



新增功能

Sun™ Studio 8

Sun Microsystems, Inc.
www.sun.com

部件号码 817-5792-10
2004 年 4 月, 修订 A

关于本文档的建议请发到: <http://www.sun.com/hwdocs/feedback>

版权所有 © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

本次分发可能包含由第三方开发的内容。

第三方软件（包括字体技术）的版权归 Sun 供应商所有并由其授权。该产品的部分内容出自 Cray90（Cray Research, Inc. 的产品）。

libdwarf 和 libredblack 在 2000 年注册的版权归 Silicon Graphics, Inc. 所有，您可以依据 GNU Lesser General Public License 从 <http://www.sgi.com> 获得。

该产品的部分内容可能出自 Berkeley BSD 系统，由加州大学 (University of California) 授权。UNIX 是在美国和其它国家（地区）的注册商标，由 X/Open Company, Ltd. 独家授权。

Sun、Sun Microsystems、Sun 徽标、Java、Sun ONE Studio、Solaris 徽标和 Sun ONE 徽标是 Sun Microsystems, Inc. 在美国和其它国家（地区）的商标或注册商标。

Netscape 和 Netscape Navigator 是 Netscape Communications Corporation 在美国和其它国家（地区）的商标或注册商标。

所有的 SPARC 商标均需获得授权才能使用，它们是 SPARC International, Inc. 在美国和其它国家（地区）的商标或注册商标。带有 SPARC 商标的产品所基于的体系结构是由 Sun Microsystems, Inc. 开发的。

该服务手册所涵盖的产品以及涉及的信息均受制于美国的出口控制法，并有可能要遵守其它国家（地区）的出口或进口法规。严禁将产品用于核、导弹、生化武器或核海事等最终目的或交给有此类企图的最终用户，无论是直接还是间接。严禁将产品出口或再次出口到美国禁运的国家（地区）或美国出口排除列表中的实体，包括但不限于被拒绝的人员以及专门指定的公民列表。

本文档按“原样”提供，对所有明示或默示的条件、陈述和担保，包括对适销性、特殊用途的适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。



目录

开始之前	5
排版惯例	5
Shell 提示符	6
访问 Sun Studio 软件和手册页	7
访问编译器和工具文档	9
访问相关的 Solaris 文档	11
开发人员资源	11
与 Sun 技术支持联系	11
发送意见	12
1. Sun Studio 8 新特性	13
集成开发环境 (IDE)	14
启动 IDE	14
性能分析工具	14
数据集合特性	15
数据表示特性	15
文档	15

2. Sun ONE Studio 8, 编译器集合新特性 17

C 编译器 18

C++ 编译器 22

Fortran 编译器 28

dbx 命令行调试程序 31

区间运算 31

Sun Performance Library 32

dmake 33

性能分析工具 34

文档 35

开始之前

《新增功能》描述了 Sun™ Studio 8 软件发行版本的新特性。

Sun Studio 8 包括集成开发环境 (IDE)，提供了创建、编辑、生成、调试和分析 C、C++ 或 Fortran 应用程序性能的模块。还包括一组基本 Java™ 语言支持模块，如果需要进行 JNI（Java™ 本地接口）开发则可以启用该模块。

除 IDE 之外，Sun Studio 8 包括了与 Sun ONE Studio 8 编译器集合一起发行的 C、C++ 和 Fortran 编译器和库。

排版惯例

表 P-1 排版惯例

字体	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑您的 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 % You have mail.
AaBbCc123	输入的内容，以便与计算机屏幕输出相区别	% su Password:
<i>AaBbCc123</i>	书名、新词或术语以及要强调的词	请阅读 《 <i>用户指南</i> 》的第 6 章。 这些称作类选项。 您 <i>必须</i> 是超级用户才能执行此操作。
<i>AaBbCc123</i>	命令行变量；用实际名称或值替换	要删除文件，请键入 <code>rm filename</code> 。

表 P-2 代码惯例

代码符号	含义	表示法	代码示例
[]	括号包含可选参数。	O[n]	-O4, -O
{ }	大括号包含所需选项的选项集合。	d{y n}	-dy
	“pipe”或“bar”符号用于分隔参数，所以只能选择两者之一。	B{dynamic static}	-Bstatic
:	与逗号一样，分号有时可用于分隔参数。	Rdir[:dir]	-R/local/libs:/U/a
...	省略号表示一系列省略。	xinline= <i>fl</i> [... <i>fn</i>]	-xinline=alpha,dos

Shell 提示符

Shell	提示符
C shell	<i>machine-name</i> %
C shell 超级用户	<i>machine-name</i> #
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 的超级用户	#

访问 Sun Studio 软件和手册页

编译器和工具以及它们的手册页并没有安装到标准的 `/usr/bin/` 和 `/usr/share/man` 目录中。要访问编译器和工具，必须正确设置 `PATH` 环境变量（请参阅第 7 页的“访问编译器和工具”）。要访问手册页，必须正确设置 `MANPATH` 环境变量（请参阅第 8 页的“访问手册页”）。

关于 `PATH` 变量的更多信息，请参阅 `csh(1)`、`sh(1)` 和 `ksh(1)` 手册页。关于 `MANPATH` 变量的更多信息，请参阅 `man(1)` 手册页。关于设置 `PATH` 变量和 `MANPATH` 变量以访问此发行版本的更多信息，请参阅安装指南或询问系统管理员。

注 – 本节中的信息假设 Sun ONE Studio 编译器和工具安装在 `/opt` 目录中。如果软件没有安装在 `/opt` 中，请咨询系统管理员以获取系统中的等价路径。

访问编译器和工具

使用下列步骤来决定是否需要更改 `PATH` 变量以访问编译器和工具。

▼ 要决定是否需要设置 `PATH` 环境变量

1. 通过在命令提示符后输入下列内容以显示 `PATH` 变量的当前值。

```
% echo $PATH
```

2. 查看输出中是否有包含 `/opt/SUNWspro/bin/` 的路径字符串。

如果找到该路径，您的 `PATH` 变量已经设置好，可以访问编译器和工具了。如果没有找到该路径，按照下一步中的指示来设置 `PATH` 环境变量。

▼ 要设置 `PATH` 环境变量以访问编译器和工具

1. 如果使用的是 **C shell**，请编辑起始 `.cshrc` 文件。如果使用的是 **Bourne shell** 或 **Korn shell**，请编辑起始 `.profile` 文件。
2. 将下列内容增加到 `PATH` 环境变量。如果已安装 Sun ONE Studio 软件或 Forte Developer 软件，则将以下路径增加到这些安装的路径之前。

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

访问手册页

使用下列步骤来决定是否需要更改 `MANPATH` 变量以访问手册页。

▼ 要决定是否需要设置 `MANPATH` 环境变量

1. 通过在命令提示符后输入下列内容以请求 `dbx` 手册页。

```
% man dbx
```

2. 如果有输出的话，请查看输出。

如果 `dbx(1)` 手册页无法找到或者显示的手册页不是用于安装软件的当前版本，请按照下一步中的指示来设置 `MANPATH` 环境变量。

▼ 要设置 `MANPATH` 环境变量以访问手册页

1. 如果使用的是 **C shell**，请编辑起始 `.cshrc` 文件。如果使用的是 **Bourne shell** 或 **Korn shell**，请编辑起始 `.profile` 文件。
2. 将下列内容增加到 `MANPATH` 环境变量。

