



新增功能

Sun™ Studio 9

Sun Microsystems, Inc.
www.sun.com

文件号码 817-7890
2004 年 8 月

关于本文档的建议请发送到: <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。

对于本文档中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家 / 地区申请的一项或多项其他专利或待批专利。

本文档及其所述产品按照限制其使用、复制、分发和反编译的许可证进行分发。未经 Sun 及其许可方（如果有）的事先书面许可，不得以任何形式、任何手段复制本产品或文档的任何部分。

第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家 / 地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、docs.sun.com、Solaris 和 Java 是 Sun Microsystems, Inc 在美国和其他国家 / 地区的商标或注册商标。

所有 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家 / 地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

OPEN LOOK 和 Sun™ 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

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、适用性和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。



请回收



Adobe PostScript

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	6
访问编译器和工具文档	8
访问相关的 Solaris 文档	10
开发人员资源	11
与 Sun 技术支持联系	11
Sun 欢迎您提出意见和建议	11
Sun Studio 9 新增功能与增强特性	13
C 编译器	14
C++ 编译器	19
Fortran 编译器	23
命令行调试程序 dbx	27
区间运算	28
Sun 性能库	28
dmake	29
性能分析工具	29
集成开发环境 (IDE)	31
文档	31

开始之前

《新增功能》介绍了此 Sun™ Studio 9 软件发行版的新增功能，其中包括 C、C++ 和 Fortran 编译器、库、以及工具中的新增功能。

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 <code>.login</code> 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。 您必须是超级用户才能执行此操作。
<code>AaBbCc123</code>	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
	分隔变量的 " " 或 "-" 符号，只能选择其一。	B{dynamic static}	Bstatic
:	与逗号一样，冒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=fl[,...fn]	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 7 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 目录下，请向系统管理员获取系统中的等效路径。

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 shel，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 shel，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 9 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 9 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 9 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 目录下，请向系统管理员获取系统中的等效路径。

- 大多数的手册都可以从 docs.sun.comSM web 站点上获得。下列书目只能从您所安装的软件中找到：
 - 《标准 C++ 库类参考》
 - 《标准 C++ 库用户指南》
 - 《Tools.h++ 库类参考》
 - 《Tools.h++ 用户指南》

- 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、产品或其他资料承担责任。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 目录下，请向系统管理员获取系统中的等效路径。

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">• 《标准 C++ 库类参考》• 《标准 C++ 库用户指南》• 《Tools.h++ 库类参考》• 《Tools.h++ 用户指南》	HTML，位于安装的软件中的文档索引 file:/opt/SUNWspro/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/SUNWspro/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/SUNWspro/docs/index.html` 和 `http://docs.sun.com` 上获得。如果软件没有安装在 `/opt` 下，请向系统管理员获取系统中的等效路径。

文档标题	描述
<i>Fortran 编程指南</i>	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。
<i>Fortran 库参考</i>	详细说明 Fortran 库和内部例程
<i>Fortran 用户指南</i>	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。
<i>C 用户指南</i>	描述 cc 编译器的编译时环境和命令行选项。
<i>C++ 用户指南</i>	描述 CC 编译器的编译时环境和命令行选项。
<i>数值计算指南</i>	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	描述
Solaris 参考手册集合	请参阅手册页部分的标题。	提供关于 Solaris™ 操作环境的信息。
Solaris 软件开发者集合	<i>链接程序和库指南</i>	描述了 Solaris™ 链接编辑器和运行时链接程序的操作。
Solaris 软件开发者集合	<i>多线程编程指南</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>

Sun 欢迎您提出意见和建议

Sun 致力于提高文档质量，并欢迎您提出宝贵的意见和建议。请将您的意见发送至以下 URL

<http://www.sun.com/hwdocs/feedback>

请提供文档的文件号码 (817-7890)。

当您提供意见和建议时，可能需要在表单中提供文档英文版本的标题和文件号码。本文档英文版本的文件号码和标题是：817-6691, What's new。

Sun Studio 9 新增功能与增强特性

Sun™ Studio 9 将代替 Sun™ Studio 8。Sun Studio 9 发行版中的新增功能包括对以下编译器、库及工具的更新：

- C 编译器
- C++ 编译器
- Fortran 编译器
- Sun 性能库
- 分布式 make 公用程序，dmake
- dbx 命令行调试器
- 性能分析工具
- 集成开发环境 (IDE)
- 文档

在大多数章节中，我们提供了一个表，其中列出了该组件的新增功能。该表分两列，左边一列提供了功能的简短说明，而右边一列显示了更详细的说明。

注—要查找本章中介绍的 Sun Studio 9 文档，另请参见随产品软件一起安装的、位于 `/opt/SUNWspro/docs/index.html` 的文档索引。如果软件不是安装在 `/opt` 目录中，请与系统管理员联系，以了解系统或网络上的等效路径。

