

インテル® Fortran コンパイラー 18.0 for Linux* リリースノート (インテル® Parallel Studio XE 2018)

このドキュメントでは、新機能、変更された機能、注意事項、および製品ドキュメントに記述されていない既知の問題について説明します。

パッケージに含まれるライセンスと本リリースノートの「著作権と商標について」をお読みください。本リリースのインテル® Fortran コンパイラー 18.0 についての詳細は、次のリンクを参照してください。

- ・ [動作環境](#)
- ・ [使用方法](#)
- ・ [ドキュメント](#)
- ・ [インテルが提供するデバッグ・ソリューション](#)
- ・ [日本語のサポート](#)
- ・ [サンプル](#)
- ・ [再配布可能なライブラリー](#)
- ・ [テクニカルサポート](#)
- ・ [互換性](#)
- ・ [新機能と変更された機能](#)
- ・ [新規および変更されたコンパイラー・オプション](#)
- ・ [終了予定のサポート](#)
- ・ [終了したサポート](#)
- ・ [既知の問題](#)
- ・ [Fortran 2008 および Fortran 2015 機能の概要](#)
- ・ [著作権と商標について](#)

変更履歴

このセクションでは製品アップデートにおける重要な変更内容を説明します。

Update 3 (インテル® Fortran コンパイラー 18.0.3)

- ・ [投機的実行のサイドチャネル問題を軽減するための変更](#)
- ・ [__INTEL_LIBIRC_DEBUG 環境変数](#)
- ・ [報告された問題を修正](#)

Update 2 (インテル® Fortran コンパイラー 18.0.2)

- ・ [投機的実行のサイドチャネル問題を軽減するための変更と新しい -mindirect-branch オプション](#)

- ・ 特定のマイクロアーキテクチャーの指定で新しい開発コード名をサポート
- ・ 報告された問題を修正

Update 1 (インテル® Fortran コンパイラー 18.0.1)

- ・ 日本語版を含む最初のアップデート
- ・ 報告された問題を修正

インテル® Fortran コンパイラー 17.0 以降 (インテル® Fortran コンパイラー 18.0.0 での変更)

- ・ 新しいオプション `-assume contiguous_pointer` と `-assume contiguous_assumed_shape`
- ・ 新しいオプション `-qopt-zmm-usage`
- ・ `-check contiguous` による CONTIGUOUS ポインターへの連続していないポインター代入の診断
- ・ Control-Flow Enforcement Technology (CET) サポート
- ・ SVMML を強制する新しいオプション `-fimf-use-svml`
- ・ SVMML 呼び出しをコンパイル時に割り当て
- ・ `-openmp` オプションの動作の変更
- ・ MEMKIND 高帯域幅メモリー・ディレクティブ
- ・ `-init=[keyword]` による初期化方法の拡張
- ・ すべての `-o*` オプションを `-qo*` オプションに変更
- ・ インストール・イメージからオフライン・ドキュメントを削除
- ・ インテル® Xeon Phi™ 製品ファミリーへのオフロードの新機能と変更された機能
- ・ OpenMP* TR4 Version 5.0 Preview 1 の機能
- ・ プロファイルに基づく最適化のハードウェア・ベースのイベント・サンプリング
- ・ 新規および変更されたコンパイラー・オプション
- ・ Fortran 2008 の機能をサポート
- ・ 報告された問題を修正

[先頭へ戻る](#)

動作環境

アーキテクチャー名についての説明は、「[インテル® アーキテクチャー・プラットフォームの用語](#)」(英語)を参照してください。

- ・ インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) 対応のインテル® 64 アーキテクチャー・ベースのプロセッサを搭載したコンピューター (第 2 世代以降のインテル® Core™ i3/i5/i7 プロセッサ、インテル® Xeon® プロセッサ E3/E5 ファミリー、または互換性のあるインテル以外のプロセッサ)
 - 64 ビット・アプリケーションおよびインテル® Xeon Phi™ プロセッサに作業をオフロードするアプリケーションの開発は、64 ビット・バージョンの OS でのみサポートしています。32 ビット・アプリケーションの開発も、64 ビット・バージョンの OS でのみサポートしています。

