

Oracle® Developer Studio 12.6：发行说明



文件号码 E83250-01
2017 年 7 月

文件号码 E83250-01

版权所有 © 2017, 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
Oracle Developer Studio 12.6 发行说明	9
系统要求	9
要求的系统软件包	10
安装说明	12
功能终止声明	12
第三方软件信息	12
发送使用数据到 Oracle	12
此发行版的文档	12
此发行版中的已知问题、限制和解决方法	13
编译器问题	13
工具问题	19
安装和卸载问题	24

使用本文档

- 概述—提供有关此 Oracle Developer Studio 12.6 发行版的已知问题和其他问题的信息。
- 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://www.oracle.com/pls/topic/lookup?ctx=E83242-01>

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 12.6 发行说明

此文档提供在安装和使用产品之前需要了解的信息。有关安装说明，请参见 [《Oracle Developer Studio 12.6：安装指南》](#)。

系统要求

Oracle Developer Studio 12.6 软件可以安装在基于 SPARC 或基于 x86 的平台上的 Oracle Solaris 10 或 Oracle Solaris 11.3 操作系统中，也可以安装在 Oracle Linux 操作系统中。

表 1 系统要求

系统资源	基于 SPARC 的系统上的 Oracle Solaris	基于 x86 的系统上的 Oracle Solaris	基于 x86 的系统上的 Linux
操作系统	Oracle Solaris 11.3（需要 SRU20 或更高版本） Oracle Solaris 10 1/13	Oracle Solaris 11.3（需要 SRU20 或更高版本） Oracle Solaris 10 1/13	OL 6.7+、UEK 3、UEK 4 OL 7.2+、UEK 3、UEK 4 RH 6.7+、7.2+
CPU	SPARC M 系列 SPARC T3 系统或更高版本	Intel Nehalem 系统或更高版本	Intel Nehalem 系统或更高版本
内存	最低：4 GB	最低：4 GB	最低：4 GB
安装程序所需的临时磁盘空间	2.2 GB	2 GB	2 GB
已安装产品需要的磁盘空间 使用 df -k 命令检查您的磁盘空间	1.2 GB	1.0 GB	1.0 GB
交换空间	Oracle Solaris 10：最低 8 GB Oracle Solaris 11：请参见 Oracle Solaris 文档了解交换大小建议内容	Oracle Solaris 10：最低 8 GB Oracle Solaris 11：请参见 Oracle Solaris 文档了解交换大小建议内容	最低：8 GB
系统软件相关项	Oracle Solaris 11：系统/头文件、sunpro-incorporation 和	Oracle Solaris 11：系统/头文件、sunpro-incorporation 和	Linux：开发/库软件包组，包括 glibc.i686、glibc-devel 和 glibc-devel.i686

系统资源	基于 SPARC 的系统上的 Oracle Solaris	基于 x86 的系统上的 Oracle Solaris	基于 x86 的系统上的 Linux
	developer-studio-utilities 软件包 Oracle Solaris 10: Entire Solaris Software Group Plus OEM Support、Entire Solaris Software Group 或 Developer Solaris Software Group	developer-studio-utilities 软件包 Oracle Solaris 10: Entire Solaris Software Group Plus OEM Support、Entire Solaris Software Group 或 Developer Solaris Software Group	
Java 版本	Java 7 (1.7.0_72) 或更高版本；Java 8 (1.8.0_20)	Java 7 (1.7.0_72) 或更高版本；Java 8 (1.8.0_20)	Java 7 (1.7.0_72) 或更高版本；Java 8 (1.8.0_20)

您可以选择在本地（客户机）系统上安装 IDE、性能分析器和 dbxtool，然后使用它们连接到运行 Oracle Developer Studio 产品的远程服务器上。客户机系统支持的配置包括以下各项：

- OS：Microsoft Windows 7 Professional 或更高版本；Ubuntu 9.10；Macintosh OS X 10.7 或更高版本
- Java：上表中指定的 Java 版本之一
- 这些组件所需的磁盘空间：300 MB

要求的系统软件包

在执行 Oracle Developer Studio 12.6 编译器和工具之前，请验证在其上运行软件的系统是否具有所需的系统软件包。下表列出了每个支持的操作系统所需的软件包。

表 2 开发者所需的系统软件

操作系统版本	系统软件包
Oracle Solaris 11.3	请参见 《Oracle Developer Studio 12.6: 安装指南》 以确保您的系统具有相应的 sunpro-incorporation 和 developer-studio-utilities 软件包版本。 检查是否已安装以下软件包： <pre> pkg:/developer/base-developer-utilities pkg:/developer/library/xprofile pkg:/diagnostic/cpu-counters pkg:/library/glib2 pkg:/library/libxml2 pkg:/library/zlib pkg:/runtime/python-27 pkg:/system/library/c++-runtime pkg:/system/library/libv12n pkg:/system/library/mmheap pkg:/system/library/openmp pkg:/system/linker pkg:/developer/library/lint pkg:/shell/bash </pre>

