

IBM XL C/C++ Advanced Edition for Linux
V9.0



XL C/C++ 入门

IBM XL C/C++ Advanced Edition for Linux
V9.0



XL C/C++ 入门

注意！

在使用本资料及其支持的产品之前，请确保阅读第 19 页的『声明』中的一般信息。

第一版

本版本适用于 IBM XL C/C++ Advanced Edition for Linux V9.0（程序号 5724-S73）及所有后续发行版和修订版，直到在新版本中另有声明为止。确保您使用的是本产品级别的正确版本。

© Copyright International Business Machines Corporation 1998, 2007. All rights reserved.

目录

关于本文档	v
本文档的适用对象	v
如何使用本文档	v
本文档中使用的约定	v
相关信息	viii
IBM XL C/C++ 出版物	viii
其他 IBM 出版物	ix
其他出版物	ix
技术支持	ix
如何发送您的意见	ix

第 1 章 XL C/C++ 简介	1
与其他 IBM 编译器的通用性	1
硬件和操作系统支持	1
高度可配置的编译器	1
语言标准一致性	2
与 GNU 的兼容性	2
源代码迁移和一致性检查	3
库	3
Mathematical Acceleration Subsystem 库	3
基本线性代数子程序	4
工具和实用程序	4
程序优化	4
64 位对象能力	5
共享内存并行化	5
OpenMP 伪指令	6
诊断列表	6
符号调试器支持	6

第 2 章 IBM XL C/C++ Advanced Edition for Linux V9.0 的新增内容	7
与 C/C++ 语言相关的更新	7
为 C - extc99 更改的缺省语言级别	7
使用 long long 数据类型进行算术转换	7
体系结构和处理器支持	7

-qtune 的新缺省设置	8
对 POWER6 处理器的新支持	8
性能和优化	8
与性能相关的编译器选项和伪指令	8
本发行版新增的内置函数	9
其他新增或已更改的编译器选项	10

第 3 章 设置和定制 XL C/C++	11
使用定制编译器配置文件	11
确定安装了哪种级别的 XL C/C++	11

第 4 章 使用 XL C/C++ 开发应用程序	13
编译器阶段	13
编辑 C/C++ 源文件	13
使用 XL C/C++ 进行编译	13
调用编译器	13
编译并行化的 XL C/C++ 应用程序	14
指定编译器选项	14
XL C/C++ 输入和输出文件	16
将已编译的应用程序与 XL C/C++ 链接在一起	16
分步进行编译和链接	16
动态和静态链接	16
运行已编译的应用程序	17
取消执行	17
设置运行时选项	17
在其他系统上运行已编译的应用程序	17
XL C/C++ 编译器诊断辅助	18
调试已编译的应用程序	18

声明	19
商标和服务标记	20
业界标准	21

索引	23
---------------------	-----------

关于本文档

本文档包含 IBM® XL C/C++ Advanced Edition for Linux® V9.0 编译器的概述和基本用法信息。

本文档的适用对象

本文档是为那些想要了解 XL C/C++ 的介绍性概述和用法信息的 C 和 C++ 开发者编写的。它假设您已经较为熟悉命令行编译器，掌握基本的 C 和 C++ 编程语言知识和基本的操作系统命令知识。对 XL C/C++ 不熟悉的程序员可以使用本文档来查找有关 XL C/C++ 编译器独有的功能和功能部件的信息。

如何使用本文档

除非另有声明，否则此参考中的所有文本均适合 C 和 C++ 语言。在这些语言之间有差别时，将会通过符合约定标准的文本和图标来指示这些差别，如『本文档中使用的约定』中所述。

本文档中通篇使用 **xlc** 和 **xlc++** 编译器调用来描述编译器的操作。但是，如果特定环境需要，您也可以采用其他形式的编译器调用命令来代替，除非另有声明，否则编译器选项用法将保持不变。

虽然本文档涵盖有关配置编译器环境以及使用 XL C/C++ 编译器来编译和链接 C 或 C++ 应用程序的信息，但它不包含下列主题：

- 编译器安装：请参阅 *XL C/C++ Installation Guide*，以获得有关安装 XL C/C++ 的信息。
- 编译器选项：请参阅 *XL C/C++ Compiler Reference*，以获得有关编译器选项的语法和用法的详细信息。
- C 或 C++ 编程语言：请参阅 *XL C/C++ Language Reference*，以获得有关 C 或 C++ 编程语言的语法、语义和 IBM 实现的信息。
- 编程主题：请参阅 *XL C/C++ Programming Guide*，以获得有关使用 XL C/C++ 开发应用程序的详细信息，它们着重于程序的可移植性和优化。

本文档中使用的约定

印刷约定

下表说明了本文档中使用的印刷约定。

表 1. 印刷约定

字型	指示	示例
粗体	小写命令、可执行文件名称、编译器选项和伪指令。	如果指定 -O3 ，那么编译器将采用 -qhot=level=0 。要防止在指定 -O3 时出现全部 HOT 优化，您必须指定 -qnohot 。

表 1. 印刷约定 (续)

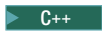
字型	指示	示例
斜体字	由用户提供实际名称或值的参数或变量。斜体字还可用于介绍新术语。	如果返回的值超过请求的 <i>size</i> ，那么确保更新 <i>size</i> 参数。
等宽字体	编程关键字和库函数、编译器内置函数、程序代码示例、命令字符串或用户定义的名称。	如果 <code>switch</code> 语句的一个或两个情况比其他条件执行特别地频繁，那么在执行 <code>switch</code> 语句之前结束这些条件并单独处理。

图标

本文档中描述的所有功能部件都同时适用于 C 和 C++ 语言。如果某个功能部件是一种语言所独有的，或者功能在语言之间有所不同，那么将使用以下图标：



该文本描述仅在 C 语言中才受支持的功能部件或描述特定于 C 语言的行为。



该文本描述仅在 C++ 语言中才受支持的功能部件或描述特定于 C++ 语言的行为。

语法图

在本文档中，自始至终都用图来举例说明 XL C/C++ 语法。本节将帮助您解释和使用这些图。

- 沿着线条的走向，从左至右、从上至下阅读语法图。

- ▶— 符号指示命令、伪指令或语句的开始。
- ▶ 符号指示命令、伪指令或语句语法在下一行继续。
- ▶— 符号指示命令、伪指令或语句是从上一行继续的。
- ▶◀ 符号指示命令、伪指令或语句的结束。

