

Oracle® Developer Studio 12.6: リリース ノート



Part No: E83249-01
2017 年 7 月

Part No: E83249-01

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

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

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.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション(人的傷害を発生させる可能性があるアプリケーションを含む)への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性(redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用了ことに起因して損害が発生しても、Oracle Corporationおよびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはオラクル およびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様とOracle Corporationとの間の契約に別段の定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様とOracle Corporationとの間の契約に定めがある場合を除いて、Oracle Corporationおよびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>) か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

目次

このドキュメントの使用方法	7
Oracle Developer Studio 12.6 リリースノート	9
システム要件	9
必要なシステムソフトウェアパッケージ	10
インストール手順	12
EOF 通知	12
サードパーティーソフトウェアの情報	12
Usage Data to Oracle	13
このリリースのドキュメント	13
このリリースでの既知の問題、制限事項、および回避策	14
コンパイラの問題	14
ツールの問題	21
インストールとアンインストールに関する問題	27

このドキュメントの使用方法

- **概要** - この Oracle Developer Studio 12.6 リリースでの既知の問題やその他の問題に関する情報を提供します。
- **対象読者** - アプリケーション開発者、システム開発者、アーキテクト、サポートエンジニア。
- **必要な知識** - プログラミング経験、ソフトウェア開発テスト、ソフトウェア製品を構築およびコンパイルできる能力。

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは <http://www.oracle.com/pls/topic/lookup?ctx=E83241-01> で入手可能です

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

Oracle Developer Studio 12.6 リリースノート

このドキュメントでは、この製品をインストールおよび使用する前に知っておく必要がある情報を提供します。インストール手順については、『[Oracle Developer Studio 12.6: インストールガイド](#)』を参照してください。

システム要件

Oracle Developer Studio 12.6 ソフトウェアは、SPARC ベースまたは x86 ベースのプラットフォームの Solaris 10 または Oracle Solaris 11.3 オペレーティングシステム、または Oracle Linux オペレーティングシステムにインストールできます。

表 1 システム要件

システムリソース	SPARC ベースシステムの Oracle Solaris	x86 ベースシステムの Oracle Solaris	x86 ベースシステムの Linux
オペレーティングシステム	Oracle Solaris 11.3 (SRU20 以降が必要) Oracle Solaris 10 1/13	Oracle Solaris 11.3 (SRU20 以降が必要) 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 シリーズ SPARC T3 システム以降	Intel Nehalem システム以 降	Intel Nehalem システム以 降
メモリー	最小: 4G バイト	最小: 4G バイト	最小: 4G バイト
インストーラに必要な一時ディ スク容量	2.2G バイト	2G バイト	2G バイト
インストールされた製品で必要 になるディスク容量 df -k コマンドを使用してディ スク容量を確認します	1.2G バイト	1.0G バイト	1.0G バイト
スワップ領域	Oracle Solaris 10: 8G バイ ト以上 Oracle Solaris 11: 推奨さ れるスワップサイズにつ いては、Oracle Solaris の ドキュメントを参照して ください	Oracle Solaris 10: 8G バイ ト以上 Oracle Solaris 11: 推奨さ れるスワップサイズにつ いては、Oracle Solaris の ドキュメントを参照して ください	最小: 8G バイト

システムリソース	SPARC ベースシステムの Oracle Solaris	x86 ベースシステムの Oracle Solaris	x86 ベースシステムの Linux
システムソフトウェアの依存関係	Oracle Solaris 11: システム/ヘッダー、sunpro-incorporation、および developer-studio-utilities パッケージ Oracle Solaris 10: Entire Solaris Software Group Plus OEM Support、Entire Solaris Software Group、または Developer Solaris Software Group	Oracle Solaris 11: システム/ヘッダー、sunpro-incorporation、および developer-studio-utilities パッケージ Oracle Solaris 10: Entire Solaris Software Group Plus OEM Support、Entire Solaris Software Group、または Developer Solaris Software Group	Linux: glibc、i686、glibc-devel、および glibc-devel.i686 を含む Development/Libraries Package Group
Java バージョン	Java 7 (1.7.0_72) 以降、Java 8 (1.8.0_20)	Java 7 (1.7.0_72) 以降、Java 8 (1.8.0_20)	Java 7 (1.7.0_72) 以降、Java 8 (1.8.0_20)

ローカル (クライアント) システムに IDE、パフォーマンスアナライザ、および dbxtool をインストールし、Oracle Developer Studio 製品が動作しているリモートサーバーに接続することもできます。クライアントシステムでサポートされている構成は次のとおりです。

- OS: Microsoft Windows 7 Professional 以降、Ubuntu 9.10、Macintosh OS X 10.7 以降
- Java: 前の表に指定されている Java バージョンのいずれか
- これらのコンポーネントに必要なディスク容量: 300M バイト

