

インテル® Visual Fortran コンパイラー 17.0 Update 4 for Windows® リリースノート (インテル® Parallel Studio XE 2017)

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® Visual Fortran コンパイラー 17.0 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](#)
- [新規および変更されたコンパイラー機能](#)
- [新規および変更されたコンパイラー・オプション](#)
- [Microsoft® Visual Studio® 統合の変更](#)
- [終了予定のサポート](#)
- [終了したサポート](#)
- [既知の問題](#)
- [Microsoft® Visual Studio® 2012/2013/2015/2017 に関する注意事項](#)
- [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 4 (インテル® Visual Fortran コンパイラー 17.0.4)

- [Microsoft® Visual Studio® 2017 をサポート](#)
- [日本語版を更新](#)
- [PARAMETER 定数と /debug-parameters /debug コンパイラー・オプションに関連した内部コンパイラー・エラーの修正](#)
- [報告された問題を修正](#)

[先頭へ戻る](#)

Update 3 - リリースなし

- Windows® 向けの Update 3 はリリースなし

[先頭へ戻る](#)

Update 2 (インテル® Visual Fortran コンパイラー 17.0.2)

- [DIR\\$ VECTOR \[NO\] MASK_READWRITE](#)
- 報告された問題を修正

[先頭へ戻る](#)

Update 1 (インテル® Visual Fortran コンパイラー 17.0.1)

- [日本語版を含む最初のアップデート。](#)
- [OpenMP* 監視スレッド](#)
- 報告された問題を修正

[先頭へ戻る](#)

インテル® Visual Fortran コンパイラー 16.0 以降 (インテル® Visual Fortran コンパイラー 17.0.0 での変更)

- インテル® Visual Fortran コンパイラーが 17.0.0 にアップデート
- [OpenMP* 4.5 のディレクティブ](#)
- コンパイラーによる最適化レポートをソースリストに追加する新しいオプション
- 関数のコード・アライメントを要求する `ATTRIBUTES code_align(n)` を追加
- 割付け配列に対する組込み代入のデフォルト動作の変更
- 論理/数値演算が混在する場合の動作を変更
- [OpenMP* 4.5 によるローカルスカラー変数のデフォルトのオフロード動作の変更](#)
- [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](#)

製品の内容

インテル® Visual Fortran コンパイラー 17.0 for Windows® には、次のコンポーネントが含まれます。

- インテル® Visual Fortran コンパイラー 17.0。IA-32 およびインテル® 64 アーキテクチャー・システムで動作するアプリケーションをビルドします。
- Microsoft® Visual Studio® で Fortran アプリケーションをデバッグするための Fortran Expression Evaluator (FEE)
- Microsoft® 開発環境への統合
- Microsoft® Visual Studio® 2013 Shell とライブラリー (評価版ライセンスでは提供されません)
- サンプルプログラム
- 各種ドキュメント

[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)
 - 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 (すべての機能およびすべてのアーキテクチャー)
- 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開発/テスト:
 - インテル® Xeon Phi™ プロセッサ (開発コード名 Knights Ferry、開発コード名 Knights Corner)
 - [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 \(インテル® MPSS\)](#)
 - オフロードコードのデバッグには Microsoft® Visual Studio® 2012 以降が必要
- Microsoft® Windows® 7、Microsoft® Windows® 8、Microsoft® Windows® 8.1、Microsoft® Windows® 10、Microsoft® Windows Server® 2012 (R2) (インテル® 64 のみ)、Microsoft® Windows Server® 2008 (R2 SP1) (インテル® 64 のみ)、Microsoft® Windows® HPC Server 2008 (R2) (インテル® 64 のみ) (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 Server® 2008 または Windows® HPC Server 2008 では、Microsoft® Visual Studio® 2015、Visual Studio® 2013、Visual Studio® 2012、または Visual Studio® 2013 Shell が必要です。
 - 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。
- IA-32 対応アプリケーション [\[3\]](#) 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) [\[6\]](#)
 - 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) [\[5\]](#)
 - 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) [\[5\]](#)
 - Microsoft® Visual Studio® 2013 Professional Edition 以上
 - Microsoft® Visual Studio® Community 2013
 - Microsoft® Visual Studio® 2012 Professional Edition 以上
 - Microsoft® Visual Studio® 2013 Shell (インテル® Visual Fortran コンパイラーの特定のライセンスに付属) ベースのインテル® Visual Fortran 開発環境 [\[1\]\[2\]](#)
- IA-32 [\[4\]](#) 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
 - Microsoft® Visual C++® Express 2012 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® Visual C++® Express 2012 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. Microsoft® Visual Studio® 2013 Shell ベースのインテル® Visual Fortran 開発環境は、インテル® Parallel Studio XE Composer Edition for Fortran Windows® のアカデミック・ライセンスと商用ライセンスに含まれます。評価版ライセンスや学生および教育関係者向けの無料ソフトウェア・プログラムで提供される「無料ツール」ライセンスには含まれません。この開発環境は、Fortran アプリケーションの編集、ビルド、デバッグに必要なものがすべて揃っています。ただし、次のような、Visual Studio® 製品の一部の機能は含まれていません。
 - リソースエディター (代用としてサードパーティー・ツールの ResEdit* (<http://www.resedit.net/> (英語)) を参照してください。)
 - Compaq* Visual Fortran プロジェクトの自動変換
 - Visual C++® や Visual Basic® などの Microsoft® の言語ツール
2. Microsoft® Visual Studio® 2013 Shell ベースのシステムにインテル® Visual Fortran 開発環境をインストールすると、Microsoft® Visual Studio® Shell は実行に必要な Microsoft® コンポーネント (.NET Framework など) がシステムに含まれているかどうか確認し、不足しているコンポーネントを自動的にダウンロードしてインストールします。
3. インテル® Visual Fortran コンパイラーは、デフォルトで、インテル® SSE2 命令対応のプロセッサが必要な IA-32 アーキテクチャー向けのアプリケーションをビルドします。コンパイラー・オプションを使用して任意の IA-32 アーキテクチャー・ベースのプロセッサ上で動作するコードを生成できます。ただし、インテル® マス・カーネル・ライブラリー (インテル® MKL) を呼び出すアプリケーションでは、インテル® SSE2 命令に対応しているプロセッサが必要です。
4. 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® 7 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 Windows® 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。
5. インテル® Visual Fortran 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。
6. インテル® Visual Fortran コンパイラーを Microsoft® Visual Studio® 2017 で使用するには、Visual Studio® から「C++ によるデスクトップ開発」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。

