インテル® Visual Fortran Composer XE 2013 Windows* 版インストール・ガイドおよびリリースノート

資料番号: 321417-004JA
2013 年 3 月 5 日

目次

1 概要 3
1.1 製品アップデート 3
1.2 インテル® Visual Fortran Composer XE 2011 からの変更点 4
1.3 製品の内容 4
1.4 動作環境 4
1.5 ドキュメント 6
1.5.1 『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』は Web から入手可能 6
1.5.2 Visual Studio* 2008 のインテル® Composer XE ヘルプに最初にアクセスしたときの遅延 6
1.6 最適化に関する注意事項 7
1.7 サンプル 7
1.8 日本語サポート 7
1.9 テクニカルサポート 7
2 インストール 8
2.1 インストール前の準備 8
2.1.1 インストールに必要なソフトウェア 8
2.1.2 64 ビット・アプリケーション用の Visual Studio* の設定 8
2.2 インストール 8
2.2.1 インストール後の再起動を推奨 9
2.2.2 クラスターでのインストール 9
2.2.3 ライセンスサーバーの使用 9
2.2.4 Microsoft* Visual Studio* 2010 用ドキュメントをインストールするための追加ステップ 9
2.3 インテル® Software Manager 10
2.4 製品の変更、更新、削除 10
2.5 サイレント・インストール/アンインストール 10
2.6 インストール先フォルダー 10

3 インテル® Visual Fortran コンパイラー 11
3.1 互換性 11
3.1.1 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 (12.0) 12
3.1.2 インテル® OpenMP* ライブラリーのスタティック・ライブラリーの提供を終了 12
3.2 新規および変更されたコンパイラー機能 12
3.2.1 Fortran 2003 の機能 12
3.2.2 Fortran 2008 の機能 12
3.2.3 Co-Array 13
3.2.4 新しい宣言子と追加された宣言子 13
3.2.5 OpenMP* 4.0 の変更 (13.1.0) 13
3.2.6 派生型のコンポーネントでの ATTRIBUTES ALIGN 宣言子の指定 (13.0.1) 14
3.2.7 ファイル・バッファリング動作の変更 (13.1) 14
3.2.8 その他の変更 15
3.3 新規および変更されたコンパイラー・オプション 15
3.3.1 インテル® Composer XE 2013 の新規および変更されたコンパイラー・オプション 15
3.3.2 新しい /Qvec-report7 コンパイラー・オプション (13.1.0) 16
3.4 並列ビルドオプション 16
3.5 既知の問題 16
3.5.1 Co-Array の問題 16
3.5.2 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 16
3.6 Microsoft* Visual Studio* 2010/2012 に関する注意事項 16
3.6.1 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 17
3.6.2 プロジェクトの依存関係の調整 18
3.6.3 Windows Server* 2012 で Visual Studio* 2012 のドキュメントを表示できない場合 18
3.7 Fortran 2003 および Fortran 2008 機能の概要 18
4 インテル® マス・カーネル・ライブラリー (インテル® MKL) 21
4.1 インテル® MKL 11.0 Update 3 の新機能 21
4.2 インテル® MKL 11.0 Update 2 の新機能 22
4.3 インテル® MKL 11.0 Update 1 の新機能 23
4.4 インテル® MKL 11.0 の新機能 23
4.5 推奨されていない (古い) 機能と削除された機能 24
4.6 既知の問題 24
4.7 権利の帰属 24
5 著作権と商標について 25

1 概要

このドキュメントでは、製品のインストール方法、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。

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。各コンポーネントの新機能の詳細は、各コンポーネントのリリースノートを参照してください。

1.1 製品アップデート

Update 3 – 2013 年 3 月

- インテル® Visual Fortran コンパイラーが 13.1.1 にアップデート
- インテル® マス・カーネル・ライブラリーが 11.0 Update 3 にアップデート
- 報告された問題の修正
  - コンパイラーの修正リスト
  - インテル® MKL の修正リスト

Update 2 – 2013 年 1 月

- インテル® Visual Fortran コンパイラーが 13.1.0 にアップデート
  - OpenMP* 4.0 の追加の宣言子、節およびプロシージャーをサポート
  - KMP_PLACE_THREADS ランタイム環境変数を追加
  - Co-Array アプリケーションのパフォーマンスを向上できるオプションを追加 (Update 1)
  - /Qvec-report7 コンパイラー・オプションを追加
  - 読み取り中の書式なしシーケンシャル・ファイルのバッファリングの変更についての説明を追加
  - GetLogicalProcessorInformation の使用例を含む ProcessorInfo サンプルを Win32.zip に追加
  - より標準の Fortran 機能を使用するように Dll.zip の DynamicLoad サンプルを更新
- インテル® マス・カーネル・ライブラリーが 11.0 Update 2 にアップデート
- 報告された問題の修正
  - コンパイラーの修正リスト: http://intel.ly/S5ulAb (英語)
  - インテル® MKL の修正リスト: http://intel.ly/S5uw3R (英語)

Update 1 – 2011 年 10 月

- インテル® Visual Fortran コンパイラーが 13.0.1 にアップデート
  - 派生型の ALLOCATABLE または POINTER コンポーネントでの ATTRIBUTES ALIGN 宣言子の指定をサポート
  - モジュール IFWINTY と KERNEL32 に Windows* API ルーチン GetLogicalProcessorInformation および GetLogicalProcessorInformationEx をサポートする宣言を追加
- インテル® マス・カーネル・ライブラリーが 11.0 Update 1 にアップデート
- Windows Server* 2012 で Visual Studio* 2012 のドキュメントを表示できない場合について情報を追加
- 報告された問題の修正
  - コンパイラーの修正リスト: http://intel.ly/S5ulAb (英語)
  - インテル® MKL の修正リスト: http://intel.ly/S5uw3R (英語)

