

Oracle® Developer Studio 12.5 发行版的新增 功能



文件号码 E71944
2016 年 7 月

文件号码 E71944

版权所有 © 2014, 2016, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，则适用以下注意事项：

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。除非您与 Oracle 签订的相应协议另行规定，否则对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的保证，亦不对其承担任何责任。除非您和 Oracle 签订的相应协议另行规定，否则对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

获得 Oracle 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

目录

使用本文档	7
1 Oracle Developer Studio 12.5 发行版简介	9
Oracle Developer Studio 概述	9
此发行版中的主要功能	10
2 C++ 编译器	13
关于 C++ 编译器	13
部分支持 C++14 标准	13
其他 C++ 编译器更改	14
3 C 编译器	15
从 -xc99 标志移到 -std 标志	15
Oracle Solaris 中 C 编译器的新缺省设置	16
__STDC_VERSION__ 的更改	16
对 Oracle Solaris 中用户应用程序的影响	17
-xlang=c89 的临时解决方法	17
C 编译器的其他更改	17
4 代码分析工具	19
关于代码分析工具	19
新增 codean 功能	19
新增 discover 功能	20
新增 uncover 功能	21
5 性能分析工具	23
关于性能分析器	23
性能分析器新功能	23
命令行工具的更改	30

对数据收集工具的更改	30
er_print 实用程序更改	32
其他命令和 API 的更改	32
6 调试工具	33
关于 dbx 调试器	33
新增和更改的 dbx 功能	33
7 Oracle Developer Studio IDE	35
关于 Oracle Developer Studio IDE	35
新增和已更改的 IDE 功能	35
可固定的终端	36
新建项目向导	37
远程开发的改进	37
混合式开发 (Java 和 C/C++) 支持	38
编辑多个文件的属性	39
支持 Doxygen C++ 注释	40
复合语句的代码折叠	40
8 OpenMP API	43
OpenMP	43
9 其他更改	45
编译器中的更改	45
编译器通用的新增和更改的功能	45
Fortran 编译器	46
新静态分析功能	46
性能库更改	48
数学库更改	48
索引	51

使用本文档

- 概述 – 介绍了此 Oracle Developer Studio 12.5 发行版的编译器和工具中的新增和更改的功能
- 目标读者 – 应用程序开发者、系统开发者、架构师、支持工程师
- 必备知识 – 编程经验、软件开发测试以及构建和编译软件产品的能力

产品文档库

有关该产品及相关产品的文档和资源，可从以下网址获得：http://docs.oracle.com/cd/E60778_01。

反馈

可以在 <http://www.oracle.com/goto/docfeedback> 上提供有关本文档的反馈。

Oracle Developer Studio 12.5 发行版简介

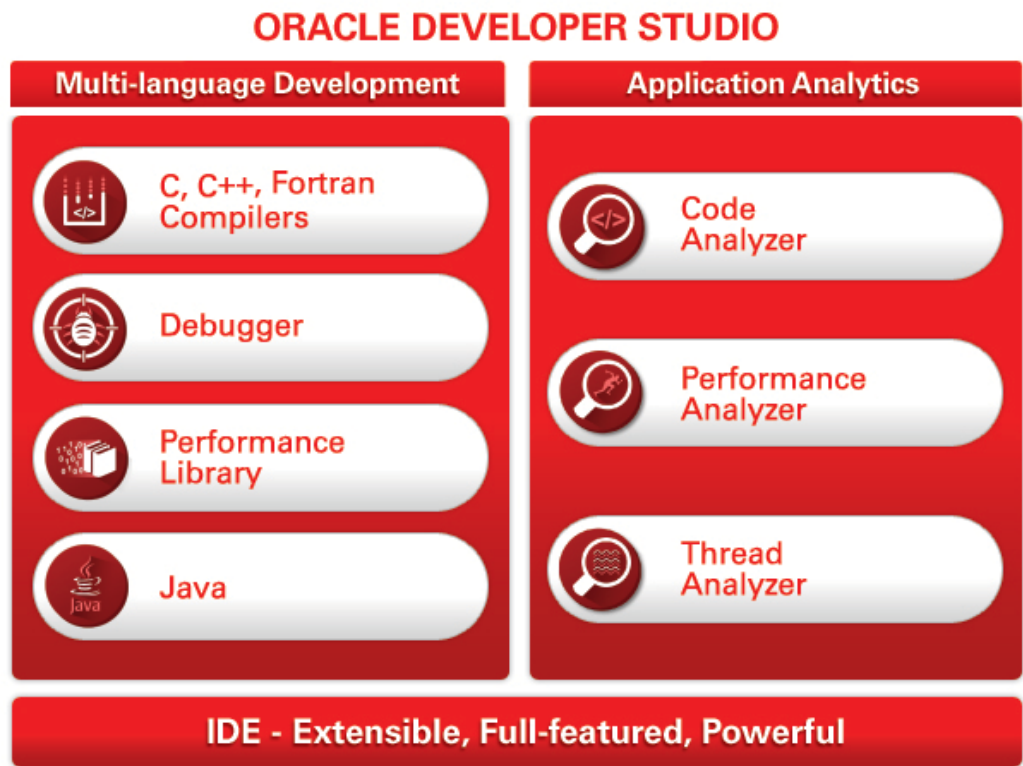
本章概述了此发行版中的主要更新。

- [“Oracle Developer Studio 概述” \[9\]](#)
- [“此发行版中的主要功能” \[10\]](#)

Oracle Developer Studio 概述

Oracle Developer Studio 包括编译器套件、分析套件以及图形集成开发环境 (integrated development environment, IDE)，该环境已调整以与那两个套件中的编译器和工具配合使用。它们一起提供一个开发环境，已进行了优化来开发在 Oracle Sun 硬件上具有最佳性能的应用程序。

图 1 图中显示了编译器和分析套件与 IDE 集成



此发行版中的主要功能

Oracle Developer Studio 12.5 提供高度优化的编译器、高级分析工具和可以识别多种语言的 IDE，以便轻松开发适用于 Oracle Solaris 和 Linux 操作系统的快速、可靠且安全的应用程序。Oracle Developer Studio 工具经过优化，可为整个硬件和软件堆栈提供有力补充，并支持开发团队更快速地编写更优质的代码。

- 加强安全屏障
 - 开发者工具中集成的 Oracle SPARC M7 芯片保护内存 (Silicon Secured Memory, SSM) 提供实时的内存访问检查，新增的库支持应用程序在定制内存分配器中使用 SSM。有关更多信息，请参见[第 4 章 代码分析工具](#)。

- 内置的源代码编译期间安全检查可帮助在开发和测试流程的早期阶段识别安全漏洞。有关更多信息，请参见[第 2 章 C++ 编译器](#)、[第 3 章 C 编译器](#)和[“编译器中的更改” \[45\]](#)。
- 应用程序运行时的堆栈溢出自动保护功能可以最大限度减少可能的安全漏洞
- IDE 中的安全编码提示会检测使用的不安全和已过时的函数，并提供关于更安全的替代方式的建议。有关更多信息，请参见[第 7 章 Oracle Developer Studio IDE](#)。
- 提高生产效率
 - 利用扩展的 GNU 支持和二进制兼容性，轻松编译开源应用程序。有关更多信息，请参见[《Oracle Developer Studio 12.5 : GCC 兼容性指南》](#)。
 - 提供常用的 C++14 功能并全面支持 C++11 和 C11，包括并发处理和原子库
 - 改进了测试功能并支持最新的 Boost 库
 - 可以轻松地对 Java、C、C++ 应用程序进行准确的性能分析。有关更多信息，请参见[第 5 章 性能分析工具](#)。
 - 可以对大型企业应用程序进行高效的代码编辑，解析时间大大减少，速度比开源替代方式最多快 7 倍。有关更多信息，请参见[第 7 章 Oracle Developer Studio IDE](#)。
- 最大限度地提高性能
 - 利用针对最新的 Oracle 系统（SPARC 和 x86，包括 SPARC S7）所做的性能优化，代码生成速度最多提高 4.5 倍
 - 使用最新的 SPARC 硬件计数器更新了性能分析器。有关更多信息，请参见[第 5 章 性能分析工具](#)。
 - 所有工具以 64 位二进制文件形式提供

C++ 编译器

本章介绍了此 Oracle Developer Studio C++ 编译器发行版中新增和更改的功能。

- [“关于 C++ 编译器” \[13\]](#)
- [“部分支持 C++14 标准” \[13\]](#)
- [“其他 C++ 编译器更改” \[14\]](#)

关于 C++ 编译器

本节提供了 Oracle Developer Studio 12.5 C++ 5.14 编译器发行版中引入的新功能和已更改功能的摘要列表。

C++ 编译器 (cc) 根据指定的命令行选项，生成面向特定操作系统、处理器、体系结构、内存模型 (32 位和 64 位)、浮点算法等等的代码。该编译器会自动将序列源代码并行化以生成在多核系统上有更好性能的二进制文件，并且还可以准备二进制文件以便其他 Oracle Developer Studio 工具更好地进行调试或分析。该编译器还支持 GNU C/C++ 兼容功能。