操作系统版本	系统软件包
	pkg:/shell/ksh93 pkg:/system/header pkg:/system/library/math pkg:/system/resource-mgmt/resource-pools pkg:/library/c++/stdc++
Oracle Solaris 10 1/13	SUNWhea SUNWarc SUNWcpp SUNWlibstdc++4 SUNWpic1h SUNWpic1r SUNWPython-devel SUNWcar SUNWpcu SUNWcsl SUNWcslr SUNWdpl SUNWgnome-base-libs SUNWlibC SUNWlibm SUNWlibms SUNWlibmsr SUNWlxml SUNWpython SUNWsprot SUNWzlib
Oracle Linux 6	开发/库软件包组和 32 位运行时支持库包括：
Oracle Linux 7	glibc glibc.i686 glibc-devel glibc-devel.i686 elfutils-libelf-devel elfutils-libelf-devel.i686 libgcc-4.8.5-4.el7.x86_64 zlib zlib.i686 libstdc++ libstdc++.i686 libgcc libgcc.i686 openssl openssl.i686

注 - developer-studio-utilities 和 studio-runtime 是“组”软件包，这是用于简化相关软件包组的安装的 IPS 快捷方式。它们完全由对其他软件包的相关项组成，但是没有自身的内容。

studio-runtime 是组软件包，它安装系统运行时支持库，这些库由通过 Oracle Developer Studio 编译器编译的程序使用。示例包括 Fortran 和 Sun 性能库。应该在已部署应用程序的系统上安装它。

developer-studio-utilities 是组软件包，它安装使用 Oracle Developer Studio 的程序员所用的 studio-runtime 和系统实用程序。示例包括 make、sccs 和汇编程序。应该在已部署 Oracle Developer Studio 12.6 的系统上安装它。

安装说明

有关所有分发的安装说明，请参见 [《Oracle Developer Studio 12.6：安装指南》](#)。

功能终止声明

[End of Features \(EOF\) Planned for Future Releases of Oracle Developer Studio](#) (Oracle Developer Studio 未来发行版计划的功能终止 (EOF)) 上列出了此发行版中已删除的功能和可能在将来发行版中删除的功能的声明。

第三方软件信息

Oracle Developer Studio 12.6 软件包括受 [《License Information User Manual Oracle Developer Studio 12.6 Last Updated: JUNE 2017》](#) 制约的第三方技术。

发送使用数据到 Oracle

“发送使用数据到 Oracle”功能会定期将您的 Oracle Developer Studio 组件使用信息发送给 Oracle Corporation。Oracle Corporation 使用此信息改进未来的 Oracle Developer Studio 软件发行版。此信息是匿名的，不会与任何个人或组织关联。

但是，如果您希望禁用“发送使用数据到 Oracle”，请将 SUNW_NO_UPDATE_NOTIFY 环境变量设置为除零 (0) 之外的任何值。

此发行版的文档

以下文档随此 Oracle Developer Studio 12.6 发行版提供。

- [《Oracle Developer Studio 12.6：安装指南》](#)。该文档包括通过 tar 文件、图形软件包安装程序和命令行软件包安装程序安装的信息。
- [《Oracle Developer Studio 12.6 发行版的新增功能》](#)。此文档介绍 Oracle Developer Studio 12.6 发行版的新增功能。
- [《Oracle Developer Studio 12.6：概述》](#)。此文档提供有关如何使用编译器和工具以及它们如何配合使用的一般说明。
- 联机帮助。联机帮助可以通过 IDE 中的 "Help"（帮助）菜单来访问，提供关于使用 IDE 中所有组件的面向任务的信息。可通过每个工具的 "Help"（帮助）菜单获得性能分析器、线程分析器、代码分析器和 dbxtool 的联机帮助。

- 这些联机参考手册页介绍了用户命令、随编译器提供的库以及其他类型的命令。手册页包含参考信息，包括命令语法、用法以及相关命令。
《[Oracle Developer Studio 12.6手册页](#)》。本文档包含 Oracle Developer Studio 12.6 发行版的所有手册页。也可通过在安装的 Oracle Developer Studio 12.6 软件中使用 man 命令访问这些文档。
- Oracle Developer Studio 12.6 手册和教程。可通过文档库访问这些文档（HTML 格式或 PDF 格式），网址为 http://docs.oracle.com/cd/E60778_01/。