- 64 ビット・バージョンの OS で 32 ビット・アプリケーションを開発する場合は、Linux* ディストリビューションからオプションのライブラリー・コンポーネント (ia32-libs、lib32gcc1、lib32stdc++6、libc6-dev-i386、gcc-multilib、g++-multilib) をインストールする必要があります。
- ・ 機能を最大限に活用できるよう、マルチコアまたはマルチプロセッサ・システムの使用を推奨します。
- ・ RAM 2GB (4GB 推奨)
- ・ 4GB のディスク空き容量 (すべての機能をインストールする場合)
- ・ インテル® Xeon Phi™ x200 製品ファミリー向けの開発/テスト
 - インテル® Xeon Phi™ x200 製品ファミリー (開発コード名 Knights Landing)
 - インテル® Xeon Phi™ x200 製品ファミリー向けインテル® Xeon Phi™ Processor Software
- ・ IA-32 対応アプリケーションまたはインテル® 64 対応アプリケーションを開発する場合は、次の Linux* ディストリビューションのいずれか (本リストは、インテル社により動作確認が行われたディストリビューションのリストです。その他のディストリビューションでも動作する可能性はありますが、推奨しません。ご質問は、[テクニカルサポート](#)までお問い合わせください。)
 - Debian* 7.0、8.0、9.0
 - Fedora* 24、25、26
 - Red Hat* Enterprise Linux* 6、7
 - SUSE* Linux* Enterprise Server 11 (SP4)、12 (SP2)
 - Ubuntu* 14.04 LTS、15.10、16.04 LTS、16.10、17.04
 - インテル® Cluster Ready
- ・ Linux* 開発ツール・コンポーネント (gcc、g++ および関連ツールを含む)。 (本リストは、インテル社により動作確認が行われたコンポーネント・バージョンのリストです。その他のバージョンでも動作する可能性はありますが、推奨しません。ご質問は、[テクニカルサポート](#)までお問い合わせください。)
 - gcc 4.3 - gcc 6
 - binutils 2.20 - 2.26
- ・ -traceback オプションを使用するには、libunwind.so が必要です。一部の Linux* ディストリビューションでは、別途入手して、インストールする必要があります。

注

- ・ インテル® コンパイラーは、さまざまな Linux* ディストリビューションと gcc バージョンで動作確認されています。一部の Linux* ディストリビューションには、動作確認されたヘッダーファイルとは異なるバージョンのものが含まれており、問題を引き起こすことがあります。使用する glibc のバージョンは、gcc のバージョンと同じでなければなりません。最良の結果を得るため、上記のディストリビューションで提供されている gcc バージョンのみを使用してください。
- ・ 非常に大きなソースファイル (数千行以上) を -O3、-ipo および -qopenmp などの高度な最適化オプションを使用してコンパイルする場合は、大量の RAM が必要になります。
- ・ 一部の最適化オプションには、アプリケーションを実行するプロセッサの種類に関する制限があります。詳細は、オプションの説明を参照してください。

[先頭へ戻る](#)

インテル® Xeon Phi™ Processor Software

インテル® Xeon Phi™ Processor Software は、インテル® Xeon Phi™ プロセッサの機能を利用するためのソフトウェアとユーティリティーのセットです。インテル® Xeon Phi™ x200 製品ファミリー (開発コード名 Knights Landing) を使用するアプリケーションを開発する場合、インテル® Fortran コンパイラーをインストールする前またはインストールした後に、インテル® Xeon Phi™ Processor Software をインストールできます。

最新バージョンのインテル® Xeon Phi™ Processor Software を使用することを推奨します。詳細は、[インテル® Xeon Phi™ Processor Software のドキュメント](#) (英語) を参照してください。

[先頭へ戻る](#)

インテル® Fortran コンパイラーの使用方法

インテル® Fortran コンパイラーの使用方法は、「インテル® Parallel Studio XE 2018: インテル® Fortran++ コンパイラー 18.0 for Linux* 入門」 ([<install-dir>/documentation_2018/ja/compiler_f/ps2018/get_started_if.htm](#)) を参照してください。

[先頭へ戻る](#)