部分支持 C++14 标准

新 C++14 标准增强了 C++，提供了其他工具来使您的代码更准确且更安全。编译器在 Oracle 硬件上保持加速的 SPARC 和 x86 性能。

这是支持 C++14 标准的第一个 Oracle Developer Studio 发行版。支持以下 C++14 标准的功能：

- 二进制文本
- 固定大小的取消分配
- deprecated 属性
- 单引号数字分隔符

- 成员初始化和聚集

其他 C++ 编译器更改

下面列出了此 5.14 发行版中特定于 C++ 编译器的新增和更改的功能。

C++ 编译器更改包括“[编译器通用的新增和更改的功能](#)”[45]中介绍的更改。

有关详细信息，请参见《[Oracle Developer Studio 12.5 : C++ 用户指南](#)》和 [cc\(1\)](#) 手册页。

- 缺省编译模式中的更改 – Oracle Solaris 中的缺省编译模式是使用 `-library=Cstd` 的 `-compat=5`（使用 Sun ABI 和 `libCstd` 库的 C++03 模式）。Linux 中的缺省编译模式是 `-std=c++03`（使用 g++ ABI 和运行时库的 C++03 模式）。
- 支持 C++11 标准功能：
Oracle Developer Studio 12.5 C++ 通过增加以下项目完成了对 C++11 的支持：
 - 并发性和原子操作

注 - 使用原子功能时需要特别注意运行时支持。有关更多信息，请参见《[Oracle Developer Studio 12.5 : C 用户指南](#)》中的“[捆绑的原子库](#)”。

- 用户定义的文字
- 新编译器选项：
 - `-pedantic` – 针对在缺省情况下接受但不遵循 C++ 标准的代码发出警告或错误。
 - `-abiopt=[mangle5|mangle6]` – 仅在 `-compat=5` 模式下可用。缺省值是 `mangle6`，用于纠正名称改编。在 Oracle Solaris SPARC 和具有 `-m32` 选项的 Oracle Solaris x86 上，可以指定 `mangle5` 以和较旧编译器的可能错误的名称改编兼容。
 - `-xcheck=noreturn` – 通知编译器当已描述为 `does_not_return` 的例程返回时添加导致运行时错误的代码。
 - `-xatomic` 指定链接哪个原子支持运行时库。

C 编译器

本章介绍对 Oracle Developer Studio 12.5 发行版的 C 编译器进行的更改。

从 -xc99 标志移到 -std 标志

随着 C11 支持的引入，语言变种不再是 C89 和 C99 之间简单的二项选择，现在有了第三个选择：C11。

通过 Oracle Solaris Studio 12.3，选择是 C99 还是 C89 并由 -xc99 标志控制：

标志	语言变种
-xc99=all	C99 语言
-xc99=none	C89 语言

通过 Oracle Solaris Studio 12.4 和 Oracle Developer Studio 12.5，语言变种的选择（C89、C99 或 C11）应使用 -std 标志控制：

标志	语言变种
-std=c11	C11 语言
-std=c99	C99 语言
-std=c89	C89/C90 语言

以下是 -xc99 标志和 -std 标志之间的简单映射：

-xc99 标志	-std 标志
-xc99	-std=c99
-xc99=all	-std=c99
-xc99=all,lib	-std=c99

-xc99 标志	-std 标志
-xc99=all,no_lib	-std=c99 -xlang=c89
-xc99=none,lib	-std=c89 -xlang=c99
-xc99=none,no_lib	-std=c89
-xc99none	-std=c89

以下用于控制语言变种的选项在 Oracle Developer Studio C 编译器中已过时或很快就会过时：-Xc、-Xa、-Xt、-xc99。

- -xc99：选择 ISO C99 或 C89 语言
- -Xc：对使用非 ISO C 构造的程序发出错误和警告
- -Xa：接受 ISO C 以及 C 语言的扩展
- -Xt：接受 ISO C 以及 K&R C 兼容性扩展
- -Xs：接受 K&R C

如果以前使用 -Xa，则应改用 -std 选项以及 -pedantic 选项。使用 -Xt 或 -Xs 的传统代码需要转换为 ISO C 语言变种。

可使用 -xlang 选项控制与标准符合性相关的特定 libc 函数的行为。在 Oracle Solaris Studio 12.4 C 编译器中，缺省行为是 C11 语言构造和 C89 库行为。在此缺省模式中，`__STDC_VERSION__` (199409L) 表示 C89 标准。

在 Oracle Developer Studio 12.5 C 编译器中，语言功能和库行为缺省为 C11 模式，`__STDC_VERSION__` (201112L) 反映了 C11。

注意：在 Oracle Solaris Studio 12.3 中，此行为由子选项 -xc99=lib 控制。

Oracle Solaris 中 C 编译器的新缺省设置

以下功能是 Oracle Solaris 中 C 编译器的新缺省设置：

- C 编译器的缺省模式已更改
- 新的缺省模式可能会影响您的应用程序
- 如果需要，仍可使用旧模式

`__STDC_VERSION__` 的更改

以下更改说明了 `__STDC_VERSION__` 的新功能：

- 缺省情况下，在 Oracle Solaris 上以前版本的 C 编译器接受它们知道的所有 C99 和 C11 功能，但仅通过将 `__STDC_VERSION__` 预定义为 199409L 来声明为符合 C89。

- 缺省情况下，新的 C 编译器将 `__STDC_VERSION__` 预定义为 201112L，声明符合 C11。

对 Oracle Solaris 中用户应用程序的影响

以下信息说明对 Oracle Solaris 平台中的用户应用程序的影响：

- 所包括的使用诸如 `_XOPEN_SOURCE`、`_POSIX_SOURCE` 和 `_POSIX_C_SOURCE` 之类的宏的文件和功能测试可能以不同方式进行解析。

例如，如果使用 `_POSIX_SOURCE`，则可能发生以下错误：

```
Compiler or options invalid for pre-UNIX 03 X/Open applications
```

问题在于，`_POSIX_SOURCE` 在技术上请求了 C89 编译器，如 [standards\(5\)](#) 手册页中所述。

如果您的应用程序测试较旧版本的标准（例如，`_POSIX_SOURCE`），可考虑更改代码以尝试较新版本（在此示例中是 `_XOPEN_SOURCE=600`）

- 测试 `__STDC_VERSION__` 的用户应用程序可能以不同方式进行解析

-xlang=c89 的临时解决方法

您可能不希望明确选择 `-std=c89`，因为这样做将会禁用 C99 和 C11 功能。

而如果选择 `-xlang=c89`，则会使新的 C 编译器接受与以前版本的 C 编译器相同的程序并定义相同的 `__STDC_VERSION__`。

C 编译器的其他更改

C 编译器的更改包括“[编译器通用的新增和更改的功能](#)”[45]中介绍的更改，以及以下其他更改。

- 许多 SIMD (`_m128{d|i}`) 内部函数是为 SPARCACE 和 SPARCACE+ 提供的。
- 新编译器选项：
 - `-xcheck=noreturn` 通知编译器当已描述为“不返回”的例程返回时添加导致运行时错误的代码。
 - `-xatomic` 指定链接哪个原子支持运行时库。
- 支持以下 C11 功能：
 - 原子对象

注 - 使用原子功能时需要特别注意运行时支持。有关更多信息，请参见 [《Oracle Developer Studio 12.5 : C 用户指南》](#) 中的“捆绑的原子库”。

- 类型通用表达式 (_Generic)

有关更多信息，请参见 [cc\(1\)](#) 手册页和 [《Oracle Developer Studio 12.5 : C 用户指南》](#)。

代码分析工具

代码分析工具套件可检测常见的编码错误（包括内存泄漏和访问违规），使开发者可以更快地编写错误更少的更好代码。

本章介绍此 Oracle Developer Studio 发行版中代码分析工具的新增功能和更改的功能，其中包含以下各节：

- “关于代码分析工具” [19]
- “新增 codean 功能” [19]
- “新增 discover 功能” [20]
- “新增 uncover 功能” [21]

关于代码分析工具

代码分析工具使用静态、动态和代码覆盖分析来检测许多常见的编码错误（包括内存泄漏和内存访问违规），从而帮助应用程序变得更加可靠。Previs 在编译时执行静态分析，discover 在应用程序运行时执行动态分析，以确定代码质量问题。uncover 工具分析代码覆盖数据以提供有关测试套件未涵盖的函数的信息，并告知如何通过覆盖这些函数受益。

使用代码分析器图形工具或 codean 命令行实用程序可以查看三种类型的分析，以便全面了解应用程序的漏洞，从而提高应用程序的正确性和可靠性。

新增 codean 功能

以下功能已添加到代码分析器命令行工具 codean。

- 新标签功能：
 - Previs、discover 和 uncover 报告的问题现在可以使用 false_positivewont_be_fixed 或 verified 标签进行标记。在管理代码分析工具找到的问题时，使用标签进行标记很有用。

- 可使用标签抑制问题。
- 新增测试套件：
 - 在通过测试套件运行应用程序时，来自 discover 的、使用 -a 选项保存的报告可由 codean 进行累计，以反映找到的问题
 - 摘要页显示测试套件揭示的函数的百分比

通常，有关代码分析器的更多信息，请参见代码分析器中的 "Help"（帮助）、[《Oracle Developer Studio 12.5：代码分析器用户指南》](#)、[《Oracle Developer Studio 12.5：代码分析器教程》](#)、[codean\(1\)](#) 手册页和 [code-analyzer\(1\)](#) 手册页。

新增 discover 功能

此发行版的 discover 内存分析工具中增加了以下功能。有关更多信息，请参见 [discover\(1\)](#) 手册页和 [《Oracle Developer Studio 12.5：Discover 和 Uncover 用户指南》](#)。