此发行版中的已知问题、限制和解决方法

本节介绍此 Oracle Developer Studio 12.6 发行版的部分已知问题，并提供有关如何解决这些问题的信息。

编译器问题

本节介绍此发行版中编译器的已知问题、问题和解决方法。

编译器共有的问题

本节介绍适用于 cc、CC 和 f95 编译器的已知问题。

硬件功能问题

Oracle Developer Studio 编译器无法在基于 UltraSPARC T1 的系统（包括 SPARC T1000 和 SPARC T2000 系统）上执行。

Oracle Solaris 中的 libsunperf

在 Oracle Solaris 上，libsunperf 软件包和修补程序不为 libsunperf.so 提供新的 LAPACK 3.6.1 例程。解决方法是使用 -staticlib=sunperf 选项进行静态链接，这将与 Oracle Developer Studio 12.6 中提供的 libsunperf.a 进行链接。

使用保留处理器的 OpenMP 处理器绑定问题

OpenMP 处理器绑定不遵守在 Linux 上使用 taskset 命令的处理器保留。

x86 编译器

使用编译行中的标志 `-xarch=amdsse4a` 或 `-msse4a` 生成可执行程序，然后运行此程序，它会导致不支持 ISA sse4a 的处理器（例如，Intel 处理器）出现分段故障。

`-xtarget=S7` 和 `-xchip=S7` 选项

编译器驱动程序支持 `-xtarget=S7` 和 `-xchip=S7` 选项，但是在编译器手册页中未说明它们。

C++ 编译器问题

本节介绍此发行版中 C++ 编译器的已知问题、问题和解决方法。

编译 BOOST 库问题

不是所有 BOOST 库都可成功地进行编译。获得的结果可能与所用的 BOOST 版本以及库使用方式有关。BOOST 库包括在 Oracle Developer Studio 编译器中；正在进行测试和改进以实现编译所有 BOOST。如果运行中遇到无法继续的问题，请报告该问题。

Oracle Solaris 上的 Apache 标准库问题

Oracle Solaris 10 8/11 及更高版本中以及在 Oracle Solaris 11、11.1 和 11.2 的初始发行版中安装的 Apache stdc++ 库无法正确使用新编译器缺省设置 `-template=no%extdef`。可能需要将选项 `-template=extdef` 添加到使用此库的 CC 命令行中。在 Oracle Solaris 11.3 SRU2 中已修复此问题。当前未提供对 Oracle Solaris 10 的修复。有关更多信息，请参见《[Understanding the Effects of the Changed Default C++ Template Compilation Model \(http://www.oracle.com/technetwork/articles/servers-storage-dev/changed-default-cpp-template-model-2292727.html\)](http://www.oracle.com/technetwork/articles/servers-storage-dev/changed-default-cpp-template-model-2292727.html)》（《了解更改后的缺省 C++ 模板编译模式的效果》）。

`-xustr=ascii_utf16_ushort` 选项与 `-std=c++11` 不兼容

如果 `-std=c++11` 或 `-std=c++14` 选项已生效，则指定标志 `-xustr=ascii_utf16_ushort` 会导致错误。

在这些模式下编译时，请使用 C++11 Unicode 字符和字符串选项。

名称改编链接问题

Oracle Developer Studio 在 `-compat=5` 模式下执行的名称改编具有错误。修复这些错误时会产生二进制文件不兼容性问题。在使用 `-m64` 的 x86/Oracle Solaris 上和 Linux 上，已在发行 Oracle Studio 之前发现并更正了错误，但在其他平台上仍存在。新选项 `-abiopt=mangle6` 更正了使用 `-m32` 选项的 SPARC/Oracle Solaris 和 x86 Oracle Solaris 上的改编错误。

请参阅《Oracle Developer Studio 12.6: C++ 用户指南》中的“`-abiopt=[mangle5|mangle6]`”。

不支持引用模板中的非全局名称空间目标

如果您使用 `-instances=extern` 编译，则使用模板和静态对象的程序会出现未定义符号的链接时错误。使用缺省设置 `-instances=global` 则不会出现问题。编译器不支持对模板中的非全局名称空间作用域目标的引用。请看以下示例：

```
static int k;
template<class T> class C {
    T foo(T t) { ... k ... }
};
```

在本示例中，一个模板类的成员引用了静态名称空间作用域的变量。请记住，名称空间作用域包含文件作用域。编译器不支持模板类的成员引用静态名称空间作用域的变量。另外，如果模板在其他的编译单元实例化，那么每个实例都会指向不同的 `k`，这破坏了 C++ 一次定义规则，代码的行为将会不可预测。

下面的方法也是可行的，但这取决于您如何使用 `k`，以及它应有的功能。第二个选项仅可供属于类成员的函数模板使用。

- 可以为变量提供外部链接属性：

```
int k; // not static
```

所有的实例都使用同一个 `k`。

- 也可以使这个变量成为类的静态成员：

```
template<class T> class C {
    static int k;
    T foo(T t) { ... k ... }
};
```

