新增功能

Sun™ Studio 11

Sun Microsystems, Inc.
www.sun.com

文件号码 819-4753-10
2005 年 11 月，修订版 A

请将有关本文档的意见和建议提交至：http://www.sun.com/hwdocs/feedback

目录

阅读本书之前

《新增功能》中介绍了 Sun™ Studio 11 软件发行版本和 Sun Studio 10 软件发行版本的新增功能，其中包括 C、 C++ 和 Fortran 编译器、库以及工具中的新增功能。

印刷约定

表 P-1 字体约定

字体[1]	含义	示例
`AaBbCc123`	命令、文件和目录的名称；计算机屏幕输出。	编辑 `.login` 文件。 使用 `ls -a` 列出所有文件。 `% You have mail.`
`AaBbCc123`	用户键入的内容，与计算机屏幕输出的显示不同。	`% su` `Password:`
AaBbCc123	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	这些称为 *class* 选项。 要删除文件，请键入 **`rm`** *filename*。
新词术语强调	新词或术语以及要强调的词。	您**必须**成为超级用户才能执行此操作。
《书名》	书名	阅读 《用户指南》的第 6 章。

1 浏览器的设置可能会与这些设置不同。

表 P-2 代码约定

代码符号	含义	表示法	代码示例
[ ]	方括号中包含可选参数。	O[*n*]	-O4, -O
{ }	花括号中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，分号有时可用于分隔参数。	R*dir*[:*dir*]	-R/local/libs:/U/a
…	省略号表示一系列的省略。	-xinline=*f1*[,…*fn*]	-xinline=alpha,dos

Shell 提示符

Shell	提示符
C shell	*machine-name*%
C shell 超级用户	*machine-name*#
Bourne shell 和 Korn shell	$
Bourne shell 和 Korn shell 超级用户	#

支持的平台

此 Sun Studio 发行版本支持使用 SPARC® 和 x86 系列处理器体系结构（UltraSPARC®、SPARC64、 AMD64、 Pentium 和 Xeon EM64T）的系统。通过访问 http://www.sun.com/bigadmin/hcl 中的硬件兼容性列表，可以了解您在使用的 Solaris 操作系统版本的支持系统。这些文档列出了实现各个平台类型的所有差别。

在本文档中，这些与 x86 有关的术语具有以下含义：

- “x86”是指较大的 64 位和 32 位 x86 兼容产品系列。
- “x64”表示有关 AMD64 或 EM64T 系统的特定 64 位信息。
- “32 位 x86”表示有关基于 x86 系统的特定 32 位信息。

有关所支持的系统，请参见硬件兼容性列表。

访问 Sun Studio 软件和手册页

Sun Studio 软件及其手册页未安装到 /usr/bin/ 和 /usr/share/man 标准目录中。要访问该软件，必须正确设置 PATH 环境变量 （请参见 第 7 页的 “访问软件”）。要访问手册页，必须正确设置 MANPATH 环境变量 （请参见 第 8 页的 “访问手册页”）。

有关 PATH 变量的详细信息，请参见 csh(1)、 sh(1)、 ksh(1) 和 bash(1) 手册页。有关 MANPATH 变量的详细信息，请参见 man(1) 手册页。有关设置 PATH 变量和 MANPATH 变量以访问此发行版本的详细信息，请参见安装指南或询问系统管理员。

注 – 本节中的信息假设 Sun Studio 软件安装在 Solaris 平台上的 /opt 目录中和 Linux 平台上的 /opt/sun 目录中。如果未将软件安装在默认目录中，请询问系统管理员以获取系统中的相应路径。

访问软件

使用以下步骤决定是否需要更改 PATH 变量以访问该软件。

决定是否需要设置 PATH 环境变量

1. **通过在命令提示符后键入以下内容以显示 PATH 变量的当前值。**

```
% echo $PATH
```

2. **在 Solaris 平台上，查看输出中是否包含有 /opt/SUNWspro/bin 的路径字符串。在 Linux 平台上，查看输出中是否包含有 /opt/sun/sunstudio11/bin 的路径字符串。**

   如果找到该路径，则说明已设置了访问该软件的 PATH 变量。如果没有找到该路径，则需要按照下一步的说明设置 PATH 环境变量。

设置 PATH 环境变量以实现对编译器和工具的访问

- **在 Solaris 平台上，将以下路径添加到 PATH 环境变量中。如果以前安装了 Forte Developer 软件、 Sun ONE Studio 软件、或其他发行版本的 Sun Studio 软件，则将以下路径添加到这些安装路径之前。**

  /opt/SUNWspro/bin

- **在 Linux 平台上，将以下路径添加到 PATH 环境变量中。**

  /opt/sun/sunstudio11/bin

访问手册页

使用以下步骤决定是否需要更改 MANPATH 变量以访问手册页。

决定是否需要设置 MANPATH 环境变量

1. **通过在命令提示符后键入以下内容以请求 dbx 手册页。**

```
% man dbx
```

2. **请查看输出 （如果有）。**

   如果找不到 dbx(1) 手册页或者显示的手册页不是软件当前版本的手册页，请按照下一步的说明来设置 MANPATH 环境变量。

设置 MANPATH 环境变量以实现对手册页的访问

- **在 Solaris 平台上，将以下路径添加到 MANPATH 环境变量中。**

  /opt/SUNWspro/man

- **在 Linux 平台上，将以下路径添加到 MANPATH 环境变量中。**

  /opt/sun/sunstudio11/man

访问集成开发环境

Sun Studio 9 集成开发环境 (integrated development environment, IDE) 提供了创建、编辑、生成、调试 C、 C++ 或 Fortran 应用程序并分析其性能的模块。

