

# Oracle® Developer Studio 12.5: Release Notes

ORACLE®

Part No: E60741  
May 2017



**Part No: E60741**

Copyright © 2016, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

**Access to Oracle Support**

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

**Référence: E60741**

Copyright © 2016, 2017, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

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.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou ce matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

**Accès aux services de support Oracle**

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

# Contents

---

<b>Using This Documentation</b> .....	7
<b>Oracle Developer Studio 12.5 Release Notes</b> .....	9
System Requirements .....	9
Required System Software Packages .....	10
Installation Instructions .....	12
Features That Have Been Removed in This Release .....	12
lock_lint Utility .....	12
OpenMP Related Environment Variables .....	12
Compiler Options for Legacy SPARC-Based Systems .....	13
F95 Interfaces with Optional Arguments in Performance Library .....	13
Legacy Fortran F77 Object Files .....	14
VMS Compatible Library libV77 .....	14
Legacy Array Intrinsic Runtime Libraries for Fortran .....	14
collect -R Option .....	14
Values amd64, generic64 Options .....	14
Automatic Tuning System Support .....	15
Features That Might Be Removed in a Future Release .....	15
C Compiler Option -Xs .....	15
C Compiler Option -Xt .....	15
Classic C++ iostreams .....	15
Compiler Option -xdebugformat=stabs .....	15
Compiler Options for Legacy AMD-Based Systems .....	16
Runtime Checking in dbx .....	16
Global Program Checking Support in Fortran .....	16
Lint -Nlevel Option .....	16
-xanalyze=code Option .....	16
fpversion Command .....	17

Static Version of the libsunmath Library .....	17
-xatomic Option .....	17
libgc Library .....	17
Fortran Interfaces to C Math Library Functions .....	17
Interval Arithmetic Support .....	18
spot Commands .....	18
bw and traps Commands .....	18
Performance Analyzer Support for MPI .....	18
-xlic_lib=sunperf Option .....	19
Third-Party Software Information .....	19
Usage Data to Oracle .....	19
Documentation For This Release .....	19
Known Problems, Limitations, and Workarounds in This Release .....	20
Compiler Issues .....	20
Tools Issues .....	27
Java Installer Issues .....	33

## Using This Documentation

---

- **Overview** – Provides information about known problems and other issues with this Oracle Developer Studio 12.5 release.
- **Audience** – Application developers, system developers, architects, support engineers.
- **Required knowledge** – Programming experience, software development testing, aptitude to build and compile software products.

## Product Documentation Library

Documentation and resources for this product and related products are available at [http://docs.oracle.com/cd/E60778\\_01](http://docs.oracle.com/cd/E60778_01).

## Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.





# Oracle Developer Studio 12.5 Release Notes

---

This document provides information that you need to know before installing and using the product. See [Oracle Developer Studio 12.5: Installation Guide](#) for installation instructions.

## System Requirements

Oracle Developer Studio 12.5 software can be installed on Oracle Solaris 10 or Oracle Solaris 11.3 operating system on SPARC-based or x86-based platforms, or on the Oracle Linux operating system.

**TABLE 1** System Requirements

System Resources	Requirements
<b>Platform</b>	Only 64-bit Oracle Solaris and Linux kernels supported
<b>Operating System</b>	Oracle Solaris 11.3 (Oracle Solaris 11.3 SRU09 or later is required)  Oracle Solaris 10 1/13  Oracle Linux 6, Oracle Linux 7  Red Hat Linux 6, Red Hat Linux 7  See “ <a href="#">Required System Software Packages</a> ” on page 10 to ensure your system has the required packages to execute the Oracle Developer Studio 12.5 software.  Refer to the <a href="#">Oracle Developer Studio 12.5: Installation Guide</a> for instructions to install the <a href="#">Oracle Developer Studio 12.5 distributions</a> .
<b>VM Support</b>	Oracle Solaris 11.3 SRU09 (or later) running as a zone, LDOM or Kernel zone.  Oracle Solaris 10 1/13 as a zone or LDOM
<b>Supported Client Systems</b>  For installing the distributions of the IDE or Performance Analyzer for remote use	Client systems running IDE, Performance Analyzer, Code Analyzer, and dbxtool must have the minimum Java version installed

System Resources	Requirements
Java Version	Java 7 or Java 8 JDK 64-bit version - minimum version determined by security updates

**Note** - The sunpro-incorporation package with the necessary updates to support the Oracle Developer Studio 12.5 release is provided in the Oracle Solaris 11.3 Release repository available to all customers. See the [Oracle Developer Studio 12.5: Installation Guide](#) for instructions on updating the sunpro-incorporation package on your Oracle Solaris 11.3 system.

Refer to the Oracle Developer Studio 12.5 Release Notes: “[Known Problems, Limitations, and Workarounds in This Release](#)” on page 20 section for other Oracle Solaris issues that might be resolved in Oracle Solaris SRUs.

## Required System Software Packages

Before you execute the Oracle Developer Studio 12.5 compilers and tools, verify that the system on which you are running the software has the required system packages. The following table lists the required software packages for each supported operating system.

