

IBM XL C for AIX V13.1.3



XL C 入門

V13.1.3

IBM XL C for AIX V13.1.3



XL C 入門

V13.1.3

注意

在使用本资料及其支持的产品之前，请参阅第 35 页的『声明』中的信息。

第一版版本

本版本适用于 IBM XL C for AIX V13.1.3（程序 5765-J06 和 5725-C71）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 IBM Corporation 1996, 2015.

目录

关于本文档	v
约定	v
相关信息	viii
IBM XL 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 简介	1
与其他 IBM 编译器的通用性	1
操作系统和硬件支持	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	3
源代码迁移和一致性检查	3
库	3
工具、实用程序和命令	4
程序优化	5
64 位对象能力	6
共享内存并行化	6
诊断报告	7
符号调试器支持	7
第 2 章 迁移应用程序	9
迁移使用事务型内存内置函数的应用程序	9
混合通过不同编译器编译的对象文件	9
第 3 章 早期发行版中新增的增强功能	11
V13.1.2 中新增的增强功能	11
内置函数	11
命令	11
编译器选项	12
V13.1 中新增的增强功能	12
支持 POWER8 处理器	12
C11 功能	13

OpenMP 4.0	14
内置函数	14
编译器选项和编译指示伪指令	18
性能和优化	20
V12.1 中新增的增强功能	20
C11 功能	20
OpenMP 3.1	21
性能和优化	22
诊断报告	22
内置函数	23
编译器选项和编译指示伪指令	24

第 4 章 设置和定制 XL C	27
使用定制编译器配置文件	27
配置编译器使用情况跟踪和报告	27

第 5 章 使用 XL C 开发应用程序	29
编译器阶段	29
编辑 C 源文件	29
使用 XL C 进行编译	29
调用编译器	29
编译并行化 XL C 应用程序	30
指定编译器选项	30
XL C 输入和输出文件	31
将已编译的应用程序与 XL C 链接在一起	32
重新链接现有的可执行文件	32
动态和静态链接	33
运行已编译的应用程序	33
XL C 编译器诊断辅助	34
调试已编译的应用程序	34
确定使用哪一个级别的 XL C	34

声明	35
商标	37

索引	39
---------------------	-----------

关于本文档

本文档包含 IBM® XL C for AIX® V13.1.3 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 的介绍性概述和用法信息的 C 开发者编写的。它假设您已经较为熟悉命令行编译器，具备 C 编程语言的基本知识以及操作系统命令的基本知识。对 XL C 不熟悉的程序员可以使用本文档来查找有关 XL C 独有的功能和功能的信息。

如何使用本文档

本文档中通篇使用 **xlc** 编译器调用来描述编译器的行为。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。

虽然本文档涵盖诸如配置编译器环境以及使用 XL C 编译器来编译和链接 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 安装指南*》。
- 编译器选项：请参阅 *XL C Compiler Reference*，了解有关编译器选项的语法和用法的详细信息。
- C 编程语言：请参阅《*XL C 语言参考*》，了解有关 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 Optimization and Programming Guide*，了解有关使用 XL C 开发应用程序的详细信息，其中着重介绍了程序可移植性和优化。

约定

印刷约定

下表说明了 IBM XL C for AIX V13.1.3 中使用的印刷约定 信息。





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。	编译器提供了基本的调用命令 xlc 以及若干其他编译器调用命令，以支持各种 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。斜体字还用于介绍新术语。	如果返回的值超过请求的 <i>size</i> ，请确保更新 <i>size</i> 参数。
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。	要编辑 并优化 myprogram.c，请输入： xlc myprogram.c -O3 。

限定元素（图标）

在语言元素描述中，如果某个功能部件是 C11 标准所独有的或者是 C 标准的 IBM 扩展，那么此信息将使用图标来描绘文本文段，如下所示：

表 2. 限定元素

限定符/图标	含义
IBM 扩展开始  	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。
IBM 扩展结束	
C11 开始  	该文本描述一个功能部件，该功能部件是作为 C11 的一部分引入至标准 C 的。
C11 结束	

语法图

在本信息中，自始至终都用图来举例说明 XL C 语法。本节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、伪指令或语句续上一行。
- ▶ 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：