[先頭へ戻る](#)

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® Xeon Phi™ コプロセッサ向けのアプリケーションを開発する場合、インテル® Visual Fortran 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、インテル® MPSS のドキュメントを参照してください。

[先頭へ戻る](#)

インテル® Visual Fortran 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® Visual Fortran コンパイラーの使用方法についての情報は、「インテル® Visual Fortran コンパイラー for Windows® ガイド」(<install-dir>\IntelSWTools\documentation_2017\ja\ps2017\getstart_comp_wf.htm) に含まれています。

[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>\IntelSWTools\documentation_2017\ja\ps2017\getstart_comp_wf.htm にあります。

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。

Windows® ベースのアプリケーションの作成についてのドキュメントは Web から入手可能

Windows® ベースのアプリケーションの作成についてのドキュメントは、Intel® Software Documentation Library Web サイトから入手できます。「[インテル® Visual Fortran を使用した Windows® ベースのアプリケーションの作成とビルド](#)」(英語) を参照してください。

Windows Server® 2012 の Microsoft® Internet Explorer® 10 でドキュメントが表示されない問題

Windows Server® 2012 では、Internet Explorer® 10 でヘルプまたはドキュメントを表示できない場合、Microsoft® Internet Explorer® のセキュリティ設定を変更すると表示されるようになります。[ツール] > [インターネット オプション] > [セキュリティ] を選択して、信頼済みサイトのリストに "about:internet" を追加します。必要に応じて、ドキュメントを参照した後に信頼済みサイトのリストから "about:internet" を削除します。

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、将来のリリースで修正される予定です。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。

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。

[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 2017 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](#)

日本語のサポート

日本語対応のインテル® コンパイラーをインストールした場合、オプションで日本語のサポートが提供されます。エラーメッセージ、仮想開発環境のダイアログ、一部のドキュメントが (英語に加えて) 日本語で提供されます。デフォルトでは、エラーメッセージとダイアログの言語はオペレーティング・システムの言語で表示されます。日本語ドキュメントは、ドキュメントの **ja** サブディレクトリーに含まれています。

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](#)

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](#)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® Visual Fortran コンパイラー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 17.0 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ク間の最適化 (/Qipo) オプションを使用して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。
- 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 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。

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

以前のリリースでは、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 12.0 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 (組み込み関数を含む) 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 13

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。

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。

インテルによる OpenMP* スタティック・ライブラリーの提供を終了

インテルによる OpenMP* スタティック・ライブラリー libiomp5mt.lib の提供が終了し、/Qopenmp-link:static コマンドライン・オプションがサポートされなくなりました。libiomp5mt.lib に対するすべての参照を、DLL インポート・ライブラリー libiomp5md.lib に変更してください。この変更に伴い、OpenMP* を使用するアプリケーションを、インテル® コンパイラーが存在しないシステムに配布する場合、インテル® コンパイラーの再配布可能コードのインストールが必要になります。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。

Fortran Expression Evaluator

Fortran Expression Evaluator (FEE) は、インテル® Visual Fortran コンパイラーとともにインストールされる Microsoft® Visual Studio® のプラグインです。Fortran コードを処理できるように、Microsoft® Visual Studio® IDE の標準デバッガーを拡張します。その他は標準デバッガーと同じです。

[先頭へ戻る](#)

新規および変更されたコンパイラー機能

一部の言語機能に関する説明はコンパイラーのドキュメントにはまだ含まれていません。必要に応じて、[Fortran 2008 Standard](#) (PDF、英語) および [Proposed draft Fortran 2015 Standard](#) を参照してください。