**TABLE 2** System Software Required by Developer

Operating System Version	System Software Packages
Oracle Solaris 11.3	<p>Please consult the <a href="#">Oracle Developer Studio 12.5: Installation Guide</a> to insure that the your system has the appropriate version of the sunpro-incorporation, developer-studio-utilities, and developer-studio-utilities package</p> <p>Check the following packages are installed:</p> <ul style="list-style-type: none"> <li>pkg:/developer/base-developer-utilities</li> <li>pkg:/developer/library/xprofile</li> <li>pkg:/diagnostic/cpu-counters</li> <li>pkg:/library/glib2</li> <li>pkg:/library/libxml2</li> <li>pkg://library/zlib</li> <li>pkg:/runtime/python-27</li> <li>pkg:/system/library/c++-runtime</li> <li>pkg:/system/library/libv12n</li> <li>pkg:/system/library/mmheap</li> <li>pkg:/system/library/openmp</li> <li>pkg:/system/linker</li> <li>pkg:/developer/library/lint</li> <li>pkg:/runtime/perl-512</li> <li>pkg:/shell/bash</li> <li>pkg:/shell/ksh93</li> <li>pkg:/system/header</li> </ul>

Operating System Version	System Software Packages
	pkg:/system/library/math pkg:/system/resource-mgmt/resource-pools pkg:/library/c++/stdcxx
Oracle Solaris 10 1/13	SUNWhea SUNWarc SUNWcpp SUNWlibstdcxx4 SUNWpiclh SUNWpiclr SUNWPython-devel SUNWcar SUNWpcpu SUNWcsl SUNWcslr SUNWdpl SUNWgnome-base-libs SUNWlibC SUNWlibm SUNWlibms SUNWlibmsr SUNWxml SUNWpython SUNWsprt SUNWzlib
Oracle Linux 6 Oracle Linux 7	Development/Libraries Package Group and 32-bit runtime support libraries including: 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

---

**Note** - `developer-studio-utilities` and `studio-runtime` are “group” packages, which are an IPS shorthand to simplify installation of groups of related packages. They consist entirely of dependences on other packages but have no content of their own.

`studio-runtime` is a group package that installs system runtime support libraries used by programs compiled using Oracle Developer Studio compilers. Examples include the Fortran and Sun Performance libraries. It should be installed on systems where applications are deployed.

`developer-studio-utilities` is a group package that installs `studio-runtime` and system utility programs used by programmers using Oracle Developer Studio. Examples include `make`, `sccs`, and assemblers. It should be installed on systems where Oracle Developer Studio 12.5 is deployed.

---

## Installation Instructions

See the [Oracle Developer Studio 12.5: Installation Guide](#) for installation instructions for all distributions.

## Features That Have Been Removed in This Release

The following features have been removed in the Oracle Developer Studio 12.5 software.

### `lock_lint` Utility

The `lock_lint` utility to analyze locks in multithreaded programs is obsolete and has been removed. You should use Thread Analyzer.

### OpenMP Related Environment Variables

The following environment variables have been deprecated:

- `STACKSIZE` - Use `OMP_STACKSIZE` instead
- `PARALLEL` - Use `OMP_NUM_THREADS` instead
- `SUNW_MP_PROCBIND` - Use `OMP_PROC_BIND` instead

- `SUNW_MP_MAX_NESTED_LEVELS` - Use `OMP_MAX_ACTIVE_LEVELS` instead

For more information about the above environment variables, see [Oracle Developer Studio 12.5: OpenMP API User's Guide](#).

## Compiler Options for Legacy SPARC-Based Systems

Support for legacy systems has been removed in this release. The systems include those based on the UltraSPARC I, II, IIe, III, IIIi, III+, IV, and IV+ processors.

Accordingly, the following options for the `cc`, `CC`, and `f95` compiler commands are removed:

### -xchip Option

The following `-xchip` option specifications are removed from this release:

```
-xchip={ultra,ultra2,ultra2i,ultra2e,ultra3,
        ultra3i,ultra3cu,ultra4,ultra4plus}
```

```
-xchip=pentium
```

### -xtarget Option

The following `-xtarget` option specifications are removed from this release:

```
-xtarget={ultra,ultra1/140,ultra1/170,ultra1/200,
          ultra2,ultra2/1170,ultra2/1200,ultra2/1300,ultra2/2170,
          ultra2/2200,ultra2/2300,ultra2e,ultra2i,ultra3,ultra3cu,
          ultra3i,ultra4,ultra4plus}
```

## F95 Interfaces with Optional Arguments in Performance Library

The optional argument feature of the `libsunperf` F95 interfaces has been removed. F95 interfaces without optional arguments continue to be supported.

## Legacy Fortran F77 Object Files

Support for object files created with the legacy F77 compiler and the `-lf77compat` linking option is discontinued in this release.

