

IBM XL C/C++ Advanced Edition V8.0 for
Linux



XL C/C++ 入门

IBM XL C/C++ Advanced Edition V8.0 for
Linux



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 29 页的『声明』中的一般信息。

第一版（2005 年 9 月）

本版本适用于 IBM® XL C/C++ Advanced Edition V8.0 for Linux™（程序 5724-M16）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。

IBM 欢迎您提出宝贵意见。您可以通过电子的形式将意见发送到下面列出的网络标识。如果需要答复，请务必提供您的完整网络地址。

- 互联网: ctscrcf@cn.ibm.com

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 可以按它认为适当的任何方式使用或分发此信息，而无须对您承担任何责任。

© Copyright International Business Machines Corporation 1990, 2005. All rights reserved.

目录

关于本文档	v
应该阅读本文档的人	v
如何使用本文档	v
本文档的结构	v
本文档中使用的约定和术语	vi
印刷约定	vi
如何阅读语法图	vi
示例	viii
相关信息	viii
IBM XL C/C++ 出版物	viii
其它文档	ix
技术支持	ix
如何发送您的意见	ix
第 1 章 XL C/C++ 功能部件的概述	1
与其它 XL 编译器的通用性	1
文档、联机帮助和技术支持	1
硬件和操作系统支持	1
高度可配置的编译器	2
语言标准一致性	2
与 GNU 的兼容性	3
源代码迁移和一致性检查	3
库	3
Mathematics 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	5
64 位对象能力	5
共享内存并行化	6
OpenMP 伪指令	6
诊断列表	6
符号调试器支持	6
第 2 章 V8.0 中的新增内容	7
性能和优化	7
特定于体系结构和处理器的代码调整	7
高性能库	7
其它与性能相关的编译器选项和伪指令	7
本发行版新增的内置函数	8
语言增强和 API 的支持	10
C、C++ 和 Fortran 的 OpenMP API V2.5 支持	10
易于使用	10
新安装和配置实用程序	10
对 IBM Tivoli License Manager 的支持	11
新的编译器选项	11
新的命令行选项	11

新增的 pragma 伪指令	12
第 3 章 设置和定制 XL C/C++	13
环境变量和 XL C/C++	13
设置编译器工作环境	13
设置缺省运行时选项	13
定制配置文件	14
确定安装哪种级别的 XL C/C++	14
第 4 章 使用 XL C/C++ 编辑、编译和链接程序	15
编译器阶段	15
编辑 C 和 C++ 源文件	15
使用 XL C/C++ 进行编译	15
编译并行化的 XL C/C++ 应用程序	17
XL C/C++ 输入文件	17
XL C/C++ 输出文件	18
指定编译器选项	19
链接 XL C/C++ 程序	19
在单独的步骤中编译和链接	20
动态和静态链接	20
第 5 章 运行 XL C/C++ 程序	21
取消执行	21
设置运行时选项	21
在其它系统上运行已编译的应用程序	21
第 6 章 XL C/C++ 编译器诊断辅助	23
编译返回码	23
XL C/C++ 编译器列表	23
头部节	24
选项节	24
源代码节	24
变换报告节	26
属性和交叉引用节	26
目标节	26
文件表节	27
编译单元收尾节	27
编译收尾节	27
调试已编译的应用程序	27
声明	29
编程接口信息	30
商标和服务标记	30
索引	33

关于本文档

《XL C/C++ 入门》对 XL C/C++ 编译器、该编译器较重要的功能部件以及这些功能部件如何才能帮助您改进软件开发效率进行了一般概述。

为了升级至本发行版的当前 XL C/C++ 用户的利益，《XL C/C++ 入门》还包括了 V8.0 的新增的或改进的功能部件的总结。

《XL C/C++ 入门》仅为了帮助您熟悉编译器。有关使用 XL C/C++ 编译器的详细信息，您将需要参阅第 viii 页的『IBM XL C/C++ 出版物』中描述的 XL C/C++ Advanced Edition V8.0 for Linux 库中的其它书籍。

应该阅读本文档的人

《XL C/C++ 入门》面向任何计划使用 IBM® XL C/C++ Advanced Edition V8.0 for Linux、熟悉 Linux® 操作系统且以前具有一些 C 和 C++ 编程经验的人。

如何使用本文档

如果您不熟悉 XL C/C++，则应查看第 1 页的第 1 章，『XL C/C++ 功能部件的概述』以熟悉 XL C/C++ 的关键功能和如何开始使用它来开发应用程序。

如果您已经是一个有经验的 XL C/C++ 用户且现在要升级至 XL C/C++ 的最新发行版，则您可能想要直接转至第 7 页的第 2 章，『V8.0 中的新增内容』以查看编译器的最新更改和功能部件增强。

本指南的剩余部分提供了使用 XL C/C++ 执行基本程序开发任务的简短概述。

本文档的结构

本指南包括以下主题：

- 第 1 页的第 1 章，『XL C/C++ 功能部件的概述』概述了 XL C/C++ 编译器的关键功能部件
- 第 7 页的第 2 章，『V8.0 中的新增内容』描述了最新版本的 XL C/C++ 所提供的新的和已更新的功能部件。
- 第 13 页的第 3 章，『设置和定制 XL C/C++』提供了有关在设置和定制 XL C/C++ 时所涉及的步骤的简短概述信息以及您可以在何处找到更详细信息的指示。
- 第 15 页的第 4 章，『使用 XL C/C++ 编辑、编译和链接程序』讨论了使用 XL C/C++ 创建和编译应用程序所涉及的基本步骤。
- 第 21 页的第 5 章，『运行 XL C/C++ 程序』描述了如何运行已编译的应用程序，包括如何设置运行时选项。
- 第 23 页的第 6 章，『XL C/C++ 编译器诊断辅助』提供了有关如何使用 XL C/C++ 编译器诊断辅助来标识和改正应用程序的编译问题的指南。

本文档中使用的约定和术语

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	命令、可执行文件名称、 <code>pragma</code> 伪指令和编译器选项。	使用 -qmkshrobj 编译器选项以从生成的目标文件创建共享目标。
斜体字	其实际名称或值将由用户提供的参数或变量。斜体字还用于介绍新的术语。	如果返回的值超过请求的 <i>size</i> ，则确保更新 <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、文件和目录名称、程序代码示例、命令字符串或用户定义的名称。	如果您使用未初始化的锁定变量调用 <code>omp_destroy_lock</code> ，则调用结果是未定义的。

如何阅读语法图

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那些图。

必须将标点符号、圆括号、算术运算符和其它特殊字符作为语法的一部分输入。

- 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、伪指令或语句结束。

语法单元图不同于完整的命令、伪指令和语句，它以 ▶—— 符号开始，以 ——▶ 符号结束。

- 必需的项出现在水平线上（主路径）。

▶▶—keyword—*required_item*——▶▶

- 可选的项显示在主路径之下。

▶▶—keyword—*optional_item*——▶▶

- 如果可以从两个或更多项中进行选择，则将它们垂直堆叠在一起。

如果必须选择这些项的其中一项，则堆叠中有一项显示在主路径上。

▶▶—keyword—*required_choice1*
 required_choice2——▶▶

如果选择这些项目之一是可选的，则所有项都堆叠在主路径之下。



缺省项显示在主路径之上。



- 主线上向左边折返的箭头指示可以重复的项。



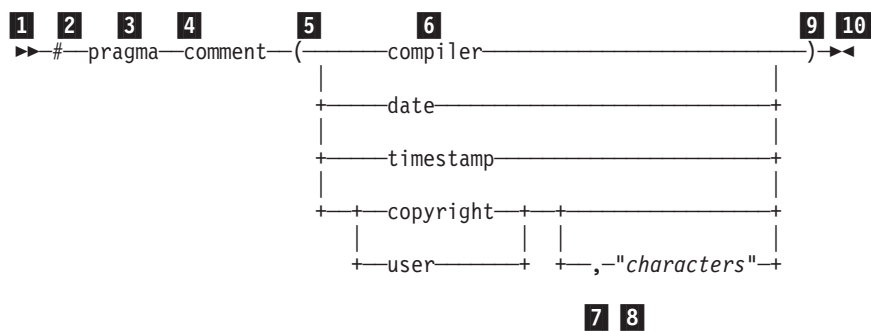
堆叠上方的重复箭头指示您可以从堆叠的项中选择多项或重复单个选项。

- 关键字以非斜体字母的形式显示，应严格按所显示的输入（例如：extern）。

变量以斜体小写字母的形式显示（例如：identifier）。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、算术运算符或其它这样的符号，则必须将它们作为语法的一部分输入。

