

インテル® C++ コンパイラー 18.0 for Windows* 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® C++ 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。

- ・ [変更履歴](#)
- ・ [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](#)
- ・ [サンプル](#)
- ・ [テクニカルサポート](#)
- ・ [18.0 の新機能と変更された機能](#)
- ・ [終了予定のサポート](#)
- ・ [終了したサポート](#)
- ・ [既知の制限事項](#)
- ・ [著作権と商標について](#)

変更履歴

Update 3 (インテル® C++ 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と新しい `-mconditional-branch` オプション。詳細は、「[インテル® コンパイラーを使用して投機的実行のサイドチャネル問題を軽減する](#)」(英語)を参照してください。
- ・ [__INTEL_LIBIRC_DEBUG](#) 環境変数

Update 2 (インテル® C++ コンパイラー 18.0.2)

- ・ インテル® コンパイラー (18.0、17.0、16.0) の Visual Studio* IDE 統合にすべての `/Qstd` オプション値に対応した [C/C++ 言語サポート] プロパティを追加
- ・ サポートするすべての VS バージョン (VS2017、VS2015、VS2013) で [C++11 サポートを有効にする (`/Qstd=c++11`)] と [C99 サポートを有効にする (`/Qstd=c++11`)] プロパティを廃止予定

- ・ 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。詳細は、「[インテル® コンパイラーを使用して投機的実行のサイドチャネル問題を軽減する](#)」(英語) を参照してください。
- ・ 報告された問題を修正

Update 1 (インテル® C++ 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
- ・ Microsoft* Visual Studio* 2017 統合の安定性が向上
- ・ 報告された問題を修正

インテル® C++ コンパイラー 17.0 以降 (インテル® C++ コンパイラー 18.0 での変更)

- ・ [新しいオプション /Qopt-zmm-usage](#)
- ・ [Control-Flow Enforcement Technology \(CET\) サポート](#)
- ・ [SVML を強制する新しいオプション /Qimf-use-svml](#)
- ・ [SVML 呼び出しをコンパイル時に割り当て](#)
- ・ [Microsoft* Visual Studio* 2012 のサポート終了](#)
- ・ [インテル® Xeon Phi™ x100 製品ファミリー \(開発コード名 Knights Corner\) のサポート終了](#)
- ・ [Microsoft* Visual Studio* 2017 をサポート](#)
- ・ [すべての -o* オプションを -qo* オプションに変更](#)
- ・ [C++ STL の並列およびベクトル実行向け Parallel STL](#)
- ・ [ハードウェアベースの PGO サポート](#)
- ・ [simd 構文の ordered ブロック向け monotonic および overlap キーワード](#)
- ・ [抽出組込み関数 \(_mm256_extract_epi8 \) の戻り型を変更](#)
- ・ [カウント付き範囲 for ループを並列ループとしてサポート](#)
- ・ [OpenMP* TR4 Version 5.0 Preview 1 の機能](#)
- ・ [OpenMP* 4.0 以降の新機能をサポート](#)
- ・ [新しい C++17 機能をサポート](#)
- ・ [C11 機能の atomic キーワードをサポート](#)
- ・ [新規および変更されたコンパイラー・オプション](#)
- ・ [32 ビット icc ラッパーは 18.0 では非推奨](#)
- ・ [インテル® Cilk™ Plus は 18.0 では非推奨](#)
- ・ [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](#)

[先頭へ戻る](#)