- 使用应用程序数据完整性 (Application Data Integrity, ADI) 进行硬件辅助检查 – 在 SPARC M7 平台上提供更快内存访问检查。此功能是在 Oracle Solaris Studio 12.4 04/15 PSE 中引入的。有关更多信息，请参见 [《Oracle Developer Studio 12.5：Discover 和 Uncover 用户指南》](#) 中的“使用芯片保护内存 (Silicon Secured Memory, SSM) 的硬件辅助检查”。
- 新 ADI 帮助应用程序库 – 新库使用 ADI 功能，适用于不使用标准的 malloc () 和 free () 调用的程序进行内存管理。用户应使用 [《Oracle Developer Studio 12.5：Discover 和 Uncover 用户指南》](#) 中的“定制内存分配器和 discover ADI 库”中所述的 ADI API 并与 libadihelper.so 库进行链接。有关更多信息，请参见 [《Oracle Developer Studio 12.5：概述》](#) 中的“Oracle Developer Studio 代码安全性检查 – Discover ADI”和 [libadiplugin\(3\)](#) 手册页。
- 减少 UMR 和 PIR 的误报 – 误报常见的原因是存在 discover 工具无法识别的非 Oracle Developer Studio 生成的库和系统调用。
- SIGCHLD 信号句柄改进 – 如果程序安装了 SIGCHLD 信号句柄，通过 discover 检测的二进制文件不再失败了。
- 内存使用量更低 – 通过 discover 检测的二进制文件使用更少内存。
- 交互调试功能改进 – 使用 dbx 时，可改进通过 discover 检测的二进制文件的交互调试功能。
- 地址空间布局随机化处理 – discover 实用程序可处理启用了地址空间布局随机化 (Address Space Layout Randomization, ASLR) 的应用程序。

新增 uncover 功能

此发行版的 Uncover 代码覆盖工具中增加了以下功能。

- 改进了检测时间和内存消耗。

有关更多信息，请参见 [uncover\(1\)](#) 手册页和 [《Oracle Developer Studio 12.5 : Discover 和 Uncover 用户指南》](#)。

性能分析工具

性能分析工具配合使用，可供您分析应用程序的行为和找到影响性能的问题症结。

本章介绍此 Oracle Developer Studio 发行版中性能分析工具的新增和更改的功能。

关于性能分析器

利用性能分析器可以深入了解应用程序的行为，从而能够找出代码中的问题方面。性能分析器可确定哪些函数、代码段和源代码行占用的系统资源最多。性能分析器可以分析单线程、多线程和多进程应用程序，然后提供分析数据以帮助确定可提高应用程序性能的方面。

性能分析器由一组命令和工具构成，包括：collect 实用程序，收集用户级程序的分析数据；er_kernel 实用程序，收集 Oracle Solaris 内核的分析数据；er_print 实用程序，以文本格式显示分析信息；以及性能分析器 GUI，以图形方式显示分析信息。

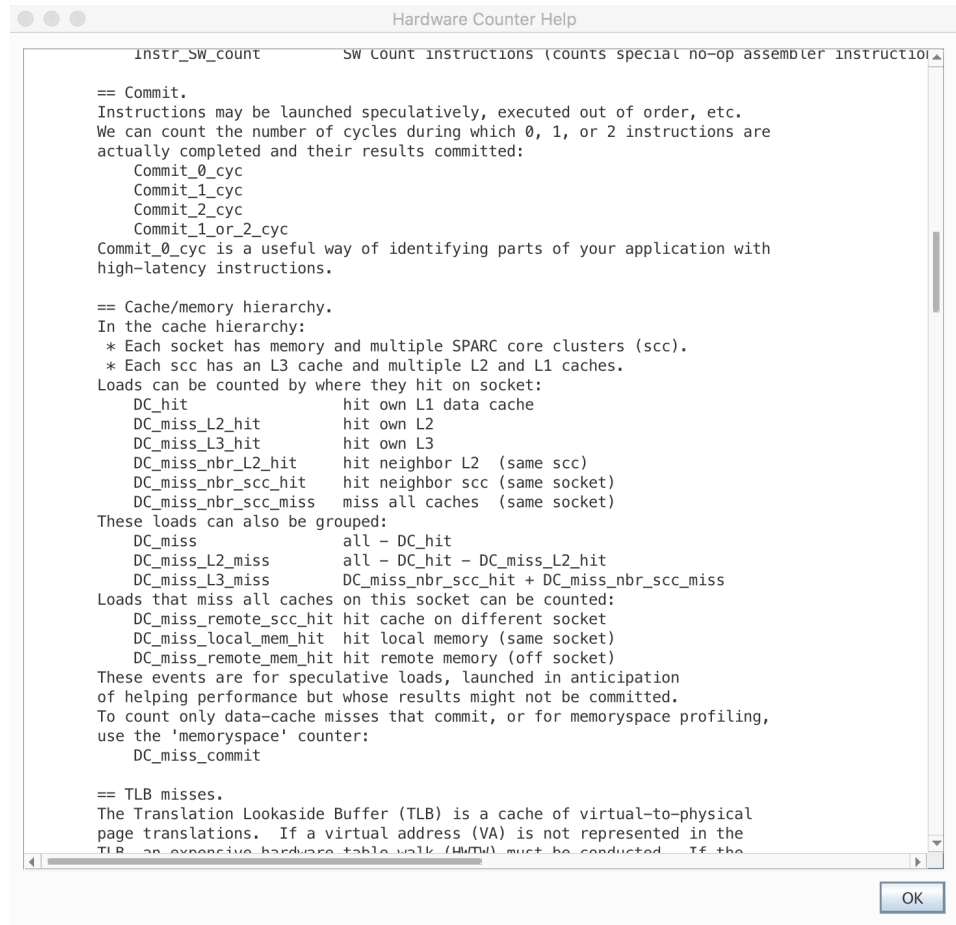
线程分析器是一款可让您专注于多线程问题的相关工具。

有关更多信息，请参阅[《Oracle Developer Studio 12.5：性能分析器》](#)和[《Oracle Developer Studio 12.5：性能分析器教程》](#)。

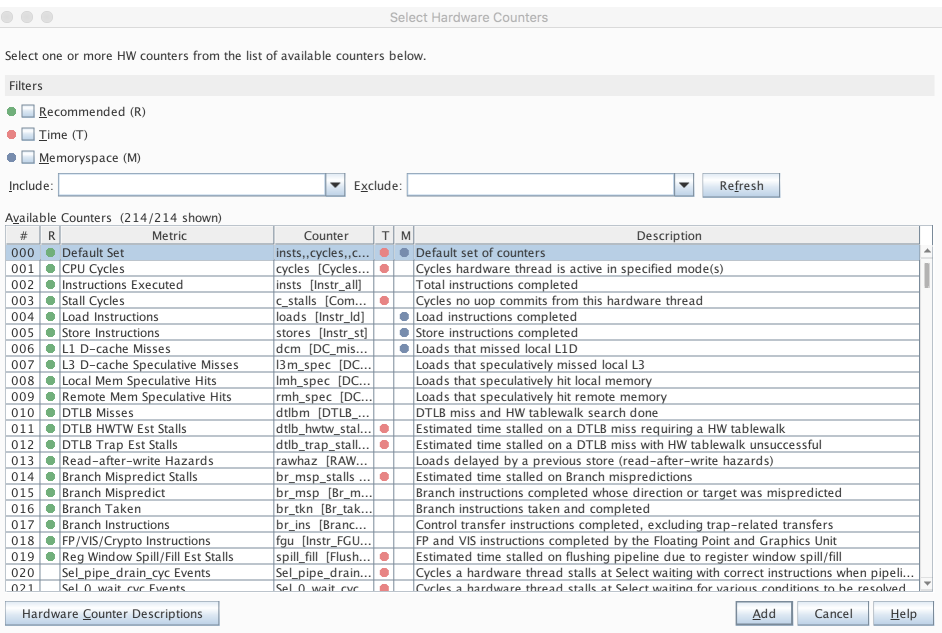
性能分析器新功能

本节概述了此发行版性能分析器和相关工具的新功能。有关更多信息，请参见性能分析器中的 "Help"（帮助）。

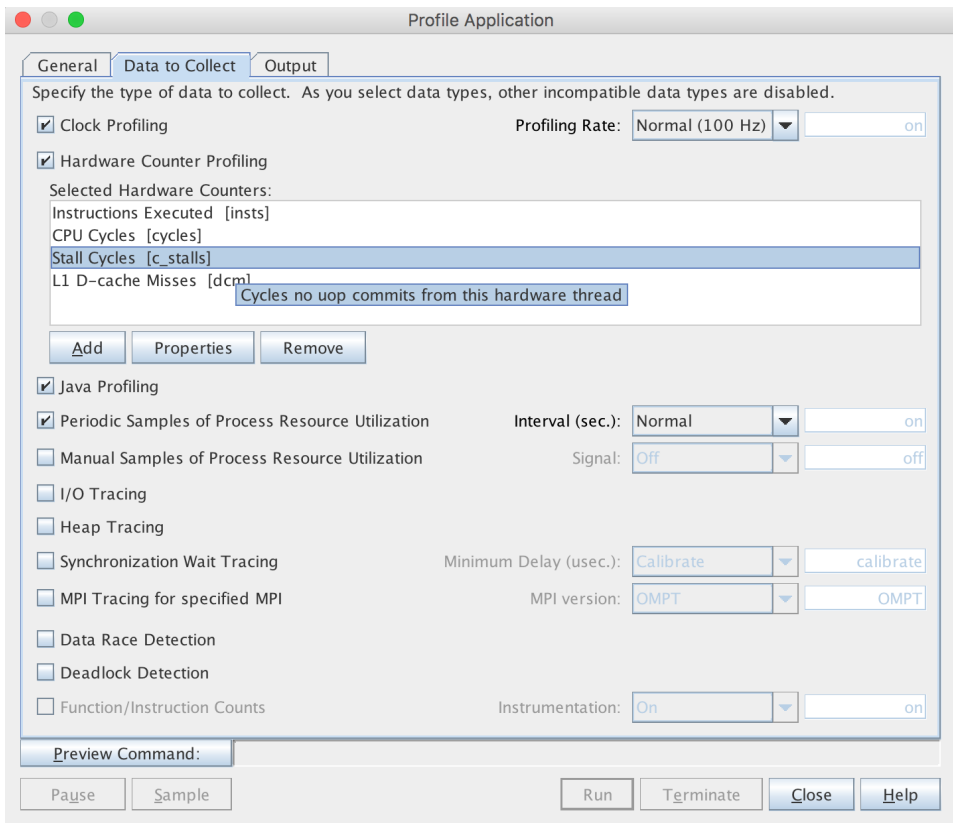
- 简化的硬件计数器分析
 - 新的特定于处理器的帮助提供重要硬件计数器的概述（仅限 SPARC）