1.2 インテル® Visual Fortran Composer XE 2011 からの変更点

- インテル® Visual Fortran コンパイラーがバージョン 13.0 にアップデート
  - インテル® OpenMP* ライブラリーのスタティック・ライブラリー libiomp5mt.lib の提供を終了しました。
  - PATH 環境変数が非常に長くなった場合、インストール後に再起動が必要になることがあります。
- インテル® マス・カーネル・ライブラリーがバージョン 11.0 にアップデート
  - インテル® MKL でインテル® Pentium® III プロセッサーのサポートが終了。インテル® MKL でサポートされる最小の命令セットはインテル® SSE2 になります。詳細は、「インテル® MKL」セクションを参照してください。
  - インテル® MKL のランタイム DLL のパスが PATH 環境変数に追加されなくなりました。インテル® MKL の DLL を使用するプログラムを実行する場合は、適切なフォルダーを PATH 環境変数に追加する必要があります。
- インテル® Parallel Debugger Extension の提供を終了
- Microsoft* Windows* 8、Microsoft* Windows Server* 2012、Microsoft* Visual Studio* 2012 のサポートを追加
  - 本リリースは、これらの Microsoft* 製品の最終リリース前に作成されたものです。Microsoft* 製品の最終リリースでインテル® Visual Fortran Composer XE の動作に影響する変更が加えられた場合は、できるだけ早くアップデートを提供する予定です。
- 次のバージョンの Windows* のサポートを終了:
  - Windows Vista*
  - Windows Server* 2003
- Microsoft* Visual Studio* 2005 のサポートが終了
- Windows* XP のサポート終了予定
- Visual Studio* の [ツール] > [オプション] > [Intel Visual Fortran (インテル(R) Visual Fortran)] ダイアログが、[ツール] > [オプション] > [Intel Composer XE (インテル(R) Composer XE)] > [Visual Fortran] に変更
- 製品のアップデートとライセンスのアクティベーションを管理するインテル® Software Manager の追加
- 報告された問題の修正

1.3 製品の内容

インテル® Visual Fortran Composer XE 2013 Windows 版* には、次のコンポーネントが含まれています。

- インテル® Visual Fortran コンパイラー XE 13.1.1。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。
- インテル® マス・カーネル・ライブラリー 11.0 Update 3
- Microsoft* 開発環境への統合
- Microsoft* Visual Studio* 2010 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

1.4 動作環境

アーキテクチャー名についての説明は、http://intel.ly/q9JVjE (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 IA-32 またはインテル® 64 アーキテクチャー・プロセッサーをベースとするコンピューター (インテル® Pentium® 4 プロセッサー以降、または互換性のあるインテル以外のプロセッサー)

    - 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ー・システムの使用を推奨します。
- RAM 1GB (2GB 推奨)
- 2GB 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* XP SP3、Microsoft* Windows* 7、Microsoft* Windows* 8、Microsoft* Windows Server* 2012、Microsoft* Windows Server* 2008、Microsoft* Windows HPC Server* 2008 (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* 2012、Microsoft* Visual Studio* 2010、Microsoft* Visual Studio* 2010 Shell、Microsoft* Visual Studio* 2008 SP1、または Visual Studio* 2008 SP1 アップデートが適用された Visual Studio* 2008 Shell が必要です。
    - Microsoft* Windows* 8 では、製品は「デスクトップ」環境にインストールされます。「新しい 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。
    - Microsoft* Windows* XP のサポート終了予定 - インテル® Visual Fortran Composer XE の将来のメジャーリリースでは、Windows* XP はサポートされなくなる予定です。
- 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* 2012 (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* 2010 (C++ コンポーネントと [X64 コンパイラおよびツール] コンポーネントがインストールされていること) [1]
    - Microsoft* Visual Studio* 2008 Standard Edition 以降 (C++ コンポーネントと [x64 コンパイラおよびツール] コンポーネントがインストールされていること) [1]
    - Microsoft* Visual Studio* 2010 Shell (インテル® 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 [2]
    - Microsoft* Visual Studio* 2008 Shell (インテル® Fortran コンパイラー 11.0、11.1、およびインテル® Visual Fortran Composer XE 2011 Update5 までに付属) ベースのインテル® Visual Fortran 開発環境
- IA-32 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++* 2010 Express Edition [2]
    - Microsoft* Visual C++* 2008 Express Edition
- インテル® 64 対応アプリケーションのビルドのみにコマンドライン・ツールを使用する場合:
    - Microsoft* Windows* Software Development Kit for Windows* 7 and .NET Framework 4.0
- Microsoft* Visual Studio* 2010 Shell には次の制限があります。
    - Windows* XP 64 ビットではサポートされていません。Windows* XP 64 ビットでは、以前のバージョンのインテル® Visual Fortran に付属の Microsoft* Visual Studio* 2008 Shell を使用できます。
    - Windows* XP では、Microsoft* Visual Studio* 2010 Shell をインストールする前に Microsoft* .NET 4.0 Framework をインストールする必要があります。詳細は、このリリースノートの「インストール」セクションを参照してください。
- ドキュメントの参照用に Adobe* Reader* 7.0 以降

注:

1. Microsoft* Visual Studio* 2008 Standard Edition では、[x64 コンパイラおよびツール] コンポーネントがデフォルトでインストールされます。Professional 以上のエディションでは、[カスタム] インストールが必要です。Microsoft* Visual Studio* 2010 では、すべてのエディションでこのコンポーネントがデフォルトでインストールされます。
2. Microsoft* Visual Studio* 2010 Shell ベースのインテル® Visual Fortran 開発環境は、インテル® Visual Fortran Composer XE のアカデミック・ライセンスと商用ライセンスに含まれています。評価版ライセンスには含まれていません。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
3. Microsoft* Visual C++* 2010 Express Edition とインテル® Visual Fortran Composer XE 2013 に付属の Microsoft* Visual Studio* 2010 Shell は共存できます。C++ と Fortran 開発環境はそれぞれ独立しています。
4. 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ーが必要な IA-32 アーキテクチャー・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・プロセッサー上で動作するコードを生成できます。ただし、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ーが必要です。
5. 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は行われていません。開発アプリケーションが、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。