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



- 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 左圆括号必须存在。
- 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")
```

示例

本文档中的示例都是用简单样式进行编码（除非另有说明），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的方法。

相关信息

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 `/opt/ibmcmp/vacpp/8.0/` 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 V8.0 for Linux Installation Guide* 中提供了有关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，并可以远程方式或本地方式访问该信息中心。*IBM XL C/C++ Advanced Edition V8.0 for Linux Installation Guide* 中提供了有关安装和访问信息中心的信息。也可以在 Web 上的以下网址处查看信息中心:

```
publib.boulder.ibm.com/infocenter/lxpcmp/index.jsp
```

- PDF 文档

缺省情况下，PDF 文档位于 `/opt/ibmcmp/vacpp/8.0/doc/language/PDF/` 目录中，也可以在 Web 上的以下网址处获得它们:

```
www.ibm.com/software/awdtools/xlcpp/library
```

本文档加上以下文件，它们组成了一套完整的 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 V8.0 for Linux Installation Guide</i> , GC09-8017-00	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++ Advanced Edition V8.0 for Linux Compiler Reference</i> , SC09-8013-00	compiler.pdf	包含关于各种编译器选项、编译指示、宏、环境变量以及内置函数（包括那些用于并行处理的函数）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 V8.0 for Linux Language Reference</i> , SC09-8016-00	language.pdf	包含关于 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 V8.0 for Linux Programming Guide</i> , SC09-8014-00	proguide.pdf	包含关于高级编程主题（如应用程序移植、使用 Fortran 代码的语言间调用、库开发、应用程序优化与并行化以及 XL C/C++ 高性能库）的信息。

这些 PDF 文件可通过 Adobe Reader 来查看和打印。如果您没有安装 Adobe Reader，则您可从以下网址处下载它：

www.adobe.com

其它文档

与 XL C/C++ 相关的更多文档，包括红皮书、白皮书、教程和其它文章，都可以在 Web 上的以下网址处获得

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

技术支持

可以从 XL C/C++ 支持页面获得其它技术支持。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 和其它支持文档。您可以在 Web 上的以下网址处找到 XL C/C++ 支持页面：

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

如果找不到所需要的内容，可以发送电子邮件至以下位置：

ctscrcf@cn.ibm.com

有关 XL C/C++ 的最新信息，请访问以下网址处的产品信息站点：

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它 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以下位置：

ctscrcf@cn.ibm.com

确保包括文档的名称、文档的章节号、XL C/C++ 的版本以及（如果适用的话）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++ 功能部件的概述

XL C/C++ Advanced Edition V8.0 for Linux 可以用于复杂且要进行大量计算的大型程序，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。它面向评估 XL C/C++ 的人以及要查找关于该产品的更多信息的新用户。

与其它 XL 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。

XL 编译器是从在各种平台（如 AIX、Linux 分发、OS/390、OS/400、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的公共代码库派生的 IBM C、C++ 和 Fortran 编译器这些较大系列的一部分。公共代码库除了符合国际编程语言标准之外，还有助于确保在多个操作系统和硬件平台之间编译器性能是一致的且容易移植程序。

XL 编译器可供在 AIX 和选择的 Linux 分发上使用。

文档、联机帮助和技术支持

本指南提供了 XL C/C++ 及其功能部件的概述。您还可以找到以下形式的更广泛的产品文档：

- 自述文件。
- 可安装的联机帮助页。
- 基于 HTML 的信息系统。
- 可移植文档格式（PDF）文档。
- 基于 Web 的联机技术支持。

有关与 XL C/C++ 一起提供的产品文档和技术支持的更多信息，请参阅：

- 第 viii 页的『IBM XL C/C++ 出版物』
- 第 ix 页的『其它文档』
- 第 ix 页的『技术支持』

硬件和操作系统支持

XL C/C++ Advanced Edition V8.0 for Linux 支持若干个 Linux 分发。有关受支持的分发和需求的完整列表，请参阅自述文件和 *XL C/C++ Advanced Edition V8.0 for Linux Installation Guide* 中的 System prerequisites。

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

为了最大限度地利用不同的硬件配置，编译器根据用于执行应用程序的机器的配置提供了许多性能调整选项。

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满足特定语言级别规范。提供编译器调用命令是为了支持所有标准化的 C 和 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**。这些 “_r” 调用指示编译器将目标文件链接和绑定至线程安全组件和库，并为编译器创建的数据和过程生成线程安全目标代码。

有关 XL C/C++ 编译器调用命令的更多信息，请参阅本书中的第 15 页的『使用 XL C/C++ 进行编译』或 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Invoking the compiler or a compiler component*。

编译器选项

您可以通过一大组提供的编译器选项来控制编译器的操作。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获取与来自其它平台的程序的兼容性以及执行许多其它公共任务（如果不执行这些任务，则需要更改源代码）。

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 C 或 C++ 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++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Compiler options reference*。

定制编译器配置文件

安装过程会在 `/etc/opt/ibmcomp/vac/8.0/vac.cfg` 处创建一个缺省的编译器配置文件。此配置文件包含几个定义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。如果是那样的话，XL C/C++ 提供了 **vac_configure** 实用程序，您可以使用该实用程序来创建其它配置文件。然后，您可以使用任何文本编辑器来修改那些文件以包含您自己的常用编译器选项设置。

有关创建和使用定制配置文件的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Customizing the configuration file*。

语言标准一致性

编译器支持 C 和 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：

- 用于支持并行化编程的 OpenMP 扩展。
- 用于支持 VMX 向量编程的语言扩展。

- GNU C 和 C++ 语言扩展的子集。

有关 C 和 C++ 语言规范和扩展的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Language Reference* 中的 Supported language standards。

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，编译器将这些选项映射至它们的对等 XL C/C++ 编译器选项。

gxlc 和 **gxlc++** 调用命令使用 `/etc/opt/ibmcomp/vac/8.0/gxlc.cfg` 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 `/etc/opt/ibmcomp/vac/8.0/gxlc.cfg` 文件以更好地满足您可能有的任何独特编译需求的需要。有关更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 Reusing GNU C/C++ compiler options with gxlc and gxlc++。

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 GCC V3.3 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来生成应用程序，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。


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 和 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，则编译器将发出警告、错误和严重错误消息。

有关更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 **-qlanglvl**。

库

XL C/C++ Advanced Edition V8.0 for Linux 附带提供了下列库：

- “SMP 运行时库”支持显式并行处理和自动并行处理。
- 用于 32 位和 64 位方式的已调整数学内部函数的 IBM Mathematics Acceleration Subsystem (MASS) 库。
- 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”包含编译器所需要的支持例程。

Mathematics Acceleration Subsystem 库

IBM Mathematics Acceleration Subsystem (MASS) 库由专门为了在 PowerPC 处理器体系结构上获取最优性能而进行了充分调整的标量和向量内部函数组成。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，也可以选择已针对特定处理器系列进行了调整的库。

MASS 库支持 32 位和 64 位编译方式，它们是线程安全的并在其相应的 libm 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 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++ Advanced Edition V8.0 for Linux 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 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++ Advanced Edition V8.0 for Linux Programming Guide* 中的 Using the Basic Linear Algebra Subprograms。