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

访问集成开发环境

Sun Studio 8 集成开发环境 (IDE) 提供了创建、编辑、生成、调试和分析 C、C++ 或 Fortran 应用程序性能模块。

IDE 需要 Sun Studio 8 的核心平台组件。如果核心平台组件没有安装到以下位置之一，则必须将 `SPRO_NETBEANS_HOME` 环境变量设置到安装核心平台组件的位置 (*installation_directory/netbeans/3.5R*)。

- 缺省安装目录 `/opt/netbeans/3.5R`
- 与 Sun Studio 8 的编译器和工具组件相同的位置（例如安装在 `/foo/SUNWspro` 中的编译器和工具组件和 `/foo/netbeans/3.5R` 中的核心平台组件）。

启动 IDE 的命令是 `sunstudio`。有关该命令的详细信息，请参阅 `sunstudio(1)` 手册页。

访问编译器和工具文档

您可以在下列位置访问文档：

- 可以在随软件一起安装的文档索引 `file:/opt/SUNWspro/docs/index.html` 中（位于本地系统或网络上）获取文档。

如果软件没有安装在 `/opt` 中，请咨询系统管理员以获取系统中的等价路径。

- 大多数的手册都可以从 `docs.sun.com`sm web 站点上获得。下列文档只能在安装的软件中获得：
 - *Standard C++ Library Class Reference*
 - *Standard C++ Library User's Guide*
 - *Tools.h++ Class Library Reference*
 - *Tools.h++ User's Guide*
- 发行版本说明可以从 `docs.sun.com` web 站点上获得。
- 在 IDE 中通过 [帮助] 菜单或窗口和对话框上的 [帮助] 按钮可以访问 IDE 所有组件的联机帮助。

您可以通过因特网在 `docs.sun.com` web 站点 (<http://docs.sun.com>) 上阅读、打印和购买 Sun Microsystems 的各种手册。如果找不到手册，请参阅和软件一起安装在本地系统或网络中的文档索引。

注 - Sun 不对本文档所提及的第三方 web 站点的可用性负责，而且 Sun 不认可也不对以上站点或资源上的任何内容、广告、产品或其它资料承担责任。此外，Sun 公司也不就因您使用或依靠以上任何站点或资源上的（或通过该站点或资源所获取的）内容、物品或服务所产生的（或所谓产生的）任何损失承担责任。

使用易读格式的文档

该文档以易读格式提供，以方便残障用户使用辅助技术进行阅读。您还可以按照下表所描述的信息找到文档的易读版本。如果软件没有安装在 /opt 中，请咨询系统管理员以获取系统中的等价路径。

文档类型	易读版本的格式和位置
手册（第三方手册除外）	HTML，位于 http://docs.sun.com
第三方手册： <ul style="list-style-type: none">• <i>Standard C++ Library Class Reference</i>• <i>Standard C++ Library User's Guide</i>• <i>Tools.h++ Class Library Reference</i>• <i>Tools.h++ User's Guide</i>	HTML，位于安装的软件中的文档索引 file:/opt/SUNWspr0/docs/index.html
自述文件和手册页	HTML，位于安装的软件中的文档索引 file:/opt/SUNWspr0/docs/index.html
联机帮助	通过 IDE 中的 [帮助] 菜单可以使用 HTML
发行说明	HTML，位于 http://docs.sun.com

相关编译器和工具文档

下表描述的相关文档可以在 <file:/opt/SUNWspr0/docs/index.html> 和 <http://docs.sun.com> 上获得。如果软件没有安装在 /opt 中，请咨询系统管理员以获取系统中的等价路径。

文档标题	说明
《Fortran 编程指南》	描述如何在 Solaris 环境中编写高效 Fortran 代码；输入/输出、库、性能、调试和并行处理。
《Fortran 库参考》	详细说明 Fortran 库和内部例程
《Fortran 用户指南》	描述 f95 编译器的编译时环境和命令行选项。也包括将传统 f77 程序迁移到 f95 的指南。
《C 用户指南》	描述 cc 编译器的编译时环境和命令行选项。
《C++ 用户指南》	描述 CC 编译器的编译时环境和命令行选项。
《数值计算指南》	描述关于浮点计算数值精确性的问题。

访问相关的 Solaris 文档

下表描述了可从 docs.sun.com web 站点上获得的相关文档。

文档集合	文档标题	说明
Solaris Reference Manual Collection	参阅手册页部分的标题。	提供关于 Solaris 操作环境的信息。
Solaris Software Developer Collection	<i>Linker and Libraries Guide</i>	描述 Solaris 链接编辑器和运行时链接程序的操作。
Solaris Software Developer Collection	<i>Multithreaded Programming Guide</i>	涵盖 POSIX 和 Solaris 线程 API、使用同步对象进行程序设计、编译多线程程序和多线程程序的查找工具。

开发人员资源

访问 <http://developers.sun.com/prodtech/cc> 以查找以下经常更新的资源：

- 关于编程技术和最佳方法的文章
- 短小编程提示的知识库
- 编译器和工具组件的文档以及与软件同时安装的文档的更正
- 支持等级信息
- 用户论坛
- 可下载代码示例
- 新技术预览

您可以在 <http://developers.sun.com> 上找到开发人员的额外资源。

与 Sun 技术支持联系

如果您有关于本产品的技术问题而本文档未予以解答，请访问：

<http://www.sun.com/service/contacting>

发送意见

Sun 致力于提高文档质量，并欢迎您提出宝贵的意见和建议。请通过电子邮件将您的意见发送至以下地址：

docfeedback@sun.com

请在电子邮件的主题行中包含文档的部件号码 (817-5792-10)。

Sun Studio 8 新特性

Sun™ Studio 8 替换了 Sun™ ONE Studio 8 编译器集合。Sun ONE Studio 8 编译器集合仅是命令行发行版本的 C、C++ 和 Fortran 编译器、库和工具。Sun Studio 8 包括了与 Sun ONE Studio 8 编译器集合发行版本一起发行的相同编译器和库，并增加了集成开发环境 (IDE) 和最新的性能分析器。

本章描述了以下新的 Sun Studio 8 特性。

- 集成开发环境 (IDE)
- 性能分析工具
- 文档

Sun Studio 8 发行版本包括了与 Sun ONE Studio 8 编译器集合发行版本相同的以下编译器、库和工具。与 Sun ONE Studio 8 编译器集合一起发行的编译器、库和工具新特性在第 2 章中描述。

- C 编译器
- C++ 编译器
- Fortran 编译器
- dbx 命令行调试程序
- Sun Performance Library™
- 区间运算

在多数节中，都会有一个列出组件新特性的表。该表有两列或三列：

- **两列的表。** 左侧列提供了对新特性的简短描述，右侧列则具有更详尽的描述。
- **三列的表。** 左侧列提供了对新特性的简短描述，中间列列出了相关命令或选项，右侧列则具有更详尽的描述。

注 – 要查找本章所述的 Sun Studio 8 文档，请参阅与产品软件同时安装在 `/opt/SUNWsprow/docs/index.html` 的文档索引。如果软件没有安装在 `/opt` 目录中，请联系系统管理员以了解系统或网络中的等价路径。

