

インテル® Visual Fortran コンパイラー 19.1 for Windows* リリースノート (インテル® Parallel Studio XE 2020)

2019年12月4日

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 19.1 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](#)
- [インストール方法](#)
- [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](#)
- [Visual Studio* 統合の変更点](#)
- [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* 2017 に関する注意事項](#)
- [Fortran 2008 および Fortran 2018 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。

インテル® Visual Fortran コンパイラー 19.0 以降 (インテル® Visual Fortran コンパイラー 19.1.0 での変更)

- [Fortran 2018 の新機能](#)
- [OpenMP* 5.0 の新機能](#)

- [新しいコンパイラー・オプション](#)
 - [/assume:\[no\]old inquire recl](#)
 - [/assume:\[no\]old ldout zero](#)
 - [/check:\[no\]udio iostat](#)
 - [/warn:\[no\]externals](#)
- [コンパイラーの動作の変更](#)
- このリリースは Visual Studio* 2019 との統合のサポートを含みます。
- 報告された問題を修正

製品の内容

インテル® Visual Fortran コンパイラー 19.1 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 19.1。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
- オンライン・ドキュメントとサンプルプログラム

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)
- 14GB 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)
- Microsoft* Windows® 10、Microsoft* Windows Server* 2016、Windows Server* 2019
- IA-32 対応アプリケーション [\[1\]](#) 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:
 - Microsoft* Visual Studio* 2017 Professional Edition 以上または Microsoft* Visual Studio* Community 2017 (「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) [\[3\]](#)
 - Microsoft* Visual Studio* 2019 Professional 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)
- 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:
 - Microsoft* Build Tools 2017
 - Microsoft* Build Tools 2019
- ドキュメントの参照用に Adobe* Reader* 7.0 以降

注:

1. 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。
2. 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* 10 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。
3. 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。

インテル® Fortran コンパイラーのインストール方法

インストール手順は、ドキュメントの一部として、インテル® ソフトウェア開発製品に同梱されています。

最新のインテル® Parallel Studio XE バージョンのインストール・ガイドは、オンラインでも利用できます。「[インテル® Parallel Studio XE 入門](#)」(英語) でインストール・ガイドを確認してください。インストール・ガイドへのリンクは、必要なインストール・パッケージをダウンロードしている間、[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](#)からも利用できます。詳細は、「[インストールに関するよくある質問](#)」(英語) を参照してください。

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は、「[インテル® Parallel Studio XE 2020: インテル® Fortran コンパイラー 19.1 for Windows* 入門](#)」(<install-dir>\documentation_2020\ja\compiler_f\ps2020\getstart_wf.htm) を参照してください。

ドキュメント

製品ドキュメントへのリンクは、<install-dir>\documentation_2020\ja\compiler_f\ps2020\getstart_wf.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。[ヘルプ] メニューから表示可能な入門ガイドにあるリンクを使用して、オンライン・ドキュメントにアクセスすることもできます。

インテル® コンパイラー・ドキュメントをローカルマシンに追加するには、[こちらの記事](#) (英語) の手順に従ってください。

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、インテル® ソフトウェア・ドキュメント・ライブラリーの 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。

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。

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 2020 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。

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。

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。

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。

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。

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 19.1 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 (/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。

- インテル® Fortran コンパイラー 18.0 以前のバージョンの `regcall` 呼び出し規約でビルドされたオブジェクトは、19.0 以降で再コンパイルする必要があります。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® Fortran コンパイラー 19.0 以降に同梱の `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。`regcall` は、次の属性で暗黙的に使用されています。
`!DIR$ ATTRIBUTES [att,] VECTOR [:clause] [, att]... :: routine-name`

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。

新規および変更されたコンパイラー機能

Fortran 2018 の新機能

- `IMPLICIT` 文の拡張によりすべての外部プロシーチャーを `EXTERNAL` で宣言する指定を許可
- `GENERIC` 文の拡張により汎用インターフェイスの宣言での使用を許可
- 変数の局所性を `DO CONCURRENT` 文で指定可能に
- `E`、`D`、`EN`、`ES`、および `G` 編集記述子の拡張によりゼロ幅のフィールドを許可、`F` 編集記述子に類似
- データ編集記述子の指数の幅 `e` をゼロに、ゼロ幅のフィールドに類似