## VMS Compatible Library `libv77`

The `libv77` library formerly contained two routines that conflict with the Fortran 90 intrinsic routines and do not conform to the Y2K requirement for 4-digit year. They were provided as optional routines for VMS compatibility only and have been removed in this release.

## Legacy Array Intrinsic Runtime Libraries for Fortran

The libraries `libfmaxlai`, `libfmaxvai`, `libfminlai`, `libfminvai`, `libfprodai` and `libfsumai` on SPARC platform have not been used by the Studio Fortran compilers since the Sun Studio 10 release in 2005.

These libraries are removed from linking in this release. Object files produced by Studio compilers earlier than the Sun Studio 10 release can no longer be used and must be recompiled with newer Studio compilers. Older executables compiled with Oracle Solaris Studio 12.3 and earlier and have dependencies on these runtime libraries will continue to work. If you have old object files that require any of these libraries and recompiling is not feasible, you should keep the old compiler installation or copy the specific libraries you need from the old compiler installation into the new compiler installation.

## `collect -R` Option

The `collect -R` option was formerly used to display a README file for the new features of the Performance Analyzer tools and is now removed.

## Values `amd64`, `generic64` Options

The x86 compiler and assembler(`fbe`) options `-xarch=amd64` and `-xarch=generic64` are deprecated. These options are replaced with the option `-m64`.

## Automatic Tuning System Support

Automatic Tuning System (ATS) facility has been removed.

## Features That Might Be Removed in a Future Release

The following features might be removed in a future release.

### C Compiler Option -Xs

The -Xs option might be removed in a future release of Oracle Developer Studio or Oracle Solaris. If your C code requires -Xs to build and compile correctly, you should migrate the code to conform with at least the C99 dialect of the ISO C standard. In other words, you should be able to compile with the -std=c99 option.

### C Compiler Option -Xt

The -Xt option might be removed in a future release of Oracle Developer Studio or Oracle Solaris. If your C code requires -Xt to build and compile correctly, you should migrate the code to conform with at least the C99 dialect of the ISO C standard. In other words, you should be able to compile with the -std=c99 option.

### Classic C++ iostreams

Classic iostreams (`libiostream`) is the original 1986 version of iostreams, which was replaced in the 1998 C++ standard. The library is specified with the `-library=iostream` option. Classic iostreams was not a standard, so no two implementations of this library are the same and code that uses it is not portable and is not compatible with the C++ standard library. Use the iostream functions provided by the C++ Standard library.

### Compiler Option -xdebugformat=stabs

The `-xdebugformat=stabs` for all compilers might be removed in a future release. The only debugger format option will be `-xdebugformat=dwarf`, which is currently the default.

## Compiler Options for Legacy AMD-Based Systems

Legacy Oracle/SUN systems soon to be EOL'd include AMD based systems. Accordingly, the following options for the cc, CC, and f95 compiler commands might be removed in the future Oracle Developer Studio releases:

- -xtarget={opteron|barcelona}
- -xarch={pentium\_proa|sse2a|sse3a|amdsse4a}
- -xchip={opteron|amdfam10}
- -march={sse4a|3dnow}
- -msse4a, -m3dnow

## Runtime Checking in dbx

The feature Runtime Checking (RTC) in dbx might be removed. You can use the `discover` tool to do memory checking during runtime.

## Global Program Checking Support in Fortran

All `-Xlist` options used for global program checking and program listing will be removed in future release.

## Lint `-Nlevel` Option

The `-Nlevel` option is obsolete and might be removed in a future release.

## `-xanalyze=code` Option

The `-xanalyze=code` is synonymous with the compiler option `-xprewise`. `-xanalyze=code` is deprecated and might be removed in a future release.



## fpversion Command

The `-kstat` option supersedes the `fpversion` command and might be removed in a future release.

## Static Version of the `libsunmath` Library

`libsunmath.a` might be removed in a future release. The `libsunmath.so` library will continue to be available. It is recommended to replace any use of the static library with the dynamic library, i.e. replace `-Bstatic -lsunmath -Bdynamic` with `-lsunmath`.

## `-xatomic` Option

When the Oracle Solaris operating system includes a compatible standard interface and the Linux bundled library conforms to a standard interface, the need for the `-xatomic` option will be redundant. Therefore, the `-xatomic` option might not be available in future releases of Oracle Developer Studio.

## `libgc` Library

The `libgc` library in Oracle Developer Studio is an obsolete garbage collection library and might be removed in a future release. You can get a modern open-source version of `libgc` here: <http://www.hboehm.info/gc/>

## Fortran Interfaces to C Math Library Functions

With the exception of the random number functions, all non-intrinsic functions and subroutines listed in the `libm_single(3F)`, `libm_double(3F)`, and `libm_quad(3F)` manual pages are deprecated and might be removed in a future release. Fortran programs might invoke the corresponding C functions instead using the C-Fortran interface features described in the Fortran Programming Guide. For example, consider the following Fortran program:

```
double precision d_tanpi
write (*, *) d_tanpi(0.75d0)
end
```

The previous program can be equivalently be written as such:

```
interface
  function tanpi(x)
    double precision, bind(c) :: tanpi
    double precision, value :: x
  end function
end interface
write (*, *) tanpi(0.75d0)
end
```

Existing Fortran programs compiled by Oracle Developer Studio 12.5 and earlier Oracle Solaris Studio releases need not be rewritten in this way provided the corresponding versions of the runtime shared objects with which such programs are linked are available on each platform on which those programs are run.