1.5 ドキュメント

製品ドキュメントは、「インストール先フォルダー」で示されているように、`Documentation` フォルダーに保存されています。

1.5.1 『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』は Web から入手可能

コンパイラー・ドキュメントの「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」は、Intel® Software Documentation Library Web サイトに移動しました。http://intel.ly/WinApp (英語) から PDF 形式のドキュメントをダウンロードできます。

1.5.2 Visual Studio* 2008 のインテル® Composer XE ヘルプに最初にアクセスしたときの遅延

Microsoft* Visual Studio* 2008 にインストールされたヘルプ・ドキュメントに最初にアクセスしたとき、表示に時間がかかる場合があります。これは、Visual Studio* でヘルプを表示する前に、新しいヘルプをコレクションに統合して、コレクションの索引を再生成するためです。Visual Studio* にすでに統合されているヘルプとインストールするヘルプのサイズに応じて、新しくヘルプが表示されるまで数分以上かかる場合があります。

1.6 最適化に関する注意事項

最適化に関する注意事項
インテル® コンパイラーは、互換マイクロプロセッサー向けには、インテル製マイクロプロセッサー向けと同等レベルの最適化が行われない可能性があります。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® SSE2)、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3)、ストリーミング SIMD 拡張命令 3 補足命令 (SSSE3) 命令セットに関連する最適化およびその他の最適化が含まれます。インテルでは、インテル製ではないマイクロプロセッサーに対して、最適化の提供、機能、効果を保証していません。本製品のマイクロプロセッサー固有の最適化は、インテル製マイクロプロセッサーでの使用を目的としています。インテル® マイクロアーキテクチャーに非固有の特定の最適化は、インテル製マイクロプロセッサー向けに予約されています。この注意事項の適用対象である特定の命令セットの詳細は、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。 改訂 #20110804

1.7 サンプル

製品コンポーネントのサンプルは、「インストール先フォルダー」の説明にある `Samples` フォルダーに用意されています。

1.8 日本語サポート

インテル® コンパイラーは、日本語と英語の両方を備えたインストーラーで日本語をサポートしています。エラーメッセージ、ビジュアル開発環境ダイアログ、ドキュメントの一部が英語のほかに日本語でも提供されています。エラーメッセージやダイアログの言語は、システムの言語設定に依存します。日本語版ドキュメントは、Documentation および Samples ディレクトリー以下の `ja_JP` サブディレクトリーにあります。

日本語サポートはすべての製品アップデートで提供されているわけではありません。

日本語サポート版を英語のオペレーティング・システムで使用する場合や日本語のオペレーティング・システムで英語サポート版を使用する場合は、http://intel.ly/pla2A5 (英語) の説明を参照してください。

1.9 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: 代理店がテクニカルサポートを提供している場合は、インテルではなく代理店にお問い合わせください。

2 インストール

2.1 インストール前の準備

2.1.1 インストールに必要なソフトウェア

製品に付属する Microsoft* Visual Studio* 2010 Shell をインストールする場合、インテル® Visual Fortran Composer XE 2013 をインストールする前に、追加で Microsoft* ソフトウェアのインストールが必要な場合があります。Microsoft* Visual Studio* 2010 Shell は、Windows* XP 64 ビットではサポートされていません。

Microsoft* Visual Studio* 2010 Shell のインストールには Microsoft* .NET 4.0 Framework が必要です。Microsoft* .NET 4.0 Framework は、次のリンクからインストーラーをダウンロードできます。

- .NET 4.0 Framework 32 ビットおよび 64 ビット

Windows* 7、Windows* 8 または Windows Server* 2008 にインストールする場合は、Shell のインストール時に .NET Framework 4.0 が (システムにインストールされていない場合) 自動的にダウンロードされインストールされます。この処理に失敗すると、エラーメッセージが表示され、インテル® Visual Fortran Composer はインストールされません。Shell のインストールに失敗した場合は、上記のリンクから .NET 4.0 Framework をダウンロードしてやり直してください。

DVD または Visual Studio* 2010 Shell 付属製品のダウンロード版を使用してインテル® Visual Fortran Composer XE 2013 をインストールする場合、マシンに Visual Studio* 2010 がインストールされていないと、インストーラーは Visual Studio* 2010 Shell をインストールしようとします。Visual Studio* 2010 Shell をインストールしない場合は、「カスタム・インストール」を選択して、Visual Studio* 2010 Shell のチェックをオフにするか、"_novsshell.exe" (Visual Studio* 2010 Shell なし) のインストーラーをダウンロードして使用してください。

2.1.2 64 ビット・アプリケーション用の Visual Studio* の設定

Microsoft* Visual Studio* 2008 を使用し、64 ビット・アプリケーション (インテル® 64 アーキテクチャー向け) を開発する場合は、Visual Studio* の構成を変更して、64 ビット・サポートを追加します。

Visual Studio* 2008 Standard Edition または Visual Studio* 2008 Shell を使用する場合は、インテル® 64 対応アプリケーションのビルド用に構成を変更する必要はありません。その他のエディションの場合は、次の操作を行ってください。

1. [コントロール パネル] の [プログラムの追加と削除] から [Microsoft Visual Studio 2008] を選択し、[変更と削除] をクリックします。[Visual Studio メンテナンス モード] ウィンドウが表示されます。[次へ] をクリックします。
2. [機能の追加と削除] をクリックします。
3. [選択した機能をインストールします] で [言語ツール] の [Visual C++] を展開します。
4. [x64 コンパイラおよびツール] ボックスがオンになっていない場合は、オンにし、[更新] をクリックします。ボックスがオンの場合は、[キャンセル] をクリックします。

Microsoft* Visual Studio* 2010 または Visual Studio* 2012 を使用している場合、このステップは必要ありません。

2.2 インストール

本製品のインストールには、有効なライセンスファイルまたはシリアル番号が必要です。本製品を評価する場合には、インストール時に [製品を評価する (シリアル番号不要)] オプションを選択してください。