这些片段是语法单元图，不同于完整的命令、伪指令或语句，它们以 |— 符号开始，以 —| 符号结束。

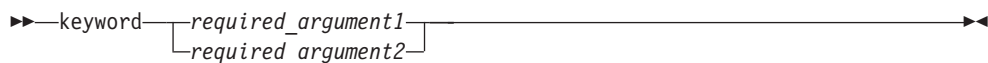
- 必需的项显示在水平线（主路径）上：



- 可选的项显示在主路径之下：



- 如果可以从两个或更多项中进行选择，那么将它们垂直堆叠在一起。
如果必须选择这些项的其中一项，那么堆叠中有一项显示在主路径上。



如果这些项是可选的，那么所有项都会显示在主路径下。



- 主线上向左折返的箭头（重复箭头）指示您可以从堆叠的项中选择多项或重复单个项。如果分隔符为非空白字符，那么还会指出该分隔符：



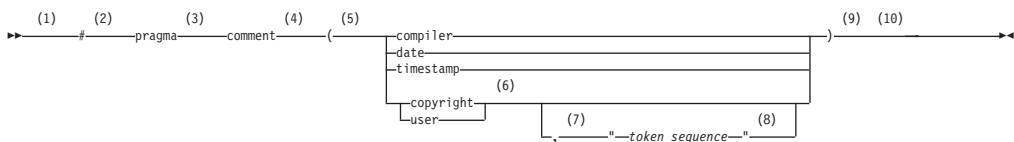
- 缺省项显示在主路径之上。



- 关键字以非斜体字母的形式显示，应严格按所显示的进行输入。
- 变量以斜体小写字母的形式显示。它们表示用户提供的名称或值。
- 如果显示了标点符号、圆括号、算术运算符或其他这样的符号，那么必须将它们作为语法的一部分输入。

样本语法图

以下语法图示例显示了 **#pragma comment** 伪指令的语法。



注:

- 这是语法图的开始。
- 符号 **#** 必须首先出现。
- 关键字 **pragma** 必须跟在 **#** 符号之后。
- 编译指示的名称 **comment** 必须跟在关键字 **pragma** 之后。
- 必须提供左圆括号。
- 输入的注释类型必须仅为以下所指示的类型之一：**compiler**、**date**、**timestamp**、**copyright** 或 **user**。
- 在注释类型 **copyright** 或 **user** 与可选字符串之间必须有一个逗号。
- 字符串必须跟在逗号之后。必须用双引号将字符串括起。
- 需要右圆括号。

10 这是语法图的结束。
按照以上显示的图表，**#pragma comment** 伪指令的下列示例在语法上是正确的：

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

示例

除非另有说明，否则本文档中的示例都是用简单样式进行编码，此样式不会尝试保存存储器、检查错误、获取快速性能或演示获取特定结果的所有可能方法。

相关信息

下列各节提供了有关涉及 XL C/C++ 的文档信息：

- 『IBM XL C/C++ 出版物』
- 第 ix 页的『其他 IBM 出版物』
- 第 ix 页的『其他出版物』

IBM XL C/C++ 出版物

XL C/C++ 提供以下形式的产品文档：

- 自述文件

自述文件包含最新的信息，其中包括对产品文档的更改和纠错。缺省情况下，自述文件位于 XL C/C++ 目录和安装 CD 的根目录中。

- 可安装的联机帮助页

为编译器调用和随产品提供的所有命令行实用程序提供了联机帮助页。*IBM XL C/C++ Advanced Edition for Linux V9.0 Installation Guide* 中提供了安装和访问联机帮助页的指示信息。

- 信息中心

可在网络上启动可搜索的 HTML 文件的信息中心，并可通过远程方式或本地方式访问该信息中心。*IBM XL C/C++ Advanced Edition for Linux V9.0 Installation Guide* 中提供了安装和访问联机信息中心的信息。此外，您还可以在 Web 上查看该信息中心，网址为 <http://publib.boulder.ibm.com/infocenter/lnxphelp/v9v111/index.jsp>。

- PDF 文档

缺省情况下，PDF 文档位于 `/opt/ibmcomp/vacpp/9.0/doc/LANG/pdf/` 目录，其中 *LANG* 是以下其中一项：`en_US`、`zh_CN` 或 `ja_JP`。您也可以在 Web 上获得该 PDF 文件，网址为 <http://www.ibm.com/software/awdtools/xlcpp/library>。

下列文件组成了一套完整的 XL C/C++ 产品手册：

表 2. XL C/C++ PDF 文件

文档标题	PDF 文件名	描述
<i>IBM XL C/C++ Advanced Edition for Linux V9.0 Installation Guide</i> , GC23-5893-00	install.pdf	包含有关安装 XL C/C++ 和配置环境以执行基本的编译和程序的信息。

表 2. XL C/C++ PDF 文件 (续)

文档标题	PDF 文件名	描述
<i>IBM XL C/C++ Advanced Edition for Linux V9.0 入门</i> , G151-0789-00	getstart.pdf	包含 XL C/C++ 产品的简介, 提供了有关设置和配置环境、编译和链接程序以及对编译错误进行故障诊断的信息。
<i>IBM XL C/C++ Advanced Edition for Linux V9.0 Compiler Reference</i> , SC23-5889-00	compiler.pdf	包含有关各种编译器选项、编译指示、宏、环境变量和内置函数 (包括那些用于并行处理的函数) 的信息。
<i>IBM XL C/C++ Advanced Edition for Linux V9.0 Language Reference</i> , SC23-5892-00	langref.pdf	包含有关 IBM 支持的 C 和 C++ 编程语言 (包括为了获取对非专有标准的可移植性和一致性的语言扩展) 的信息。
<i>IBM XL C/C++ Advanced Edition for Linux V9.0 Programming Guide</i> , SC23-5890-00	progguide.pdf	包含有关下列高级编程主题的信息, 例如应用程序移植、使用 Fortran 代码的语言间调用、库开发、应用程序优化和并行化以及 XL C/C++ 高性能库。

要阅读 PDF 文件, 请使用 Adobe® Reader。如果没有安装 Adobe Reader, 那么可以依据许可条款从 Adobe Web 站点下载该软件, 网址为 <http://www.adobe.com>。

与 XL C/C++ 相关的更多文档, 包括红皮书、白皮书、教程和其他文章, 都可以在 Web 上获得, 网址为:

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

其他 IBM 出版物

- *ESSL for Linux on POWER V4.2 Guide and Reference*, SA22-7904, 已在 <http://publib.boulder.ibm.com/infocenter/clresctr/index.jsp> 中提供

其他出版物

- *Using the GNU Compiler Collection*, 已在 <http://gcc.gnu.org/onlinedocs> 中提供

技术支持

可以从 XL C/C++ 支持页面获得其他技术支持, 网址为: <http://www.ibm.com/software/awdtools/xlcpp/support>。此页面提供一个具有搜索能力的门户网站, 可以在该网站找到大量的技术支持 FAQ 和其他支持文档。

如果找不到所需的内容, 那么可以发送电子邮件至 compinfo@ca.ibm.com。

有关 XL C/C++ 的最新信息, 请访问产品信息站点, 网址为: <http://www.ibm.com/software/awdtools/xlcpp>。

如何发送您的意见

您的反馈对于帮助提供准确和高质量的信息非常重要。如果您对本文档或任何其他 XL C/C++ 文档有任何意见, 请将您的意见通过电子邮件发送至 compinfo@ca.ibm.com。

请提供正确的信息，如文档的名称、文档的章节号、XL C/C++ 的版本以及您有意见的文本的特定位置，例如，页号或表号。

第 1 章 XL C/C++ 简介

IBM XL C/C++ Advanced Edition for Linux V9.0 是一种高性能的高级编译器，可以用于开发复杂且要进行大量计算的程序，包括使用 Fortran 程序进行的语言间调用。