静态类成员具有外部链接属性。每个 `C<T>::foo` 的实例使用不同的 `k`。其他函数可以共享 `C<T>::k` 的实例。此选项可能是您需要的选项。

名称空间内的 `#pragma align` 需要改编名称

在名称空间内使用 `#pragma align` 时，必须使用改编名称。例如，在下面的代码中，`#pragma align` 语句是无效的。要更正此问题，应将 `#pragma align` 语句中的 `a`、`b` 和 `c` 替换为其改编名称。

编译为 C++11 时，可以使用标准关键字 `alignas` 以避免 `pragma` 出现问题。

```
namespace foo {
    #pragma align 8 (a, b, c) // has no effect
    //use mangled names: #pragma align 8 (__1cDfooBa_, __1cDfooBb_, __1cDfooBc_)
    static char a;
    static char b;
    static char c;
}
```

Fortran 编译器问题

本节介绍此发行版中 Fortran 编译器的已知问题和问题。

- 不换行打印行末尾之前的空格不会影响输出位置。
输出语句格式末尾的 `X` 编辑描述符不会影响输出记录中后续字符的位置。如果输出语句中有 `ADVANCE='NO'` 并且有更多字符要通过后续输出语句传送到同一记录，则会导致差异。
在很多情况下，可以通过添加空白字符串编辑描述符，而不是 `nX` 编辑描述符来解决此问题。它们并不完全相同，因为空白字符串编辑描述符实际上会将空白字符包含在记录中，而 `nX` 仅跳过后 `n` 个字符，缺省情况下通常导致跳过的位中产生空白。
- 一行中有两个连续 `&` 号的有效代码会被拒绝。
Fortran 标准禁止有一个 `&` 号的空续行。但是，仍可以创建同一行中有两个 `&` 号的空续行，这不在标准限制范围之内。编译器不会处理这种情况，而会显示一条错误消息。解决方法是删除该行，这只会影响程序可读性而并不添加任何语义。BOZ 常量有时会被截断。
- 在某些相对更复杂的情况下，如数组构造，BOZ 常量可能会被截成 4 个字节的缺省整数大小，即使它所应指定给的相应项是一个 8 个字节的整数实体。解决方法是在数组构造中使用正确类型和种类的常量，而不要使用 BOZ 常量。
- 更正新发行版中的舍入算法可能会导致列表控制的、名称列表控制的和格式化写入的 `ROUND='NEAREST'` 和 `ROUCH='COMPATIBLE'` 的输出与以前发行版中的输出不同。该舍入差异应该仅在最低有效位上。

Oracle Developer Studio 12.6 缺少 Oracle Solaris libfsu 修补程序

Oracle Solaris 上的 libfsu 版本可能没有使“参数化派生类型”功能正常运行所需的最新修补程序。因此，使用此功能的程序无法在 Oracle Solaris 上正常运行，即使它们在 Linux 上可以运行。