動作環境

- ・ 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・プロセッサをベースとするコンピューター (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)
- ・ 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® グラフィックス・テクノロジーへのオフロードまたはネイティブサポートの開発/テスト
 - 次のプロセッサ・モデルをサポートしています。
 - § インテル® Xeon® プロセッサ E3-1285 v3 および E3-1285L v3 (インテル® C226 チップセット) (インテル® HD グラフィックス P4700)
 - § 第 5 世代インテル® Core™ プロセッサ (インテル® Iris® Pro グラフィックス、インテル® HD グラフィックス)
 - § 第 4 世代インテル® Core™ プロセッサ (インテル® Iris® Pro グラフィックス、インテル® Iris® グラフィックス、またはインテル® HD グラフィックス 4200+ 番台)
 - § 注: リストされているチップセットのインテル® Xeon® プロセッサのみサポートしています。ほかのチップセットのインテル® Xeon® プロセッサはサポートしていません。前世代のインテル® Core™ プロセッサはサポートしていません。インテル® Celeron® プロセッサおよび Intel Atom® プロセッサとの互換性はありません。
 - インテル® グラフィックス・テクノロジー対応の最新の 32 ビットまたは 64 ビット・グラフィックス・ドライバー (インテル® ダウンロード・センター (英語) から入手できます)
 - Windows* 用 binutils (<http://intel.ly/1fHX7xO> (英語) から入手できます)
 - § binutils をインストールした後、ld.exe を含むディレクトリーを PATH に追加する必要があります。
- ・ Microsoft* Windows* 7 (SP1)、Microsoft* Windows* 8、Microsoft* Windows* 8.1、Microsoft* Windows® 10、Microsoft* Windows Server* 2012 (R2) (エンベデッド・エディションはサポートされていません)、Windows Server* 2016[1]
 - Microsoft* Windows* 8、Microsoft* Windows* 8.1 および Microsoft* Windows Server* 2012 では、製品は「デスクトップ」環境にインストールされます。「Windows* 8 UI」アプリケーションの開発はサポートされていません。[2]
- ・ 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のビルドに、Microsoft* Visual Studio* 開発環境あるいはコマンドライン・ツールを使用する場合は、次のいずれか:
 - Microsoft* Visual Studio* 2017 Professional Edition 以上 (「C++ によるデスクトップ開発」コンポーネントがインストールされていること)
 - Microsoft* Visual Studio* 2015 Professional Edition 以上 (「Visual C++* 2015 用の共通ツール」コンポーネントがインストールされていること) [3]
 - Microsoft* Visual Studio* Community 2015 以上 (「Visual C++* 2015 用の共通ツール」コンポーネントがインストールされていること) [3]
 - Microsoft* Visual Studio* 2013 Professional Edition 以上 (C++ コンポーネントがインストールされていること)
 - Microsoft* Visual Studio* Community 2013 以上 (C++ コンポーネントがインストールされていること)
- ・ IA-32 [1] アーキテクチャー・アプリケーションのビルドに、コマンドライン・ツールのみを使用する場合は、次のいずれか:
 - Microsoft* Visual C++* Express 2015 for Windows* Desktop
 - Microsoft* Visual C++* Express 2013 for Windows* Desktop

- ・ インテル® 64 対応アプリケーションのビルドに、コマンドライン・ツールのみを使用する場合は、次のいずれか:
 - Microsoft® Visual C++® Express 2015 for Windows® Desktop
 - Microsoft® Visual C++® Express 2013 for Windows® Desktop
 - Microsoft® Windows® Software Development Kit for Windows® 8
- ・ ドキュメントの参照用に Adobe® Reader® 7.0 以降

注

1. アプリケーションは、上記の開発用と同じ Windows® バージョンで実行できます。また、Windows® XP よりも前の非エンベデッドの Microsoft® Windows® 32 ビット・バージョンでも実行できますが、インテルではこれらの互換性テストを行っていません。開発アプリケーションは、古いバージョンの Windows® にはない Win32 API ルーチンを使用している可能性があります。アプリケーションの互換性テストをご自身の責任で行ってください。アプリケーションを実行するには、特定のランタイム DLL をターゲットシステムにコピーしなければならないことがあります。
2. インテル® C++ コンパイラーは、Windows® 8 UI アプリケーションの開発をサポートしていません。インテルでは、ユーザーの皆様のご意見を常に参考にしています。例えば、Windows® 8 UI アプリケーションにインテル® C++ コンパイラーまたはその他のインテル® ソフトウェア開発製品の機能を利用したい方は、[Online Service Center](#) (英語) からご意見をお送りください。インテル® ソフトウェア開発製品に関するご意見・ご要望は英語でのみ受け付けています。
3. インテル® C++ コンパイラーを Microsoft® Visual Studio® 2015 で使用するには、Visual Studio® から「Visual C++® 2015 用の共通ツール」コンポーネントをインストールする必要があります。[こちらの記事](#) (英語) の説明を参照してください。
4. Microsoft® Visual Studio® 2017 でインテル® C++ コンパイラーを使用する方法については、[こちらの記事](#) (英語) の説明を参照してください。
5. 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® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS)

インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS) は、インテル® C++ コンパイラーのインストール前またはインストール後にインストールできます。最新バージョンのインテル® MPSS を使用することを推奨します。インテル® Parallel Studio XE for Windows® を登録すると、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 (<http://registrationcenter.intel.com>) から入手できます。

ユーザー空間およびカーネルドライバーのインストールに必要な手順については、Intel® MPSS のドキュメントを参照してください。

[先頭へ戻る](#)

Intel® C++ コンパイラーの使用法

コマンドラインおよび Microsoft* Visual Studio* からの Intel® C++ コンパイラーの使用法は、「Intel® Parallel Studio XE 2018: Intel® C++ コンパイラー 18.0 for Windows* 入門」(<*install-dir*>%documentation_2018%ja%compiler_c%ps2018%get_started_wc.htm) を参照してください。

[先頭へ戻る](#)

ドキュメント

製品ドキュメントは、<*install-dir*>%documentation_2018%ja%compiler_c%ps2018%get_started_wc.htm からリンクされています。

Microsoft* Visual Studio* のオンラインヘルプ形式

新しい [ヘルプ] メニュー項目は、入門ドキュメントにジャンプします。
F1 キーで表示される状況依存ヘルプは**利用できません**。

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

Intel® 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。Intel® Parallel Studio XE のコンポーネントのコア・ドキュメントは、[Intel® 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、[Intel® 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**。これには、日本語ドキュメントも含まれています。

- ・ [ヘルプ設定の設定] が [ブラウザで起動] に設定されている場合、[ツール] > [オプション] > [F# ツール] または [ツール] > [オプション] > [Intellitrace] で *F1* キーを押すと、ブラウザが 2 つ開きます。
- ・ Chrome*: 検索またはキーワードからトピックを表示すると、目次が同期しません。[トピックを同期] も動作しません。
- ・ Firefox*: 目次が表示されないことがあります。検索の大文字と小文字は区別されません。
- ・ Safari*: Windows* の反応が遅くなります。

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。

[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 XE 2018 Composer Edition for C++ - 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](#)

テクニカルサポート

インストール時に製品の登録を行わなかった場合は、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 (<http://registrationcenter.intel.com>) で登録してください。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: 販売代理店がこの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](#)

新機能と変更された機能

このバージョンでは、次の機能が新たに追加または大幅に拡張されています。これらの機能に関する詳細は、ドキュメントを参照してください。

`__INTEL_LIBIRC_DEBUG` 環境変数

OS/HW が正しく動作せず、CPUID の libirc のキャッシュサイズが 0 の場合、パフォーマンスが低下する可能性があります。その場合、環境変数 `__INTEL_LIBIRC_DEBUG=1` を定義して、原因を特定し、OS/HW 問題に関する情報を取得できます。

例: `export __INTEL_LIBIRC_DEBUG=1`

特定の CPU の指定で新しい開発コード名をサポート

次の新しい cpuid (開発コード名) をサポート

- ・ haswell
- ・ broadwell
- ・ skylake
- ・ skylake_avx512
- ・ knl
- ・ knm

次の関数属性で利用可能 (関数を特定の cpu 開発コード名向けに手動で指定):

```
C/C++ __attribute__((cpu_dispatch()))  
C/C++ __attribute__((cpu_specific()))  
C/C++ __attribute__((vector(processor()))
```