- RN 編集記述子を Fortran 2018 および ISO/IEC/IEEE 60559:2011 で指定された最近接丸めに
- EX 編集記述子で浮動小数点値の 16 進形式の出力を許可。16 進形式の浮動小数点値を入力で許可。
- SIZE= を非前進 I/O で指定
- 保留中の非同期操作で INQUIRE 文の SIZE= および POS= の値を標準化
- INQUIRE 文の RECL= 指定子に割り当てられる値を標準化
- CMLPX 組込み関数の新しい形式、最初の引数が COMPLEX 型の場合、KIND= キーワードは不要
- SIGN 関数の引数で異なる種類を指定可能に
- 組込みプロシージャーの INTEGER および LOGICAL 引数でデフォルトの種類が不要に
- 名前付き定数 STAT_FAILED_IMAGE および STAT_UNLOCKED_FAILED_IMAGE を組込み ISO_FORTRAN_ENV モジュールで定義
- 非ブロック DO 文および算術 IF 文は Fortran 2018 で削除。インテル® Fortran コンパイラーは FORTRAN 標準で削除された機能をフルサポート。
- COMMON、EQUIVALENCE および BLOCKDATA 文が古い機能に
- ラベル付き DO ループが古い機能に
- DO CONCURRENT 構文の変数の局所性を DO CONCURRENT 文で宣言可能に
- 組込みプロシージャーの固有名が古い機能に
- FAIL IMAGE 文により実際のイメージ失敗を待つ必要なく失敗したイメージの回復コードのデバッグを許可
- オプション引数 STAT= を ATOMIC_REF および ATOMIC_DEFINE 組込みプロシージャーに追加
- オプションの STAT= および ERRMSG= 指定子を MOVE_ALLOC 組込みプロシージャー、イメージセレクター、CRITICAL 文と構文に追加
- ATOMIC_ADD、ATOMIC_AND、ATOMIC_CAS、ATOMIC_FETCH_ADD、ATOMIC_FETCH_AND、ATOMIC_FETCH_OR、ATOMIC_FETCH_XOR、ATOMIC_OR、および ATOMIC_XOR アトミック・サブルーチンを実装
- CO_BROADCAST、CO_MAX、CO_MIN、CO_REDUCE、および CO_SUM 集合サブルーチンを実装
- ランク引き継ぎ仮引数の操作を許可する SELECT RANK 構文を実装
- Fortran 2018 で必要な非標準組込みプロシージャーとモジュールの使用をコンパイラーで診断
- ISO_C_BINDING、IEEE_ARITHMETIC、および IEEE_EXCEPTIONS 組込みモジュールの変形組込み関数を宣言式で許可
- IEEE_GET_ROUNDING_MODE および IEEE_SET_ROUNDING_MODE 組込みモジュール・プロシージャーでオプション引数 RADIX を指定可能に
- オプションの ROUND 引数を IEEE_ARITHMETIC 組込みモジュールの IEEE_RINT 関数に追加
- IEEE_ARITHMETIC 組込みモジュールに IEEE_FMA、IEEE_SIGN_BIT、IEEE_NEXT_UP および IEEE_NEXT_DOWN 関数を追加
- IEEE_MAX、IEEE_MIN、IEEE_MAX_MAG、および IEEE_MIN_MAG 組込みモジュール・プロシージャーを実装
- IEEE_INT および IEEE_REAL 組込みモジュール・プロシージャーを実装
- IEEE_EXCEPTIONS 組込みモジュールに IEEE_GET_MODES および IEEE_SET_MODES 組込みモジュール・プロシージャーを使用した IEEE_MODES の保存および回復に使用できる新しい派生型、IEEE_MODES_TYPE を追加
- 新しい丸めモード、IEEE_AWAY を追加
- SUBNORMAL が DENORMAL と同義に
- IEEE_QUIET_EQ、IEEE_QUIET_NE、IEEE_QUIET_LT、IEEE_QUIET_LE、IEEE_QUIET_GT、IEEE_QUIET_GE、IEEE_SIGNALING_EQ、IEEE_SIGNALING_NE、IEEE_SIGNALING_GT、