必要なシステムソフトウェアパッケージ

Oracle Developer Studio 12.6 のコンパイラやツールを実行する前に、ソフトウェアの実行元となるシステムに、必要なシステムパッケージが含まれていることを確認してください。次の表は、サポートされている各オペレーティングシステムに必要なソフトウェアパッケージを一覧表示しています。

表 2 開発者が必要とするシステムソフトウェア

オペレーティングシステムのバージョン	システムソフトウェアパッケージ
Oracle Solaris 11.3	『Oracle Developer Studio 12.6: インストールガイド』を参照して、適切なバージョンの sunpro-incorporation および developer-studio-utilities パッケージがシステムに含まれていることを確認してください。 次のパッケージがインストールされていることを確認します。 <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</pre>

オペレーティングシステムのバージョン	システムソフトウェアパッケージ
	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 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 SUNWcpu SUNWcsl SUNWcslr SUNWdpl SUNWgnome-base-libs SUNWlibC SUNWlibm SUNWlibms SUNWlibmsr SUNWlxml SUNWpython SUNWsprot SUNWzlib
Oracle Linux 6 Oracle Linux 7	開発/ライブラリパッケージグループおよび次のライブラリを含む 32 ビット実行時サポートライブラリ: 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

注記 - `developer-studio-utilities` および `studio-runtime` は「グループ」パッケージであり、関連するパッケージのグループのインストールを単純化するための IPS 省略形です。これらは完全にほかのパッケージへの依存関係から構成されており、独自の内容は一切持ちません。

`studio-runtime` は、Oracle Developer Studio コンパイラを使用してコンパイルされたプログラムで使用されるシステム実行時サポータライブラリをインストールするグループパッケージです。例として、Fortran ライブラリや Sun Performance ライブラリなどが挙げられます。アプリケーションの配備先となるシステムでは、これをインストールすべきです。

`developer-studio-utilities` は、`studio-runtime` と、Oracle Developer Studio を使用するプログラマによって使用されるシステムユーティリティプログラムをインストールするグループパッケージです。例として、`make`、`sccs`、アセンブラなどが挙げられます。Oracle Developer Studio 12.6 の配備先となるシステムでは、これをインストールすべきです。

インストール手順

すべてのディストリビューションのインストール手順については、『[Oracle Developer Studio 12.6: インストールガイド](#)』を参照してください。

EOF 通知

このリリースで削除された機能および将来のリリースで削除される可能性のある機能のお知らせは、『[End of Features \(EOF\) Planned for Future Releases of Oracle Developer Studio](#)』に記載されています。

サードパーティーソフトウェアの情報

Oracle Developer Studio 12.6 ソフトウェアには、『[License Information User Manual Oracle Developer Studio 12.6 Last Updated: JUNE 2017](#)』によって管理されるサードパーティーのテクノロジーが含まれています。

Usage Data to Oracle

使用状況をオラクルに送付する機能は、Oracle Developer Studio コンポーネントの使用状況に関する情報を、Oracle Corporation に定期的に送信します。この情報は、Oracle Developer Studio ソフトウェアの今後のリリースを改善する目的で、Oracle Corporation によって使用されます。この情報は匿名であり、この情報を個人や組織に関連付けることは一切できません。

ただし、Usage Data to Oracle を無効にする場合は、SUNW_NO_UPDATE_NOTIFY 環境変数をゼロ (0) 以外の任意の値に設定してください。

このリリースのドキュメント

この Oracle Developer Studio 12.6 リリースでは、次のドキュメントを利用できます。

- 『Oracle Developer Studio 12.6: インストールガイド』。このドキュメントには、tar ファイル、グラフィカルパッケージインストーラ、およびコマンド行パッケージインストーラからインストールするための情報があります。
- 『Oracle Developer Studio 12.6 リリースの新機能』。このドキュメントは、Oracle Developer Studio 12.6 リリースの新機能について説明しています。
- 『Oracle Developer Studio 12.6: 概要』このドキュメントでは、コンパイラやツールを使用する方法、およびそれらを連係させて使用する方法について一般的な説明をしています。
- オンラインヘルプ。IDE の「ヘルプ」メニュー経由で使用可能なオンラインヘルプは、IDE のすべてのコンポーネントの使用方法に関するタスク指向の情報を提供します。パフォーマンスアナライザ、スレッドアナライザ、コードアナライザ、および dbxtool のオンラインヘルプは、各ツールの「ヘルプ」メニュー経由で使用できます。
- ユーザーコマンド、コンパイラに付属しているライブラリ、およびその他のタイプのコマンドを説明するオンラインのリファレンスマニュアルページです。コマンドの構文、使用法、関連コマンドなど、参考情報が含まれています。
『Oracle Developer Studio 12.6 マニュアルページ』。このドキュメントには、Oracle Developer Studio 12.6 リリースのすべてのマニュアルページが含まれています。これらのドキュメントには、インストールされた Oracle Developer Studio 12.6 ソフトウェアの man コマンドを使用してアクセスすることもできます。
- Oracle Developer Studio 12.6 のマニュアルやチュートリアル。これらのドキュメントは、ドキュメントライブラリ (http://docs.oracle.com/cd/E60778_01/) から HTML 形式または PDF 形式でアクセスできます