集成开发环境 (IDE)

Sun Studio 8 IDE 提供了创建、编辑、生成、调试和分析 C、C++ 或 Fortran 应用程序性能模块。还包括一组基本 Java™ 语言支持模块，如果需要进行 JNI（Java™ 本地接口）开发则可以启用该模块。

该发行版本的 IDE 可用于 Solaris™ 操作系统 (Solaris OS) (SPARC™ Platform Edition) 版本 7、8 和 9，和 Solaris OS (x86 Platform Edition) 版本 7、8 和 9。

Sun Studio 8 软件由两个主要组件组成：

- 包括 IDE、编译器、工具和核心平台的 Sun Studio 组件
- Core Platform 运行的 Java™ 2 Platform, Standard Edition (J2SE) 技术

缺省情况下 Sun Studio 8 组件安装在 /opt 目录中，不过安装期间可以指定替换的位置。J2SE 技术缺省安装于 /usr/jdk/j2sdk1.4.2_02 目录中，不过安装期间可以指定 /usr 中的替换目录。

IDE 的正确操作取决于能找到核心平台的 IDE，和能找到 J2SE 技术的核心平台。

启动 IDE

启动 IDE 的命令是 sunstudio。有关该命令的详细信息，请参阅 sunstudio(1) 手册页。（要显示 sunstudio(1) 手册页，必须将 /installation-directory/SUNWspro/man 置于 \$MANPATH 中。）

性能分析工具

以下几节列出了 Sun Studio 8 发行版本性能分析器工具的新数据集和演示特性。更多信息请参阅以下手册页：

- analyzer(1)
- collect(1)
- collector(1)
- er_print(1)
- libcollector(3)

数据集合特性

下表描述了新的或更改的数据集合功能。

- 现在性能分析器既可以做为集成到 IDE 中的 NetBeans 模块也可以做为源于 shell 的命令使用。
- 现在性能分析器具有允许从 GUI 选择并记录实验的 [收集器] 窗口。
- 数据集合错误检测和报告已经增强。

有关更多信息，请参阅 `collect(1)`、`collector(1)` 和 `libcollector(3)` 手册页。

数据表示特性

下表描述了新的或更改的数据表示功能。

- 性能分析器紧接着已注释源码和反汇编显示了指示高度量代码行的刻度线。
- 性能分析器在时间线中设置函数的颜色时具有附加的灵活性。
- 性能分析器在 CPU 空闲状态的时间线中显示时钟分析数据时，具有附加的灵活性。
- `er_print` 和性能分析器现在可以从不同对象文件接合数据类型，因此类型报告和显示不显示相同结构的重复条目。对象类型报告的 `er_print` 格式已改善。新 `data_layout er_print` 命令和对应的 [分析器] 标签已实现。
- `er_print` 现在具有改善的帮助消息，同时命令按照与手册页上显示的相同顺序列出。

有关更多信息，请参阅 `analyzer(1)` 和 `er_print(1)` 手册页以及性能分析器联机帮助。

文档

本节描述了 Sun Studio 8 文档新特性。

- 残障人士用户通过辅助技术同样可以阅读 Sun Studio 8 产品文档。有关更多信息请参阅第 10 页的“使用易读格式的文档”。

Sun ONE Studio 8，编译器集合 新特性

本章描述了 Sun ONE Studio 8，编译器集合编译器和命令行工具的新增特性。本发行版本主要提高了 C、C++ 和 Fortran 语言系统的性能和可移植性，并且支持 C99 语法的子集和 dbx 命令行调试程序中的 OpenMP™ 程序。

本章中所述的编译器、库和工具包括在 Sun Studio 8 发行版本中

本章包含以下各节：

- C 编译器
- C++ 编译器
- Fortran 编译器
- dbx 命令行调试程序
- Sun Performance Library
- 区间运算
- 性能分析工具（也在 Sun Studio 8 发行版本中更新）
- 文档

在多数节中，都会有一个列出组件新特性的表。该表有两列或三列：

- **两列的表。** 左侧列提供了对新特性的简短描述，右侧列则具有更详尽的描述。
- **三列的表。** 左侧列提供了对新特性的简短描述，中间列列出了相关命令或选项，右侧列则具有更详尽的描述。

注 — 要查找本章所述的 Sun ONE Studio 8，编译器集合文档，请参阅与产品软件同时安装在 `/opt/SUNWsprow/docs/index.html` 的文档索引。如果软件没有安装在 `/opt` 目录中，请联系系统管理员以了解系统或网络中的等价路径。

C 编译器

本节列出了本发行版本中 C 编译器的新特性。下表中包括下列新特性：

- 表 2-1 一般增强
- 表 2-2 更快的编译
- 表 2-3 增强的性能
- 表 2-4 更便捷的调试

关于在本节中引用的特定编译器选项的更多信息，请参阅《C 用户指南》或 `cc(1)` 手册页。

表 2-1 列出了 C 编译器的一般增强。

表 2-1 C 编译器的一般增强

特性	说明
变量作用域不再需要链接程序映射文件： <code>-xldscope</code>	现在可以使用两种不同的方法来控制动态库符号的输出。该功能称为链接程序作用域且由链接映射文件支持。首先，可以在代码中嵌入新的声明说明符。 只要在代码中直接添加 <code>__global</code> 、 <code>__symbolic</code> 、和 <code>__hidden</code> ，就可以不必使用映射文件。其次，可以通过在命令行指定 <code>-xldscope</code> 来覆盖变量作用域的缺省设置。
附加 C99 特性的实现	本发行版本增加了对以下 ISO/IEC 9899:1999（本文档中简称为 C99）特性的支持。下表仅详细描述了在本发行版本中实现的 C99 特性，它是所有已实现的 C99 特性的子集。要查看在过去和当前 C 编译器发行版本中实现的所有 C99 特性的完整列表，请参阅《C 用户指南》。其中列出了每一项的 C99 标准子节编号。 <ul style="list-style-type: none">• 6.2.5 <code>_Bool</code>• 6.2.5 <code>_Complex</code> 类型 本发行版本支持 <code>_Complex</code> 的部分实现。在 Solaris 7、8 和 9 操作系统中必须用 <code>-lcplxsupp</code> 链接。<ul style="list-style-type: none">• 6.3.2.1 不限制仅在左值进行数组到指针的转化• 6.4.4.2 十六进制浮点文字• 6.5.2.5 复合文字• 6.7.2 类型说明符• 6.10.6 STDC pragmas• 6.10.8 <code>__STDC_IEC_559</code> 和 <code>__STDC_IEC_559_COMPLEX</code> 宏

表 2-1 C 编译器的一般增强 (后续)