工具和实用程序

xlc_install

此交互式实用程序帮助您在系统上安装 XL C/C++。

new_install

在安装 XL C/C++ 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它编译器配置文件，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定制集合。

cleanpdf 命令

与概要定向反馈相关的命令，用于管理 PDFDIR 目录。从指定的目录、PDFDIR 目录或当前目录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，在将两个或多个 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

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

showpdf 命令

显示在概要定向反馈练习运行（在选项 **-qpdf1** 和 **-qshowpdf** 下的编译）中执行的所有过程的调用和块计数的命令。

gxlc 和 gxlc++ 实用程序

两种调用方法，它们将 GNU C 或 GNU C++ 调用命令转换为对应的 **xlc** 或 **xlc** 命令并调用 XL C/C++ 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 XL C/C++ 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、浮点和其它类型的操作的优化。
- 根据程序将在何处运行，针对特定机器类或非常具体的机器配置优化程序。

优化变换可以在运行时给予应用程序更好的整体性能。C 和 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。
- 重构生成的目标代码以最佳地使用 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++ 启用了编程模型（如 OpenMP），这允许您编写高性能代码。

可能的话，在尝试对代码进行优化之前，应在没有优化的情况下测试并调试代码。

有关优化技巧的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Programming Guide* 中的 *Optimizing your applications*。

有关与优化相关的编译器选项的总结，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Options for performance optimization*。

64 位对象能力

XL C/C++ 编译器的 64 位对象能力满足了更大存储器需求和更大处理能力的需要。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 位对象格式来满足 64 位可执行文件的需要。链接程序绑定 64 位对象以创建 64 位的可执行文件。注意，绑定在一起的对象必须具有相同对象格式。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
- 尝试在 32 位平台上运行 64 位应用程序

在 64 位和 32 位平台上，32 位可执行文件将像它们当前在 32 位平台上一样继续运行。

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++ Advanced Edition V8.0 for Linux Programming Guide* 中的 *Using 32-bit and 64-bit modes*。

共享内存并行化

XL C/C++ Advanced Edition V8.0 for Linux 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（OpenMP）
- 指示编译器自动生成共享内存并行化
- 基于消息传递的共享或分布式内存并行化（MPI）
- POSIX 线程（Pthread）并行化
- 使用 `fork()` 和 `exec()` 的低级 UNIX 并行化

有关更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux 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++ 和许多其它 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，请访问 www.openmp.org。

诊断列表

编译器输出列表具有您可以包括或省略的可选节。有关适用的编译器选项及该列表本身的更多信息，请参阅第 23 页的『XL C/C++ 编译器列表』。

符号调试器支持

当调试程序时，可以使用 **gdb** 或其它任何符号调试器。

第 2 章 V8.0 中的新增内容

XL C/C++ Advanced Edition V8.0 for Linux 中的新功能部件和增强功能分为四个类别:

- 『性能和优化』
- 第 10 页的『语言增强和 API 的支持』
- 第 10 页的『易于使用』
- 第 11 页的『新的编译器选项』

性能和优化

许多新的功能部件和增强功能都可归入优化和性能调整类别。

特定于体系结构和处理器的代码调整

-qarch 编译器选项控制为指定的机器体系结构生成的特定指令。**-qtune** 编译器选项调整指令、调度和其它优化方法以在指定的硬件上增强性能。这些选项共同起作用以生成对于指定的体系结构可获得最佳性能的应用程序代码。

XL C/C++ V8.0 增大了可用于 **-qarch** 编译器选项的子选项列表，以支持那些支持 VMX 指令集的处理器和新提供的 POWER5+ 处理器。提供了下列新的 **-qarch** 选项:

- **-qarch=ppc64v**
- **-qarch=pwr5x**

高性能库

XL C/C++ 包括已充分调整的数学函数，它们可以在很大程度上提高具有大量数学计算的应用程序的性能。这些函数是通过以下高性能库提供的:

Mathematical Acceleration Subsystem (MASS)

MASS 库提供高性能的标量和向量函数来执行普通的数学计算。XL C/C++ Advanced Edition V8.0 for Linux 附带提供的 MASS 库引入了新的标量和向量函数以及对 POWER5 处理器体系结构的新支持。

有关使用 MASS 库的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Programming Guide* 中的 Using the Mathematical Acceleration Subsystem。

基本线性代数子程序 (BLAS)

XL C/C++ Advanced Edition V8.0 for Linux 引入了高性能代数函数的 BLAS 集合。您可以使用这些函数执行以下操作:

- 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++ Advanced Edition V8.0 for Linux Programming Guide* 中的 Using the Basic Linear Algebra Subprograms。

其它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且在以上各节中尚未提及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的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 Options for performance optimization。

表 3. 其它与性能相关的编译器选项和伪指令

选项 / 伪指令	描述
-qfloat	-qfloat 添加了下列新的子选项: -qfloat=relax 此子选项降低严格的 IEEE 符合性以换取更快的速度，通常是通过除去某些不重要的浮点算术运算（如涉及右边的零的加法和减法）来实现的。 -qfloat=norelax 这是缺省设置。将严格与 IEEE 保持一致。
-qipa	-qipa 添加了下列新的子选项: -qipa=clonearch=arch{,arch} 指定一个或多个将生成同一指令集的多个版本的处理器体系结构。 XL C/C++ 允许您指定多个将生成指令集的特定处理器体系结构。在运行时，应用程序将检测操作环境的特定体系结构并选择专门用于该体系结构的指令集。 -qipa=cloneproc=name{name} 指定要为由 clonearch 子选项指定的处理器体系结构克隆的一个或多个函数的名称
-O	指定 -O3 编译器选项现在指示编译器也采用 -qhot=level=0 编译器选项设置。 指定 -O4 或 -O5 编译器选项现在指示编译器也采用 -qhot=level=1 编译器选项设置。

本发行版新增的内置函数

下表列示了本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++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 Built-in functions for POWER and PowerPC architectures。

表 4. XL C/C++ 的内置函数

函数	描述
<code>void __builtin_return_address (unsigned int level);</code>	返回当前函数或它的一个调用程序的返回地址，其中 <i>level</i> 是常量文字，指示要在调用堆栈中扫描的帧数。
<code>void __builtin_frame_address (unsigned int level);</code>	返回当前函数或它的一个调用程序的函数帧地址，其中 <i>level</i> 是常量文字，指示要在调用堆栈中扫描的帧数
<code>int __compare_and_swap(volatile int* addr, int* old_val_addr, int new_val);</code>	执行将单字变量的内容与存储的旧值进行比较的原子操作。
<code>int __compare_and_swaplp(volatile long* addr, long* old_val_addr, long new_val);</code>	执行将双字变量的内容与存储的旧值进行比较的原子操作。

表 4. XL C/C++ 的内置函数 (续)

函数	描述
<code>int __fetch_and_add(volatile int* addr, int val);</code>	在单个原子操作中将由 <i>addr</i> 指定的单字增加由 <i>val</i> 指定的数量。
<code>long __fetch_and_addlp(volatile long* addr, long val);</code>	在单个原子操作中将由 <i>addr</i> 指定的双字增加由 <i>val</i> 指定的数量。
<code>unsigned int __fetch_and_and(volatile unsigned int* addr, unsigned int val);</code>	在单个原子操作中, 通过将该值与输入 <i>val</i> 参数进行“与”运算, 清除由 <i>addr</i> 指定的单字中的位。
<code>unsigned long __fetch_and_andlp(volatile unsigned long* addr, unsigned long val);</code>	在单个原子操作中, 通过将该值与输入 <i>val</i> 参数进行“与”运算, 清除由 <i>addr</i> 指定的双字中的位。
<code>unsigned int __fetch_and_or(volatile unsigned int* addr, unsigned int val);</code>	在单个原子操作中, 通过将该值与输入 <i>val</i> 参数进行“或”运算, 设置由 <i>addr</i> 指定的单字中的位。
<code>unsigned long __fetch_and_orlp(volatile unsigned long* addr, unsigned long val);</code>	在单个原子操作中, 通过将该值与输入 <i>val</i> 参数进行“或”运算, 设置由 <i>addr</i> 指定的双字中的位。
<code>unsigned int __fetch_and_swap(volatile unsigned int* addr, unsigned int val);</code>	在单个原子操作中, 将由 <i>addr</i> 指定的单字设置为该值或输入 <i>val</i> 参数并返回内存位置的原始内容。
