

IBM XL C/C++ for Linux V10.1



XL C/C++ 入门

版本 10.1

IBM XL C/C++ for Linux V10.1



XL C/C++ 入门

版本 10.1

注意

在使用本资料及其支持的产品之前，请参阅第 27 页的『声明』中的信息。

第一版本

本版本适用于 IBM XL C/C++ for Linux V10.1（程序号 5724-U83）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 1996, 2008. All rights reserved.

目录

关于本文档	v
约定	v
相关信息	viii
IBM XL C/C++ 信息	viii
标准和规范	ix
其他 IBM 信息	x
其他信息	x
技术支持	x
如何发送您的意见	x
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
工具和实用程序	4
程序优化	5
64 位对象能力	5
共享内存并行化	6
诊断列表	6
符号调试器支持	7
第 2 章 IBM XL C/C++ for Linux V10.1	
 的新增内容	9
操作系统支持	9
C++0x	9
与其他 XL C/C++ 语言相关的更新	11
OpenMP 3.0	11

性能和优化	11
新增或已更改的编译器选项和伪指令	12
版本 9.0 中新增的增强功能	13
与 C/C++ 语言相关的更新	13
体系结构和处理器支持	14
性能和优化	14
其他新增或已更改的编译器选项	16

第 3 章 设置和定制 XL C/C++	19
使用定制编译器配置文件	19

第 4 章 使用 XL C/C++ 开发应用程序	21
编译器阶段	21
编辑 C/C++ 源文件	21
使用 XL C/C++ 进行编译	21
调用编译器	22
编译并行化 XL C/C++ 应用程序	22
指定编译器选项	23
XL C/C++ 输入和输出文件	24
将已编译的应用程序与 XL C/C++ 链接在一起	24
动态和静态链接	25
运行已编译的应用程序	25
XL C/C++ 编译器诊断辅助	26
调试已编译的应用程序	26
确定安装哪种级别的 XL C/C++	26

声明	27
商标和服务标记	29

索引	31
---------------------	-----------

关于本文档

本文档包含 IBM® XL C/C++ for Linux® V10.1 编译器的概述和基本用法信息。

应该阅读本文档的人

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++ 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『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* 以获取有关 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++ Optimization and 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。

约定

印刷约定

下表说明了 IBM XL C/C++ for Linux V10.1 信息中使用的印刷约定。

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。	编译器提供基本的调用命令 xlc 和 xlc (xlc++) 以及其他一些编译器调用命令，以支持各种 C/C++ 语言级别和编译环境。
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。







表 1. 印刷约定 (续)

字型	指示	示例
下划线	编译器选项或伪指令的参数的缺省设置。	nomaf <u>maf</u>
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。	要编译和优化 myprogram.c, 请输入: xlc myprogram.c -O3。

限定元素 (图标)

本信息中描述的大多数功能部件都同时适用于 C 和 C++ 语言。在其中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的语言元素描述中, 此信息将使用图标以描绘文本段, 如下所示:

表 2. 限定元素

限定符/图标	含义
仅限于 C, 或仅限于 C 开始   仅限于 C 结束	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。
仅限于 C++, 或仅限于 C++ 开始   仅限于 C++ 结束	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。
IBM 扩展开始   IBM 扩展结束	该文本描述一个功能部件, 该功能部件是标准语言规范的 IBM 扩展。

语法图

在本信息中, 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。

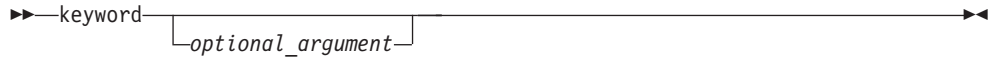
—> 符号指示命令、伪指令或语句的结束。

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：



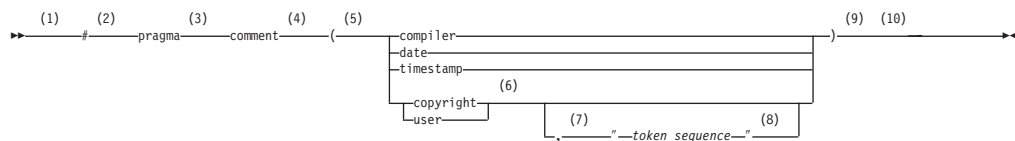
- 缺省项显示在主路径之上。



- 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:

- 1 这是语法图的开始。
- 2 符号 # 必须首先出现。
- 3 关键字 pragma 必须跟在 # 符号之后。
- 4 编译指示的名称 comment 必须跟在关键字 pragma 之后。
- 5 必须提供左圆括号。
- 6 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。
- 7 在注释类型 copyright 或 user 与可选字符串之间必须有一个逗号。
- 8 字符串必须跟在逗号之后。必须用双引号将字符串括起。
- 9 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")
```

本信息中的示例

除非另有说明, 否则本信息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。

安装信息的示例已标注为示例或基本示例。基本示例旨在描述基本或缺省安装期间执行的过程, 它们只需要进行极小的修改或甚至不需要修改。

相关信息

下列各节提供了 XL C/C++ 的相关信息:

IBM XL C/C++ 信息

XL C/C++ 提供以下形式的产品信息:

- 自述文件

自述文件包含最新的信息, 其中包括对产品信息的更改和纠错。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++ for Linux V10.1 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++ for Linux V10.1 Installation Guide* 中提供了安装和访问联机信息中心的指示信息。

可在 Web 上查看的信息中心，网址为 <http://publib.boulder.ibm.com/infocenter/lnxphelp/v101v121/index.jsp>。

• PDF 文档

缺省情况下，PDF 文档位于 `/opt/ibmcmp/vacpp/10.1/doc/LANG/pdf/` 目录，其中 *LANG* 是以下其中一项：`en_US`、`zh_CN` 或 `ja_JP`。另外，Web 上也提供了 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++ 产品信息：

表 3. XL C/C++ PDF 文件

文档标题	PDF 文件名	描述
<i>IBM XL C/C++ for Linux V10.1 Installation Guide</i> , GC23-8553-00	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。
《 <i>IBM XL C/C++ for Linux V10.1 入门</i> 》，G151-1048-00	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++ for Linux V10.1 Compiler Reference</i> , SC23-8549-00	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++ for Linux V10.1 Language Reference</i> , SC23-8552-00	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++ for Linux V10.1 Optimization and Programming Guide</i> , SC23-8550-00	proguide.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++ 相关的更多信息，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>

标准和规范

XL C/C++ 设计用于支持下列标准和规范。您可以参阅这些标准，以获得在本信息中找到的某些功能部件的精确定义。

- *Information Technology – Programming languages – C, ISO/IEC 9899:1990*，也称为 C89。
- *Information Technology – Programming languages – C, ISO/IEC 9899:1999*，也称为 C99。

- *Information Technology – Programming languages – C++, ISO/IEC 14882:1998*, 也称为 C++98。
- *Information Technology – Programming languages – C++, ISO/IEC 14882:2003(E)*, 也称为标准 C++。
- *Information Technology – Programming languages – Extensions for the programming language C to support new character data types, ISO/IEC DTR 19769*. C 标准委员会已经接受此草稿技术报告, 您可以在以下网站上获得该报告: <http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1040.pdf>。
- *Draft Technical Report on C++ Library Extensions, ISO/IEC DTR 19768*. 此草稿技术报告已经提交给 C++ 标准委员会, 您可以在以下网站上获得该报告: <http://www.open-std.org/JTC1/SC22/WG21/docs/papers/2005/n1836.pdf>。
- *Altivec Technology Programming Interface Manual*, Motorola Inc. 用于支持向量处理技术的向量数据类型的规范可在以下网站上获得: http://www.freescale.com/files/32bit/doc/ref_manual/ALTIVECPIM.pdf。

其他 IBM 信息

- 在 <http://publib.boulder.ibm.com/clresctr/windows/public/esslbooks.html> 处提供的 *ESSL for AIX V4.3 - ESSL for Linux on POWER V4.3 Guide and Reference (SA22-7904)*

其他信息

- 在 <http://gcc.gnu.org/onlinedocs> 处提供的 *Using the GNU Compiler Collection*

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说明和其他支持信息。

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。如果您对本资料或任何其他 XL C/C++ 资料有任何意见, 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, 如资料的名称、资料的章节号、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++ for Linux V10.1 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

IBM XL C/C++ for Linux V10.1 是较大的 IBM C、C++ 和 Fortran 编译器系列的一部分。

XL C/C++ 与 XL Fortran 一起组成 XL 编译器系列。

这些编译器是从共享用于大量平台和编程语言的编译器功能和优化技术的公共代码库派生的。编程环境包括 IBM AIX®、IBM Blue Gene/L™、IBM Blue Gene/P™、Cell Broadband Engine™ 体系结构、IBMi、选定的 Linux 分发版、IBM z/OS® 和 IBM z/VM®。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

IBM XL C/C++ for Linux V10.1 支持多个 Linux 分发版。有关完整的需求列表，请参阅自述文件和 *XL C/C++ Installation Guide* 中的“Before installing XL C/C++”。

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

您可以使用各种编译器调用命令和选项来定制编译器以满足您的独特编译需求。