特性	说明
支持 VIS™ 开发者工具包: <code>-xvis</code> (SPARC®)	<p>在使用 VIS 指令集软件开发工具包 (VSDK) 中定义的汇编语言模板时, 可以使用 <code>-xvis=[yes no]</code> 选项。</p> <p>VIS 指令集是 SPARC v9 指令集的扩展。尽管 UltraSPARC® 是 64 位处理器, 但在很多情况下数据都限制在 8 位或 16 位范围内, 特别是多媒体应用程序中。VIS 指令可以用一条指令处理 4 个 16 位数据, 这个特性使得处理诸如图像、线性代数、信号处理、音频、视频以及网络等新媒体的应用程序的性能大大提高。</p> <p>有关 VSDK 的更多信息, 请参阅 http://www.sun.com/processors/documentation.html。</p>
为从属线程提供更大的缺省栈	<p>现在为从属线程提供了更大的缺省栈空间。所有从属线程都有同样的栈大小, 对于 32 位应用程序该值缺省为 4MB, 对于 64 位应用程序, 该值缺省为 8MB。可以通过环境变量 <code>STACKSIZE</code> 来设置该缺省值。</p>
增强 <code>-xprofile</code> (SPARC)	<p><code>-xprofile</code> 选项提供以下增强功能:</p> <ul style="list-style-type: none"> • 支持分析共享库 • 使用 <code>-xprofile=collect -mt</code> 选项的线程安全的配置文件集合 • 对在单个配置文件目录下分析多个程序或多个共享库的支持有所增强 <p>通过设置 <code>-xprofile=use</code> 选项, 编译器能够在配置目录下为多个拥有不同基名的目标文件找到配置数据。在找不到目标文件的配置文件数据的情况下, 编译器会使用新的选项: <code>-xprofile_pathmap=collect-prefix:use-prefix</code>。</p>
支持 UTF-16 字符串文字: <code>-xustr</code>	<p>如果需要支持使用 ISO10646 UTF-16 字符串文字的国际化程序, 那么就要指定 <code>-xustr=ascii_utf16_ushort</code> 选项。也就是说, 如果您的代码包含 16 位字符组成的字符串文字, 那么就需要使用该选项。如果不指定该选项, 编译器既不生成、也不识别 16 位的字符串文字。该选项使 U“<i>ASCII_string</i>”字符串文字被识别为无符号短整数数组。因为这样的字符串还不属于任何标准, 所以该选项的作用是使非标准 C 得以识别。</p>

表 2-2 列出了支持更快编译的 C 编译器新特性。

表 2-2 支持更快编译的 C 语言新特性

特性	说明
更快的分析 (SPARC)	在使用阶段, 同时使用 <code>-xprofile_ircache[=path]</code> 和 <code>-xprofile=collect use</code> 会重用收集阶段保存的编译数据, 从而可以减少编译时间。 在编译大程序时, 由于中间数据的保存, 使得使用阶段的编译时间大大减少。所保存的数据会占用相当大的磁盘空间。
预编译头文件: <code>-xpch</code>	本发行版本的编译器新增了预编译头文件的特性。预编译头文件的作用是减少源代码共享同一组包含文件的应用程序的编译时间, 而且这些包含文件往往有大量的源代码。预编译头文件的工作机理是, 首先从一个源文件收集一组头文件信息, 然后在重新编译该源文件或者其它有同样头文件的源文件时就可以使用这些收集到的信息。要使用这个特性, 需要指定 <code>-xpch</code> 和 <code>-xpchstop</code> 选项, 并同时使用 <code>#pragma hdrstop</code> 指令。
使用多个处理器: <code>-xjobs=n (SPARC)</code>	指定 <code>-xjobs=n</code> 选项用来设定编译器生成多少个进程来完成它的任务。在多 CPU 机器上, 该选项可以减少生成时间。目前, <code>-xjobs</code> 只在与 <code>-xipo</code> 一起使用时才有效。如果指定 <code>-xjobs=n</code> , 过程间调用优化器在编译不同的文件时会用 <code>n</code> 作为它能启动的代码生成器实例的最大数。

表 2-3 列出了支持增强性能的 C 编译器的新特性。

表 2-3 增强性能的 C 语言新特性

特性	说明
使用链接程序支持的线程局部存储来增强运行时性能： -xthreadvar	<p>可以使用编译器新增的链接程序支持的线程局部存储特性来完成以下功能：</p> <ul style="list-style-type: none">快速实现 POSIX 接口中的分配线程数据。将多进程序转换成多线程程序。将使用线程局部存储的 Windows 应用程序移植到 Solaris。快速实现 OpenMP 程序中的线程私有变量。 <p>通过声明线程局部变量，编译器现在可以支持线程局部存储。这个声明包括一个普通的变量声明和额外的 <code>__thread</code> 变量说明符，另外还需要在命令行加上 <code>-xthreadvar</code> 选项。</p>
通过减少缺页以增强运行时性能：-xF	<p>使用 <code>-xF</code> 新特性可以使链接程序对变量和函数进行重组以达到最优。这有助于解决以下会对运行时性能产生负面影响的问题：</p> <ul style="list-style-type: none">在内存中存放位置很近的无关变量会造成缓存和页的争用。在内存中存放位置很远的相关变量会造成过大的工作集。未用到的弱变量副本会造成过大的工作集，从而降低有效数据密度。
增强运行时： -xlinkopt (SPARC)	<p>当指定 <code>-xlinkopt</code> 命令时，C++ 编译器可以对可重定位对象文件进行链接时优化。</p> <p>指定 <code>-xlinkopt</code> 选项时，编译器会在链接时进行一些额外的优化，并且不会修改所链接的 <code>.o</code> 文件。这些优化仅仅在可执行程序中才出现。当编译整个程序并且使用配置文件反馈时，<code>-xlinkopt</code> 选项才最有效。</p>
增强运行时： -xpagesize= <i>n</i> (SPARC)	<p>为栈设置内存页面大小。<i>n</i> 可以是 8K、64K、512K、4M、32M、256M、2G、16G 或 <code>default</code>。您必须为目标平台上的 Solaris OS 指定有效的页面大小，诸如 <code>getpagesize(3C)</code> 的返回值。如果不指定有效的页面大小，运行时该请求就会被忽视。您可以使用 <code>pmap(1)</code> 或 <code>meminfo(2)</code> 来决定目标平台的页面大小。</p> <p>注意，该特性只在 Solaris 9 OS 中可用。使用该选项编译的程序将无法链接到早期的 Solaris OS 环境中。</p> <p>该选项其实是代表 <code>-xpagesize_stack</code> 和 <code>-xpagesize_heap</code> 的宏。</p>
基于硬件计数器的分析： -xhwcprof (SPARC)	<p>使用 <code>-xhwcprof=[enable disable]</code> 选项可以使编译器支持基于硬件计数器的分析。</p> <p>启用 <code>-xhwcprof</code> 时，编译器会生成信息来帮助工具匹配硬件计数器的数据引用，并忽略相关指令引发的事件。相关的数据类型和结构成员可能也会随着符号信息（由 <code>-g</code> 生成）一起标识。性能分析中会用到这些信息，因为仅从基于代码地址、源代码语句或例程配置文件中很难识别出这些信息。</p>

表 2-4 列出了支持更便捷调试的 C 编译器的新特性

表 2-4 支持更便捷调试的 C 语言新特性

特性	说明
DWARF 格式的调试程序信息： -xdebugformat	如在 <i>DWARF 调试信息格式</i> 中指定的那样，C 编译器会将调试信息从 Stabs 格式迁移为 DWARF 格式。如果要维护读取调试信息的软件，那么可以通过使用该选项将工具从 stabs 格式转换为 DWARF 格式。在新发行版本中，缺省设置为 <code>-xdebugformat=stabs</code> 。 移植工具时，使用 <code>-xdebugformat=dwarf</code> 选项是访问新格式的一个途径。除非您要维护读取调试信息的软件，或者某个工具要求使用这些格式之一的调试信息，否则没有必要使用这个选项。
对调试 OpenMP 程序的支持： -xopenmp=noopt	如果使用 dbx 调试 OpenMP 程序，那么编译时选用 <code>-g</code> 和 <code>-xopenmp=noopt</code> 可以让您在并行区设置断点并查看变量内容。