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

IDE 是否可以正确操作取决于 IDE 能否找到核心平台。 sunstudio 命令会查找两个位置的核心平台：

- 该命令首先查找 Solaris 平台上的默认安装目录 /opt/netbeans/3.5V11 和 Linux 平台上的默认安装目录 /opt/sun/netbeans/3.5V11。
- 如果该命令在默认目录中找不到核心平台，则它会假设包含 IDE 的目录和包含核心平台的目录均安装在同一位置上。例如，在 Solaris 平台上，如果包含 IDE 的目录路径是 /foo/SUNWspro，则该命令会在 /foo/netbeans/3.5V11 中查找核心平台。在 Linux 平台上，如果包含 IDE 的目录的路径是 /foo/sunstudio11，则该命令会在 /foo/netbeans/3.5V11 中查找核心平台。

如果核心平台未安装在 sunstudio 命令查找它的任一位置上，则客户端系统上的每个用户必须将环境变量 SPRO_NETBEANS_HOME 设置为安装核心平台的位置 (/*installation_directory*/netbeans/3.5V11)。

在 Solaris 平台上，IDE 的每个用户还必须将 /*installation_directory*/SUNWspro/bin 添加到其他任何 Forte Developer 软件、 Sun ONE Studio 软件或 Sun Studio 软件发行版本路径前面的 $PATH 中。在 Linux 平台上， IDE 的每个用户还必须将 /*installation_directory*/sunstudio11/bin 添加到其他任何 Sun Studio 软件发行版本路径前面的 $PATH 中。

路径 /*installation_directory*/netbeans/3.5V11/bin 不能添加到用户的 $PATH 中。

访问 Sun Studio 文档

您可以访问以下位置的文档：

- 可以通过随软件一起安装在本地系统或网络上的文档索引获取该文档，位置为 Solaris 平台上的 file:/opt/SUNWspro/docs/zh/index.html 和 Linux 平台上的 file:/opt/sun/sunstudio11/docs/zh/index.html。

  如果未将软件安装在 Solaris 平台上的 /opt 目录中或 Linux 平台上的 /opt/sun 目录中，请询问系统管理员以获取系统中的相应路径。
- 大多数的手册都可以从 docs.sun.com℠ Web 站点获取。以下书目则只能通过您所安装的软件中获取：
  - 《标准 C++ 库类参考》
  - 《标准 C++ 库用户指南》
  - 《Tools.h++ 类库参考》
  - 《Tools.h++ 用户指南》
- 发行说明可从 docs.sun.com Web 站点获取。
- 在 IDE 中通过“帮助”菜单以及许多窗口和对话框上的“帮助”按钮，可以访问 IDE 所有组件的联机帮助。

您可以通过 Internet 访问 docs.sun.com Web 站点 (http://docs.sun.com) 以阅读、打印和购买 Sun Microsystems 的各种手册。如果找不到手册，请参见与软件一起安装在本地系统或网络中的文档索引。

注 – Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

采用易读格式的文档

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述，找到文档的易读版本。如果未将软件安装在 /opt 目录中，请询问系统管理员以获取系统中的相应路径。

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： • 《标准 C++ 库类参考》 • 《标准 C++ 库用户指南》 • 《Tools.h++ 类库参考》 • 《Tools.h++ 用户指南》	HTML，位于所安装软件中的文档索引 file:/opt/SUNWspro/docs/zh/index.html
自述文件	HTML，位于开发者门户网站 http://developers.sun.com/prodtech/cc/documentation/ss11/cn/mr/READMEs 中
手册页	HTML，位于安装的软件上的文档索引，位置为 Solaris 平台上的 file:/opt/SUNWspro/docs/zh/index.html 和 Linux 平台上的 file:/opt/sun/sunstudio11/docs/zh/index.html。
联机帮助	HTML，可通过 IDE 中的 “帮助”菜单和 “帮助”按钮访问
发行说明	HTML，位于 http://docs.sun.com

相关文档

下表描述的相关文档可以在 `file:/opt/SUNWspro/docs/zh/index.html` 和 `http://docs.sun.com` 上获取。如果未将软件安装在 `/opt` 目录中，请询问系统管理员以获取系统中的相应路径。

文档标题	描述
《使用 dbx 调试程序》	描述了如何使用 `dbx` 命令行调试器来调试用 C、C++、Fortran 和 Java™ 编程语言编写的程序。
《Fortran 编程指南》	描述了如何在 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》	详细说明了 Fortran 库和内部例程
《Fortran 用户指南》	描述了 `f95` 编译器的编译时环境和命令行选项。还包括关于将传统的 `f77` 程序迁移到 `f95` 的说明。
《C 用户指南》	描述了 `cc` 编译器的编译时环境和命令行选项。
《C++ 用户指南》	描述了 `CC` 编译器的编译时环境和命令行选项。
《性能分析器》	描述了如何使用收集器和性能分析器来执行大范围性能数据的统计分析，以及跟踪各种系统调用，并在函数、源代码行和指令级将这些数据与程序结构相关联。

访问相关的 Solaris 文档

下表描述了可从 `docs.sun.com` Web 站点上获取的相关文档。

文档集合	文档标题	描述
Solaris 参考手册集合	请参见手册页部分的标题。	提供有关 Solaris™ 操作环境的信息。
Solaris 软件开发者集合	《链接程序和库指南》	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	《多线程编程指南》	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。

开发者资源

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

- 有关编程技术和最佳实例的文章
- 有关编程小技巧的知识库
- 有关编译器和工具组件的文档以及与软件安装在一起的文档的更正信息
- 有关支持级别的信息
- 用户论坛
- 可下载的代码样例
- 新技术预览