## Interval Arithmetic Support

The Oracle Developer Studio and earlier Oracle Solaris Studio releases provided support for interval arithmetic on Oracle Solaris SPARC and supported Linux systems in the Fortran 95 and C++ compilers. The interval arithmetic support might be removed in future Oracle Developer Studio releases.

## spot Commands

rip, spot, spot\_cmds, spot\_cmds\_timing, and spot\_diff commands might be removed in a future release for all platforms.

## bw and traps Commands

bw and traps commands might be removed in a future release for all platforms.

## Performance Analyzer Support for MPI

The support for MPI in the Performance Analyzer might be removed in a future release.

## **-xlic\_lib=sunperf Option**

This option is obsolete and might be removed in a future release. Use `-library=sunperf` to link with the Sun Performance Library.

## **Third-Party Software Information**

Oracle Developer Studio 12.5 software includes third-party technologies governed by the [License Information User Manual Oracle Developer Studio 12.5 Last Updated: JULY 2016](#).

## **Usage Data to Oracle**

The Usage Data to Oracle feature periodically sends information on your usage of Oracle Developer Studio components to Oracle Corporation. This information is used by Oracle Corporation to improve future Oracle Developer Studio software releases. This information is anonymous and cannot be associated to any individual or organization.

However, if you wish to disable Usage Data to Oracle, set the `SUNW_NO_UPDATE_NOTIFY` environment variable to any value other than zero (`0`).

## **Documentation For This Release**

The following documentation is available with this Oracle Developer Studio 12.5 release.

- [Oracle Developer Studio 12.5: Installation Guide](#). This document includes information for installing from a tar file, a graphical package installer, and a command-line package installer.
- [What's New in the Oracle Developer Studio 12.5 Release](#). This document describes the new features of the Oracle Developer Studio 12.5 release.
- [Oracle Developer Studio 12.5: Overview](#). This document provides a general description about how to use the compilers and tools and how they work together.
- Online help. The online help available through the Help menu in the IDE provides task-oriented information on using all of the components of the IDE. Online help for Performance Analyzer, Thread Analyzer, Code Analyzer, and `dbxtool` is available through the Help menu in each tool.

- These online reference manual pages describe user commands, libraries that are supplied with the compilers, and other types of commands. Man pages contain reference information, including command syntax, usage, and related commands.  
*Oracle Developer Studio 12.5 Man Pages.* This document contains all the man pages for the Oracle Developer Studio 12.5 release. You can also access these documents through the man command in the installed Oracle Developer Studio 12.5 software.
- Oracle Developer Studio 12.5 manuals and tutorials. You can access these documents in HTML format or PDF format through the documentation library at [http://docs.oracle.com/cd/E60778\\_01/](http://docs.oracle.com/cd/E60778_01/).

## Known Problems, Limitations, and Workarounds in This Release

This section describes some of the known issues at the time of this Oracle Developer Studio 12.5 release, and provides information about how to work around these problems.

### Compiler Issues

This section describes known issues, problems, and workarounds for the compilers in this release.

#### Issues Common to the Compilers

This section describes known issues that apply to the cc, CC, and f95 compilers.

#### Problem with OpenMP Processor Binding Using Reserved Processors

OpenMP processor binding does not respect processor reservations using the taskset command on Linux.

#### x86 Compiler

Using flag `-xarch=amdsse4a` or `-msse4a` in the compilation line to generate an executable program, then when running this program, it causes a segmentation fault on a processor which does not support ISA sse4a, for example, an Intel processor.

## **-xtarget=S7 and -xchip=S7 Options**

-xtarget=S7 and -xchip=S7 options are supported in the compiler driver but not documented in the compiler man pages.

## **UltraSPARC T1 Systems**

Oracle Developer Studio 12.5 compilers will not execute on UltraSPARC T1 systems, including SPARC T1000 and SPARC T2000 systems. You will receive the following error if you try:

```
ld.so.1: cc: fatal: cc: hardware capability (CA_SUNW_HW_1) unsupported: 0x20
```

## **C++ Compiler Issues**

This section describes known issues, problems, and workarounds for the C++ compiler in this release.

### **Problems Compiling BOOST Libraries**

Not all of BOOST libraries compile successfully. The results you get might depend on the version of BOOST you use, and how you exercise the libraries. BOOST libraries are included in Oracle Developer Studio compiler testing and progress is being made toward compiling all of BOOST. If you run into a problem that prevents you from making progress, please report it.