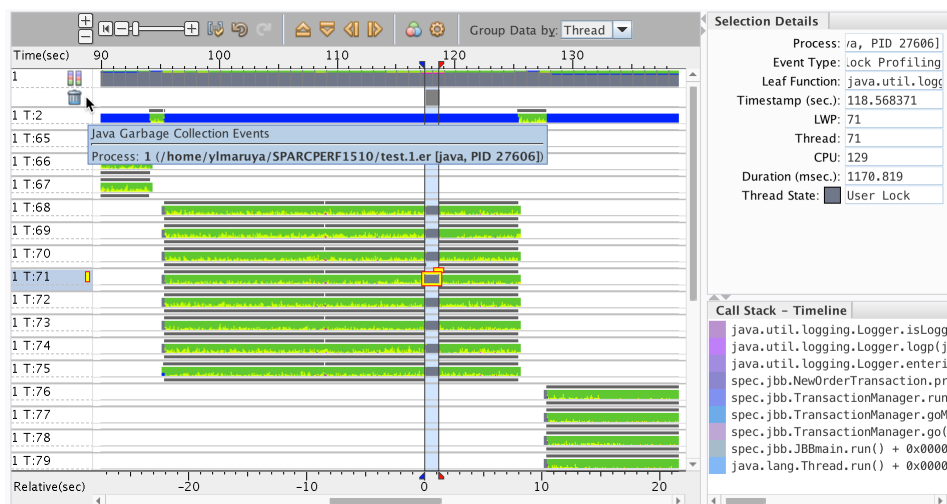
- 新的 "Select Hardware Counters" (选择硬件计数器) 对话框更详细地描述计数器。另外，它还更便于您过滤和添加计数器。



- 新增了一个用来自动选择合适分析率的 auto 选项。
- 更新了受支持硬件的缺省硬件计数器集。
- 更新了受支持平台的内存空间分析视图。（在 x86 系统上，内存空间分析至少需要 Oracle Solaris 11.3）
- 简化了用于选择硬件计数器的工作流



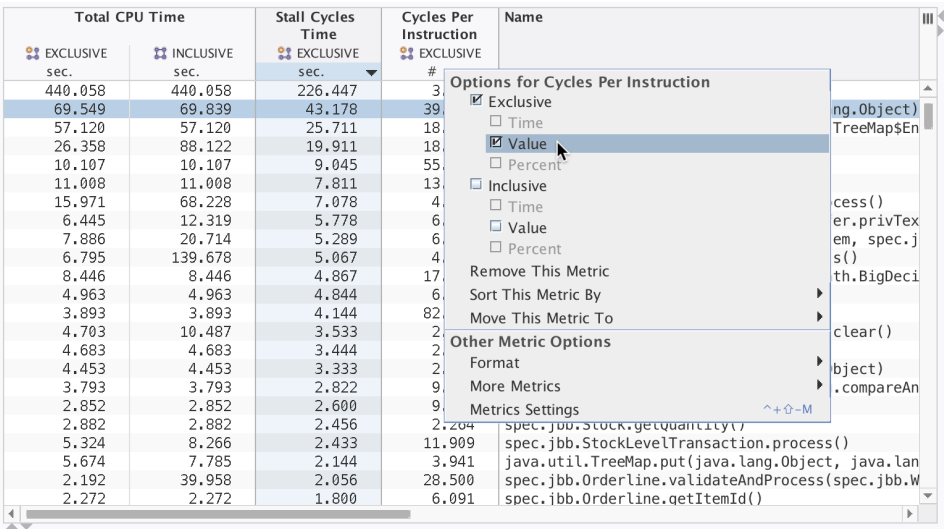
- **Java 分析增强功能**
 - 有关 Java 垃圾收集器事件的信息现在显示在 Overview（概述）和 Timeline（时间线）中。



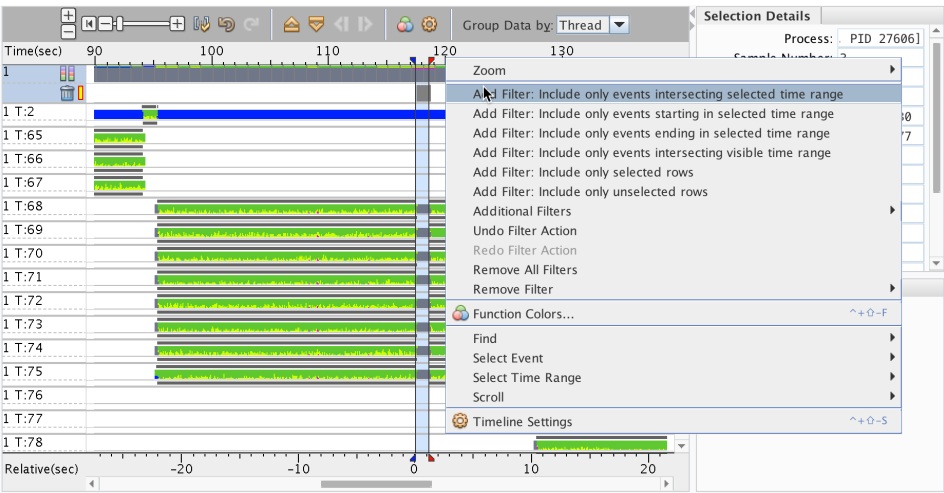
- 改进了性能度量对函数、源、字节代码和机器代码的归属。
- 度量显示增强功能

大多数数据视图都改进了度量显示，使您可以更方便地控制数据的显示方式：

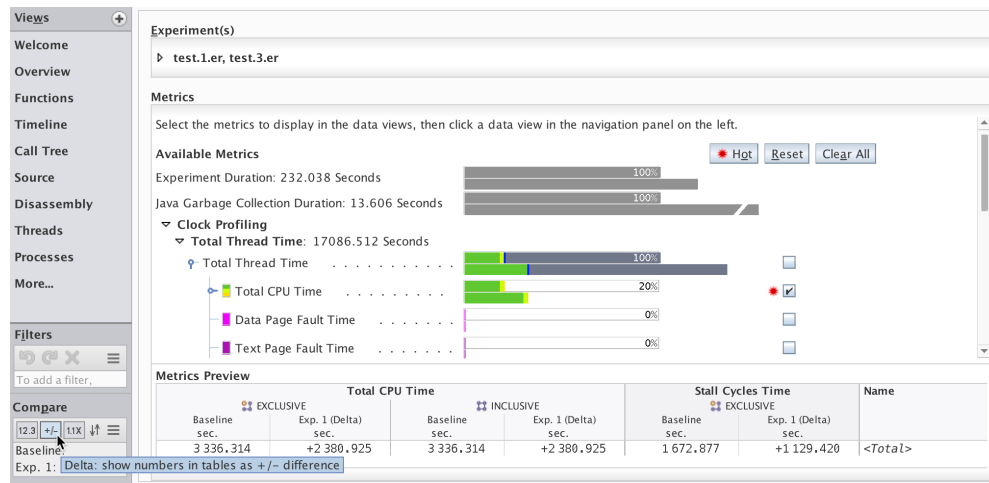
 - 数据列进行了分组，可以更好地显示包含和独占度量。
 - 列标题在顶角提供了控件，您可以单击这些控件来配置度量或者从视图中全部删除这些度量。
 - 以前位于 "Overview"（概述）屏幕中用于为显示度量选择时间或百分比的控件现在通过列标题控件提供。



- 时间线增强功能
 - 现在可以从标尺中选择时间范围。
 - 现在，可以从标尺中的上下文菜单中访问按时间或按行过滤。



- 比较实验增强功能
- 借助 "Performance Analyzer" (性能分析器) 主窗口中新增的 "Compare" (比较) 面板，可以在查看比较的数据时，使用按钮在绝对值、增量和比率模式之间进行切换。