Fortran 2008 の機能

- 組込み型の型宣言
- ポインターの初期化
- 暗黙形状配列
- EXIT 文の構文名の拡張
- 内部プロシージャーの BIND(C) のサポート

割付け配列に対する組込み代入のデフォルト動作の変更 (17.0)

以前のリリースでは、コンパイラーはデフォルトで、割付け配列に対する組込み代入では、代入される配列は値と同じ形状に割付け済みであると仮定していました。形状が一致しない場合に Fortran 2003 の自動 (再) 割付けを行うには、/assume:realloc_lhs コンパイラー・オプション (/standard-semantics に含まれる) を指定する必要がありました。

コンパイラー 17.0 では、デフォルトの動作が Fortran 2003 標準と一致するように変更され、必要であれば割付け配列は組込み代入で自動的に (再) 割付けされるようになりました。この変更はパフォーマンスに多少影響します。古い動作に戻す場合は、/assume:norealloc_lhs または新しい /nostandard-realloc_lhs コンパイラー・オプションを指定します。

論理/数値演算が混在する場合の動作の変更 (17.0)

Fortran 標準規格は、論理データ型と数値 (integer/real/complex) データ型が混在する代入および演算を禁止しています。標準規格の拡張として、インテル® Visual Fortran コンパイラーではこの混在を許可していますが、これまで動作規則の文書化が不十分で、コンテキストに依存して(一貫しない)実装が行われていました。

バージョン 17.0 では、論理/数値演算が混在する場合のコンパイラーの実装が一貫するように変更されました。場合によっては、新しい動作が以前のバージョンの実装と異なるため、以前の動作が正しいと仮定しているプログラムでは結果が変わる可能性があります。

バージョン 17.0 の動作は次のとおりです。

- 論理値が数値コンテキストで使用された場合、"fpscomp logicals" コンパイラー・オプションの設定に応じて、true 値は整数 -1 または +1 に変換されます。false 値は整数 0 に変換されます。
- 数値が論理コンテキストで使用された場合、最初に整数に変換されます (必要な場合)。そして、"fpscomp logicals" コンパイラー・オプションの設定に応じて、整数値をどのように扱うかが決まります。"nologicals" が有効な場合、奇数値は true、偶数値は false として扱われます。"logicals" が有効な場合、非ゼロ値は true、ゼロ値は false として扱われます。
- 数値が論理変数に代入された場合、"fpscomp logicals" の設定に応じて値が .TRUE. または .FALSE. に変換され、新しい論理値が代入されます。以前のバージョンでは、バイナリー値が変換されずに直接コピーされることがありました。
- 論理値が数値変数に代入された場合、上記のように、最初に整数に変換された後、組み込み代入の通常の規則が適用されます。

インテル® Visual Fortran コンパイラーのデフォルトは "fpscomp nologicals" ですが、"standard- semantics" オプションを指定すると "fpscomp logicals" が設定されることに注意してください。

プログラムがこの拡張による影響を受けるかどうか確認するには、標準警告を有効にして (/stand) プログラムをビルドします。古い動作に戻す場合は、/assume:old_logical_assign を指定します。

PARAMETER 定数と /debug-parameters および /debug コンパイラー・オプションに関連した内部コンパイラー・エラーの修正 (17.0 Update 4)

17.0 Update 4 で PARAMETER 定数と /debug-parameters および /debug コンパイラー・オプションに関連した内部コンパイラー・エラー (内部トラッキング ID: DPD200418283) が修正されました。

詳細は、「[PARAMETER 定数と /debug-parameters /debug コンパイラー・オプションに関連した内部コンパイラー・エラー](#)」を参照してください。

OpenMP* 4.5 によるローカルスカラー変数のデフォルトのオフロード動作の変更

OpenMP* 4.5 の DEFAULTMAP (TOFROM:SCALAR) 節のサポートにより、ローカルスカラー変数のデフォルトの動作が変わりました。以前のリリースでは、ローカルスカラー変数はデフォルトでオフロードされました。17.0 では、ローカルスカラー変数はマップされず、代わりに暗黙的に FIRSTPRIVATE 属性が指定されます。17.0 でローカルスカラー変数をオフロードするには、DEFAULTMAP (TOFROM:SCALAR) 節を使用する必要があります。詳細は、『インテル® Fortran 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。

OpenMP* 監視スレッド

バージョン 17.0.1 では、OpenMP* 監視スレッド (ヘルパースレッド) が生成されません。

OpenMP* 機能

OpenMP* 4.0 (英語) および OpenMP* 4.5 (英語) の次のディレクティブ、節、プロシージャーがコンパイラーでサポートされました。

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* 4.5 のディレクティブ:

- TARGET ENTER DATA
- TARGET EXIT DATA
- TASKLOOP

節:

- OMP TARGET および OMP TARGET UPDATE ディレクティブの DEPEND
- OMP TARGET および OMP TARGET UPDATE ディレクティブの NOWAIT
- OMP SIMD ディレクティブの SIMDLLEN
- OMP ORDERED ディレクティブの SIMD
- OMP DECLARE SIMD (proc-name) ディレクティブの PROCESSOR(cpuid)
- OMP TARGET ディレクティブの DEFAULTMAP (TOFROM:SCALAR)

processor 節を OMP DECLARE SIMD に追加

インテル® Visual Fortran コンパイラー 17.0 には、プログラマーが OpenMP* SIMD で YMM/ZMM レジスターを使用できる、OMP DECLARE SIMD の拡張が含まれています。PROCESSOR(cpuid) 節は、指定したプロセッサ向けのルーチンのベクトルバージョンを生成するようにコンパイラーに指示します。詳細は、『インテル® Fortran 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。

!\$OMP DO SCHEDULE 節の SIMD 修飾子と NONMONOTONIC 修飾子

インテル® Visual Fortran コンパイラー 17.0 には、DO 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 OMP DO SCHEDULE 節の新しい 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® Fortran 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。

OpenMP* 4.5 で定義されている taskloop および do across ループのサポート

インテル® Visual Fortran コンパイラー 17.0 では、for/do ループを並列化する新しいループ構造がサポートされました。"taskloop" は cilk_for ループに似ていて、インテルのタスク実行モデルの下で動的な分割統治ループ分割を可能にします。"doacross" は、ループ伝播の依存があるループの並列化を可能にします。

新しいインテル® Xeon Phi™ プロセッサ/コプロセッサへのオフロード機能

- OpenMP* 4.5 節の変更
 - 結合構造または複合構造の場合、if 節でディレクティブ名修飾子をサポート
 - `if[[directive-name-modifier :] scalar-logical-expression]` 構造が `directive-name-modifier` で指定された場合、if 節はその構造のセマンティクスにのみ適用されます。その他の場合、if 節を適用できるすべての構造に適用されます。
例: `!$omp target parallel for if(target : do_offload_compute)`
 - `use_device_ptr(list)` 節を `!$omp target data` に実装
 - `is_device_ptr(list)` 節を `!$omp target` に実装
- 結合 `target` 構造のサポート
 - `!$omp target parallel`
 - `!$omp target parallel for`
 - `!$omp target simd`
 - `!$omp target parallel for simd`

omp declare simd linear 節の新しい修飾子

omp declare simd ディレクティブの `linear` 節を新しい修飾子で拡張

linear (*linear-list* [: *linear-step*])

linear-list は次のいずれかです。

list

modifier (*list*)

modifier は次のいずれかです。

ref

val

uval

- すべての *list* 項目は各 SIMD レーンで同時に呼び出される関数の仮引数でなければなりません。
- *modifier* を指定しない場合や **val** または **uval** 修飾子を指定した場合、各レーンの各 *list* 項目の値は、関数に入るとき *list* 項目の値 + レーンの論理番号 × *linear-step* に相当します。
- **uval** 修飾子を指定した場合、各呼び出しは各 SIMD レーンと同じメモリー位置を使用します。このメモリー位置は論理的な最終レーンの最後の値で更新されます。
- **ref** 修飾子を指定した場合、各レーンの各 *list* 項目のメモリー位置は、レーンの論理番号 × *linear-step* でインデックスされた関数に入るときメモリー位置の配列に相当します。

新しいディレクティブと追加されたディレクティブ

ATTRIBUTES `code_align(n)`

コンパイラー 17.0 では、関数のコード・アライメントを要求する `ATTRIBUTES code_align(n)` ディレクティブを指定することができます。"n" は最小アライメント境界 (バイト単位) で、1 から 4096 の範囲の 2 の累乗の値でなければなりません (例: 1、2、4、8、16、32、64、128 など)。n = 1 はアライメントを行いません。n は必ず指定する必要があります。

PROCESSOR(cpuid) 節の拡張

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) 基本命令、競合検出命令、指数および逆数命令、プリフェッチ命令、および RDSEED および ADX (Multi-Precision Add-Carry Instruction Extensions) 命令を含むインテル® アドバンスド・ベクトル・エクステンション 2 (インテル® AVX2) 対応第 2 世代インテル® Xeon Phi™ プロセッサ・ファミリー向けに、新しい *cpuid* キーワード "mic_avx512" を追加しました。新しい *cpuid* キーワードの使用方法は、『インテル® Fortran コンパイラー 17.0 デベロッパ・ガイドおよびリファレンス』を参照してください。

DIR\$ VECTOR [NO] MASK_READWRITE

ベクトライザーによるメモリー・スペキュレーションを有効/無効にする DIR\$ VECTOR [NO] MASK_READWRITE 節をサポートします。

MASK_READWRITE | NOMASK_READWRITE

条件文内でのマスク付きロード/ストア操作の生成を有効/無効にします。

MASK_READWRITE 節は、メモリーのスペキュレーションを無効にし、条件文内でマスク付きロード/ストア操作を生成するようにコンパイラーに指示します。

NOMASK_READWRITE 節は、メモリーのスペキュレーションを有効にし、条件文内でマスクなしロード/ストア操作を生成するようにコンパイラーに指示します。

[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。

- [/assume: \[no\]old_logical_assign](#)
- [/fp:consistent](#)
- [/Qalign-loops\[:n\]](#)
- [/Qopt-report-annotate \[: text | html \]](#)
- [/Qopt-report-annotate-position: \[caller | callee | both \]](#)
- [/\[no\]standard-realloc-lhs](#)
- [/Zo\[-\]](#)