DVD で製品を受け取った場合、製品 DVD を DVD ドライブに挿入します。自動でインストールが開始されます。自動で開始されない場合は、Windows* エクスプローラーで DVD ドライブのトップレベル・ディレクトリーを開き、setup.exe をダブルクリックします。

製品のダウンロード版を購入した場合は、ダウンロードしたファイル (.EXE) をダブルクリックして、インストールを開始します。利用可能なダウンロード・ファイルには各種あり、それぞれ異なるコンポーネントの組み合わせを提供していることに注意してください。ダウンロード・ページを注意深くお読みになり、適切なファイルを選択してください。

新しいバージョンをインストールする前に古いバージョンをアンインストールする必要はありません。新しいバージョンは古いバージョンと共存可能です。以前のバージョンの削除は、このバージョンをインストールする前でも後でも行うことができます。

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 でシリアル番号を登録すると、製品のアップデートや以前のバージョンを利用できます。

2.2.1 インストール後の再起動を推奨

インテル® Visual Fortran Composer XE をインストールすると、(インテル® MKL を除く) コンパイラーのランタイム DLL が含まれるフォルダー名が PATH 環境変数に追加されます。一部のシステムでは、PATH の長さが非常に長くなると (2048-4096 文字)、システムを再起動するまでコマンドラインが動作しなくなることがあります。インテル® Visual Fortran Composer XE をインストールした後は、システムを再起動することを推奨します。

2.2.2 クラスターでのインストール

インストールするマシンに Microsoft* Compute Cluster Pack のライセンスがあり、クラスターメンバーの場合、「フル・インストール」を選択すると、そのクラスターのアクセス可能なすべてのノードに製品がインストールされます。「カスタム・インストール」を選択すると、現在のノードのみにインストールするオプションを選択できます。

2.2.3 ライセンスサーバーの使用

「フローティング・ライセンス」を購入された場合は、ライセンスファイルまたはライセンスサーバーを使用したインストール方法について http://intel.ly/oPEdEe (英語) を参照してください。この記事には、多様なシステムにインストールすることができる FLEXlm* ライセンス・マネージャーに関する情報も記述されています。

2.2.4 Microsoft* Visual Studio* 2010 用ドキュメントをインストールするための追加ステップ

Microsoft* Visual Studio* 2010 がインストールされているシステムにインテル® Visual Fortran Composer XE 2013 を初めてインストールするとき、Visual Studio* 2010 のドキュメントの「ローカルストア」を初期化するかどうか確認するメッセージが表示されます (初期化を行っていない場合)。「ヘルプ ライブラリ マネージャー」によってインテル® Visual Fortran Composer XE 2013 ヘルプ・ドキュメントが Visual Studio* 2010 内に登録されます。「ヘルプ ライブラリ マネージャー」のインストール・ウィザードの説明に従って、Visual Studio* 2010 用のインテル® Visual Fortran Composer XE 2013 ヘルプ・ドキュメントをインストールします。

このステップは 1 回のみ実行する必要があります。将来インテル® Visual Fortran Composer XE 2013 のアップデートをインストールするときに、「ヘルプ ライブラリ マネージャー」を使用してドキュメントを再登録する必要はありません。

詳細は、http://msdn.microsoft.com/en-us/library/dd264831.aspx を参照するか、microsoft.com で「ヘルプ ライブラリ マネージャー」を検索してください。

2.3 インテル® Software Manager

インテル® Software Manager は、製品アップデートの配信方法を簡素化し、現在インストールされているすべてのインテル® ソフトウェア製品のライセンス情報とステータスを表示します。

将来の製品設計の参考のため、製品使用状況に関する匿名情報をインテルに提供する、インテル® ソフトウェア向上プログラムに参加できます。このプログラムは、デフォルトで無効になっていますが、インストール中または後から有効にして参加できます。参加はいつでも取りやめることができます。詳細は、http://intel.ly/SoftwareImprovementProgram (英語) を参照してください。

2.4 製品の変更、更新、削除

Windows* のコントロールパネルの [プログラムの追加と削除] / [プログラムと機能] でインストールまたは削除する製品コンポーネントを変更します。インストールした製品に応じて、以下のいずれかのエントリーが表示されます。

- インテル® Visual Fortran Composer XE 2013 Windows* 版
- インテル® Composer XE 2013 Windows* 版
- インテル® Parallel Studio XE 2013 Windows* 版

コンパイラーのインストールの一部として Microsoft* Visual Studio* 2010 Shell をインストールした場合、以下の追加エントリーが表示されます。

- Microsoft* Visual Studio* 2010 Shell (Integrated) - JPN
- インテル(R) Visual Fortran 用 Microsoft* Visual Studio* 2010 ファイル
- Microsoft Visual Studio 2010 Remote Debugger – JPN

製品を完全に削除する場合を除き、これらのエントリーは削除しないでください。

製品のアップデート・バージョンをインストールする際、古いバージョンを最初にアンインストールする必要はありません。アップデートを最初にインストールする場合、古いバージョンを置換するか、システムで古いバージョンと新しいバージョンの両方を使用するかを選択します。この選択は、将来のアップデートにも適用されます。Microsoft* Visual Studio* の [ツール] > [オプション] > [Intel Composer XE (インテル(R) Composer XE)] > [Visual Fortran (インテル(R) Visual Fortran)] > [Compiler (コンパイラー)] ダイアログから、使用するコンパイラーのバージョンを選択できます。バージョン 12.0 (インテル® Visual Fortran Composer XE 2011) よりも古いコンパイラーは、Visual Studio* で選択できません。インストールされているすべてのバージョンをコマンドラインから使用できます。