本节在较高层次讨论了 XL C/C++ 编译器的功能部件。它面向要对该编译器进行评测的人员以及需了解关于该产品的更多信息的新用户。

与其他 IBM 编译器的通用性

XL C/C++ 与 XL C 和 XL Fortran 一起组成 XL 编译器系列。

IBM XL C/C++ Advanced Edition for Linux V9.0 是较大的 IBM C、C++ 和 Fortran 编译器系列的一部分。

这些编译器是从公共代码库派生的，该代码库在各种平台（如 AIX®、i5/OS®、选定的 Linux 分发版、z/OS® 和 z/VM® 操作系统）和编程语言之间共享编译器功能和优化技术。公共代码库除了符合国际编程语言标准之外，还有助于在多个操作系统和硬件平台上保持编译器性能的一致且使程序更加容易移植。

硬件和操作系统支持

XL C/C++ V9.0 支持 若干 Linux 分发版。请参阅自述文件和 *XL C/C++ Installation Guide* 中的“Before installing XL C/C++”，以获得必备条件的完整列表。

编译器、编译器的库及其生成的对象程序将在具有必需的软件和磁盘空间的 POWER3™、POWER4™、POWER5™、POWER5+™、POWER6™、PowerPC® 和 PowerPC 970 系统上运行。

要最大程度地利用各种受支持的硬件配置，编译器为性能调整应用程序（尤其是将用于执行已编译的应用程序的硬件类型）提供了一些选项。

高度可配置的编译器

XL C/C++ 为您提供了大量功能部件，以允许您根据自己的独特编译要求来定制编译器。

编译器调用命令

XL C/C++ 提供了几个可以用来调用编译器的不同命令，例如，**xlc**、**xlc++** 和 **xlc**。每个调用命令都是唯一的，它们指示编译器定制编译输出以遵循特定语言级别规范。提供编译器调用命令是为了支持所有标准化的 C/C++ 语言级别和许多流行的语言扩展。

编译器还提供了大多数调用命令的相应 “_r” 版本，例如，**xlc_r** 和 **xlc_r**。

“_r” 调用指示编译器将对象文件链接和绑定至线程安全组件与库，并为编译器创建的数据和过程生成线程安全对象代码。

有关 XL C/C++ 编译器调用命令的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的“Invoking the compiler”。

编译器选项

您可以从大量编译器选项中选择某个选项来控制编译器行为。不同类别的选项有助于您调试应用程序、优化和调整应用程序性能、选择语言级别和扩展以与其他 C 或 C++ 编译器支持的非标准功能部件和行为相兼容，以及执行许多其他常规任务。如果不执行这些任务，那么需要更改源代码。

XL C/C++ 使您可以通过环境变量、编译器配置文件、命令行选项和程序源文件中嵌入的编译器伪指令语句的组合来指定编译器选项。

有关 XL C/C++ 编译器选项的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的“Compiler options reference”。

定制编译器配置文件

安装过程会创建缺省的编译器配置文件，该文件包含定义了编译器选项缺省设置的节。

编译过程可能需要频繁地进行调用以指定除 XL C/C++ 提供的缺省设置之外的编译器选项设置。如果是这样，那么您可以使用 `makefile` 来定义编译器选项设置，也可以创建定制配置文件来定义自己的常用编译器选项设置集合。

有关更多信息，请参阅第 11 页的『使用定制编译器配置文件』。

语言标准一致性

编译器支持 C/C++ 的以下编程语言规范：

- ISO/IEC 9899:1999（称为 C99）
- ISO/IEC 9899:1990（称为 C89）
- ISO/IEC 14882:2003（称为标准 C++）
- ISO/IEC 14882:1998，C++ 语言的第一个正式规范（称为 C++98）

除了标准化语言级别以外，XL C/C++ 还支持语言扩展，包括：

- 用于支持可移植的并行化编程的 OpenMP V2.5 扩展
- 用于支持向量编程的语言扩展
- GNU C 和 C++ 语言扩展的子集

与 GNU 的兼容性

XL C/C++ 支持 GNU 编译器命令选项的子集以便于移植用 `gcc` 和 `g++` 开发的应用程序。

在将 `gxlc` 或 `gxlc++` 调用命令与选择的 GNU 编译器选项一起使用时，可获得此支持。在调用编译器之前，编译器尽可能地将 GNU 选项映射至它们的对等 XL C/C++ 编译器选项。

这些调用命令使用纯文本配置文件来控制 GNU 至 XL C/C++ 选项映射和缺省值。您可以定制此配置文件以更好地满足您可能有的任何独特的编译需求。请参阅 *XL C/C++ Compiler Reference* 中的“Reusing GNU C/C++ compiler options with `gxlc` and `gxlc++`”，以便获得更多信息。

XL C/C++ 将 GNU C 和 GNU C++ 头文件与 GNU C 和 C++ 运行时库一起使用，以生成与用 GNU 编译器集合（GCC）生成的代码是二进制兼容的代码。应用程序的一

些部分可用 XL C/C++ 构建并可将它们与用 GCC 构建的一些部分组合在一起生成应用程序，并且该应用程序能正常运行就好像它是单独用 GCC 构建的一样。

要获取与 GCC 编译的代码的二进制兼容性，用 XL C/C++ 编译的程序应包括与驻留在同一系统上的 GNU 编译器所使用的一样的头文件。要确保系统中存在头文件和运行时库的适当版本，在安装 XL C/C++ 之前，必须先安装必备的 GCC 编译器。

有关此关系的一些值得注意的其他方面是：

- IBM 内置函数与 GNU C 内置函数可以共存。
- C 和 C++ 程序的编译使用 GNU C 和 GNU C++ 头文件。
- 编译将 GNU 汇编程序用于汇编程序输入文件。
- 已编译的 C 代码将链接至 GNU C 运行时库。
- 已编译的 C++ 代码将链接至 GNU C 和 GNU C++ 运行时库。
- 调试使用随 IBM Cell Broadband Engine 软件开发工具包 2.0 提供的 GNU 调试器、**gdb**


源代码迁移和一致性检查

XL C/C++ 通过提供指示编译器将应用程序编译到特定语言级别的编译器调用命令，帮助保护您在现有 C/C++ 源代码中的投资。您也可以使用 **-qlanglvl** 编译器选项来指定一个给定的语言级别，如果程序源文件中的语言或语言扩展元素不符合该语言级别，那么编译器将发出警告、错误和严重错误消息。

请参阅 *XL C/C++ Compiler Reference* 中的 “**-qlanglvl**”，以获得更多信息。

库

XL C/C++ 附带提供了下列库：

- SMP 运行时库，它支持显式并行处理和自动并行处理。
- Mathematical Acceleration Subsystem (MASS) 库，它用于 32 位和 64 位方式的已调整数学内部函数。
- 已调整的代数函数的基本线性代数子系统 (BLAS) 库。
-  C++ 运行时库，它包含编译器所需要的支持例程。

Mathematical Acceleration Subsystem 库

Mathematical Acceleration Subsystem (MASS) 库由为了在受支持的处理器体系结构上获取最佳性能而专门进行调整的标量和向量数学内部函数组成。您可以选择 MASS 库来在各种各样的处理器上支持高性能计算，也可以选择为支持特定处理器系列而调整的库。

MASS 库函数支持 32 位和 64 位编译方式，它们是线程安全的并在缺省 libm 数学库例程上提供了改进的性能。当您请求对应用程序执行特定级别的优化时会自动调用这些库。您也可以显式地调用 MASS 库函数，而不必考虑优化选项是否生效。