IEEE_SIGNALING_GE、IEEE_SIGNALING_LT、および IEEE_SIGNALING_LE 組込みモジュール・プロシーチャーを実装

OpenMP* 5.0 の新機能

- SIMD デイレクティブの IF 節
- SIMD デイレクティブの NONTEMPORAL 節
- SIMD および TARGET SIMD 構文の包括/排他スキャン操作

コンパイラーの動作の変更

- 以前のコンパイラー・バージョンでは、派生型宣言の中で型引数を宣言する前に PRIVATE 文または SEQUENCE 文を使用することができました。これは誤りです。インテル® Fortran コンパイラー 19.1 では、この動作は許可されなくなりました。
- 呼び出しが成功した場合、INQUIRE 文はフルパスおよび標準化されたファイルパスを使用します。変更の例:
"bar/foo"としてファイルを開く。
変更前:
INQUIRE 文で "./bar/foo"、"bar//foo" または "bar/../bar/foo" が開かれているかどうか問い合わせる。動作しない。
変更後:
INQUIRE 文で "./bar/foo"、"bar//foo" または "bar/../bar/foo" が開かれているかどうか問い合わせる。動作する。
- コンパイラー 19.0 では、ISO_C_BINDING 組込みモジュールの C_F_POINTER 例外を含むすべてのプロシーチャーは Fortran 2018 標準に従って PURE にされていました。解釈 18/007 で、C_F_PROCPOINTER PURE にすることは誤りであったことが示されました。コンパイラー 19.1 では、この解釈に従って C_F_PROCPOINTER IMPURE にするように変更されました。

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。次の新しいコンパイラー・オプションが追加されました。

- [/assume:\[no\]old inquire recl](#)
- [/assume:\[no\]old ldout zero](#)
- [/check:\[no\]udio iostat](#)
- [/warn:\[no\]externals](#)

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。

`/assume:[no]old_inquire_recl`

未接続のユニットまたはストリームアクセス向けに接続されたユニットへの INQUIRE 文の RECL= 指定子の値を指定します。

`/assume: [no]old_ldout_zero`

list-directed 出力で生成される浮動小数点ゼロの形式 (`old_ldout_zero` は指数形式、`no_old_ldout_zero` は分数形式) を指定します。

`/check: [no]udio_iostat`

ユーザー定義の派生型入力/出力プロシージャーが実行されたときに標準規格チェックを行うかどうかを指定します。

`/warn: [no]externals`

明示的なインターフェイスの (または `EXTERNAL` と宣言されていない) 仮プロシージャーやプロシージャー呼び出しで警告を出力するかどうかを指定します。

Visual Studio* 統合の変更点

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。

終了予定のサポート

終了したサポート

Microsoft* Visual Studio* 2013 および Microsoft* Visual Studio* 2015 のサポートを終了

Microsoft* Visual Studio* 2013 および Microsoft* Visual Studio* 2015 はインテル® Visual Fortran コンパイラー 19.1 でサポートされません。

Microsoft* Windows* 7、Windows Server* 2008 R2 SP1、Windows Server* 2012 のサポートを終了

Microsoft* Windows* 7、Windows Server* 2008 R2 SP1、Windows Server* 2012 はインテル® Visual Fortran コンパイラー 19.1 でサポートされません。

ループ・プロファイラー

ループ・プロファイラー機能はインテル® Visual Fortran コンパイラー 19.1 で削除されました。ループ・プロファイラーに関連する次のコンパイラー・オプションは削除されました: /Qprofile-loops:keyword、/Qprofile-loops-report=value、/Qprofile-functions、/Qguide-profile。

Microsoft* Visual Studio *Shell

Microsoft* は、Microsoft* Visual Studio* Shell 2017 (Isolated) を提供しないことを発表しました。そのため、Microsoft* Visual Studio* Shell は製品に同梱されなくなります。Shell (Integrated) は完全な Microsoft* Visual Studio* バンドルの一部として利用できます。Microsoft* Visual Studio* 製品の詳細は、<https://visualstudio.microsoft.com/ja/vs/> を参照してください。

