

Sun ONE Studio
C++程序员手册

高波 林宇 译

Sun™ ONE
Open Net Environment

 *Sun*
microsystems
We make the net work.

Sun ONE Studio C++程序员手册

Sun Microsystems, Inc 著

高波 林宇 译



机械工业出版社
China Machine Press

作为 Forte Tools 系列产品之一, Forte Developer C++ 7.0 是功能强大易于使用的 C++ 编译器, 主要用于在 UltraSPARC 工作站和 Sun 企业服务器上, 帮助开发商将 C/C++ 应用扩展到服务器架构上, 并使之运行在 Solaris 操作环境下。

本书分三部分来讲述 Forte Developer C++ 7.0。第一部分介绍 Forte Developer C++ 7.0 编译器的一般特性, 帮助读者熟悉编译器。第二部分详细讲述了 C++ 和 C 在编程思想上的差异, 并结合操作系统的特性, 教会读者如何在 UNIX 环境下编写 C++ 程序, 这是本书的核心内容。第三部分介绍了如何使用库。本书附录给出了 C++ 编译器选项和 Pragma 参考, 并介绍了 Sun ONE Studio 7。

本书适合于熟悉 C++ 并对 Solaris 运行环境及 UNIX 命令有所了解的编程开发人员。

Authorized translation from the English language edition published by Sun Microsystems, Inc.

Copyright © 2002 by Sun Microsystems, Inc.

All right reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

本书中文简体字版由 Sun Microsystems, Inc. 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

Sun ONE Studio C++ 程序员手册/美国太阳微系统公司著; 高波等译. —北京: 机械工业出版社, 2002. 12

书名原文: C++ User's Guide: Forte Developer 7

ISBN 7-111-11092-7

I. S... II. ①美... ②高... III. C 语言—编译器 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 081232 号

机械工业出版社 (北京市西城区百万庄南大街 22 号 邮政编码 100037)

责任编辑: 侯春望

中国人民解放军后勤指挥学院印刷厂印刷 • 新华书店北京发行所发行

2002 年 12 月第 1 版第 1 次印刷

787mm×1092mm¹/16•19.5 印张

印数: 0 001 - 3 000 册

定价: 48.00 元 (附两张光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换。

阅读本手册之前

本手册将指导你使用 Forte Developer C++编译器，并提供了关于命令行编译器选项的详细信息。适合于熟悉 C++工作原理和对 Solaris 操作环境及 UNIX 命令有所了解的编程人员。

本手册的结构

本手册包括以下几个主题：

Sun 工作站的 C++编译器 第 1 章是有关编译器的介绍性内容，例如所遵循的标准和新的特性。第 2 章解释如何使用编译器，第 3 章讨论怎样使用编译器的命令行选项。

编写 C++程序 第 4 章讨论如何编译那些被其他编译器普遍接受的非标准代码。第 5 章对创建和组织头文件及模板定义给出了一些建议。第 6 章讨论如何创建和使用模板，第 7 章解释编译模板的各个选项。第 8 章讨论异常处理。第 9 章给出强制类型转换操作的有关信息。第 10 章讨论对 Sun 工作站的 C++编译器的性能有较大影响的技术。第 11 章提供了关于创建多线程程序的有关信息。

库 第 12 章解释如何使用编译器所提供的库。第 13 章讨论 C++的标准库。第 14 章讨论标准 `iostream` 库（用于兼容模式）。第 15 章讨论复数库（用于兼容模式）。第 16 章提供关于创建库的有关信息。

编译器选项 附录 A 给出编译器选项的详细信息。

Progmas 附录 B 提供了有关 Progmas 的信息。

术语表 术语表定义了本书中用到的 C++及其相关的术语。

约定

代码约定

代码符号	含义	符号	代码例子
[]	中括号中内容为可选参数	<code>-compat[=n]</code>	<code>-compat=4</code>
{ }	大括号中内容为必需的选项	<code>d{yln}</code>	<code>dy</code>
	管道符号用于将可能被选用的参数分隔开	<code>B{dynamic static}</code>	<code>Bstatic</code>

(续)