请参阅 *XL C/C++ Programming Guide* 中的 “Using the Mathematical Acceleration Subsystem”，以便获得更多信息。

基本线性代数子程序

libxlopt 库中附带提供了高性能代数函数的基本线性代数子程序（BLAS）集合。这些功能使您能够执行以下操作：

- 计算一般矩阵或其转置矩阵的矩阵向量乘积。
- 对一般矩阵或其转置矩阵执行组合的矩阵乘法和加法。

有关使用 BLAS 函数的更多信息，请参阅 *XL C/C++ Programming Guide* 中的“Using the Basic Linear Algebra Subprograms”。

工具和实用程序

new_install

在安装 IBM XL C/C++ Advanced Edition for Linux V9.0 之后，运行此实用程序将会配置编译器以在系统上使用。

vac_configure

使用此实用程序来创建其他编译器配置文件，然后可以修改这些文件以包含您自己的编译器选项缺省设置的定制集合。

cleanpdf 命令

与概要定向反馈（PDF）相关的命令 **cleanpdf** 将从写入概要定向反馈数据的目录中除去所有概要分析信息。

mergepdf 命令

与概要定向反馈（PDF）相关的命令 **mergepdf** 在将两个或更多 PDF 记录合并为一个记录时，提供了权衡它们的重要性的能力。PDF 记录必须从相同的可执行文件派生。

resetpdf 命令

cleanpdf 命令的当前行为与 **resetpdf** 命令的行为相同，保留它是为了与其他平台上的较早发行版兼容。

showpdf 命令

showpdf 命令显示在运行概要定向反馈练习（使用选项 **-qpdf1** 和 **-qshowpdf** 进行的编译）时执行的所有过程的调用和块计数。

gxlc 和 gxlc++ 实用程序

gxlc 和 **gxlc++** 调用将 GNU C 或 GNU C++ 调用命令转换为相应的 **xlc** 或 **xlc++** 命令，然后再调用 IBM XL C/C++ Advanced Edition for Linux V9.0 编译器。这些实用程序的目的是减少对用 GNU 编译器构建的现有应用程序的 **makefile** 的更改，并使得向 IBM XL C/C++ Advanced Edition for Linux V9.0 的过渡更方便。

程序优化

XL C/C++ 提供了几个可以帮助您控制程序优化的编译器选项。使用这些选项，您可以执行以下操作：

- 选择不同级别的编译器优化
- 控制对循环、浮点和其他类型的操作的优化
- 根据程序将在何处运行，针对特定类别的机器或非常具体的机器配置优化程序

优化变换可以给予应用程序更好的整体执行性能。C/C++ 提供了一些针对各种受支持硬件的优化变换的组合。这些变换能够：

- 减少为关键操作执行的指令数。
- 重构生成的对象代码以最佳地使用 PowerPC 体系结构。
- 改进内存子系统的使用。
- 利用体系结构的能力来处理大量的共享内存并行化。

因为编译器支持完善的程序分析和变换，所以付出相对较小的开发努力就可能获得显著的性能提高。而且，XL C/C++ 启用了诸如 OpenMP 之类的编程模型，这使您可以编写高性能并行代码。

有关更多信息，请参阅：

- *XL C/C++ Programming Guide* 中的 “Optimizing your applications”
- *XL C/C++ Compiler Reference* 中的 “Optimization and tuning options”
- *XL C/C++ Compiler Reference* 中的 “Compiler built-in functions”

64 位对象能力

XL C/C++ 编译器的 64 位对象能力满足了用户对存储器容量需求和处理能力的不断增长的需要。Linux 操作系统提供一个环境，该环境允许您开发和执行通过使用 64 位地址空间来利用 64 位处理器的程序。

为了支持可以适合 64 位地址空间的更大的可执行文件，使用了单独的 64 位对象格式。链接程序绑定这些对象以创建 64 位的可执行文件。绑定在一起的对象必须具有相同的对象格式。以下方案是不允许的，将不能装入和/或执行：

- 从 32 位库或共享库中引用符号的 64 位对象或可执行文件
- 从 64 位库或共享库中引用符号的 32 位对象或可执行文件
- 尝试显式地装入 32 位模块的 64 位可执行文件
- 尝试显式地装入 64 位模块的 32 位可执行文件

XL C/C++ 主要通过使用 **-q64** 和 **-qarch** 编译器选项来支持 64 位方式。此组合确定目标体系结构的位方式和指令集。

有关更多信息，请参阅 *XL C/C++ Programming Guide* 中的 “Using 32-bit and 64-bit modes”。

共享内存并行化

XL C/C++ 支持多处理器系统体系结构的应用程序开发。您可以使用下列任何一种方法通过 XL C/C++ 来开发并行化应用程序：

- 基于伪指令的共享内存并行化
- 指示编译器自动生成共享内存并行化
- 基于共享或分布式内存并行化（MPI）的消息传递

有关更多信息，请参阅 *XL C/C++ Programming Guide* 中的 Parallelizing your programs。

OpenMP 伪指令

OpenMP 伪指令是一组基于 API 的命令，它们受 XL C/C++ 和许多其他 IBM 以及非 IBM C、C++ 和 Fortran 编译器的支持。

可以使用 OpenMP 伪指令来指示编译器如何并行化特定循环。源代码中存在这些伪指令可使得编译器无需对并行代码执行任何并行分析。OpenMP 伪指令要求存在 Pthread 库以提供用于并行化的必要基础结构。

OpenMP 伪指令解决并行化应用程序的三个重要问题：

1. 子句和伪指令可用于限定作用域的变量。通常，不应共享变量；即，每个处理器都应具有它自己的变量副本。
2. 工作共享伪指令指定应如何将包含在代码的并行区域中的工作分布在各个 SMP 处理器上。
3. 伪指令可用来控制处理器之间的同步。

XL C/C++ 支持 OpenMP API V2.5 规范。

有关更多信息，请参阅：

- *XL C/C++ Programming Guide* 中的 “Optimizing your applications”
- www.openmp.org

诊断列表

编译器输出列表可以提供重要的信息，以帮助您更有效地开发和调试应用程序。

列表信息已加到您可以包括或省略的可选节中。有关适用的编译器选项以及该列表本身的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的 “Compiler messages and listings”。

符号调试器支持

您可以指示 XL C/C++ 在已编译的对象中包含调试信息。该信息可以由 **gdb** 或任何其他符号调试器检查，以便帮助您调试程序。

第 2 章 IBM XL C/C++ Advanced Edition for Linux V9.0 的新增内容

本节描述了 IBM XL C/C++ Advanced Edition for Linux V9.0 中的新增功能部件和增强功能。

与 C/C++ 语言相关的更新

此发行版更改 C 编译的缺省语言级别，并且引进了在使用 long long 数据类型进行算术转换时的新行为。

为 C - extc99 更改的缺省语言级别

当使用 **xlc** 调用来调用 C 编译器时，缺省 **-qlanglvl** 编译器选项设置现在是 **extc99**。此更改将允许您使用 C99 功能部件和头文件，而不必明确地指定 **extc99** 子选项。