▶—keyword—required_argument—▶

- 可选的项显示在主路径之下：

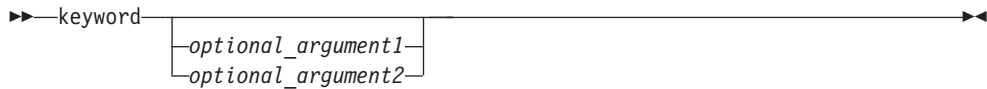
▶—keyword—
 |optional_argument|—▶

- 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。

▶—keyword—
 |required_argument1|
 |required_argument2|—▶

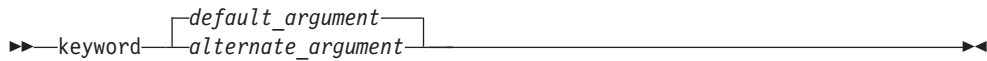
如果这些项是可选的，那么所有项都会显示在主路径下。



- 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：



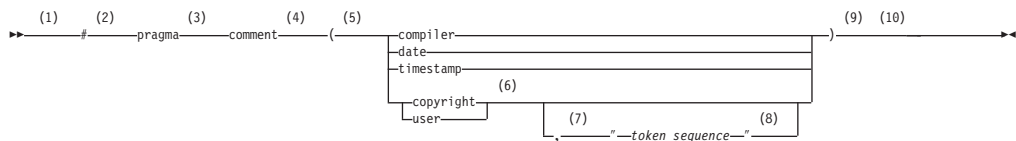
- 缺省项显示在主路径之上。



- 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。
- 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")
```

语法语句的示例

EXAMPLE *char_constant* {*a|b*}[*c|d*]*e*[,*e*]*... name_list*{*name_list*}*...*

以下列表说明语法语句:

- 输入关键字 EXAMPLE。
- 输入 *char_constant* 的值。
- 输入 *a* 或 *b* 的值,但不能同时输入。
- (可选)输入 *c* 或 *d* 的值。
- 至少输入 *e* 的一个值。如果您输入多个值,那么必须用逗号对每个值进行分隔。
- (可选)至少输入 *name_list* 的一个 *name* 的值。如果您输入多个值,那么必须用逗号对每个 *name* 进行分隔。

注: 语法语句和语法图表示中均使用同一个示例。

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 的相关信息:

IBM XL C 信息

XL C 以下列格式提供产品信息:

- 《快速入门指南》

《快速入门指南》(quickstart.pdf) 旨在帮助您了解 IBM XL C for AIX V13.1.3。缺省情况下,《快速入门指南》位于 XL C 目录和安装 DVD 的 \quickstart 目录下。

- 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,自述文件文件位于 XL C 目录和 DVD 安装的根目录和子目录下。

- 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 for AIX V13.1.3 安装指南》中提供了安装和访问联机帮助页的指示信息。

- 在线产品文档

可以在 IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSGH2K_13.1.3/com.ibm.compilers.aix.doc/welcome.html) 查看完全可搜索的基于 HTML 的文档。

- PDF 文档

Web 上也提供了 PDF 文件，网址为 <http://www.ibm.com/support/docview.wss?uid=swg27036590>。

下列文件组成了一套完整的 XL C 产品信息：

表 3. XL C PDF 文件

文档标题	PDF 文件名	描述
《IBM XL C for AIX V13.1.3 安装指南》，S151-2066-02	install.pdf	包含有关安装 XL C 和配置环境以执行基本的编译和程序的信息。
《IBM XL C for AIX V13.1.3 入门》，S151-2061-02	getstart.pdf	包含 XL 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。
IBM XL C for AIX V13.1.3 Compiler Reference, SC27-4239-02	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。
IBM XL C for AIX V13.1.3 语言参考, SC27-4240-02	langref.pdf	包含有关 IBM 支持的 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。
IBM XL C for AIX V13.1.3 Optimization and Programming Guide, SC27-4241-02	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 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>。

可以在 Web（网址为 <http://www.ibm.com/support/docview.wss?uid=swg27036590>）上获得与 XL C 相关的更多信息（包括 IBM Redbooks® 出版物、白皮书和其他文章）。

有关 C/C++ 的更多信息，请参阅 C/C++ café，网址为 <https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=5894415f-be62-4bc0-81c5-3956e82276f3>。

标准和规范

XL C 旨在支持下列标准和规范。您可以查阅这些标准和规范以获取此信息中出现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 9899:2011*，也称为 C11。（部分支持）

- *AltiVec Technology Programming Interface Manual*, Motorola Inc. 此针对向量数据类型的规范用于支持向量处理技术，可以在 http://www.freescale.com/files/32bit/doc/ref_manual/ALTIVECPIM.pdf 中找到此规范。
- *Information Technology - Programming Languages - Extension for the programming language C to support decimal floating-point arithmetic, ISO/IEC WDTR 24732*. 此草稿技术报告已提交给 C++ 标准委员会，可以在 <http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1176.pdf> 中找到此报告。
- *ANSI/IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985*.
- *OpenMP Application Program Interface Version 3.1*（完全支持）、*OpenMP Application Program Interface Version 4.0*（部分支持）可以在 <http://www.openmp.org> 中找到

其他 IBM 信息

- *Parallel Environment for AIX: Operation and Use*
- IBM Systems 信息中心（位于 <http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/com.ibm.aix.doc/doc/base/aixparent.htm>）是 AIX 信息的资源。

您可以找到特定 AIX 系统的下列书籍：

- *AIX Commands Reference, Volumes 1 - 6*
- *Technical Reference: Base Operating System and Extensions, Volumes 1 & 2*
- *AIX National Language Support Guide and Reference*
- *AIX General Programming Concepts: Writing and Debugging Programs*
- *AIX Assembler 语言参考*

其他信息

- *Using the GNU Compiler Collection*，网址为 <http://gcc.gnu.org/onlinedocs>

技术支持

可以从 XL C 支持页面获得其他技术支持，网址为：http://www.ibm.com/support/entry/portal/product/rational/xl_c_for_aix。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。

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

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

如何发送您的意见

您的反馈对于帮助我们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 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。

请确保包含以下信息：资料的名称、资料的章节号、XL C 的版本以及您的意见文本的特定位置，例如页号或表号。

第 1 章 XL C 简介

IBM XL C for AIX V13.1.3 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 C++ 和 Fortran 程序进行的语言间调用。

本节包含有关较高级别的 XL C 编译器的功能的信息。它面向要对该编译器进行评测的人员以及需要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 for AIX V13.1.3 是较大的 IBM C、C++ 和 Fortran 编译器系列的一部分。XL C 与 XL C++ 和 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[®]/Q、IBM i、所选 Linux 分发版、IBM z/OS[®] 和 IBM z/VM[®]。符合国际编程语言标准的公共代码库有助于支持一致的编译器性能，简化多个操作系统和硬件平台之间代码的可移植性。

操作系统和硬件支持

本节描述了 IBM XL C for AIX V13.1.3 支持的操作系统和硬件。

IBM XL C for AIX V13.1.3 支持下列操作系统：

- AIX V6.1 TL 2 Service Pack 5 或更高版本
- AIX V7.1
- AIX V7.2
- IBM i V7.1 PASE V7.1
- IBM i V7.2 PASE V7.2

有关完整的需求列表，请参阅自述文件和《XL C 安装指南》中的“在安装 XL C 之前”。