<code>double __frim(double val);</code>	以双精度格式获取输入 <i>val</i> , 将 <i>val</i> 向下舍入为下一个较小的整数值, 并以双精度格式返回结果。仅对 POWER5+ 处理器有效。
<code>float __frims(float val);</code>	以浮点格式获取输入 <i>val</i> , 将 <i>val</i> 向下舍入为下一个较小的整数值, 并以浮点格式返回结果。仅对 POWER5+ 处理器有效。
<code>double __frin(double val);</code>	以双精度格式获取输入 <i>val</i> , 将 <i>val</i> 四舍五入为最接近的整数值, 并以双精度格式返回结果。仅对 POWER5+ 处理器有效。
<code>float __frins(float val);</code>	以浮点格式获取输入 <i>val</i> , 将 <i>val</i> 四舍五入为最接近的整数值, 并以浮点格式返回结果。仅对 POWER5+ 处理器有效。
<code>double __frip(double val);</code>	以双精度格式获取输入 <i>val</i> , 将 <i>val</i> 向上舍入为下一个较大的整数值, 并以双精度格式返回结果。仅对 POWER5+ 处理器有效。
<code>float __frips(float val);</code>	以浮点格式获取输入 <i>val</i> , 将 <i>val</i> 向上舍入为下一个较大的整数值, 并以浮点格式返回结果。仅对 POWER5+ 处理器有效。
<code>double __friz(double val);</code>	以双精度格式获取输入 <i>val</i> , 将 <i>val</i> 舍入为最接近零的下一个整数值, 并以双精度格式返回结果。仅对 POWER5+ 处理器有效。
<code>float __frizs(float val);</code>	以浮点格式获取输入 <i>val</i> , 将 <i>val</i> 舍入为最接近零的下一个整数值, 并以浮点格式返回结果。仅对 POWER5+ 处理器有效。

表 4. XL C/C++ 的内置函数 (续)

函数	描述
<code>long __ldarx(volatile long* addr);</code>	生成“装入双字并保留索引”（ldarx）指令。可以将此指令与随后的 stwcx. 指令配合使用以在指定的内存位置实施“读-修改-写”。
<code>int __lwarx(volatile int* addr);</code>	生成“装入单字并保留索引”（lwarx）指令。可以将此指令与随后的 stwcx. 指令配合使用以在指定的内存位置实施“读-修改-写”。
<code>int __stdcx(volatile long* addr, long val);</code>	生成“存储双字条件索引”（stdcx.）指令。可以将此指令与前面的 ldarx 指令配合使用以在指定的内存位置实施“读-修改-写”。
<code>int __stwcx(volatile int* addr, int val);</code>	生成“存储单字条件索引”（stwcx.）指令。可以将此指令与前面的 lwarx 指令配合使用以在指定的内存位置实施“读-修改-写”。
<code>unsigned long __mftb();</code>	生成“从时基移动”（mftb）硬件指令。
<code>unsigned int __mftbu();</code>	生成“从时基上面移动”（mftb）硬件指令。

语言增强和 API 的支持

API 和语言增强可以在您开发应用程序时提供更多方便和灵活性，并使您能够更容易地开发可充分利用硬件平台能力的代码。

C、C++ 和 Fortran 的 OpenMP API V2.5 支持

XL C/C++ 现在支持 OpenMP API V2.5 标准。OpenMP 规范的这个最新级别将先前的 C/C++ 和 Fortran OpenMP 规范组合成 C/C++ 和 Fortran 的单个规范并解决了它们之间先前的不一致性。

OpenMP 应用程序编程接口（API）是可移植且可伸缩的编程模型，它提供用于在 C、C++ 和 Fortran 应用程序中开发用户定向的共享内存并行化的标准接口。该规范由 OpenMP 组织定义，该组织由计算机硬件和软件供应商组成，其中包括 IBM。

可以在以下网址处找到关于 OpenMP 规范的更多信息：

www.openmp.org

易于使用

XL C/C++ 包括以下新的功能部件以帮助您更容易使用编译器来开发应用程序。

新安装和配置实用程序

XL C/C++ 的本发行版引入了 **xlc_install** 和 **new_install** 实用程序，以帮助您轻松地安装和配置编译器以供在系统上初次使用。

对 IBM Tivoli License Manager 的支持

IBM Tivoli License Manager (ITLM) 是一个基于 Web 的解决方案，它可以帮助您在受支持的系统上管理软件使用测量和许可证分配服务。通常，ITLM 识别并监视系统上已安装并正在使用的产品。

IBM XL C/C++ Advanced Edition V8.0 for Linux 启用了 ITLM 以支持库存和使用情况特征符，这意味着 ITLM 不但能够检测 XL C/C++ 的产品安装，并且还能检测它的使用情况。

注：ITLM 不是 XL C/C++ 编译器产品的一部分，必须另外购买和安装。

一旦安装并激活了 ITLM，ITLM 就会扫描系统以找到指示系统上是否已安装给定产品的产品库存特征符。ITLM 还标识产品的版本、发行版和修订版级别。XL C/C++ 的特征符文件将安装到以下目录：

缺省安装

`/opt/ibmcmp/vac/8.0`

非缺省安装

`compiler/vac/8.0`，其中 *compiler* 是由 **--prefix** 安装选项指定的目标安装目录。

有关 IBM Tivoli License Manager Web 的更多信息，请参阅：

www.ibm.com/software/tivoli/products/license-mgr

新的编译器选项

可以在命令行上或通过应用程序源文件中嵌入的伪指令指定编译器选项。

新的命令行选项

下表总结了 XL C/C++ 的新增命令行选项。您可以在 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Compiler options reference* 中找到所有编译器选项的详细语法和使用信息。

选项	描述和注释
-qasm	-qasm 编译器选项现在添加了新的功能。现在您不仅可以此编译器选项来控制如何解释程序中的内联汇编程序语句，而且还可以控制是否为 <code>asm</code> 语句发出代码。
-qasm_as	-qasm_as 编译器选项的语法稍有更改。
-qdump_class_hierarchy	当此选项生效时，编译器将每个类对象的层次结构和虚函数表布局的表示转储到文件。
-qlist	-qlist 编译器选项添加了 offset 和 nooffset 子选项。指定 -qlist=offset 将指示编译器显示与过程开始处的目标列表偏移量，而不是显示与代码生成开始处的目标列表偏移量。
-qmakedep	-qmakedep 编译器选项添加了新的 gcc 子选项。指定 -qmakedep=gcc 将指示编译器以类似于 GNU C/C++ 编译器使用的格式生成 <code>make</code> 依赖性信息。
-MF	这个新的编译器选项为 -qmakedep 或 -M 选项生成的 <code>make</code> 依赖性文件指定文件名。

选项	描述和注释
-qppline	这个新的编译器选项允许在预处理的输出中生成 <code>#line</code> 伪指令。 -qnoppline 编译器选项禁止生成 <code>#line</code> 伪指令。
-qreserved_reg	这个新的编译器选项允许您保留一个或多个寄存器名称。除非用作堆栈指针、帧指针或在某些其它固定角色中使用，否则不能在编译期间使用保留的寄存器。
-qsourcetype	<p>本发行版添加了 assembler-with-cpp 作为 -qsourcetype 编译器选项的新子选项。</p> <p>通常，编译器通过文件的 .S 文件名后缀识别需要预处理的汇编程序源文件。编译器预处理 .S 源文件，然后将预处理器输出发送至汇编程序。</p> <p>在命令行上指定 -qsourcetype=assembler-with-cpp filename 指示编译器将出现在 assembler-with-cpp 后面的所有文件名看作需要预处理的汇编程序源文件，而不考虑文件名后缀。</p>
-qtplinst	这个新的编译器选项管理编译器如何执行模板的隐式实例化。
-qversion	指定 -qversion 编译器选项将返回正式的编译器产品名称和版本。

新增的 `pragma` 伪指令

下表总结了 XL C/C++ 的新增 `pragma` 伪指令选项。您可以在 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 `XL C/C++ Pragas` 中找到详细语法和使用信息。

# <code>pragma</code> 伪指令	描述和注释
<code>altivec_vrsave</code>	当 <code>altivec_vrsave</code> 伪指令生效时，函数的序言和结尾将包括用于维护 <code>VRSAVE</code> 寄存器的代码。仅当 -qaltivec 生效时此编译指示才有效，必须仅在函数中使用此编译指示且它仅影响它所在的函数。
<code>STDC CX_LIMITED_RANGE</code>	<code>STDC CX_LIMITED_RANGE</code> 伪指令指示编译器，在它控制的作用域内，将仅用值调用复数除法和绝对值，以便中间计算结果不会溢出或失去意义。

第 3 章 设置和定制 XL C/C++

本节提供关于安装和定制 XL C/C++ 的简短概述信息，并指出了如何找到描述特定安装和定制主题的更详细信息的其它文档。

环境变量和 XL C/C++

XL C/C++ 使用一些环境变量来控制编译器操作的各个方面。环境变量分为两个基本类别：

- 为编译器定义基本工作环境的环境变量。
- 定义运行时编译器选项缺省值的环境变量。

设置编译器工作环境

这些环境变量定义编译器的基本工作环境，包括指定您选择的本地语言或定义库或临时文件的位置。有关完整的信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Setting up the compilation environment*。

LANG 指定用来显示诊断消息和编译器列表的缺省本地语言语言环境。此环境变量还会影响运行时行为。

MANPATH

指定系统、编译器和第三方联机帮助页的搜索路径。

NLSPATH

指定可以找到消息目录的一个或多个目录位置。此环境变量还会影响运行时行为。

PDFDIR

指定当使用 **-qpdf** 选项编译时存储概要定向反馈信息的目录位置。

TMPDIR

指定编译器将存储程序编译期间创建的临时文件的目录位置。此环境变量还会影响运行时行为。

设置缺省运行时选项

这些环境变量定义编译器要使用的运行时编译器选项缺省值，除非它们被命令行上或位于程序源代码中的伪指令中指定的编译器选项设置显式地覆盖。有关完整的信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Setting up the compilation environment*。

XL_NOCLONEARCH

指示程序仅执行类属代码，其中类属代码是未对特定处理器体系结构建立版本的已编译目标代码。您可以设置 **XL_NOCLONEARCH** 环境变量来帮助您调试应用程序。

XLSMPOPTS

XLSMPOPTS 环境变量允许您指定影响 SMP 执行的运行时选项。

OMP_DYNAMIC、**OMP_NESTED**、**OMP_NUM_THREADS** 和 **OMP_SCHEDULE**

这些环境变量是 OpenMP 标准的一部分。它们允许您指定应用程序将如何执行各节并行代码。

定制配置文件

配置文件是纯文本文件，它指定编译器选项和调用的缺省设置。XL C/C++ 在编译器安装期间在文件 `/etc/opt/ibmcomp/vac/8.0/vac.cfg` 中提供缺省配置。

如果您正在单用户系统上运行，或编译环境已具有编译脚本或 `makefile`，则可能要让缺省配置文件保持原样。

或者，您可以创建其它定制配置文件来满足特定应用程序或应用程序组所需要的特殊编译要求。

有关创建和使用定制配置文件的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Customizing the configuration file*。

确定安装哪种级别的 XL C/C++

您可能不确定在特定机器上安装哪种级别的 XL C/C++。如果要与软件支持机构联系，您将需要知道此信息。

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