当使用新的缺省 **-qlanglvl=extc99** 设置进行编译时，您可能会遇到以下方面的问题：

- 在 C99 中可以使用 **restrict** 来限定指针，因此 **restrict** 不能用作标识。
- C99 中处理 **long long** 数据的方式与 C89 中处理 **long long** 数据的方式不同。
- C99 头文件定义了新宏：头文件 **limits.h** 中的 **LLONG_MAX** 和头文件 **stdarg.h** 中的 **va_copy**。
- 宏 **__STD_VERSION__** changes 的值从 199409 更改为 19990。

要还原至先前的 **xlc** 行为，请在调用编译器时指定 **-qlanglvl=extc89**。

使用 long long 数据类型进行算术转换

在使用 XL C/C++ V9.0 的这个发行版的情况下，当您使用 long long 数据类型执行某些算术运算时，编译器行为会更改。

采用算术表达式。其中：

- 一个操作数具有 long long int 类型或 long long 类型，以及
- 其他操作数具有 unsigned long int 类型，但它的值在 long long int 或 long long 中无法表示。

XL C/C++ 的前发行版将这两类操作数转换为 long long 类型。

自本发行版起，编译器会将这两类操作数转换为 unsigned long long int 类型或 unsigned long long 类型。此新行为与 GCC 编译器行为一致。

有关更多信息，请参阅 *XL C/C++ Language Reference* 中的 Integral and floating-point promotions。

体系结构和处理器支持

-qarch 和 **-qtune** 编译器选项控制由编译器生成的代码。这些编译器选项调整指令、调度和其他优化，以便为指定的目标处理器或一系列处理器提供最佳性能。

-qtune 的新缺省设置

新的缺省 **-qtune** 设置为:

- **-qtune=balanced**

-qtune=balanced 子选项是此发行版的新增选项, 当您指定某些 **-qarch** 设置时, 该子选项将成为缺省的 **-qtune** 设置。使用 **-qtune=balanced** 会指示编译器调整生成的代码, 以便在一系列新型处理器体系结构上获得最佳性能, 这些体系结构包括 POWER6。

对 POWER6 处理器的新支持

XL C/C++ 9.0 扩展 **-qarch** 和 **-qtune** 子选项的列表, 以支持新近提供的 POWER6 处理器。

现在提供了下列 **-qarch** 和 **-qtune** 选项:

- **-qarch=pwr6**
- **-qarch=pwr6e**
- **-qtune=pwr6**

-qipa 编译器选项还添加新的体系结构克隆子选项, 以支持对 POWER6 处理器的过程间分析 (IPA) 优化:

- **-qipa=clonearch=pwr6**

性能和优化

许多新的功能部件和增强功能都可归入性能和优化调整类别。

与性能相关的编译器选项和伪指令

下表中的条目描述新增或已更改的编译器选项和伪指令。

此处提供的信息只是一个简短的概述。有关这些编译器选项以及其他与性能相关的编译器选项的更多信息, 请参阅 *XL C/C++ Compiler Reference* 中的 Optimization and tuning options。

表 3. 与性能相关的编译器选项和伪指令

选项/伪指令	描述
-qalias= <u>global</u> <u>noglobal</u>	这些新的 -qalias 子选项启用或禁用链接时优化期间在编译单元上应用特定于语言的别名判别规则。
-qalias= <u>restrict</u> <u>norestrict</u>	这些新的 -qalias 子选项对严格限定指针启用或禁用优化。通常, 指定 -qalias=restrict 将会提高使用严格限定指针的代码的性能。您可以使用 -qalias=norestrict , 以保留与使用 V9.0 以前版本的编译器所编译代码的兼容性。
-qnofdpr1-qfdpr	指定 -qfdpr 选项指示编译器将优化信息存储在所创建的对象文件中。反馈定向程序重构 (FDPR) 性能调整实用程序将使用此信息。
-qfloat= <u>fenv</u> <u>nofenv</u>	这些新的 -qfloat 子选项会向编译器发出通知, 告知代码是否对浮点硬件环境具有依赖性, 例如以明示方式读取或写入浮点状态或控制寄存器。指定 -qfloat=nofenv 将指示代码不依赖硬件环境, 从而允许编译器执行主动优化。

表 3. 与性能相关的编译器选项和伪指令 (续)

选项/伪指令	描述
-qfloat= gcclongdouble nogcclongdouble	这些新的 -qfloat 子选项仅在 -qldbl128 选项生效时才有效。它们指示编译器使用 GCC 提供的或 IBM 提供的库函数来进行 128 位长双精度运算。
-qfloat= hscmplx nohscmplx	指定 -qfloat=hscmplx 将进一步优化那些涉及复杂除法和复杂绝对值的运算。
-qfloat= rngchk norngchk	指定 -qfloat=rngchk 将为软件除法运算和内联 sqrt 运算启用对输入自变量的范围检查。指定 -qfloat=norngchk 将指示编译器跳过范围检查，从而在某些环境中可以获得更佳性能。指定 -qnostrict 编译器选项设置 -qfloat=norngchk 。
-qipa=clonearch=pwr6	-qipa=clonearch 编译器选项现在包括新的 pwr6 子选项，以支持对 POWER6 处理器的过程间分析 (IPA) 优化。
-qipa=threads= [auto noauto number]	此新的 -qipa 子选项允许您指定在第二个 IPA 遍数期间编译器可以将多少线程分配给代码生成。
-qnoldbl128 qldbl128	指定 -qldbl128 会增大从 64 位到 128 位的 long double 型大小。
-qpdf	-qpdf 选项现在可以用于提供有关特定对象的概要定向反馈。有关更多信息，请参阅 <i>XL C/C++ Programming Guide</i> 中的 “Object level profile-directed feedback”。
-qsmp= threshold=n	当 -qsmp=auto 生效时，此新的子选项允许您在编译器考虑将循环中所需的工作量进行自动并行化处理之前，指定该工作量。
#pragma expected_value(param, value)	使用 #pragma expected_value 伪指令，指定函数调用中传递的参数在运行时期间很可能采用的值。编译器可以使用此信息来执行某些优化，例如，函数克隆和内联。

本发行版新增的内置函数

本节列示了此发行版新增的内置函数。有关由 XL C/C++ 提供的内置函数的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的 “Compiler built-in functions”。

转换函数

这些新函数将 long double 数据类型从 IBM 样式转换为 GCC 样式。

- long double __ibm2gccldbl (long double);
- _Complex long double __ibm2gccldbl_cmplx (_Complex long double);

PowerPC 高速缓存控制

PowerPC 体系结构指定 **dcbst** 和 **dcbf** 高速缓存复制指令。下列新的内置函数使程序员可以直接访问这些指令。

- void __dcbst(const void* addr); /* Data Cache Block Store */
- void __dcbf(const void* addr); /* Data Cache Block Flush */

POWER6 预取扩展和高速缓存控制

POWER6 处理器具有高速缓存控制和流预取扩展功能，且支持存储流预取和预取深度控制。XL C/C++ 提供下列新的内置函数，使程序员可以直接访问这些指令。

- `void __dcbfl(const void* addr);` /* pwr6 - Data Cache Block Flush from L1 data cache only */
- `void __protected_unlimited_stream_set(unsigned int direction, const void* addr, unsigned int ID);` /* Supported by pwr5 and pwr6 */
- `void __protected_unlimited_store_stream_set(unsigned int direction, const void* addr, unsigned int ID);` /* Supported by pwr6 */
- `void __protected_store_stream_set(unsigned int direction, const void* addr, unsigned int ID);` /* Supported by pwr6 */
- `void __protected_stream_count_depth(unsigned int unit_cnt, unsigned int prefetch_depth, unsigned int ID);` /* Supported by pwr6 */