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 19 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 (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 V3.0 的某些功能扩展
- 用于支持向量编程的语言扩展
- GNU C 和 C++ 语言扩展的子集

有关 C/C++ 语言规范和扩展的更多信息，请参阅 *XL C/C++ Language Reference* 中的“Language levels and language extensions”。

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++` 编译器开发的应用程序。

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++”，以获取更多信息。

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。

有关此关系的其他一些值得注意的方面是：

- 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 GNU 调试器、随 IBM Software Developer Kit (SDK) for Multicore Acceleration V3.0 提供的 **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代码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。

有关更多信息，请参阅 *XL C/C++ Compiler Reference* 中的“qlanglvl”。

库

XL C/C++ 包括了包含多个库的运行时环境。

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。

有关更多信息，请参阅 *XL C/C++ Optimization and Programming Guide* 中的“Using the Mathematical Acceleration Subsystem”。

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++ Optimization and Programming Guide* 中的“Using the Basic Linear Algebra Subprograms”。

其他库

随 XL C/C++ 附带提供了下列各项:

- SMP 运行时库, 它支持显式并行处理和自动并行处理。请参阅 *XL C/C++ Optimization and Programming Guide* 中的“SMP Runtime Library”。
- C++ 运行时库包含编译器所需要的支持例程。

对 Boost 库的支持

除了 XL C/C++ 附带提供的库之外, 该产品的此版本还支持 Boost V1.34.1 库。提供了用于修改 Boost 1.34.1 库的补丁文件, 以便构建这些库并将其与 XL C/C++ 应用程序配合使用。补丁或修改文件不会扩展 Boost 库, 也不会向其提供其他功能。

要下载补丁文件和了解有关对这些库的支持的更多信息, 请参阅 XL C/C++ 库页面 (网址: <http://www.ibm.com/software/awdtools/xlcpp/library>) 上的相关链接。

工具和实用程序

XL C/C++ 附带提供了多种工具和实用程序。

new_install

在安装 IBM XL C/C++ for Linux V10.1 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, 以包含您自己定制的编译器选项缺省设置集合。

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, 它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 命令显示在概要定向反馈练习运行（使用选项 **-qpdf1** 和 **-qshowpdf** 进行的编译）中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++ for Linux V10.1 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++ for Linux V10.1 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或性能的编译器选项。

使用这些选项，您可以执行下列任务：

- 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。这些变换提供下列好处：

- 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

有关更多信息，请参阅以下相关主题：

- *XL C/C++ Optimization and Programming Guide* 中的“Optimizing your applications”
- *XL C/C++ Compiler Reference* 中的“Optimizing 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++ Optimization and 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++ Optimization and 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处理器上。
3. 伪指令可用来控制处理器之间的同步。

自此发行版起，XL C/C++ 支持 OpenMP API V3.0 规范。请参阅第 11 页的『OpenMP 3.0』，以了解此功能部件带来的各项更改的概述。

有关程序性能优化的更多信息，请参阅：

- *XL C/C++ Optimization and 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++ for Linux V10.1 的新增内容

本节描述了 IBM XL C/C++ for Linux V10.1 中的新增功能部件和增强功能。

操作系统支持

IBM XL C/C++ for Linux V10.1 现在支持 IBM Power Systems 服务器支持的下列这些操作系统: Red Hat Enterprise Linux 5.2 (RHEL 5.2) 和 SUSE Linux Enterprise Server 10 SP 2 (SLES10 SP2)。

预定义的宏

现在有 4 个新的宏:

`__ILP32 __ILP32__`

仅在编译用于其中 long int、int 和指针均使用 32 位的目标时才定义为 1。否则, 不会对其进行定义。

`__LP64 __LP64__`

仅在编译用于其中 long int 和指针均使用 64 位且 int 使用 32 位的目标时才定义为 1。否则, 不会对其进行定义。

编译器不再支持 `__C99_COMPLEX_HEADER__` 宏。

有关 XL C/C++ 的预定义宏的完整列表, 请参阅 *XL C/C++ Compiler Reference* 中的“Compiler predefined macros”。

C++0x

此发行版引进了对 C++ 编程语言 (特别是 C++0x) 的新版本标准的支持。此标准尚未被正式采用, 但我们开始支持其某些功能。

具体地说, 在此发行版中:

- 我们添加了新的语言级别
- 我们为与 long long 数据类型的算术转换引进了新的整数提升规则
- C++ 预处理器现在支持 C99 功能

新的语言级别 - **extended0x**

当调用 C++ 编译器时, 缺省 `-qlanglvl` 编译器选项仍是 **extended**。

在此发行版中, 新的子选项已添加至 `-qlanglvl` 选项。`-qlanglvl=extended0x` 用于允许用户对 C++0x 的、当前受 XL C/C++ 支持的任何功能部件的早期实现进行试验。

C++ 环境下的 C99 long long

在使用 XL C/C++ V10.1 的此发行版的情况下, 当使用整数数据类型执行某些算术运算时, 编译器行为会改变。具体地说, 整数提升规则已更改。

先前，在 C++（并且作为对 C89 的扩展）中，当使用 **-qlonglong** 进行编译时，无后缀整数将提升为它所适合的此列表中的第一个类型：