您可以通过访问 http://developers.sun.com 找到其他开发者资源。

联系 Sun 技术支持

如果您遇到通过本文档无法解决的技术问题，请访问以下网址：

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

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。您可以通过以下网址提交您的意见和建议：

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

请在您的反馈信息中注明文档的文件号码 (819-4753-10)。

第1章

Sun Studio 11 新增功能与增强功能

Sun™ Studio 11 发行版本中包含了以下编译器、库以及工具的更新：

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

在大多数章节中都提供了一个表，其中列出了该组件的新增功能。该表分为两列，左列提供了功能的简短描述，右列则显示了更为详细的描述。

注 – 要查找本章介绍的 Sun Studio 11 文档，请参见随产品软件一起安装的文档索引，位置为 Solaris 平台上的 `/opt/SUNWspro/docs/zh/index.html` 和 Linux 平台上的 `/opt/sun/sunstudio11/docs/zh/index.html`。如果未将软件安装在 `/opt` 目录中，请询问系统管理员以获取系统或网络上的相应路径。

通用 C、 C++ 及 Fortran 功能

C、C++ 或 Fortran 编译器提供了以下新增功能。有关特定于每种语言的新功能的信息，请参见以下新增功能列表。

有关这些新增功能的详细信息，请参见每种语言的用户指南或手册页。

表 1-1 C 编译器新增功能

功能	描述
适用于 x86 开发的 `-xarch` 新标志	目前， `-xarch` 选项支持将以下新标志用于 x86 平台开发： • `amd64a` • `pentium_proa` • `ssea` • `sse2a`。
支持 x86 `-xpagesize` 选项	目前，为 x86 平台和 SPARC 平台启用了 `-xpagesize`、`-xpagesize_heap` 以及 `-xpagesize_stack` 选项。
支持 x86 内存模型	使用 `-xmodel` 新选项，可以在 64 位 AMD 体系结构中指定内核、小型或中型内存模型。
支持 SSE/SSE2 的整型介质内部函数	本发行版本支持 SSE2 128 位 XMM 寄存器整型介质指令的内部函数。在源代码中包含 `sunmedia_intrin.h` 头文件，并指定 `-xbuiltin` 选项以使用这些函数。并且，这些内部函数需要 SSE2 的支持，因此，请指定诸如 `-xarch=sse2`、 `-xarch=amd64` 或 `-xtarget=opteron` 等选项。
用于 x86 SSE2 平台的 `-xvector` 新标志	使用 `-xvector` 选项，可自动生成对向量库函数的调用和/或生成 SIMD （Single Instruction Multiple Data，单指令多数据）指令。
用于 SPARC 平台的二进制优化器	使用 `-xbinopt` 新选项，编译器可以准备由 `binopt` (1) 二进制优化器进一步优化的二进制文件。
SPARC `-xtarget` 和 `-xchip` 的新值	`-xtarget` 的新标志 `ultra3iplus`、`ultra4plus` 和 `ultraT1` 连同 `-xchip` 的新标志 `ultra3iplus`、`ultra4plus` 以及 `ultraT1` 为 UltraSPARC IIIiplus、 UltraSPARC T1 以及 UltraSPARC IVplus 处理器提供了代码生成。
`STACKSIZE` 环境变量的增强功能	已增强了 `STACKSIZE` 环境变量的语法，可接受表示从属线程堆栈大小的单位关键字：B 表示字节； K 表示千字节； M 表示兆字节； G 表示千兆字节。 例如， `setenv STACKSIZE 8192` 表示将从属线程堆栈大小设置为 8 MB ； 1235B 表示将从属线程堆栈大小设置为 1235 字节；1235G 表示将其设置为 1235 千兆字节。默认情况下，没有后缀字母的整数值仍然表示千字节。
OpenMP 自动确定作用域	目前，C 和 C++ 程序提供了自动确定作用域功能。在《Sun Studio OpenMP API User's Guide》的第 3 章介绍了该功能。

C 编译器

除了之前在第 14 页的“通用 C、C++ 及 Fortran 功能”下面列出的新增功能外，C 编译器还提供了以下新增功能。

表 1-2 C 编译器新增功能

功能	描述
新的调试器信息的默认格式	默认情况下，C 编译器现在会使用 DWARF 格式生成调试器信息。这种更改应该是透明的，因为 `dbx` 和性能分析软件易于接受并倾向于使用 DWARF 格式。通过指定 `-xdebugformat=stabs`，仍然可以使用 stabs 格式生成调试器信息。
两个新的 pragma	• c99 使用 `c99` (`implicit` | `no%implicit`) pragma 可查找隐式函数声明。 • `[no_]warn_missing_parameter_info` 使用 `[no_]warn_missing_parameter_info` pragma 可查找不包含任何参数类型信息的函数声明。

C++ 编译器

除了之前在第 14 页的 “通用 C、 C++ 及 Fortran 功能”下面列出的新增功能外， C++ 编译器还提供了以下新增功能。

表 1-3 C++ 编译器新增功能

功能	描述
新的调试器信息格式	目前， C++ 编译器可以使用 DWARF 格式生成调试器信息。默认格式仍然为 stabs 格式，但可通过将新选项 `-xdebugformat` 设置为 `-xdebugformat=dwarf` 来生成 DWARF 数据。
调用函数模板中的相关静态函数	C++ 标准规定，取决于模板参数的函数调用只能引用具有外部链接的可见函数声明。如果应用程序代码取决于忽略此规则并从函数模板中调用相关静态函数的编译器，则需要指定 `-features=[no%]tmplrefstatic`。