```
xlc -qversion
```

第 4 章 使用 XL C/C++ 编辑、编译和链接程序

基本的 C 和 C++ 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）和运行过程的重复循环。

先决条件信息:

1. 必须首先确保正确地配置所有 Linux 设置（例如，某些环境变量和存储器限制），然后才能使用此编译器。有关更多信息，请参阅第 13 页的『环境变量和 XL C/C++』。
2. 要更多地了解有关编写 C 和 C++ 程序的信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Language Reference*。

编译器阶段

典型的编译器调用命令按顺序执行下列某些或全部程序。对于链接时优化，将在编译期间多次执行某些阶段。当每个程序运行时，结果按顺序发送至下一个步骤。

1. 预处理器
2. 编译器，包括以下阶段:
 - a. 前端解析和语义分析
 - b. 循环变换
 - c. 过程间分析
 - d. 优化
 - e. 寄存器分配
 - f. 最终的汇编
3. 汇编程序（用于在进行预处理后的 **.s** 文件和 **.S** 文件）
4. 链接程序 **ld**

要查看编译器如何单步执行这些阶段，当编译应用程序时指定 **-qphsinfo** 和 **-v** 编译器选项。

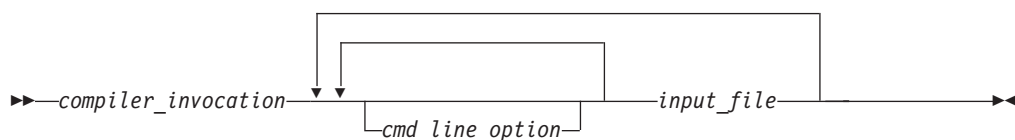
编辑 C 和 C++ 源文件

要创建 C 和 C++ 源程序，您可以使用任何可用的文本编辑器，如 **vi** 或 **emacs**。除非配置文件或 **-qsourcetype** 编译器选项定义了其它非标准文件名后缀，否则源程序必须使用可识别的文件名后缀。请参阅第 17 页的『XL C/C++ 输入文件』，以获取 XL C/C++ 可识别的文件名后缀列表。

要使 C 或 C++ 源程序成为有效的程序，它必须符合在 *XL C/C++ Advanced Edition V8.0 for Linux Language Reference* 中指定的语言定义。

使用 XL C/C++ 进行编译

要编译源程序，请通过下面显示的语法使用一个编译器调用命令:



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** 文件并将目标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，您应考虑使用 **xlC** 或 **xlC++** 或者其线程安全等效项进行编译。

xlC 和 **xlC++** 将编译诸如 C 或 C++ 的程序源代码，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。其它基本编译器调用命令的目的主要是为 C 或 C++ 语言的不同级别和扩展提供显式编译支持。

除了基本编译器调用命令以外，XL C/C++ 还提供了许多基本编译器调用的专用变体。基本编译器调用的变体是通过将后缀连接至该调用命令的名称来命名的。调用变体的后缀意义为：

_r 线程安全调用变体，它支持将 POSIX Pthread API 用于多线程应用程序，包括使用 **-qsmp** 或使用包含 OpenMP 程序并行化伪指令的源代码编译的应用程序。

表 5. XL C/C++ 编译器调用命令

基本调用	基本调用上的变体	描述
xlC xlC++	xlC_r xlC++_r	调用编译器以便将源文件作为 C++ 语言源代码编译。
xlC	xlC_r	调用编译器以便将源文件作为 C 源代码编译。
c99	c99_r	调用编译器以便严格遵循 ISO C99 标准（ISO/IEC 14882:1999）编译源文件。 注： ISO C99 标准也指定运行时库中的功能部件。这些功能部件在系统上当前安装的运行时库中可能不受支持。
c89	c89_r	调用编译器以便严格遵循 ISO C89 标准（ISO/IEC 9899:1990）编译源文件。
cc	cc_r	调用编译器以用于不要求符合 C89 或 C99 的旧的 C 代码。
gxlc		在将 GNU C 命令行选项转换为 XL C/C++ 选项之后调用编译器。 注： 并非每个 GNU C 选项都具有完全等价的 XL C/C++ 选项。
gxlc gxlc++		在将 GNU C++ 命令行选项转换为 XL C/C++ 选项之后调用编译器。 注： 并非每个 GNU C++ 选项都具有完全等价的 XL C/C++ 选项。

编译并行化的 XL C/C++ 应用程序

XL C/C++ 提供线程安全编译调用，您可以在编译要在多处理器环境下使用的并行化应用程序时使用这些调用。

- xlc_r
- xlc++_r
- xlc_r
- c99_r
- c89_r
- cc_r

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。

注：单独使用这些命令中的任何一个并不暗示并行化。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。而您只能将 **-qsmp** 选项与这六个调用命令中的一个一起指定。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。

XL C/C++ 输入文件

编译器的输入文件为：

源文件（**.c** 后缀表示 C 语言，**.C .cc .cp .cpp .cxx .c++** 后缀表示 C++ 语言）

编译器将具有这些后缀的文件视为要编译的 C 或 C++ 源文件。

编译器按您在命令行上指定的顺序编译源文件。如果编译器找不到指定的源文件，则编译器生成一条错误消息并继续编译下一个文件（如果下一个文件存在的话）。

如果您有不符合标准 C 或 C++ 文件命名约定的 C 或 C++ 源文件，则您可以使用 **-+** 编译器选项来指示编译器像处理 C 或 C++ 源文件一样处理这样的文件。当 **-+** 编译器选项生效时，除了有 **.a**、**.o**、**.so**、**.s** 或 **.S** 文件名后缀的文件以外，这样的文件将作为 C++ 源文件进行编译。

包含文件也包含源代码，且常常具有与平常用于 C 或 C++ 源文件的那些后缀不同的后缀。

预处理源文件（**.i** 后缀）

编译器将预处理源文件 *filename.i* 发送至编译器，将在那里再次像对待 **.c** 或 **.C** 文件一样预处理它。预处理文件可用于检查宏和预处理器伪指令。

目标文件（**.o** 后缀）

在编译器编译源文件之后，它使用 **ld** 命令来链接产生的 **.o** 文件、您指定为输入文件的任何 **.o** 文件以及产品和系统库目录中某些 **.o** 和 **.a** 文件。然后编译器可以从这些目标文件生成单个 **.o** 目标文件或单个可执行输出文件。

汇编程序源文件（**.s** 和 **.S** 后缀）

编译器将汇编程序源文件发送至汇编程序（**as**）。汇编程序在链接时将目标文件发送至链接程序。

注：编译器首先预处理文件名后缀为 **.S** 的汇编程序源文件，然后将它发送至汇编程序。

共享目标或库文件（.so 后缀）

这些文件是可以在运行时装入并由多个进程共享的目标文件。当在链接期间指定了共享目标时，关于该目标的信息记录在输出文件中，但输出文件中实际上不包括来自共享目标的代码。

配置文件（.cfg 后缀）

配置文件的内容确定编译过程的许多方面，最普通的是确定编译器的缺省选项。可以使用配置文件来集中不同的缺省编译器选项组或使 XL C/C++ 编译器的多个级别在系统上存在。

缺省配置文件是 `/etc/opt/ibmcomp/vac/8.0/vac.cfg` 和 `/etc/opt/ibmcomp/vac/8.0/gxlc.cfg`。

概要数据文件

-qpdf1 选项生成运行时概要信息以在后续编译中使用。此信息存储在名称与模式 `*pdf*` 匹配的一个或多个隐藏文件中。

XL C/C++ 输出文件

C 和 C++ 生成的输出文件为：

可执行文件: **a.out**

缺省情况下，XL C/C++ 在当前目录中生成命名为 **a.out** 的可执行文件。

目标文件: *filename.o*

如果您指定 **-c** 编译器选项，则不生成可执行文件，而是编译器为每个指定的程序源代码输入文件生成一个目标文件，并且汇编程序为每个指定的汇编程序输入文件生成一个目标文件。缺省情况下，将使用与输出目标文件的相应源代码输入文件相同的文件名前缀将该输出目标文件保存到当前目录。

汇编程序源文件: *filename.s*

如果您指定 **-S** 编译器选项，则不会生成可执行文件，而是 XL C/C++ 编译器为每个指定的输入源文件生成一个等价的汇编程序源文件。缺省情况下，将使用与输出汇编程序源文件的相应源代码输入文件相同的文件名前缀将该输出汇编程序源文件保存到当前目录。

编译器列表文件: *filename.lst*

缺省情况下，除非您指定了一个或多个与列表相关的编译器选项，否则不会生成列表。列表文件将放置在当前目录中，其文件名前缀与源文件的相同。

cpp 预处理源文件: *filename.i*

要创建预处理源文件，在编译时指定 **-P** 选项。源文件将会被预处理但不会被编译。也可以通过 **-E** 选项重定向输出以生成包含 `#line` 伪指令的预处理文件。将为每个源文件生成一个预处理源文件 *filename.i*。缺省情况下，将使用与输出预处理器源文件的相应源代码输入文件相同的文件名前缀将该输出预处理器源文件保存到当前目录。

生成依赖性文件: *filename.u*

当 **-M** 或 **-qmakedep** 编译器选项生效时，编译器为编译的每个 C 或 C++ 源文件创建一个 `.u` 文件。可以使用由 `.u` 文件提供的依赖性信息来帮助您创建 `make` 文件。

每个 `.u` 文件包含一行表示输入文件，并包含一个条目表示每个包含文件，该文件采用以下通用格式：