编译器、编译器的库及编译器生成的对象程序将在具有所需软件和磁盘空间的 POWER5、POWER5+、POWER6[®]、POWER7[®]、POWER7+[™] 和 POWER8[®] 系统上运行。

为了利用各种受支持的硬件配置，编译器提供了选项以根据运行已编译的应用程序的硬件类型对应用程序的性能进行调整。

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 提供了几个可以用来调用编译器的不同命令，例如，**xlc**、**c99** 和 **c89**。提供编译器调用命令是为了支持大部分标准化的 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 **xlcr**。“_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 编译器调用命令的更多信息，请参阅 *XL 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。您可从对以下任务使用不同选项获得优势：

- 调试您的应用程序
- 优化和调优应用程序性能
- 选择语言级别和扩展以与其他 C 编译器支持的非标准功能和行为相兼容
- 执行将否则需要更改源代码的许多其他常见任务

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指定编译器选项。

有关 XL C 编译器选项的更多信息，请参阅 *XL 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纯文本编译器配置文件，该文件包含定义了编译器选项缺省设置的节。

如果您经常指定 XL C 的缺省设置以外的编译器选项设置，那么可使用 **makefile** 来定义设置。或者，您也可以创建定制配置文件来定义您自己经常使用的选项设置。

有关使用定制编译器配置文件的更多信息，请参阅第 27 页的『使用定制编译器配置文件』。

使用情况跟踪配置文件

使用情况和报告工具可以用于检测组织对编译器的使用是否超过了许可证权利。

编译器的使用情况跟踪和报告功能具有它自己的配置文件。主要的编译器配置文件包含指向此文件的条目。编译器产品的不同安装可以使用单个使用情况跟踪配置文件来集中管理使用情况跟踪和报告功能。

有关使用情况跟踪和报告功能的详细信息，请参阅 *XL C Compiler Reference* 中的 *Tracking and reporting compiler usage*。

语言标准一致性

本主题描述了 IBM XL C for AIX V13.1.3 支持的 C 编程语言规范。

- 部分支持 ISO/IEC 9899:2011（称为 C11）
- ISO/IEC 9899:1999（称为 C99）
- ISO/IEC 9899:1990（称为 C89）

除了标准语言级别之外，XL C 还支持以下语言扩展：

- 部分支持 OpenMP Application Program Interface V4.0
- OpenMP Application Program Interface V3.1
- 用于支持向量编程的语言扩展
- 一部分 GNU C 语言扩展

有关 C 语言规范和扩展的更多信息，请参阅《XL C 语言参考》中的语言级别和语言扩展。

与 GNU 的兼容性

XL C 支持一部分 GNU 编译器命令选项，以便于移植用 **gcc** 编译器开发的应用程序。

当 **gxlc** 调用命令与选择 GNU 编译器选项一起使用时，此支持可用。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 编译器选项。

调用命令使用纯文本配置文件来控制 GNU 至 XL C 选项映射和缺省值。可以定制此配置文件以满足您的独特编译需求。有关更多信息，请参阅 *XL C Compiler Reference* 中的 "Reusing GNU C compiler options with gxlc"。

源代码迁移和一致性检查

XL C 提供编译器调用命令，以指示编译器将应用程序代码编译到特定语言级别。

您也可以使用 **-qlanglvl** 编译器选项来指定语言级别。如果程序源中的语言或语言扩展元素不符合所指定的语言级别，那么编译器会发出诊断消息。

相关信息



-qlanglvl

库

XL C 提供一个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置函数组成。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 位编译方式并且它们的性能优于缺省的 **libm** 数学库例程。这些库是线程安全的，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 Optimization and Programming Guide* 中的 "Using the Mathematical Acceleration Subsystem"。

Basic Linear Algebra Subprograms

libxlopt 库中附带了高性能代数函数的 Basic Linear Algebra Subprograms (BLAS) 集合。可使用这些功能来:

- 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 Optimization and Programming Guide* 中的 Using the Basic Linear Algebra Subprograms。

其他库

XL C 还随附了以下库:

- 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 Optimization and Programming Guide* 中的 SMP Runtime Library。
- 内存调试运行时库用于诊断内存泄漏。请参阅 *XL C Optimization and Programming Guide* 中的 Using memory heaps。

有关库支持的更多信息, 请搜索 XL C 编译器支持页面, 网址为 http://www.ibm.com/support/entry/portal/product/rational/xl_c_for_aix。

工具、实用程序和命令

本主题介绍了 XL C 中提供的主要工具、实用程序和命令。本主题未涵盖所有编译器工具、实用程序和命令。

工具

IBM Debugger for AIX

IBM Debugger for AIX 可以帮助您检测并诊断以本地方式或远程方式运行的程序中的错误。可以通过设置已编译的特定于语言的断点、暂挂执行、单步执行代码以及检查和更改变量的内容来控制程序的执行。

调试器包含特定于给定编程语言的视图和功能。借助已编译的语言视图, 可以监视您正在调试的应用程序的变量、表达式、寄存器、内存和应用程序模块。

使用情况报告工具

使用情况报告工具生成一个报告来描述组织对编译器的使用情况。这些报告有助于确定组织对编译器的使用是否与编译器许可证权利匹配。**urt** 命令包含一些可以用于对报告进行定制的选项。有关更多信息, 请参阅 *XL C Compiler Reference* 中的 Tracking and reporting compiler usage。

实用程序

CreateExportList 实用程序

CreateExportList 实用程序将创建一个文件, 该文件包含所给定的一组对象文件中所有可导出的exportable符号的列表。有关更多信息, 请参阅 *XL C Optimization and Programming Guide* 中的 Exporting symbols with the CreateExportList utility。

gxlc 实用程序

gxlc 实用程序将 GNU C 调用命令转换为相应的 **xlc** 命令并调用 XL C 编译

器。该实用程序的目的是最大程度地减少对用 GNU 编译器构建的现有应用程序的 `makefile` 的更改并使得向 XL C 编译器的过渡更方便。有关更多信息，请参阅 *XL C Compiler Reference* 中的 *Reusing GNU C compiler options with gxc*。

命令

genhtml 命令

genhtml 命令将转换由 **-qlistfmt** 选项生成的现有 XML 诊断报告。您可以选择通过使用 **-qlistfmt** 选项来生成 XML 或 HTML 诊断报告。该报告可帮助您进行优化。有关如何使用此命令的更多信息，请参阅 *XL C Compiler Reference* 中的 **genhtml** command。

与概要文件定向反馈 (PDF) 相关的命令

cleanpdf 命令

cleanpdf 命令将从写入概要文件定向反馈数据的目录中除去所有 PDF 文件或指定的 PDF 文件。

mergepdf 命令

mergepdf 命令提供了在将两个或更多 PDF 记录合并为单个记录时对这些记录的重要性进行权衡的能力。PDF 记录必须从相同的可执行文件派生。

showpdf 命令

showpdf 命令将显示 PDF 运行期间执行的所有过程（在 **-qpdf1** 选项下进行编译）的下列类型的概要分析信息：

- 块计数器概要分析
- 调用计数器概要分析
- 值概要分析
- 高速缓存不命中概要分析（如果在 **-qpdf1** 阶段中指定了 **-qpdf1=level=2** 选项）。

可以使用文本或 XML 格式来查看前两种类型的概要分析信息。但是，只能使用 XML 格式来查看值概要分析和高速缓存不命中概要分析信息。

有关更多信息，请参阅 *XL C Compiler Reference* 中的 **-qpdf1**、**-qpdf2**。

xlcndi **xlcndi** 脚本将 XL C 安装到非缺省目录位置。有关更多信息，请参阅《*XL C 安装指南*》中的“使用 **xlcndi** 来更新高级安装”。

程序优化

XL C 提供了几个可以帮助您控制程序优化和性能的编译器选项。

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。XL C 提供了一些针对各种受支持硬件的优化变换的组合。这些变换提供下列优点：

- 减少为关键操作执行的指令数
- 重构生成的对象代码，以最优方式利用 Power Architecture® 处理器
- 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

相关信息



优化应用程序



优化和调优



编译器内置函数

64 位对象能力

XL 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。

AIX 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。绑定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。不允许使用以下方案，将无法装入和/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