Microsoft* Visual Studio* 2017 をサポート

インテル® C++ コンパイラーの統合サポートに Microsoft* Visual Studio* 2017 が追加されました。詳細は、「[動作環境](#)」を参照してください。

simd 構文の ordered ブロック向け monotonic および overlap キーワード

既存の #pragma omp ordered simd 用の新しいキーワード

- ・ #pragma omp ordered simd monotonic()
- ・ #pragma omp ordered simd overlap(expr)
- ・ #pragma omp simd reduction(=: list)

詳細は、『インテル® C++ コンパイラー 18.0 デベロッパー・ガイドおよびリファレンス』を参照してください。

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、インテル® Xeon® Platinum プロセッサー (開発コード名 Skylake) 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、インテル® Core™ マイクロアーキテクチャー/インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャーのインテル® AVX-512 をターゲットとする場合は high です。

ハードウェアベースの PGO サポート

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、『インテル® C++ コンパイラー 18.0 デベロッパー・ガイドおよびリファレンス』を参照してください。

C++ STL の並列およびベクトル実行向け Parallel STL

インテル® C++ コンパイラーとともに、ポリシーの実行をサポートする C++ 標準ライブラリーのアルゴリズムの実装である Parallel STL がインストールされます。

Parallel STL の詳細は、[入門 \(英語\)](#) と [リリースノート \(英語\)](#) を参照してください。

抽出組込み関数 (`_mm256_extract_epi8`) の戻り型を変更

256 ビット・ベクトル組込み関数 `_mm256_extract_epi8/epi16(__m256i a, const int index)` が `__int8/__int16` 値の代わりに整数値を返すように定義されました。

SVML を強制する新しいオプション `/Qimf-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 `/Qcf-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ます。新しいオプション `/Qimf-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。

OpenMP* TR4 Version 5.0 Preview 1 の機能

[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。

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。

OpenMP* 4.0 以降の新機能をサポート

- ・ taskloop 構文 `#pragma omp taskloop[clause[[,]clause]..]` をサポート
- ・ `#pragma omp for linear (list [: linear-step])` をサポート
 - `list` は `list` または `modifier(list)` のいずれか
- ・ `linear` 節で `ref val uval` 修飾子をサポート
 - 例: `linear(ref(p))`, `linear(val(i):1)`, `linear(uval(j):1)`
- ・ `#pragma omp simd simdlen(n)` をサポート
- ・ `#pragma omp ordered [simd]` をサポート
- ・ 配列全体のリダクション: `int x[n]; #pragma omp simd reduction(+:x)`
- ・ `processor` 節の拡張を `#pragma omp declare simd` に追加 (OpenMP* 4.5 の正式な機能ではありません)
- ・ `#pragma omp for schedule` 節の SIMD 修飾子と NONMONOTONIC 修飾子をサポート
 - インテル® C++ コンパイラー 18.0 には、ループの反復をチームのスレッド間でどのように分割するかについて、ユーザー制御を強化する SIMD 修飾子と NONMONOTONIC 修飾子の拡張が含まれています。詳細は、『インテル® C++ コンパイラー・デベロッパー・ガイドおよびリファレンス』を参照してください。
- ・ reduction 節で部分配列をリスト項目としてサポート
 - `reduction(reduction-identifier:list)`: リスト項目が部分配列の場合、reduction 節がセクションの個々の要素に適用されると見なして処理されず、プライベートの部分配列の要素は、連続して割り当てられます。

C++17 の機能をサポート

インテル® C++ コンパイラー 18.0 は、`/Qstd=c++17` (Windows*) または `-std=c++17` (Linux*/macOS*) コンパイラー・オプションで以下の C++17 の機能をサポートします。