其他新增或已更改的编译器选项

可以在命令行上指定编译器选项，也可以通过应用程序源文件中嵌入的伪指令来指定编译器选项。请参阅 *XL C/C++ Compiler Reference*，以获得这些编译器选项和其他编译器选项的详细描述和用法信息。

表 4. 其他新增或已更改的编译器选项

选项/伪指令	描述
-C!	指定 -C! 编译器选项可从预处理的输出中除去注释。
-qcommon -qnoccommon	在 -qcommon 生效时，未初始化的全局变量将在对象文件的公共部分中进行分配。在 -qnoccommon 生效时，未初始化的全局变量将初始化为零，并且在对象文件的数据部分中进行分配。
-qoptdebug -qnooptdebug	当与 -O3 或更高的优化级别配合使用时，新的 -qoptdebug 选项将会指示编译器生成可由符号调试器读取的优化伪码。
-qpack_semantic= ibm gnu	-qpack_semantic 选项是一个可移植性选项，它指示编译器对 #pragma pack 伪指令使用 IBM 或 GCC 语法和语义。
-qreport	当与启用自动并行化或向量化的编译器选项配合使用时， -qreport 选项现在可生成伪码列表，显示如何使程序循环并行化和向量化。如果编译器无法将给定循环并行化或向量化，那么该报告还提供诊断信息。
-qsaveopt -qnosaveopt	在前发行版中， -qsaveopt 选项可将用于编译文件的命令行选项存储到生成的对象文件中。在此发行版中，已扩展存储在对象文件中的信息，以便同时包含编译期间调用的每个编译器组件的版本和级别信息。
-qsmp=stackcheck	此新的 -qsmp 子选项将指示编译器在运行时期间检查从属线程是否存在堆栈溢出，如果剩余堆栈大小低于 XLSMPOPTS 环境变量的 stackckeck 选项指定的字节数，那么发出警告。
-qtemplatedepth=number	-qtemplatedepth 指定将由编译器处理且已递归地实例化的最大模板特例化数。
-qversion=verbose	-qversion 选项添加了新的 verbose 子选项。指定 -qversion=verbose 将指示编译器显示编译期间调用的每个编译器组件的版本和级别信息。

第 3 章 设置和定制 XL C/C++

有关完整的先决条件和安装信息，请参阅 *XL C/C++ Installation Guide*。

使用定制编译器配置文件

在 XL C/C++ 编译器安装期间创建了缺省编译器配置文件，您可以直接修改此配置文件，以便根据特定需要添加缺省选项。但是，如果稍后对该编译器应用更新，那么还需要将所有修改重新应用于新安装的配置文件。

通过创建自己的定制编译器配置文件，可以避免此情况。现在，编译器可以识别和解析您在定制配置文件以及缺省配置文件中指定的编译器设置。

如果您指示编译器使用定制配置文件，那么编译器将检查和处理该定制配置文件中的设置，然后才查看缺省系统配置文件中的设置。以后可能会影响缺省配置文件中设置的编译器更新将不会影响定制配置文件中的设置。

请参阅 *XL C/C++ Compiler Reference* 中的 “Using custom compiler configuration files”，以获得更多信息。

确定安装了哪种级别的 XL C/C++

如果联系软件支持人员以寻求帮助，那么将需要知道特定机器上安装的 XL C/C++ 的级别。

要显示系统上已安装的编译器的版本和发行版级别，使用 **-qversion** 编译器选项调用编译器。

例如，要获取详细的版本信息，请在命令行中输入下列内容：

```
xlc++ -qversion=verbose
```

第 4 章 使用 XL C/C++ 开发应用程序

基本的 C/C++ 应用程序开发就是编辑、编译和链接（缺省情况下与编译合并为单个步骤）以及运行过程的重复循环。

注意:

1. 在可以使用编译器之前，必须首先确保正确地安装和配置 XL C/C++。有关更多信息，请参阅 *XL C/C++ Installation Guide*。
2. 要了解有关编写 C/C++ 程序的信息，请参阅 *XL C/C++ Language Reference*。

编译器阶段

典型的编译器调用按顺序执行下列某些或全部活动。为了获得链接时优化，将在编译期间多次执行某些活动。当每个程序运行时，结果按顺序发送至下一个步骤。

1. 预处理源文件
2. 编译，根据所指定的编译器选项，它可能包含下列阶段：
 - a. 前端语法分析和语义分析
 - b. 高级优化
 - c. 低级优化
 - d. 寄存器分配
 - e. 最终的汇编
3. 在进行预处理后的 **.s** 文件和 **.S** 文件的程序汇编
4. 对象链接以创建可执行的应用程序

要查看编译器如何单步执行这些阶段，当您编译应用程序时，请指定 **-v** 编译器选项。要查看编译器在每个阶段所花费的时间量，请指定 **-qphsinfo**。

编辑 C/C++ 源文件

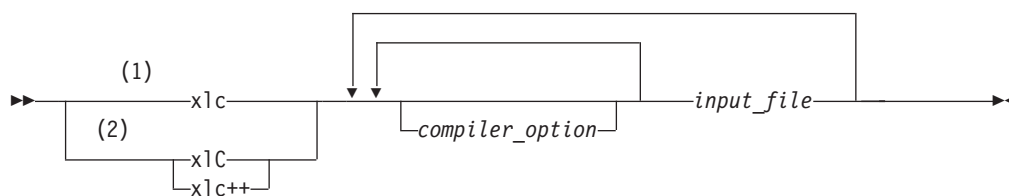
要创建 C/C++ 源程序，您可以使用任何系统可用的文本编辑器。源程序必须使用被认可的文件名后缀来保存。请参阅第 16 页的『XL C/C++ 输入和输出文件』，以获得 XL C/C++ 认可的文件名后缀列表。

要使 C 或 C++ 源程序成为有效的程序，它必须符合在 *XL C/C++ Language Reference* 中指定的语言定义。

使用 XL C/C++ 进行编译

调用编译器

要编译源程序，请使用以下所示的基本调用语法:



注:

- 1 用于编译 C 源代码的基本调用。
- 2 用于编译 C++ 源代码的基本调用。

编译器调用命令执行所有必要的步骤来编译 C 或 C++ 源文件、汇编任何 **.s** 和 **.S** 文件并将对象文件和库链接至可执行程序。

为了新的 C 或 C++ 应用程序能够工作，您应该使用 **xlc**、**xlc++**、或线程安全等效项进行编译。**xlc** 和 **xlc++** 将编译 C 或 C++ 程序源代码，但因为当 C 编译器调用链接程序时未指定 C++ 代码所需要的库，所以使用 **xlc** 编译 C++ 文件可能导致链接或运行时错误。

另外，还提供了其他调用命令以满足专门的编译需要，目的主要是为 C 或 C++ 语言的不同级别和扩展提供显式编译支持。请参阅 *XL C/C++ Compiler Reference* 中的 “Invoking the Compiler”，以获取有关可用的编译器调用命令（包括旨在帮助开发者从 GNU 编译环境迁移至 XL C/C++ 的特殊调用）的更多信息。