代码符号	含 义	符 号	代码例子
:	冒号和逗号一样，有时用于将参数分隔开	Rdir[:dir]	R/local/libs:/U/a
...	省略号表示某个序列内的省略	xinline=f/[...fn]	xinline=alpha,dos

shell 提示符

Shell	命 令
C shell	%
Bourne shell 和 Korn shell	\$
C shell、Bourne shell 和 Korn shell 超级用户	#

访问 Forte Developer 开发工具和帮助手册

Forte Developer 的产品组件和手册页没有安装在标准的 `/usr/bin/` 和 `/usr/share/man` 目录下。要获得 Forte Developer 编译器和工具，必须在 `PATH` 环境变量中有 Forte Developer 组件目录。而要获得 Forte Developer 手册页，则必须在 `MANPATH` 环境变量中有 Forte Developer 手册目录。

要获得更多关于变量 `PATH` 的信息，可参见 `cs`h (1)、`sh` (1) 和 `ksh` (1) 手册页，而要获得更多关于变量 `MANPATH` 的信息，可参见 `man` (1) 手册页。关于设置 `PATH` 和 `MANPATH` 变量来获得此 Forte Developer 版本的详细信息，可参见安装指南或者询问系统管理员。

注意 本节所述均假设 Forte Developer 产品是安装在 `/opt` 目录下的。如果你的产品软件没有安装在 `/opt` 目录下，请向系统管理员请求一个在你的系统上等价的路径。

访问 Forte Developer 编译器和工具

采用以下步骤判定是否需要改变 `PATH` 变量来访问 Forte Developer 的编译器和工具

- 判定是否需要设置 `PATH` 环境变量。

1) 在命令提示符后面输入以下命令，显示 `PATH` 变量的当前值。

```
% echo $PATH
```

2) 在输出结果中寻找包含 `/opt/SUNWspro/bin/` 的路径。

如果找到，表明 `PATH` 变量已经设置为可访问 Forte Developer 开发工具的值；否则，请按

面所述步骤设置 PATH 环境变量。

- 设置 PATH 环境变量使其能访问 Forte Developer 编译器和工具。

1) 如果使用 C shell, 则编辑用户自身的.cshrc 文件; 如果使用 Bourne shell 或 Korn shell, 则编辑用户自身的.profile 文件。

- 2) 将以下字符串加入 PATH 环境变量中:

```
/opt/SUNWspro/bin
```

访问 Forte Developer 手册页

采用以下步骤判定是否需要改变 MANPATH 变量来访问 Forte Developer 手册页。

- 判定是否需要设置 MANPATH 环境变量。

- 1) 在命令提示符后面输入以下命令, 请求 dbx 手册页。

```
% man dbx
```

- 2) 检查输出。

如果不能找到 dbx (1) 手册页, 或者显示的手册页不是对应于当前所装软件的版本, 则按下面所述步骤设置 MANPATH 环境变量。

- 设置 MANPATH 环境变量使其能访问 Forte Developer 手册页。

1) 如果使用 C shell, 则编辑用户自身的.cshrc 文件; 如果使用 Bourne shell 或 Korn shell, 则编辑用户自身的.profile 文件。

- 2) 将以下字符串加入 MANPATH 环境变量中:

```
/opt/SUNWspro/man
```

访问 Forte Developer 文档

可以在以下位置访问 Forte Developer 产品文档:

- 据文档索引可以获得产品文档, 而文档索引与产品一起安装在本地系统或者网络上:
`/opt/SUNWspro/docs/index.html`。

如果产品软件没有安装在/opt 目录下, 则应向系统管理员请求你的系统上的等价路径。

- 大多数手册页都可以从 docs.sun.com 网站获得, 以下是仅从所安装产品中就可获得的:
 - *Standard C++ Class reference* (标准 C++类库参考)。
 - *Standard C++ Library User's Guide* (标准 C++库用户指南)。
 - *Tools.h++ Class Library Reference* (Tools.h++类库参考)。
 - *Tools.h++ User's Guide* (Tools.h++用户指南)。

docs.sun.com 网站 (<http://docs.sun.com>) 使得可以通过互联网来阅读、打印和购买 Sun

Microsystems 手册。如果不能找到所需手册，可参见与产品一同安装在本地系统或网络上的文档索引。

注意 Sun 对本文所提到的第三方网站的可用性不负任何责任，而且对来自这类网站或来源的内容、广告、产品或其他资料不作任何保证，若因为使用来自这些网站或来源的内容、产品、服务等造成危害或损失，Sun 将不承担任何责任。