- 显式地尝试装入 32 位模块的 64 位可执行文件
- 显式地尝试装入 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 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 Optimization and Programming Guide* 中的“Using 32-bit and 64-bit modes”。

共享内存并行化

XL 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 来开发并行化应用程序：

- 基于伪指令的共享内存并行化（OpenMP 或 SMP）
- 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

并行编程工具都以线程概念为基础。并行编程充分利用多处理器系统的优势，同时保持与现有单处理器系统的完全二进制兼容性。这意味着在单处理器系统上运行的多线程程序可以利用多处理器系统而无需重新编译。

有关更多信息，请参阅 *XL C Optimization and Programming Guide* 中的 *Parallelizing your program*。

OpenMP 伪指令

OpenMP 伪指令是一组基于 API 的命令，它们受 XL 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的代码块。如果源代码中存在这些伪指令，那么编译器就无需对并行代码执行任何相关性分析。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 伪指令解决了应用程序并行化方面的以下重要问题：

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，每个线程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代码并行区域中包含的工作分布在各个线程上。
3. 伪指令可用来控制线程之间的同步。

IBM XL C for AIX V13.1 支持 OpenMP API V4.0 规范。有关详细信息，请参阅 第 14 页的『OpenMP 4.0』。

相关信息



优化应用程序



并行编程的 OpenMP API 规范

诊断报告

编译器列表、XML 报告和 HTML 报告提供了一些重要信息，以帮助您更有效地开发和调试应用程序。

列表信息已整理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 Compiler Reference* 中的编译器消息和列表。

您还可以从编译器中获取 XML 或 HTML 格式的诊断信息。XML 和 HTML 报告提供有关编译器可以执行或无法执行的优化的信息。可使用此信息来在调整应用程序时减少编程工作量，对于高性能应用程序更是如此。该报告通过 XML 模式进行定义，由您创建的用于读取和分析结果的工具能够轻易使用该报告。有关此报告以及如何进行使用的详细信息，请参阅 *XL C Optimization and Programming Guide* 中的 "Using reports to diagnose optimization opportunities"。

符号调试器支持

您可使用 **-g** 编辑器选项的不同级别来指示 XL C 将调试信息包含在已编译的对象中。

调试信息可以由 **dbx**、IBM Debugger for AIX、 或任何其他支持 AIX XCOFF 可执行文件格式 上受支持的符号调试器检查，以便帮助您调试程序。

相关信息



第 2 章 迁移应用程序

本部分列出了在迁移那些使用其他版本的 XL C 编译的应用程序时的重要注意事项。

迁移使用事务型内存内置函数的应用程序

从 IBM XL C for AIX V13.1.2 开始，要使用事务型内存内置函数，必须在源代码中包含一个头文件。此外，如果使用了事务开始和结束内置函数的数字返回值，那么必须使用 IBM XL C for AIX V13.1.3 提供的宏返回值代替这些数字返回值。

事务型内存内置函数所需的新头文件

如果使用了任何事务型内存内置函数，那么必须在源代码中包含 `htmxlintrin.h` 文件。

更改了事务开始和结束内置函数的返回值

事务开始和结束内置函数的返回值不再是数字。您必须使用以下返回值来更新自己的程序：

`__TM_begin`

如果成功执行，那么此函数将返回 `_HTM_TBEGIN_STARTED`；否则，将返回其他值。

`__TM_end`

如果在指令开始执行前线程处于事务状态，那么此函数将返回 `_HTM_TBEGIN_STARTED`；否则，将返回其他值。

`__TM_simple_begin`

如果成功执行，那么此函数将返回 `_HTM_TBEGIN_STARTED`；否则，将返回其他值。

相关信息

事务型内存内置函数



事务型内存内置函数

混合通过不同编译器编译的对象文件

使用不同编译器编译的大多数对象文件都可以链接在一起。但是，在某些情况下，对象文件不兼容，必须重新进行编译。

请注意以下限制：

- AIX、Linux 大尾数法分发版和 Linux 小尾数法分发版之间不存在二进制兼容性。
- 请勿将使用大尾数法编译器编译的对象文件与使用小尾数法编译器编译的对象文件混合在一起。
- 如果在编译期间使用了 `-qipa` 选项，请勿将使用不同编译器版本编译的对象文件和库文件混合在一起。`-qipa` 选项会指示编译器对这些对象文件和库文件执行 IPA 链接。IPA 链接可能无法处理不匹配的版本。

相关信息:



-qipa

第 3 章 早期发行版中新增的增强功能

本部分描述了早期发行版中新增的增强功能。这些增强功能同样适用于当前发行版。

V13.1.2 中新增的增强功能

本部分描述了 IBM XL C for AIX V13.1.2 中新增的功能和增强功能。这些功能和增强功能同样适用于更高发行版。

内置函数

本部分描述了 IBM XL C for AIX V13.1.2 新增的或已更改的内置函数。

新增的内置函数

vec_mergee

合并两个向量的偶数编号元素的值。

vec_mergeo

合并两个向量的奇数编号元素的值。

vec_revb

返回一个向量，该向量包含该参数的对应元素的字节（采用逆向字节顺序）。

vec_reve

返回一个向量，该向量包含该参数的元素（采用逆向元素顺序）。

vec_xl(a, b)

从由位移 *a* 和指针 *b* 指定的内存地址中装入 16 字节向量。

vec_xl_be(a, b)

从由位移 *a* 和指针 *b* 指定的内存地址中装入 16 字节向量。

vec_xst(a, b,c)

将 16 字节向量 *a* 的元素存储到有效的地址，通过将 *b* 中提供的位移加到 *c* 所提供的地址上来获取此有效地址。此有效地址不会截断至 16 字节的倍数。

vec_xst_be(a, b,c)

将 16 字节向量 *a* 的元素以大尾数法元素顺序存储到有效地址中，通过将 *b* 中提供的位移加到 *c* 所提供的地址上来获取此有效地址。此有效地址不会截断至 16 字节的倍数。

相关信息



编译器内置函数

命令

本部分描述了新的、已更改的或已除去的编译器命令。

resetpdf

已除去此命令。建议您使用 **cleanpdf** 命令代替。**resetpdf** 命令的行为与 **cleanpdf** 命令的行为相同。有关更多信息，请参阅“XL C Compiler Reference”中的 -qpdf1 和 -qpdf2。

编译器选项

本部分描述了新的或已更改的编译器选项。

-qfloat

新增了下列子选项:

subnormals

此子选项向编译器断言代码使用非正规浮点值（也称为非规范化浮点值）。

nosubnormals

此子选项向编译器断言代码未使用非正规浮点值（也称为非规范化浮点值）。

不论是否指定此子选项，您程序的行为都不会改变，但编译器会使用此信息来尽可能地提高性能。这些子选项仅在 POWER8 处理器上生效。要使用 **-qfloat=subnormals** 或 **-qfloat=nosubnormals**，还必须指定 **-qarch=pwr8** 和 **-qtune=pwr8** 选项。

-qstrict=guards

指定 **-qstrict=guards** 选项时由 XL C 执行的操作数量已增多。当 **-qstrict=guards** 选项生效时，编译器的行为如下:

- 编译器不会使操作越过安全机制。
- 如果编译器遇到 if 语句，并且此类语句包含可在编译时解决的指针回绕检查，那么编译器将不会除去这类检查或相关操作。

V13.1 中新增的增强功能

本部分描述了 V13.1 中编译器新增的功能和增强功能。这些功能和增强功能同样适用于更高版本。

支持 POWER8 处理器

XL C for AIX V13.1 支持 POWER8 处理器。

为了支持 POWER8 处理器而引入的新功能和增强功能分为以下几个类别:

- POWER8 处理器的 MASS 库
- POWER8 处理器的编译器选项
- POWER8 处理器的内置函数

POWER8 处理器的 Mathematical Acceleration Subsystem (MASS) 库

向量库

向量 MASS 库 **libmassvp8.a** 包含已针对 POWER8 体系结构进行调整的向量函数。可以在 32 位方式或 64 位方式下使用这些函数。

有关向量库的更多信息，请参阅 *XL C Optimization and Programming Guide* 中的 Using the vector libraries。

SIMD 库

MASS SIMD 库 **libmass_simdp8.a** 包含一组已加速的常用数学内置函数，它们的性能优于相应的标准系统库函数。

有关 SIMD 库的更多信息，请参阅 *XL C Optimization and Programming Guide* 中的 Using the SIMD libraries。

POWER8 处理器的编译器选项

-qarch 编译器选项指定生成代码所针对的处理器体系结构。**-qtune** 编译器选项调整指令选择、调度和其他与体系结构有关的性能增强功能，以在特定的硬件体系结构中以最优状态运行。

新的 **-qarch=pwr8** 子选项将生成对象代码，该对象代码包含将在 POWER8 硬件平台上运行的指令。如果指定了新的 **-qtune=pwr8** 子选项，那么将针对 POWER8 硬件平台对优化进行调整。

有关更多信息，请参阅 *XL C Compiler Reference* 中的 **-qarch** 和 *XL C Compiler Reference* 中的 **-qtune**。

POWER8 处理器的内置函数

添加了新硬件内置函数，以支持下列 POWER8 处理器功能：

- 用于向量处理的 POWER8 函数
- POWER8 二进制编码的十进制函数
- POWER8 密码术函数
- POWER8 四字四则运算函数
- POWER8 负荷和保留/存储条件指令
- POWER8 高速缓存和数据预取控制函数