```
int
long int
unsigned long int
long long int
unsigned long long
```

从此发行版开始，当使用 **-qlanglvl=extended0x** 进行编译时，编译器立即将无后缀整数提升为它所适合的此列表中的第一个类型：

```
int
long int
long long int
unsigned long long
```

注：与 C99 标准在 C 编译器中的实现相似，如果值不符合 long long 类型，但符合 unsigned long long，那么 C++ 将允许从 long long 提升至 unsigned long long。在此情况下，将生成一条消息。

已添加宏 `__C99_LLONG`，以便与 C99 兼容。在 **-qlanglvl=extended0x** 的情况下，此宏定义为 1；在其他情况下，则取消定义。

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

预处理器更改

对 C++ 预处理器所作的下列更改使其更容易将代码从 C 移植至 C++：

- 常规字符串文字现在可与宽字符串文字并置。
- `#line <integer>` 预处理器伪指令具有较大的上限。对于 C++，它已经从 32767 增大至 2147483647。
- C++ 现在支持 `_Pragma` 运算符。
- 这些宏现在适用于 C++ 和 C：
 - `__C99_MACRO_WITH_VA_ARGS`（也可与 **-qlanglvl=extended** 配合使用）
 - `__C99_MAX_LINE_NUMBER`（也可与 **-qlanglvl=extended** 配合使用）
 - `__C99_PRAGMA_OPERATOR`
 - `__C99_MIXED_STRING_CONCAT`

注：除另有说明之外，否则，仅当使用 **-qlanglvl=extended0x** 进行编译时，才提供这些 C++ 预处理器更改。

有关 XL C/C++ 支持的语言标准的其他信息，请参阅 *XL C/C++ Language Reference* 中的“Language levels and extensions”。

与其他 XL C/C++ 语言相关的更新

向量数据类型

向量数据类型现在可以使用一些能与基本数据类型配合使用的运算符，例如：

- 一元运算符
- 二目运算符
- 关系运算符

OpenMP 3.0

在此发行版中，XL C/C++ 支持 OpenMP API V3.0 规范。XL C/C++ 实现基于 IBM 对 OpenMP Application Program Interface Draft 3.0 Public Comment 的解释。

版本 2.5 与版本 3.0 之间的主要差别是：

- 增加任务级别并行化。新的 OpenMP 构造 TASK 和 TASKWAIT 使用户能够对非常规算法（例如，现有 OpenMP 构造无法处理的指针跟踪或递归算法）进行并行化。
- for 循环现在可以包含 unsigned int 和 pointer 类型以及 signed int 的 var 值。
- 堆栈大小控制。现在可以使用新的环境变量 OMP_STACKSIZE 控制由 OMP 运行时库所创建线程的堆栈大小。
- 用户可以使用新的环境变量 OMP_WAIT_POLICY 与 OMP_SET_POLICY 向所需的等待线程的行为提供暗示。
- 存储器复用。已除去对 PRIVATE 子句的一些限制。出现在并行构造的 reduction 子句中的列表项现在也可以出现在工作共享构造上 private 子句中。
- 安排。新的 SCHEDULE 属性 auto 允许编译器和运行时系统控制安排。
- 具有 STATIC 时间表的连续循环构造现在可以使用 nowait。
- 嵌套支持 - COLLAPSE 子句已经被添加至 DO、FOR、PARALLELL FOR 和 PARALLEL DO 伪指令以允许精确循环嵌套的并行化。这意味着可以使嵌套中的多个循环并行化。
- THREADPRIVATE 伪指令现在可适用于类作用域以及文件和块作用域的变量。
- 规范格式的迭代器循环（其中包括具有随机存取迭代器的循环）的并行化。

有关更多信息，请参阅：

- XL C/C++ Optimization and Programming Guide 中的“Using OpenMP directives”
- www.openmp.org

性能和优化

某些功能部件和增强功能可以帮助调整和优化应用程序性能。

对 -qstrict 的增强

许多子选项已经被添加至 -qstrict 选项，以允许对违反严格程序语义的优化和变换进行更细颗粒度控制。在前发行版中，-qstrict 选项禁止所有违反严格程序语义的变换。如果您使用不带子选项的 -qstrict，那么它仍然会执行该行为。相似地，在前发行版中，-qnostrict 已允许那些可更改程序语义的变换。由于较高级别的优化可能需要放松严格程序语义，所以添加子选项使您可以放松所选规则，以便获得特定的较快代码的好处，而无须关闭所有语义验证。

有 16 个新子选项可单独地或通过子选项组使用。这些组是:

all 禁用所有语义更改变换，其中包括由其他子选项控制的那些变换。

ieeeftp 控制单独的操作是否符合 IEEE 754 语义。

order 控制是否可采用或许违反程序语言语义的方式对单个操作重新排序。

precision
控制可能影响程序结果精度的优化和变换。

exceptions
控制可能影响程序生成的运行时异常的优化和变换。

有关这些子选项的详细信息，请参阅 *XL C/C++ Compiler Reference* 中的“-qstrict”。