ドキュメント

製品ドキュメントへのリンクは、[<install-dir>/documentation_2018/ja/compiler_f/ps2018/get_started_if.htm](#) にあります。

インストール・イメージからオフライン・コア・ドキュメントを削除

インテル® Parallel Studio XE のインストール・イメージからオフライン・コア・ドキュメントが削除されました。インテル® Parallel Studio XE のコンポーネントのコア・ドキュメントは、[インテル® ソフトウェア・ドキュメント・ライブラリー](#) (英語) からオンラインで参照できます。また、[インテル® ソフトウェア開発製品レジストレーション・センター](#) から、日本語ドキュメントを含むオフライン・ドキュメントをダウンロードすることもできます: [Product List > Intel® Parallel Studio XE Documentation](#)。

デベロッパー・ガイドおよびリファレンス、新機能とリリースノート、インストール・ガイド

すべてのツール・コンポーネントのデベロッパー・ガイドおよびリファレンス、新機能とリリースノート、インストール・ガイドは、[Intel® Parallel Studio XE Support > Documentation](#) (英語) から入手できます。

[先頭へ戻る](#)

日本語のサポート

日本語のサポートは、すべてのアップデートではなく、一部のアップデートで提供されます。

日本語オペレーティング・システムで英語のサポートを使用する (または英語オペレーティング・システムで日本語のサポートを使用する) 方法については、[こちらの記事](#) (英語) を参照してください。

[先頭へ戻る](#)

インテルが提供するデバッグ・ソリューション

インテルが提供するデバッグ・ソリューションは GNU* GDB ベースです。詳細は、「[インテル® Parallel Studio 2018 Composer Edition for Fortran - デバッグ・ソリューション・リリースノート](#)」(英語) を参照してください。

[先頭へ戻る](#)

サンプル

製品のサンプルは、「[インテル® ソフトウェア製品のサンプルとチュートリアル](#)」(英語) からダウンロードできます。

[先頭へ戻る](#)

再配布可能なライブラリー

詳細は、「[インテル® Parallel Studio XE の再配布ライブラリー](#)」(英語) を参照してください。

[先頭へ戻る](#)

テクニカルサポート

[インテル® ソフトウェア開発製品レジストレーション・センター](#)でライセンスを登録してください。登録を行うことで、サポートサービス期間中 (通常は 1 年間)、製品アップデートと新しいバージョンの入手を含む無償テクニカルサポートが提供されます。

テクニカルサポート、製品のアップデート、ユーザーフォーラム、FAQ、ヒント、およびその他のサポート情報は、<http://www.intel.com/software/products/support/> (英語) を参照してください。

注: 販売代理店が製品のテクニカルサポートを提供している場合、インテルではなく販売代理店にお問い合わせください。

[先頭へ戻る](#)

互換性

一般に、インテル® Fortran コンパイラー for Linux* の以前のバージョン (8.0 以降) でコンパイルされたオブジェクト・コードおよびモジュールは、バージョン 18.0 でもそのまま使用できます。ただし、次の例外があります。

- バージョン 12.0 よりも前のコンパイラーでビルドされた CLASS キーワードを使用して多相変数を宣言しているソースは再コンパイルする必要があります。
- マルチファイルのプロシージャー間の最適化 (-ipo) オプションを使用してビルドされたオブジェクトは、最新のバージョンで再コンパイルする必要があります。
- バージョン 12.0 よりも前のコンパイラーでビルドされた REAL(16)、REAL*16、COMPLEX(16)、COMPLEX*32 データ型を使用しているオブジェクトは再コンパイルする必要があります。
- バージョン 10.0 よりも前のコンパイラーでインテル® 64 アーキテクチャー用にビルドされたモジュール変数を含むオブジェクトは再コンパイルする必要があります。Fortran 以外のソースからこれらの変数を参照する場合、不正な先頭の下線を削除するように外部名を変更する必要があります。
- バージョン 11.0 よりも前のコンパイラーでコンパイルされた、派生型宣言の外部で ATTRIBUTES ALIGN ディレクティブを指定したモジュールは再コンパイルする必要があります。この問題を検出すると、コンパイラーはメッセージを表示します。
- 派生型宣言の内部で ATTRIBUTES ALIGN ディレクティブを指定したモジュールは 13.0.1 以前のコンパイラーでは使用できません。
- Fortran 2008 のサブモジュール機能を実装するため、バイナリー .mod ファイルの内部フォーマットが大幅に変更されました。このため、バージョン 16.0 以降の Fortran コンパイラーで作成されたモジュールファイルは、バージョン 15.0 以前の Fortran コンパイラーで使用することはできません。
- インテル® Xeon Phi™ x100 製品ファミリー向けにコンパイル/ビルドされたオブジェクト/ライブラリーは、インテル® Xeon Phi™ x200 製品ファミリー向けにコンパイル/ビルドされたオブジェクト/ライブラリーと互換性がありません。