- POWER8 事务型内存函数
- POWER8 预取函数

有关 XL C 提供的内置函数的更多信息，请参阅 *XL C Compiler Reference* 中的 Compiler built-in functions。

C11 功能

除了现有 C11 功能之外，此发行版的 XL C 中还支持新增 C11 功能。

注：IBM 支持 C11（批准之前又称为 C1X）的所选功能。IBM 将继续开发和实现此标准的功能。此语言级别的实现基于 IBM 对该标准的解释。在 IBM 实现所有 C11 功能（包括支持新的 C11 标准库）之前，该实现可能会随着发行版的不同而改变。IBM 在实现 C11 功能时，不会尝试在源代码、二进制代码或列表以及其他编译器接口方面保持与先前发行版的兼容性。

AIX V13.1 中引入了下列功能：

- typedef 重新声明

- 类属选择

typedef 重新声明

通过使用 typedef 重新声明，您可以重新定义同一作用域中先前的 typedef 名称，以引用同一类型。

 XL C 编译器支持所有类型，包括便于修改的类型。

 有关更多信息，请参阅《XL C 语言参考》中的“typedef 定义”。

类属选择

类属选择提供了一个机制，以在编译时根据给定类型名称选择表达式。一般用法是定义类型类属宏。有关更多信息，请参阅 Generic selection (C11)。

OpenMP 4.0

XL C for AIX V13.1 部分支持 OpenMP Application Program Interface V4.0 规范。XL C 实现基于 IBM 对 OpenMP Application Program Interface V4.0 的解释。

此版本的 XL C 支持以下 OpenMP 4.0 功能：

- update 和 capture 子句增强功能
- OMP_DISPLAY_ENV 环境变量

update 和 capture 子句增强功能

已扩展 atomic 构造的 update 和 capture 子句以支持更多表达式形式。

OMP_DISPLAY_ENV 环境变量

可使用环境变量 OMP_DISPLAY_ENV 来显示与环境变量关联的内部控制变量 (ICV) 的值以及有关运行时库的特定于构建的信息。

相关信息

- *XL C Compiler Reference* 中的 "OpenMP environment variables"
- *XL C Compiler Reference* 中的 "Pragma directives for parallel processing"
- The OpenMP API specification for parallel programming

内置函数

以下主要类别的内置函数 是此发行版新增的。

注：仅当设置或隐含了 **-qarch=pwr8** 时，POWER8 内置函数才有效。

用于向量处理的 POWER8 内置函数

增加了以下向量内置函数：

- 向量 gather-bits-by-bytes 双字函数
 - vec_gbb
- 向量计数前导零函数
 - vec_cntlz
- 向量填充计数函数
 - vec_popcnt

- 扩展的向量逻辑运算函数
 - `vec_eqv`
 - `vec_nand`
 - `vec_orc`
- 128 位整数增减函数
 - `vec_add_u128`
 - `vec_sub_u128`
 - `vec_adde_u128`
 - `vec_sube_u128`
 - `vec_addc_u128`
 - `vec_subc_u128`
 - `vec_addec_u128`
 - `vec_subec_u128`
 - `vec_bperm`

已扩展以下内置函数以支持双字类型:

- 向量压缩函数
 - `vec_pack`
 - `vec_packs`
 - `vec_packsu`
- 向量解压缩函数
 - `vec_unpackh`
 - `vec_unpackl`
- 向量增减函数
 - `vec_add`
 - `vec_sub`
- 向量最大和最小函数
 - `vec_max`
 - `vec_min`
- 向量切换和旋转函数
 - `vec_rl`
 - `vec_sl`
 - `vec_sr`
 - `vec_sra`
- 向量比较函数
 - `vec_cmpeq`
 - `vec_cmpgt`
 - `vec_cmpge`
 - `vec_cmplt`
 - `vec_cmple`

二进制编码的十进制内置函数

添加了下列内置函数以支持二进制编码的十进制 (BCD) 四则运算和比较。前三种类型的内置函数是 POWER8 内置函数。当 **-qarch** 设置为目标 POWER8 或 POWER7 处理器时, BCD 负荷函数和存储函数有效:

- BCD 增减函数
 - `__bcdadd`
 - `__bcdsub`
- 针对溢出函数的 BCD 测试增减
 - `__bcdadd_ofl`
 - `__bcdsub_ofl`
 - `__bcd_invalid`
- BCD 比较函数
 - `__bcdcmpeq`
 - `__bcdcmpgt`
 - `__bcdcmpge`
 - `__bcdcmplt`
 - `__bcdcmple`
- BCD 负荷函数和存储函数
 - `__vec_ldrmb`
 - `__vec_strmb`

POWER8 密码术内置函数

已提供以下内置函数以执行密码操作:

- 高级加密标准 (AES) functions
 - `__vcipher`
 - `__vcipherlast`
 - `__vncipher`
 - `__vncipherlast`
 - `__vsbox`
- 安全散列算法 (SHA) 函数
 - `__vshasigmad`
 - `__vshasigmaw`
- 其他函数
 - `__vpmsumb`
 - `__vpmsumh`
 - `__vpmsumw`
 - `__vpmsumd`
 - `__vpermxor`

POWER8 非向量内置函数

已添加以下内置函数以改进高速缓存的效率:

- `__dcbtna`
- `__icbt`

已使用以下函数扩展负荷和存储内置函数以支持更多类型:

- `__lqarx`
- `__lharx`
- `__lbarx`
- `__stqcx`
- `__sthcx`
- `__stbcx`

POWER8 事务型内存内置函数

事务型内存是用于并行编程的模型。在此模型中，您可以指定要自动处理的指示信息或语句的块操作。

可使用以下内置函数来标记事务的开始或结束，并对故障原因进行诊断:

- 事务开始和结束函数
 - `__TM_begin`
 - `__TM_end`
 - `__TM_simple_begin`
- 事务异常中止函数
 - `__TM_abort`
 - `__TM_named_abort`
- 事务查询函数
 - `__TM_failure_address`
 - `__TM_failure_code`
 - `__TM_is_conflict`
 - `__TM_is_failure_persistent`
 - `__TM_is_footprint_exceeded`
 - `__TM_is_illegal`
 - `__TM_is_named_user_abort`
 - `__TM_is_nested_too_deep`
 - `__TM_is_user_abort`
 - `__TM_nesting_depth`

POWER8 预取内置函数

以下内置函数采用直觉、可移植和优化友好方式显示 Data Stream Control Register (DSCR) 问题状态控制:

- 瞬态属性启用函数

- __hardware_transient_enable
- __load_transient_enable
- __software_transient_enable
- __store_transient_enable
- 单元计数启用和设置函数
 - __hardware_unit_count_enable
 - __software_unit_count_enable
 - __set_prefetch_unit_count
- 预取深度函数
 - __default_prefetch_depth
 - __depth_attainment_urgency
- 负荷流启用和禁用函数
 - __load_stream_disable
 - __stride_n_stream_enable
- DSCR 函数
 - __prefetch_get_dscr_register
 - __prefetch_set_dscr_register

相关信息:

- *XL C Compiler Reference* 中的 Compiler built-in functions
- -qarch

编译器选项和编译指示伪指令

本节描述新增的或已更改的编译器选项和编译指示伪指令。

可以在命令行上指定编译器选项。还可以通过应用程序源文件中嵌入的编译指示伪指令来修改编译器行为。有关 XL C 编译器选项的详细描述和用法信息，请参阅 *XL C Compiler Reference*。

-qarch

该选项缺省值已更新为 **pwr4**。表示旧硬件系列的子选项将以静默方式升级到更新的体系结构。

增加或更新了下列子选项:

-qarch=pwr7

此子选项将生成对象代码，该对象代码包含将在 POWER7、POWER7+ 或 POWER8 硬件平台上运行的指令。

-qarch=pwr8

此子选项将生成对象代码，该对象代码包含将在 POWER8 硬件平台上运行的指令。

-qcheck

增加或更新了下列子选项:

-qcheck=stackclobber

此子选项将检测您的程序中某种类型的堆栈溢出。

-qcheck=unset

此子选项将检查在运行时设置自动变量之前使用的这些变量。

-qdbgfmt=dwarf4

此子选项将使用 DWARF 4 格式生成调试信息。

-qhelp 此选项将显示编译器的联机帮助页。**-qinfo**

编译器不会针对下列文件发出参考消息：

- 编译器的标准搜索路径中的文件和系统头文件。
- 编译器的标准搜索路径中的文件最终包括的文件和系统头文件。

增加或更新了下列子选项：

-qinfo=mt

此子选项通知您可能需要同步的位置。

-qinfo=unset

此子选项将检测设置自动变量之前使用的这些变量，并使用编译时的参考消息对其进行标记。

-qpdf1=unique

此子选项将在运行时期为每个进程创建唯一的 PDF 文件。

-qprefetch=dscr

此子选项有助于改进应用程序的运行时性能。可根据系统体系结构指定 dscr 的值。

-qsimd=auto

此子选项控制自动，该操作由不推荐使用的 **-qhot=simd** 选项执行。

-qtune

已更新选项缺省值。

增加或更新了下列子选项：

-qtune=pwr7

此子选项指定针对 POWER7 或 POWER7+ 硬件平台对优化进行调整。

-qtune=pwr8

此子选项指定针对 POWER8 硬件平台对优化进行调整。

SMT 子选项

新的 **-qtune** 同步多线程技术 (SMT) 子选项允许您指定目标 SMT 来为在该方式下获得最佳性能而定向优化。