Fortran 编译器

在此发行版本中， Fortran 95 编译器中的新增功能仅限于之前在第 14 页的 “通用 C、C++ 及 Fortran 功能”下面列出的新增功能。

命令行调试器 dbx

表 1-4 dbx 新增功能

功能	描述
支持 Linux 操作系统上的 AMD64 体系结构	64 位 dbx 目前支持 AMD64 体系结构。在 Linux 操作系统上，64 位 dbx 无法调试 32 位程序。要在 Linux 操作系统上调试 32 位程序，请使用 `-x exec32` 选项运行 dbx 命令来启动 32 位 dbx。
支持生成 DWARF 符号调试信息的 Sun Studio C 编译器	默认情况下，使用 C 编译器的 DWARF 符号调试信息。
`thr_create` 和 `thr_exit` 事件	当创建某个线程或创建指定了 *thread_id* 的线程时，便会发生 `thr_create` 事件。当线程退出时，便会发生 `thr_exit` 事件。
`step_abflow` 环境变量	如果将此环境变量设置为 stop 时，在单步执行时， dbx 会在 `longjmp()`、`siglongjmp()` 和 throw 语句处停止。如果设置为 ignore 时，dbx 不会检测 `longjmp()` 和 `siglongjmp()` 的异常控制流变化。
`intercept` 和 `unintercept` 命令的修订语法	该语法现在包括用于清除截取列表和排除列表的 -set 选项，并且会将这些列表设置为仅截取或排除特定类型的抛出。
支持调试使用 Java™ 2 Platform, Standard Edition v 5.0 Update 3 中的 `javac` 编译器编译的 Java 程序	支持调试使用 J2SE 5.0 Update 3 编译的 Java 程序，但具有以下限制： - 部分支持泛型 - 不支持自动装箱和拆箱 - 不支持静态导入

OpenMP API

表 1-5 OpenMP API 新增功能

功能	描述
OpenMP 2.5	OpenMP 实现已升级到 2.5 规范。有关详细信息，请参见 OpenMP 网站 `http://www.openmp.org/`。
C++ 自动确定作用域	目前，为 C++ 程序以及 C 和 Fortran 95 启用了自动确定变量作用域功能。《OpenMP User's Guide》的第 3 章介绍了自动确定作用域。

区间运算

本发行版本中没有区间运算的新增功能。

Sun 性能库

表 1-6 Sun 性能库新增功能

功能	描述
x86 平台上的 Sun 性能库	本发行版本中的 Sun 性能库包含基于 64 位 x86 系统的 Solaris 操作系统库。 64 位 x86 版的 Sun 性能库在功能上与 SPARC v9 版相同，但以下内容除外： • 四精度例程 （dqdoti、 dqdota）不可用。 • 区间 BLAS 例程不可用。 • 带有 64 位整型参数的例程不可用，即，DAXPY() 可用，但 DAXPY_64() 不可用。 • 可移植性能库功能在基于 x86 的系统上的 Solaris 操作系统上不可用。 在 amd64 库中，已将许多经常调用的 BLAS 内核进行了优化。还进一步优化了某些内部 FFT 例程。与 SPARC v9 版本类似，对许多例程进行了并行优化。
SPARC 平台上的 Sun 性能库	为最新的 UltraSPARC 处理器进行了 BLAS 和 FFT 改进。

dmake

表 1-7 dmake 新增功能

功能	描述
兼容模式	为了与 GNU make 兼容而添加了 `-x SUN_MAKE_COMPAT_MODE` 命令行选项和 `SUN_MAKE_COMPAT_MODE` 环境变量。

性能分析工具

表 1-8 性能分析工具新增功能

功能	描述
改进了对所显示标签的控制	分析器的标签机制经过重新设计后具有更大的灵活性。只有适用于至少一种装入实验的标签才可用，并显示一组默认的标签而不是所有的标签，尤其对于较大的实验。可以在 .er.rc 文件中使用标签指令设置默认的标签。可以使用 “设置数据表示”对话框中的 “标签”标签来添加或删除所显示的标签。
“过滤器”对话框中的“高级”标签	目前，“过滤器”对话框中包含一个“高级”标签，可以在其中键入过滤表达式。也可以使用 AND 和 OR 运算符以及反映 “函数”标签、“DataObjects”标签、“DataLayout”标签或“MemoryObject”标签中的一项或多项选择的词组来建立表达式。
“时间线”标签目前会考虑过滤情况	“时间线”标签仅显示通过当前过滤器设置的事件。
改进了后续过程的处理方式	分析器和 `er_print` 实用程序将处理 `.er.rc` 文件中的 en_desc on|off 指令。如果该指令指定了 `on`，则会立即读取所有后续的实验；如果该指令指定了 `off`，则仅读取创建的实验。
改进了分析器中新窗口的行为	现在，通过单击 “新建窗口”工具栏按钮或选择 “文件” → “创建新窗口”打开的其他分析器窗口彼此之间更加清晰地被隔开。它们将共享装入的实验，但是您可以在每个窗口中单独设置过滤、度量和排序等。
新的 “内存对象”标签和报告	可以使用分析器中的新标签来显示缓存线、页面等的性能数据。内存对象可以使用若干个新的 `er_print` 命令。可以在 “设置数据表示”对话框的 “标签”标签中单击 “添加自定义对象”来创建自定义内存对象。
Linux 上的硬件计数器分析	支持的 Linux 系统提供硬件计数器溢出分析。此支持要求您安装 Perfctr 修补程序。
改进了 MPI 分析的处理方式	可以识别指定 MPI 的 LAM 和 MPICH 版本的进程级别的其他变量。