可访问格式的产品文档

Forte Developer 7 产品文档提供的是可访问格式，通过使用一些辅助技术，它对于有缺陷的使用者来说也同样可读。你可以在下表所述内容中找到文档的可访问版本。如果产品软件不是安装在/opt 目录下，请向系统管理员请求一个在你的系统上等价的路径。

文档类型	可访问版本的格式和位置
手册（除第三方手册）	http://docs.sun.com 上的 HTML
第三方手册： <ul style="list-style-type: none"> • <i>Standard C++ Class reference</i>（标准 C++ 类库参考） • <i>Standard C++ Library User's Guide</i>（标准 C++ 库用户指南） • <i>Tools.h++ Class Library Reference</i>（Tools.h++ 类库参考） • <i>Tools.h++ User's Guide</i>（Tools.h++ 用户指南） 	安装产品内的 HTML，可通过 file:/opt/SUNWspro/docs/index.html/ 上的文档索引获得
Readme 文件和手册页	安装产品内的 HTML，可通过 file:/opt/SUNWspro/docs/index.html/ 上的文档索引获得
发行注释	产品 CD 上的文本文件，在 /cdrom/devpro_v10n1_sparc/release_notes.txt 中

有关 Forte Developer 文档

下表所述为在 file:/opt/SUNWspro/docs/index.html 下可获得的有关文档。如果产品软件不是安装在/opt 目录下，请向系统管理员请求一个在你的系统上等价的路径。

文档名	说明
<i>Numerical Computation Guide</i> （数值计算指南）	关于浮点计算的数值精度的描述性文章

访问有关 Solaris 文档

下表所述为在 docs.sun.com 网站可获得的有关文档。

文档集	文档标题	说 明
Solaris 参考手册包	参见手册页部分的标题	提供有关 Solaris 运行环境的信息
Solaris 软件开发包	<i>Linker and Libraries Guide</i> (连接器和库指南)	描述 Solaris 连接编辑器和运行时间连接器的操作
Solaris 软件开发包	<i>Multithreaded Programming Guide</i> (多线程编程指南)	包括 POSIX 和 Solaris 线程 API, 同步对象编程, 编译多线程程序和为多线程程序寻找工具

访问与 C++有关的手册页

本手册提供了 C++库的可用手册页列表。下表列出了其他与 C++有关的手册页列表。

标 题	说 明
C++filt	依次复制每个文件名, 在对看似 C++修饰名称的符号进行译码后, 将其写入标准输出
dem	修复一个或多个指定的 C++名称
fbe	从汇编语言源文件中创建对象文件
fpversion	打印系统 CPU 和 FPU 的信息
gprof	生成程序的执行配置文件
ild	增量连接, 允许将修改后的对象代码插入已生成的可执行文件中
inline	扩展内联过程调用汇编器
lex	生成词法分析程序
rpcgen	生成 C/C++代码以实现 RPC 协议
sigfpe	允许对具体 SIGFPE 代码进行信号处理
stdarg	处理变量参数列表
varargs	处理变量参数列表
version	显示对象文件或二进制的版本标识
yacc	为执行 LALR (1) 解析算法的一个简单自动开关将上下文无关的语法转换为一系列图表

参考书

以下是针对 C++ 语言的参考书:

《*The C++ Programming Language*》(C++程序设计语言)[⊗]3rd edition, Bjarne Stroustrup (Addison-Wesley, 1997).

《*The C++ Standard Library*》, Nicolai Josuttis (Addison-Wesley, 1999).

《*Generic Programming and the STL*》, Matthew Austern (Addison-Wesley, 1999).

《*Standard C++ IOStreams and Locales*》, Angelika Langer and Klaus Kreft (Addison-Wesley, 2000).

《*Thinking in C++*》(C++编程思想)[⊗], *Volume 1, Second Edition*, Bruce Eckel (Prentice Hall, 2000).

《*The Annotated C++ Reference Manual*》, Margaret A. Ellis and Bjarne Stroustrup, (Addison-Wesley, 1990).

《*Design Patterns: Elements of Reusable Object-Oriented Software*》, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides (Addison-Wesley, 1995).