-qunroll=*n*

此子选项提示编译器由于因素 *n* 而展开循环。如果循环次数少于 *n* 个迭代，那么将完全对其展开循环。

-qvisibility

此选项指定实体的可视性属性。实体可视性属性描述了在一个模块中定义的实体是否可在以及如何在其他模块中引用或使用。可视性属性只影响具有外部链接的实体，并且不会增加其他实体的可视性。

新增或已更改的编译指示伪指令

#pragma GCC visibility push, #pragma GCC visibility pop

这一对编译指示伪指令是等效于 **-qvisibility** 选项的编译指示。编译指示伪指令用于指定外部链接符号的可视性属性。

性能和优化

其他功能和增强功能可以帮助进行性能调整和应用程序优化。

IBM

实体的可视性属性

实体可视性属性描述了在一个模块中定义的实体是否可在以及如何在其他模块中引用或使用。通过对实体使用可视性属性，可以获得以下好处：

- 降低共享库的大小
- 减少符号冲突的可能性
- 允许进一步对编译和链接阶段进行优化
- 改进动态链接的效率

有关更多信息，请参阅 *XL C Optimization and Programming Guide* 中的 *Using visibility attributes (IBM extension)*。

IBM

有关性能调整和程序优化的更多信息，请参阅 *XL C Optimization and Programming Guide* 中的 *Optimizing your applications* 和 *Coding your application to improve performance*。

V12.1 中新增的增强功能

本部分描述了 V12.1 中编译器新增的功能和增强功能。这些功能和增强功能同样适用于更高版本。

C11 功能

XL C V12.1 引入了对 C11 的选定功能的支持，这是新增的 C 编程语言标准。

注：IBM 支持 C11（批准之前又称为 C1X）的所选功能。IBM 将继续开发和实现此标准的功能。此语言级别的实现基于 IBM 对该标准的解释。在 IBM 实现所有 C11 功能（包括支持新的 C11 标准库）之前，该实现可能会随着发行版的不同而改变。IBM 在实现 C11 功能时，不会尝试在源代码、二进制代码或列表以及其他编译器接口方面保持与先前发行版的兼容性。

XL C V12.1 中引入了下列 C11 功能：

- 匿名结构
- 复杂类型初始化
- 新的语言级别 - **extc1x**
- **_Noreturn** 函数说明符

- 静态断言

匿名结构

此功能允许在 **extc1x** 语言级别声明匿名结构。有关更多信息，请参阅《XL C 语言参考》中的“匿名结构”。

复杂类型初始化

将在标准头文件 `complex.h` 中定义宏 `CMPLX`、`CMPLXF` 和 `CMPLXL`，以允许在 **extc1x** 语言级别对复杂类型进行初始化。有关更多信息，请参阅《XL C 语言参考》中的“复杂类型的初始化 (C11)”。

新的语言级别 - extc1x

在此发行版中，新的子选项已添加至 **-qlanglvl** 选项。可以使用 **-qlanglvl=extc1x** 来启用 XL C 当前所支持的 C11 功能。

_Noreturn 函数说明符

`_Noreturn` 函数说明符用于声明不会将函数返回至其调用者。您可以使用此函数说明符来定义自己的不进行返回的函数。编译器可以通过忽略该函数返回时所出现的情况来生成更好的代码。有关更多信息，请参阅《XL C 语言参考》中的“`_Noreturn` 函数说明符”。

静态断言

在 C 语言中增加静态断言具有下列优点：

- 库可以在编译时检测一般用法错误。
- C 标准库的实现可以检测和诊断一般用法错误，从而提高可用性。

可以声明静态断言以在编译时检查重要的程序不变量。

有关更多信息，请参阅《XL C 语言参考》中的“`_Static_assert` 声明 (C11)”。

OpenMP 3.1

XL C V12.1 支持 OpenMP Application Program Interface V3.1 规范。XL C 实现基于 IBM 对 OpenMP Application Program Interface V3.1 的解释。

OpenMP 3.1 包括下列对 OpenMP 3.0 的更新：

- 将 `final` 和 `mergeable` 子句添加到 `task` 构造以支持优化。
- 添加 `taskyield` 构造以允许用户指定程序可以在何处执行任务切换。
- 添加 `omp_in_final` 运行时库函数以支持最终任务区域的特例化。
- 扩展 `atomic` 构造以包括 `read`、`write` 和 `capture` 格式；添加 `update` 子句以应用 `atomic` 构造的现有格式。
- 添加两个归约运算符： `min` 和 `max`。
- 允许在 `firstprivate` 子句中指定用常量限定的类型。
- 添加 `OMP_PROC_BIND` 环境变量以控制是否允许 OpenMP 线程在处理器之间移动。
- 扩展 `OMP_NUM_THREADS` 环境变量以指定要用于嵌套并行区域的线程数。

相关信息

- *XL C Compiler Reference* 中的 "OpenMP environment variables"
- *XL C Compiler Reference* 中的 "Pragma directives for parallel processing"
- www.openmp.org

性能和优化

XL C V12.1 中的其他功能和增强功能可以帮助进行性能调整 and 应用程序优化。

有关编译器优化的报告

列表报告中提供了许多增强功能，以便为您提供有关编译器如何优化代码的更多信息。可以使用此信息来从编译器的优化功能中获取更多的优点。有关这些所增强报告的更多详细信息，请参阅『诊断报告』。

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

诊断报告

XL C V12.1 中添加的新诊断报告可以帮助您确定改进代码性能的时机。

HTML 格式的编译器报告

此时，可能会获取有关下列内容的 XML 或 HTML 格式的信息：编译器之前能够执行的优化以及未执行的优化。此信息可用来在调整应用程序时减少编程工作量，对于高性能应用程序更是如此。

可以使用 **-qlistfmt** 选项及其相关联的子选项来生成 XML 或 HTML 报告。缺省情况下，现在此选项在您未指定内容的类型时生成所有可用内容。

要查看已生成的 XML 报告的 HTML 版本，现在可以使用 **genhtml** 工具。有关如何使用此工具的更多信息，请参阅 *XL C Compiler Reference* 中的 **genhtml** command。

有关此报告以及如何进行使用的详细信息，请参阅 *XL C Optimization and Programming Guide* 中的 "Using reports to diagnose optimization opportunities"。

对概要分析报告的增强功能

对列表文件增加了新节以帮助对程序进行分析。将 **-qreport** 选项与 **-qpdf2** 选项配合使用时，可以将下列各节添加至列表文件中标题为 PDF Report 的节中：

概要分析数据的相关性

本节显示 **-qpdf1** 阶段中概要分析数据与源代码的相关性。相关性由范围 0 到 100 中的数字指示。该数字越大，概要分析数据与源代码的相关性就越高，并且使用概要分析数据可以获得的性能增益更多。

缺少概要分析数据

本节可能包括有关缺少概要分析数据的警告消息。将对编译器找不到其概要分析数据的每个函数发出该警告消息。

过时的概要分析数据

本节可能包括有关过时的概要分析数据的警告消息。编译器将对 **-qpdf1** 阶段之

后进行了修改的每个函数发出该警告消息。当优化级别从 **-qpdf1** 阶段更改为 **-qpdf2** 阶段时，也会发出该警告消息。

有关概要文件定向反馈的详细信息，请参阅 *XL C Optimization and Programming Guide* 中的 Using profile-directed feedback。

有关列表文件的其他信息，请参阅 *XL C Compiler Reference* 中的 Compiler listings。

对 **showpdf** 报告的增强功能

除了当前提供的块计数器和调用计数器概要分析信息之外，您还可以使用 **showpdf** 实用程序来查看高速缓存不命中概要分析和值概要分析信息。只能使用 XML 格式来显示值概要分析和高速缓存不命中概要分析信息。然而，可以使用文本或 XML 格式来显示所有其他类型的概要分析信息。在此发行版中，概要文件定向反馈 (PDF) 信息保存在两个文件中。一个文件是 **-qpdf1** 阶段期间生成的 PDF 映射文件，另一个文件是执行所产生的应用程序期间生成的 PDF 文件。可以运行 **showpdf** 实用程序来显示这两个文件中包含的 PDF 信息。有关更多信息，请参阅 *XL C Optimization and Programming Guide* 中的 Viewing profiling information with showpdf。

新增和增强的诊断选项

下表中的条目描述新增或已更改的编译器选项和伪指令，它们使您可以对编译器列表进行控制。

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 Compiler Reference* 中的 "Listings, messages and compiler information"。

表 4. 与列表相关的编译器选项和伪指令

选项/伪指令	描述
-qlistfmt	<p>增强了 -qlistfmt 选项，以生成 HTML 报告以及 XML 报告，这些包含有关下列内容的信息：编译器执行的优化以及未执行的优化。</p> <p>此选项的缺省行为已更改。现在，如果您未指定内容的特定类型，那么该选项将生成所有可用内容，而非不生成任何内容。</p>

内置函数

本节描述自 XL C 以来 V12.1 新增的内置函数。

GCC 原子内存访问内置函数 (IBM 扩展)

此发行版中新增了用于进行原子内存访问的 XL C 内置函数，这些函数的行为对应于 GNU 编译器集合 (GCC) 所提供的函数的行为。在具有多个线程的程序中，您可以使用这些函数以原子方式安全地修改一个线程中的数据，而不会与其他线程发生冲突。

有关 XL C 提供的内置函数的更多信息，请参阅 *XL C Compiler Reference* 中的 Compiler built-in functions。

编译器选项和编译指示伪指令

本节描述 V12.1 中新增或已更改的编译器选项和编译指示伪指令。

可以在命令行上指定编译器选项。还可以通过应用程序源文件中嵌入的编译指示伪指令来修改编译器行为。请参阅 *XL 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。

新增或已更改的编译器选项