```
file_name.o :file_name.c
file_name.o :include_file_name
```

将根据编译器搜索 `#include` 预处理器伪指令的顺序规则来列示包含文件。如果找不到包含文件，则不会将它添加至 `.u` 文件。不具有 `include` 语句的文件所生成的输出文件将包含一个仅列示输入文件名的行。

概要数据文件 (*.pdf*)

它们是 `-qpdf1` 编译器选项生成的概要定向反馈文件。将在后续编译中使用它们以根据实际执行结果来调整优化。

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、描述要生成的目标代码、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。
- 在编译器配置文件内找到的节中
- 在源代码中使用伪指令语句
- 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，则最后指定的那个选项起作用。

注：`-I` 编译器选项是一种特殊情况。编译器在搜索在命令行上用 `-I` 指定的目录之前，它首先搜索在 `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。

具有累积行为的其它选项是 `-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++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Compiler options reference*。

链接 XL C/C++ 程序

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：

```
x1C file1.C file2.o file3.C
```

编译并生成目标文件 `file1.o` 和 `file3.o`，然后将所有目标文件（包括 `file2.o`）提交至链接程序以生成一个可执行文件。

在链接之后，遵循第 21 页的第 5 章，『运行 XL C/C++ 程序』中的指示信息来执行程序。

在单独的步骤中编译和链接

要生成稍后可以链接的目标文件，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 appropriate libraries
```