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。

新しい /Qalign-loops[:n]、/Qalign-loops- コンパイラー・オプション

この新しいオプションは、2 の累乗のバイト境界でループをアライメントします。

新しい /Qopt-report-annotate [: text | html] コンパイラー・オプション

このオプションは、アノテーション付きソースリスト機能を有効にし、その形式 (text または html) を指定します。デフォルト形式は text です。

新しい /Qopt-report-annotate-position: [caller | callee | both] コンパイラー・オプション

このオプションは、最適化メッセージを表示するアノテーション付きソースの位置を指定します。値は、"caller"、"callee" または "both" です。デフォルト値は caller です。

新しい /fp:consistent コンパイラー・オプション

このオプションを指定すると、ドライバーはより一貫した浮動小数点結果を提供するオプションの組み合わせをセットします。

新しい /Zo[-] コンパイラー・オプション

このオプションは、Windows® の最適化コードのデバッグサポートを向上します。

新しい /assume: [no]old_logical_assign コンパイラー・オプション

このオプションは、論理値を数値変数へ代入する代入文と、数値を論理変数へ代入する代入文の動作を指定します。「[論理/数値演算が混在する場合の動作の変更](#)」も参照してください。

新しい /[no]standard-realloc-lhs コンパイラー・オプション

このオプションは、/assume:[no]realloc-lhs の代替形式です。「[割付け配列に対する組み込み代入のデフォルト動作の変更](#)」も参照してください。

[先頭へ戻る](#)

Visual Studio® 統合の変更点

Microsoft® Visual Studio® 2017 をサポート

インテル® Visual Fortran 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。

Microsoft® Visual Studio® IDE でソースコードのナビゲーションが向上

モジュール/プロシージャーを容易に参照できるように、Visual Studio® IDE に (ソリューション エクスプローラー ビューに似た) "ツリービュー" が追加されました。詳細は、コンパイラー・ドキュメントを参照してください。

Microsoft® Visual Studio® のオンラインヘルプ形式の変更

オンラインヘルプ形式がブラウザーベースになりました。Microsoft® Visual Studio® の [ヘルプ] メニューからインテルのドキュメントを参照する場合、または F1 キー、ダイアログボックスにあるヘルプボタン、その他の GUI で状況依存ヘルプを参照する場合、デフォルトのブラウザーに対応するヘルプトピックが表示されます。デフォルトのブラウザーによっては、次のような既知の問題があります。

- [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。

既存のコードからインテル® Fortran プロジェクトを新規作成

Visual Studio® で [ファイル] > [新規作成] > [Fortran Project From Existing Code (既存のコードからインテル(R) Fortran プロジェクトを作成)] を利用できるようになりました。選択したフォルダーからソースを追加した新しい Fortran プロジェクトが作成されます。プロジェクト・ウィザードでプロジェクトの種類とプラットフォームをカスタマイズできます。

[先頭へ戻る](#)

終了予定のサポート

Microsoft® Visual Studio® 2012 のサポート終了予定

Microsoft® Visual Studio® 2012 のサポートは、将来のリリースで終了する予定です。

[先頭へ戻る](#)

終了したサポート

IA-32 ホスト・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。

Microsoft® Visual Studio® 2010 のサポートを終了

Microsoft® Visual Studio® 2010 のサポートを終了しました。

[先頭へ戻る](#)

既知の問題

日本語ファイル名に関するコマンドライン診断表示の問題

コンパイル診断で日本語が含まれているファイル名は、ネイティブのインテル® 64 対応アプリケーション用コンパイラを使用して、Windows® コマンドラインでコンパイルした場合に正しく表示されません。Visual Studio® を使用する場合やインテル® 64 対応アプリケーション用クロスコンパイラまたは IA-32 対応アプリケーション用コンパイラを使用する場合は、この問題は発生しません。

Fortran を含む言語が混在したプログラムがデバッグできない

Visual Studio® 2012 以降で、.NET マネージ・コード・アプリケーションから呼び出される Fortran コードのデバッグを有効にするには、[ツール] > [オプション] > [デバッグ] > [全般] で [Managed C++ 互換モード] チェックボックスをオフにします。