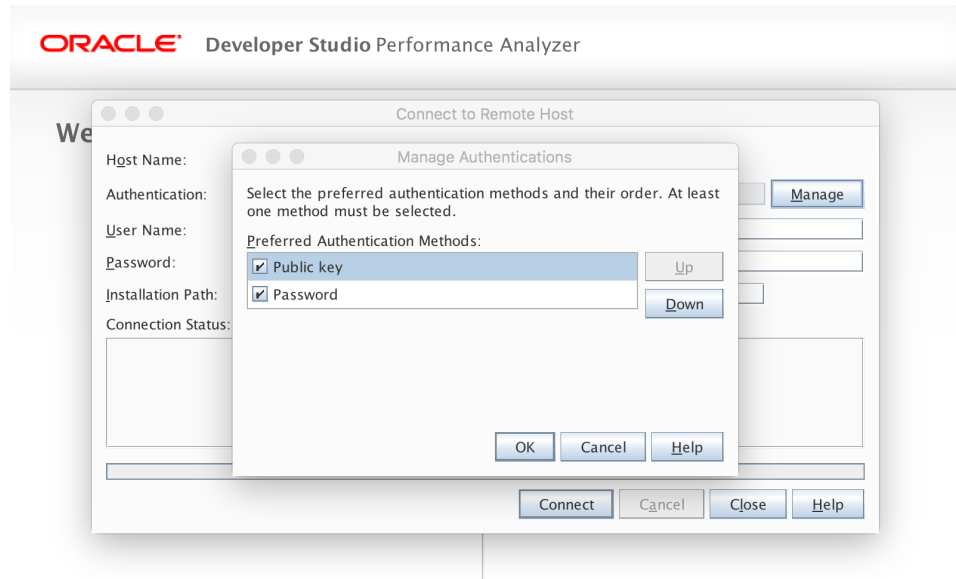
有关更多信息，请参见比较实验。

■ 调用树增强功能

"Call Tree" (调用树) 视图新增了 "Copy All" (全部复制) 选项，该选项以文本形式复制调用树，您可以将它们粘贴到一个文本文件中。

■ 远程分析器增强功能

- 远程分析器现在支持多种验证方法。



- 远程分析器提供更好的错误消息并改善了性能。

命令行工具的更改

本节介绍各个命令行性能分析工具的更改。有关更多信息，请参见每个命令行工具相应的手册页。

对数据收集工具的更改

数据收集工具包括 `collect` 命令、`dbx collector` 命令和 `er_kernel` 命令。这些工具全都用于分析程序，以收集数据和创建实验，性能分析器或 `er_print` 可以读取此类数据和实验。所有数据收集工具有如下更改。

- 对 Java 实验自动完成 Java 垃圾收集的跟踪。
- 对于 Java，通过可选的 `<scope>` 修饰符对 `synctrace` 功能进行了扩展；`<scope>` 对于本机 API 跟踪是 `n`，对于 Java API 跟踪是 `j`，或对于这两个 API 跟踪是 `nj`。`nj` 是缺省值。
- Oracle SPARC 和 x86（包括 Haswell-E/EP）上的硬件计数器有更新的支持和内存空间视图。

- 新的硬件计数器选项 `-auto` 可帮助选择适当的分析速率。
- 使用 `collect -h` 和 `er_kernel-h` 可获得有关使用 SPARC 硬件计数器的新指南。
- 与 `dlopen()`、`dlopen()`、`dlclose()`、`exit()` 和 `Exit()` 有关的问题已解决。
- 当指定 `CLONE_VM` 时在 Linux 上创建的线程不会被跟踪。
- 线程的最大数量限制已删除。

collect 实用程序更改

`collect` 实用程序是在运行时分析应用程序的工具，可以收集数据和创建实验，且性能分析器或 `er_print` 可以读取此类数据和实验。

除了所有数据收集工具通用的更改外，本发行版中的 `collect` 实用程序还有如下更改：

- 允许对 Java 目标进行堆跟踪，但它仅跟踪本机分配，不跟踪 Java 分配
- 不再识别 `-R` 参数。

dbx collector 的更改

`dbx collector` 是 `dbx` 调试器的子命令，可用于收集性能数据。有关更多信息，请参见 [collector\(1\)](#) 手册页。

除了所有数据收集工具通用的更改外，本发行版中的 `dbx collector` 命令还有如下更改：

- 12.4 版中的多个错误已修复。

er_kernel 实用程序更改

`er_kernel` 命令分析 Oracle Solaris 内核，并生成可在性能分析器或 `er_print` 中检查的实验。

除了所有数据收集工具通用的更改外，本发行版中的 `er_kernel` 实用程序还有如下更改：

- 可以更准确地跟踪进程创建和终止。
- `er_kernel -h` 输出的格式已改进。

有关更多信息，请参见 [er_kernel\(1\)](#) 手册页。

er_print 实用程序更改

er_print 实用程序会生成纯文本版本、可由性能分析器显示的数据视图。输出显示标准输出中。

er_print 实用程序在此发行版中进行了如下更改：

er_print 实用程序有如下更改：

- 12.4 版中的多个错误已修复。
- 使用 Oracle Developer Studio 进行编译时所使用的标志显示在源报告和反汇编报告中。
- 在实验标题中报告计算机型号信息。
- overview 命令已实现。

有关更多信息，请参见 [er_print\(1\)](#) 手册页。

其他命令和 API 的更改

libcollector API 有如下更新：

- libcollectorAPI.a 和 libfcollector.a 的静态版本现已可用。
- 对手册页进行了重写，以更好地描述 Java API。请参见 [libcollector\(3\)](#)。

调试工具

Oracle Developer Studio 提供了命令行 dbx 调试器以及用于使用 dbx 的 dbxtool 图形工具。调试器还集成到 IDE 中。有关使用 IDE 进行调试的更多信息，请参见[第 7 章 Oracle Developer Studio IDE](#)。

本章包含有关调试工具中新增功能的以下主题：

- [“关于 dbx 调试器” \[33\]](#)
- [“新增和更改的 dbx 功能” \[33\]](#)

关于 dbx 调试器

dbx 调试器是一个交互式源代码级事后和实时调试工具。可以在命令行通过 dbxtool 图形界面和在 Oracle Developer Studio IDE 中使用该调试器。dbx 调试器是可脚本化并且多线程感知的。

新增和更改的 dbx 功能

以下是 dbx 中添加或更改的功能。有关更多信息，请参见 [《Oracle Developer Studio 12.5：使用 dbx 调试程序》](#)、[dbx\(1\)](#) 手册页和 dbx 帮助文件。要访问 dbx 帮助文件，请键入以下命令：

```
% dbx
(dbx) help
```

- dbx 版本已从 8.0 更新到 8.1。
- 支持压缩调试信息。请参见 [《Oracle Developer Studio 12.5：使用 dbx 调试程序》](#) 中的“压缩调试节（仅限 Oracle Solaris）”。
- 用于 whatis 命令的新选项 -a。-a 选项仅输出数据成员。
- 新的 dbx 环境变量 output_data_member_only。如果设置为 on，则仅输出数据成员。
- 增加了处理与位置无关的可执行文件 (Position Independent Executable, PIE) 的支持。

- 增加了对 SPARC 上 Oracle Solaris 11 的芯片保护内存 (Secured Silicon Memory, SSM) 的支持。使用 dbx 命令 `adi` 可使用此功能。
- 支持以下 C++11 功能：用户定义的文本。
- 支持以下 C++14 功能：二进制文本；单引号数字分隔符。
- 支持以下 C11 功能：类型通用表达式。
- 对以下寄存器名称进行了更改，以在 Intel 平台中更好地保持一致：

原名称	新名称
<code>fs_base</code>	<code>fsbase</code>
<code>gs_base</code>	<code>gsbase</code>
<code>fcwd</code>	<code>fcw</code>
<code>fswd</code>	<code>fsw</code>
<code>ftw</code>	<code>fctw</code>
<code>mxcr_mask</code>	已删除
<code>uesp</code>	已删除

有关更多信息，请在 dbx 下发出 `help changes` 命令来访问 dbx 帮助文件。

Oracle Developer Studio IDE

Oracle Developer Studio IDE 为喜欢图形编程环境的用户集成了 Oracle Developer Studio 的许多组件。

本章阐述下列主题：

- [“关于 Oracle Developer Studio IDE” \[35\]](#)
- [“新增和已更改的 IDE 功能” \[35\]](#)

关于 Oracle Developer Studio IDE

Oracle Developer Studio 提供了图形集成开发环境 (integrated development environment, IDE)，其生成于 NetBeans 平台之上，配置为使用 Oracle Developer Studio C、C++ 和 Fortran 编译器、dmake 分布式 make 命令和 dbx 调试器。IDE 还与分析套件的一些分析器工具集成，从而您可以分析代码而不离开 IDE。

用于启动 IDE 的命令是 devstudio。有关此命令的详细信息，请参见 devstudio(1) 手册页。

有关 IDE 的完整文档，请参见 IDE 中的帮助。有关使用 IDE 基本功能的逐步说明，请参见 [《Oracle Developer Studio 12.5：IDE 快速入门教程》](#)。