このリリースでの既知の問題、制限事項、および回避策

このセクションでは、この Oracle Developer Studio 12.6 リリースの時点でのいくつかの既知の問題、およびこれらの問題を回避する方法について説明します。

コンパイラの問題

ここでは、このリリースでのコンパイラに関する既知の問題および回避策について説明します。

コンパイラに共通する問題

このセクションでは、cc、CC、および f95 コンパイラに当てはまる既知の問題について説明します。

ハードウェア機能の問題

Oracle Developer Studio コンパイラは、SPARC T1000 および SPARC T2000 システムを含む UltraSPARC T1 ベースシステムでは動作しません。

Oracle Solaris での libsunperf

Oracle Solaris では、libsunperf パッケージおよびパッチは、libsunperf.so 用の新しい LAPACK 3.6.1 ルーチンを提供しません。回避策として、Oracle Developer Studio 12.6 で入手可能な、libsunperf.a とのリンクを行う `-staticlib=sunperf` オプションを使用して静的にリンクします。

予約済みプロセッサを使用した、OpenMP プロセッサのバインドに関する問題

Linux の OpenMP プロセッサのバインドでは、`taskset` コマンドを使用したプロセッサ予約が考慮されません。

x86 コンパイラ

コンパイル行でフラグ `-xarch=amdssse4a` または `-msse4a` を使用して実行可能プログラムを生成した場合、このプログラムの実行時に、Intel プロセッサなどの、ISA sse4a をサポートしないプロセッサで、セグメント例外が発生します。

-xtarget=S7 および -xchip=S7 オプション

-xtarget=S7 および -xchip=S7 オプションは、コンパイラドライバではサポートされていますが、コンパイラのマニュアルページには記載されていません。

C++ コンパイラの問題

このセクションでは、このリリースでの C++ コンパイラに関する既知の問題および回避策について説明します。

BOOST ライブラリのコンパイルに関する問題

すべての BOOST ライブラリが正常にコンパイルされるとは限りません。コンパイルの結果は、使用している BOOST のバージョン、およびライブラリ使用方法によって異なります。BOOST ライブラリは Oracle Developer Studio コンパイラのテストに含まれており、すべての BOOST をコンパイルできるように対応が進められています。作業の進行を妨げる問題が発生した場合はそれを報告してください。

Oracle Solaris での Apache 標準ライブラリの問題

Oracle Solaris 10 8/11 以降、および Oracle Solaris 11、11.1、11.2 の初期リリースにインストールされている Apache stdcxx ライブラリは、新しいコンパイラのデフォルトである `-template=no%extdef` を指定すると正しく動作しません。このライブラリを使用する `cc` コマンド行にオプション `-template=extdef` を追加する必要がある可能性があります。この問題は Oracle Solaris 11.3 SRU2 で修正されました。Oracle Solaris 10 用の修正は、現時点では利用可能になっていません。詳細は、[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) を参照してください。

-xustr=ascii_utf16_ushort オプションは -std=c++11 と互換性がない

`-std=c++11` または `-std=c++14` オプションが有効である場合、フラグ `-xustr=ascii_utf16_ushort` を指定するとエラーになります。

これらのモードでコンパイルするときは、C++11 Unicode 文字および文字列のオプションを使用します。

リンク時の名前符号化の問題

-compat=5 モードの Oracle Developer Studio によって実行される名前符号化には、エラーが含まれています。エラーを修正すると、バイナリの非互換性が作成されました。-m64 を使用した x86/Oracle Solaris と Linux の Oracle Studio リリース以前にこれらのエラーは検出され修正されましたが、ほかのプラットフォームではそのまま残されています。新しいオプション -abiopt=mangle6 は、SPARC/Oracle Solaris および -m32 オプションを使用した x86 Oracle Solaris 上での符号化エラーを修正します。

『Oracle Developer Studio 12.6: C++ ユーザーズガイド』の「-abiopt=[mangle5|mangle6]」を参照してください。

大域的ではない名前空間のオブジェクトをテンプレートから参照できない

プログラムでテンプレートと静的オブジェクトを使用していると、-instances=extern を指定してコンパイルした場合に未定義シンボルのリンク時エラーが発生します。これは、デフォルト設定の -instances=global では問題になりません。コンパイラは、大域的でない名前空間スコープのオブジェクトに対するテンプレートからの参照をサポートしません。次の例を考えてみましょう。