编译并行化的 XL C/C++ 应用程序

XL C/C++ 提供线程安全编译器调用命令，您可以在编译用于多处理器环境中的并行化应用程序时使用这些调用命令。这些调用类似于其相应的基本编译器调用，只是它们会将已编译的对象链接和绑定至线程安全组件和库。

普通的 XL C/C++ 线程安全编译器调用包括:

- **xlc_r**
- **xlc++_r**
- **xlc_r**

XL C/C++ 提供其他线程安全调用来满足特定的编译需求。请参阅 *XL C/C++ Compiler Reference* 中的 “Invoking the Compiler”，以便获得更多信息。

注: 单独使用这些命令中的任何一个并不表示并行化。为了让编译器能够识别 OpenMP 伪指令并激活并行化，您还必须指定 **-qsmp** 编译器选项。而您仅应该将 **-qsmp** 选项与这些线程安全调用命令之一一起指定。当您指定 **-qsmp** 时，驱动程序链接将指向配置文件的活动中 **smp** 库行上所指定的库。

指定编译器选项

编译器选项执行各种功能，如设置编译器特征、描述要生成的对象代码、控制发出的诊断消息和执行某些预处理器功能。

您可以用以下方式指定编译器选项:

- 在命令行上使用命令行编译器选项

- 在源代码中使用伪指令语句
- 在 `makefile` 中
- 在编译器配置文件内找到的节中
- 或通过使用这些技术的任何组合

当指定了多个编译器选项时，有可能发生选项冲突和不兼容。为了以一致的方式解决这些冲突，编译器对大多数选项通常应用以下一般优先级顺序：

1. 源文件中的伪指令语句覆盖命令行设置
2. 命令行编译器选项设置覆盖配置文件设置
3. 配置文件设置覆盖缺省设置

通常，如果调用编译器时在命令行上多次指定了相同的编译器选项，那么最后指定的那个选项起作用。

注：某些编译器选项并不按照以上所述的优先级顺序。

例如，`-I` 编译器选项是一种特殊情况。编译器在搜索命令行上用 `-I` 指定的目录之前，首先搜索在 `vac.cfg` 文件中用 `-I` 指定的任何目录。该选项是累积的而不是优先的。

请参阅 *XL C/C++ Compiler Reference*，以获取有关编译器选项及其用法的更多信息。

具有累积行为的其他选项是 `-R` 和 `-l`（小写 L）。

还可以将编译器选项传递至链接程序、汇编程序和预处理器。请参阅 *XL C/C++ Compiler Reference* 中的“Compiler options reference”，以获取有关编译器选项和如何指定它们的更多信息。

通过 `gxlc` 和 `gxlc++` 复用 GNU C/C++ 编译器选项

XL C/C++ 包括各种帮助您从 GNU C/C++ 编译器转换到 XL C/C++ 的功能部件。`gxlc` 和 `gxlc++` 实用程序就是其中之一。

每个 `gxlc` 和 `gxlc++` 实用程序都接受 GNU C 或 C++ 编译器选项并将它们转换成相应的 XL C/C++ 选项。两个实用程序都使用 XL C/C++ 选项来创建 `xlc` 或 `xlc++` 调用命令，然后实用程序使用该命令来调用编译器。提供这些实用程序是为了帮助您复用为先前用 GNU C/C++ 开发的应用程序创建的 `makefile`。然而，为了充分利用 XL C/C++ 的能力，应使用 XL C/C++ 调用命令及其相关选项。

`gxlc` 和 `gxlc++` 的操作由配置文件 `gxlc.cfg` 控制。此文件中显示具有 XL C/C++ 对应选项的 GNU C/C++ 选项。并非每个 GNU 选项都具有对应的 XL C/C++ 选项。`gxlc` 和 `gxlc++` 会对未转换的输入选项返回一个警告。

`gxlc` 和 `gxlc++` 选项映射是可修改的。有关使用 `gxlc` 或 `gxlc++` 配置文件的信息，请参阅 *XL C/C++ Compiler Reference* 中的 *Reusing GNU C/C++ compiler options with gxlc and gxlc++*。

XL C/C++ 输入和输出文件

XL C/C++ 可识别下面列示的文件类型。有关这些文件类型和编译器使用的其他文件类型的详细信息，请参阅 *XL C/C++ Compiler Reference* 中的“Types of input files”和“Types of output files”。

表 5. 输入文件类型

文件扩展名	描述
.c	C 源文件
.C、.cc、.cp、.cpp、.cxx 和 .c++	C++ 源文件
.i	预处理的源文件
.o	对象文件
.s	汇编程序文件
.S	未预处理的汇编程序文件
.so	共享对象或库文件

表 6. 输出文件类型

文件扩展名	描述
a.out	用于编译器创建的可执行文件的缺省名称
.d	生成依赖性文件
.i	预处理的源文件
.lst	列表文件
.o	对象文件
.s	汇编程序文件
.so	共享对象或库文件

将已编译的应用程序与 XL C/C++ 链接在一起

缺省情况下，不需要执行任何特殊操作来链接 XL C/C++ 程序。编译器调用命令自动调用链接程序来生成可执行输出文件。例如，运行以下命令：

```
xlc++ file1.C file2.o file3.C
```

编译并生成对象文件 file1.o 和 file3.o，然后将所有对象文件（包括 file2.o）提交至链接程序以生成一个可执行文件。

分步进行编译和链接

要生成稍后可以链接的对象文件，使用 **-c** 选项。

```
xlc++ -c file1.C           # Produce one object file (file1.o)
xlc++ -c file2.C file3.C   # Or multiple object files (file1.o, file3.o)
xlc++ file1.o file2.o file3.o # Link object files with default libraries
```

动态和静态链接

XL C/C++ 允许程序利用操作系统设施进行动态和静态链接：

- 动态链接意味着在程序首次运行时找到并装入某些外部例程的代码。当编译使用共享库的程序时，缺省情况下，共享库自动链接至程序。

如果多个程序使用共享库中的例程，那么动态链接的程序占用较少的磁盘空间和虚拟内存。在链接期间，它们不需要任何特殊的预防措施来避免与库例程的命名冲突。如果几个程序同时使用相同的共享例程，那么它们的性能可能超过静态链接的程序。它们还允许您升级共享库中的例程而不需要重新链接。

因为该链接形式是缺省值，所以您不需要其他选项来启用该链接。

- 静态链接意味着由程序调用的所有例程的代码成为可执行文件的一部分。

可以将静态链接的程序移至系统并在系统上运行它们，而不需要 XL C/C++ 运行时库。如果它们执行许多对库例程的调用或调用许多小例程，那么它们的性能可能超过动态链接的程序。如果您要避免与库例程产生命名冲突，那么在为程序中的数据对象和例程选择名称时它们确实需要一些预防措施。如果分别在不同级别的操作系统编译和运行它们，那么它们也可能不生效。

有关编译和链接程序的更多信息，请参阅：

- *XL C/C++ Compiler Reference* 中的 “Linking”
- *XL C/C++ Programming Guide* 中的 “Constructing a library”