通常最好的方法是通过编译器调用命令执行链接程序，因为这会自动将某些额外的 **ld** 选项和库名传递至链接程序。

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，则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，则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。

因为此格式的链接是缺省值，所以您不需要其它选项来打开它。

- 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++ 库。如果它们执行许多对库例程的调用或调用许多小例程，则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，则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。如果您在操作系统的一个级别上编译它们而在操作系统的不同级别上运行它们，则它们也可能不工作。

有关链接您的程序的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 *Invoking the linkage editor*。

此外，有关编译和链接库的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Programming Guide* 中的 *Constructing a library*。

第 5 章 运行 XL C/C++ 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。

可以通过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来运行程序。

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，则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。其它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，请参阅第 13 页的『环境变量和 XL C/C++』。

在其它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，则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 页面上的 Download 部分获取最新的“XL C/C++ 运行时环境”安装映像以及许可证发放和使用信息：

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

第 6 章 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来帮助标识和更正这些问题。

本节提供由 XL C/C++ 提供的主要诊断消息的简短概述。有关可以帮助您解决应用程序的问题的相关编译器选项的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 Options for error checking and debugging 和 Options that control listings and messages。

编译返回码

在编译结束时，在以下任何一种情况下，编译器将返回码设置为零：

- 未发出任何消息。
- 诊断到的所有错误的最高严重性级别小于 **-qhalt** 编译器选项的设置，并且错误数未达到 **-qmaxerr** 编译器选项所设置的限制。。
- 未发出 **-qhaltonmsg** 编译器选项所指定的消息。

否则，编译器将返回码设置为下列其中一个值：

返回码	错误类型
1	发生了严重性级别比 -qhalt 编译器选项的设置更高的错误。
40	发生了选项错误或不可恢复错误。
41	发生了配置文件错误。
250	发生了内存不够错误。编译器调用命令不能再分配任何内存供它使用。
251	发生了“已接收到信号”错误。即，发生了不可恢复错误或中断信号。
252	发生了“找不到文件”错误。
253	发生了输入 / 输出错误 - 不能读取或写入文件。
254	发生了派生错误。不能创建新的进程。
255	当进程运行时检测到错误。

注：还会显示返回码来表示运行时错误。

XL C/C++ 编译器列表

将根据 **-qlist**、**-qsource**、**-qxref**、**-qattr**、**-qreport** 和 **-qlistopt** 编译器选项的设置，在输出列表中生成诊断信息。**-S** 选项在单独的文件中生成汇编程序列表。

如果编译器在编译应用程序时遇到编程错误，则编译器将一条诊断消息发送到标准错误设备，如果选择了适当的编译器选项，则还会发送到一个列表文件。

要通过列表的帮助找到问题的原因，可以参阅以下各项：

- 源代码节（以查看源程序上下文中的任何编译错误）
- 属性和交叉引用节（以查找错误命名或在未声明的情况下使用的数据对象，或查找不匹配的参数）

- 变换和目标节（以查看生成的代码是否类似于您所期望的代码）

标题标识列表的每个主要节。节标题前有一串“大于”（>）号，以便您可以很容易找到它的开头：

```
>>>> section name
```

可以通过指定适当的编译器选项来选择哪些节会出现在列表中。有关这些选项的更多信息，请参阅 *XL C/C++ Advanced Edition V8.0 for Linux Compiler Reference* 中的 Options for error checking and debugging 和 Options that control listings and messages。

头部节

列表文件有一个包含下列项的头部节：

- 由下列各项组成的编译器标识：
 - 编译器名称
 - 版本号
 - 发行版号
 - 修订版号
 - 修订数
- 源文件名称
- 编译日期
- 编译时间

头部节总是在列表中存在，它是第一行且仅出现一次。当存在多个编译单元时，将对每个编译单元重复下列各节。

选项节

选项节总是在列表中存在。每个编译单元都有一个单独的节。它指示对编译单元生效的指定选项。当您遇到有冲突的选项时此信息很有用。如果您指定 **-qlistopt** 编译器选项，则此节列示所有选项的设置。

源代码节

源代码节包含具有行号和文件号（可选）的输入源代码行。文件号指示产生源代码行的源文件（或包含文件）。所有主文件的源代码行（不是来自包含文件的那些行）都未打印文件号。每个包含文件都有一个与它相关联的文件号，来自包含文件的源代码行都打印了该文件号。文件号位于左边，行号位于文件号的右边，源代码行文本位于行号的右边。编译器将与每个文件相关的行进行编号。仅当 **-qsource** 编译器选项生效时，源代码行和与其相关联的号码才会出现。

如果 **-qsource** 选项生效，则错误消息散布在源列表中。在编译过程中生成的错误消息包含下列各项：

- 源代码行
- 指向出错列的指示符行
- 错误消息。

例如：

如果 **-qnosource** 选项生效，则错误消息全部位于源代码节中，错误消息包括下列各项：

- 用引号括起的文件名
- 错误的行号和列位置
- 错误消息。

```
"loop.c", line 9.16: 1506-277 (S) Syntax error: possible missing ',' or ']'?
```

C 和 C++ 诊断消息的格式为:

其中:

severity_letter 指示问题的严重性，如上一节所述。

编译错误可以具有下列严重性级别，这些严重性级别作为某些错误消息的一部分显示：

S 严重错误。由于下列其中一种原因, 编译失败:

- 存在编译器不能更正的情况。
- 内部编译器表已溢出。程序的处理停止，XL C/C++ 未生成目标文件。
- 包含文件不存在。程序的处理停止，XL C/C++ 未生成目标文件。
- 检测到不可恢复的程序错误。停止处理源文件，XL C/C++ 未生成目标文件。通常可以通过修正编译期间报告的任何程序错误来改正此错误。

E 仅适用于 C 编译。编译器在源代码中检测到错误并尝试改正它。编译器将继续编译应用程序，但可能不会生成您期望的结果。

W 警告消息。编译器在源代码中检测到潜在的问题，但不会尝试改正它。编译器将继续编译应用程序，但可能不会生成您期望的结果。

参考消息。它不指示任何错误，只是一些您应知道以避免意外行为的信息。

第 6 章 XL C/C++ 编译器诊断辅助 25

缺省情况下，如果编译器遇到严重错误（严重性级别 S），则它停止而不会生成输出文件。您可以通过使用 **-qhalt** 选项指定不同的严重性，使编译器因为严重性较小的错误而停止。例如，使用 **-qhalt=w** 时，如果编译器遇到严重性为 W 或更高严重性的任何错误，则它停止。此技术可以减少检查程序的语法和语义有效性所需要的编译时间量。可以通过使用 **-qflag** 选项来限制较低严重性的消息而不停止编译器。

变换报告节

如果 **-qreport** 选项生效，则编译器生成一个变换报告，显示 XL C/C++ 是如何对程序进行优化的。本节显示对应于原始源代码的伪码，以便您可以查看 **-qhot** 或 **-qsmp** 选项生成的并行化和循环变换。