[先頭へ戻る](#)

REAL(16) および COMPLEX(16) データ型のスタック・アライメントの変更

バージョン 12.0 よりも古いコンパイラーでは、REAL(16) または COMPLEX(16) (REAL*16 または COMPLEX*32) 項目が値で渡される場合、スタックアドレスは 4 バイトでアラインされます。パフォーマンスを向上するため、バージョン 12 以降のコンパイラーは、これらの項目を 16 バイトでアラインし、引数が 16 バイト境界でアラインされていると仮定します。これは、gcc と互換性があります。

この変更は、主にコンパイラーにより生成される REAL(16) 値の計算を行うライブラリー・ルーチン (組込み関数を含む) の呼び出しに影響します。以前のバージョンでコンパイルしたコードをバージョン 12 のライブラリーとリンクする場合、またはアプリケーションをインテルのランタイム・ライブラリーの共有バージョンにリンクする場合、正しくない結果が返される可能性があります。

バージョン 12.0 よりも古いコンパイラーでコンパイルされている場合、この問題を回避するには、REAL(16) および COMPLEX(16) データ型を使用しているすべての Fortran ソースを再コンパイルしてください。

[先頭へ戻る](#)

新機能と変更された機能

インテル® Fortran コンパイラーは、Fortran 2008 標準のすべての機能をサポートします。インテル® Fortran コンパイラー 18.0 で追加された Fortran 2008 機能は、以下のとおりです。必要に応じて、[Fortran 2008 標準](#) (PDF、英語) および [Fortran 2015 標準の草案](#) (PDF、英語) を参照してください。

Fortran 2008 の機能

- ・ ISO_FORTRAN_ENV の COMPILER_OPTIONS と COMPILER_VERSION
- ・ 三角組込み関数と双曲線組込み関数への COMPLEX 引数
- ・ FINDLOC 組込み関数
- ・ MAXLOC 組込み関数および MINLOC 組込み関数の BACK オプション引数
- ・ PROCEDURE リストにおける複数の型バインド・プロシージャ
- ・ ポインター仮引数への非ポインターデータ項目の引き渡し
- ・ 割り当て可能な左辺 (LHS) による多相代入
- ・ 再帰型と前方参照の割付けコンポーネント
- ・ DATA 文の制限の排除

インテル® Xeon Phi™ 製品ファミリーへのオフロードの新機能と変更された機能

インテルは、ユーザーの課題に最適なソリューションを提供するため、インテル製品の市場を継続的に評価しています。この評価プロセスの一環として、インテルはインテル® Xeon Phi™ コプロセッサ 7200 (開発コード名 Knights Landing) を市場に投入しないことを決定しました。

- ・ インテル® Xeon Phi™ 7200 プロセッサの迅速な採用状況を考慮し、インテルは一般市場に Knights Landing (開発コード名) コプロセッサを展開しないことを決めました。
- ・ インテル® Xeon Phi™ プロセッサは、引き続きインテルのソリューション・ポートフォリオの主要要素として、ユーザーに最も魅力的で競争力のあるソリューションを提供します。

インテル® Xeon Phi™ x100 製品ファミリー (開発コード名 Knights Corner) のサポートは、このリリースで終了しました。インテル® Fortran コンパイラーは、通信にクラスター・ファブリックを利用するクラスター構成において、インテル® Xeon Phi™ x200 製品ファミリーベースのシステムへのオフロードをサポートします。このサポートに関して、新しい機能と変更された機能があります。