《*C++ Primer*》, Third Edition, Stanley B. Lippman and Josee Lajoie (Addison-Wesley, 1998).

《*Effective C++—50 Ways to Improve Your Programs and Designs*》, Second Edition, Scott Meyers (Addison-Wesley, 1998).

《*More Effective C++—35 Ways to Improve Your Programs and Designs*》, Scott Meyers (Addison-Wesley, 1996).

《*Efficient C++: Performance Programming Techniques*》, Dov Bulka and David Mayhew (Addison-Wesley, 2000).

联系方式:

Sun 希望不断改进它的文档, 欢迎你评价和建议用电子邮件发至 Sun 的如下地址:
docfeedback@sun.com。

[⊗] 该书已由机械工业出版社引进出版。——编者注

目 录

阅读本手册之前

第一部分 C++编译器

第 1 章 C++编译器概述	2	2.6.1 交换区的大小	15
1.1 所遵循的标准	2	2.6.2 增大交换区	15
1.2 C++ Readme 文件	2	2.6.3 控制虚拟内存	16
1.3 手册页	3	2.6.4 内存的需求	16
1.4 许可	3	2.7 简化命令	17
1.5 C++编译器的新特性	3	2.7.1 在 C shell 中使用别名	17
1.6 C++工具	5	2.7.2 使用 CCFLAGS 来指定编译选项	17
1.7 本地语言的支持	5	2.7.3 使用 make	17
第 2 章 使用 C++编译器	6	第 3 章 使用 C++的编译器选项	20
2.1 开始使用	6	3.1 语法	20
2.2 运行编译器	7	3.2 通用的法则	20
2.2.1 命令语法	7	3.3 按功能概述选项	21
2.2.2 文件命名惯例	8	3.3.1 代码生成选项	21
2.2.3 使用多个源文件	8	3.3.2 调试选项	21
2.3 用不同版本的编译器编译	9	3.3.3 浮点选项	22
2.4 编译和连接	10	3.3.4 语言选项	23
2.4.1 编译连接的顺序	10	3.3.5 库选项	23
2.4.2 单独的编译和连接	10	3.3.6 许可选项	24
2.4.3 一致编译和连接	10	3.3.7 过时的选项	24
2.4.4 SPARC V9 的编译	11	3.3.8 输出选项	25
2.4.5 诊断编译器	12	3.3.9 性能选项	26
2.4.6 理解编译器的组织	12	3.3.10 预处理器选项	27
2.5 预处理指令和名称	14	3.3.11 配置选项	27
2.5.1 Pragmas	14	3.3.12 参考选项	27
2.5.2 #define 所列的可变参数	14	3.3.13 源选项	27
2.5.3 预定义的名字	15	3.3.14 模板选项	28
2.5.4 #error	15	3.3.15 线程选项	28
2.6 内存需求	15	第二部分 编写 C++程序	
		第 4 章 语言扩展	30
		4.1 重载更少限定性的虚函数	30

4.2	enum 类型和变量的前置声明	30	6.6.4	部分特殊化	46
4.3	使用不完整的 enum 类型	31	6.7	模板使用中的问题	47
4.4	将 enum 名作为范围限定词	31	6.7.1	非本地命名解析和实例化	47
4.5	使用匿名 struct 体声明	31	6.7.2	本地类型作为模板参数	47
4.6	传递匿名类实例的地址	33	6.7.3	模板函数的友元声明	48
4.7	声明静态的命名空间函数作为类的友元	33	6.7.4	在模板定义中使用限定名	50
4.8	为函数名使用预定义_func_符号	34	6.7.5	嵌套模板声明	51
第 5 章	程序组织	35	6.7.6	引用静态变量和静态函数	51
5.1	头文件	35	6.7.7	在同一目录中建立多个使用模板的程序	51
5.1.1	语言适应性头文件	35	第 7 章	编译模板	55
5.1.2	幂等头文件	36	7.1	verbose 编译	55
5.2	模板定义	36	7.2	模板命令	55
5.2.1	包含式模板定义	37	7.3	模板实例的放置和连接	55
5.2.2	分离式模板定义	37	7.3.1	外部实例	56
第 6 章	创建和使用模板	39	7.3.2	静态实例	56
6.1	函数模板	39	7.3.3	全局实例	56
6.1.1	函数模板声明	39	7.3.4	显式实例	57
6.1.2	函数模板定义	39	7.3.5	半显式实例	57
6.1.3	使用函数模板	40	7.4	模板库	57
6.2	类模板	40	7.4.1	库结构	57
6.2.1	类模板声明	40	7.4.2	写入模板库	57
6.2.2	类模板定义	40	7.4.3	读取多个模板库	58
6.2.3	类模板成员定义	41	7.4.4	共享模板库	58
6.2.4	使用类模板	42	7.5	模板定义搜索	58
6.3	模板实例化	42	7.5.1	源文件定位规则	58
6.3.1	隐式模板实例化	43	7.5.2	定义搜索路径	59
6.3.2	类整体实例化	43	7.6	模板实例的自动一致性	59
6.3.3	显式模板实例化	43	7.7	编译时间实例化	59
6.4	模板合成	44	7.8	模板选项文件	59
6.5	默认模板参数	45	7.8.1	注释	60
6.6	模板特殊化	45	7.8.2	文件包含	60
6.6.1	模板特殊化声明	45	7.8.3	源文件扩展名	60
6.6.2	模板特殊化定义	45	7.8.4	定义源位置	60
6.6.3	模板特殊化的使用和实例化	46	7.8.5	模板特殊化入口	62