詳細は、[インテル® デベロッパー・ゾーンの記事](#) (英語) を参照してください。

IA-32 ホスト・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、インテル® Parallel Studio XE 2018 で終了しました。

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

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。

既知の問題

コマンドプロンプトから psxevars.bat を実行すると警告メッセージが表示される

コマンドプロンプトから psxevars.bat を実行してコンパイラー環境を設定すると警告またはエラーメッセージが表示されます。このメッセージは無視してかまいません。psxevars.bat の実行後に環境は正しく設定されています。"The system cannot find the file specified." のような警告メッセージが表示されても無視してください。

メッセージを無視し、コマンドプロンプトを引き続き使用してかまいません。

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ーを使用して、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ーを使用する場合、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。

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

PARAMETER 定数と /debug-parameters および /debug コンパイラー・オプションと一緒に使用すると、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。

```
01 module sample_mod
02 implicit none
03 integer, parameter :: isize=32
04
05 contains
06
07 subroutine example1
08 call example2(isize)
09 end subroutine example1
10
11 subroutine example2(jsize)
12 integer, intent(in) :: jsize
13 write(*,*) jsize
14 end subroutine example2
15
16 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。

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® Visual Fortran コンパイラーでは、この使用方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
```

32 ビットの Co-Array Fortran が 64 ビットの Microsoft* Windows* で動作しない

32 ビットの Co-Array Fortran は、必要なユーティリティー "mpiexec.exe" と "smpd.exe" が正しく動作しないため、64 ビットの Microsoft* Windows® 10 または Microsoft* Windows Server* 2012 R2 では動作しません。これは、互換性の問題です。詳細は、「[32 ビットの Co-Array Fortran が 64 ビットの Microsoft* Windows* で動作しない](#)」(英語) を参照してください。

Fortran Co-Array アプリケーションが FAIL IMAGE の実行後にハングする

現在の FAIL IMAGE 実装には FAIL IMAGE 文を実行した後に発生する既知の問題があります。詳細は、[この記事](#) (英語) を参照してください。

Microsoft* Visual Studio* 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* 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

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ーを展開します。
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_COMPILER19)\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_COMPILER19)\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® 10 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* 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。
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。

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* ファイルの既定のアプリにブラウザ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 [既定のプログラムの設定] で .htm ファイルを任意のブラウザに関連付けます。または、[設定] > [システム] > [既定のアプリ] > [既定のアプリの選択] で .htm ファイルの既定のアプリとして任意のブラウザを選択します。

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。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。

Fortran プロジェクトの空のクラスビュー

Fortran プロジェクトで空のクラスビューで問題が発生することがあります。この問題を回避するには、手動で既存の中間ファイル (.fdz とその他のファイル) をすべて削除し、[ツール] > [オプション] > [テキスト エディター] > [Fortran] > [Advanced (詳細)] で次の設定を確認します。

- [Collect Call/Callers graph information (呼び出し先/呼び出し元情報の収集)] と [Collect Object Browser information (オブジェクト・ブラウザー情報の収集)] が True に設定されていること
- [Disable Database (データベースの無効化)] が False に設定されていること

Microsoft* Visual Studio* 2017 の Fortran デバッガーに関する問題

Microsoft* Visual Studio* 2017 バージョン 15.7.x または 15.8.x で Fortran プログラムをデバッグするとクラッシュします。また、ブレークポイントで Fortran 配列の値が表示されません。詳細は、[こちらの記事](#) (英語) を参照してください。

Fortran 2008 および Fortran 2018 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と Fortran 2018 標準のほとんどの機能をサポートします。現在のバージョンでサポートされている新しい Fortran 2018 の機能は、[Fortran 2018 の新機能](#) セクションを参照してください。

必要に応じて、[Fortran 2008 標準](#) (PDF、英語) および [Fortran 2018 標準](#) (PDF、英語) を参照してください。

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。

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られません。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 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。

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。