表 1-8 性能分析工具新增功能 *（续）*

功能	描述
Java 模式已被 View 模式所替换	Java 模式已被 View 模式所替换 View 模式的用户、专家和计算机设置对应于 Java 模式的打开、专家和关闭设置。View 模式适用于 Java 程序（尤其是 OpenMP）以外的编程模型。会显示一条警告来接受 `javamode` 命令。
`er_print` 命令更改	已更改了 `er_print` 实用程序的各种命令。已重命名了影响数据对象的命令，并且可以更加一致地处理有关度量的命令。新的 `procstat` 命令用于打印有关数据处理的信息。使用 `filters` 新命令可以指定过滤表达式。已添加了定义过滤器并计算内存对象索引的新表达式语法。
“函数”标签、“DataObjects”标签、“DataLayout”标签中的多项选择标签	在“函数”标签、“DataObjects”标签、“DataLayout”标签和“MemoryObjects”标签中，目前可以进行多项选择。
改进了过滤	除了选择实验和过滤要显示其度量的样本、线程、LWP 和 CPU 外，现在还可以指定一个过滤表达式，对于要包含在显示中的任何数据记录表示为 true 值。

集成开发环境 (Integrated Development Environment, IDE)

表 1-9 IDE 新增功能

功能	描述
支持 J2SE 5.0 Update 3	IDE 目前可以运行 J2SE 5.0 Update 3

文档

有关 Sun Studio 11 文档的更新信息，请参见开发者门户网站中的最新信息页 (`http://developers.sun.com/prodtech/cc/support_index.html`)。

第2章

Sun Studio 10 新增功能与增强功能

Sun™ Studio 10 将替代 Sun™ Studio 9。Sun Studio 10 发行版本中的新增功能包括对以下编译器、库以及工具的更新：

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

在大多数的章节中都提供了一个表，其中列出了该组件的新增功能。该表分为两列，左列提供了功能的简短描述，右列则显示了更为详细的内容。

注 – 要查找本章中介绍的 Sun Studio 10 文档，请参见随产品软件一起安装的文档索引，位于 `/opt/SUNWspro/docs/zh/index.html` 中。如果未将软件安装在 `/opt` 目录中，请询问系统管理员以获取系统或网络上的相应路径。

C 编译器

表 2-1 C 编译器的新增功能

功能	描述
OpenMP 并行编程 API	目前在运行 Solaris 操作系统的基于 32 位和 64 位 x86 系统上启用了此 API。
新的 `-xarch` 选项	`-xarch=amd64` 用于指定 64 位 AMD 指令集的编译。如果指定了 `-xarch=amd64`，则 C 编译器目前可以预定义 `__amd64` 和 `__x86_64`。
新的 `-xtarget` 选项	`-xtarget=opteron` 用于指定 32 位 AMD 编译中的 `-xarch`、`-xchip` 以及 `-xcache` 设置。
基于 x86 系统的新 `-xregs` 标志	利用 `-xregs` 选项 `-xregs=[no%]frameptr` 的仅限于 x86 的新标志，您可以将框架指针寄存器用作未分配的被调用方保存寄存器以提高应用程序的运行时性能。
`lint` 的新 `-Xarch=amd64` 选项	C 实用程序 `lint` 目前接受新的选项 `-Xarch=amd64`。有关详细信息，请参见 `lint`(1) 手册页。
基于 x86 系统的 `-xarch=generic64`	除支持传统的 SPARC 平台外，现有的 `-xarch=generic64` 选项目前还支持 x86 平台。
基于 x86 系统的 `-xipo`	`-xipo` 选项目前可以在基于 x86 的系统上使用。

注 – 要生成 64 位代码，必须在命令行中的 `-fast` 和 `-xtarget` 的右侧指定 `-xarch=amd64`。例如，指定 `cc -fast -xarch=amd64` 或 `cc -xtarget=opteron -xarch=amd64`。新的 `-xtarget=opteron` 选项不会自动生成 64 位代码。该选项将扩展为 `-xarch=sse2`、`-xchip=opteron` 以及 `-xcache=64/64/2:1024/64/16`，它产生 32 位代码。`-fast` 选项也会产生 32 位代码，因为该选项是定义 `-xtarget=native` 的宏。

C++ 编译器

表 2-2 C++ 编译器的新增功能

功能	描述
OpenMP 并行编程 API	目前在运行 Solaris 操作系统的基于 32 位和 64 位 x86 系统上启用了此 API。
新的 `-xarch` 选项	`-xarch=amd64` 用于指定 64 位 AMD 指令集的编译。如果指定了 `-xarch=amd64`，C++ 编译器目前会预定义 `__amd64` 和 `__x86_64`。
新的 `-xtarget` 选项	`-xtarget=opteron` 指定 32 位 AMD 编译的 `-xarch`、`-xchip` 以及 `-xcache` 设置。
基于 x86 系统的新 `-xregs` 标志	利用 `-xregs` 选项 `-xregs=[no%]frameptr` 的仅限于 x86 的新标志，您可以将框架指针寄存器用作未分配的被调用方保存寄存器以提高应用程序的运行时性能。
基于 x86 系统的 `-xarch=generic64`	除支持传统的 SPARC 平台外，现有的 `-xarch=generic64` 选项目前还支持 x86 平台。
基于 x86 系统的 `-xipo`	`-xipo` 选项目前可以在基于 x86 的系统上使用。
模板－模板参数	可以指定带参数的模板定义，这些参数就是模板本身，而不是类型或值。请回想一下，在类型上实例化的模板本身就是一个类型。例如，请参见第 23 页的 “模板－模板参数示例”。
嵌套类的访问规则	默认模式下，本发行版本的 C++ 编译器允许嵌套类具有成员函数访问成员类的相同权限。有关详细信息，请参见第 24 页的 “嵌套类的访问规则”。