インテル® Xeon Phi™ 製品ファミリーをターゲットとするプログラム向けの事前定義済みマクロの変更

インテル® Xeon Phi™ x200 製品ファミリーへのオフロードプログラムをサポートする、新しい事前定義済みマクロ `__AVX512F__`、`__AVX512CD__`、`__AVX512ER__`、`__AVX512PF__` を利用できます。詳細は、製品ドキュメントを参照してください。

ターゲットコンパイル向けに、事前定義済みマクロ `__TARGET_ARCH_MIC` が定義されました。

古いインテル® Xeon Phi™ x100 製品ファミリーをターゲットとする事前定義済みマクロ `__k10m__`、`__KNC__`、`__KNC`、`__MIC__`、`__MIC` は、このリリースからターゲットコンパイル向けに定義されません。

オフロード・コンパイル・ターゲットのデフォルトを変更

このリリースでは、`-qoffload-arch` コンパイラー・オプションのデフォルトが、インテル® Xeon Phi™ x200 製品ファミリーをターゲットとする `mic-avx512` に変更されました。

LD_LIBRARY_PATH

ホスト上のターゲット (インテル® Xeon Phi™ x100 製品ファミリー) ライブラリーの場所は `MIC_LD_LIBRARY_PATH` 環境変数で指定し、ホスト・ライブラリーのみ `LD_LIBRARY_PATH` 環境変数で指定します。

インストール・イメージ・ディレクトリーの変更

Linux* では、`linux/bin`、`linux/compiler/include`、および `<install-dir>/compilers_and_libraries_2018/Linux` の `lib` サブディレクトリーから "intel64_mic" フォルダが削除されました。

新しい環境変数 OFFLOAD_NODES

クラスター内でファブリックを利用してオフロードする場合、`OFFLOAD_NODES=<machines>` を使用してオフロード可能な特定のインテル® Xeon Phi™ x200 製品ファミリーベースのシステムを指定できます。<machines> には、カンマ区切りの名前またはアドレスのリストです。

プロファイルに基づく最適化のハードウェア・ベースのイベント・サンプリング

プロファイルに基づく最適化 (PGO) のハードウェア・ベースのイベント・サンプリングは、インテル® コンパイラーとインテル® VTune™ Amplifier を利用して PGO の多くの利点を得ることができる、新しい低オーバーヘッドのモデルです。インテル® VTune™ Amplifier がサポートされているシステムでデータを収集することができます。詳細は、各製品のデベロッパー・ガイドを参照してください。

OpenMP* 機能

[OpenMP* Technical Report 4 : Version 5.0 Preview 1 \(英語\)](#) 仕様のタスク・リダクション言語機能がサポートされました。

TASKGROUP に TASK_REDUCTION 節が追加されました。

TASK に IN_REDUCTION 節が追加されました。

TASKLOOP に REDUCTION 節と IN_REDUCTION 節が追加されました。

OpenMP* SIMD ディレクティブの新しいキーワード monotonic、overlap、および lastprivate:

```
!$omp ordered simd overlap(overlap_index)
```

```
!$omp ordered simd monotonic([var:step]s)
```

```
!$omp simd lastprivate(conditional:[vars])
```

詳細は、コンパイラー・ドキュメントまたは上記の OpenMP* 仕様へのリンクを参照してください。

新しいディレクティブと追加されたディレクティブ

MEMKIND 高帯域幅メモリー・ディレクティブ

新しい MEMKIND ディレクティブを ALLOCATE 文の直前に追加することで、高帯域幅メモリー (HBW) または非 HBW メモリー割り当てを有効にできます。このディレクティブは、インテル® Xeon Phi™ x200 製品ファミリー (開発コード名 Knights Landing) をターゲットとするインテル® 64 アーキテクチャー・ベースの Linux* システムにのみ適用されます。

```
!DIR$ MEMKIND:{DDR | HBW} [,ALIGN=n]
```

[先頭へ戻る](#)