### **Apache Standard Library Issue on Oracle Solaris**

The Apache `stdcxx` library installed in Oracle Solaris 10 8/11 and later, and in the initial releases of Oracle Solaris 11, 11.1, and 11.2, does not work correctly with the new compiler default of `-template=no%extdef`. You will probably need to add the option `-template=extdef` to CC command lines that use this library. The problem is fixed in Oracle Solaris 11.3 SRU2. No fix for Oracle Solaris 10 is currently available. For more information, see [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).

The Apache `stdcxx` library installed in Oracle Solaris 10 8/11 has a syntax error in header `stdcxx4/loc/_moneyunct.h`. This error was not seen by earlier compilers, but is caught by the Oracle Developer Studio 12.5 C++ compiler. There is no way to disable the error detection.

---

**Note** - The syntax error problem was fixed in Oracle Solaris 10 1/13 and Oracle Solaris 11.1.

---

### **Ambiguity: Constructor Call or Pointer-to-Function**

Some C++ statements could potentially be interpreted as a declaration or as an expression-statement. The C++ disambiguation rule is that if a statement can be a declaration, it is a declaration.

In Oracle Solaris Studio 12.2 and earlier, the compiler misinterpreted cases like the following:

```

struct S {
    S();
};
struct T {
    T( const S& );
};
T v( S() );    // ???
    
```

The programmer probably intended the last line to define a variable `v` initialized with a temporary of type `S`. Previous versions of the compiler interpreted the statement that way.

But the construct "`S()`" in a declaration context can also be an abstract declarator (that is, one without an identifier) meaning "function with no parameters returning of value of type `S`." In that case, it is automatically converted to the function pointer "`S(*)()`". The statement is thus also valid as a declaration of a function `v` having a parameter of function-pointer type, returning a value of type `T`.

The compiler now makes the correct interpretation, which might not be what the programmer intended.

There are two ways to modify the code to make it unambiguous:

```

T v1( S() ); // v1 is an initialized object
T v2( S(*)() ); // v2 is a function
    
```

The extra pair of parentheses in the first line is not valid syntax for `v1` as a function declaration, so the only possible meaning is "an object of type `T` initialized with a temporary value of type `S`."

Similarly, the construct "`S(*)()`" cannot possibly be a value, so the only possible meaning is as a function declaration.

The first line can also be written as:

```
T v1 = S();
```

Although the meaning is completely clear, this form of initialization can sometimes result in extra temporaries being created, although it usually does not.

Writing code like the following is not recommended because the meaning is not clear, and different compilers might give different results.

```
T v( S() ); // not recommended
```

## Name Mangling Linking Problems

The name mangling performed by Oracle Developer Studio in `-compat=5` mode has errors. Fixing the errors would have created binary incompatibilities. The errors were found and corrected prior to Oracle Studio releases on x86/Oracle Solaris with `-m64` and on Linux, but remain on the other platforms. A new option, `-abiopt=mangle6` corrects the mangling errors on sparc/Oracle Solaris and x86 Oracle Solaris with the `-m32` option.

Refer to “`-abiopt=[mangle5|mangle6]`” in *Oracle Developer Studio 12.5: C++ User’s Guide*.

## No Support For Referencing a Non-Global Namespace Object From a Template

A program using templates and static objects causes link-time errors of undefined symbols if you compile with `-instances=extern`. This is not a problem with the default setting `-instances=global`. The compiler does not support references to non-global namespace-scope objects from templates. Consider the following example:

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

In this example, a member of a template class references a static namespace-scope variable. Keep in mind that namespace scope includes file scope. The compiler does not support a member of a template class referencing a static namespace-scope variable. In addition, if the template is instantiated from different compilation units, each instance refers to a different `k`, which means that the C++ One-Definition Rule is violated and the code has undefined behavior.

Depending on how you want to use `k` and the effect it should have, the following alternatives are possible. The second option only is available for function templates that are class members.

1. You can give the variable external linkage:

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

All instances use the same copy of `k`.

2. You can make the variable a static member of the class:

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

Static class members have external linkage. Each instance of `C<T>::foo` uses a different `k`. An instance of `C<T>::k` can be shared by other functions. This option is probably what you want.

### **#pragma align Inside Namespace Requires Mangled Names**

When you use `#pragma align` inside a namespace, you must use mangled names. For example, in the following code, the `#pragma align` statement has no effect. To correct the problem, replace `a`, `b`, and `c` in the `#pragma align` statement with their mangled names.

When compiling as C++11, you can use the standard keyword `alignas` to avoid problems with pragmas.

```
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;
}
```

## **C Compiler Issues**

This section describes known issues and problems for the C compiler in this release.

### **Standard Macro FLT\_ROUNDS Behavior**

The implementation of standard C macro `FLT_ROUNDS` that is supplied with the C compiler for SPARC Linux behaves as the C standard specifies and returns the appropriate value for the current rounding mode. It does not always return the value 1 as specified in the man page in SPARC Linux that describes the functions defined in the standard C header file `fenv.h`.



**-xustr=ascii\_utf16\_ushort Option is Incompatible with -std=c11**

Specifying the flag `-xustr=ascii_utf16_ushort` results in an error if `-std=c11` (the compiler default) is in effect.