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

この例では、テンプレートクラスのメンバーは静的な名前空間スコープ変数を参照します。名前空間スコープはファイルスコープを含むことに注意してください。コンパイラは、静的な名前空間スコープ変数を参照するテンプレートクラスのメンバーをサポートしません。複数のコンパイル単位からテンプレートがインスタンス化されると、各インスタンスは異なる k を参照します。つまり、C++ 単一定義規則違反が発生し、コードは定義されていない動作を起こします。

k の使用方法やその効果に応じて、次の代替案が考えられます。2 番目のオプションは、クラスのメンバーの関数テンプレートにのみ使用できます。

- 変数に外部リンケージを持たせる

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

すべてのインスタンスは、k の同じコピーを使用します。

- 変数をクラスの静的メンバーにする

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

静的なクラスメンバーは外部リンケージを持ちます。C<T>::foo のインスタンスが使用する k はそれぞれ異なります。C<T>::k のインスタンスは、ほかの関数で共有することができます。通常はこのオプションが使用されます。

名前空間内の #pragma align と符号化名

名前空間内で #pragma align を使用する場合は、符号化名を使用する必要があります。たとえば、次のコードでは、#pragma align 文は何の働きもしません。この問題を解決するには、#pragma align 文の a、b、および c を符号化された名前に変更します。

C++11 としてコンパイルする場合、標準のキーワード alignas を使用すれば、プラグマの問題を回避できます。

```
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 コンパイラの問題

このセクションでは、このリリースでの Fortran コンパイラの既知の問題について説明します。

- 前進なし印刷行の末尾の前にある空白が、出力の位置に影響を与えない。
出力文の書式の末尾に X 編集記述子を指定しても、出力レコード内の次の文字の位置がその影響を受けません。これによって違いが生まれるのは、出力文に **ADVANCE='NO'** が含まれていて、かつ後続の出力文によって同じレコードに転送される文字がほかに存在している場合です。
多くの場合、これを回避するには、**nX** 編集記述子の代わりに空文字列編集記述子を使用します。これらは厳密には同じではありませんが、それは、空文字列編集記述子の場合には実際に空文字がレコード内に挿入されるのに対し、**nX** では次の *n* 個の文字がスキップされるだけであり、それによって通常は、それらのスキップされた位置がデフォルトで空になるからです。
- 2つの継続アンパサンドから成る行が存在していると、有効なコードが拒否される。
単一のアンパサンドを含む空の継続行は、Fortran 規格で禁止されています。しかし、同じ行に2つのアンパサンドが含まれていると、規格の制限の対象から外れ、空の継続行を引き続き作成できます。コンパイラではそのような場合は処理されず、エラーが発生します。回避方法は、その行を削除することであり、それを行っ

ても、プログラムの可読性に影響があるだけであり、意味が付加されることは一切ありません。BOZ 定数が切り捨てられることがある。

- 配列構成のように比較的複雑な一部のシナリオでは、BOZ 定数が、その代入先となると思われる対応する項目が 8 バイト整数エンティティであっても、デフォルト整数サイズの 4 バイトに切り捨てられる可能性があります。回避方法は、BOZ 定数の代わりに正しい型と種別を持つ定数を、配列構成で使用する事です。
- 新しいリリースの丸めアルゴリズムの修正によって、以前のリリースの出力と比べて、ROUND='NEAREST' および ROUND='COMPATIBLE' を指定した並び指示、名前並び指示、およびフォーマットされた書き込みの出力に差異が生じることがあります。丸めの差異は、最下位の数字のみである必要があります。

Oracle Developer Studio 12.6 用の Oracle Solaris libfsu パッチの不足

Oracle Solaris での libfsu のバージョンには、パラメータ化された派生型機能で正しく動作させるために必要な最新のパッチが含まれていない可能性があります。そのため、この機能を使用するプログラムが Linux で動作するときでも Oracle Solaris では正しく動作しません。

この機能を Oracle Solaris で動作させるためには、-Bstatic オプションを使用して libfsu のアーカイブバージョンとリンクします。

libsunmath ライブラリが Linux でサポートされていない

Linux では、libsunmath ライブラリはサポートされていません。そのため、一部の非組み込み数学関数が Linux プラットフォームでのコンパイル時に未解決となります。

libsunmath に含まれる Fortran での呼び出し可能な関数は次のとおりです。