マネージ・コード・アプリケーションの場合は、プロジェクト・プロパティにある [デバッグ] > [アンマネージ コード デバッグを有効にする] も確認します。

パラメーター化された派生型で文字長引数の特定の使用方法が完全に実装されていない

パラメーター化された派生型 (PDT) では、文字長引数の次の使用法は完全に実装されていません。

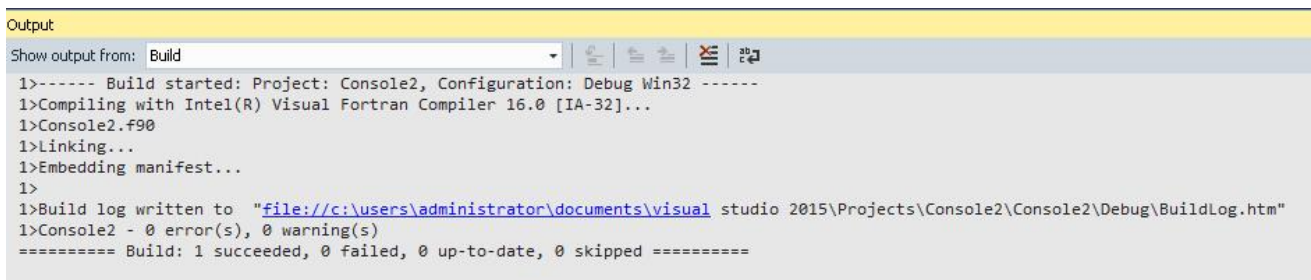
- 文字長引数を含む PDT 引数定数
- %RE と %IM は未実装

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

インテル® Parallel Studio XE 2016 を日本語版 Windows® にインストールすると、Microsoft® Visual Studio® 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ことがあります。この問題の詳細は、[こちら](#) (英語) を参照してください。

Microsoft® Visual Studio® 2015 Update 1 IDE 統合で BuildLog.htm へのリンクが正しく表示されない問題

インテル® Visual Fortran コンパイラー IDE の Visual Studio® 2015 Update 1 統合で、出力ウィンドウのビルドログのリンクが正しく表示されない問題があります。次の画面のように、パスの最初のスペースより前の部分のみがリンクに含まれます。



```
Output
Show output from: Build
1>----- Build started: Project: Console2, Configuration: Debug Win32 -----
1>Compiling with Intel(R) Visual Fortran Compiler 16.0 [IA-32]...
1>Console2.f90
1>Linking...
1>Embedding manifest...
1>
1>Build log written to "file://c:\users\administrator\documents\visual studio 2015\Projects\Console2\Console2\Debug\BuildLog.htm"
1>Console2 - 0 error(s), 0 warning(s)
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

この問題を回避するには、出力ウィンドウからリンクをコピーし、Web ブラウザーを使用してリンクを開いてください。

PARAMETER 定数と /debug-parameters および /debug コンパイラー・オプションに関連した内部コンパイラー・エラー

(Update 4 で修正済み) PARAMETER 定数と /debug-parameters および /debug コンパイラー・オプションを一緒に使用すると、内部コンパイラー・エラーが発生することがあります。

以下に、内部エラーが発生する使用例を示します。

```
module sample_mod
  implicit none
  integer, parameter :: isize=32

  contains
```

```

subroutine example1
  call example2(ysize)
end subroutine example1

subroutine example2(jsize)
  integer, intent(in) :: jsize
  write(*,*) jsize
end subroutine example2

end module sample_mod

```

Visual Studio® の Debug 構成のビルドでは、/debug-parameter オプションと /debug オプションが一緒に指定されることが多く、ビルド中にいくつかの内部エラーが発生することがあります。

この問題は、次のいずれか方法で回避できます。

1. Visual Studio® では、[プロパティ] > [Fortran] > [Debugging (デバッグ)] > [Information for PARAMETER Constants (PARAMETER 定数の情報)] を [None (なし)] に設定して、/debug-parameters を無効にします。
2. Visual Studio® では、任意の /debug-parameters 設定 (上記の 1 の手順を参照) と一緒に使用する場合、[プロパティ] > [Fortran] > [Debugging (デバッグ)] > [Debug Information Format (デバッグ情報の形式)] を [Line Numbers Only (行番号のみ)] に設定して、/debug:minimal 設定を使用します。
3. コンパイラー・コマンドラインでは、/debug:full を使用する場合、コマンドラインまたはアプリケーションのビルドスクリプトからすべての /debug-parameters を削除します。
4. コンパイラー・コマンドラインでは、任意の /debug-parameters 設定と一緒に指定する場合、コマンドラインまたはアプリケーションのビルドスクリプトで /debug:minimal を使用します。

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。

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

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® Visual 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](#)

Microsoft® Visual Studio® 2012/2013/2015/2017 に関する注意事項

Microsoft® Visual Studio® 2010 ではいくつかの変更がありました。そのほとんどは、メインプログラムが C/C++ の言語が混在したアプリケーションのビルドに影響するものです。これらの変更は、Visual Studio® 2012/2013/2015/2017 にも適用されます。

インテル® Fortran ランタイム・ライブラリーを参照するための Microsoft® Visual C++® の設定

以前のリリースでは、インテル® Visual Fortran コンパイラーの LIB フォルダを C/C++ プロジェクトで利用できるようにするために [ツール] > [オプション] > [プロジェクトおよびソリューション] > [Visual C++ ディレクトリ] で設定を行っていました。Visual Studio® 2010 以降では、この方法が変更されています。

1. Visual Studio® で C++ プロジェクトを含むソリューションを開き、[表示] > [プロパティ マネージャー] を選択します。[表示] メニューの直下に [プロパティ マネージャー] が見つからない場合は、[表示] > [その他のウィンドウ] の下にあります。[プロパティ マネージャー] ダイアログボックスが表示されます。これは、[プロパティ] ウィンドウや [プロパティ ページ] とは関係ありません。
2. プロパティ・ツリーの Debug | Win32 の横にある三角または + 記号をクリックしてこのフォルダを展開します。
3. Microsoft.Cpp.Win32.user をダブルクリックします。
4. [VC++ ディレクトリ] を選択します。
5. [ライブラリ ディレクトリ] の右側のフィールドをクリックします。
6. ドロップダウンから <編集...> を選択します。
7. [新しい行] ボタンをクリックするか、Ctrl+Insert キーを押します。

8. 表示された新しいフィールドに、次のように入力します。