新規および変更されたコンパイラー・オプション

詳細は、コンパイラーのドキュメントを参照してください。

- ・ [-openmp を -qopenmp に変更](#)
- ・ [-init=\[keyword\]](#)
- ・ [-o* オプションを -qo* オプションに変更](#)
- ・ [-qoffload-arch=\[arch\]](#)
- ・ [-prof-gen-sampling、-prof-use-sampling](#)
- ・ [-assume contiguous_pointer オプションと -assume contiguous_assumed_shape オプション](#)
- ・ [-check contiguous](#)
- ・ [-fimf-use-svml](#)
- ・ [-fimf-force-dynamic-target](#)
- ・ [-cf-potection](#)

廃止予定のコンパイラー・オプションのリストは、ドキュメントのコンパイラー・オプションのセクションを参照してください。

-openmp を -qopenmp オプションに変更

OpenMP* 機能を有効にするには、-qopenmp オプションを使用してください。古い -openmp オプションは、コンパイラーで受け付けなくなりました。

-init=[keyword] オプションの拡張

スカラーと配列の初期化で、これまでの ZERO や SNAN に加えて、HUGE と MINUS_HUGE {表現可能な正/負の最大整数または実数値}、TINY と MINUS_Tiny {表現可能な正/負の最小整数または実数値}、INFINITY と MINUS_INFINITY {正/負の IEEE 無限大} もサポートされるようになりました。

すべての -o* オプションを -qo* オプションに変更

以前のバージョンで非推奨とされていたすべての -o* オプションが、このバージョンでは -qo* オプションに置き換えられました。ただし、Linux* と macOS* で出力ファイルを変更する -o オプションは変更されません。

古い -o* オプションを使用すると、次のようなコンパイラー診断が出力されます。

```
$ ifort -openmp example.f90
```

```
ifort: コマンドライン・エラー: オプション '-openmp' はサポートされていません。オプション '-qopenmp' を使用してください。
```

影響のあるオプション:

- offload-attribute-target=<target>
- offload-option,<target>,<tool>,<opts>
- [no-]offload[=<arg>]
- [no-]openmp
- openmp-lib=<arg>
- openmp-link=<arg>
- [no-]openmp-offload
- [no-]openmp-simd
- openmp-stubs
- openmp-threadprivate=<arg>
- openmp-report[=<level>]
- openmp-task=<arg>
- opt-args-in-regs=<arg>
- [no-]opt-assume-safe-padding
- opt-block-factor=<arg>
- [no-]opt-calloc
- [no-]opt-class-analysis
- [no-]opt-dynamic-align

-[no-]opt-gather-scatter-unroll
-[no-]opt-jump-tables=<arg>
-opt-malloc-options=<arg>
-[no-]opt-matmul
-[no-]opt-mem-layout-trans=<arg>
-[no-]opt-multi-version-aggressive
-[no-]opt-prefetch[=<val>]
-opt-prefetch-distance=<arg>
-opt-ra-region-strategy[=<arg>]
-[no-]opt-report-embed
-opt-report-file=<arg>
-opt-report-filter=<arg>
-opt-report-format=<arg>
-opt-report-phase=<arg>
-opt-report-routine=<arg>
-opt-report-help
-opt-report[=<arg>]
-opt-report-per-object
-opt-streaming-cache-evict=<arg>
-opt-streaming-stores=<arg>
-[no-]opt-subscript-in-range
-opt-threads-per-core=<arg>

-qoffload-arch=[arch] のデフォルトを変更

このリリースでは、-qoffload-arch=[arch] コンパイラー・オプションのデフォルトが、インテル® Xeon Phi™ x200 製品ファミリーをターゲットとする mic-avx512 に変更されました。

新しいコンパイラー・オプション -prof-gen-sampling、-prof-use-sampling

これらの新しいプロファイルに基づく最適化オプションは、インテル® コンパイラーとインテル® VTune™ Amplifier を利用してハードウェア・ベースのイベント・サンプリングを行います。

新しいオプション -assume contiguous_pointer と -assume contiguous_assumed_shape

これらの新しいオプションは、すべての形状引継ぎ配列やポインターにユニットストライドがあることをアサートします。

新しいオプション -check contiguous

このオプションは、CONTIGUOUS ポインターへの連続していないポインター代入の診断に役立ちます。-check all オプションには、このチェックが含まれます。