-g 已将 **-g** 选项扩张为具有新级别，以改进对已优化的程序的调试。

-qhaltormsg

以前仅受 C++ 编译器支持的 **-qhaltormsg** 选项现在受 XL C 支持。如果生成了指定的错误消息，那么此选项将在生成任何对象文件、可执行文件或汇编程序源文件之前停止编译。还增加了负数形式的 **-qnohaltormsg**。

-qinclude

增加了否定形式的 **-qnoinclude**，以忽略先前指定的 **-qinclude** 选项。

-qinfo **-qinfo=all** 现在对除了 **als** 和 **ppt** 之外的所有组启用所有诊断消息。

-qlanglvl

增加或更新了下列子选项：

-qlanglvl=extc1x

此子选项启用当前受支持的所有 C11 功能以及其他特定于实现的语言扩展。

-qlanglvl=textafterendif

此子选项禁止显示当您代码从允许 **#endif** 或 **#else** 后面存在额外文本的编译器移植到 IBM XL C 编译器时发出的警告消息。

有关 C11 功能的更多信息，请参阅第 20 页的『C11 功能』。

-qlistfmt

增强了 **-qlistfmt** 选项以生成 HTML 报告以及 XML 报告，这些报告包含有关下列内容的信息：编译器执行的优化以及未执行的优化。

-qlistfmt 的缺省行为已更改。在此发行版中，如果您未指定内容的特定类型，那么该选项将生成所有可用内容，而非不生成任何内容。

-qoptfile

新的选项 **-qoptfile** 指定一个文件，该文件包含要用于编译的其他命令行选项的列表。

-qpvc **-qpvc=large** 现在启用大型 TOC 访问并在目录大于 64 KB 时阻止 TOC 溢出情况出现。

-qshowpdf

缺省值已从 **-qnoshowpdf** 更改为 **-qshowpdf**。

新增或已更改的编译指示伪指令

#pragma ibm max_iterations

增加了 **max_iterations** 编译指示。它指定所选循环的循环迭代大概最大次数。

#pragma ibm min_iterations

增加了 **min_iterations** 编译指示。它指定所选循环的循环迭代大概最小次数。

#pragma simd_level

增加了 **simd_level** 编译指示。它控制各个循环的向量指令的编译器代码生成。

第 4 章 设置和定制 XL C

本部分描述了如何根据您的自身的需求来设置和定制编译器。

有关 XL C 的完整先决条件和安装信息，请参阅《XL C 安装指南》中的“在安装 XL C 之前”。

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。

可以进行下列选择来定制编译器设置：

- XL C 编译器安装进程会创建缺省编译器配置文件。可以直接修改此配置文件，以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，那么必须将所有修改重新应用于新安装的配置文件。
- 可以创建您自己的定制配置文件，用于覆盖或补充缺省配置文件。编译器可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。编译器更新（以后可能会影响缺省配置文件中的设置）不会影响定制配置文件中的设置。

相关信息



使用定制编译器配置文件

配置编译器使用情况跟踪和报告

除了编译器配置文件，对于使用情况跟踪和报告功能，还有一个单独的配置文件。缺省情况下，会禁用使用情况跟踪，但可以通过在此配置文件中修改条目来启用该功能。通过使用此文件，还可配置使用情况跟踪的其他各个方面。

虽然编译器配置文件不同于使用情况跟踪配置文件，但是它包含指定使用情况跟踪配置文件的位置的条目，以便编译器可找到此文件。

有关如何配置使用情况跟踪和报告功能的更多信息，请参阅 *XL C Compiler Reference* 中的 *Tracking and reporting compiler usage*。

第 5 章 使用 XL C 开发应用程序

C 应用程序开发就是编辑、编译、链接和运行过程的重复循环。缺省情况下，编译和链接将合并为一个步骤。

注意事项:

- 请确保已正确安装和配置 XL C，然后才能使用该编译器。有关更多信息，请参阅《XL C 安装指南》。
- 要了解有关编写 C 程序的信息，请参阅《XL C 语言参考》。

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。当每个编译组件运行时，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. 在预处理汇编（.s）文件和未预处理的汇编程序（.S）文件之后对这些文件进行汇编
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 源文件

要创建 C 源程序，您可以使用系统上提供的任何文本编辑器，如 **vi** 或 **emacs**。

源程序必须使用被认可的文件名后缀来保存。请参阅第 31 页的『XL C 输入和输出文件』，以获得 XL C 可识别的后缀列表。

要使 C 源程序成为有效的程序，它必须符合在《XL C 语言参考》中指定的语言定义。

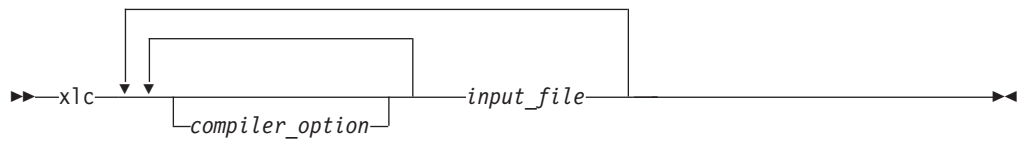
使用 XL C 进行编译

XL C 是命令行编译器。您可以根据特定 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 源文件或预处理的文件（.i 或 .ii），汇编任何 .s 和 .S 文件，并将对象文件和库链接到可执行程序。

要编译 C 源程序，请使用以下基本调用语法：



对于大多数应用程序，请使用 **xlc** 或线程安全等效项进行编译。

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 语言的不同级别和扩展提供显式编译支持。有关可用编译器调用命令（包括旨在帮助开发者从 GNU 编译环境迁移至 XL C 的特殊调用）的更多信息，请参阅“XL C Compiler Reference”中的“Invoking the compiler”。

编译并行化 XL C 应用程序

XL C 提供线程安全编译器调用命令以编译要用于多处理器环境中的并行应用程序。

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。一般 XL C 线程安全编译器调用如下所示：

- xlc_r
- xlc_r7
- xlc128_r
- xlc128_r7

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

注：单独使用这些命令中的任何一个并不表示并行化。为使编译器能够识别 SMP 或 OpenMP 伪指令并激活并行化，您还必须指定 **-qsmp** 编译器选项。相应地，在使用线程安全调用时应当仅指定 **-qsmp** 选项。当您指定 **-qsmp** 时，驱动程序将链接配置文件的活动节中 smp 库行上指定的库。

有关并行应用程序的更多信息，请参阅“*XL 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。

您可以采用以下方法的其中一种或任何组合指定编译器选项：

- 通过命令行
- 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

还可以将选项传递至链接程序、汇编程序和预处理器。

编译器选项的优先级顺序

当指定了多个编译器选项时，有可能会发生选项冲突和不兼容性问题。为了以一致的方式解决这些冲突，编译器会对大多数选项应用以下一般优先级顺序：

1. 源文件中的伪指令语句将覆盖命令行设置。
2. 命令行中的编译器选项设置将覆盖配置文件设置。
3. 配置文件设置将覆盖缺省设置。

通常，如果在调用编译器时在命令行中多次指定了同一编译器选项，那么最后指定的那个选项将生效。

注：某些编译器选项（例如，`-I` 选项）并不按照以上所述的优先级顺序。编译器在搜索命令行上用 `-I` 指定的目录之前，首先会搜索在 `xl.c.cfg` 文件中用 `-I` 指定的任何目录。`-I` 选项是累积的而不是优先的。

相关信息



编译器选项参考

通过 `gxlc` 来复用 GNU C 编译器选项

XL C 包括各种帮助您从 GNU C 编译器过渡到 XL C 的功能，其中包括 `gxlc` 命令。

`gxlc` 实用程序接受 GNU C 编译器选项并将它们转换成相当的 XL C 选项。此实用程序使用 XL C 选项来创建 `xlc` 调用命令，然后使用该命令来调用编译器。提供 `gxlc` 实用程序是为了帮助您复用为先前用 GNU C 开发的应用程序创建的 `makefile`。然而，为了充分利用 XL C 的功能，您可将 `xlc` 和 `cc` 调用命令与其相关联的选项一起使用。

`gxlc` 的操作由 `gxlc.cfg` 配置文件控制。此文件中显示具有 XL C 等效项的 GNU C 选项。并非每个 GNU 选项都具有对应的 XL C 选项。`gxlc` 实用程序为它不能转换的任何 GNU C 选项返回一个警告。

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

XL C 输入和输出文件

本主题描述了 XL C 可识别的文件类型。

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 Compiler Reference* 中的 *Types of input files* 和 *XL C Compiler Reference* 中的 *Types of output files*。

表 5. 输入文件类型

文件扩展名	描述
<code>.a</code>	归档或库文件
<code>.c</code>	C 源文件
<code>.i</code>	预处理的源文件
<code>.o</code>	对象文件
<code>.s</code>	汇编程序文件
<code>.S</code>	未预处理的汇编程序文件

表 5. 输入文件类型 (续)

文件扩展名	描述
.so	共享对象文件

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	适合包含在 <code>makefile</code> 中的目标文件
.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	共享对象文件
.u	生成依赖性文件

将已编译的应用程序与 **XL 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** 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，可以使用 `xlc` 来编译 `file1.c` 和 `file3.c`，以生成对象文件 `file1.o` 和 `file3.o`；之后，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。