新しいバージョンのコンパイラーを削除した場合、以前のバージョンの Microsoft Visual Studio への統合を再インストールする必要があります。

2.5 サイレント・インストール/アンインストール

コンパイラーの自動インストール/アンインストールについては、http://intel.ly/qAwdvR (英語) を参照してください。

2.6 インストール先フォルダー

インストール・フォルダーの構成を以下に示します。一部含まれていないフォルダーもあります。システム環境変数 `IFORT_COMPILER13` を使用して、マシンにインストールされている最新バージョンのインテル® Visual Fortran Composer XE 2013 を検出できます。

- `C:\Program Files\Intel\Composer XE 2013`
  - `bin`

        - ia32
        - ia32_intel64
        - intel64
    - compiler
        - include
            - ia32
            - intel64
        - lib
            - ia32
            - intel64
    - Documentation
    - help
    - mkl
        - benchmarks
        - bin
        - examples
        - include
        - interfaces
        - lib
        - tests
        - tools
    - redist
    - Samples

bin、include および lib 配下のフォルダーは次のとおりです。

- ia32: IA-32 上で動作するアプリケーションのビルドに使用するファイル
- intel64: インテル® 64 上で動作するアプリケーションのビルドに使用するファイル
- ia32_intel64: IA-32 上での実行用のコンパイラー。インテル®64 上で動作するアプリケーションをビルドします。

英語以外の Windows* システムにインストールする場合、Program Files フォルダー名が異なる場合があります。インテル® 64 アーキテクチャー・システムでは、フォルダー名は Program Files (X86) またはそれに相当する名前です。

デフォルトでは、アップデートによって既存のディレクトリーの内容が置換されます。最初のアップデートをインストールするときに、以前のインストールとは別に新しいアップデートをインストールして、システムに両方のファイルを残すオプションを選択できます。両方を残すオプションを選択した場合、古いアップデートのトップレベルのフォルダー名は Composer XE 2013.*nnn* (*nnn* はアップデート番号) に変更されます。

3 インテル® Visual Fortran コンパイラー

このセクションでは、インテル® Visual Fortran コンパイラーの変更点、新機能、および最新情報をまとめています。

3.1 互換性

一般に、インテル® 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 13.0 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を使用してビルドされた 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を使用してビルドされた 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を使用してインテル® 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を使用してコンパイルされた、派生型宣言の外部で ATTRIBUTES ALIGN 宣言子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。この問題が発生した場合、問題を通知するメッセージが表示されます。
- 派生型宣言の内部で ATTRIBUTES ALIGN 宣言子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。

3.1.1 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 (12.0)

以前のリリースでは、REAL(16) または COMPLEX(16) (REAL*16 または COMPLEX*32) 項目が値で渡されたとき、スタックアドレスは 4 バイトでアラインされていました。パフォーマンスを向上させるため、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ます。引数は 16 バイト境界でアラインされます。

この変更は、主にライブラリーが生成した REAL(16) 値の計算を行うライブラリー (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。

3.1.2 インテル® OpenMP* ライブラリーのスタティック・ライブラリーの提供を終了

インテル® OpenMP* ライブラリーのスタティック・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ることがあります。詳細は、http://intel.ly/PwYFvl (英語) を参照してください。

3.2 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 2003 規格 (http://j3-fortran.org/doc/2003_Committee_Draft/04-007.pdf (英語)) および Fortran 2008 規格 (http://j3-fortran.org/doc/standing/links/007.pdf (英語)) を参照してください。

3.2.1 Fortran 2003 の機能

- 多相変数のデフォルトの初期化
- 外部プロシージャーを参照する場合、汎用インターフェイス・ブロックの MODULE PROCEDURE からキーワード MODULE を省略

3.2.2 Fortran 2008 の機能

- ATOMIC_DEFINE および ATOMIC_REF 組込み関数

3.2.3 Co-Array

共有メモリー環境で Co-Array を使用するプログラムの実行に特別なプロシージャーは必要ありません。実行ファイルを実行するだけでかまいません。根本的な並列化の実装にはインテル® MPI が使用されます。コンパイラーをインストールすると、共有メモリーでの実行に必要なインテル® MPI ランタイム・ライブラリーが自動的にインストールされます。

`/coarray:distributed` オプションを使用するには、インテル® Cluster Studio のライセンスが必要です。Windows* 上で分散 Co-Array アプリケーションを実行する方法については、http://intel.ly/oZurZS (英語) を参照してください。

現在、インテル® MPI 以外の MPI 実装や OpenMP* を使用した Co-Array アプリケーションの使用はサポートされていません。

デフォルトでは、作成されるイメージの数は現在のシステムの実行ユニットの数と同じです。メインプログラムをコンパイルする ifort コマンドで `/Qcoarray-num-images:<n>` オプションを指定することで、この設定を変更することができます。また、環境変数 `FOR_COARRAY_NUM_IMAGES` でイメージ数を指定することもできます。

3.2.3.1 Co-Array のパフォーマンスを向上させるコンパイラー・オプション (13.0.1)

Co-Array アプリケーションのパフォーマンスを向上させる作業は現在も行われています。その成果の一部は 13.0.1 (Composer XE 2013 Update 1) コンパイラーに実装されていますが、デフォルトでは無効になっています。実装されている最適化を有効にするには、アプリケーションをコンパイルするときに次のコンパイラー・オプションを指定します。

```
/switch:coarray_opts
```

現時点では、このオプションを追加することで Co-Array 値を別のイメージからコピーする際のパフォーマンスが向上します。このオプションを使用してもパフォーマンスが大幅に向上しないアプリケーションもあります。アプリケーションをコンパイルするときに、このオプションを指定することを推奨します。エラーが発生した場合は、テクニカルサポートにお知らせください。