C 编译器

本节列出了该发行版的 C 编译器的新增功能。下表列出了这些新增功能：

- 表 1-1 一般增强特性
- 表 1-2 增强的硬件平台支持
- 表 1-3 改进的性能和优化选项
- 表 1-4 通过 Lint 公用程序进行新的安全检查

有关本节中引用的特定编译器选项的详细信息，另请参见《C 用户指南》或 cc(1) 手册页。

表 1-1 列出了 C 编译器的一般增强特性。

表 1-1 C 编译器的一般增强特性

功能	描述
其他 C99 功能的实现	<p>该发行版增加了对以下 ISO/IEC 9899:1999（在本文中简称 C99）功能的支持。下表仅详细列出此发行版中实现的 C99 功能，它是所有实现的 C99 功能的子集。有关 C 编译器的过去和当前发行版中实现的所有 C99 功能的完整列表，另请参见《C 用户指南》。我们为此 Sun Studio 9 发行版中支持的每个新项列出了 C99 标准中各小节的编号。</p> <ul style="list-style-type: none">• 5.2.4.2.2: 支持 FLT_EVAL_METHOD 宏。该宏和新的 -flt_eval 编译选项一起，确定了编译器是否将浮点表达式作为 long double 进行计算，或是否根据表达式中类型与常量的组合来对其进行计算。• 6.4.3: 支持四位和八位通用字符名 (UCN)，它们可用于标识符、字符常量和字符串文字中，以指定 C 基本字符集中不存在的字符。UCN \Uxxxxxxxx 指定了其八位短标识符（根据 ISO/IEC 10646 的规定）为 xxxxxxxx 的字符。同样，通用字符名 \unxxx 指定了其四位短标识符为 nxxx（其八位短标识符为 0000xxxx）的字符。• 6.7.4: 支持内联函数和 extern 内联函数• 6.7.8: 支持指定的初始化函数，它为初始化稀疏数组和结构提供了一种方法，这在数值和系统编程中很常见。
通过新的 -features 编译选项改进了与旧二进制的兼容性	<p>现在，您无需更改旧二进制的行为即可将旧的 C 和 C++ 二进制（C/C++ 5.6 之前的版本）与新的 C 和 C++ 二进制链接在一起。如果希望新的二进制与包含 extern 内联函数的旧 C 和 C++ 二进制之间进行兼容，请使用 -features=no%extinl 编译选项。</p> <p>要获得符合标准的行为，必须使用当前编译器对旧代码重新进行编译。</p>

表 1-1 C 编译器的一般增强特性 (续)

功能	描述
较大的从属线程缺省栈大小	现在, 从属线程的缺省栈大小更大一些。所有从属线程的栈大小都相同。缺省情况下, 对于 32 位应用程序, 栈大小为 4 兆字节; 对于 64 位应用程序, 栈大小为 8 兆字节。该大小使用环境变量 STACKSIZE 进行设置。
改进的 <code>-xprofile</code> (SPARC®)	<code>-xprofile</code> 选项进行了以下改进: <ul style="list-style-type: none">• 支持文件配置共享库• 使用 <code>-xprofile=collect -mt</code> 的线程安全配置文件收集• 改进了对单个配置文件目录中多个程序或共享库文件配置的支持。 通过设置 <code>-xprofile=use</code> , 编译器能够在包含具有非唯一基名的多个对象文件数据的配置文件目录中查找配置文件数据。如果编译器无法找到对象文件的配置文件数据, 它将提供一个新选项 <code>-xprofile_pathmap=collect-prefix:use-prefix</code> 。
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应用程序, 请指定 <code>-xustr=ascii_utf16_ushort</code> 。换句话说, 如果您的代码包含由 16 位字符组成的字符串文字, 请使用此选项。如果不使用该选项, 则编译器既不会生成、也无法识别 16 位字符的字符串文字。此选项使系统能够将 <code>U"ASCII_string"</code> 字符串文字识别为无符号短类型数组。由于这类字符串还不属于任何标准, 因此此选项可以识别非标准 C。
自动生成的预编译头文件	该版本的 C 编译器扩展了预编译头文件功能, 以便使编译器的某个部分能够自动生成预编译头文件。您还可以手动生成预编译头文件, 但是, 如果您对编译器的新增功能感兴趣的话, 另请参见 <code>cc(1)</code> 手册页中 <code>-xpch</code> 选项的相关说明, 以了解更多信息。另请参见 <code>CCadmin(1)</code> 手册页。

表 1-2 列出了 C 编译器中支持快速编译的新增功能。

表 1-2 增强的硬件平台支持

功能	描述
支持 SPARC® 平台的更多标志	现在, <code>-xchip</code> 和 <code>-xtarget</code> 选项支持 <code>ultra3i</code> 和 <code>ultra4</code> 作为其值, 因此您可以生成针对 UltraSPARC IIIi 和 UltraSPARC IV 处理器进行了优化的应用程序。
支持 x86 平台的更多标志	<p>C 编译器支持 <code>-xarch</code>、<code>-xtarget</code> 和 <code>-xchip</code> 编译选项的新标志用于将在 x86 平台上运行的代码。这些新标志经过精心设计, 可充分利用 Pentium 3 和 Pentium 4 芯片以及 Solaris™ 软件对 x86 平台上 <code>sse</code> 和 <code>sse2</code> 指令的支持。新标志如下所示:</p> <ul style="list-style-type: none">• <code>-xchip=pentium3</code> 针对 Pentium 3 类型处理器进行了优化• <code>-xchip=pentium4</code> 针对 Pentium 4 类型处理器进行了优化• <code>-xarch=sse</code> 将 <code>sse</code> 指令集添加到 <code>pentium_pro</code> 指令集体系结构中• <code>-xarch=sse2</code> 将 <code>sse2</code> 指令集添加到 <code>sse</code> 允许的指令集中• <code>-xtarget=pentium3</code> 设置了 <code>-xarch=sse</code>、<code>-xchip=pentium3</code> 和 <code>-xcache=16/32/4:256/32/4</code>• <code>-xtarget=pentium4</code> 设置了 <code>-xarch=sse2</code>、<code>-xchip=pentium4</code> 和 <code>-xcache=8/64/4:256/128/8</code> <p>您可以按照下面这些原则来确定哪种选项组合最适合您的编译要求:</p> <ul style="list-style-type: none">• 如果您要构建在采用 Solaris 9 更新版 6 或更高版本的 Pentium 3 或 Pentium 4 机器上运行的应用程序, 请根据情况使用 <code>-xtarget=pentium3</code> 或 <code>-xtarget=pentium4</code> 进行编译。• 如果您要构建在采用 Solaris 9 更新版 5 或更低版本的 Pentium 3 或 Pentium 4 机器上运行的应用程序, 请设置 <code>-xarch=pentium_pro</code> (而并非您所认为的 <code>pentium3</code> 或 <code>pentium4</code>), 因为 Solaris 9 更新版 5 或更低版本的操作系统不支持 <code>sse</code> 和 <code>sse2</code> 指令。使用 <code>-xtarget=pentium3</code> 或 <code>-xtarget=pentium4</code> 时, 请将 <code>-xchip</code> 和 <code>-xcache</code> 设置为使用相同的值, 具体取决于目标机器。• 如果在目标机器上进行构建, 则指定 <code>-fast</code>、<code>-xarch=native</code> 或 <code>-xtarget=native</code> 将自动扩展至上述相应的 <code>-xchip</code>、<code>-xarch</code> 和 <code>-xtarget</code> 标志设置。