运行已编译的应用程序

由 XL C/C++ 编译器生成的程序可执行文件的缺省文件名是 **a.out**。可以使用 **-o** 编译器选项来选择不同的名称。

要运行程序，请在命令行上输入程序可执行文件的名称和任何运行时自变量。

因为您可能偶尔会执行错误的命令，所以应避免对程序可执行文件指定与系统命令或 shell 命令相同的名称，如 **test** 或 **cp**。如果您决定使用与系统命令或 shell 命令相同的名称来命名程序可执行文件，那么应该通过指定程序可执行文件所在目录的路径名（如 **./test**）来执行程序。

取消执行

要暂挂正在运行的程序，当程序处于前台时按 **Ctrl+Z** 键。使用 **fg** 命令来恢复运行。

要取消正在运行的程序，当程序处于前台时按 **Ctrl+C** 键。

设置运行时选项

您可以使用环境变量设置来控制某些运行时选项和使用 XL C/C++ 编译器创建的应用程序的行为。其他环境变量不控制实际的运行时行为，但可以影响应用程序的运行方式。

有关环境变量和它们在运行时如何影响应用程序的更多信息，请参阅 *XL C/C++ Installation Guide*。

在其他系统上运行已编译的应用程序

如果要在没有安装编译器的另一个系统上运行用 XL C/C++ 编译器开发的应用程序，那么将需要在该系统上安装运行时环境。

您可以从以下网址处的 XL C/C++ Support 页面获取最新的 XL C/C++ 运行时环境 PTF 映像以及许可证发放和用法信息：

www.ibm.com/software/awdtools/xlcpp/support

XL C/C++ 编译器诊断辅助

当编译应用程序遇到问题时，XL C/C++ 会发出诊断消息。您可以使用这些消息和编译器输出列表中提供的其他信息来帮助识别和更正这类问题。

当您运行应用程序时，XL C/C++ 运行时还将为某些不受支持的操作（特别是与 I/O 有关的操作）发出消息。

有关可帮助解决应用程序问题的列表、诊断和相关编译器选项的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的下列主题：

- “Compiler messages and listings”
- “Error checking and debugging options”
- “Listings, messages, and compiler information options”

调试已编译的应用程序

在编译时指定 **-g** 或 **-qlinedebug** 编译器选项将指示 XL C/C++ 编译器把调试信息包括在已编译的输出中。

然后，可以使用 **gdb** 或任何其他符号调试器，来单步执行和检查已编译的应用程序的行为。

已优化的应用程序在调试时，将会给您带来一些特殊的挑战。当调试高度优化的应用程序时，您应该考虑使用 **-qoptdebug** 编译器选项。有关调试的更多信息，请参阅 *XL C/C++ Programming Guide* 中的 “Optimizing your applications”。

声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务信息，请向您当地 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

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

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

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

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：INTERNATIONAL BUSINESS MACHINES CORPORATION “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证，因此本条款可能不适用于您。

本信息中可能包含有技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。该 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue

Markham, Ontario L6G 1C7
Canada

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能全面地作举例说明，这些示例包含个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。为了开发、使用、营销或分发符合此操作平台（编写样本程序的操作平台）的应用程序编程接口的应用程序，您可以用任何方式复制、修改和分发这些样本程序，而不必向 IBM 公司付款。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明：

©（贵公司的名称）（年）。此部分代码是根据 IBM 公司样本程序衍生出来的。© Copyright IBM Corp. 1998, 2007. All rights reserved.

商标和服务标记

文本中标识的公司、产品或服务名称可能是 IBM 或其他公司的商标或服务标记。有关 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标的信息位于 <http://www.ibm.com/legal/copytrade.shtml> 中。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

Intel 是 Intel Corporation 或其附属机构在美国和其他国家或地区的商标或注册商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的注册商标。

其他公司、产品和服务名称可能是其他公司的商标或服务标记。

业界标准

支持下列标准:

- C 语言符合信息系统编程语言 C 的国际标准 (ISO/IEC 9899-1990)。
- 此外, C 语言还符合信息系统编程语言 C 的国际标准 (ISO/IEC 9899-1999 (E))。
- C++ 语言符合信息系统编程语言 C++ 的国际标准 (ISO/IEC 14882:1998)。
- 此外, C++ 语言还符合信息系统编程语言 C++ 的国际标准 (ISO/IEC 14882:2003 (E))。
- C 语言和 C++ 语言均符合 OpenMP C 和 C++ 应用程序编程接口版本 2.5 。

索引

[B]

- 编辑源文件 13
- 编译
 - 活动顺序 13
 - SMP 程序 14
- 编译器
 - 调用 13
 - 控制行为 14
 - 运行 13
- 编译器选项
 - 冲突和不兼容 15
 - 规范方法 14
- 并行化 5

[C]

- 程序
 - 运行 17

[D]

- 代码优化 4
- 调试 18
- 调试器支持 18
 - 符号 6
 - 输出列表 18
- 调试信息, 生成 18
- 调试已编译的应用程序 18
- 调用编译器 13
- 调用程序 17
- 定制
 - 与 GNU 兼容 2
- 动态链接 16
- 对象文件 16
 - 创建 16
 - 链接 16
- 多处理器系统 5

[F]

- 符号调试器支持 6

[G]

- 工具 4
 - 配置文件实用程序 4
 - 新安装配置实用程序 4
 - cleanpdf 实用程序 4
 - gxlc 和 gxlc++ 实用程序 4

- 工具 (续)
 - mergepdf 实用程序 4
 - new_install 实用程序 4
 - resetpdf 实用程序 4
 - showpdf 实用程序 4
 - xlc_configure 4
- 共享对象文件 16
- 共享内存并行化 5
- 归档文件 16

[H]

- 汇编程序
 - 源 (.S) 文件 16
 - 源 (.s) 文件 16

[J]

- 静态链接 16

[K]

- 可执行文件 16
- 库 16

[L]

- 链接
 - 动态 16
 - 静态 16
- 链接过程 16
- 列表 16

[Q]

- 迁移
 - 源代码 15

[S]

- 实用程序 4
 - cleanpdf 4
 - gxlc 和 gxlc++ 4
 - mergepdf 4
 - new_install 4
 - resetpdf 4
 - showpdf 4
 - xlc_configure 4
- 输出文件 16

- 输入文件 16

[W]

- 文件
 - 编辑源代码 13
 - 输出 16
 - 输入 16
- 问题确定 18

[X]

- 性能
 - 优化变换 5

[Y]

- 优化
 - 程序 4
- 语言支持 2
- 源级别调试支持 6
- 源文件 16
- 运行编译器 13
- 运行时
 - 库 16
- 运行时环境 17
- 运行时选项 17

[Z]

- 执行编译器 13
- 执行程序 17
- 执行链接程序 16

[数字]

- 64 位环境 5

G

- GNU
 - 兼容性 2

M

- mod 文件 16

O

OpenMP 6

S

SMP

程序, 编译 14

SMP 程序 5

V

vac.cfg 文件 15

X

XL C/C++ 的级别, 确定 11

xlc_configure 4

[特别字符]

.a 文件 16

.c 和 .C 文件 16

.i 文件 16

.lst 文件 16

.mod 文件 16

.o 文件 16

.S 文件 16

.s 文件 16

.so 文件 16



程序号: 5724-S73

G151-0789-00