- ・ `static_assert` のメッセージ省略を許可 (N3928)
- ・ 範囲 `for` ループの制限緩和 (N3994)
- ・ 以前の主要バージョンのコンパイラーとの比較を含む、サポートしている機能の最新リストは、『[インテル® C++ コンパイラーでサポートされる C++17 の機能](#)』を参照してください。

C11 の機能をサポート

インテル® C++ コンパイラーは、`/Qstd=c11` (Windows*) または `-std=c11` (Linux*/macOS*) コンパイラー・オプションで以下の C11 の機能をサポートします。

- ・ `_Atomic` および `__attribute__((atomic))` キーワードを含むすべての C11 の機能

- ・ 以前の主要バージョンのコンパイラーとの比較を含む、サポートしている機能の最新リストは、「[インテル® C++ コンパイラーにおける C11 サポート](#)」を参照してください。

新規および変更されたコンパイラー・オプション

コンパイラー・オプションの詳細は、『[インテル® C++ コンパイラー 18.0 デベロッパー・ガイドおよびリファレンス](#)』の「コンパイラー・オプション」セクションを参照してください。

- ・ **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。

Windows* では、/Qoffload-option コンパイラー・オプションでターゲット・コンパイラーに渡されるコンパイラー・オプションに影響します。

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。

```
$ icc -openmp example.c
icc: 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:

- [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>

廃止予定のコンパイラー・オプションのリストは、『[インテル® C++ コンパイラー 18.0 デベロッパー・ガイドおよびリファレンス](#)』の「[コンパイラー・オプション](#)」セクションを参照してください。

[先頭へ戻る](#)

終了予定のサポート

インテル® Cilk™ Plus は 18.0 では非推奨

インテル® Cilk™ Plus は、インテル® C++ コンパイラー 18.0 では非推奨の古い機能です。プロセッサ・グラフィックスへのオフロードには、OpenMP* ベースの構文を使用することを推奨します。詳細は、『[インテル® Cilk™ Plus の代わりに OpenMP* またはインテル® TBB を使用するためのアプリケーションの移行](#)』を参照してください。

32 ビット icc ラッパーは 18.0 では非推奨

icl: リマーク #10421: IA-32 ターゲット・ラッパー・バイナリー 'icl' は古いオプションです。コンパイラー・スタートアップ・スクリプトまたは適切なインテル(R) 64 コンパイラー・バイナリーで '/Qm32' オプションを使用して目的のアーキテクチャーをターゲットにしてください。

終了したサポート

Microsoft* Visual Studio* 2012 は 18.0 ではサポート終了

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

- Visual Studio* の [Code Generation [Intel C++]] (コード生成 [インテル(R) C++]) > [Offload Target Architecture (オフロード・ターゲット・アーキテクチャー)] から (/Qoffload-Arch): "mic" が削除されました。

- ・ [Platform Toolset (プラットフォーム・ツールセット)] が "Intel C++ Compiler 17.0" に設定されているプロジェクトで "mic" 値が指定されている場合、"Intel C++ Compiler 18.0" プラットフォーム・ツールセットへアップグレード後に "既定値" に更新されます。

IA-32 ホストへのインストールのサポートを終了

IA-32 ホストへのインストールのサポートを終了しました。32 ビット・ターゲット用コード生成のサポートは 64 ビット・ホストでサポートされます (/Qm32 コンパイラー・オプションを使用)。

`_GFX_enqueue` を削除

`_GFX_enqueue` は削除されました。 `_GFX_offload` に変更してください。

[先頭へ戻る](#)

既知の制限事項

インテル® Xeon Phi™ コプロセッサをターゲットとするオフロード・アプリケーションには Microsoft* Visual Studio* 2015 または 2017 が必要

インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS) のインテル® Xeon Phi™ x200 製品ファミリーのサポートの制限により、現在 Microsoft* Visual Studio* 2013 ではオフロード・アプリケーションを実行できません。Microsoft* Visual Studio* 2013 でオフロード・アプリケーションをビルドすることはできますが、実行すると次のエラーになります: "The remote process indicated that the following libraries could not be loaded libioffload_target.so.5 libiomp5.so libcilkrts.so.5 offload error: cannot start process on the device 0 (error code 19)" (リモートプロセスは、次のライブラリーをロードできませんでした: libioffload_target.so.5 libiomp5.so libcilkrts.so.5。オフロードエラー: デバイス 0 でプロセスを開始できません (エラーコード 19))。

Microsoft* Visual Studio* 2013 でビルドしたオフロード・アプリケーションの実行には、インテル® コンパイラーのインテル(R) 64 VS2015 環境またはインテル(R) 64 VS2017 環境から、Microsoft* Visual Studio* 2015 または 2017 を 'devenv' として実行する必要があります。

このエラーメッセージは、ショートカット・メニューから VS2015/VS2017 を実行した場合のみ出力され、インテル® Parallel Studio XE 環境のコマンドラインから実行した場合には出力されません。

VS2013 は別の問題でも失敗し、メッセージが表示されます。インテル® Xeon Phi™ コプロセッサ向けのオフロード・アプリケーションのビルドと実行には、Microsoft* Visual Studio* 2015 または 2017 のみを使用することを推奨します。

この制限は、インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS) の将来のバージョンで修正される可能性があります。

ポインターチェッカーにダイナミック・ランタイム・ライブラリーが必要

`/Qcheck-pointers` オプションを使用する場合は、ランタイム・ライブラリー `libchkp.dll` をリンクする必要があります。`/MD` のようなオプションを `/Qcheck-pointers` とともに使用すると、設定に関係なくこのダイナミック・ライブラリーがリンクされることに注意してください。詳細は、<http://intel.ly/1jV0eWD> (英語) を参照してください。

日本語版 Windows* にインストールすると IDE からインテル® コンパイラーのヘルプ・ドキュメントを起動できない

複数のペインを含むドキュメントが Visual Studio* 内のブラウザーで正しく表示されない

Visual Studio* 内のブラウザーには複数のペインを含むドキュメントが正しく表示されない制限があります (左のペインに目次が表示されますが、右のペインにコンテンツが表示されません)。回避策: Visual Studio* の [ヘルプ] メニューから同じドキュメントにアクセスします。

異なるコンパイラーを使用して 256 ビット・ベクトル型引数をコンパイルするとランタイムにアクセス違反が発生するコードが生成される

2 つの異なるコンパイラー (Microsoft* Visual C++* 2013 コンパイラーおよびインテル® C++ コンパイラー 15.0 以降) を使用してアプリケーションを作成した場合、アライメントされていないデータアクセスによる一般保護違反が発生することがあります。この問題は、呼び出し元を Visual C++* でビルドし、256 ビット・ベクトル型引数を参照で渡して、インテル® C++ コンパイラーでビルドした関数とその引数にアクセスすると発生します。

原因は、256 ビット・ベクトル型引数のアライメントが一致しないためです。

`<code>` にインテル® AVX 以降の新しいコード値 (CORE-AVX-I, CORE-AVX2, その他) を指定して `/Qx<code>` コンパイラー・オプションを使用する場合は、アプリケーションのソースコードで `__mm256_stream_*` (非テンポラルデータのロード/ストア組込み関数) が明示的に使用されない限り、実際にはアライメントされていないアクセス命令がこれらのインスタンスで使用されるため、この問題は発生しません。

Visual Studio* の既知の問題

・ インテル® C++ コンパイラーの Visual Studio* 2017 統合の問題

Microsoft* Visual Studio* 2017 との統合ではいくつかの問題が見つかっています。これらは間欠的な問題で、すべてのシステムで発生するわけではありません。最新の Visual Studio* 2017 バージョン 15.3.3 では、統合に関する問題は報告されていません。詳細は、「[インテル® ソフトウェア開発ツールの Microsoft* Visual Studio* 2017 統合に関する問題](#)」を参照してください。