表 1-3 列出了支持增强性能的 C 编译器的新特性。

表 1-3 改进的性能和优化选项

特性	描述
编译器选项的全新缺省设置及其扩展	<p>以下编译选项的缺省设置已更改:</p> <ul style="list-style-type: none">• SPARC[®] 平台上的 <code>-xarch: v8plus</code>。对于当前正在使用的几乎所有机器来说, 新的缺省设置可产生更高的运行时性能。但是, 专门用于在 UltraSPARC 之前的计算机上进行部署的应用程序将不再使用该缺省选项来执行; 此时应使用 <code>-xarch=v8</code> 进行编译, 以确保应用程序能够在 UltraSPARC 之前的计算机上执行。• SPARC[®] 平台上的 <code>-xcode: abs44</code> (用于 v9) 和 <code>abs32</code> (用于 v8)。• SPARC[®] 平台上的 <code>-xmemalign: 8i</code> (用于 v8) 和 <code>8s</code> (用于 v9)。• SPARC[®] 平台上的 <code>-xprefetch: auto,explicit</code>。此项更改将对本来具有非线性内存访问模式的应用程序造成负面影响。要禁用此更改, 请指定 <code>-xprefetch=no%auto,no%explicit</code>。 <p>以下选项和宏的扩展已更改:</p> <ul style="list-style-type: none">• 现在, <code>-fast</code> 选项将在其扩展中包括新的选项 <code>-xlibmopt</code> (如下所示)。• 现在, <code>-O</code> 宏将扩展到 <code>-xO3</code>, 而不是 <code>-xO2</code>。缺省设置的更改将产生更高的运行时性能。但是, <code>-xO3</code> 可能不适用于那些依赖于将所有变量自动视为 <code>volatile</code> 的程序。可能依赖于此假定的典型程序是那些实现其自己的同步基元的设备驱动程序以及早期多线程应用程序。其解决办法是使用 <code>-xO2</code> 进行编译, 而不使用 <code>-O</code>。
新的优化编译选项	<p>新编译选项如下所示:</p> <ul style="list-style-type: none">• <code>-xlibmopt</code> 和 <code>-xnolibmopt</code>: <code>-xlibmopt</code> 选项使编译器能够使用优化的数学例程库。在使用 <code>-xlibmopt</code> 选项时, 必须通过指定 <code>-fround=nearest</code> 来使用缺省舍入模式。数学例程库针对性能进行了优化, 并且通常会生成速度更快的代码。这样生成的代码可能与普通数学库生成的代码稍有不同, 不同之处通常在最后一位上。 <p>通过在命令行上指定新的 <code>-xnolibmopt</code> 选项, 您可以显式关闭此库。</p> <ul style="list-style-type: none">• <code>-xipo_archive</code>: 新的 <code>-xipo_archive</code> 选项使编译器能够通过使用 <code>-xipo</code> 进行编译以及在生成可执行文件之前驻留在归档库 (.a) 中的对象文件来优化传递给链接程序的对象文件。库中包含的、在编译进程中进行了优化的任何对象文件都将被其已优化的版本替换。• <code>-xprefetch_auto_type</code>: 使用新的 <code>-xprefetch_auto_type</code> 选项能够以与生成直接内存访问预取相同的方式来生成由选项 <code>-xprefetch_level=[1 2 3]</code> 指示的间接循环预取。 <p>诸如 <code>-xdepend</code>、<code>-xrestrict</code> 和 <code>-xalias_level</code> 等选项可以改进 <code>-xprefetch_auto_type</code> 的优化优势。它们会影响计算间接预取候选项的主动性, 进而影响自动间接预取插入的主动性, 因为它们有助于更好地消除内存别名信息的歧义。</p>

表 1-4 介绍了 lint 公用程序中包括的新的安全检查功能。

表 1-4 通过 Lint 公用程序进行新的安全检查

特性	描述
用于 lint 的新 <code>-errsecurity</code> 选项	<p>Sun Studio 9 发行版的 lint 公用程序采用了新的安全检查工具。您可以在进行编译之前使用新的 <code>-errsecurity</code> 选项，以检查您的代码是否符合安全要求。</p> <pre>-errsecurity[= {core standard extended %none}]</pre> <p><code>lint -errsecurity=core</code></p> <p>检查源代码结构，这些结构几乎总是不安全或难以验证。这一级别的检查包括：</p> <ul style="list-style-type: none">• 将变量格式字符串与 <code>printf()</code> 和 <code>scanf()</code> 函数系列一起使用• 在 <code>scanf()</code> 函数中使用无界字符串 (<code>%s</code>) 格式• 使用不带安全用法的函数：<code>gets()</code>、<code>cftime()</code>、<code>ascftime()</code>、<code>creat()</code>• <code>open()</code> 与 <code>O_CREAT</code> 的使用不正确 <p>将在这一级别产生警告的源代码视为错误。应更改有问题的源代码。在所有情况下，均可使用更简单、更安全的可选方法。</p> <pre>lint -errsecurity=standard</pre> <p>包括核心级别的所有检查以及可能安全的结构，但存在更好的可选方法。在检查新编写的代码时，建议使用这一级别的检查。这一级别的其他检查包括：</p> <ul style="list-style-type: none">• 使用除 <code>strcpy()</code> 之外的其他字符串复制函数• 使用弱随机数函数• 使用不安全的函数来生成临时文件• 使用 <code>fopen()</code> 来创建文件• 使用可调用 shell 的函数 <p>用新的代码或经过重大修改的代码来替换在这一级别产生警告的源代码。根据导致应用程序不稳定的风险来权衡解决传统代码中的这些警告。</p>