新增和已更改的 IDE 功能

以下是 Oracle Developer Studio IDE 中添加或更改的功能：

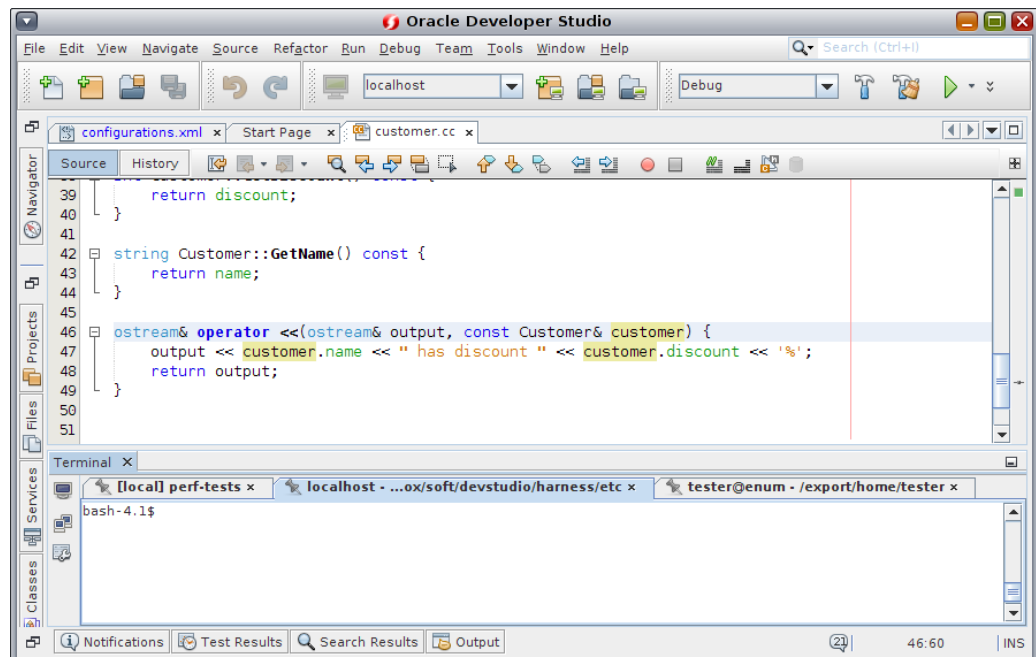
- 新的可固定终端 – 有关更多信息，请参见[“可固定的终端” \[36\]](#)。
- 新建基于现有源代码的项目向导，使预构建和构建步骤更清晰易懂。 – 有关更多信息，请参见[“新建项目向导” \[37\]](#)。
- 远程开发已改进。 – 有关更多信息，请参见[“远程开发的改进” \[37\]](#)。
- 混合式开发 (Java 和 C/C++) 支持。 – 有关更多信息，请参见[“混合式开发 \(Java 和 C/C++\) 支持” \[38\]](#)。

- 编辑多个文件的属性 – 有关更多信息，请参见[“编辑多个文件的属性” \[39\]](#)。
- 支持 Doxygen C++ 注释 – 有关更多信息，请参见[“支持 Doxygen C++ 注释” \[40\]](#)。
- 复合语句的代码折叠 – 有关更多信息，请参见[“复合语句的代码折叠” \[40\]](#)。
- 从编辑器中的编译器提示导航到输出日志 – 有关更多信息，请参见 IDE 帮助主题“从编译器提示导航到构建日志”。
- SendTo 实用程序 – 有关更多信息，请参见 IDE 帮助主题“SendTo 实用程序”。
- 生成缺少的开关子句
- 调用图增强功能 – 有关更多信息，请参见 IDE 帮助主题“使用调用图”。
- 新审计和提示 – 包括安全编码建议 – 在 IDE 中，选择 "Options" (选项) -> "Editor" (编辑器) -> "Hints" (提示)，然后选择 C/C++ 语言以查看许多可用的新提示。有关更多信息，请参见 IDE 帮助主题“使用静态代码分析器”。
- 新重构 – 有关更多信息，请参见 IDE 帮助主题“通过重构引入函数和变量”。

可固定的终端

"Pin Tab" (固定标签) 操作已添加到 "Terminal" (终端) 弹出菜单。

图 2 可固定的终端

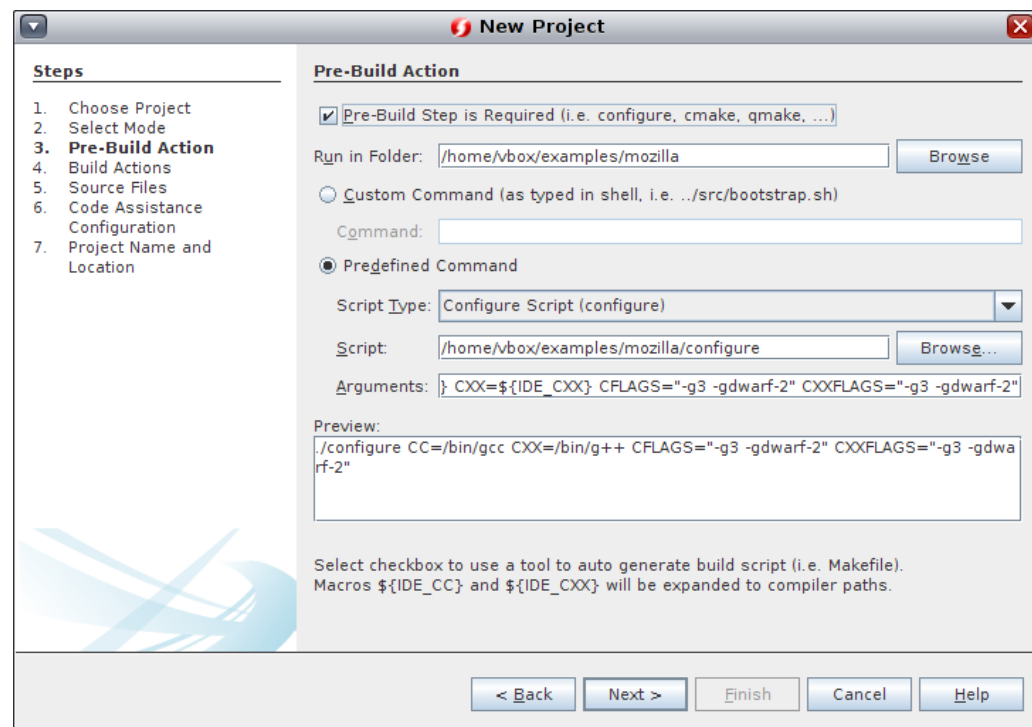


有关更多信息，请参见 IDE 帮助主题“固定终端”。

新建项目向导

在新建基于现有源代码的项目向导中，可在以后使用新的属性选项卡 "Pre-Build"（预构建）定制 "Pre-Build"（预构建）步骤。可从 "Project Properties"（项目属性）-> "Build"（构建）-> "Pre-Build"（预构建）访问此选项卡。下图显示了新建项目向导，在 "Pre-Build Action"（预构建操作）选项卡中：

图 3 新建项目向导

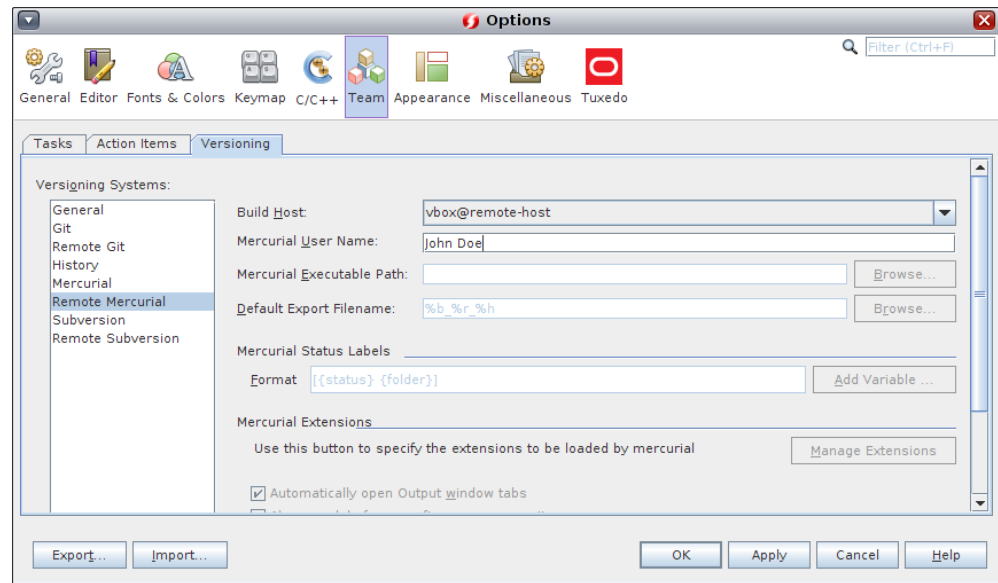


远程开发的改进

为远程开发添加了以下功能：

- 远程模式中的 SVN、Git 和 Mercurial 支持 – 此功能使您可以像本地用户那样在完全远程模式中使用 VCS。可以在 "Tools"（工具）-> "Options"（选项）-> "Team"

(团队) -> "Remote Mercurial/Git/Subversion" (远程 Mercurial/Git/Subversion) 以及 "Tools" (工具) -> "Options" (选项) -> "Fonts & Colors" (字体和颜色) -> "Remote Mercurial/GIT/Subversion" (远程 Mercurial/GIT/Subversion) 中定制远程 VCS 首选项。以下屏幕抓图显示了 "Options" (选项) 窗口中的 "Versioning" (版本控制) 选项：



- 完全远程项目的代码帮助支持
- 新建远程主机设置向导的新选项

有关更多信息，请参见 IDE 帮助主题“将完全远程模式与版本控制系统结合使用”和“为 C/C++/Fortran 开发配置远程主机”。

混合式开发 (Java 和 C/C++) 支持

IDE 可以帮助您将 C/C++ 项目与使用 Java 本地接口 (Java Native Interface, JNI) 和 Java 本地访问 (Java Native Access, JNA) 技术的 Java 项目进行集成。使用这些技术，可以从 Java 程序调用本机 C 和 C++ 程序。