```
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_shufans, 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_llmwcra, i_llmwcra, i_lmwcra, i_lmwcra,
i_mwcra, i_mwcra, i_set_addrans, i_set_lcrans, i_set_mwcra,
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_mwcra, r_mwcra,
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, smwcra, standard_arithmetic, u_addrans, u_lcrans,
u_llmwcra, u_llmwcra, u_lmwcra, u_lmwcra, u_mwcra, u_mwcra,
u_shufrans_

```

Fortran のパディング値の変更

Oracle Developer Studio 12.6 でコンパイルされたコードは以前のリリースを使用した場合に比べ、`-pad=common` コンパイラオプションを使って生成されたパディング値が異なる可能性があります。同じ共通ブロックを参照する 2 つ以上のファイルを、`-pad=common` を使用してコンパイルする場合、それらのコンパイル時に同じ Oracle Developer Studio リリースの f95 コンパイラを使用する必要があります。

Fortran 77 ライブラリが削除された

ここでは、Oracle Solaris Studio 12.2 リリースで廃止対象の FORTRAN 77 ライブラリが削除されたことに注意してください。これは、レガシー Sun WorkShop f77 コンパイラでコンパイルされた、共有ライブラリ `libF77`、`libM77`、および `libFposix` に依存する古い実行可能ファイルが動作しないことを意味しています。

配列組み込み関数における大域レジスタの使用

配列組み込み関数の `OT_PRODUCT` および `MATMUL` は、各 SPARC プラットフォームアーキテクチャー用に高度に調整されています。このため、これらの関数は大域レジスタの `%g2`、`%g3`、および `%g4` をスクラッチレジスタとして利用します。

区間演算の場合は、配列組み込みの `ANY`、`ALL`、`COUNT`、`MAXVAL`、`MINVAL`、`SUM`、`PRODUCT` も影響を受けます。

上記の配列組み込み関数を呼び出す場合に、これらのレジスタが一時記憶領域として利用できることを前提にしたユーザーコードを作成しないでください。これらのレジスタ内のデータは、配列組み込み関数を呼び出したときに上書きされます。

アーカイブライブラリ内の F95 モジュールが実行可能ファイルに含まれない

デバグ `dbx` では、コンパイルに使用されたすべてのオブジェクトファイルが実行可能ファイルの中に含まれている必要があります。通常、ユーザーが追加の作業を実行しなくても、プログラムはこの要件を満たしています。例外となるのは、モジュールを含むアーカイブを使用している場合です。プログラムがモジュールを使用するが、モジュール内の手順または変数をいずれも参照しない場合は、結果として生じるオブジェクトファイルには、モジュール内で定義されるシンボルへの参照は含まれません。オブジェクトファイル内で定義されているシンボルへの参照がある場合のみ、リンカーはアーカイブから取得したオブジェクトファイルをリンクします。そのような参照がない場合、オブジェクトファイルは実行可能ファイルに含められません。使用されたモジュールに関連付けられているデバグ情報を見つけようとすると、`dbx` が警告を出力します。デバグ情報が見つからないシンボルに関する情報は提供できません。

この問題を回避するためには、`-u` リンカーオプションを使用します。このオプションは、1つのシンボルをそのオプション引数として取ります。そのシンボルを未定義のリンカーシンボルのセットに追加し、問題を解決します。モジュールと関連付けられているリンカーシンボルは通常、小文字の文字列に下線が後続するモジュール名です。

たとえば、モジュール `MODULE_1` を含むオブジェクトファイルをアーカイブから取り出すには、リンカーオプション `-u module_1_` を指定します。`f95` コマンドを使用してリンクを実行する場合、コマンド行で `-Qoption ld -umodule_1_` を使用してください。

Linux プラットフォームの `gethrtime(3F)`

システムの省電力が有効になっている場合、AMD プロセッサのクロックレートを正確に取得するための信頼できる方法はありません。結果として、Linux プラットフォーム上で `gethrtime(3F)` (Fortran コンパイラの、Oracle Solaris `gethrtime(3C)` 関数

の Linux 版) に基づく時間関数を使用して実際の時間を高精度で取得する場合、精度の高い値が得られるのは、AMD システムの省電力が無効になっている場合だけです。省電力機能が無効にするには、システムをリブートしなければいけない可能性があります。

ツールの問題

このセクションでは、デバッグツールおよびパフォーマンス分析ツールの既知の制限事項について説明します。

dbx の制限事項と非互換性

dbx には次の制限事項があります。

- libC.so.5 または libC.so.4 の古いコピーを使用すると、C++ 例外の領域で dbx の問題が発生することがあります。不正なスタブや未処理の例外に関する警告メッセージが出力されることがあります。

回避策: 最新の libC.so.5 をすべてのシステムにインストールしてください。

- Fortran ユーザーは、実行時検査をフル活用できるように、-stackvar オプションを使用してコンパイルするようにしてください。
- 一部のプログラムは、-stackvar オプションを使用すると正しく動作しないことがあります。そのような場合には、RTC を使用せずに配列添字検査を有効にする -c コンパイラオプションを試してください。
- dbx コマンド行インタプリタは、Code Set Independence (CSI) をサポートしない旧バージョンの Korn シェル (ksh) です。マルチバイト文字は、dbx コマンド行に入力すると誤って解釈される場合があります。
- dbx の次の機能は、Linux OS では使用できません。
 - 修正継続
 - マルチスレッドアプリケーションでのパフォーマンスデータの収集。
 - 次のイベントのブレークポイント
 - fault
 - lastrites
 - lwp_exit
 - sysin
 - sysout
 - sync
 - インデックス DWARF (コンパイラオプション -xs=no)
- Linux プラットフォームでのプログラムのデバッグでは、次の問題が発生する可能性があります。

- TLS アクセスはサポートされていません。
- 32 ビットプログラムをデバッグするには、`-x exec32` オプションを指定して `dbx` を起動する必要があります。
- Linux プラットフォームの場合、Korn シェルの `pipe` 演算子には制約があります。ターゲットプロセスにアクセスする必要がある `dbx` コマンドはパイプラインの一部として機能しません。たとえば、次のコマンドではおそらく `dbx` がハングアップします。