表 1-4 (续) 通过 Lint 公用程序进行新的安全检查

特性	描述
用于 lint 的新 <code>-errsecurity</code> 选项 (续)	<code>lint -errsecurity=extended</code> 包含一套最全面的检查, 包括核心级别和标准级别中的所有内容。此外, 还生成了有关在某些情况下可能不安全的结构的许多警告。这一级别的检查作为查看代码的辅助方法时非常有用, 但不必将其用作可接受源代码必须符合的标准。这一级别的其他检查包括: <ul style="list-style-type: none">• 在循环内部对 <code>getc()</code> 或 <code>fgetc()</code> 的调用• 对易于导致路径名竞争情况的函数的使用• 对 <code>exec()</code> 系列函数的使用• <code>stat()</code> 与其他函数之间的竞争情况 查看在这一级别产生警告的源代码, 以确定可能的安全性问题是否仍然存在。 如果未指定 <code>-errsecurity</code> 的设置, 则编译器会将其设置为 <code>-errsecurity=%none</code> 。如果指定了不带参数的 <code>-errsecurity</code> , 则编译器会将其设置为 <code>-errsecurity=standard</code> 。

C++ 编译器

本节列出了此发行版本中 C++ 编译器的新特性。下表中包括下列新特性:

- 表 1-5 一般增强特性
- 表 1-6 增强的硬件平台支持
- 表 1-7 新的和增强的优化选项

有关在本节中引用的特定编译器选项的更多信息, 另请参见 《C++ 用户指南》或 CC(1) 手册页。

表 1-5 列出了 C++ 编译器（5.6 版）的一般增强特性。

表 1-5 C++ 编译器的一般增强特性

特性	描述
外部链接的内联函数	<p>C++ 标准规定，除非已声明为静态，否则内联（inline）函数类似于非内联函数，具有外部链接。如今，C++ 5.6 首次在缺省情况下就使内联函数具有外部链接。如果内联函数必须在外部生成（例如，需要其地址），则只会会有一个副本被链接到最终程序中。以前，需要副本的每个对象文件都拥有自己的、具有局部链接的副本。</p> <p><code>extern inline</code> 函数的实现可与早期编译器版本创建的二进制文件兼容，也就是说，程序行为与过去一样，均符合标准。旧的二进制可能拥有内联函数的多个局部副本，但新代码最多只能拥有 <code>extern inline</code> 函数的一个副本。</p> <p><code>extern inline</code> 函数的实现可与内联函数的 C99 版兼容，该内联函数使用了此发行版中包含的 C 5.6 编译器。也就是说，按照 <code>extern</code> 内联函数的 C 和 C++ 规则，可以同时 C 和 C++ 文件中定义相同的内联函数，并且外部函数将只有一个副本显示在最终程序中。</p>
增强的 UTF-16 支持	<p>5.5 版的 C++ 编译器引入了对 UTF-16 字符串文字的支持。此发行版扩展了对使用语法 <code>U'x'</code> 的 UTF-16 字符串文字的支持，该语法类似于字符串的 <code>U"x"</code> 语法。必须使用相同的 <code>-xustr</code> 选项才能识别 UTF-16 字符串文字。</p> <p>此外，该发行版还支持 UTF-16 字符和字符串文字中的数值换码符，它们类似于普通字符串文字和字符串中的数值换码符。例如：</p> <pre>U"ab\123ef" // 字符的八进制表示 U'\x456' // 字符的十六进制表示</pre> <p>有关详细信息，另请参见 C++ 手册页 CC(1) 中有关 <code>-xustr</code> 的说明。</p>
自动生成的预编译头文件	<p>该版本的 C++ 编译器扩展了预编译头文件功能，以便使编译器的某个部分能够自动生成预编译头文件。您还可以手动生成预编译头文件，但是，如果您对编译器的新增功能感兴趣的话，另请参见 CC(1) 手册页中 <code>-xpch</code> 选项的相关说明，以了解更多信息。另请参见 CCadmin(1) 手册页。</p>

表 1-6 列出了支持更快速编译的 C++ 编译器新特性。

表 1-6 增强的硬件平台支持

特性	描述
支持 SPARC® 平台的更多标志	现在, <code>-xchip</code> 和 <code>-xtarget</code> 选项支持 <code>ultra3i</code> 和 <code>ultra4</code> 作为其值, 因此您可以生成针对 UltraSPARC IIIi 和 UltraSPARC IV 处理器进行了优化的应用程序。
支持 x86 平台的更多标志	<p>C 编译器支持 <code>-xarch</code>、<code>-xtarget</code> 和 <code>-xchip</code> 编译选项的新标志用于将在 x86 平台上运行的代码。这些新标志经过精心设计, 可充分利用 Pentium 3 和 Pentium 4 芯片以及 Solaris™ 软件对 x86 平台上 <code>sse</code> 和 <code>sse2</code> 指令的支持。新标志如下所示:</p> <ul style="list-style-type: none">• <code>-xchip=pentium3</code> 针对 Pentium 3 类型处理器进行了优化• <code>-xchip=pentium4</code> 针对 Pentium 4 类型处理器进行了优化• <code>-xarch=sse</code> 将 <code>sse</code> 指令集添加到 <code>pentium_pro</code> 指令集体系结构中• <code>-xarch=sse2</code> 将 <code>sse2</code> 指令集添加到 <code>sse</code> 允许的指令集中• <code>-xtarget=pentium3</code> 设置了 <code>-xarch=sse</code>、<code>-xchip=pentium3</code> 和 <code>-xcache=16/32/4:256/32/4</code>• <code>-xtarget=pentium4</code> 设置了 <code>-xarch=sse2</code>、<code>-xchip=pentium4</code> 和 <code>-xcache=8/64/4:256/128/8</code> <p>您可以按照下面这些原则来确定哪种选项组合最适合您的编译要求:</p> <ul style="list-style-type: none">• 如果您要构建在采用 Solaris 9 更新版 6 的 Pentium 3 或 Pentium 4 机器上运行的应用程序, 请根据情况使用 <code>-xtarget=pentium3</code> 或 <code>-xtarget=pentium4</code> 进行编译。• 如果您要构建在采用 Solaris 9 更新版 5 或更低版本的 Pentium 3 或 Pentium 4 机器上运行的应用程序, 请设置 <code>-xarch=pentium_pro</code> (而非您所认为的 <code>pentium3</code> 或 <code>pentium4</code>), 因为 Solaris 9 更新版 5 或更低版本的操作系统不支持 <code>sse</code> 和 <code>sse2</code> 指令。使用 <code>-xtarget=pentium3</code> 或 <code>-xtarget=pentium4</code> 时, 请将 <code>-xchip</code> 和 <code>-xcache</code> 设置为使用相同的值, 具体取决于目标机器。• 如果在目标机器上进行构建, 则指定 <code>-fast</code>、<code>-xarch=native</code> 或 <code>-xtarget=native</code> 将自动扩展至上述相应的 <code>-xchip</code>、<code>-xarch</code> 和 <code>-xtarget</code> 标志设置。