属性和交叉引用节

本节提供关于在编译单元中使用的实体的信息。如果 **-qxref** 或 **-qattr** 编译器选项生效，则该节存在。根据生效的选项，本节包含全部或部分关于在编译单元中使用的实体的下列信息：

- 实体的名称
- 实体的属性（如果 **-qattr** 生效）。属性信息可以包括下列详细信息的所有或部分：
 - 数据类型
 - 名称的类
 - 名称的相对地址
 - 对齐
 - 维数
 - 对于数组，它是否是可分配的
 - 它是指针、目标还是整数指针
 - 它是否是参数
 - 它是否是易失的
 - 对于哑自变量，它的意向、它是否是值以及它是否是可选的
 - 专用、公用或受保护模块
- 指示您在何处定义、引用或修改实体的坐标。如果您声明了实体，则坐标将用 \$ 进行标记。如果您初始化了实体，则坐标将用 * 进行标记。如果您在相同位置声明并初始化了实体，则坐标将用 & 进行标记。如果设置了实体，则坐标将用 @ 进行标记。如果引用了实体，则不会标记坐标。

如果您指定带 **-qxref** 或 **-qattr** 的 **full** 子选项，则 C 和 C++ 会报告编译单元中的所有实体。如果您未指定此子选项，则仅出现您实际使用的实体。

目标节

仅当 **-qlist** 编译器选项生效时，XL C/C++ 才生成此节。它包含伪汇编程序目标代码列表，该列表显示源代码行号、以十六进制表示法表示的指令偏移量、指令的汇编程序助记符和指令的十六进制值。在右边，该节还显示指令的循环时间和编译器的中间语言。最后，将显示总循环时间（直线执行时间）和生成的机器指令总数。每个编译单元都有一个单独的节。

注：要获取真实的汇编程序列表，在编译应用程序时指定 **-S** 编译器选项。汇编程序列表将命名为 *filename.s*。

文件表节

本节包含一个表，该表显示使用的每个主源文件和包含文件的文件号和文件名。它还列示了引用包含文件的主源文件的行号。本节总是存在。

编译单元收尾节

这是每个编译单元的列表的最后一节。它包含诊断总结并指示单元是否编译成功。如果文件仅包含一个编译单元，则本节在列表中不存在。

编译收尾节

编译收尾节仅在列表的末尾出现一次。完成编译时，XL C/C++ 提供一个编译总结：读取的源记录数、编译开始时间、编译结束时间、总编译时间、总 CPU 时间和虚拟 CPU 时间。本节在列表中总是存在。

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** 或其它任何的符号调试器来单步执行和检查已编译的应用程序的行为。

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：

IBM Corporation
Lab Director
IBM Canada Limited
8200 Warden Avenue
Markham, Ontario, Canada
L6G 1C7

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。

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：

本信息可能包含用源语言编写的样本应用程序，该程序举例说明了在各种操作平台上的编程技术。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。

编程接口信息

编程接口信息用来帮助您使用此程序来创建应用程序软件。

通用编程接口允许用户编写获取此程序工具的服务的应用软件。

然而，本信息还可能包含诊断、修改和调整信息。诊断、修改和调整信息用来帮助您调试应用程序软件。

注：不要将此诊断、修改和调整信息用作编程接口，因为它将来还会有更改。

商标和服务标记

本出版物中使用的下列各项是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标：

AIX	IBM	OS/390
OS/400	POWER3	POWER4
POWER5	POWER5+	PowerPC
z/OS	z/VM	

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

OpenMP 是 OpenMP Architecture Review Board 的商标。

UNIX 是 Open Group 在美国和其他国家或地区的注册商标。

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

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。

索引

[B]

- 帮助系统 1
- 编辑源文件 15
- 编译
 - 描述如何编译程序 15
 - SMP 程序 17
- 编译器列表 23
- 编译器列表中编译单元收尾节 27
- 编译器列表中的变换报告节 26
- 编译器列表中的交叉引用节 26
- 编译器列表中的收尾节 27
- 编译器列表中的属性节 26
- 编译器列表中的头部节 24
- 编译器列表中的文件表节 27
- 编译器列表中的选项节 24
- 编译器列表中的源代码节 24
- 编译器选项
 - 编译器列表中的节 24
 - 请参阅 在索引开始处列示在特殊字符下的各个选项
- 并行化 6
- 不可恢复错误 25

[C]

- 参考消息 25
- 参数
 - 请参阅 自变量
- 错误消息
 - 格式的解释 25
 - 在编译器列表中 24

[D]

- 代码优化 5
- 调试 23, 27
- 调试器支持 6, 27
- 调用编译器 15
- 调用程序 21
- 定制配置文件（包括缺省编译器选项） 14
- 动态链接 20
- 多处理器系统 6

[F]

- 符号调试器支持 6

[G]

- 概要分析数据文件 18
- 工具 4
 - 安装实用程序 4
 - 配置文件实用程序 4
 - 新安装配置实用程序 4
 - cleanpdf 4
 - gxlc 和 gxlc++ 4
 - mergepdf 4
 - new_install 4
 - resetpdf 4
 - showpdf 4
 - vac_configure 4
 - xlc_install 4
- 共享目标文件 17
- 共享内存并行化 6
- 归档文件 17

[H]

- 环境变量
 - 编译时 13
 - 运行时 13
- 汇编程序
 - 源 (.S) 文件 17
 - 源 (.s) 文件 17, 18

[J]

- 警告错误 25
- 静态链接 20

[K]

- 可执行文件 18
- 库 17

[L]

- 联机编译器帮助 1
- 联机文档 1
- 链接 19
 - 动态 20
 - 静态 20
- 列表文件 18

[M]

- 命令行选项
 - 请参阅 编译器选项
- 目标文件 17, 18

[P]

- 配置文件 14, 17

[Q]

- 缺省值
 - 定制编译器缺省值 14

[S]

- 示例程序
 - 请参阅 样本程序
- 输出文件 18
- 输入文件 17

[W]

- 文本编辑器 15
- 文档, 联机格式 1
- 文件
 - 编辑源代码 15
 - 输出 18
 - 输入 17
- 问题确定 23

[Y]

- 严重错误 25
- 优化 5
- 语言支持 2
- 源代码一致性检查 3
- 源级别调试支持 6
- 源文件 17
- 运行编译器 15
- 运行程序 21
- 运行时
 - 库 17
- 运行时环境 21
- 运行时选项 21

[Z]

执行编译器 15
执行程序 21

[数字]

64 位环境 5

A

ANSI
 检查与 C/C++ 标准的一致性 3
a.out 文件 18

E

E 错误严重性 25

H

HTML 文档 1

I

I 错误严重性 25
ISO
 检查与 C/C++ 标准的一致性 3
I/O
 请参阅 输入 / 输出

M

makefile
 作为缺省选项的替代项的配置文件 14
mod 文件 17, 18

O

OpenMP 6

P

PDF 文档 1

S

S 错误严重性 25
SMP
 程序, 编译 17
SMP 程序 6

U

U 错误严重性 25

W

W 错误严重性 25

X

XL C/C++ 的级别, 确定 14
XL C/C++ 消息的 15xx 标识 25

[特别字符]

-qattr 编译器选项 26
-qlist 编译器选项 26
-qlistopt 编译器选项 24
-qreport 编译器选项 26
-qsource 编译器选项 24
-qxref 编译器选项 26
.a 文件 17
.c 和 .C 文件 17
.cfg 文件 17
.lst 文件 18
.mod 文件 17, 18
.o 文件 17, 18
.S 文件 17
.s 文件 17, 18
.so 文件 17
/etc/opt/ibmcomp/vacpp/8.0/vac.cfg 配置文件
14



程序号: 5724-M16

S151-0265-00

