

インテル® C++ コンパイラー 19.0 Update 4 for Windows* リリースノート (インテル® Parallel Studio XE 2019 Update 4)

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載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 19.0 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](#)
- [19.0 の新機能と変更された機能](#)
- [C++ STL の並列およびベクトル実行向け Parallel STL](#)
- [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

インテル® C++ コンパイラー 19.0.3 以降 (インテル® C++ コンパイラー 19.0.4 での変更)

- Microsoft* Visual Studio 2019* をサポート
- 報告された問題を修正

インテル® C++ コンパイラー 19.0.2 以降 (インテル® C++ コンパイラー 19.0.3 での変更)

- 報告された問題を修正

インテル® C++ コンパイラー 19.0.1 以降 (インテル® C++ コンパイラー 19.0.2 での変更)

- インテル® C++ コンパイラー 19.0 Update 2 には機能とセキュリティーに関する更新が含まれます。ユーザーは最新のバージョンに更新する必要があります。

インテル® C++ コンパイラー 19.0 以降 (インテル® C++ コンパイラー 19.0.1 での変更)

- [#pragma omp simd 向けの精度に影響しない simd オプション](#)
- [-\[Q\]x / -\[Q\]ax / -\[m\]tune / -\[m\]arch オプションで新しい開発コード名をサポート](#)

インテル® C++ コンパイラー 18.0 以降 (インテル® C++ コンパイラー 19.0 での変更)

- [regcall 呼び出し規約の変更](#)
- [OpenMP* parallel プラグマのユーザー定義のインダクションをサポート](#)
- [排他スキャン simd をサポート](#)
- [Visual Studio* 2017 Build Tools をサポート](#)
- [/Qopenmp-simd をデフォルトで設定](#)
- [/GS では読み出し直後に canary バイトをクリア](#)
- [/Qrcc オプションを廃止](#)
- [cannonlake オプションをサポート](#)
- [Visual Studio* IDE の変更点](#)
- [Windows* で GNU* 形式の asm キーワードをサポート](#)
- 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 /Qindirect-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緩和](#)」を参照してください。
- [新しい C++17 機能をサポート](#)
- [OpenMP* TR6 Version 5.0 Preview 2 の部分サポートを拡張](#)
- [新規および変更されたコンパイラー・オプション](#)

[先頭へ戻る](#)

動作環境

- 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)
- Microsoft* Windows* 7 Professional SP1 (将来のリリースで廃止予定)、Microsoft* Windows® 10、Microsoft* Windows Server* 2012 R2、Windows Server* 2016 (エンベデッド・エディションはサポートされていません)。Microsoft* Windows* 8.x はサポートされていません [\[1\]](#)
 - 「Windows* 8 Modern 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* 2019 Professional Edition、Enterprise Edition、Community Edition (「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* 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) [\[4\]](#)
- 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 Build Tools 2017
 - Microsoft Build Tools 2019

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、オンライン・サービス・センター (<http://www.intel.com/supporttickets> (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。

[先頭へ戻る](#)

インテル® C++ コンパイラーの使用方法

コマンドラインおよび Microsoft* Visual Studio* からのインテル® C++ コンパイラーの使用方法は、「インテル® Parallel Studio XE 2019: インテル® C++ コンパイラー 19.0 for Windows* 入門」(<install-dir>\documentation_2019\ja\compiler_c\ps2019\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_2019\ja\compiler_c\ps2019\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

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.

- [ヘルプ設定の設定] が [ブラウザーで起動] 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、ブラウザーが 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す。
- **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 2019 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。

#pragma omp simd 向けの精度に影響しない simd オプション

現在、「#pragma omp simd」は浮動小数点値と浮動小数点例外の精度に影響しない設定をオーバーライドします。次のオプションは、SIMD ループでも精度に影響しない値と例外を生成するようにこの動作を変更します。

- Qsimd-honor-fp-model[-]: 選択されている浮動小数点モデルに従って SIMD ループをベクトル化します。
- Qsimd-serialize-fp-reduction[-]: SIMD ループをベクトル化する際に浮動小数点リダクションをシリアル化します。

OpenMP* SIMD 仕様と FP モデルフラグは、この要件に関して矛盾することがあります。コンパイラーはデフォルトで OpenMP* 仕様に従ってループをベクトル化します。この新しいフラグを使用することで、プログラマーはコンパイラーが FP モデルフラグに従ってループをシリアル化するように設定をオーバーライドできます。

注 1: /Qsimd-honor-fp-model が使用され、OpenMP* SIMD リダクションがループ全体のシリアル化の唯一の原因である場合、/Qsimd-serialize-fp-reduction を使用すると、シリアル化されるリダクション処理を除くループ全体がベクトル化されます。