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详细信息，请参阅 *XL C/C++ Compiler Reference* 中的“Optimization and tuning options”。

表 4. 与性能相关的编译器选项和伪指令

选项/伪指令	描述
-qstrict	已经添加许多新的子选项，以便您能够对程序语义规则的放松进行更多控制，从而获得一些性能益处。
-qfloat	一些 -qfloat 子选项受 -qstrict 的新子选项影响。
-qreport	列表现在包含有关为每个循环创建多少个流以及哪些循环由于 non-stride-one 引用而无法 SIMD 矢量化信息。您可以使用此信息来提高应用程序的性能。
-qsmp	当 -qsmp=omp 生效时，OpenMP API 3.0 的其他功能现在可用。有关更多信息，请参阅第 11 页的『OpenMP 3.0』。

有关性能调整和程序优化的其他信息，请参阅 *XL C/C++ Optimization and Programming Guide* 中的“Optimizing your applications”。

新增或已更改的编译器选项和伪指令

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 5. 新增或已更改的编译器选项和伪指令

选项/伪指令	描述
-qstrict	许多子选项已经被添加至 -qstrict 选项，以允许对违反严格程序语义的优化和变换进行更多控制。有关更多信息，请参阅第 11 页的『性能和优化』。

表 5. 新增或已更改的编译器选项和伪指令 (续)

选项/伪指令	描述
-qshowmacros	当与 -E 选项一起使用时, -qshowmacros 选项会用宏定义替换预处理的输出。提供了子选项用于更准确地控制预定义宏和用户定义的宏的发出。
-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 选项现在报告循环中的流数量, 并且在循环由于 non-stride-one 引用而无法 SIMD 向量化时生成信息。
-qsmp	当 -qsmp=omp 生效时, OpenMP API 3.0 的其他功能现在可用。有关更多信息, 请参阅第 11 页的『OpenMP 3.0』。
-qtimestamps	此选项可用于从已生成的二元中除去时间戳记。
-qtls	线程本地存储器支持已经增强为包括 <code>__attribute__((tls-model("string")))</code> , 其中 <i>string</i> 是 local-exec、initial-exec、local-dynamic 或 global-dynamic。
-qinfo	子选项 als 和 noals 已添加至 qinfo 选项以报告 (或不报告) ANSI 别名判别规则的可能违例。

版本 9.0 中新增的增强功能

本节描述了新增功能部件和对先前版本 9.0 中编译器的增强。

与 C/C++ 语言相关的更新

C 编译的缺省语言级别已更改, 并且引入了在使用 long long 数据类型进行算术转换时的新行为。

为 C 更改的缺省语言级别 – extc99

当使用 **xlc** 调用来调用 C 编译器时, 缺省 **-qlanglvl** 编译器选项设置是 **extc99**。此更改允许您使用 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 __STDC_VERSION__ 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++ 版本 9.0 的情况下, 当使用 long long 数据类型执行某些算术运算时, 编译器行为会改变。

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

- 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++ 的前发行版已将这两类操作数转换为 `long long` 类型。

编译器现在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++ 版本 9.0 已扩展 **-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

已经进行多处增强以帮助调整性能和优化程序。

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 6. 与性能相关的编译器选项和伪指令

选项/伪指令	描述
-qalias= global noglobal	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应用程序的别名判别规则。
-qalias= restrict norestrict	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。
-qnofdprl-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= fenv nofenv	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和控制寄存器。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。
-qfloat= gcc longdouble nogcc longdouble	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++ Optimization and 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。

版本 9.0 中的内置函数

一些内置函数已添加在版本 9.0。