为了在 Oracle Solaris 上使用此功能，可以使用 `-Bstatic` 选项与 `libfsu` 归档版本链接。

libsunmath 库在 Linux 上不受支持

`libsunmath` 库在 Linux 上不受支持。因此，在 Linux 平台上进行编译时，某些非内部数学函数无法解析。

`libsunmath` 中 Fortran 可调用的函数如下：

```
abrupt_underflow, convert_external, d_acos, d_acosd, d_acosh,
d_acosp, d_acospi, d_addran, d_addrans, d_aint, d_anint, d_annuity,
d_asin, d_asind, d_asinh, d_asinp, d_asinpi, d_atan2, d_atan2d,
d_atan2pi, d_atan, d_atand, d_atanh, d_atanp, d_atanpi, d_cbrt,
d_ceil, d_compound, d_copysign, d_cos, d_cosd, d_cosh, d_cosp,
d_cospi, d_erf, d_erfc, d_exp10, d_exp2, d_exp, d_expm1, d_fabs,
d_floor, d_fmod, d_get_addrans, d_hypot, d_infinity, d_init_addrans,
d_j0, d_j1, d_jn, d_lcran, d_lcrans, d_lgamma, d_lgamma_r, d_log10,
d_log1p, d_log2, d_log, d_logb, d_max_normal, d_max_subnormal,
d_min_normal, d_min_subnormal, d_mwcran, d_mwcrans, d_nextafter, d_pow,
d_quiet_nan, d_remainder, d_rint, d_scalb, d_scalbn, d_set_addrans,
d_shufrans, d_signaling_nan, d_significand, d_sin, d_sincos,
d_sincosd, d_sincosp, d_sincospi, d_sind, d_sinh, d_sinp, d_sinpi,
d_sqrt, d_tan, d_tand, d_tanh, d_tanp, d_tanpi, d_y0, d_y1, d_yn,
gradual_underflow, i_addran, i_addrans, i_get_addrans, i_get_lcrans,
i_get_mwcrans, i_init_addrans, i_init_lcrans, i_init_mwcrans,
i_lcran, i_lcrans, i_llmwcran, i_llmwcrans, i_lmwcran, i_lmwcrans,
i_mwcran, i_mwcrans, i_set_addrans, i_set_lcrans, i_set_mwcrans,
i_shufrans, id_finite, id_fp_class, id_ilogb, id_rint, id_isinf,
id_isnan, id_isnormal, id_issubnormal, id_iszero, id_nint, id_signbit,
ieee_flags, ieee_handler, ieee_retrospective, iq_finite, iq_fp_class,
iq_ilogb, iq_isinf, iq_isnan, iq_isnormal, iq_issubnormal, iq_iszero,
iq_signbit, ir_finite, ir_fp_class, ir_ilogb, ir_rint, ir_isinf,
ir_isnan, ir_isnormal, ir_issubnormal, ir_iszero, ir_nint, ir_signbit,
nonstandard_arithmetic, q_copysign, q_fabs, q_fmod, q_infinity,
q_max_normal, q_max_subnormal, q_min_normal, q_min_subnormal,
q_nextafter, q_quiet_nan, q_remainder, q_scalbn, q_signaling_nan,
r_acos, r_acosd, r_acosh, r_acosp, r_acospi, r_addran, r_addrans, r_aint,
r_anint, r_annuity, r_asin, r_asind, r_asinh, r_asinp, r_asinpi, r_atan2,
r_atan2d, r_atan2pi, r_atan, r_atand, r_atanh, r_atanp, r_atanpi, r_cbrt,
r_ceil, r_compound, r_copysign, r_cos, r_cosd, r_cosh, r_cosp, r_cospi,
r_erf, r_erfc, r_exp10, r_exp2, r_exp, r_expm1, r_fabs, r_floor, r_fmod,
r_get_addrans, r_hypot, r_infinity, r_init_addrans, r_j0, r_j1, r_jn,
r_lcran, r_lcrans, r_lgamma, r_lgamma_r, r_log10, r_log1p, r_log2, r_log,
r_logb, r_max_normal, r_max_subnormal, r_min_normal, r_min_subnormal,
```

`r_mwcran`、`r_mwcrans`、`r_nextafter`、`r_pow`、`r_quiet_nan`、`r_remainder`、
`r_rint`、`r_scalb`、`r_scalbn`、`r_set_addrans`、`r_shufrans`、`r_signaling_nan`、
`r_significand`、`r_sin`、`r_sincos`、`r_sincosd`、`r_sincosp`、`r_sincospi`、
`r_sind`、`r_sinh`、`r_sinp`、`r_sinpi`、`r_sqrt`、`r_tan`、`r_tand`、`r_tanh`、`r_tanp`、
`r_tanpi`、`r_y0`、`r_y1`、`r_yn`、`sigfpe`、`smwcran`、`standard_arithmetic`、
`u_addrans`、`u_lcrans`、`u_llmwcran`、`u_llmwcrans`、`u_lmwcran`、`u_lmwcrans`、
`u_mwcran`、`u_mwcrans`、`u_shufrans`

Fortran 填充值更改

与以前的发行版相比，使用 `-pad=common` 编译器选项生成的填充值在使用 Oracle Developer Studio 12.6 编译的代码中可能是不同的。如果使用 `-pad=common` 编译引用同一通用块的两个或更多文件，则必须使用同一 Oracle Developer Studio 发行版中的 `f95` 编译器编译它们。

Fortran 77 库已删除

需注意 Oracle Solaris Studio 12.2 发行版已删除了过时的 FORTRAN 77 库。这意味着使用依赖于共享库 `libF77`、`libM77` 和 `libFposix` 的传统 Sun WorkShop f77 编译器编译的旧的可执行文件将不会运行。

数组内部函数使用全局寄存器

数组内部函数 `OT_PRODUCT` 和 `MATMUL` 针对相应 SPARC 平台体系结构进行了高度优化。因此，它们使用全局寄存器 `%g2`、`%g3` 和 `%g4` 作为临时寄存器。

对于区间运算，还会影响数组内部函数 `ANY`、`ALL`、`COUNT`、`MAXVAL`、`MINVAL`、`SUM`、`PRODUCT`。

如果调用了上述所列的数组内部函数，则用户代码不应该认为这些寄存器可用于暂时存储。当调用数组内部函数时，这些寄存器中的数据将被覆盖。

归档库中的 F95 模块不包括在可执行文件中

调试器 `dbx` 要求编译中使用的所有对象文件都包含在可执行文件中。通常，无需用户执行额外操作，程序即可满足此要求。但使用含有模块的归档文件时例外。如果程序使用了一个模块，但没有引用模块中的任何过程或变量，则产生的对象文件不会包含对模块中定义的符号的引用。只有对目标文件中定义的符号具有引用时，链接器才会链接归档文件中的对象文件。如果没有此类引用，则可执行文件中将不包括对象文件。`dbx` 将在