```
$(IFORT_COMPILER17)\compiler\lib\ia32
```

9. [OK] をクリックします。もう一度 [OK] をクリックして、[プロパティ ページ] も閉じます。
10. Visual Studio® のメニューから [ファイル] > [すべてを保存] を選択します。

インテル® 64 (x64) 構成でビルドする場合は、次の手順を実行してください。

1. [プロパティ マネージャー] を開いて、Debug | x64 フォルダーを展開します。
2. Microsoft.Cpp.x64.user をダブルクリックします。
3. [VC++ ディレクトリ] を選択します。
4. [ライブラリ ディレクトリ] の右側のフィールドをクリックします。
5. ドロップダウンから <編集...> を選択します。
6. [新しい行] ボタンをクリックするか、Ctrl+Insert キーを押します。
7. 表示された新しいフィールドに、次のように入力します。

```
$(IFORT_COMPILER17)\compiler\lib\intel64
```

8. [OK] をクリックします。もう一度 [OK] をクリックして、[プロパティ ページ] も閉じます。
9. Visual Studio® のメニューから [ファイル] > [すべてを保存] を選択します。

[ソリューション エクスプローラー] タブをクリックするか、Ctrl+Alt+L キーを押して [ソリューション エクスプローラー] を表示します。

Debug | x64 フォルダーに Microsoft.Cpp.x64.user プロパティ・ページが見つからない場合は、フォルダーを右クリックして [新しいプロジェクト プロパティ シートの追加] を選択します。そして、MsBuild 4.0 プロパティ・ページの場所を参照します。Windows® 7 および Windows® 8 では、通常以下の場所にあります。

```
C:\Users\\AppData\Local\Microsoft\MSBuild\v4.0
```

これらのパスを表示するためには、隠しファイルと隠しフォルダーの表示を有効にする必要があります。

Microsoft.Cpp.x64.user.props を選択して [開く] をクリックします。後は、上記の手順に従ってください。

プロジェクトの依存関係の調整

以前のバージョンの Visual Studio® から依存関係が設定されているプロジェクトを変換する場合、既存のプロジェクトの依存関係は Visual Studio® 2012/2013/2015/2017 によって参照に変換されます。C/C++ プロジェクトで Fortran プロジェクトを参照している場合、C/C++ プロジェクトのビルドで MSB4075 エラーが発生することがあります。この問題を解決するには、次の操作を行います。

1. C/C++ プロジェクトを右クリックして、[参照] を選択します。
2. 参照リストに Fortran プロジェクトがある場合は、プロジェクトを選択してから [参照の削除] をクリックします。参照リストにあるすべての Fortran プロジェクトに対してこの操作を行います。[OK] をクリックします。
3. ほかの C/C++ プロジェクトでも上記の手順を実行します。

これにより、プロジェクトの依存関係が更新されます。

1. C/C++ プロジェクトを右クリックして、[プロジェクトの依存関係] を選択します。(Visual Studio® 2013 では、[ビルド依存関係] > [プロジェクト依存関係] を選択します。)
2. このプロジェクトと依存関係のあるプロジェクトのチェックボックスをすべてオンにします。
3. [OK] をクリックします。
4. 依存関係のあるほかの C/C++ プロジェクトでも上記の手順を実行します。

以前のバージョンの Visual Studio® とは異なり、Visual Studio® 2012 は依存関係のあるプロジェクトの出力ライブラリーを自動でリンクしません。そのため、親プロジェクトのプロパティ・ページで [Linker (リンカー)] > [Additional Directories (追加のライブラリー・ディレクトリー)] からこれらのライブラリーを明示的に追加する必要があります。必要に応じて、Visual Studio® のマクロである \$(ConfigurationName) と \$(PlatformName) を使用してパスを指定することができます。次に例を示します。