注 2: このオプションは、ループの自動ベクトル化には影響しません。

-[Q]x / -[Q]ax / -[m]tune / -[m]arch オプションで新しい開発コード名をサポート

サポートされた開発コード名: cascadelake, kabylake, coffeelake, amberlake, whiskeylake。

regcall 呼び出し規約の変更

-regcall コマンドライン・オプションまたは `__regcall` 関数属性を使用して以前のバージョン (インテル® C++ コンパイラー 18.0 以前) の `regcall` 呼び出し規約でビルドされたオブジェクトは、再コンパイルする必要があります。また、インテルの `libirng` ライブラリーは `regcall` を使用しています。インテル® C++ コンパイラー 19.0 に同梱の `libirng` ライブラリーは、古いバージョンと互換性がありません。さらに、次の `vector` 属性は暗黙的に `regcall` を使用するため、影響を受けます: `__attribute__((vector(clauses)))`。

Visual Studio* 2017 Build Tools をサポート

インテル® C++ コンパイラーを `compilervars` 環境で使用し、Visual Studio* 2017 IDE 全体をインストールしなくても、Visual Studio* 2017 Build Tools 内で MSBuild とインテル® C++ コンパイラー・ツールセットを使用して Visual Studio* C++ プロジェクトをビルドできるようになりました。

vector プラグマの `nodynamic_align` 節と `vectorlength` 節

- 明示的な動的アライメント構文

```
#pragma vector dynamic_align[(pointer)]
#pragma vector nodynamic_align
```

ポインターを指定しない場合、コンパイラーは通常どおりに振る舞います (アライメントする必要があるポインターを自動的に判断するか、ピールループを生成しません)。ポインターを指定すると、コンパイラーはそのポインターのピールループを生成します。`nodynamic_align` 節を指定すると、コンパイラーはピールループを生成しません。

- ```
#pragma vector vectorlength(vl1,vl2, ..., vlN)
#pragma vector vectorlength(vl1,vl2, ..., vlN)
```

ベクトライザーは、コストモデルに応じてリストから最適なベクトル長を選択します。リストに最適なベクトル長がない場合、ループはスカラーのままとなります。このプラグマはベクトル化を強制しないため、すべてのループに対して安全に使用できます。

## GNU\* 形式のインライン asm

```
__asm("nop" :); _asm("nop" :); asm("nop" :); __asm { nop }; _asm { nop }; asm
{ nop };
```

## Visual Studio\* IDE の変更点

- Visual Studio\* の状況依存ヘルプが復活

F1 を押すと、ダウンロードしたドキュメントやシステムにインストールされているドキュメントを Visual Studio\* の状況依存ヘルプで表示できます。

- /Q[a]x、-[a]x プロパティの変更点専用拡張命令セットのコード生成
- [Intel Processor-Specific Optimization (専用拡張命令セットのコード生成)] プロパティに icelake と cannonlake オプションを追加
  - 削除されたオプション: /QxSSE2、/QxSSE3、/QxCORE-AVX-1、/QxSSEE4.1
- /tune、-mtune プロパティの変更点
  - icelake と cannonlake オプションを追加
- C++ 言語標準
  - [構成プロパティ] > [C/C++] > [Language [Intel C++]] (言語 [インテル(R) C++]) から [Enable C99 Support (C99 サポートを有効にする)] (/Qstd=c99) と [Enable C++11 support (C++11 サポートを有効にする)] (/Qstd=c++11) を削除
  - Visual C++\* との互換性のため Visual Studio\* 2017 で /std オプションを利用可能に
  - Visual Studio\* のサポートするすべてのバージョンで /Qstd を利用可能に

## OpenMP\* TR6 Version 5.0 Draft の機能

[OpenMP\\* Technical Report 6: Version 5.0 Preview 2](#) (英語) 仕様の言語機能がサポートされました。

- 明示的な包括スキャン構文\*  
`#pragma omp simd reduction[parallel](inscan, operator:list)`  
`#pragma omp scan inclusive(item-list)`
- 明示的な排他スキャン構文\*  
`#pragma omp simd reduction[parallel](inscan, operator:list)`  
`#pragma omp scan exclusive(item-list)`  
プリフィクス `sum` は、ベクトル実行中に正しく計算されます。
- OpenMP\* 並列プラグマの UDI  
`#pragma omp declare induction ( induction-id : induction-type :step-  
type : inductor ) [collector( collector )]`

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。

## C++17 の機能をサポート

インテル® C++ コンパイラー 19.0 は、/Qstd=c++17(Windows\*) または -std=c++17(Linux\*/macOS\*) オプションで次の機能をサポートします。

- 畳み込み式 ([N4295](#) (英語))
- インライン変数 ([P0386R2](#) (英語))