```
where | head -1
```

回避策:

- Ctrl-C キーを入力して新しい `dbx` プロンプトを表示します。
- `dbx` は多くの情報をキャッシュに格納するので、前述の例の場合、次の一連のコマンドが動作します。

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

- リダイレクトコマンドでファイルに出力してから、ファイルの内容を表示します。

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

- `dbx` は、GNU C および C++ コンパイラで次の機能をサポートしません。
 - VL 配列 (gcc 4.1 以前)
 - OpenMP
 - RTTI
 - テンプレートの定義
 - デフォルト引数
 - `using` ディレクティブ
 - `friend`
- Oracle Linux 6 の `dbx` には、次の問題があります。
 - システムライブラリ内で間接参照シンボルが使用されていると、`dbx` がブレークポイントを実際の関数にではなくその参照に設定する場合があります。
 - gcc 4.4.4 コンパイラを使用してコンパイルされたデバッグコードの場合、`dbx` は `-D` コンパイラオプションを使用して定義されたマクロを認識しないことに注意してください。
- デバッグ情報のスタブ形式は Oracle Developer Studio でサポートされていますが、Oracle はこの形式をいずれは廃止して、DWARF 形式を優先させる可能性があるとして公表しています。Oracle は新機能または既存の機能の機能拡張のためにスタブのサポートを実装する必要がありません。

次のデバッグ機能はスタブ形式でサポートされません。

- C++11 および C++14 の新機能。

- C11 の新機能
- 最適化されたコードのパラメータおよびローカル変数。
- マクロ (コードが `-g3` オプションを指定してコンパイルされている場合)。
- デバッグ情報のサブセット (通常、`-xdebuginfo={...}` を指定してコンパイルされている場合)。
- オブジェクト指向 Fortran
- dbx がプロセスに接続されているときのデータ収集の問題については、[26 ページの「dbx attach を使用したプロファイリング \(collect -P\)」](#)を参照してください。
- GNU 実行時ライブラリを使用する場合、例外処理のデバッグサポートが制限されます。
- 次の機能は、discover 計測型バイナリではサポートされていません。
 - 修正継続
 - カスタマイズ変数の実行時チェック (dbxenv)
- デバッグ対象が HTML モードで計測された場合、discover の分析は dbx コンソールに表示されません。回避策として、計測中はテキストモードオプション `-w` を使用します。

パフォーマンスアナライザおよび er_print ユーティリティの制限事項

このセクションでは、パフォーマンスアナライザツールおよび er_print ユーティリティの既知の問題について説明します。

- Linux では、`collect -P` コマンドを使用して実行中のマルチスレッドプロセス (Java など) に接続すると、データが正しく収集されません。
- Java 7U65 ではパフォーマンスアナライザがクラッシュします。パフォーマンスアナライザツールを使用する場合は、Java 7U75 以上を使用してください。
- 共有オブジェクトの「ライブラリの可視性」機能が、フィルタリングと組み合わせた場合に必ずしも正しく動作しません。
- パフォーマンスアナライザおよび er_print は、jar ファイルに埋め込まれている共有オブジェクトを見つけることができません。また、いくつかの状況で、パフォーマンスアナライザおよび er_print が、Java クラスファイルまたはソースファイルを見つけることができません。これらの問題を回避するには、`addpath` を使用して、ファイルが含まれているディレクトリを指します。詳細は、パフォーマンスアナライザのヘルプのトラブルシューティングのセクションを参照してください。
- 実験を比較する際には、次の問題が発生する可能性があります。

「ソース」および「逆アセンブリ」ビューで、包括的メトリックのみが预期されているときに、包括的なメトリックおよび排他的なメトリックが表示されます。

- CPU の可変クロックレートに関連する次の問題が発生することがあります。
 - 可変クロックレートを持つシステムでは、プロセッサが最大速度より低い速度で実行されてメトリックが回数に変換される場合、サイクルベースのカウンタのハードウェアカウントプロファイリングメトリックが低く報告されます。
 - 異なるクロック周波数で動作する複数の CPU を持つシステムの実験は、1 つの CPU のクロックレートに基づいて進められるため、ほかの CPU 上のイベントが過大または過小にカウントされる可能性があります。
- SPARC T5、M5、および M6 システムで多数のスレッドを使用するプロファイリング OpenMP アプリケーションが、読み込むことができない非常に大きな実験を生成することがあります。回避策は、`collect` を実行する前に、環境変数 `SP_COLLECTOR_NO_OMP` を設定することです。この変数を設定すると、結果の実験の OpenMP 構文およびメトリックが表示されず、「ユーザー」、「上級」、および「マシン」ビューモードがすべて同じように表示されます。