表 1-7 列出了支持更轻松移植的 C++ 编译器的新特性：

表 1-7 新的和增强的优化选项

特性	描述
编译器选项的全新缺省设置及其扩展	<p>以下编译选项的缺省设置已更改：</p> <ul style="list-style-type: none">• SPARC® 平台上的 <code>-xarch: v8plus</code>。对于当前正在使用的几乎所有的机器来说，新的缺省设置可产生更高的运行时性能。但是，专门用于在 UltraSPARC 之前的计算机上进行部署的应用程序将不再使用该缺省选项来执行；此时应使用 <code>-xarch=v8</code> 进行编译，以确保应用程序能够在 UltraSPARC 之前的计算机上执行。• SPARC® 平台上的 <code>-xcode: abs44</code>（用于 v9）和 <code>abs32</code>（用于 v8）。• SPARC® 平台上的 <code>-xmemalign: 8i</code>（用于 v8）和 <code>8s</code>（用于 v9）• SPARC® 平台上的 <code>-xprefetch: auto, explicit</code>。此项更改将对本来具有非线性内存访问模式的应用程序造成负面影响。要禁用此更改，请指定 <code>-xprefetch=no%auto, no%explicit</code>。 <p>以下宏的扩展已更改：</p> <ul style="list-style-type: none">• 现在，<code>-O</code> 宏将扩展到 <code>-xO3</code>，而不是 <code>-xO2</code>。缺省设置的更改将产生更高的运行时性能。但是，<code>-xO3</code> 可能不适用于那些依赖于将所有变量自动视为 <code>volatile</code> 的程序。可能依赖于此假定的典型程序是那些实现其自己的同步基元的设备驱动程序以及早期多线程应用程序。其解决办法是使用 <code>-xO2</code> 进行编译，而不使用 <code>-O</code>。
新的循环优化编译选项	<p>现在，C++ 编译器支持以下选项，以优化其计算可并行进行的循环。只有当您指定 <code>-xO3</code> 或更高的优化级别时，这些选项才有效。</p> <ul style="list-style-type: none">• <code>-xautopar</code>• <code>-xvector</code>• <code>-xdepend</code> <p>有关详细信息，另请参见 C++ 手册页 CC(1) 中有关 <code>-xautopar</code>、<code>-xvector</code> 和 <code>-xdepend</code> 的说明。</p>
新的特定函数优化级别控制	<p>您可以将 <code>#pragma opt</code> 指令与命令行选项 <code>-xmaxopt</code> 组合在一起，以指定编译器对个别函数应用的优化级别。当您需降低特定函数的优化级别（例如，为了避免类似于消除堆栈的代码增强）或提高特定函数的优化级别时，该组合非常有用。</p>
循环的预取优化	<p><code>-xprefetch_auto_type</code>：使用新的 <code>-xprefetch_auto_type</code> 选项能够以与生成直接内存访问预取相同的方式来生成由选项 <code>-xprefetch_level=[1 2 3]</code> 指示的间接循环预取。</p> <p>诸如 <code>-xdepend</code>、<code>-xrestrict</code> 和 <code>-xalias_level</code> 等选项可以改进 <code>-xprefetch_auto_type</code> 的优化优势。它们会影响计算间接预取候选项的主动性，进而影响自动间接预取插入的主动性，因为它们有助于更好地消除内存别名信息的歧义</p>

表 1-7 新的和增强的优化选项 (续)

特性	描述
限定指针优化	<p>C++ 不支持 C99 中引入的 <code>restrict</code> 关键字。但现在 C++ 编译器接受 C 编译器选项 <code>-xrestrict</code>。</p> <p>该选项作出了有关编译中函数的断言，即指针类型的函数参数并非指相同或重叠的对象。此选项对 C++ 比对 C 稍微更危险一些，因为该断言对 C++ 标准库中的某些函数来说是不正确的。</p>

Fortran 编译器

表 1-8 列出了此发行版的 Fortran 编译器的新功能和增强功能，其中包括以下几个方面：

- Solaris™ OS x86 平台上 f95 的新编译功能
- 改进的运行时性能
- 新的 Fortran 2003 命令行内在函数
- 已更改的 f95 编译器命令行选项缺省设置
- 缺省 SPARC® 体系结构中所做的更改
- OpenMP 库的增强
- 新的 f95 编译器命令行选项