有关 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 7. 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 -qnocommon	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 -qnocommon 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 -qnoptdebug	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。

表 7. 其他新增或已更改的编译器选项 (续)

选项/伪指令	描述
-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。
-qsaveoptl-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。在版本 9.0 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 stackcheck 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* 中的“Before installing”。

使用定制编译器配置文件

您可以通过修改缺省配置文件或创建自己的配置文件来定制编译器设置和选项。

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。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”，以获取更多信息。

第 4 章 使用 XL C/C++ 开发应用程序

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。

注意事项:

1. 必须首先确保正确地安装和配置 XL C/C++ 才能使用编译器。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. 在预处理组合件（.s）文件和未预处理的汇编程序（.S）文件之后对它们进行汇编
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，请在编译应用程序时指定 **-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 24 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。

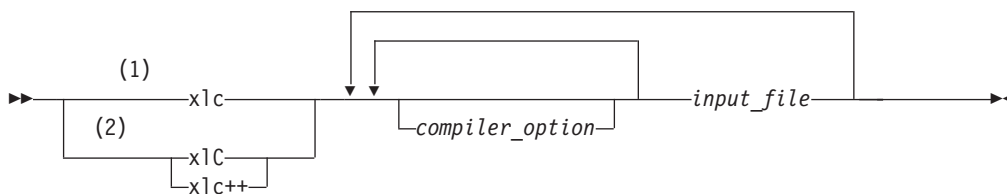
使用 XL C/C++ 进行编译

XL C/C++ 是命令行编译器。您可以根据特定 C/C++ 应用程序的需要选择调用命令和选项。

调用编译器

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。

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。

为了新的应用程序能够工作，您应该使用 **xlc**、**xlc++**、或线程安全等效项进行编译。您可以使用 **xlc++** 来编译 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（其中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 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”，以获取更多信息。

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。

有关并行化应用程序的更多信息，请参阅 *XL C/C++ Optimization and Programming Guide* 中的“Parallelizing your programs”。

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。有关编译器选项和如何指定它们的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的“Compiler options reference”。

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，其中包括 **gxlc** 和 **gxlc++**。

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当的 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。

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

XL C/C++ 输入和输出文件

XL C/C++ 可识别这些文件类型。

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 *XL C/C++ Compiler Reference* 中的“Types of output files”。

表 8. 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 9. 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 file1.C 和 file3.C 以生成对象文件 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
```