C++ 编译器

本节列出了本发行版本中 C++ 编译器的新特性。下表中包括下列新特性：

- 表 2-5 一般增强
- 表 2-6 更快的编译
- 表 2-7 更便捷的移植
- 表 2-8 增强的性能
- 表 2-9 增加的警告和错误控制

关于在本节中引用的特定编译器选项的更多信息，请参阅《C++ 用户指南》或 `cc(1)` 手册页。

表 2-5 列出了 C++ 编译器（5.5 版本）的一般增强。

表 2-5 C++ 编译器的一般增强

特性	说明
不再需要模板缓存： -instances	<p>此发行版本 C++ 编译器极大地改善了模板实例化。使用缺省模板实例化模型的程序现在可以在同一目录下生成多个程序。</p> <p>现在多数具有 -instances=static 且依赖于可替代实例化模型的程序都可以使用新的缺省实例化模型。</p> <p>对模板实例化进行的增强和更改通过避免模板缓冲而减少了编译时间，或者通过避免重复的静态函数而降低可执行程序的大小。</p>
变量作用域不再需要链接程序映射文件： -xldscope	<p>现在可以使用两种不同的方法来控制动态库符号的输出。这项功能称为链接程序作用域，链接程序映射文件早就对它提供了支持。首先，可以在代码中嵌入新的声明说明符。</p> <p>只要在代码中直接添加 <code>__global</code>、<code>__symbolic</code>、和 <code>__hidden</code>，就可以不必使用映射文件。其次，可以通过在命令行指定 <code>-xldscope</code> 来覆盖变量作用域的缺省设置。</p>
强大的宏诊断新工具： -xdumpmacros	<p>此发行版本新引入了两个 <code>pragma</code> 和一个编译器选项来帮助您跟踪应用程序中的宏。其中包括定义在系统头文件中的宏。</p> <p>可以在命令行使用 <code>-xdumpmacros</code> 选项来查看宏定义，也可以查看宏的定义位置、宏的取消定义位置以及宏在程序中的使用位置。如果要缩小范围，可以在源码中直接使用 <code>dumpmacros</code> 和 <code>end_dumpmacros</code> 这两个新 <code>pragma</code>。</p>
支持 VIS 开发者工具包： -xvis	<p>在使用 VIS 指令集软件开发工具包 (VSDK) 中定义的汇编语言模板时，可以使用 <code>-xvis=[yes no]</code> 选项。缺省为 <code>-xvis=no</code>。</p> <p>关于 VSDK 的更多信息，请参阅 http://www.sun.com/processors/vis</p>
支持 C99 运行时库和环境： -xlang	<p>在支持 C99 标准 (ISO/IEC 9899:1999, 编程语言 - C) 的操作系统上，<code>-xlang=c99</code> 指定调用 C 库函数的 C 和 C++ 代码的 C99 运行时行为特性。有些 C99 行为特性，例如 C 复数类型，要求 C 编译器和 <code>-xc99=%all</code> 选项一起使用，而有些行为特性，例如 <code>printf</code>，就没有这样的限制。</p> <p>兼容模式 (<code>-compat=4</code>) 对 C99 尚不支持。</p>

表 2-5 C++ 编译器的一般增强 (后续)

特性	说明
支持 UTF-16 字符串文字: <code>-xustr</code>	如果需要支持使用 ISO10646 UTF-16 字符串文字的国际化程序, 那么就要指定 <code>-xustr=ascii_utf16_ushort</code> 选项。也就是说, 如果您的代码包含 16 位字符组成的字符串文字, 那么就需要使用该选项。如果不指定该选项, 编译器既不生成、也不识别 16 位的字符串文字。该选项使 U “...” 字符串文字被识别为无符号短整数数组。因为这样的字符串还不属于任何标准, 所以该选项的作用是使非标准 C++ 得以识别。
为 OpenMP™ 提供的扩展支持: <code>-xopenmp</code>	C++ 编译器为显式并行的 OpenMP 接口继续提供支持。关于 <code>-xopenmp</code> 选项的详细信息, 请参阅 CC(1) 手册。编译器对 OpenMP 功能进行了扩展, 增加了以下功能: <ul style="list-style-type: none"> • 允许在 OpenMP 数据子句中包含类对象。 • 允许在类成员函数中出现 OpenMP pragma。
增强: <code>-xprofile</code>	<code>-xprofile</code> 选项提供以下增强功能: <ul style="list-style-type: none"> • 支持分析共享库 • 线程安全的配置文件集合使用 <code>-xprofile=collect -mt</code> • 对在单个配置文件目录下分析多个程序或多个共享库的支持有所增强。

表 2-6 列出了支持更快编译的 C++ 编译器新特性。

表 2-6 支持更快编译的 C++ 语言新特性

特性	说明
加速语法检查: <code>-xe</code>	当指定 <code>-xe</code> 时, 编译器只检查语法和文法错误, 并且不生成任何对象代码。 如果不需要编译器生成对象文件, 就可以使用 <code>-xe</code> 选项。例如, 如果要使用删除代码的方法来确定导致错误消息的原因, 使用 <code>-xe</code> 可以加速编辑、编译周期。
更快的分析: <code>-xprofile_ircache</code>	在使用阶段, 同时使用 <code>-xprofile_ircache[=path]</code> 和 <code>-xprofile=collect use</code> 会重用收集阶段保存的编译数据, 从而可以减少编译时间。 在编译大程序时, 由于中间数据的保存, 使得使用阶段的编译时间大大减少。所保存的数据会占用相当大的磁盘空间。
不再使用冗余的模板实例化: <code>-instlib=filename</code>	使用 <code>-instlib=filename</code> 选项可以避免在函数库和当前对象中生成重复的模板实例化。一般来说, 如果程序与库共享大量的实例, 则可以试试 <code>-instlib=filename</code> 选项, 您会发现编译时间会减少。 使用 <code>filename</code> 参数以指定包含现有模块实例的库 (已知的)。 <code>filename</code> 参数必须包含正斜杠 “/” 字符。与当前目录相关的路径要使用点斜杠 “./”。 <code>-instlib=filename</code> 选项不是缺省选项, 只有在指定该选项之后才能使用。该选项可被多次指定和累积。

表 2-6 支持更快编译的 C++ 语言新特性 (后续)