```
..\FLIB\$(ConfigurationName)\FLIB.lib
```

\$(ConfigurationName) は Release または Debug に置換されます。同様に、\$(PlatformName) は Win32 または x64 に置換されます。

Windows Server® 2012 と Windows Server® 2008 で Visual Studio® 2012 のドキュメントが表示されない問題

Windows Server® 2012 と Windows Server® 2008 では、Visual Studio® 2012 のヘルプまたはドキュメントを表示できない場合、Microsoft® Internet Explorer® のセキュリティ設定を変更すると表示されるようになります。[ツール] > [インターネット オプション] > [セキュリティ] を選択して、[インターネット] ゾーンで [MIME スニффイングを有効にする] および [アクティブ スクリプト] を有効にします。

Microsoft® Visual Studio® IDE からインテル® Parallel Studio XE のドキュメントを開くと「この種類のリンク (ファイル) を開くアプリがインストールされていません」エラーになる問題

.htm* ファイルの規定のアプリにブラウザーが設定されていない場合、Microsoft® Visual Studio® IDE で [ヘルプ] > [Intel Compilers and Libraries (インテル(R) コンパイラーおよびライブラリー)] からドキュメントを開くと、「この種類のリンク (ファイル) を開くアプリがインストールされていません」エラーが表示されます。

この問題を解決するには、[コントロール パネル] > [表示方法:] を [大きいアイコン] または [小さいアイコン] に設定し、[既定のプログラム] > [ファイルの種類またはプロトコルのプログラムへの関連付け] または [規定のプログラムの設定] して .htm ファイルを任意のブラウザーに関連付けます。または、[設定] > [システム] > [規定のアプリ] > [規定のアプリの選択] で .htm ファイルの規定のアプリとして任意のブラウザーを選択します。

[先頭へ戻る](#)

Fortran 2008 および Fortran 2015 機能の概要

インテル® Visual Fortran コンパイラーは、Fortran 2008 標準規格の多くの機能と Proposed draft Fortran 2015 標準規格の一部の機能をサポートします。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。現在のコンパイラーでは、以下の Fortran 2008 機能がサポートされています。

- 配列の最大次元数が 31 次元に (Fortran 2008 では 15 次元)
- Co-Array
 - CODIMENSION 属性
 - SYNC ALL 文
 - SYNC IMAGES 文
 - SYNC MEMORY 文
 - CRITICAL および END CRITICAL 文
 - LOCK および UNLOCK 文
 - ERROR STOP 文
 - ALLOCATE および DEALLOCATE で Co-Array を指定
 - 組み込みプロシージャ: ATOMIC_DEFINE、ATOMIC_REF、IMAGE_INDEX、LCBOUND、NUM_IMAGES、THIS_IMAGE、UCBOUND
- CONTIGUOUS 属性
- ALLOCATE の MOLD キーワード
- DO CONCURRENT
- OPEN の NEWUNIT キーワード
- GO および GO.d フォーマット編集記述子
- 無制限のフォーマット項目繰り返しカウント指定子
- CONTAINS セクションは空にすることも可能
- 組み込みプロシージャ: BESSEL_J0、BESSEL_J1、BESSEL_JN、BESSEL_YN、BGE、BGT、BLE、BLT、DSHIFTL、DSHIFTR、ERF、ERFC、ERFC_SCALED、GAMMA、HYPOT、IALL、IANY、IPARITY、IS_CONTIGUOUS、LEADZ、LOG_GAMMA、MASKL、MASKR、MERGE_BITS、NORM2、PARITY、POPCNT、POPPAR、SHIFTA、SHIFTL、SHIFTR、STORAGE_SIZE、TRAILZ
- 組み込みモジュール ISO_FORTRAN_ENV の追加: ATOMIC_INT_KIND、ATOMIC_LOGICAL_KIND、CHARACTER_KINDS、INTEGER_KINDS、INT8、INT16、INT32、INT64、LOCK_TYPE、LOGICAL_KINDS、REAL_KINDS、REAL32、REAL64、REAL128、STAT_LOCKED、STAT_LOCKED_OTHER_IMAGE、STAT_UNLOCKED
- ALLOCATABLE または POINTER 属性を持たない OPTIONAL 仮引数は、対応する実引数に ALLOCATABLE 属性があるのに割り当てられない場合、POINTER 属性があるのに関連付けが解除されている場合、または NULL 組み込み関数への参照の場合、無視されます。
- 仮引数がプロシージャ・ポインターの場合、そのポインターの有効な参照先か、または組み込み関数 NULL への参照である実引数に関連付けられます。実引数がポインターでない場合、仮引数に INTENT (IN) 属性が含まれていなければなりません。
- BLOCK 構造
- EXECUTE_COMMAND_LINE 組み込みサブルーチン
- サブモジュール
- IMPURE
- 組み込み型の型宣言
- ポインターの初期化
- 暗黙形状配列
- EXIT 文の構文名の拡張
- 内部プロシージャの BIND(C) のサポート

現在のバージョンでは、次の Proposed draft Fortran 2015 の機能がサポートされています。

- 「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® Visual 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 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、Internet Explorer、Microsoft Edge、Visual Basic、Visual C++、Visual Studio、Windows、および Windows Server 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。

Google および Chrome は Google Inc. の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。

© 2017 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。