将来のメジャーバージョンでは、これらの最適化はデフォルトで有効になり、上記のオプションは削除される予定です。

3.2.3.2 Co-Array の既知の問題

このバージョンでは、以下の機能は完全には動作しません。

- 派生型 Co-Array の ALLOCATABLE または POINTER コンポーネントの別のイメージの値へのアクセス。一部は動作します。

3.2.4 新しい宣言子と追加された宣言子

インテル® Composer XE 2013 では、次のコンパイラー宣言子が追加、変更されています。詳細は、ドキュメントを参照してください。

- ATTRIBUTES CVF
- ORDERED/END ORDERED
- SIMD VECTORLENGTHFOR

3.2.5 OpenMP* 4.0 の変更 (13.1.0)

13.1.0 (Composer XE 2013 Update 2) コンパイラーでは、OpenMP* 4.0 で追加される次の宣言子とプロシージャーがサポートされました。製品のドキュメントが更新されるまで、http://intel.ly/VPV24X (英語) を参照してください。

SIMD 宣言子:

- OMP SIMD
- OMP DECLARE SIMD
- OMP DO SIMD
- OMP PARALLEL DO SIMD

プロシージャー:

- OMP_GET_PROC_BIND

3.2.5.1KMP_PLACE_THREADS 環境変数 (13.1.0)

この環境変数を使用すると、ユーザーは明示的なアフィニティー設定やプロセス・アフィニティー・マスクを記述する代わりに、OpenMP* アプリケーションで使用するコア数およびコアごとのスレッド数を簡単に指定することができます。

構文

```
value = ( int [ "C" | "T" ] [ delim ] | delim ) [ int [ "T" ]
[ delim ] ] [ int [ "O" ] ];

delim = "," | "x";
```

効果

使用するコア数、オフセット値 (オプション)、コアごとのスレッド数を指定します。"C" はコア数、"T" はスレッド数、"O" はオフセットを示します。コア数またはスレッド数のいずれかを指定する必要があります。省略した場合、デフォルト値は利用可能なコア数 (スレッド数) です。

例

5C,3T,10 - オフセット 1 で 5 コア、コアごとに 3 スレッド使用します。
5,3,1 - 上記と同じです。
24 - 最初の 24 コア、コアごとに利用可能なすべてのスレッドを使用します。
2T - すべてのコア、コアごとに 2 スレッド使用します。
,2 - 上記と同じです。
3x2 - 3 コア、コアごとに 2 スレッド使用します。
4C120 - オフセット 12 で 4 コア、コアごとに利用可能なすべてのスレッドを使用します。

3.2.6 派生型のコンポーネントでの ATTRIBUTES ALIGN 宣言子の指定 (13.0.1)

コンパイラー 13.0.1 では、派生型の ALLOCATABLE または POINTER コンポーネントに ATTRIBUTES ALIGN 宣言子が指定されます。宣言子は派生型宣言内に配置しなければなりません。拡張型の場合、宣言子は親の型のコンポーネントを指定してはなりません。

この宣言子が指定されると、コンパイラーは明示的な ALLOCATE または (ALLOCATABLE コンポーネントに対する) Fortran 言語規則に従った暗黙の割り当てによりコンポーネントが割り当てられたときに指定されたアライメントを適用します。

派生型コンポーネントに ATTRIBUTES ALIGN 宣言子を含むモジュールはバージョン 13.0.1 よりも前のコンパイラーで使用できません。

3.2.7 ファイル・バッファリング動作の変更 (13.1)

インテル® Visual Fortran Composer XE 2013 (コンパイラー 13.0) 以前のバージョンでは、Fortran ランタイム・ライブラリーは、可変長の書式なしシーケンシャル・ファイルのレ

コードを読み取るときにすべての入力をバッファリングしていました。このデフォルトのバッファリングは、任意のサイズの可変長レコードをメモリーに保持できる大きな内部バッファーを割り当てることで行われます。非常に大きなレコードの場合、メモリーが過度に使用され、最悪の場合は利用可能なメモリーを使い果たす可能性があります。しかし、レコードを読み取るときのデフォルトのバッファリング動作を変更する方法は用意されていませんでした (レコードを書き込むときにレコードのバッファリングを要求または拒否することは可能でした)。

このデフォルトのバッファリング動作は、インテル® Visual Fortran Composer XE 2013 で変更され、これらのレコードはすべてデフォルトではバッファリング**されず**、ディスクからユーザープログラムの変数に直接読み込まれるようになりました。この変更はメモリーを確保する必要があるプログラムを支援するために行われたものですが、多くの小さなコンポーネントで構成されているレコードを読み取るときにパフォーマンスが低下する場合があります。実際、一部のユーザーから、パフォーマンスの低下が報告されました。

このため、インテル® Visual Fortran Composer XE 2013 Update 2 (コンパイラー 13.1) では、ユーザーがこれらの可変長書式なしレコードをバッファリングするかどうかを選択できる手法が提供されました。デフォルトの動作は 13.0 と同じで、これらのレコードはデフォルトではバッファリングされません。13.1 でこの種の I/O を使用したときにパフォーマンスが低下する場合は、レコードの出力のバッファリングを有効にするかどうかを選択する場合と同じ方法で入力のバッファリングを有効にすることができます。

- ファイルの OPEN 文で BUFFERED="YES" を指定する
- 環境変数 FORT_BUFFERED に YES、TRUE、またはゼロ以外の整数値を指定する
- コンパイラーのコマンドラインで `-assume buffered_io` を指定する

これらのメカニズムは、これまで、可変長書式なしシーケンシャル・ファイルの書き込みを行う場合にのみ適用されていたものです。これらのメカニズムを使用すると、Fortran ランタイム・ライブラリーは、ファイルのレコードのサイズに関係なく、ファイルの入力レコードをすべてバッファリングします。

つまり、13.0 より前のデフォルトの動作に戻ることになります。

3.2.8 その他の変更

- G フォーマット編集記述子の出力が、より適切に Fortran 2008 規格に準拠するように変更されました。この変更には、選択されたフォーマットでの丸めの効果と -0 に丸められた値の表現が含まれます。
- D、E、G、EN、ES フォーマットを使用した出力で指数フィールドが暗黙の指数幅をオーバーフローすると、出力フィールドはアスタリスクで埋められます。以前のバージョンでは、指数の表示が規格に準拠していませんでした。
- コンパイラーは RANDOM_NUMBER および RANF 組込みサブルーチンへの参照をベクトル化するようになりました。
- (13.0.1) モジュール IFWINTY と KERNEL32 に Windows* API ルーチン GetLogicalProcessorInformation および GetLogicalProcessorInformationEx をサポートする宣言が追加されました。