第 8 章 异常处理	65	11.4.5 对象锁定	93
8.1 同步和异步异常	65	11.4.6 MT-Safe 类	95
8.2 指定运行时错误	65	11.4.7 对象析构	95
8.3 禁止异常	66	11.4.8 一个应用程序实例	96
8.4 使用运行时函数和预定义的异常	66		
8.5 带有信号和 Setjmp/Longjmp 的 混合异常	67	第三部分 库	
8.6 建立带有异常的共享库	68	第 12 章 使用库	102
第 9 章 Cast 操作符	69	12.1 C 库	102
9.1 const_cast	69	12.2 C++编译器提供的库	102
9.2 reinterpret_cast	70	12.2.1 C++库的描述	103
9.3 static_cast	71	12.2.2 访问 C++库的手册页	104
9.4 dynamic_cast	71	12.2.3 默认的 C++库	104
9.4.1 继承层次的向上类型转换	72	12.3 相关的库选项	105
9.4.2 转换为 void*	72	12.4 使用类库	106
9.4.3 继承层次的向下和水平类型转换	72	12.4.1 iostream 库	106
第 10 章 增强程序性能	76	12.4.2 复数库	107
10.1 避免临时对象	76	12.4.3 连接 C++库	108
10.2 使用内联函数	76	12.5 静态连接标准库	109
10.3 使用默认操作	77	12.6 使用共享库	110
10.4 使用数值类	77	12.7 替换 C++标准库	111
10.4.1 选择直接传递类	78	12.7.1 哪些可以被替换	111
10.4.2 在各种处理器上直接传递类	78	12.7.2 哪些不能被替换	112
10.5 缓存成员变量	79	12.7.3 安装替换的库	112
第 11 章 多线程程序	80	12.7.4 使用替换的库	112
11.1 建立多线程程序	80	12.7.5 标准头文件的实现	112
11.1.1 隐式的多线程编译	80	第 13 章 使用 C++的标准库	115
11.1.2 使用 C++支持的线程和信号库	81	13.1 C++标准库头文件	116
11.2 在多线程程序中使用异常	81	13.2 C++标准库手册页	117
11.3 在线程间共享 C++标准库对象	81	13.3 STLport	126
11.4 在多线程环境中使用经典的 iostream	84	第 14 章 使用经典的 iostream 库	128
11.4.1 MT-Safe iostream 库的组织	84	14.1 共享 libiostream	128
11.4.2 iostream 库的接口变化	89	14.2 预定义的 iostream	129
11.4.3 全局和静态数据	92	14.3 iostream 交互的基本结构	129
11.4.4 顺序执行	93	14.4 使用经典的 iostream 库	130
		14.4.1 使用 iostream 输出	131