```
xlc file1.c file2.o file3.c
```

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 (file2.o, file3.o)
xlc file1.o file2.o file3.o # Link object files with default libraries
```

相关信息



链接

重新链接现有的可执行文件

链接程序接受可执行文件作为输入，所以可以将现有的可执行文件与更新的对象文件链接在一起。

然而，您不能重新链接先前使用 **-qipa** 选项链接的可执行文件。

如果您有一个由几个源文件组成的程序，并且仅对其中某些源文件进行了局部更改，那么您不必再次编译每个文件。而是可以在编译更改的文件时，包括该可执行文件作为最新的输入文件：

```
xlc -omansion front_door.c entry_hall.c parlor.c sitting_room.c \
    master_bath.c kitchen.c dining_room.c pantry.c utility_room.c
```

```
vi kitchen.c # Fix problem in OVEN function
```

```
xlC -o newmansion kitchen.c mansion
```

限制第二次要编译和链接的文件数可以减少编译时间、磁盘活动和内存使用量。

注：除非您执行链接很有经验，否则应该避免此类链接操作。如果未正确操作，那么可能导致接口错误和其他问题。如果确实遇到问题，那么使用 **-qextchk** 编译器选项进行编译可以帮助您诊断链接问题。

动态和静态链接

您可以使用 **XL C** 以利用操作系统工具来执行动态和静态链接。

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。通过使用动态链接，您可以升级共享库中的例程而不需要重新链接。该链接形式是缺省值，因此无需任何其他选项。

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** 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。

注：如果分别在操作系统的不同级别上编译和运行动态和静态链接程序，那么这些程序可能不生效。

运行已编译的应用程序

在对程序进行编译和链接之后，可以在命令行上运行生成的可执行文件。

由 **XL 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。

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。

要运行程序，请在命令行上输入程序可执行文件的名称以及运行时参数。

取消执行

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

要取消正在运行的程序，请在程序处于前台时按 **Ctrl+C**。

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 编译器所创建的应用程序的行为。某些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。

有关环境变量以及它们在运行时如何影响应用程序的更多信息，请参阅《XL C 安装指南》。

在其他系统上运行已编译的应用程序

通常，在使用较早版本的 AIX 的系统上链接的应用程序将可以在更新版本的 AIX 上运行。但是，在使用较新版本的 AIX 的系统上链接的应用程序将不一定能够在较早版本的 AIX 上运行。

XL C 编译器诊断辅助

当编译应用程序遇到问题时，XL 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 Compiler Reference* 中的下列主题：

- 编译器消息和列表
- 错误检查和调试选项
- 列表、消息和编译器信息选项

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 编译的应用程序。

在编译时，您可使用 **-g** 或 **-qlinedebug** 选项以指示 XL C 编译器把调试信息包括在已编译的输出中。对于 **-g**，还可以使用其他级别来实现调试功能与编译器优化之间的平衡。有关调试选项的更多信息，请参阅 *XL C Compiler Reference* 中的 "Error checking and debugging"。

然后，可以使用 **dbx**、IBM Debugger for AIX、或其他任何支持 AIX XCOFF 可执行文件格式的符号调试器，以单步调试和检查已编译的应用程序的行为。

在调试应用程序时，已优化的应用程序会给您带来一些特殊的挑战。如果您需要调试已优化的应用程序，那么可以考虑将 **-g** 选项的 **-gN** 形式与任意优化选项结合使用。该形式的 **-g** 选项根据 N 的值在完全优化与完全调试支持之间提供不同的权衡级别。有关调试已优化的代码的更多信息，请参阅 "*XL C Optimization and Programming Guide*" 中的 "Debugging optimized code"。

确定使用哪一个级别的 XL C

要显示正在使用的 XL C 的版本和发行版级别，请使用 **-qversion** 编译器选项调用编译器。

例如，要获取详细的版本信息，请输入以下命令：

```
xlc -qversion=verbose
```

声明

编程接口：规划的编程接口允许客户编写程序，以获取 IBM XL C for AIX 的服务。

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, MD-NC119
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：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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。那些 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：

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

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。样本程序“按现状”提供，不附有任何种类的保证。对于因使用样本程序所引起的任何损害，IBM 概不负责。

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, 2015 年。

隐私策略注意事项：

IBM 软件产品，包括软件即服务解决方案（“软件产品”），可能使用 cookie 或其他技术收集产品使用信息，以帮助改善最终用户体验、定制最终用户互动或其他目的。在

许多情况下，软件产品不会收集任何个人可标识信息。我们的某些软件产品可以帮助您收集个人可标识信息。如果此软件产品使用 cookie 收集个人可标识信息，那么关于本产品使用 cookie 的具体信息作如下之规定。

此软件产品不使用 cookie 或其他技术收集个人可标识信息。

如果为此软件产品部署的配置为作为客户的您提供了通过 cookie 和其他技术从最终用户处收集个人身份信息的功能，那么您应自行寻求有关适用于此类数据收集的任何法律（包括对于声明和同意的任何要求）的法律意见。

有关将各种技术（包括 cookie）用于这些目的的更多信息，请参阅 IBM's Privacy Policy（网址为：<http://www.ibm.com/privacy>）和 IBM's Online Privacy Statement（网址为：<http://www.ibm.com/privacy/details> 中标题为“Cookies, Web Beacons and Other Technologies”的部分，以及『IBM Software Products and Software-as-a-Service Privacy Statement』（网址为：<http://www.ibm.com/software/info/product-privacy>）。

商标

IBM、IBM 徽标和 [ibm.com](http://www.ibm.com) 是 International Business Machines Corp. 在全球许多管辖区域注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。Web 站点“Copyright and trademark information” (<http://www.ibm.com/legal/copytrade.shtml>) 提供了 IBM 商标的最新列表。

Adobe 和 Adobe 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。

UNIX 是 Open Group 在美国和其他国家或地区的注册商标。

索引

[B]

- 编辑源文件 29
- 编译
 - 活动顺序 29
 - SMP 程序 30
- 编译器
 - 调用 29
 - 运行 29
- 编译器伪指令
 - 新增或已更改 12, 18, 24
- 编译器选项
 - 冲突和不兼容 31
 - 规范方法 30
 - 新增或已更改 12, 18, 24
- 并行化 6

[C]

- 程序
 - 运行 33

[D]

- 代码优化 5
- 调试 34
- 调试器支持 34
 - 符号 8
 - 输出列表 34
- 调试信息, 生成 34
- 调试已编译的应用程序 34
- 调用编译器 29
- 调用程序 33
- 调用命令 30
- 定制
 - 与 GNU 兼容 3
- 动态链接 33
- 对象文件 31
 - 创建 32
 - 链接 32
- 多处理器系统 6

[F]

- 符号调试器支持 8

[G]

- 工具
 - 调试器 4

工具 (续)

- 定制安装 5
- cleanpdf 实用程序 5
- CreateExportList 4
- gxlc 实用程序 4
- IBM 调试器 4
- install 5
- mergepdf 实用程序 5
- showpdf 实用程序 5
- xlcndi 5
- 共享对象文件 31
- 共享内存并行化 6

[H]

- 汇编程序
 - 源 (.S) 文件 31
 - 源 (.s) 文件 31

[J]

- 基本示例, 描述的 viii
- 静态链接 33

[K]

- 可执行文件 31
- 库 31

[L]

- 链接
 - 动态 33
 - 静态 33
- 链接过程 32
- 列表 31

[M]

- 命令 4

[N]

- 内置函数 11, 14, 23

[Q]

- 迁移
 - 源代码 30

[S]

- 实用程序
 - 定制安装 5
 - cleanpdf 5
 - CreateExportList 4
 - gxlc 4
 - IBM 调试器 4
 - install 5
 - mergepdf 5
 - showpdf 5
 - xlcndi 5
- 输出文件 31
- 输入文件 31

[W]

- 文件
 - 编辑源代码 29
 - 输出 31
 - 输入 31
- 问题确定 34

[X]

- 性能 20
 - 优化变换 5

[Y]

- 优化 20
 - 程序 5
- 与 GNU 兼容 3
- 语言标准 2
- 语言支持 2
- 源级别调试支持 8
- 源文件 31
- 运行编译器 29
- 运行时环境 34
- 运行时库 31
- 运行时选项 34

[Z]

- 执行程序 33
- 执行链接程序 32

[数字]

- 64 位环境 6

A

a.out 文件 31

D

dbx 调试器 8, 34

O

OpenMP 7

S

SMP

程序, 编译 30

SMP 程序 6

X

xl.ccfg 文件 30

[特别字符]

.a 文件 31

.c 文件 31

.i 文件 31

.lst 文件 31

.mod 文件 31

.o 文件 31

.S 文件 31

.s 文件 31



程序号: 5765-J06 和 5725-C71

Printed in China

S151-2061-02