尝试查找与所用模块关联的调试信息时发出警告。对于缺少调试信息的符号，则无法提供有关这些符号的信息。

使用 `-u` 链接程序选项可以解决这个问题。此选项使用一个符号作为其选项参数。它会将该符号添加到未定义的链接程序符号集中，这就需要解析此符号。与模块关联的链接程序符号通常是模块名称，其所有字母均为小写，后面跟有一条下划线。

例如，为了强制包含模块 `MODULE_1` 的对象文件被归档文件采用，请指定链接程序选项 `-u module_1_`。如果使用 `f95` 命令进行链接，请在命令行上使用 `-Qoption ld -umodule_1_`。

Linux 平台上的 gethrtime(3F)

启用系统节能功能时，没有可靠方式可以精确地获得 AMD 处理器上的时钟速率。因此，使用基于 `gethrtime(3F)` 的计时函数（Fortran 编译器的 Linux 版 Oracle Solaris `gethrtime(3C)` 函数）在 Linux 平台上获得高精度实际时间的方法只有在禁用了节能功能的 AMD 系统上才会精确。要禁用节能功能，可能需要重新引导系统。

工具问题

本节列出了调试工具和性能分析工具的已知限制。

dbx 限制和不兼容情况

dbx 具有以下限制：

- 使用 `libc.so.5` 或者 `libc.so.4` 的旧副本可能会在 C++ 异常区域中引起 dbx 问题。可能会出现关于错误的 `stab` 和未处理的异常等警告消息。
解决方法：在所有系统上安装最新的 `libc.so.5`。
- Fortran 用户应该用 `-stackvar` 选项进行编译，以便充分利用运行时检查。
- 某些程序可能无法正常使用 `-stackvar` 选项。在这样的情况下，请尝试使用 `-c` 编译器选项，它将在不使用运行时检查的情况下启用数组下标检查。
- dbx 命令行解释器是旧版本的 Korn shell (`ksh`)，不支持代码集独立 (Code Set Independence, CSI)。当在 dbx 命令行上键入多字节字符时，会发生解释错误。
- Linux OS 上不能使用以下 dbx 功能：
 - 修复并继续
 - 多线程应用程序上的性能数据收集。
 - 在下列事件中设置断点：
 - `fault`

- lastrites
- lwp_exit
- sysin
- sysout
- sync
- 索引 DWARF (编译器选项 -xs=no)
- 在 Linux 平台上调试程序时可能会发生下面的问题：
 - 不支持 TLS 访问。
 - 要调试 32 位程序，必须使用 -x exec32 选项启动 dbx。
 - Korn shell 中的管道操作符仅限于 Linux 平台。任何需要访问目标进程的 dbx 命令不能作为管道的一部分使用。例如，下面的命令可能会导致 dbx 挂起：

```
where | head -1
```

解决方法：

- 按 Ctrl-C 组合键以显示新的 dbx 提示符。
- dbx 可缓存大量信息，对于上述示例，下列命令序列会起作用：

```
where
where | head -1
```

- 将命令输出重定向到文件，然后显示文件内容。

```
(dbx) > bag where
(dbx) cat bag
```

- dbx 不支持 GNU C 和 C++ 编译器的下列功能：
 - VL 数组 (gcc 4.1 及之前)
 - OpenMP
 - RTTI
 - 模板定义
 - 缺省参数
 - using 指令
 - friend
- Oracle Linux 6 上的 dbx 存在以下问题：
 - 系统库中使用的间接引用符号有时会导致 dbx 在引用处不是实际函数处设置断点。
 - 在调试使用 gcc 4.4.4 编译器编译的代码时，请注意 dbx 看不到使用 -D 调试器选项定义的宏。
- Oracle Developer Studio 支持调试信息的 stabs 格式，但 Oracle 已宣布可能最终将停用该格式以支持 DWARF 格式。Oracle 不需要为新增功能或现有功能的增强实施 stabs 支持。

Stabs 格式不支持以下调试功能。

- C++11 和 C++14 新增功能。
- C11 新增功能
- 优化后代码中的参数和局部变量。
- 宏（当使用 -g3 选项编译代码时）。
- 调试信息的子集（通常在使用 -xdebuginfo={...} 编译时）。
- 面向对象的 Fortran
- 有关将 dbx 连接到进程时的数据收集问题的信息，请参见[“dbx attach 分析 \(collect -P\)” \[23\]](#)。
- 如果使用 GNU 运行时库，则对异常处理的调试支持是有限的。
- 不支持对 discover 检测过的二进制文件使用以下功能：
 - 修复并继续
 - 运行时检查定制变量 (dbxenv)
- 如果是在 HTML 模式下检测了调试对象，discover 分析将不会显示在 dbx 控制台中。解决方法是在检测期间使用文本模式选项 -w。