3.3 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。

3.3.1 インテル® Composer XE 2013 の新規および変更されたコンパイラー・オプション

- /align:array8byte
- /align:array16byte

- /align:array32byte
- /align:array64byte
- /align:array128byte
- /align:array256byte
- /assume:[no]std_intent_in
- /Qdiag-enable:sc-enums
- /Qdiag-enable:sc-{full|concise|precise}
- /Qdiag-enable:sc-single-file
- /Qguide-profile:<file|dir>[,[file|dir],...]
- /Qimf-domain-exclusion:classlist[:funclist]
- /Qvec-report6
- /Qvec-report7 (13.1.0)

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。

3.3.2 新しい /Qvec-report7 コンパイラー・オプション (13.1.0)

/Qvec-report7 オプションは、ロード、ストア、型変換、その他についての統計とメトリクスを含む、ループのベクトル化に関するより詳細な情報を出力します。この情報は、指定したループをベクトル化した場合にどの程度パフォーマンスの向上が見込めるか理解するのに役立ちます。

このオプションを指定したときに出力される情報は、http://intel.ly/XeSkW6 (英語) から入手できる新しいベクトル化解析ツールで処理することもできます。

3.4 並列ビルドオプション

Visual Studio* ビルド環境に、マルチコアまたはマルチプロセッサー・システムで未解決の依存性がないソースを並列ビルドできる機能が追加されました。この機能を利用すると、大規模なプロジェクトのビルドに必要な時間を短縮できます。

この機能を有効にするには、プロジェクトのプロパティー・ページを開いて、[Fortran] > [General (全般)] > [Multi-processor Compilation (マルチプロセッサーのコンパイル)] で [Yes (はい)] を選択します。

3.5 既知の問題

3.5.1 Co-Array の問題

Fortran 2008 Co-Array サポートの既知の問題の一覧は、「Co-Array の既知の問題」を参照してください。

3.5.2 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または IA-32 対応アプリケーション用コンパイラーを使用する場合は、この問題は発生しません。

3.6 Microsoft* Visual Studio* 2010/2012 に関する注意事項

Microsoft* Visual Studio* 2010 によりいくつかの変更があります。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 にも適用されます。

3.6.1 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® Fortran の LIB フォルダーを 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 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. プロパティー・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_COMPILER13)\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_COMPILER13)\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 プロパティー・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 プロパティー・ページの場所を参照します。Windows* XP では、通常以下の場所にあります。

```
C:\Documents and Settings\<username>\Local Settings\Application Data
\Microsoft\MSBuild\v4.0
```