有关在本节中引用的特定编译器选项的更多信息，另请参见《Fortran 用户指南》或 f95(1) 手册页。

表 1-8 Fortran 编译器的新功能和增强功能

特性	描述
Solaris OS x86 平台上 f95 的新编译功能	<p>使用值为 generic、native、386、486、pentium、pentium_pro、pentium3 或 pentium4 的 -xtarget 进行编译，以便在 Solaris x86 平台上生成可执行文件。x86 平台上的缺省值为 -xtarget=generic</p> <p>以下 f95 功能尚未在 x86 平台上实现，并且仅可在 SPARC® 平台上使用：</p> <ul style="list-style-type: none">• 区间运算（编译器选项 -xia 和 -xinterval）• 四精度（128 位）算法• IEEE 内模块 IEEE_EXCEPTIONS、IEEE_ARITHMETIC 和 IEEE_FEATURES• sun_io_handler 模块• 诸如 -autopar、-parallel、-explicitpar 和 openmp 等并行化选项。 <p>以下 f95 命令行选项仅可在 x86 平台上使用，而不能在 SPARC® 平台上使用：</p> <ul style="list-style-type: none">• -fprecision、-fstore、-nofstore <p>以下 f95 命令行选项仅可在 SPARC® 平台上使用，而不能在 x86 平台上使用：</p> <ul style="list-style-type: none">• -xcode、-xmemalign、-xprefetch、-xcheck、-xia、-xinterval、-xipo、-xjobs、-xlang、-xlinkopt、-xloopinfo、-xpagesize、-xprofile_ircache、-xreduction、-xvector、-depend、-openmp、-parallel、e-autopar、-explicitpar、-vpara、-XlistMP <p>此外，在 x86 平台上，-fast 选项可扩展为包括已添加的选项 -nofstore。</p>

表 1-8 Fortran 编译器的新功能和增强功能

特性	描述
Solaris OS x86 平台上 f95 的新编译功能 (续)	<p>Fortran 编译器支持 <code>-xarch</code>、<code>-xtarget</code> 和 <code>-xchip</code> 编译选项的新标志用于将在 x86 平台上运行的代码。这些新标志经过精心设计, 可充分利用 Pentium 3 和 Pentium 4 芯片以及 Solaris™ 软件对 x86 平台上 sse 和 sse2 指令的支持。新标志如下所示:</p> <ul style="list-style-type: none"> • <code>-xchip=pentium3</code> 针对 Pentium 3 类型处理器进行了优化 • <code>-xchip=pentium4</code> 针对 Pentium 4 类型处理器进行了优化 • <code>-xarch=sse</code> 将 sse 指令集添加到 <code>pentium_pro</code> 指令集体系结构中 • <code>-xarch=sse2</code> 将 sse2 指令集添加到 sse 允许的指令集中 • <code>-xtarget=pentium3</code> 设置了 <code>-xarch=sse</code>、<code>-xchip=pentium3</code> 和 <code>-xcache=16/32/4:256/32/4</code> • <code>-xtarget=pentium4</code> 设置了 <code>-xarch=sse2</code>、<code>-xchip=pentium4</code> 和 <code>-xcache=8/64/4:256/128/8</code> • <code>-fns</code> 仅可在 <code>pentium3</code> 或 <code>pentium4</code> 处理器上启用。当 <code>-xarch</code> 不是 <code>sse</code> 或 <code>sse2</code> 时, 将忽略 <code>-fns=yes</code>。否则, 对于 SSE 和 SSE2 浮点指令, <code>-fns=yes</code> 意味着下溢将清零 (FTZ), 并且非规格化操作数将按零处理 (DAZ)。<code>-fns=yes</code> 不会影响传统的 x86 浮点指令。例如, 对 <code>long double</code> 操作数或结果进行浮点运算可以利用传统的 x86 浮点指令, 并且这些运算不会受 <code>-fns=yes</code> 的影响。 <p><i>x86 的特殊注意事项:</i></p> <p>使用 <code>-xarch=sse</code> 或 <code>-xarch=sse2</code> 进行编译以便在 Solaris™ x86 SSE/SSE2 平台上运行的程序必须只能在支持 SSE/SSE2 的平台上运行。在不支持 SSE/SSE2 的平台上运行此类程序可能会导致在没有任何明确警告消息的情况下出现段故障或不正确的结果。以后可能会提供 OS 和编译器的补丁程序, 以避免在不支持 SSE/SSE2 的平台上执行 SSE/SSE2 编译的二进制。支持 SSE/SSE2 的 x86 平台包括在 Pentium 4 兼容处理器上运行的 Solaris 9 更新版 6。</p> <p>此外, 这一警告消息还扩展到了采用 <code>.i1</code> 内联汇编语言函数或 <code>__asm()</code> 汇编程序代码 (利用 SSE/SSE2 指令) 的程序。</p> <p>在尝试运行为目标运行时平台编译的二进制时, 请与您的系统管理员联系, 以确定这些平台是否支持 SSE/SSE2。</p> <p>改进的运行时性能</p> <p>此发行版将大大改进大多数应用程序的运行时性能。为了获得最佳的效果, 请使用较高的优化级别 <code>-xO4</code> 或 <code>-xO5</code> 进行编译。现在, 在这些级别上, 编译器可能内联了所包含的过程, 以及具有假定形状、可分配或指针参数的过程。</p>

表 1-8 Fortran 编译器的新功能和增强功能

特性	描述
新的 Fortran 2003 命令行内在函数	<p>Fortran 2003 标准草案引入了三个新的用于处理命令行参数和环境变量的内函数。此发行版的 f95 编译器已实现了这些内函数。这些新的内函数包括：</p> <ul style="list-style-type: none"> • GET_COMMAND(command, length, status) 以命令的形式返回调用该程序的整个命令行。 • GET_COMMAND_ARGUMENT(number, value, length, status) 以值的形式返回命令行参数。 • GET_ENVIRONMENT_VARIABLE(name, value, length, status, trim_name) 返回环境变量的值。
已更改的命令行选项缺省设置	<p>此发行版的 f95 更改了以下命令行选项缺省设置。</p> <ul style="list-style-type: none"> • -xprefetch 缺省值为 -xprefetch=no%auto,explicit。 • -xmemalign 的缺省值为 -xmemalign=8i, -xarch=v9 和 v9a 除外，其缺省值为 -xmemalign=8f。
缺省 SPARC® 体系结构中所做的更改	<p>缺省 SPARC® 体系结构不再是 V7。此 Sun Studio 9 发行版中限制了对 -xarch=v7 的支持。新的缺省值为 V8PLUS (UltraSPARC)。编译时使用 -xarch=v7 将被视为 -xarch=v8，因为 Solaris 8 OS 仅支持 -xarch=v8 或更高版本。</p>
OpenMP 库的增强	<p>OpenMP 库在以下几个方面得到了增强：</p> <ul style="list-style-type: none"> • OMP_NUM_THREADS 和多任务库的最大线程数已从 128 提高到 256。 • 此发行版 Fortran 95 编译器的共享内存并行编程的 OpenMP API 实现在并行区域中采用了自动变量作用域功能。有关详细信息，另请参见《OpenMP API 用户指南》。（对于此发行版，OpenMP 仅在 SPARC® 平台上实现。）