- 列挙型クラスの構文規則 ([P0138R2](#) (英語))
- 廃止予定の動的例外仕様の削除 ([P0003R5](#) (英語))
- 例外仕様を型システムの一部にする ([P0012R1](#) (英語))
- constexpr ラムダ式 ([P0170R1](#) (英語))
- \*this のラムダ・キャプチャー ([P0018R3](#) (英語))
- if constexpr 文 ([P0292R2](#) (英語))
- 構造化バインド([P0217R3](#) (英語))
- if 文と switch 文の変数と条件を分離 ([P0305R1](#) (英語))
- 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。

## C++14 の機能をサポート

インテル® C++ コンパイラー 19.0 は、`/Qstd=c++14` (Windows\*) または `-std=c++14` (Linux\*/macOS\*) オプションで次の機能をサポートします。

- 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++14 の機能](#)」を参照してください。

## C++11 の機能をサポート

インテル® C++ コンパイラー 19.0 は、`/Qstd=c++11` (Windows\*) または `-std=c++11` (Linux\*/macOS\*) オプションで次の機能をサポートします。

- 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++11 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。

- 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 19.0 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。

- `/Qopenmp-simd` をデフォルトで設定
- `/GS` ではデフォルトで読み出し直後に canary バイトをクリア
- 新しい `/Q[a]xcannonlake` オプション
- 新しい `/tune=cannonlake` オプション
- `/Qrcd` オプションは、切り捨ての代わりに最近値への丸めを使用することで、「高速な」浮動小数点から整数への変換を可能にします。これは廃止予定のオプションです。



廃止予定のコンパイラー・オプションのリストは、『[Intel® C++ コンパイラー 19.0 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](#)

## C++ STL の並列およびベクトル実行向け Parallel STL

Intel® 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。

### 機能/API の変更点

- さらに多くのアルゴリズムで並列およびベクトル実行ポリシーをサポート: `find_first_of`、`is_heap`、`is_heap_until`、`replace`、`replace_if`。
- さらに多くのアルゴリズムでベクトル実行ポリシーをサポート: `remove`、`remove_if`。
- さらに多くのアルゴリズムで並列実行ポリシーをサポート: `partial_sort`。

詳細は、<https://software.intel.com/en-us/get-started-with-pstl> (英語) を参照してください。

## 終了予定のサポート

### Intel® Cilk™ Plus は 18.0 以降では非推奨

Intel® Cilk™ Plus は、Intel® C++ コンパイラー 18.0 以降では非推奨の古い機能です。詳細は、「[Intel® Cilk™ Plus アプリケーションを OpenMP\\* もしくは Intel® TBB へ移行する](#)」を参照してください。

## 終了したサポート

Intel® グラフィックス・テクノロジーへのオフロードサポートは終了しました。

[先頭へ戻る](#)

## 既知の制限事項

### Parallel STL

`unseq` および `par_unseq` ポリシーは、`'#pragma omp simd'` または `'#pragma simd'` をサポートするコンパイラーでのみ有効です。並列およびベクトル実行は、ランダム・アクセス・イテレーターが提供される場合にのみアルゴリズムのサブセットでサポートされ、残りはシリアル実行のままとなります。コンパイラーによっては、`zip_iterator` は `unseq` および `par_unseq` ポリシーで動作しないことがあります。

## 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\* 内のブラウザで正しく表示されない

Visual Studio\* 内のブラウザ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が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® コンパイラーの Visual Studio\* 統合をインストールした環境で、ファイル エクスプローラーからプロジェクトを開くと Visual Studio\* 2017 がフリーズすることがあります。この問題を回避するには、Visual Studio\* からプロジェクトを開きます。
- ICL/MSVC コンパイルが混在するプロジェクトでは、Visual Studio\* 2017 Update 4 以降で次の問題が発生することがあります: 「The "ConformanceMode" parameter is not supported by the "CL" task. Verify the parameter exists on the task, and it is a settable public instance property. ("ConformanceMode" パラメーターは "CL" タスクではサポートされていません。タスクにパラメーターが存在し、設定可能なパブリック・インスタンス・プロパティであることを確認します。)」

この問題を回避するには、Visual Studio\* 2017 を更新するか、`<vs2017installdir>\Common7\IDE\VC\VCTargets\Platforms\<platform>\PlatformToolsets\Intel`

C++ Compiler 19.0\Toolset.targets ファイルの 2 つの <CL> セクションにある「ConformanceMode = "%(ClCompile.ConformanceMode) "」行をコメントアウトします。

- *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 (*mispdb110.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 (*mispdb110.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) 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 *mispdb\*.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、コンパイラは .pdb ファイルを生成しません。

## 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・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・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られません。詳細については、[http://www.intel.co.jp/jp/products/processor\\_number/](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、Cilk 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 またはその子会社の商標です。

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。

© 2019 Intel Corporation. 無断での引用、転載を禁じます。

[先頭へ戻る](#)