SVML を強制する新しいオプション -fimf-use-svml

新しいオプションは、スカラー数学演算に LIBM が使用されている場合、SVML の使用を強制します。これにより、ベクトルコードとスカラーコードの計算結果がビット単位で同じになることが保証されます。この機能を利用して、コンパイラーは -fp:precise FP モデルで数学関数をベクトル化し、ベクトル化されたコードはスカラーコードと同じ結果を生成します。

Control-Flow Enforcement Technology (CET) サポート

Control-flow Enforcement Technology (CET) は、ROP (Return-Oriented Programming) や COP/JOP (Call/Jump-Oriented Programming) などの脆弱性を悪用する特定の攻撃からプログラムを防御します。詳細は、[プレビュー・ドキュメント](#) (英語) を参照してください。

新しいコンパイラー・オプション -cf-potection[:keyword] により、コンパイラーの CET サポートを有効にできます。

SVML 呼び出しをコンパイル時に割り当て

SVML 関数に対するコンパイラーのデフォルトの動作が変更され、CPU 固有の SVML エントリーの呼び出しが実行されるようになりました。新しいオプション -fimf-force-dynamic-target を指定すると、以前の動作に戻して動的に SVML をディスパッチできます。

新しいオプション -qopt-zmm-usage

新しいオプション -qopt-zmm-usage:low|high を使用して、コンパイラーにより生成される zmm コードをチューニングできます。引数値 low は、Skylake Server (開発コード名) マイクロアーキテクチャー・ベースのターゲット上のエンタープライズ・アプリケーションなどで、インテル® アドバンスド・ベクトル・エクステンション 2 (インテル® AVX2) ISA からインテル® アドバンスド・ベクトル・エクステンション 512 (インテル® AVX-512) ISA へのスムーズな移行を可能にします。ZMM 命令向けのチューニングには、#pragma omp simd simdlen() などの明示的なベクトル構文を使用することを推奨します。引数値 high は、幅広いベクトル操作を利用して命令ごとの計算量を高める、ベクトル演算が主体の HPC コードなどのアプリケーションに適しています。デフォルト値は、Skylake Server (開発コード名) マイクロアーキテクチャーをターゲットとする場合は low で、コンパイルターゲットを組み合わせる場合は high です。

特定のマイクロアーキテクチャーの指定で新しい開発コード名をサポート

指定した開発コード名のインテル® マイクロアーキテクチャーをサポートするプロセッサ向けに最適化されたコードを生成します。

-x、-ax、-mtune、-march、-arch オプションで次の新しい開発コード名をサポート:

BROADWELL
CANNONLAKE
HASWELL
ICELAKE
IVYBRIDGE

KNL
KNM
SANDYBRIDGE
SILVERMONT
SKYLAKE
SKYLAKE-AVX512

[先頭へ戻る](#)

終了予定のサポート

[先頭へ戻る](#)

終了したサポート

Red Hat* Enterprise Linux* 5 のサポートを終了

このオペレーティング・システム・バージョンのサポートを終了しました。新しいバージョンのオペレーティング・システムに移行してください。

32 ビット・ホストへのインストールのサポートを終了

32 ビット・ホストへのインストールのサポートは、このリリースで終了しました。32 ビット・ターゲット用コードの生成は 64 ビット・ホストでサポートされます (-m32 コンパイラー・オプションを使用)。

インテル® Xeon Phi™ x100 製品ファミリー (開発コード名 Knights Corner) のサポート終了

インテル® Xeon Phi™ x100 製品ファミリー (開発コード名 Knights Corner) は、2017 年 1 月に終息したため、インテル® Parallel Studio XE 2017 でのみサポートされます。インテル® Xeon Phi™ x100 製品ファミリーに対するインテル® Parallel Studio XE 2017 のサポートは、製品終息から 3 年後の 2020 年 1 月に終了します。有効なサポートサービスをお持ちの方にサポートが提供されます。

[先頭へ戻る](#)

