the default location.

For this scenario, use the following steps:



1. In order to avoid future dependency errors, uninstall the existing run-time-related packages by issuing the following commands:  
**`rpm -e xlsmp.msg.rte-1.3.7-2 --nodeps`**
2. In order to replace the existing run-time-related packages, install the XL C/C++ V7.0 compiler by issuing the following commands:  
**`rpm -ivh xlsmp.msg.rte-1.5.1-0.ppc64pseries.rpm`**  
**`rpm -ivh xlsmp.rte-1.5.1-0.ppc64pseries.rpm --force`**  
**`rpm -ivh xlsmp.lib-1.5.1-0.ppc64pseries.rpm`**  
**`rpm -ivt xlmass.lib-4.1.1-0.ppc64pseries.rpm --prefix xlmass_path`**  
**`rpm -ivh vac.lic-7.0.1-0.ppc64pseries.rpm --prefix vaclic_path`**  
**`rpm -ivh vac.lib-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path`**  
**`rpm -ivh vac.cmp-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path`**  
**`rpm -ivh vacpp.rte-7.0.1-0.ppc64pseries.rpm --force`**  
**`rpm -ivh vacpp.rte.lnk-7.0.1-0.ppc64pseries.rpm`**  
**`rpm -ivh vacpp.cmp-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path`**  
**`rpm -ivh vacpp.help-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path`**  
**`rpm -ivh vacpp.samples-7.0.1-0.ppc64pseries.rpm --prefix vacpp_path`**
3. Invoke the **`vac_configure`** utility as follows:  
**`vacpp_path/vac/7.0/bin/vac_configure`**  
**`-gcc /usr`**  
**`-gcc64 /usr`**  
**`-mass xlmass_path/xlmass/4.1`**  
**`-smprt /opt/ibmcmp/xlsmp/1.5`**  
**`-vac vacpp_path/vac/7.0`**  
**`-vacpp vacpp_path/vacpp/7.0`**  
**`-vacpprt /opt/ibmcmp/vacpp/7.0`**  
**`-vaclic vaclic_path/vac/7.0`**  
**`vacpp_path/vac/7.0/etc/vac.base.cfg`**  
**`-o /etc/opt/ibmcmp/vac/7.0/vac.cfg`**

---

## Trying out a PTF update to the compiler

When applying PTF updates, you can test the PTF update before uninstalling the previous Fix level. You must install the PTF to a different location (follow the instructions in “Installation” on page 4). In this case, there will be no need to reinstall `vac.lic`.

Later, when you have verified that you want the PTF and are ready to uninstall the previous version, you can follow the instructions in “Uninstalling XL C/C++ V7.0” on page 15.

### Notes:

1. Do not uninstall `vac.lic`. If you try to uninstall this package, you will receive an error message because this package is a prerequisite for other packages that are installed on your system.
2. Do not use the **`new_install`** utility if you do not uninstall the previous version of the compiler. You must use **`vac_configure`** utility.

3. If you do not uninstall the base version of the compiler from the system, and attempt to apply the PTF in the same location, the installation fails. The RPM utility prevents you from overwriting the existing packages because the location of the files are the same.

---

## Chapter 3. Troubleshooting the configuration

This chapter describes possible problems you may encounter when you configure XL C/C++ V7.0.

---

### Error: Could not determine location of 32-bit or 64-bit GCC (RHEL3 U3, RHEL4)

#### Scenario

You are running either the **new\_install** or the **vac\_configure** utility to configure the compiler on a computer running RHEL3 U3 or RHEL4 when you get at least one of the following error messages:

#### Error messages

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

#### Explanation

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

- glibc-devel
- libstdc++-devel

On RHEL3 U3 and RHEL4, you cannot tell whether the 32-bit or the 64-bit version of GCC is installed by querying the packages because both the 32-bit and 64-bit packages have exactly the same RPM name.

#### Action

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

#### Example

This example uses instances of the classic "Hello World" test case. For instructions to create the C/C++ files required to perform this test, see "Testing the installation" on page 15.

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

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

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

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

3. If the programs cannot be compiled successfully, it means that you need to install the necessary packages and configure the compiler again:
  - a. If you get an error message from 32-bit mode testing, install the required packages, using the following commands:

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

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

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

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

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

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

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

- c. Run `new_install` or `vac_configure` again.

---

## Error: Could not run the license acceptance tool (RHEL4, SLES9, Y-HPC)

### Scenario

You are attempting to run the `new_install` utility on a computer running any supported Linux distribution other than RHEL3 U3 and you get the following error message.

### Error message

```
ERROR: Could not run the license acceptance tool. Please ensure you have installed either libgcj, or both java2 and java2-jre RPM(s) from your operating system installation media.
```

### Explanation

Java run-time environment is required to run the `new_install` utility.

### Action

Verify that the Java Runtime Environment is installed. For instructions, refer to “Verifying that the required packages are on the system” on page 3. Then run `new_install` again.

---

## Problem: There is not enough debugging information

### Scenario

You are experiencing unexpected results when running `new_install` or `vac_configure` and you need more information than the system is providing.

### Action

Use one of the following options to get a more verbose output:

- **-v** (verbose) provides some debug information.
- **-vv** (very verbose) provides a more detailed debug information than **-v**.



---

## Notices

This information was developed for products and services offered in the U.S.A.

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

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

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

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

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

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

Lab Director  
IBM Canada Ltd. Laboratory  
B3/KB7/8200/MKM  
8200 Warden Avenue  
Markham, Ontario L6G 1C7  
Canada

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

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

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

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

#### COPYRIGHT LICENSE:

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

---

## Trademarks and service marks

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

- BladeCenter
- eServer
- IBM
- iSeries
- OpenPower



- PowerPC
- pSeries
- VisualAge

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

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

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







Program Number: 5724-K77

GC09-7946-01