To avoid the error, you must specify an option that changes the C language dialect accepted by the compiler to an earlier version of C. The recommended options to use are `-std=c99` or `-std=c89`.

The `-xustr=ascii_utf16_ushort` option will also be accepted if you use any of the following options that change the C language dialect: `-ansi`, `-Xc`, `-Xa`, `-Xt`, or `-Xs`.

**Fortran Compiler Issues**

This section describes known issues and problems for the Fortran compiler in this release.

- Blank space before the end of a no advance print line does not affect output position.  
Having the X edit descriptor at the end of a format of an output statement does not affect the position of the next character in the output record. This causes a difference if the output statement has `ADVANCE='NO'` and there are more characters to be transferred to the same record by subsequent output statements.  
  
In many cases, this can be worked around by having a blank character string edit descriptor instead of the `n X` edit descriptor. They are not exactly the same since the blank character string edit descriptor actually causes blank characters to go into the record whereas the `n X` only skips over the next `n` characters, usually causing blanks to be in those skipped positions by default.
- Valid code rejected when a line consists of two continuation ampersands.  
An empty continuation line with a single ampersand is forbidden by the Fortran standard. However, with two ampersands on the same line, an empty continuation line can still be created without falling under the standard restriction. The compiler does not handle that case and gives an error. The workaround is to delete that line which only affects the readability of the program without adding any semantics. BOZ constants sometimes get truncated.
- In some relatively more complex scenarios, such as an array construct, a BOZ constant might get truncated to the default integer size of 4 bytes even though the corresponding item it is supposed to be assigned to is an 8-byte integer entity. The workaround is to use constants of correct type and kind in array construct instead of BOZ constants.
- Correction of rounding algorithm in the new release might cause differences in outputs of list-directed, name-list directed, and formatted-writes using the `ROUND='NEAREST'` and `ROUCH='COMPATIBLE'` compared to outputs in previous releases. The rounding difference should only be in the least-significant digit.

## Fortran Padding Value Change

Padding values that are generated using the `-pad=common` compiler option might be different in code that is compiled with Oracle Developer Studio 12.5 compared to prior releases. If you use `-pad=common` to compile two or more files that reference the same common block, you must compile them with the `f95` compiler from the same Oracle Developer Studio release.

## Fortran 77 Libraries Removed

This is a reminder that the Oracle Solaris Studio 12.2 release removed the obsolete FORTRAN 77 libraries. This means that old executables compiled with the legacy Sun WorkShop `f77` compiler that depend on the shared libraries `libF77`, `libM77` and `libFposix` will not run.

## Array Intrinsic Functions Use Global Registers

The array intrinsic functions `OT_PRODUCT` and `MATMUL` are highly tuned for the appropriate SPARC platform architectures. As a result, they use the global registers `%g2`, `%g3`, and `%g4` as scratch registers.

For interval arithmetic, the array intrinsics `ANY`, `ALL`, `COUNT`, `MAXVAL`, `MINVAL`, `SUM`, `PRODUCT` are also affected.

User code should not assume these registers are available for temporary storage if calls are made to the array intrinsics listed above. Data in these registers will be overwritten when the array intrinsics are called.

## F95 Modules in Archive Libraries Not Included in Executable

The debugger `dbx` requires all object files used in the compilation to be included in the executable file. Usually, programs satisfy this requirement with no extra work on the part of the user. An exceptional case arises from the use of archives containing modules. If a program uses a module, but does not reference any of the procedures or variables in the module, the resulting object file will not contain references to the symbols defined in the module. The linker only links with a object file from an archive if there is a reference to a symbol defined in the object file. If there is no such reference, the object file will not be included in the executable file. `dbx` will give a warning when it tries to find the debugging information associated with the module that was used. It will not be able to provide information about the symbols whose debugging information is missing.

Use the `-u` linker option to work around this problem. This option takes a symbol as its option argument. It adds that symbol to the set of undefined linker symbols, so it will have to be resolved. The linker symbol associated with a module is normally the module name with all letters in lower case followed by an underscore.

For example, to force the object file containing the module `MODULE_1` to be taken from an archive, specify the linker option `-u module_1`. If linking using the `f95` command, use `-Option ld -umodule_1` on the command line.

### **gethrtime(3F) on Linux Platforms**

There is no reliable way to accurately obtain the clock rate on AMD processors when system power saving is enabled. As a result, using timing functions based on `gethrtime(3F)` (the Fortran compiler's Linux version of the Oracle Solaris `gethrtime(3C)` function) to get high resolution real time on Linux platforms will only be accurate on AMD systems with power saving disabled. A reboot of the system might be required to disable the power-saving features.

## **Tools Issues**

This section lists known limitations of the debugging tools and performance analysis tools.