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++ Optimization and Programming Guide* 中的“Constructing a library”

动态和静态链接

XL C/C++ 允许程序利用操作系统工具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将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。

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** 编译器选项来选择另一名称。

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，那么应该通过指定该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，控制某些运行时选项以及使用 XL C/C++ 编译器所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，那么将需要在此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。

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

您可以使用符号调试器来调试采用 XL C/C++ 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。有关调试选项的更多信息，请参阅 *XL C/C++ Compiler Reference* 中的“Error checking and debugging”。

然后，可以使用 **gdb** 或其他任何符号调试器，以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。有关优化代码的更多信息，请参阅 *XL C/C++ Optimization and Programming Guide* 中的“Optimizing your applications”。

确定安装哪种级别的 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
```

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 公司的样本程序衍生出来的。© Copyright IBM Corp. 1998, 2008. All rights reserved.

商标和服务标记

IBM、IBM 徽标和 `ibm.com` 是 International Business Machines Corporation 在美国和/或其他国家或地区的商标或注册商标。在此信息中，首次出现时这些 IBM 商标项和其他 IBM 商标项均标记有相应符号（® 或 ™），指示此信息发布时 IBM 拥有的美国注册的商标或普通法商标。此类商标在其他国家或地区也可能是已注册商标或普通法商标。Web 上提供有 IBM 商标的当前完整列表，网址为 <http://www.ibm.com/legal/copytrade.shtml>。

Adobe、Adobe 徽标、PostScript 和 PostScript 徽标是 Adobe Systems Incorporated 在美国和/或其他国家或地区的注册商标或商标。

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

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

Cell Broadband Engine 是 Sony Computer Entertainment, Inc. 在美国和/或其他国家或地区的商标，并且依据许可证使用。

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

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[B]

- 编辑源文件 21
- 编译
 - 活动顺序 21
 - SMP 程序 22
- 编译器
 - 调用 22
 - 控制行为 23
 - 运行 22
- 编译器选项
 - 冲突和不兼容 23
 - 规范方法 23
- 并行化 6, 11

[C]

- 程序
 - 运行 25

[D]

- 代码优化 5
- 调试 26
- 调试器支持 26
 - 符号 7
 - 输出列表 26
- 调试信息, 生成 26
- 调试已编译的应用程序 26
- 调用编译器 22
- 调用程序 25
- 调用命令 22
- 定制
 - 与 GNU 兼容 2
- 动态链接 25
- 对象文件 24
 - 创建 24
 - 链接 24
- 多处理器系统 6, 11

[F]

- 符号调试器支持 7

[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

- 工具 (续)
 - gxlc 和 gxlc++ 实用程序 5
 - mergepdf 实用程序 4
 - new_install 实用程序 4
 - resetpdf 实用程序 4
 - showpdf 实用程序 5
 - xlconfig 4
- 共享对象文件 24
- 共享内存并行化 6, 11
- 归档文件 24

[H]

- 汇编程序
 - 源 (.S) 文件 24
 - 源 (.s) 文件 24

[J]

- 基本示例, 描述的 viii
- 静态链接 25

[K]

- 可执行文件 24
- 库 24

[L]

- 链接
 - 动态 25
 - 静态 25
- 链接过程 24
- 列表 24

[N]

- 内置函数 15

[Q]

- 迁移
 - 源代码 23

[S]

- 实用程序 4
 - cleanpdf 4
 - gxlc 和 gxlc++ 5

- 实用程序 (续)
 - mergepdf 4
 - new_install 4
 - resetpdf 4
 - showpdf 5
 - xlconfig 4
- 输出文件 24
- 输入文件 24

[W]

- 文件
 - 编辑源代码 21
 - 输出 24
 - 输入 24
- 问题确定 26

[X]

- 性能
 - 优化变换 5

[Y]

- 优化
 - 程序 5
- 语言标准 2
- 语言支持 2
- 源级别调试支持 7
- 源文件 24
- 运行编译器 22
- 运行时
 - 库 24
- 运行时环境 26
- 运行时选项 25

[Z]

- 执行程序 25
- 执行链接程序 24

[数字]

- 64 位环境 5

G

- GNU
 - 兼容性 2

M

mod 文件 24

O

OMP 伪指令 11

OpenMP 6

S

SMP

程序, 编译 22

SMP 程序 6

V

vac.cfg 文件 23

X

XL C/C++ 的级别, 确定 26

xl_c_configure 4

[特别字符]

.a 文件 24

.c 和 .C 文件 24

.i 文件 24

.lst 文件 24

.mod 文件 24

.o 文件 24

.S 文件 24

.s 文件 24

.so 文件 24



程序号: 5724-U83

中国印刷

G151-1048-00