リモートホスト上のプロファイリングアプリケーションの制限事項

パスフレーズを使用したリモートログインはまだサポートされていません。

collect ユーティリティ

このセクションでは、`collect` ユーティリティ、およびパフォーマンスアナライザツールのデータ収集の既知の問題について説明します。

- JDK 1.7.0_40 から JDK 1.7.0_59 を使用した場合、JDK のバグが原因で、Java および C++ のアプリケーションが混在したプロファイリングが完了しないことがあります。Java から C++ への呼び出しのスタックが正しく巻き戻されません。
- Linux のヒープトレースで `calloc` への呼び出しがトレースされません。
- `-std=c++11` を指定してコンパイルされたバイナリに対して、カウントデータ (`collect -c`) の収集が動作しません。
- SPARC T4/T5 M5/M6 でのクロックプロファイリングでは、Oracle Solaris の `setitimer` (ITIMER_REALPROF, ...) のバグにより、不規則なサンプルが発生する可能性があります。
- 属性システムが 1 に設定されたハードウェアカウンタプロファイリングには、次のような既知の問題があります。`usleep()` で、一部のシステム時間が正しくカウントされない可能性があります。Oracle Solaris 10 でシステム属性を使用すると、アプリケーションがハングアップする可能性があります。
- Oracle Solaris のバグにより、クロックプロファイリングでユーザー CPU 時間が待機 CPU 時間として間違って報告される可能性があります。
- カーネルゾーンでのプロファイリングシグナルについて、次の問題が発生する可能性があります。

- カーネルゾーンでクロックプロファイリングシグナルが不規則に配信されてしまい、その結果データが失われます。
- Westmere カーネルゾーンでハードウェアカウンタ `dtlb_misses.any` が動作しません
- x86 上のメモリー空間プロファイリングは、カーネルゾーンではサポートされません
- SPARC では、ハードウェアカウンタプロファイリングのサンプルが何秒も遅れる可能性があります。サンプルが失われるため、実際より少なくカウントされます
- カーネルゾーン内のクロックプロファイリングサンプルから間違ったマイクロステートが報告される可能性があります
- カーネルゾーンを実行しているシステムの制御ドメイン上で、HWC プロファイリングのサンプルが失われる可能性があります。
- カーネルゾーン上で HWC プロファイリングのサンプルが失われる可能性があります。
- Linux 上で SPECJBB2015 のプロファイリングを行うと、クラッシュまたはハングアップが発生する可能性があります。
- x86 上 (Oracle Solaris と Linux の両方) で、スタック巻き戻しのさまざまな特殊ケースでエラーが発生する可能性があります。

er_kernel ユーティリティー

このセクションでは、`er_kernel` ユーティリティーの既知の問題について説明します。

- リブートを誘発する Oracle VM のバグを回避するために、Oracle VM の配下で実行されている Oracle Solaris システムでは、`er_kernel` が実行されません
- 呼び出しスタックで 1 つのフレームが省略されることがありますが、それは特に、リーフ関数とそのエピローグ内にある場合に発生します。
- カーネルゾーンを実行しているシステムの制御ドメイン上で、ハードウェアカウンタプロファイリングのサンプルが失われる可能性があります。
- カーネルゾーン上で、ハードウェアカウンタプロファイリングのサンプルが失われる可能性があります。
- カーネルゾーン内の `er_kernel` によるハードウェアカウンタプロファイリングでは、ハードウェアカウンタが 1 つしかサポートされない可能性があります。

dbx collector を使用したプロファイリング

このセクションでは、`dbx collector` コマンドを使用したときにアプリケーションのプロファイリングで発生する問題について説明します。

- Java で `dbx collector` を使用する場合は制限事項があります。ターゲットに Java クラスファイルを指定できません。代わりに、JVM をターゲットとして指定し、クラスファイルを `dbx run` コマンドのパラメータとして指定する必要があります。例:

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

- Linux 上で、`dbx` または `collect -P` を使用した Java アプリケーションのプロファイリングが失敗することがあります。

dbx attach を使用したプロファイリング (collect -P)

このセクションでは、`dbx attach` コマンドを使用したときに、実行中のアプリケーションのプロファイリングで発生する問題について説明します。

- Linux では、`dbx` または `collect -P` を使用した Java プログラムへの接続が失敗することがあります。
- Linux では、ブロッキング呼び出しまたは非ブロッキング呼び出しを行なっているターゲットに接続すると、ターゲットで障害が発生することがあります。