### **dbx Limitations and Incompatibilities**

dbx has the following limitations:

- Using older copies of `libc.so.5` or `libc.so.4` might cause problems for dbx in the area of C++ exceptions. Warning messages about bad stabs and unhandled exceptions might result.  
Workaround: Install the latest `libc.so.5` on all systems.
- Fortran users should compile with the `-stackvar` option to take full advantage of runtime checking.
- Some programs might not work properly with the `-stackvar` option. In such cases, try the `-C` compiler option, which enables array subscript checking without RTC.
- The dbx command line interpreter is an older version of the Korn shell (ksh) that does not support Code Set Independence (CSI). Multi-byte characters can be misinterpreted when typed on the dbx command line.
- The following features of dbx are not available on the Linux OS:
  - Fix and continue
  - Performance data collection on multithreaded applications.

- Pretty-print
- Breakpoints on the following events:
  - `fault`
  - `lastrites`
  - `lwp_exit`
  - `sysin`
  - `sysout`
  - `sync`
- Index DWARF (compiler option `-xs=no`)
- The following problems might occur when debugging programs on Linux platforms:
  - TLS access is not supported.
  - To debug 32-bit programs, you must start `dbx` with the `-x exec32` option.
  - The pipe operator in the Korn shell is limited on Linux platforms. Any `dbx` command that needs to access the target process does not work as part of a pipeline. For example, the following command is likely to cause `dbx` to hang:

```
where | head -1
```

Workarounds:

- Type `Ctrl-C` to display a new `dbx` prompt.
- `dbx` caches a lot of information, so for the above example, the following sequence of commands works:
  - ```
where
```

```
where | head -1
```
  - Redirect command output to a file and then display the file contents.
    - ```
(dbx) > bag where
```