既知の問題

OpenMP* 4.5 の OMP THREADPRIVATE と共通ブロック名の特定の使用方法が診断されない

OpenMP* 4.5 の規則では、共通ブロック名を指定する THREADPRIVATE ディレクティブが 1 つのプログラムユニットにある場合、同じ名前を指定する COMMON 文を含むすべてのプログラ

ムユニットで、最後の該当する COMMON 文の後に THREADPRIVATE ディレクティブがなければなりません。インテル® Fortran コンパイラーでは、この使用法が適切に診断されません。

例えば、次のプログラムは OpenMP* 4.5 仕様に準拠していませんが、ifort は OMP THREADPRIVATE 文に続く 2 つの COMMON 文に対してエラーメッセージを出力しません。

```
PROGRAM ex1
  COMMON /common_blk1/x
  !$OMP THREADPRIVATE(/common_blk1/)

  COMMON /common_blk1/y
  COMMON /common_blk1/z

END PROGRAM
```

[先頭へ戻る](#)

__INTEL_LIBIRC_DEBUG 環境変数

古い仮想マシン・ソフトウェアと最新のハードウェアを使用するシステムでは、まれにパフォーマンスが低下することがあります。OS/HW が正しく動作せず、CPUID の libirc のキャッシュサイズが 0 になります。この場合、新しい環境変数 __INTEL_LIBIRC_DEBUG を 1 に設定すると、ランタイムはアポートしてユーザーに通知できるようになります。この環境変数を設定することで、原因を特定して、OS/HW 問題に関する情報を取得できます。

[先頭へ戻る](#)

Fortran 2008 および Fortran 2015 機能の概要

インテル® Fortran コンパイラーは、Fortran 2008 標準のすべての機能をサポートします。また、Fortran 2015 標準の草案のいくつかの機能もサポートします。その他の機能は将来のリリースでサポートされる予定です。このバージョンでサポートされる Fortran 2015 標準の草案の機能は、次のとおりです。

- ・ 「Technical Specification 29113 Further Interoperability with C」のすべての機能。次の機能を含みます。
 - 型引き継ぎ (TYPE(*))
 - ランク引き継ぎ (DIMENSION(..))
 - 互換性のある仮引数の制約の緩和
 - Fortran で使用される C コード操作「C 記述子」を定義する C インクルード・ファイル ISO_Fortran_binding.H

[先頭へ戻る](#)

著作権と商標について

最適化に関する注意事項

インテル® コンパイラーでは、インテル® マイクロプロセッサに限定されない最適化に関して、他社製マイクロプロセッサ用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサ依存の最適化は、インテル® マイクロプロセッサでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサ用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

注意事項の改訂 #20110804

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品適格性、あらゆる特許権、著作権、その他知的財産権の非侵害性への保証を含む) に関してもいかなる責任も負いません。インテルによる書面での合意がない限り、インテル製品は、インテル製品の欠陥や故障によって人身事故が発生するような用途向けに使用することを前提としたものではありません。

インテル製品は、予告なく仕様や説明が変更されることがあります。機能または命令の一覧で「留保」または「未定義」と記されているものがありますが、その「機能が存在しない」あるいは「性質が留保付である」という状態を設計の前提にしないでください。これらの項目は、インテルが将来のために留保しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一切責任を負いません。この情報は予告なく変更されることがあります。この情報だけに基づいて設計を最終的なものとししないでください。

本資料で説明されている製品には、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本資料で紹介されている資料番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725 (アメリカ合衆国) までご連絡いただくか、<http://www.intel.com/design/literature.htm> (英語) を参照してください。

インテル・プロセッサ・ナンバーはパフォーマンスの指標ではありません。プロセッサ・ナンバーは同一プロセッサ・ファミリー内の製品の機能を区別します。異なるプロセッサ・ファミリー間の機能の区別には用いません。詳細については、http://www.intel.co.jp/jp/products/processor_number/ を参照してください。

インテル® Fortran コンパイラーは、インテルのソフトウェア使用許諾契約書 (EULA) の下で提供されます。

詳細は、製品に含まれるライセンスを確認してください。

Intel、インテル、Intel ロゴ、Intel Core、Xeon、Intel Xeon Phi、VTune は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

© 2018 Intel Corporation. 無断での引用、転載を禁じます。

[先頭へ戻る](#)