- Linux では、マルチスレッドのターゲットに接続すると、接続の時点ですでに作成されているスレッドのデータが正しく記録されません。欠落しているデータに関する警告が表示されません。JVM はマルチスレッドであるため、これには Java ターゲットが含まれます。
- コレクタライブラリ `libcollector.so` を事前にロードせずに、`LD_PRELOAD` なしで開始された実行中のプロセスに `dbx` を接続すると、さまざまなエラーが発生する可能性があります。どのトレーシングデータも収集できません。同期はトレーシング、ヒープトレーシング、入出力トレーシング、または MPI トレーシングを待機します。トレースデータの収集は各種ライブラリに割り込むことで行われますが、`libcollector.so` が事前にロードされていなければ、割り込みを行えません。
- `dbx` がプロセスに接続されたあとにターゲットプログラムがシグナルハンドラをインストールし、そのシグナルハンドラが Oracle Solaris の `SIGPROF` シグナルおよび `SIGEMT` シグナル、または Linux の `SIGIO` シグナルを転送しない場合、すべてまたは一部のプロファイリングデータが失われることがあります。
- ターゲットプログラムが `libcpc.so` を使用している場合、コレクタとプログラムが両方ともこのライブラリを使用するため、ハードウェアカウンタ実験が失敗することがあります。
- ターゲットプログラムが `setitimer(2)` を呼び出す場合、コレクタとプログラムの両方がタイマーを使用しているため、クロックプロファイリング実験に失敗する場合があります。

- `malloc` ライブラリに呼び出しを行なっているターゲットに接続すると、ターゲットで障害が発生することがあります。接続しているときにハードウェアカウンタを要求すると、そのような障害が発生する可能性が非常に高くなります。

IDE の制限事項

このセクションでは、IDE の既知の制限事項について説明します。

リモートホストでバイナリからプロジェクトを作成するには、リモートホストで Oracle Solaris Studio 12.3 または 12.4、あるいは Oracle Developer Studio 12.5 が利用可能である必要があります。この機能は、以前のリリースではサポートされません。

コード分析ツールの制限事項

このセクションでは、コードアナライザツールの既知の制限事項について説明します。

- コードの計測が適切でない場合、誤検知または検知漏れが見つかる可能性があります。詳細については、『[Oracle Developer Studio 12.6: Discover および Uncover ユーザーズガイド](#)』の「Supported Binaries」を参照してください。
- プログラムのサイズ (テキスト + データ) が `-xmodel=small` ($2^{32} - 2^{24} - 1$) の最大範囲を超えるほどに大きい場合、`discover` および `uncover` ツールは `-xmodel=medium` を指定してコンパイルされた x86_64 バイナリに対して動作しません。
- `discover` ツールは、Oracle Solaris 11.3 でリンカーオプション `-z ancillary` を指定して作成されたデバッグ情報を含むファイルである付属ファイルに対して動作しません。
- Oracle Linux でコード分析ツールを使用するには、元のバイナリを `-xannotate=[yes]` オプションを指定してコンパイルおよびリンクする必要があります。
- SSM 対応マシンのカーネルゾーンで Discover SSM を使用する場合は、そのゾーンの構成に `host-compatible=native` を含めるようにしてください。

インストールとアンインストールに関する問題

このセクションでは、Java インストーラの既知の問題および回避方法について説明します。

Java インストーラを使用した Oracle Instant Client のインストール

非対話型モードのコマンド行 `./developerstudio.sh --non-interactive` では、OIC コンポーネントがインストールされません。

これを回避するには、非対話型モードのインストール用コマンド行で、`--install-components` オプションのあとに、次のように、OIC と、すべてのコンパイラおよびツールのコンポーネントを指定する必要があります。

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

さらに、日本語または中国語のロケールが必要な場合は、コンポーネント `japanese-localization` および `simplified-chinese-localization` を (コンマで区切って) `--install-components` の行に追加してください。

日本語ロケールが設定されているときにアンインストーラが動作しない

ユーザー環境でロケールを日本語に設定している場合、アンインストーラで日本語文字が正しく表示されません。回避策として、日本語ロケールの設定を解除し、`LC_ALL` 環境を `C` に設定します。

```
$ set LC_ALL=C
```

ロケールに意味がある場合は、`--locale` インストーラフラグを使用します。

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