性能分析器和 er_print 实用程序限制

本节介绍性能分析器工具和 er_print 实用程序的已知问题。

- 在 Linux 上，使用 collect -P 命令连接到正在运行的多线程进程（包括 Java），将不会正确收集数据。
- 性能分析器对 Java 7U65 崩溃。在使用性能分析器工具时，请至少使用 Java 7U75。
- 共享对象的 "Library Visibility"（库可见性）功能有时无法与过滤正常使用。
- 性能分析器和 er_print 找不到 jar 文件中嵌入的共享对象。此外，在有些情况下，性能分析器和 er_print 找不到 Java 类文件或源文件。要解决这些问题，请使用 addpath 指向包含相应文件的目录。有关更详细的信息，请参见性能分析器 "Help"（帮助）中的 "Troubleshooting"（故障排除）部分。
- 在比较实验时，可能会看到以下问题：
 - "Source"（源）和 "Disassembly"（反汇编）视图显示包含度量和独占度量，而不是如预期地仅显示包含度量。
- 可能会看到与具有可变时钟速率的 CPU 相关的以下问题：
 - 在采用可变时钟速率的系统上，基于周期的计数器的硬件计数器分析度量在处理器以低于最大速率运行时低报，并且度量会转换为时间。
 - 在采用了以不同时钟频率运行的多个 CPU 的系统上，将根据其中一个 CPU 的时钟速率处理实验，这可能导致对其他 CPU 的计数过多或不足。
- 分析在 SPARC T5、M5 和 M6 系统上使用许多线程的 OpenMP 应用程序时，可能生成非常大而无法读取的实验。解决方法是在运行 collect 之前设置环境变量 SP_COLLECTOR_NO_OMP。请注意，在设置此变量时，无法在结果实验中看到

OpenMP 构造和度量, "User" (用户)、"Expert" (专家) 和 "Machine" (计算机) 视图全都看起来一个样。

在远程主机上分析应用程序的限制

尚不支持使用口令短语远程登录。

collect 实用程序

本节介绍性能分析器工具的 collect 实用程序和数据收集的已知问题。

- 通过 JDK 1.7.0_59 使用 JDK 1.7.0_40 时, 由于 JDK 中存在错误, 对混合了 Java 和 C++ 的应用程序的分析可能不完整。从 Java 到 C++ 的调用的堆栈未正确展开。
- Linux 上的堆跟踪不跟踪对 calloc 的调用。
- 收集计数数据 (collect -c) 对使用 -std=c++11 编译的二进制文件无效。
- 由于 Oracle Solaris setitimer (ITIMER_REALPROF, ...) 中的错误所致, SPARC T4/T5 M5/M6 上的时钟分析可能具有不规则的样例。
- 属性系统等于 1 的硬件计数器分析具有以下已知问题。有些系统时间可能未对 usleep() 正确计数。在 Oracle Solaris 10 上, 使用系统属性可挂起应用程序。
- 由于 Oracle Solaris 中的错误所致, 时钟分析可能将 "User CPU Time" (用户 CPU 时间) 错误地报告为 "Wait CPU Time" (等待 CPU 时间)
- 通过内核区域中的分析信号, 您可能会看到以下问题:
 - 内核区域中的时钟分析信号无规律地传送, 从而导致数据丢失。
 - 硬件计数器 dtlb_misses.any 对 Westmere 内核区域不起作用
 - 内核区域不支持 x86 上的内存空间分析
 - 在 SPARC 上, 硬件计数器分析样例可能延迟许多秒; 丢失的样例将导致少计
 - 内核区域中的时钟分析样例可能报告错误的微观状态
 - 运行内核区域的系统控制域上的 HWC 分析样例可能丢失。
 - 内核区域上的 HWC 分析样例可能丢失。
- 在 Linux 上分析 SPECJBB2015 会导致崩溃或挂起。
- 堆栈展开的各种极端情况可能会在 x86 以及 Oracle Solaris 和 Linux 上失败

er_kernel 实用程序

本节介绍 er_kernel 实用程序的已知问题。

- er_kernel 拒绝在运行于 Oracle VM 下的 Oracle Solaris 系统上运行, 以避免会导致重新引导的 Oracle VM 错误

- 调用堆栈有时可能会省略某个帧，通常出现在叶函数位于其尾声中时
- 运行内核区域的系统控制域上的硬件计数器分析样例可能丢失。
- 内核区域上的硬件计数器分析样例可能丢失。
- 内核区域中使用 `er_kernel` 的硬件计数器分析可能仅支持单个硬件计数器

dbx collector 分析

本节介绍使用 `dbx collector` 命令时的应用程序分析问题。

- 在 Java 上使用 `dbx collector` 有限制。无法为目标指定 Java 类文件。因此，必须指定 JVM 作为目标，并指定类文件作为 `dbx run` 命令的一个参数。例如：

```
> dbx /path-to-your-jdk/bin/java
(dbx) collector enable
(dbx) collector java on
(dbx) run [JVM-options] [Java-class-file-to-execute]
```