表 1-8 Fortran 编译器的新功能和增强功能

特性	描述
新的 f95 编译器命令行选项	<p>以下 f95 命令行选项在此发行版中为新选项。有关详细信息，另请参见 f95(1) 手册页。</p> <ul style="list-style-type: none"> • <code>-xipo_archive={ none readonly writeback }</code> 允许交叉文件优化，以包括归档 (.a) 库。（仅适用于 SPARC®） • <code>-xipo_archive=none</code> 未处理任何归档文件。 • <code>-xipo_archive=readonly</code> 编译器可以通过使用 <code>-xipo</code> 进行编译、在生成可执行文件之前驻留在归档库 (.a) 中的对象文件来优化传递给链接程序的对象文件。 • <code>-xipo_archive=writeback</code> 编译器可以通过使用 <code>-xipo</code> 进行编译、在生成可执行文件之前驻留在归档库 (.a) 中的对象文件来优化传递给链接程序的对象文件。库中包含的、在编译进程中进行了优化的任何对象文件都将被其已优化的版本替换。 如果未指定 <code>-xipo</code> 的设置，则编译器会将其设置为 <code>-xipo_archive=none</code>。 • <code>-xprefetch_auto_type=[no%]indirect_array_access</code> 生成间接预取，以间接访问数据数组。（仅适用于 SPARC®） • <code>[no%]indirect_array_access</code> 以与生成直接内存访问预取相同的方式来生成 [不生成] 由选项 <code>-xprefetch_level=[1 2 3]</code> 指示的间接循环预取。 如果未指定 <code>-xprefetch_auto_type</code> 的设置，则编译器会将其设置为 <code>-xprefetch_auto_type=[no%]indirect_array_access</code>。 <code>-xprefetch</code> 选项仅可在 SPARC® 平台上使用 诸如 <code>-xdepend</code>、<code>-xrestrict</code> 和 <code>-xalias_level</code> 等选项可影响计算间接预取候选项的主动性，进而影响自动间接预取插入的主动性，因为它们可以更好地消除内存别名信息的歧义。 • <code>-xprofile_pathmap=collect_prefix:use_prefix</code> 设置配置文件数据文件的路径映射。当文件配置的目录不是以前通过 <code>-xprofile=collect</code> 进行编译时所使用的目录时，请将 <code>-xprofile_pathmap</code> 选项与 <code>-xprofile=use</code> 选项一起使用。

命令行调试程序 dbx

以下新功能已添加到 Sun Studio 9 发行版的 dbx 中：

- 在 Linux 平台上支持 gcc 和 g++ 编译器
- 在 Solaris™ OS (x86 Platform Edition) 上支持 Fortran

区间运算

在此编译器集合发行版本中没有新的区间运算功能。

Sun 性能库

Sun Performance Library™ 是一组经过优化、用于解决线性代数和其他数值密集型问题的高速数学子例程。Sun 性能库是建立在可从 Netlib (<http://www.netlib.org>) 获得的公共领应用程序集合之上。将这些例程增强并捆绑后即成为 Sun 性能库。

表 1-9 列出了本发行版的 Sun 性能库新特性。有关更多信息，另请参见《Sun 性能库用户指南》和 3p 手册页。

表 1-9 Sun 性能库新特性

特性	描述
为 x86 发行的 Sun 性能库	<p>此发行版的 Sun 性能库包括 Solaris/x86 平台的库。存在两种版本：</p> <ul style="list-style-type: none">• 对支持 SSE2 指令集的系统使用这些指令的高性能版本。• 适用于不支持 SSE2 的系统的兼容版本。 <p>x86 版的 Sun 性能库与 SPARC® 版在功能上相同，但以下项除外：</p> <ul style="list-style-type: none">• 四精度例程 (dqdoti、dqdota) 不可用• 区间 BLAS 例程不可用• x86 库为单线程• 仅可使用 32 位寻址功能• 可移植库性能特性在 Solaris/x86 上不可用 <p>SSE2 支持需要以下版本的 Solaris/x86：</p> <ul style="list-style-type: none">• Solaris 10 build 48 (或更高版本)• Solaris 9 build 6 更新版 5 (或更高版本) <p>要与高性能 SSE2 优化的库进行链接，请使用 <code>-xarch=sse2</code> 标志。例如：</p> <pre>f95 -xarch=sse2 example.f -xlic_lib=sunperf 或 cc -xarch=sse2 example.c -xlic_lib=sunperf</pre>

dmake

dmake 是一个命令行工具，与 make(1) 兼容。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，则需要使用 \$(MAKE)。dmake 将分析 makefile 并决定能够并行生成哪些目标，以及在您设置的众多主机上分布这些目标的 build 版本。有关其他详细信息，另请参见 dmake(1) 手册页。表 1-10 列出了 Sun Studio 9 发行版中 dmake 的新特性。

表 1-10 dmake 新特性

特性	描述
dmake 中针对 Solaris 所做的性能、可靠性和可用性方面的改进	makefile 分析器比原来的版本快 10 倍，而比 GNU make 快 3 倍。Build 版本运行更快速，并且更稳定。此外，日志文件也更具有可读性。
Linux dmake 实现	在串行、并行和分布式模式下为 Linux build 实现完整的 dmake 功能。因此，无需对 makefile 做重大变更即可在 Linux 上生成 Solaris™ 应用程序。一个 build 版本可以同时分发给 Linux 和 Solaris™ 系统。