```
(dbx) cat bag
```
- `dbx` does not support the following features for the GNU C and C++ compilers:
  - VL array (`gcc 4.1` and before)
  - OpenMP
  - RTTI
  - Template definition
  - Default argument
  - `using` directive
  - `friend`
- The following issues exist for `dbx` on Oracle Linux 6:

- Indirect reference symbols used in system libraries can sometimes cause dbx to set breakpoints at the reference rather than the actual function.
- When debugging code compiled with gcc 4.4.4 compilers, be aware that dbx does not see macros defined using the -D compiler option.
- The stabs format for debugging information is supported in Oracle Developer Studio, but Oracle has announced that the format might eventually be discontinued in favor of DWARF format. Oracle is not required to implement stabs support for new features or for enhancements to existing features.

The following debugging features are not supported by the stabs format.

- New C++11 and new C++14 features.
- New C11 features
- Parameters and local variables in optimized code.
- Macros (when code is compiled with the -g3 option).
- Subsets of debugging info (generally when compiled with -xdebuginfo={...}).
- Object Oriented Fortran
- See [“dbx attach Profiling \(collect -P\)” on page 32](#) for information about data collection problems when dbx is attached to a process.
- Debugging support for exception handling is limited if GNU runtime libraries are used.

## Performance Analyzer and er\_print Utility Limitations

This section describes known problems with the Performance Analyzer tool and er\_print utility.

- Performance Analyzer crashes with Java 7U65. Use at least Java 7U75 when using the Performance Analyzer tool.
- The Library Visibility functionality for shared objects does not always work properly in conjunction with filtering.
- Performance Analyzer and er\_print cannot find shared objects embedded in jar files. In addition, under some circumstances Performance Analyzer and er\_print cannot find Java class files or source files. To work around these issues, use addpath to point to the directories containing the files. See the Troubleshooting section of the Help in Performance Analyzer for more detailed information.
- When you compare experiments you might see the following issue:  
Source and Disassembly views show inclusive and exclusive metrics instead of only inclusive metrics as expected.
- You might see the following issues related to CPUs with variable clock rate:

- On a system with a variable clock rate, hardware counter profiling metrics for cycle-based counters are under-reported when the processor runs at less than maximum speed and the metrics are converted to times.
- On systems with multiple CPUs running at different clock frequencies, experiments will be processed based on the clock rate for one CPU, which can lead to over-counting or under-counting on other CPUs.
- Profiling OpenMP applications that use many threads on SPARC T5, M5, and M6 systems might produce very large experiments that cannot be read. The workaround is to set the environment variable `SP_COLLECTOR_NO_OMP` before you run `collect`. Note that when you set this variable, you cannot see OpenMP constructs and metrics in the resulting experiment and the User, Expert, and Machine view modes all look the same.

## Limitations for Profiling Applications on a Remote Host

Remote login with a pass phrase is not yet supported.

## `collect` Utility

This section describes known problems with the `collect` utility and data collection for the Performance Analyzer tool.

- Profiling of mixed Java and C++ applications might not be complete when using JDK 1.7.0\_40 through JDK 1.7.0\_59 due to a bug in the JDK. The stack for calls from Java to C++ are not properly unwound.
- Heap tracing on Linux does not trace calls to `calloc`.
- Collection of count data (`collect -c`) does not work for binaries that were compiled with `-std=c++11`.
- Clock profiling on SPARC T4/T5 M5/M6 might have irregular samples due to a bug in Oracle Solaris `setitimer(ITIMER_REALPROF, ...)`
- Hardware counter profiling with attribute `system=1` has the following known issues. Some system time might not be counted correctly for `usleep()`. On Oracle Solaris 10, using the system attribute can hang the application.
- Clock profiling might incorrectly report User CPU Time as Wait CPU Time due to a bug in Oracle Solaris
- You might see following problems with profiling signals in kernel zones:
  - Clock-profiling signals in kernel zones are erratically delivered, leading to missing data.
  - Hardware counter `dtlb_misses.any` doesn't work on Westmere kernel zones
  - Memoryspace profiling on x86 is not supported on kernel zones

- On SPARC, hardware counter profiling samples might be delayed by many seconds; lost samples will result in undercounts
- Clock-profiling samples in kernel zones might report the wrong microstate
- HWC-profiling samples on a the control domain of a system which is running kernel zones might be lost.
- HWC-profiling samples on a kernel zone might be lost.
- Profiling SPECJBB2015 on Linux can cause a crash or hang.
- Various cornercases of stack unwind might fail on x86, both Oracle Solaris and Linux

### **er\_kernel Utility**

This section describes known problems with the `er_kernel` utility.

- `er_kernel` refuses to run on an Oracle Solaris system that is running under Oracle VM to avoid an Oracle VM bug that causes a reboot
- Callstacks might sometimes omits a frame, typically when the leaf function is within its epilogue
- Hardware counter profiling samples on the control domain of a system which is running kernel zones might be lost.
- Hardware counter profiling samples on a kernel zone might be lost.
- Hardware counter profiling with `er_kernel` in a kernel zone might only support a single hardware counter

### **dbx collector Profiling**

This section describes problems with profiling an application when using the `dbx collector` command.

- There is a limitation in using the `dbx collector` on Java. You cannot specify a Java class file for the target. Instead, you must specify the JVM as the target and specify the class file as a parameter to the `dbx run` command. For example:

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

- On Linux, profiling of any Java applications with `dbx` or `collect -P` might fail.

## **dbx attach Profiling (collect -P)**

This section describes problems with profiling a running application when using the `dbx attach` command:

- If you attach `dbx` to a running process that was started without `LD_PRELOAD` to preload the collector library `libcollector.so`, a number of errors can occur. You cannot collect any tracing data: synchronization wait tracing, heap tracing, I/O tracing, or MPI tracing. Tracing data is collected by interposing on various libraries, and if `libcollector.so` is not preloaded, the interposition cannot be done.
- If the target program installs a signal handler after `dbx` is attached to the process, and the signal handler does not pass on the `SIGPROF` and `SIGEMT` signals in Oracle Solaris or the `SIGIO` signal in Linux, some or all profiling data might be lost.
- If the target program uses `libpcp.so`, hardware-counter experiments might fail because both the collector and the program are using the library.
- If the target program calls `setitimer(2)`, clock profiling experiments can be corrupted because both the collector and the program are using the timer.
- Attaching to a target that is in the middle of a call into the `malloc` library might cause the target to fail. Requesting hardware counters when attaching greatly increases the probability of such failures.
- On Linux, attaching to a Java program with `dbx` or `collect -P` might fail.
- On Linux, attaching to a target that is in the middle of a blocking or non-blocking call might cause the target to fail.
- On Linux, attaching to a multithreaded target will not properly record data for threads that are already created at the time of the attach. No warning about missing data is provided. Note that this includes any Java targets because the JVM is multithreaded.

## **IDE Limitations**

This section describes known limitations in the IDE.

To create a project from binary on a remote host, the remote host must have Oracle Solaris Studio 12.3 or 12.4 or Oracle Developer Studio 12.5 available. This feature is not supported on earlier releases.

## **Code Analysis Tools Limitations**

This section describes known limitations to the code analysis tools.



- False positives or negatives might be found if you do not properly instrument your code. For more information, see [“Supported Binaries” in Oracle Developer Studio 12.5: Discover and Uncover User’s Guide](#).
- The discover and uncover tools do not work with an x86\_64 binary compiled with `-xmodel=medium` when the program size (text + data) is large enough to extend beyond the maximum range of `-xmodel=small` ( $2^{32} - 2^{24} - 1$ ).
- The discover tool doesn't work on an ancillary file, which is a file containing debug information created with the linker option `-z ancillary` in Oracle Solaris 11.3.
- To use any of the Code Analysis tools on Oracle Linux, the original binary must be compiled and linked with the `-xannotate=[yes]` option.
- To use Discover SSM in a kernel zone of an SSM capable machine, the configuration of the zone should include `host-compatible=native`

## Java Installer Issues

This section describes known issues, problems, and workarounds for the Java Installer.

### Installation of Oracle Instant Client (OIC)

The command line, non-interactive mode `./developerstudio.sh --non-interactive` will not install the OIC component.

To avoid this, the command line, non-interactive mode for installation must specify OIC together with all the Compiler and Tools components after the `--install-components` option as follows:

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

In addition, if a Japanese or Chinese locale is required, add the components (comma-separated) `japanese-localization` and `simplified-chinese-localization` to the `--install-components` line.