A.2.24	-fstore	181	A.2.59	-O <i>filename</i>	196
A.2.25	-ftrap= <i>t[,t...]</i>	181	A.2.60	+p	196
A.2.26	-G	182	A.2.61	-P	197
A.2.27	-g	183	A.2.62	-p	197
A.2.28	-g0	184	A.2.63	-pentium	197
A.2.29	-H	184	A.2.64	-pg	197
A.2.30	-h[<i>name</i>]	184	A.2.65	-PIC	197
A.2.31	-help	185	A.2.66	-pic	197
A.2.32	-I <i>pathname</i>	185	A.2.67	-pta	198
A.2.33	-I-	185	A.2.68	-ptipath	198
A.2.34	-i	187	A.2.69	-pto	198
A.2.35	-inline	187	A.2.70	-ptr	198
A.2.36	-instances= <i>a</i>	187	A.2.71	-ptv	198
A.2.37	-keptmp	188	A.2.72	-Qoption <i>phase option[,option...]</i>	198
A.2.38	-KPIC	188	A.2.73	-qoption <i>phase option</i>	199
A.2.39	-Kpic	188	A.2.74	-qp	199
A.2.40	-L <i>path</i>	188	A.2.75	-Qproduce <i>sourcetype</i>	199
A.2.41	-llib	189	A.2.76	-qproduce <i>sourcetype</i>	200
A.2.42	-libmieee	189	A.2.77	-Rpathname[: <i>pathname...</i>]	200
A.2.43	-libmil	189	A.2.78	-readme	200
A.2.44	-library = <i>l[,l...]</i>	189	A.2.79	-S	200
A.2.45	-mc	193	A.2.80	-s	201
A.2.46	-migration	193	A.2.81	-sb	201
A.2.47	-misalign	193	A.2.82	-sbfast	201
A.2.48	-mr[, <i>string</i>]	194	A.2.83	-staticlib= <i>l[,l...]</i>	201
A.2.49	-mt	194	A.2.84	-temp= <i>path</i>	202
A.2.50	-native	195	A.2.85	-template= <i>opt[,opt...]</i>	203
A.2.51	-noex	195	A.2.86	-time	203
A.2.52	-nofstore	195	A.2.87	-U <i>name</i>	203
A.2.53	-nolib	195	A.2.88	-unroll= <i>n</i>	204
A.2.54	-nolibmil	195	A.2.89	-V	204
A.2.55	-noqueue	195	A.2.90	-v	204
A.2.56	-norunpath	195	A.2.91	-vdelx	204
A.2.57	-O	196	A.2.92	-verbose= <i>v[,v...]</i>	204
A.2.58	-O <i>level</i>	196	A.2.93	+w	205

A.2.94	+w2	205	A.2.128	-xnolibmil	227
A.2.95	-w	206	A.2.129	-xnolibmopt	227
A.2.96	-xa	206	A.2.130	-xopenmp[= <i>i</i>]	227
A.2.97	-xalias_level[= <i>n</i>]	206	A.2.131	-xOlevel	228
A.2.98	-xar	208	A.2.132	-xpg	230
A.2.99	-xarch= <i>isa</i>	209	A.2.133	-xprefetch[= <i>a</i> , <i>a</i>]	230
A.2.100	-xbuiltin[={%all%none}]	211	A.2.134	-xprefetch_level[= <i>i</i>]	232
A.2.101	-xcache= <i>c</i>	212	A.2.135	-xprofile= <i>p</i>	233
A.2.102	-xcg89	213	A.2.136	-xregs= <i>r</i> {, <i>r</i> ...}	234
A.2.103	-xcg92	213	A.2.137	-xs	235
A.2.104	-xcheck[= <i>i</i>]	214	A.2.138	-xsafe=mem	235
A.2.105	-xchip= <i>c</i>	214	A.2.139	-xsb	236
A.2.106	-xcode= <i>a</i>	215	A.2.140	-xsbfast	236
A.2.107	-xcrossfile[= <i>n</i>]	216	A.2.141	-xspace	236
A.2.108	-xF	217	A.2.142	-xtarget= <i>t</i>	236
A.2.109	-xhelp=flags	217	A.2.143	-xtime	242
A.2.110	-xhelp=readme	217	A.2.144	-xunroll= <i>n</i>	242
A.2.111	-xia	218	A.2.145	-xtrigraphs[={yesno}]	242
A.2.112	-xildoff	218	A.2.146	-xwe	243
A.2.113	-xildon	218	A.2.147	-z[<i>arg</i>]	244
A.2.114	-xinline[=func_spec [<i>func_spec</i> ...]]	219	附录 B Pragma	245	
A.2.115	-xipo[={0 1 2}]	220	B.1 Pragma 的格式	245	
A.2.116	-xlang= <i>language</i> [, <i>language</i>]	221	B.2 Pragma 参考	246	
A.2.117	-xlibmieee	222	B.2.1 #pragma align	246	
A.2.118	-xlibmil	222	B.2.2 #pragma init	246	
A.2.119	-xlibmopt	223	B.2.3 #pragma fini	247	
A.2.120	-xlic_lib=sunperf	223	B.2.4 #pragma ident	247	
A.2.121	-xlicinfo	224	B.2.5 #pragma no_side_effect	247	
A.2.122	-Xm	224	B.2.6 #pragma pack(<i>n</i>)	248	
A.2.123	-xM	224	B.2.7 #pragma returns_new_memory	249	
A.2.124	-xM1	224	B.2.8 #pragma unknown_control_flow	249	
A.2.125	-xMerge	224	B.2.9 #pragma weak	250	
A.2.126	-xnativeconnect[= <i>i</i>]	225	附录 C Sun ONE Studio 7, Solaris 企业版 浏览向导 (原 Forte Developer, 企业版 7)	252	
A.2.127	-xnolib	225			