如果您的 Java 项目使用 JNI 方法，则可以基于 Java 项目生成 C++ JNI 项目。JNI 项目以 Java 代码实现所有本机接口。有关如何创建 C++ JNI 库项目以及关联的 Java 和本机项目支持的 IDE 功能的信息，请参见 IDE 帮助主题“在混合式开发项目中使用 JNI”和“在混合式开发中使用 JNA”。

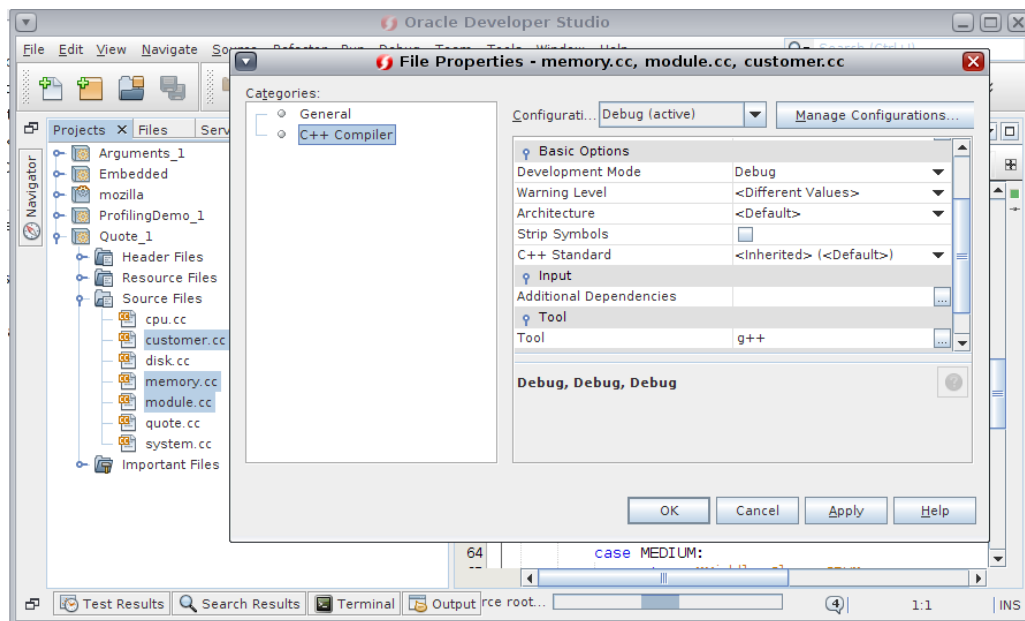
要使用混合式开发，您需要执行以下前提步骤。

1. 检查以确保有 Java JDK，并使用系统中安装的 Java JDK 运行 IDE 以使用此功能。
IDE 通常只需要 Java JRE。
2. 从 IDE 更新中心安装 Java SE 插件。
应安装五个模块。
3. 记下用户目录并注意以下要求：
 - 将所需的所有其他模块安装在指定的用户目录中。
 - 只有运行包含与其他模块关联的用户目录的 IDE 时，混合式开发才能工作。
 - 如果未安装到正确的用户目录，此功能将不起作用，或要求将 Java SE 重新安装到新的用户目录中。

编辑多个文件的属性

现在可以在项目中编辑多个文件的属性。按住 Shift 并单击可在 "Projects" (项目) 窗口中选择多个文件，然后单击右键并选择 "Properties" (属性)。此时将显示 "Properties" (属性) 窗口：

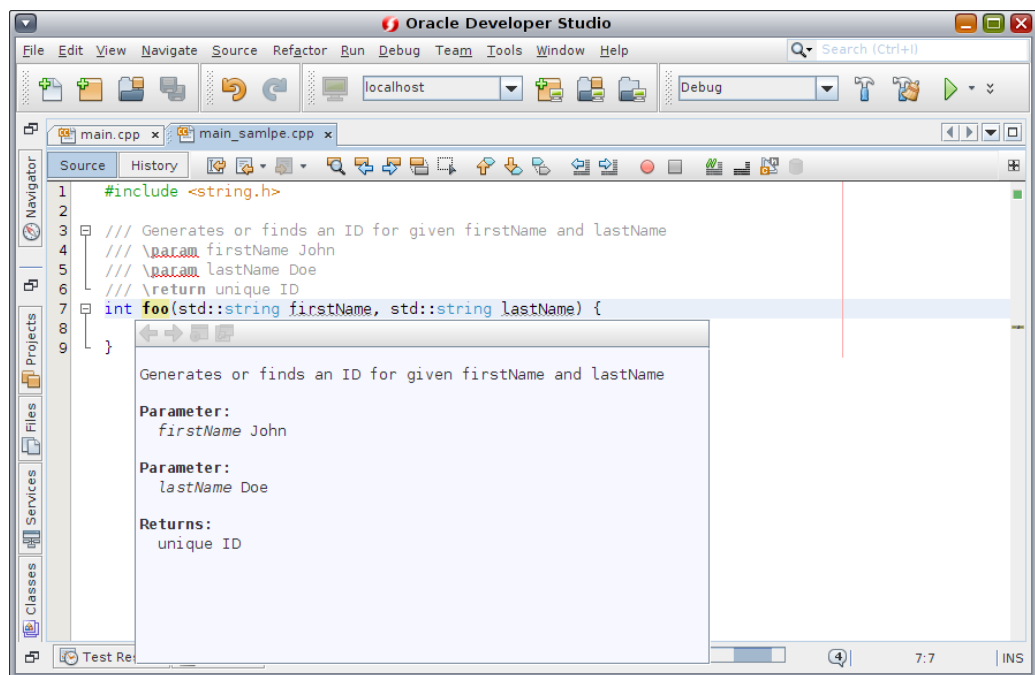
图 4 编辑多个文件的属性



支持 Doxygen C++ 注释

项目现在可以使用 doxygen 样式的单行 "///" 注释进行注释。

图 5 使用 Doxygen C++ 注释

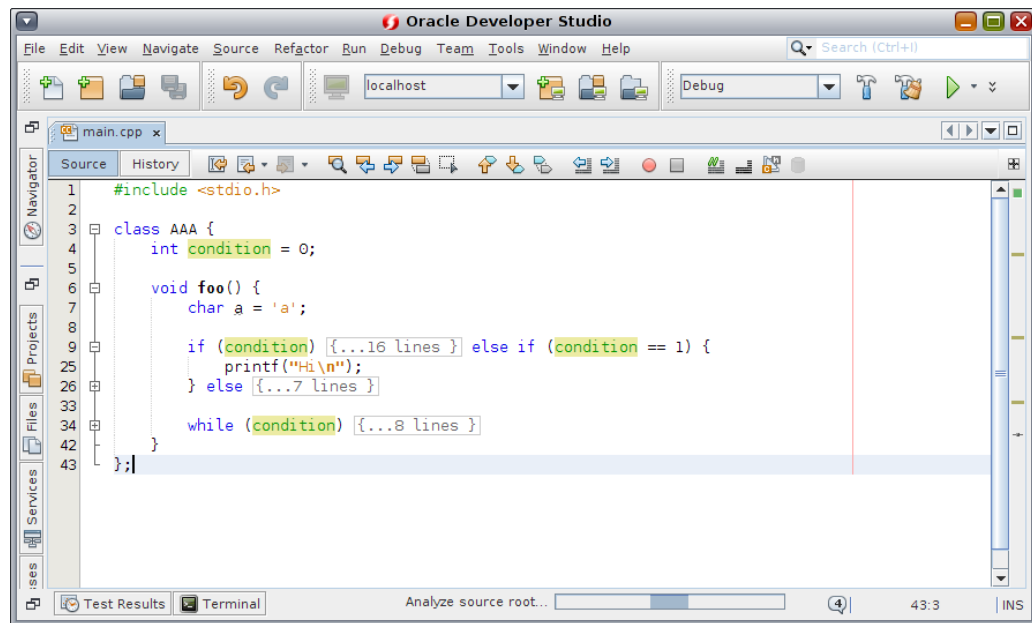


有关更多信息，请参见 IDE 帮助主题“向代码添加注释”。

复合语句的代码折叠

现在可以折叠复合语句，如 if-else、do-while 等。

图 6 IDE 中的代码折叠



有关更多信息，请参见 IDE 帮助主题“折叠 C 和 C++ 文件中的代码块”。

OpenMP API

本章介绍此 Oracle Developer Studio 发行版中 OpenMP API 支持的更改。

OpenMP

本节讨论 OpenMP API 的新增功能和更新。

- OpenMP 和 autopar 程序的缺省线程数是 `cores_per_chip` 数量的倍数。计算线程数的算法从 `cores_per_chip` 开始，然后检查该数量的连续倍数，选取不超过 32 并且不超过计算机上的核心数的最大数量。

有关 OpenMP 的更多信息，请参见 [《Oracle Developer Studio 12.5 : OpenMP API 用户指南》](#)。

其他更改

本章介绍 Oracle Developer Studio 软件的其他组件的新增和更改的功能。

- “编译器中的更改” [45]
- “性能库更改” [48]

编译器中的更改

以下部分介绍编译器中的更改，包括以下主题：

- “编译器通用的新增和更改的功能” [45]
- “Fortran 编译器” [46]
- “新静态分析功能” [46]