注 – 要生成 64 位代码，必须在命令行中的 `-fast` 和 `-xtarget` 的右侧指定 `-xarch=amd64`。例如，指定 `CC -fast -xarch=amd64` 或 `CC-xtarget=opteron -xarch=amd64`。新的 `-xtarget=opteron` 选项不会自动生成 64 位代码。该选项将扩展为 `-xarch=sse2`、`-xchip=opteron` 以及 `-xcache=64/64/2:1024/64/16`，它产生 32 位代码。`-fast` 选项也会产生 32 位代码，因为该选项是定义 `-xtarget=native` 的宏。

模板－模板参数示例

本节提供了两个代码示例，一个示例不使用模板－模板参数，另一个示例则使用模板－模板参数。

本示例不使用模板－模板参数，因为 MyClass<int> 是一个类型。

```
template<typename T> class MyClass { ... };
std::list< MyClass<int> > x;
```

本示例中，类模板 C 具有一个是类模板的参数，对象 x 是一个将类模板 A 作为其参数的 C 的实例。 c 的成员 y 具有 A<int> 类型。

```
// ordinary class template
template<typename T> class A {
    T x;
};
// class template having a template parameter
template < template<typename U> class V > class C {
    V<int> y;
// instantiate C on template
C<A> x;
```

嵌套类的访问规则

在默认标准模式下， C++ 编译器目前允许嵌套类访问封装类的专有成员。

C++ 标准规定嵌套类不具有对封装类成员的特殊访问权限。但是，大多数用户认为这种限制不合理，因为成员函数拥有专有成员的访问权限，因此，成员类也应该有此权限。在下面的示例中，函数 foo 试图访问类 outer 的专有成员。根据 C++ 标准，该函数不具有访问权限，除非它被声明为友元函数：

```
class outer {
    int i; // private in outer
    class inner {
        int foo(outer* p) {
            return p->i; // invalid
        }
    };
};
```

C++ 委员会目前正准备采纳对访问规则的更改，以便向成员类授予与成员函数相同的访问权限。由于预料到将会更改语言规则，很多编译器已经实现了这种规则。

要恢复旧的编译器行为，禁用这种访问权限，请使用编译器选项 -features=no%nestedaccess。默认设置为 -features=nestedaccess。

Fortran 编译器

表 2-3 Fortran 编译器的新增功能

功能	描述
OpenMP 并行编程 API	目前在运行 Solaris 操作系统的基于 32 位和 64 位 x86 系统上启用了此 API。
新的 `-xarch` 选项	`-xarch=amd64` 用于指定 64 位 AMD 指令集的编译。如果指定了 `-xarch=amd64`，则 Fortran 编译器目前会预定义 `__amd64` 和 `__x86_64`。
新的 `-xtarget` 选项	`-xtarget=opteron` 指定 32 位 AMD 编译的 `-xarch`、`-xchip` 以及 `-xcache` 设置。
基于 x86 系统的 `-xarch=generic64`	除支持传统的 SPARC 平台外，现有的 `-xarch=generic64` 选项目前还支持 x86 平台。
基于 x86 系统的 `-xipo`	`-xipo` 选项目前可以在基于 x86 的系统上使用。
Big-endian 和 little-endian 平台间的二进制（未格式化）文件共享	在基于 SPARCA 和 x86 的系统间移动时，新的编译器标志 `-xfilebyteorder` 支持二进制 I/O 文件。标志识别未格式化 I/O 文件的字节顺序和字节对齐。有关详细信息，请参见第 25 页的 “Big-endian 和 Little-endian 平台间的二进制文件共享”。

注 – 要生成 64 位代码，必须在命令行中的 `-fast` 和 `-xtarget` 的右侧指定 `-xarch=amd64`。例如，指定 `f95 -fast -xarch=amd64` 或 `f95 -xtarget=opteron -xarch=amd64`。新的 `-xtarget=opteron` 选项不会自动生成 64 位代码。该选项将扩展为 `-xarch=sse2`、`-xchip=opteron` 以及 `-xcache=64/64/2:1024/64/16`，它产生 32 位代码。`-fast` 选项也会产生 32 位代码，因为该选项是定义 `-xtarget=native` 的宏。

Big-endian 和 Little-endian 平台间的二进制文件共享

在基于 SPARCA 和 x86 的系统间移动时，新的编译器标志 `-xfilebyteorder` 支持二进制 I/O 文件。标志识别未格式化 I/O 文件的字节顺序和字节对齐。

标志的语法是：

-xfilebyteorder=
{[little*max_align*:%all,*unitno*,*filename*},[big*max_align*:{%all,*unitno*,*filename*}],[native:{%all,*unitno*,*filename*}]}:

max_align	目标平台的最大字节对齐。值为 1、2、4、8 以及 16。对齐适用于 Fortran VAX 结构和 Fortran 95 派生类型，这些派生类型使用依赖于平台的对齐来获得与 C 结构的兼容性。
little*max_align*:{%all,*unitno*,*filename*}	文件或单元编号的列表，文件或单元编号是最大字节对齐为 *max_align* 的系统上使用的 "little-endian" 文件。例如，little4 描述了 32 位 x86 文件，而 little16 描述了 64 位 x86 文件。
big*max_align*:{%all,*unitno*,*filename*}	文件或单元编号的列表，文件或单元编号是最大字节对齐为 *max_align* 的系统上使用的 "big-endian" 文件。
native:{%all,*unitno*,*filename*}	文件或单元编号的列表，文件或单元编号是编译处理器系统上使用的相同字节顺序和对齐的本机文件。
%all	指定所有的文件和逻辑单元，以 "SCRATCH" 打开的或在该选项中显式命名的内容除外。它可以用来描述该标志非显式列出的默认文件。%all 只能出现一次。
unitno	该应用程序打开的 Fortran 逻辑单元编号。
filename	该应用程序打开的 Fortran 文件名称。