- Microsoft* Visual Studio* 2017 ではルーチンに対してガイド付き自動並列化の解析を利用できません

Visual Studio* 2017 バージョン 15.4 以降では、エディターで選択したルーチンに対して、コンテキストメニューの [Intel Compiler (インテル(R) コンパイラー)] > [Guided Auto Parallelism (ガイド付き自動並列化)] > [Run Analysis on Routine... (ルーチン ... の解析を実行)] を利用できません。行の範囲、ファイル、プロジェクトの解析は実行できます。この問題は、将来のリリースで修正される予定です。

- MSVCP90D.dll (またはその他の Microsoft* ランタイム DLL) が見つからない

サンプル・プロジェクト (および Microsoft* Visual C++* プロジェクト) を実行するときに Microsoft* Visual Studio* のランタイム DLL が見つからない場合、ランタイムエラーが発生します。これは、マニフェスト・ファイルや SXS アセンブリーが見つからないことが原因です。この問題を解決するには、使用しているバージョンの Microsoft* Visual Studio* の `redist` フォルダ (デフォルトの場所は `c:\program files[(x86)]\Microsoft Visual Studio X.X\VC\redist`) に移動します。amd64、x86、Debug_NonRedist サブフォルダで、必要なランタイムが含まれているフォルダを探します (デバッグ・ライブラリーを探す場合は、ファイル名の最後が D のファイルが含まれているフォルダを探します)。必要なランタイムが含まれているフォルダが見つかったら、そのフォルダの (.manifest ファイルを含む) すべての内容を、実行する .exe ファイルのあるフォルダにコピーします。

- 警告 #31001: pdb にアクセスするための dll (mspdb110.dll など) が指定されたパスに存在しません。

Microsoft* Visual Studio* Express を使用して、IA-32 向けのコンパイルで /debug ([Fortran] > [General (全般)] > [Debug Information Format (デバッグ情報の形式)]) を有効にすると、次の警告が出力されることがあります。

警告 #31001: pdb にアクセスするための dll (mspdb110.dll など) が指定されたパスに存在しません。

通常は設定エラーです。コンパイルは /Zi の代わりに /Z7 を使用して継続されますが、プログラムをリンクするときに同様のエラーが発生する可能性があります。64 ビットの Windows* アプリケーション (つまり、インテル® Visual Fortran コンパイラー) は 64 ビットの dll を使用する必要がありますが ([https://msdn.microsoft.com/ja-jp/library/windows/desktop/aa384231\(v=vs.85\).aspx](https://msdn.microsoft.com/ja-jp/library/windows/desktop/aa384231(v=vs.85).aspx))、Microsoft* Visual Studio* Express は 32 ビット・バージョンの mspdb*.dll のみ提供しており、64 ビット・バージョンがないため、この警告が表示されます。この警告が表示されても、実行ファイルは正常にビルドされます。ただし、デバッガーで使用されるデバッグ情報はオブジェクト・ファイル (.obj) に埋め込まれます。/Z7 が有効な場合、コンパイラーは .pdb ファイルを生成しません。

- ・ インテル® Parallel Studio XE 2018 Beta のインテル® C++ および Fortran コンパイラの Visual Studio* 2017 統合
 - VS2017 では、ベース・プラットフォーム・ツールセットが制限されます。利用可能なベース・プラットフォーム・ツールセットは v141 および v141_xp です。v140、v140_xp、v120、v120_xp などの古いビルド環境を使用したい場合は、対応する Visual Studio* バージョンへのインテル® C++ コンパイラの統合をインストールしてプロジェクトをビルドするか、サポートされる古いバージョンの Visual Studio* のベース・プラットフォーム・ツールセットを使用してください。
 - VS2017 では、インテル® Fortran モジュールウィザードによって生成されるファイルは、Fortran プロジェクトに自動的に追加されません。[プロジェクト] > [既存の項目の追加] を選択して、ファイルを手動で指定する必要があります。
 - VS 2017 でインテル® C++ コンパイラーとインテル® スレッディング・ビルディング・ブロック (インテル® TBB) を利用してオフロード・アプリケーションをコンパイルするには、プロジェクト・プロパティの [C/C++] > [Command Line (コマンドライン)] > [Additional Options (追加のオプション)] に /Qtbb オプションを追加します。

インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャーの既知の問題

- ・ ランタイム型情報 (RTTI) は未サポート

仮想共有メモリー・プログラミングでは、ランタイム型情報 (RTTI) はサポートされていません。特に、`dynamic_cast<>` と `typeid()` の使用はサポートされていません。

インテル® グラフィックス・テクノロジーへのオフロードの既知の問題

- ・ gfx_linker: : error : command 'ld.exe' exited with non-zero exit code -107374170

x64 プロジェクトでインテル® グラフィックス・テクノロジーへコードをオフロードすると、binutils に含まれている `ld.exe` でリンカーエラーが表示されることがあります。この問題を解決するには、64 ビット用の binutils bin ディレクトリーを `PATH` 環境変数に追加して、Microsoft* Visual Studio* を再起動してください。

- ・ オフロードコードのホストバージョンが並列化されない

コンパイラーは、`#pragma offload` 以下に並列ループのターゲットバージョンとホストバージョンの両方を生成します。ホストバージョンは、オフロードが実行できない場合 (通常は、ターゲットシステムにインテル® グラフィックス・テクノロジーが有効なユニットがない場合) に実行されます。並列ループは、オフロードの並列セマンティクスを含む `cilk_for` の並列構文または配列表記文を使用して指定する必要があります。ターゲットバージョンはターゲット実行で並列化されますが、現在、ホスト側のバックアップ・バージョンが並列化されない制限があります。そのため、`cilk_for` を使用する場合、オフロード実行が行われないと、バックアップ・コードの実行パフォーマンスに大きく影響する可能性があることに注意してください。配列表記文は現在ホスト側で並列コードを生成しないため、パフォーマンスに影響はありません。これは既知の問題で、将来のリリースで修正される予定です。

- ・ インテル® グラフィックス・テクノロジーへのオフロードの既知の制限事項
 - Windows* 7 では、オフロードが行われたときにディスプレイをロックできません。アクティブ・ディスプレイが必要です。
 - オフロードコードでは、次の機能を使用できません。
 - § 例外処理
 - § RTTI
 - § *longjmp/setjmp*
 - § VLA
 - § 変数引数リスト
 - § 仮想関数、関数ポインター、その他の間接呼び出しまたはジャンプ
 - § 共有仮想メモリー
 - § 配列や構造体のようなポインターを含むデータ構造
 - § ポインターまたは参照型のグローバル変数
 - § OpenMP*
 - § *cilk_spawn* または *cilk_sync*
 - § インテル® Cilk™ Plus のレデューサー
 - § ANSI C ランタイム・ライブラリー呼び出し (SVML、*math.h*、*mathimf.h* 呼び出し、およびその他いくつかの例外あり)
 - 64 ビット浮動小数点演算および整数演算は非効率

Microsoft* Windows* 10 日本語版の Microsoft Edge* でドキュメントが表示されない問題

- ・ Microsoft* Windows* 10 日本語版では、Microsoft Edge* でインテル® コンパイラー・ドキュメントを表示すると、左上の [目次]、[キーワード]、[検索] ボタンが動作しません。ボタンをクリックすると、空白ページが表示されます。
- ・ この問題は現在調査中です。回避策として、インテル® コンパイラー・ドキュメントの表示には Internet Explorer* を使用してください。デフォルトのブラウザを Internet Explorer* に設定するには、Google* で「Internet Explorer* を通常使うブラウザにする」を検索してください。

[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® C++ 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 Atom、Celeron、Intel Core、Iris、Xeon、Intel Xeon Phi、Cilk、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。

Microsoft および Windows は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。

* 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](#)

コンパイラーの最適化に関する詳細は、[最適化に関する注意事項](#)を参照してください。