- 在 Linux 上，使用 `dbx` 或 `collect -P` 分析任何 Java 应用程序都可能失败。

dbx attach 分析 (collect -P)

本节介绍使用 `dbx attach` 命令时的运行中应用程序分析问题：

- 在 Linux 上，使用 `dbx` 或 `collect -P` 连接到 Java 程序可能会失败。
- 在 Linux 上，连接到正在进行阻塞调用或非阻塞调用的目标可能导致目标失败。
- 在 Linux 上，连接到多线程目标将不会正确记录在连接时已创建的线程的数据。不提供有关丢失数据的警告。请注意，其中包括所有 Java 目标，因为 JVM 是多线程的。
- 如果将 `dbx` 连接到正在运行且启动时未使用 `LD_PRELOAD` 预先装入收集器库 `libcollector.so` 的进程，则将发生一系列错误。无法收集任何跟踪数据：同步等待跟踪、堆跟踪、I/O 跟踪或 MPI 跟踪。跟踪数据是通过对各库执行插入操作而收集的。如果没有预先装入 `libcollector.so`，将无法执行插入操作。
- 如果在 `dbx` 连接到进程之前目标程序安装了信号处理程序，并且信号处理程序未在 Oracle Solaris 上传递 `SIGPROF` 和 `SIGEMT` 信号（或者未在 Linux 上传递 `SIGIO` 信号），则可能会丢失部分或全部分析数据。
- 如果目标程序使用 `libcpc.so`，则硬件计数器实验可能会失败，这是因为收集器和程序都在使用该库。
- 如果目标程序调用 `setitimer(2)`，则由于收集器和程序同时使用计时器，可能会使时钟分析实验中断。
- 连接到正在调用 `malloc` 库的目标可能导致目标失败。在连接时请求硬件计数器会极大地提高此类失败的概率。

IDE 限制

本节介绍 IDE 中的已知限制。

要在远程主机上根据二进制文件创建项目，远程主机必须提供 Oracle Solaris Studio 12.3 或 12.4 或者 Oracle Developer Studio 12.5。较早的发行版不支持此功能。

代码分析工具限制

本节介绍代码分析工具的已知限制。

- 如果未正确检测您的代码，则可能会发现误报或漏报。有关更多信息，请参见《[Oracle Developer Studio 12.6: Discover 和 Uncover 用户指南](#)》中的“Supported Binaries”。
- 当程序大小（文本 + 数据）大到超过 `-xmodel=small` 的最大范围 ($2^{32} - 2^{24} - 1$) 时，discover 和 uncover 工具无法处理使用 `-xmodel=medium` 编译的 x86_64 二进制文件。
- discover 工具对辅助文件无效；在 Oracle Solaris 11.3 中，辅助文件包含使用链接程序选项 `-z ancillary` 创建的调试信息。
- 要在 Oracle Linux 上使用任何代码分析工具，都必须编译原始二进制文件并使用 `-xannotate=[yes]` 选项进行链接。
- 要在支持 SSM 的计算机的内核区域中使用 "Discover SSM"，则区域的配置应该包括 `host-compatible=native`

安装和卸载问题

本节介绍 Java 安装程序的已知问题、问题和解决方法。

使用 Java 安装程序安装 Oracle Instant Client

命令行、非交互模式 `./developerstudio.sh --non-interactive` 将不安装 OIC 组件。

要避免此问题，用于安装的命令行、非交互模式必须在 `--install-components` 选项之后一起指定 OIC 以及所有编译器和工具组件，如下所示：

```
./developerstudio.sh --non-interactive --install-components oic, dbxtool, code-analyzer-tool, performance-and-thread-analysis-tools, performance-library, c-and-cpp-compilers, dmake
```


此外，如果需要日语或中文语言环境，请将组件（用逗号分隔）`japanese-localization` 和 `simplified-chinese-localization` 添加到 `--install-components` 行。

设置了日语语言环境时，卸载程序无法运行

如果您环境中的语言环境设置为日语，卸载程序无法正确显示日语字符。解决方法是取消设置日语语言环境，并将您的 `LC_ALL` 环境设置为 `C`。

```
$ set LC_ALL=C
```

如果您的语言环境适用，请使用 `--locale` 安装程序标志。

```
$ LC_ALL=C ./developerstudio.sh --locale=ja
```