特性	说明
生成函数: -template= geninlinefuncs	一般情况下, C++ 编译器不会生成内联模板函数, 除非该函数既被调用了, 又不能内联。然而, 如果指定 -template=geninlinefuncs, 编译器会将以前未生成的显式实例化类模板的内联成员函数实例化。两种情况下, 对这些函数的链接都是局部的。
预编译头文件: -xpch	本发行版本的编译器新增了预编译头文件的特性。预编译头文件的作用是减少源代码共享同一组包含文件的应用程序的编译时间, 而且这些包含文件往往有大量的源代码。预编译头文件的工作机理是, 首先从一个源文件收集一组头文件信息, 然后在重新编译该源文件或者其它有同样头文件的源文件时就可以使用这些收集到的信息。要使用这个特性, 需要指定 -xpch 和 -xpchstop 选项, 并同时使用 #pragma hdrstop 指令。
使用多个处理器: -xjobs=n	指定 -xjobs=n 选项用来设定编译器生成多少个进程来完成它的任务。在多 CPU 机器上, 该选项可以减少生成时间。目前, -xjobs 只在与 -xipo 选项一起使用时才有效。如果指定 -xjobs=n, 过程间调用优化器在编译不同的文件时会用 n 作为它能启动的代码生成器实例的最大数。

表 2-7 列出了支持更便捷移植的 C++ 编译器的新特性:

表 2-7 支持更便捷移植的 C++ 语言新特性

特性	说明
简化移植: -xmemalign	使用 -xmemalign 选项用来控制编译器对数据对齐的假设。通过控制可能会出现非对齐内存访问的代码和出现非对齐内存访问时的处理程序, 可以更轻松的将程序移植到 Solaris 操作系统 (OS)。-xmemalign 选项也用来优化过分对齐的数据, 以及访问过分压缩的结构。
设置 char 的符号: -xchar	有些系统将字符类型定义成无符号类型, -xchar[={signed unsigned u}] 选项使得易于迁移这些系统上的程序。如果不是从这样的系统中迁移, 最好不要使用该选项。只有那些依赖字符类型符号的程序才需要重写, 它们要改写成显式指定有符号或者无符号。
调试移植代码: -xport64	使用新的 -xport64 选项来帮助将代码移植到 64 位环境。具体来说, 该选项会对以下情况提示警告信息: 值的截断 (包括指针), 符号扩展以及对位压缩的更改。将代码从诸如 V7 (ILP32) 的 32 位体系结构移植到诸如 V9 (LP64) 的 64 位体系结构时, 这些更改是很通常的。如果数据截断是由显式类型转换引起, 则附加选项当显式类型转换导致数据截断时, 也可以使用 -xnocastwarn 禁用 64 位编译模式下的截断警告。

表 2-8 列出了支持增强性能的 C++ 编译器的新特性：

表 2-8 增强性能的 C++ 语言新特性

特性	说明
支持数据线程局部存储的链接程序： -xthreadvar (SPARC)	<p>可以使用编译器新增的链接程序支持的线程局部存储特性来完成以下功能：</p> <ul style="list-style-type: none">• 快速实现 POSIX 接口中的分配线程数据。• 将多进程程序转换成多线程程序。• 将使用线程局部存储的 Windows 应用程序移植到 Solaris。• 快速实现 OpenMP 中的线程私有变量。 <p>通过声明线程局部变量，编译器现在可以支持线程局部存储。这个声明包括一个普通的变量声明和额外的 <code>__thread</code> 变量说明符，另外还需要在命令行加上 <code>-xthreadvar</code> 选项。</p>
减少缺页：-xF	<p>使用 <code>-xF</code> 新特性可以使链接程序对变量和函数进行重组以达到最优。这有助于解决以下会对运行时性能产生负面影响的问题：</p> <ul style="list-style-type: none">• 在内存中存放位置很近的无关变量会造成缓存和页的争用。• 在内存中存放位置很远的相关变量会造成过大的工作集。• 未用到的弱变量副本会造成过大的工作集，从而降低有效数据密度。
新的 pragmas	<p>C++ 编译器目前支持四个新的 <code>pragma</code> 命令，它们可以增强代码的优化。有关以下 <code>pragma</code> 命令的全部信息，请参见《C++ 用户指南》：</p> <ul style="list-style-type: none">• <code>#pragma does_not_read_global_data</code>• <code>#pragma does_not_return</code>• <code>#pragma does_not_write_global_data</code>• <code>#pragma rarely_called</code>
增强运行时： -xlinkopt	<p>当指定 <code>-xlinkopt</code> 选项时，C++ 编译器可以对可重定位目标文件进行链接时优化。请参阅 CC(1) 手册页。</p> <p>指定 <code>-xlinkopt</code> 选项时，编译器会在链接时进行一些额外的优化，并且不会修改所链接的 <code>.o</code> 文件。这些优化仅仅在可执行程序中出现。当编译整个程序并且使用配置文件反馈时，<code>-xlinkopt</code> 选项才最有效。</p>
增强运行时： -xpagesize= <i>n</i>	<p>使用 <code>-xpagesize=<i>n</i></code> 选项设置用于堆栈和堆的首选页大小。<i>n</i> 可以是 8K、64K、512K、4M、32M、256M、2G、16G 或缺省。您必须为目标平台上的 Solaris 操作环境指定有效的页面大小，诸如 <code>getpagesize(3C)</code> 的返回值。如果不指定有效的页面大小，运行时该请求就会被忽视。您可以使用 <code>pmap(1)</code> 或 <code>meminfo(2)</code> 来决定目标平台的页面大小。</p> <p>该功能只在 Solaris 9 OS 中可用。使用该选项编译的程序将无法链接到早期的 Solaris OS 环境中。</p>

表 2-9 列出了 C++ 编译器的新增错误和警告控制：

表 2-9 C++ 编译器的新增错误和警告控制

特性	说明
过滤警告消息： -erroff	使用新增的 -erroff 选项可以禁止编译器前端输出警告消息。错误消息和驱动程序输出的消息都不会受到影响。您还可以使用 -erroff 挑选出特定的警告消息，以便只有它被禁止，或者只有它可以输出。
终止编译：-errtags, -errwarn	现在使用 -errtags 和 -errwarn 编译选项可以使编译器在输出某条特定警告后终止。首先指定 -errtags=yes 来获取该特定警告的标记，然后指定 -errwarn=tag，其中 tag 是 -errtags 为该警告消息返回的唯一标识符。 也可以通过指定 -errwarn=%all 使编译器输出任何警告时都终止编译过程。另见 CC(1) 手册页中的 -xwe。
增强的标准库名称过滤 特性：-filt= [no%]stdlib	缺省情况下，将设置 -filt=[no%]stdlib 选项，用来简化链接程序和编译器输出的错误消息中来自标准库的名称。这能帮助您更容易识别出标准库函数名。指定 -filt=no%stdlib 关掉过滤特性。

Fortran 编译器

Sun ONE Studio 8, 编译器集合发行版本提供了与传统 Fortran 77 程序兼容的 Fortran 95 编译器 f95。关于将传统的 Fortran 77 程序移植到 Fortran 95 编译器的详细信息, 请参阅《Fortran 用户指南》中的“FORTRAN 77 兼容性: 迁移到 Fortran 95”一章。

表 2-10 列出了 Fortran 95 编译器的新特性。有关详细信息, 请参阅《Fortran 用户指南》、《Fortran 编程指南》和《Fortran 库参考》。