C.1 启动 Sun ONE Studio 7——用于		C.4.1 准备一个用于调试的程序	266
Solaris IDE 的企业版	252	C.4.2 开始调试程序	266
C.1.1 获得产品许可	253	C.4.3 打开调试器窗口	268
C.1.2 安装 Sun ONE Studio 7——用于		C.4.4 单步执行代码	269
Solaris 产品企业版	253	C.4.5 设置断点	269
C.1.3 访问 Sun ONE Studio 7——用于		C.4.6 运行程序	269
Solaris 在线产品的企业版	253	C.4.7 使用 Fix 和 Continue	271
C.1.4 系统需求	253	C.4.8 使用运行时检查	272
C.1.5 开始浏览	254	C.4.9 设置监视	273
C.2 创建代码	257	C.4.10 调试 Java 程序	274
C.2.1 Makefile 向导	257	C.4.11 其他调试特性	278
C.2.2 启动 Build	258	C.5 性能分析器	279
C.2.3 其他的创建特性	259	C.5.1 性能分析	279
C.3 本地连接器工具	260	C.5.2 分析数据	279
C.3.1 运用范例	260	C.6 用于 Solaris 的 Sun ONE Studio 7	
C.3.2 设置环境	261	企业版的工具	283
C.3.3 创建共享库	261	C.6.1 X-Designer 软件	283
C.3.4 创建本地连接器描述文件	262	C.6.2 支持 CVS	284
C.3.5 选择用于 Java API 的接口	263	C.6.3 具体语言的特性	284
C.3.6 生成本地连接器组件	264	C.7 获得更多信息	285
C.3.7 执行文件状态程序	265	术语表	286
C.4 调试器	266		

本书表格

表 2-1	C++编译器可识别的文件后缀名	8
表 2-2	C++编译系统的组件	13
表 3-1	选项语法格式的例子	20
表 3-2	代码生成选项	21
表 3-3	调试选项	22
表 3-4	浮点选项	22
表 3-5	语言选项	23
表 3-6	库选项	23
表 3-7	许可选项	24
表 3-8	过时的选项	24
表 3-9	输出选项	25
表 3-10	性能选项	26
表 3-11	预处理器选项	27
表 3-12	配置选项	27
表 3-13	参考选项	27
表 3-14	源选项	28
表 3-15	模板选项	28
表 3-16	线程选项	28
表 10-1	不同体系结构传递 struct 和联合体的方式	78
表 11-1	iostream 原核心类	84
表 11-2	MT-Safe 的可重入公有函数	85
表 12-1	和 C++编译器一起提供的库	102
表 12-2	连接 C++库的编译器选项	109
表 12-3	头文件搜索的例子	113
表 13-1	C++标准库的头文件	116
表 13-2	C++标准库手册	117
表 14-1	iostream 例程的头文件	130
表 14-2	iostream 预定义的操纵符	140
表 14-3	iostream 手册页概述	146
表 14-4	iostream 术语	147