性能分析工具

表 1-11 列出了 Sun Studio 9 发行版性能分析工具中的新数据集合和演示特性。有关详细信息，另请参见以下手册页：

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

表 1-11 列出了 Sun Studio 9 性能分析器中的新特性和增强特性。

表 1-11 性能分析工具新特性

特性	描述
新的 Linux 分发	<p>除了 Sun Studio 9 for Solaris™ 之外,性能分析器现在还可用于 Sun Studio 9 for Linux 中。支持以下 Linux 操作系统:</p> <ul style="list-style-type: none">• Java™ Desktop System 1.0• SuSE Linux Enterprise Server 8• RedHat Enterprise Linux 3 <p>这两种操作系统上可用的公用程序相同,但有一点除外,即 <code>er_kernel</code> 不包括在 Linux 分发中。Linux 上的 <code>collect</code> 命令更有限。仅可使用基于时钟的分析和堆跟踪;有关详细信息,另请参见 <code>collect</code> 手册页。在 Linux 下可以对多线程应用程序进行分析,但在 RedHat 版的 Linux 操作系统下进行分析时,可以看到当前存在的高数据差异。</p>
数据空间分析	<p>现在,针对 SPARC® 平台的 C 程序可以进行数据空间分析。数据空间配置文件是一个数据集,诸如缓存缺失等与内存有关的事件将针对导致这些事件的数据对象引用在该集中进行报告,它不仅仅是出现与内存有关的事件的指令。</p> <p>数据空间分析信息的分析可显示在命令行上,也可显示在分析器 GUI 中,如下所示:</p> <ul style="list-style-type: none">• <code>er_print</code> 命令具有三个与数据空间分析相关的新选项: <code>data_objects</code>、<code>data_osingle</code> 和 <code>data_olayout</code>• 现在,该分析器包括两个与数据空间分析相关的新标签,即“数据对象”和“数据布局”。如果数据空间配置文件出现在实验中,则这些标签将自动显示。
后续进程	<p>后续进程的记录已得到增强,以包括记录所有后续进程的功能,而不仅仅是使用 <code>fork</code> 和 <code>exec</code> 命令创建的进程及其变异进程。为了支持增强的功能, <code>collect -F</code> 命令现在有了新的选项:</p> <pre>collect -F all.</pre> <p>使用 <code>-F all</code> (并非 <code>-F on</code>) 处理后续进程,例如系统呼叫均使用代码字母“c”命名。</p> <p>使用命令行公用程序 <code>er_print</code> 或在分析器 GUI 中可显式选择后续进程的数据以便进行显示。</p> <p>有关详细信息,另请参见 <code>collect(1)</code> 手册页。</p>
数据收集输出重定向	<p><code>collect</code> 命令具有一个新的选项,即 <code>collect -O file</code>,它可以将所有输出从 <code>collect</code> 重定向到已指定的 <code>file</code>。该命令不会重定向来自派生目标的输出。</p>
增强的分析器命令行参数	<p>现在,分析器命令(启动脚本)可接受对较长的参数使用双短横线,特别是 <code>--jdkhome</code> 和 <code>--fontsize</code>。</p>
分析器 API 共享库的新软件包	<p>分析器 API 的共享库已置于单独的软件包中,以便对其单独并免费进行分发。</p>

表 1-11 性能分析工具新特性 (续)

特性	描述
支持 <code>collect</code> 命令的说明文件	<code>collect</code> 命令具有新的命令行选项: <code>collect -C comment</code> 。将 <code>comment</code> 添加到该实验的说明文件中。可以应用多达 10 个 <code>-c</code> 参数。
实验预览和实验标题中的说明	实验预览和实验标题将显示该实验中任何说明文件的内容
增强的源码和反汇编显示	带注释的源码和反汇编改进了替代源上下文中的代码处理。以红色斜体字显示的索引行将指示从另一个文件插入代码的位置。借助 [源码] 标签, 在索引行上单击鼠标将打开替代源文件中的 [源码] 窗口。
增强的 <code>er_src</code> 命令	现在, 命令行公用程序 <code>er_src</code> 可以显示函数列表、处理 Java <code>.class</code> 文件并显示来自替代源上下文中的源码和反汇编。
Java™ 方法签名	Java™ 长名称格式可显示完整的方法签名, 而不仅仅是函数名称本身。
进行堆跟踪时 <code>mmap</code> 调用包括的内容	在进行堆跟踪时, 对 <code>mmap</code> 的调用将按照内存分配来处理。

集成开发环境 (IDE)

以下新功能已添加到 Sun Studio 9 发行版的 IDE 中:

- 新的 `ss_attach` 功能使您能够在其开始执行时即捕获程序, 并附加了 `dbx` 调试器以立即开始对其进行调试, 而不是在运行进程后再连接到调试器。
- 源码编辑器中的 [快速浏览] 组合框使您能够导航到类方法、函数、`#define` 或源文件的其他元素。

文档

本节介绍了 Sun Studio 9 文档的新特性。

- 《OpenMP API 用户指南》已进行扩展, 以包括两章新的内容。第 5 章介绍了通过 Fortran 95 `__AUTO` 子句进行自动数据作用域设置。第 6 章说明了 OpenMP 程序的性能, 并提供了一些用于改进性能的方法的一般建议。
- 《C 用户指南》包含两部分新的附录: 附录 A “按功能分组的编译器选项”和附录 D “C99 实现定义的信息”。附录 A 的内容原来在选项参考一章的开头, 但现在单独划分成一个附录, 以增强其明显程度。

- *性能分析器*手册新增了一章内容，标题为“了解带注释的源码和反汇编数据”。本章介绍了各种不同的注释，如索引行、编译器注释、特殊行（如外联函数），并说明了如何识别注释与最初源码之间的显示差异。
- Sun 开发人员门户站点上提供了有关性能分析器的教程，网址是：
<http://developers.sun.com/prodtech/cc>
- *Sun WorkShop to Sun Studio Migration* 帮助集具有有关文件比较和文件合并的新主题。
- 新的 *编译器与工具*帮助集介绍了 Sun Studio 发行版中包括的编译器和工具。每个主题均简单说明了某个组件，并列出了该组件的文档。