该选项不适用于以 STATUS=scratch 打开的文件。这些文件的 I/O 操作始终带有本机处理器的字节顺序和字节对齐。

编译器命令行上未指定 -xfilebyteorder 时，第一个默认值是 -xfilebyteorder=native:%all。必须为该选项至少指定一个参数。即，至少出现 little:、big: 或 native: 参数之一。

该标志未显式声明的文件假定为本机文件。例如，使用 -xfilebyteorder=little4:zfile.out 编辑时声明的 zfile.out 是 little-endian 32 位 x86 文件（符合 4 字节最大数据对齐规则），其他的文件是本机文件。

如果指定给文件的字节顺序与本机处理器的字节顺序相同，而指定的对齐却与本机处理器不同，则即使没有字节交换，也会使用适当的填充。例如，这可能会出现在下面的情况中：使用 -xarch=amd64 对 64 位 x86 进行编译，且指定 -xfilebyteorder=little4:*filename*。

Big-endian 和 little-endian 平台间共享的数据记录中的声明类型必须具有相同大小。例如，使用 `-xtypemap=integer:64、real:64、double:64` 编译的 x86 可执行文件不能读取由 SPARC 可执行文件（使用 `-xyptemap=integer:64、real:64、double:128` 进行编译）生成的文件，因为默认双精度数据类型的大小不同。

共享的 I/O 文件不能包含 VAX `UNION/MAP` 数据结构，因为编译器无法知道如何解释 `UNION` 数据。使用 `-xfilebyteorder` 标志声明包含 `UNION` 数据的文件时，将产生运行时错误。

命令行调试器 dbx

表 2-4 dbx 的新增功能

功能	描述
AMD64 体系结构支持	64 位 dbx 目前支持 AMD64 体系结构。

与基于 SPARC 系统的 Sun Studio 软件一样，基于 x86 系统的 Sun Studio 软件包含两个 dbx 二进制文件：一个是 32 位 dbx，只能调试 32 位程序；一个是 64 位 dbx，可以调试 32 位和 64 位程序。

启动 dbx 后，它会决定执行哪一个二进制文件。在 64 位 Solaris 操作系统上，默认值是 64 位 dbx。

OpenMP API

表 2-5 OpenMP API 的新增功能

功能	描述
运行 Solaris 10 操作系统的基于 x86 系统的可用性。	基于 SPARC 系统的 Solaris 操作系统上已经可用的 OpenMP API 功能同样可以用在运行 Solaris 10 操作系统基于 32 位或 64 位 x86 系统上的 Sun Studio 编译器中。
`libmtsk`	多任务库 `libmtsk` 现在是一个共享库，并且是 Solaris 10 操作系统的一部分。
嵌套并行操作	本发行版本支持嵌套并行操作。默认情况下，该操作处于禁用状态，需要设置 `OMP_NESTED` 环境变量，对 `omp_set_nested()` 函数进行运行时调用来启用此操作。启用嵌套并行操作后，不能忽略在并行区域中对大多数 `omp_` 函数的调用。调整并行环境（例如，`omp_set_num_threads()` 或 `omp_set_dynamic()`）的调用仅影响线程遇到的相同或内部嵌套级别的后续并行区域。
线程的默认行为	现在线程的默认行为是 SLEEP。早期的默认行为是 SPIN。要恢复早期的默认行为，请使用 `SUNW_MP_THR_IDLE=SPIN`。
`SUNW_MP_NUM_POOL_THREADS` 环境变量	`SUNW_MP_NUM_POOL_THREADS` 指定线程池的大小（最大线程数）。线程池仅包含非用户的线程 — `libmtsk` 库创建的线程。不包含用户线程，例如主线程。将 `SUNW_MP_NUM_POOL_THREADS` 设置成 0 会强制清空线程池，并由一个线程来执行所有的并行区域。指定的值必须为非负整数。默认值是 1023。该环境变量可以防止单一进程创建过多的线程（这种情况可能会发生，例如，带有递归嵌套并行区域的线程）。
`SUNW_MP_MAX_NESTED_LEVELS` 环境变量	`SUNW_MP_MAX_NESTED_LEVELS` 指定活动并行区域的最大深度。具有超过 `SUNW_MP_MAX_NESTED_LEVELS` 活动嵌套深度的任何并行区域都将由单一线程执行。其值必须是正整数。默认值是 4。最外层并行区域的深度级别是 1。
`SUNW_MP_GUIDED_WEIGHT` 环境变量	`SUNW_MP_GUIDED_WEIGHT` 设置 `libmtsk` 所使用的加权值来循环 `GUIDED` 调度。`libmtsk` 使用以下公式来计算 `GUIDED` 循环的块大小：*chunk_size*= *num_unassigned_iterations*/(*weight***num_threads*) 其中，*num_unassigned_iterations* 是循环中未分配给任何线程的迭代数，*weight* 是浮点常量（本发行版本的默认值是 2.0，早期版本的默认值是 1.0），*num_threads* 是用于执行循环的线程数。为 `SUNW_MP_GUIDED_WEIGHT` 指定的值必须是正的、非零浮点常量。`libmtsk` 将使用该值作为 `GUIDED` 块大小计算中的加权值。