表 2-10 Fortran 95 编译器新特性

特性	选项	说明
Fortran 2000 特性		<p>在本 Fortran 95 编译器发行版本中, 已实现了 Fortran 2000 草案标准中的以下特性, 并且在 http://www.dkuug.dk/jtc1/sc22/open/n3501.pdf 可以找到它们的 PDF 格式:</p> <ul style="list-style-type: none">• 异常和 IEEE 算法• 与 C 的互操作性• PROTECTED 属性• ASYNCHRONOUS I/O 说明符
增强与传统 f77 的兼容性		<p>很多新特性增强了 Fortran 95 编译器与传统的 Fortran 77 编译器 f77 的兼容性。这其中包括:</p> <ul style="list-style-type: none">• 变量格式表达式 (VFE's)• 长标识符• -arg=loc• -vax 编译器选项
I/O 错误处理程序		<p>两个新函数使用户可以为逻辑单元上的带格式输入指定自己的错误处理例程。当检测到格式错误时, 运行时 I/O 库会调用特定的由用户提供的处理例程, 同时将数据指向输入行中导致错误的字符。处理例程可以提供一个新字符并允许 I/O 操作在检测到错误的点上使用新字符后继续; 或者采用缺省的 Fortran 错误处理操作。</p> <p>新例程 SET_IO_ERR_HANDLER(3f) 和 GET_IO_ERR_HANDLER(3f) 是模块化的子例程, 并需要在调用这些例程的例程中使用 USE SUN_IO_HANDLERS。要详细了解这些例程, 请参手册页。</p>

表 2-10 Fortran 95 编译器新特性 (后续)

特性	选项	说明
无符号整数		<p>在本发行版本中, Fortran 95 编译器接受一种新的数据类型: UNSIGNED, 作为对语言的一种扩展。四种 KIND 参数值可以使用 UNSIGNED: 1、2、4 和 8, 分别对应 1、2、4 和 8 字节无符号整数。</p> <p>无符号整常数的形式是以大写或小写字母 U 结尾的数字串, 后面可以跟一个下划线和 KIND 参数。</p>
首选栈 / 堆页面大小	-xpagesize	<p>新的编译器选项, -xpagesize, 使运行中的程序可以在程序启动时设定首选栈和堆的页面大小。例如, -xpagesize=4M 将首选 Solaris 9 操作系统 (OS) 堆栈和堆的页面大小设定为 4 MB。从预先设定的数值集中选择。</p> <p>栈或堆的页面大小可以通过 -xpagesize_stack 和 -xpagesize_heap 分别设定。</p> <p>该功能只在 Solaris 9 OS 中可用。使用该标志编译的程序将无法链接到早期的 Solaris OS 环境中。)</p>
更快的分析	xprofile_ircache=path	<p>本发行版本引入了新的命令行选项 -xprofile_ircache= path, 来加速配置文件反馈中的 use 编译阶段。指定该标志后, 编译器会在收集编译阶段 -xprofile=collect 将中间数据保存到路径中, 以便后面的 -xprofile=use 阶段重新使用, 无须再次生成该信息。对于大型程序, 这可以在 -xprofile=use 阶段节省大量的编译时间。</p>
增强的“已知库”	-xknown_lib	<p>-xknown_lib 选项已增强, 从基本线性代数子程序库 (BLAS) 中加入了更多例程, 同时引入了三个子选项。编译器能够识别选择 BLAS 库例程的调用, 并且能为 Sun Performance Library 实现自由地进行正确优化。</p>

表 2-10 Fortran 95 编译器新特性 (后续)

特性	选项	说明
链接时优化	-xlinkopt	使用新的 -xlinkopt 标志编译和链接可以调用一个后优化器, 在链接时生成的二进制目标代码中运用一些高级性能优化。 当与配置文件反馈一起用于编译整个程序时, 这个选项非常有效。
局部变量的初始化	-xcheck=init_local	这是对 -xcheck 选项标志新的扩展, 启用了特殊的局部变量初始化。使用 -xcheck=init_local 编译时, 如果一个变量在程序赋值前使用, 编译器会使用一个容易导致算术异常的数值来初始化局部变量。由 ALLOCATE 语句分配的内存也会以这种方式初始化。SAVE 变量、模块变量和 COMMON 块中的变量不会被初始化。
增强的 -openmp 选项	-openmp	-openmp 选项标记经过增强, 便于调试 OpenMP 程序。要使用 dbx 调试 OpenMP 应用程序, 请使用 -openmp=noopt -g 进行编译。 然后就可以使用 dbx 在并行区内设置断点, 并显示变量的内容。
多进程编译	-xjobs= <i>n</i>	带 -xipo 指定 -xjobs= <i>n</i> , 过程间优化器调用至多 <i>n</i> 个代码生成器实例来编译命令行所列的文件。该选项可以大大缩短多 CPU 机器上较大应用程序的生成时间。
使用 PRAGMA ASSUME 进行断言	-xassume_control	ASSUME pragma 是本发行版本编译器的新特性。该 pragma 提示编译器在编译过程中某些点上的条件程序员确信为真。这有助于编译器更好地优化代码。程序员也可以使用断言在程序运行时检查程序的合理性。新的 -xassume_control 标志决定处理 ASSUME pragmas 的方式。
显式线程化程序的 OpenMP 支持。	-xopenmp	本发行版本中的 OpenMP API 实现支持显式线程化的程序。

dbx 命令行调试程序

表 2-11 列出了本发行版本 dbx 命令行调试程序的新特性。关于这些特性的更多信息，请参阅《使用 dbx 调试程序》手册。

表 2-11 dbx 新特性

特性	说明
调试混合语言代码的程序	dbx 现在支持以下 C99 语言类型： <ul style="list-style-type: none">• complex• 虚数• 双精度复数• 双精度虚数• 长双精度复数• 长双精度虚数 您可以打印包含这些类型的变量和表达式的值。
对调试 OpenMP 程序的支持	dbx 现在支持调试 Fortran 95、C++ 和 C 中的 OpenMP 程序。dbx 可以正确显示 Fortran 95、C++ 和 C 编译器所产生的 OpenMp 编码的线程、堆栈、函数、参数和变量。
detach 命令的新的 -stop 选项	detach -stop 命令将 dbx 从目标程序中分离出来，同时将进程保留在停止状态。此 -stop 选项允许临时使用其它建立于 /proc 基础上的调试工具，这些工具可能由于专用访问而被阻止。
新的 -resumeone 事件修饰符	事件处理程序的新 -resumeone 修饰符帮助多线程程序中的函数调用条件。

区间运算

在此编译器集合发行版本中没有新的区间运算特性。

Sun Performance Library

Sun Performance Library™ 是一系列优化的、高速的数学子例程，它们用于解决线性代数问题和其它数值密集问题。Sun Performance Library 是建立在可从 Netlib 获得 (<http://www.netlib.org>) 的公共领域应用集合之上。将这些例程增强并捆绑后即成为 Sun Performance Library。

表 2-12 列出了本发行版本的 Sun Performance Library 新特性。更多信息请参阅 *Sun Performance Library User's Guide* 和 3p 手册页。