编译器通用的新增和更改的功能

下面列出了自上一发行版起对 C、C++ 和 Fortran 编译器进行的更改。可在编译器手册页中找到详细信息。特定于 C++ 编译器的更改已在[第 2 章 C++ 编译器](#)中进行了详细介绍。特定于 C 编译器的更改已在[第 3 章 C 编译器](#)中进行了详细介绍。

新硬件上的应用程序性能

每一个 Oracle Developer Studio 发行版都包括对 Oracle Sun 硬件服务器的性能改进。此发行版中包括对 SPARC M6、SPARC M7、SPARC T7、SPARC S7 和 Intel Broadwell/avx2_i 处理器的扩展支持。

其他编译器更改

以下列表介绍了影响 C、C++ 和 Fortran 编译器的其他更改：

- 支持 SPARC M6、M7、T7 和 S7 处理器。
- 支持 x86 数据空间分析。
- -xcheck 有新的缺省 -xcheck=stkovfl。
- 新编译器选项：
 - -features=[no]mergestrings 可使编译器将字符串字母以及其他合适的 const 或只读数据放入链接程序从中删除重复字符串的二进制文件的特定部分。此选项仅在 SPARC 中可用。
 - -xsecure_code_analysis 启用编译器安全代码分析，以便在编译时查找和显示可能的内存安全违规。
 - -xtarget=S7, -xchip=S7 在编译器驱动程序中受支持。

Fortran 编译器

Fortran 编译器通过针对 Fortran77、Fortran90 和 Fortran95 标准的创纪录运行时性能和兼容性选项来支持技术和科学应用程序开发。包括大多数 Fortran 2003 功能和 OpenMP 4.0 支持。Fortran 编译器与 C 和 C++ 编译器使用相同的高性能代码生成技术，从而确保结果应用程序为最新的基于 SPARC 和 x86 的 Oracle 系统生成最高性能的并行代码。

Fortran 编译器的更改包括“[编译器通用的新增和更改的功能](#)” [45]中介绍的更改，以及以下更改：

- 自由源代码格式行的最大长度从 132 个字符增加到了 250 个字符。

有关更多信息，请参见 [f95\(1\)](#) 手册页和《[Oracle Developer Studio 12.5 : Fortran 用户指南](#)》。

新静态分析功能

缺省情况下，对 C 和 C++ 编译器生成内存安全检查和编译时警告。

包括以下消息标记：

- SEC_UNINITIALIZED_MEM_READ
- SEC_UNINITIALIZED_BITOP
- SEC_UNDEFINED_RETURN_VALUE
- SEC_ARR_OUTSIDE_BOUND_READ
- SEC_ARR_OUTSIDE_BOUND_WRITE
- SEC_FREED_PTR_RETURN

- SEC_READ_FREED_PTR
- SEC_WRITE_FREED_PTR
- SEC_NULL_PTR_DEREF

向 `stderr` 发出警告并发出所有其他编译时错误和警告。它们由 `-erroff`、`-errtags` 和 `-errwarn` 命令行选项以及 `error_messages()` pragma 控制，与所有其他编译时消息类似。

如何使用 `-errwarn` 命令行选项

`-errwarn` 命令行选项可用于将内存安全检查警告变为致命错误。例如：

- `-errwarn=SEC_NULL_PTR_DEREF` 使任何空指针解除引用变为致命错误。
- `-errwarn=%all` 使所有编译时警告变为致命错误。

如何禁用静态错误检查

静态错误检查与编译器后端处理并行运行。在某些情况下，如负载沉重的系统或包含极复杂的流量控制的超大型模块，编译时间可能会增加。在编译时间绝对重要的情况下，可通过命令行选项 `-xsecure_code_analysis=no` 禁用静态错误检查，但这是以抑制可能严重的诊断消息为代价的。

另外，可以使用 Oracle Developer Studio 的缺省 config 文件功能禁用整个站点的静态错误检查。例如，可以使用配置文件 `cc.defaults` 或 `CC.defaults`。也可以使用 `c89.defaults` 或 `c99.defaults`。安装路径是 `install-dir/lib/compilers/etc/config.`。

可以使用 `SPRO_DEFAULTS_PATH` 环境变量禁用缺省的静态错误检查，无需对 `makefile` 进行更改：`SPRO_DEFAULTS_PATH=path`，其中 `path` 是包含缺省文件的目录。

使用 `error_messages()` Pragma 启用和禁用 SEC 消息

可使用 `error_messages()` pragma 根据源文件的指定区域选择性地启用和禁用 SEC 消息。

以下示例说明如何在特定语句上禁用特定 SEC 警告：

```
cat foo.c:
    <some code>
#pragma error_messages (off, tag-of-interest)
    <statement of interest>
#pragma error_messages (default|on, tag-of-interest)
    <remainder of code>
```

其中 `tag-of-interest` 是 SEC 前端警告之一。

禁用代码区域中的所有 SEC 警告：

```
cat foo.c:
<some code>
#pragma error_messages (off, SEC_UNINITIALIZED_MEM_READ,SEC_UNINITIALIZED_BITOP,
SEC_UNDEFINED_RETURN_VALUE,
SEC_ARR_OUTSIDE_BOUND_READ,SEC_ARR_OUTSIDE_BOUND_WRITE,SEC_DOUBLE_FREE,SEC_FREED_PTR_RETURN,
SEC_READ_FREED_PTR,SEC_WRITE_FREED_PTR,SEC_NULL_PTR_DEREF)
    line-or-lines-of-interest
#pragma error_messages (default|on, SEC_UNINITIALIZED_MEM_READ,SEC_UNINITIALIZED_BITOP,
SEC_UNDEFINED_RETURN_VALUE,
SEC_ARR_OUTSIDE_BOUND_READ,SEC_ARR_OUTSIDE_BOUND_WRITE,SEC_DOUBLE_FREE,SEC_FREED_PTR_RETURN,
SEC_READ_FREED_PTR,SEC_WRITE_FREED_PTR,SEC_NULL_PTR_DEREF)
<remainder of code>
```

性能库更改

Oracle Developer Studio 性能库是一组优化的高速数学子例程，用于解决线性代数和其他数字密集型问题。Oracle Developer Studio 性能库以来自 <http://www.netlib.org> 上的 Netlib 的公共域子例程集合为基础。Oracle 增强了这些公共域子例程，并将其捆绑成 Oracle Developer Studio 性能库。

在此发行版中，进行了以下更改：

- Oracle Developer Studio 性能库中的 LAPACK 升级为版本 3.5.0。在 Oracle Developer Studio 性能库中实现了 LAPACK 3.5.0 中的所有新功能，包括以下功能：
 - 删除 Fortran 95 接口的可选参数。
 - 基本 LAPACK 例程已更新为与 LAPACK 3.5.0 匹配
- STACKSIZE 环境变量的新的最低要求设置为每个线程 8 MB。

Oracle Developer Studio 性能库将大数据分为包含在硬件高速缓存中的多个块，然后将这些数据块复制到堆栈以便更好地进行定位。硬件运行速度更快意味着高速缓存更大，因此数据块和堆栈更大。为了更好地容纳当前硬件，每个线程的堆栈大小最低要求是所有平台上均为 8 MB。

有关更多信息，请参见《[Oracle Developer Studio 12.5：性能库用户指南](#)》。

数学库更改

在 Oracle Solaris Studio 12.4 和编译器的早期发行版中，<sunmath.h> 头文件包括要从 Fortran 调用的“包装”函数的 C/C++ 原型，以调用 libsunmath 中的多个非标准数学函数和浮点实用程序例程。这些“包装”函数本身就可以调用可从 C/C++ 直接调用的相应函

数，因此 C/C++ 程序没有必要使用它们。例如，“包装”函数 `r_atan2d_` 等效于 C 函数 `atan2df`。

在 Oracle Developer Studio 12.5 中，“包装”函数已过时。其在 `<sunmath.h>` 中的原型现在由预处理器宏 `__SUNMATH_DEPRECATED` 保护；在 C 程序中使用这些函数会生成警告：

```
example% cat func.c
#include <sunmath.h>
float func(float x, float y) {
    return r_atan2d_(&x, &y);
}
example% cc -c func.c
"func.c", line 4: warning: implicit function declaration:
r_atan2d_
example%
```

所生成的程序可能会产生错误结果。应将此类程序重写为直接使用相应的 C 函数：

```
return atan2df(x, y);
```

另外，可以在包含 `<sunmath.h>` 之前添加 `#define __SUNMATH_DEPRECATED prior`，或将 `-D__SUNMATH_DEPRECATED` 添加到在命令行上指定的编译器标志，但是请注意，“包装”函数在未来的发行版中将被删除。

索引

B

编译器, 13
 C, 17
 c++, 13, 14
 C++14 标准, 13
 Fortran, 46
 通用新功能, 45

C

codean 的更改, 19
collect 实用程序, 31

D

代码分析工具, 19
 Discover, 20
 Previser, 46
 Uncover, 21
 代码分析器, 19
 静态分析, 46
dbx, 33
 更改, 33
dbx collector 命令, 31
discover
 命令更改, 20

E

er_kernel 实用程序, 31
er_print 命令, 32

I

IDE (集成开发环境), 35

更改, 35

K

库, 45
 OpenMP, 43
 性能, 48

L

libcollector API, 32

S

数据收集, 30

U

uncover 命令更改, 21

X

性能分析器, 23, 23

Z

主要功能, 10