区间运算

本发行版本中没有新增的区间运算功能。

Sun 性能库

表 2-6 Sun 性能库的新增功能

功能	描述
64 位 Solaris 操作系统支持	本发行版本中的 Sun 性能库包含对基于 x86 系统的 64 位 Solaris 操作系统的支持。

64 位 x86 版的 Sun 性能库与 SPARC v9 版在功能上相同，但存在以下不同情况：

- 四精度例程 （`dqdoti`、 `dqdota`）不可用。
- 区间 BLAS 例程不可用。
- 带有 64 位整型参数的例程不可用。例如，`DAXPY()` 可用，而 `DAXPY_64()` 不可用。

要与高性能 amd64 优化的库进行链接，请使用 `-xarch=amd64` 标志。例如：

```
f95 -xarch=amd64 example.f -xlic_lib=sunperf
```

dmake

表 2-7 dmake 的新增功能

功能	描述
新的 DMAKE_OUTPUT_MODE 环境变量	利用新的环境变量或 makefile 宏， DMAKE_OUTPUT_MODE 可以更改日志文件的格式。默认情况下，或者将 DMAKE_OUTPUT_MODE 设置为 TXT1 时，dmake 将向日志文件打印附加的系统信息，并重复执行带有输出的命令。将 DMAKE_OUTPUT_MODE 设置为 TXT2 时，将省略系统信息，且从不重复执行命令。有关详细信息，请参见 dmake(1) 手册页的 ENVIRONMENT/MACROS 一节。（请注意，手册页中说明的环境变量不正确； DMAKE_OUTPUT_MODE 正确的值应当是 TXT1 和 TXT2。）
Unix2003 遵循性	可以通过设置 DMAKE_COMPAT_MODE=POSIX 来强制遵循 Unix2003。
网格引擎支持	通过设置 DMAKE_MODE=grid 来指定网格引擎支持。
系统重载控制	使用 DMAKE_ADJUST_MAX_JOBS 控制系统重载。
提高内存的使用情况	本发行版本中包含了对内存使用的改善。

性能分析工具

表 2-8 性能分析工具的新功能

功能	描述
对实验格式的更改	已对实验格式进行更改。目前，日志中包含了一个提供目标大小 （以位计算）的条目。此外，版本由 9.1 更改为 9.2，因此旧工具将无法读取新的实验，但使用 Sun Studio 10 工具可以读取旧的实验。
er_kernel 实用程序	新的 er_kernel 实用程序现在只适用于 Solaris 10 操作系统。 DTrace 权限要求使用该 er_kernel 实用程序。
增强的性能度量精度	性能分析器和 er_print 实用程序的度量精确度已由小数点后一位增加到两位。
直接编辑实验说明文件	性能分析器中已添加了对实验说明文件的直接编辑功能。
显示函数名称的新选项	目前，在性能分析器和 er_print 命令中可以使用新的选项来显示函数名称。
增强的度量选项	性能分析器中的度量选项已得到增强。您可以一次性选择或清除所有度量的显示。

表 2-8 性能分析工具的新功能 *（续）*

功能	描述
收集器 GUI 的变更	用于下面后续进程的菜单已移动到“收集实验”标签中。除了 on 和 off 选项，目前菜单还支持所有的选项及扩展硬件计数器的溢出分析功能。
硬件计数器溢出分析的增强功能	硬件计数器溢出分析这项功能已得到增强，它可以与大多数的处理器一起使用，包括基于 x86 的处理器。在 dbx 和性能分析器 GUI 中使用 `collect -h` 命令、`collector hwprofile` 命令时，可以使用此项增强功能。
新的 `appendfile` 选项	已将 `appendfile` 选项添加到 `er_print` 实用程序中。利用该选项，可将 `er_print` 实用程序的输出添加到现有文件的末尾。
`er_src` 实用程序默认行为的变更	已将 `er_src` 实用程序的默认行为更改为以下命令的相同行为：`er_src -source all -1 object`。
J2SE 技术的位置	目前，性能分析器和 `collect` 实用程序使用 J2SE 技术的默认位置（产品安装程序安装它的位置）。
新的 `collect -J java_args` 选项	`collect -J java_args` 选项提供了一种方法，可以向正用于分析的 Java 安装传递标志参数。
暂停和恢复期间的抽样行为变更	抽样数据产生在暂停之前、恢复之后，但不产生于收集器暂停的时候。
JVM 函数的伪函数	Java 模式下 Java 虚拟机 （JVM）[1] 函数的伪函数名称由 `<JVM-Overhead>` 更改为 `<JVM-System>`。
`<Unknown>` 子类型	Java 函数 `<Unknown>` 子类型的名称已发生更改，更易于理解。
`.er.rc` 文件路径	处理后的 `.er.rc` 文件的路径目前分别显示在性能分析器的“错误/警告日志”窗口以及 `er_print` 和 `er_src` 实用程序的 `stderr` 中。
`JDK_1_4_2_HOME` 环境变量	用于定义 Java 路径以进行数据收集的 `JDK_1_4_2_HOME` 环境变量，现已废弃。
堆分析	由于无法在 JVM 1.5 中得到支持，现已废弃对 Java 程序的堆分析。
`collect -j` 扩展选项	`collect` 实用程序将接受 on 或 off 的值，以及指向 Java 安装的路径以用于分析。

1 术语“Java 虚拟机”和“JVM”指 Java™ 平台的虚拟机。

集成开发环境 (Integrated Development Environment, IDE)

表 2-9 IDE 的新增功能

功能	描述
脚本执行功能	目前，您可以直接从 IDE 执行脚本。
Linux 操作系统上的 ss_attach	ss_attach 功能目前可用在运行于 Linux 操作系统上的 Sun Studio 软件中

文档

有关最新的 Sun Studio 10 文档信息，请参见开发者门户网站中的最新消息页，网址为 `http://developers.sun.com/prodtech/cc/support_index.html`。