表 2-12 Sun Performance Library 新特性

特性	说明
性能改善	<p>该发行版本的 Sun Performance Library 包括以下性能改善。</p> <ul style="list-style-type: none">• BLAS 和 FFT 的性能改善包括：改善的 US-III 小问题尺寸的 GEMM 性能，以及在 V9 库中使用 32 位 FFT 例程时改善的小问题尺寸的 FFT 性能• 稀疏处理的性能改善包括：增强的 Sun Performance Library 稀疏处理的单 CPU 性能，以及增强的并行 Sun Performance Library 稀疏处理• 稀疏 BLAS 性能改善包括：并行稀疏矩阵和向量间运算，以及改善的小问题尺寸的性能
可移植库性能	<p>本 Sun Performance Library 发行版本内部更改了如何简化最佳性能。在运行时，用于优化执行正在运行的 SPARC 硬件平台的 Sun Performance Library 的一个版本可以动态装入。这种情况只能在链接 Sun Performance Library 共享库版本时才发生，这是缺省设置。</p>
稀疏处理的新特性	<p>稀疏处理现在包括 Hermitian 正有限向量支持。</p>
合并的并行模型	<p>本发行版本的 Sun Performance Library 包括合并的并行模型，减少了 Sun Performance Library 发送的库的数量，并且减少了 Sun Performance Library 的大小。</p> <p>合并的并行模型简化了 Sun Performance Library 串行或并行行为的链接。</p>
区间 BLAS 手册页已移到 man3pi 文件夹	<p>区间 BLAS 手册页已经移到 man3pi 文件夹。</p> <p>对于 Fortran 95 接口和在每一个区间 BLAS 例程中使用的参数类型，请参阅 3pi 手册页中独立例程的那一段。例如，若要显示 constructv_i.3pi 例程的手册页，请输入</p> <pre>man -s 3pi constructv_i</pre> <p>例程的名字必须小写。</p>

dmake

dmake 是一个命令行工具，兼容 make(1)。dmake 能够以分布、并行或串行模式生成目标。如果使用标准 make(1) 公用程序，在对 makefile 修改的情况下对 dmake 的转变很小。dmake 是 make 公用程序的一个超集。对于嵌套 make，如果一个顶级的 makefile 称为 make，您需要使用 \$(MAKE)。dmake 分析 makefile 并决定能够并行生成哪些目标，以及在您设置的主机分布这些目标的生成版本。详细信息请参阅 man dmake。

表 2-13 dmake 新特性

特性	说明
dmake 内存使用减少了	结果由多种因素决定，内存堆的使用已减小 50% 到 60%。
增加了一致性	dmake 现在同 Solaris make 一致
dmake 现在自动调节并行作业限制以防止过载	环境变量 DMAKE_ADJUST_MAX_JOBS 可以设置为自动调节并行作业的限制以防止过载。 <ul style="list-style-type: none">• 如果设置为 YES，dmake 可根据当前系统的负载来调节并行作业限制。如果系统没有过载，那么 dmake 会使用用户定义的限制。如果系统过载，那么 dmake 会设置当前限制，使其少于用户定义的限制。如果没有设置此变量，那么 dmake 可根据当前系统的负载来调节并行作业限制。这是 dmake 的缺省设置。• NO 使得 dmake 关闭自动调节机制。

性能分析工具

表 2-14 列出了 Sun ONE Studio 8，编译器集合发行版本性能分析器工具的新数据集合和演示特性。更多信息请参阅以下手册页：

- `collect(1)`
- `collector(1)`
- `er_print(1)`
- `libcollector(3)`

表 2-14 性能分析工具新特性

特性	说明
支持 Java™ 程序基于时钟的分析和硬件计数器溢出分析	现在 Java 编程语言完全支持基于时钟的分析和硬件计数器溢出分析，以及同步延迟跟踪和内存分配跟踪。收集的数据既用于目标的机器表示，也用于 Java 表示。此外提供了 <code>libcollector</code> 的 Java API。
基于时钟的分析	基于时钟的分析不再限定为支持更高分辨率时钟的 Solaris 操作环境版本系统时钟分辨率的倍数。使用不带参数的 <code>collect</code> 命令时会返回支持的配置区间范围。
装入对象的归档	使用 <code>collect</code> 命令或 <code>dbx collector archive</code> 命令的 <code>-A</code> 选项可以控制装入对象的归档。
暂停和恢复数据集合的应用程序编程接口	提供了暂停和恢复独立线程数据集合的 API。
为内存访问事件计数的硬件计数器	对于为内存访问事件计数的硬件计数器来说，在计数器名称加上前缀 “+” 会激活由程序计数器的收集器和触发该事件的虚拟地址的搜索。
CPU 过滤	按 CPU 过滤已增加到性能分析器和 <code>er_print</code> 公用程序。该功能对早于 Solaris 操作环境版本 9 的版本不可用。它是在 <code>er_print</code> 命令 <code>cpu_select</code> 和 <code>cpu_list</code> 中实现的。
显示列表	性能工具显示列表按照源代码行和程序计数器的度量值排序。这些列表显示在性能分析器的 [行] 标签和 PC 标签中，是 <code>er_print</code> 公用程序使用 <code>lines</code> 命令和 <code>pcs</code> 命令生成的。性能分析器的 [汇总] 标签显示了所选源代码行的全部度量。使用 <code>lsummary</code> 命令和 <code>psummary</code> 命令可以将行和程序计数器的汇总面板显示在 <code>er_print</code> 公用程序中。

表 2-14 性能分析工具新特性 (后续)

特性	说明
[时间线] 标签	<p>[时间线选项] 对话框已与 [设置数据表示] 对话框合并为 [时间线] 标签。</p> <p>[时间线] 标签可以显示 LWP、线程或 CPU 的数据栏。使用 [设置数据表示] 对话框的 [时间线] 标签从以上三项中选择一项并显示其数据。</p> <p>[时间线] 标签中的调用栈可以按照根函数或末端函数对齐，此外也可以设置可见框架的数量。使用 [设置数据表示] 对话框的 [时间线] 标签进行选择。</p>
对象的选择	<p>对象的选择已扩展到包含源代码行、程序计数器以及函数。所选对象显示在菜单栏中，其度量显示在 [汇总] 标签中。从当前标签导航到另一标签时，将显示所选对象。特别地，切换到源码或反汇编的操作会定位到所选行或所选指令，而不是始终定位到所选函数的第一行或第一条指令。</p>
后续进程中的实验	<p>装入初始进程实验的同时会自动装入后续进程的实验，却禁用了它们的数据显示。必须使用性能分析器中的 [过滤数据] 对话框或 <code>er_print</code> 公用程序中的 <code>experiment_select</code> 命令来启用后续实验的数据显示。</p>
[泄漏列表] 标签	<p>[泄漏列表] 标签以图形方式显示了泄漏和分配的数据，并能够在泄漏（或分配）的调用栈和调用栈函数的源码（或反汇编）间进行导航。</p>
Java 模式	<p>用 Java 编程语言编写的应用程序中的实验，在显示时可能会带有设置为 <code>on</code>、<code>expert</code> 或 <code>off</code> 的 Java 模式。</p>

文档

本节描述了 Sun ONE Studio 8，编译器集合文档的新特性。

- 《使用 `dbx` 调试程序》手册中有一章名为“调试 OpenMP 应用程序”。该章描述了如何使用 `dbx` 命令行调试程序来调试使用显式并行性 OpenMP 接口的应用程序。
- 从 docs.sun.com web 站点可以获取 *FORTRAN 77 Language Reference* 的旧发行版本。该手册未对本发行版本进行更新。
- 残障人士用户通过辅助技术同样可以阅读 Sun ONE Studio 8，编译器集合的产品文档。有关更多信息请参阅第 10 页的“使用易读格式的文档”。