Windows* 7 および Windows* 8 では、通常以下の場所にあります。

```
C:\Users\<username>\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。

3.6.2 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* 2010/2012 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。

以前のバージョンの Visual Studio* とは異なり、Visual Studio* 2010/2012 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ー・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。

3.6.3 Windows Server* 2012 で Visual Studio* 2012 のドキュメントを表示できない場合

Windows Server* 2012 で 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* のセキュリティー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 スニッフィングを有効にする] および [アクティブ スクリプト] を有効にします。

3.7 Fortran 2003 および Fortran 2008 機能の概要

インテル® Fortran コンパイラーは、Fortran 2003 の多くの機能をサポートしています。現在サポートしていない Fortran 2003 機能についても、今後サポートしていく予定です。現在のコンパイラーでは、以下の Fortran 2003 機能がサポートされています。

- Fortran 文字セットが次の 8 ビット ASCII 文字を含むように拡張: ~ \ [ ] ` ^ { } | # @
- 最大長 63 文字までの名前
- 最大 256 行の文
- 角括弧 [ ] を (/ /) の代わりに配列の区切り文字として使用可能
- コンポーネント名とデフォルト初期化を含む構造コンストラクター
- 型と文字列長仕様を含む配列コンストラクター
- 名前付き PARAMETER 定数は複素定数の一部
- 列挙子
- 割り当て可能な派生型のコンポーネント
- 割り当て可能なスカラー変数
- 無指定文字長エンティティー
- PRIVATE コンポーネントの PUBLIC 型と PUBLIC コンポーネントの PRIVATE 型
- ALLOCATE と DEALLOCATE の ERRMSG キーワード
- ALLOCATE の SOURCE= キーワード
- 型拡張子
- CLASS 宣言
- 多相型エンティティー
- 継承と関連付け
- 遅延バインディングと抽象型
- 型バインド・プロシージャー
- TYPE CONTAINS 宣言
- ABSTRACT 属性
- DEFERRED 属性
- NON_OVERRIDABLE 属性
- 型バインド・プロシージャーの GENERIC キーワード
- FINAL サブルーチン
- ASYNCHRONOUS 属性および文
- BIND(C) 属性および文
- PROTECTED 属性および文
- VALUE 属性および文
- VOLATILE 属性および文
- ポインター・オブジェクトの INTENT 属性
- 多相オブジェクトのデフォルトの初期化
- 代入文の左辺と右辺の形状または長さが異なる場合に、左辺の割り当て可能な変数を再割り当て (無指定文字長でない場合、`/assume:realloc_lhs` オプションが必要)
- ポインター代入の境界の仕様と境界の再マップ
- ASSOCIATE 構造
- SELECT TYPE 構造
- すべての I/O 文で、次の数値は任意の種類で指定可能: UNIT=, IOSTAT=
- NAMELIST I/O が内部ファイルで許可
- NAMELIST グループのエンティティーの制限の緩和
- 書式付き入出力で IEEE 無限大と NaN の表現方法が変更
- FLUSH 文
- WAIT 文
- OPEN の ACCESS='STREAM' キーワード
- OPEN およびデータ転送文の ASYNCHRONOUS キーワード
- INQUIRE およびデータ転送文の ID キーワード
- データ転送文の POS キーワード
- INQUIRE の PENDING キーワード
- 次の OPEN 数値は任意の種類で指定可能: RECL=
- 次の READ および WRITE 数値は任意の種類で指定可能: REC=、SIZE=
- 次の INQUIRE 数値は任意の種類で指定可能: NEXTREC=、NUMBER=、RECL=、SIZE=

- 開始する新しい I/O が自身以外の内部ファイルを修正しない内部 I/O の場合、再帰 I/O を利用可能
- IEEE 無限大および非数は Fortran 2003 で指定されるフォーマット出力で表示
- BLANK、DECIMAL、DELIM、ENCODING、IOMSG、PAD、ROUND、SIGN、SIZE I/O キーワード
- DC、DP、RD、RC、RN、RP、RU、RZ 書式編集記述子
- I/O フォーマットで、繰り返し指定子が続く場合、P 編集記述子の後のカンマはオプション
- USE 内のユーザー定義演算子名の変更
- USE の INTRINSIC および NON_INTRINSIC キーワード
- IMPORT 文
- 割り当て可能なダミー引数
- 割り当て可能な関数結果
- PROCEDURE 宣言
- 外部プロシージャーを参照する場合、汎用インターフェイス・ブロックの MODULE PROCEDURE からキーワード MODULE を省略
- プロシージャー・ポインター
- ABSTRACT INTERFACE
- PASS 属性と NOPASS 属性
- SYSTEM_CLOCK 組込み関数の COUNT_RATE 引数が任意の種類の REAL で指定可能
- STOP 文の実行で IEEE 浮動小数点例外が発生すると警告を表示
- `/assume:noold_maxminloc` が指定された場合、ゼロサイズの配列の MAXLOC または MINLOC でゼロを返す
- 型問い合わせ組込み関数
- COMMAND_ARGUMENT_COUNT 組込み関数
- EXTENDS_TYPE_OF と SAME_TYPE_AS 組込み関数
- GET_COMMAND 組込み関数
- GET_COMMAND_ARGUMENT 組込み関数
- GET_ENVIRONMENT_VARIABLE 組込み関数
- IS_IOSTAT_END 組込み関数
- IS_IOSTAT_EOR 組込み関数
- MAX/MIN/MAXVAL/MINVAL/MAXLOC/MINLOC 組込み関数 (CHARACTER 引数)
- MOVE_ALLOC 組込み関数
- NEW_LINE 組込み関数
- SELECTED_CHAR_KIND 組込み関数
- 次の組込み関数においてオプションで KIND= 引数を指定可能: ACHAR、COUNT、IACHAR、ICHAR、INDEX、LBOUND、LEN、LEN、TRIM、MAXLOC、MINLOC、SCAN、SHAPE、SIZE、UBOUND、VERIFY
- ISO_C_BINDING 組込みモジュール
- IEEE_EXCEPTIONS、IEEE_ARITHMETIC、IEEE_FEATURES 組込みモジュール
- ISO_FORTRAN_ENV 組込みモジュール

このリリースではまだ実装されていないか、動作しない Fortran 2003 機能の一部を次にリストします。

- ユーザー定義の派生型 I/O
- パラメーター化された派生型
- 初期化式での変形組込み関数 (MERGE や SPREAD など) の使用

インテル® Fortran コンパイラーは、Fortran 2008 規格のいくつかの機能もサポートしています。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
  - 組込みプロシージャー: ATOMIC_DEFINE、ATOMIC_REF、IMAGE_INDEX、LCOBOUND、NUM_IMAGES、THIS_IMAGE、UCOBOUND
- CONTIGUOUS 属性
- ALLOCATE の MOLD キーワード
- DO CONCURRENT
- OPEN の NEWUNIT キーワード
- G0 および G0.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
- 組込みプロシージャー: 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ー・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はない場合、仮引数に INTENT (IN) 属性が含まれていなければなりません。

4 インテル® マス・カーネル・ライブラリー (インテル® MKL)

このセクションでは、インテル® マス・カーネル・ライブラリー (インテル® MKL) の変更点、新機能、および最新情報をまとめています。

4.1 インテル® MKL 11.0 Update 3 の新機能

- 報告された問題の修正 – 修正リスト
- BLAS:
  - Intel® Optimized MP LINPACK Benchmark for Clusters パッケージを HPL 2.1 に更新
- スパース BLAS:
  - インテル® アドバンスト・ベクトル・エクステンション 2 (インテル® AVX2) における DCOOMM のパフォーマンスが向上
- LAPACK:
  - ?LASET、?LACPY、?LANGE、?LANSY を並列化

- FFT:
  - インテル® AVX2 における複素数-複素数の 2 の累乗 FFT のパフォーマンスが向上
- VSL:
  - インテル® AVX2 における SFMT19937 基本乱数ジェネレーター (BRNG) のパフォーマンスが向上
- クラスター FFT:
  - ハイブリッド・モードのクラスター FFT のパフォーマンスが向上
- データ・フィッティング:
  - インテル® Xeon® プロセッサー X5570 およびインテル® Xeon® プロセッサー E5-2690 における線形および 3 次 Hermite/Bessel/Akima スプライン用 df?construct1d 関数のパフォーマンスが向上

4.2 インテル® MKL 11.0 Update 2 の新機能

- インテル® MKL Extended Eigensolver (拡張固有値ソルバー) を追加:
  - インテル® MKL Extended Eigensolver は、密行列、LAPACK 帯行列、疎行列 (CSR) 形式の標準定値固有値問題および一般化定値固有値問題を解くハイパフォーマンスなパッケージです。Feast という革新的な高速かつ安定した数値アルゴリズムをベースにしています (「権利の帰属」セクションを参照)。
- スパース BLAS:
  - 0 ベースの DCSRMM のパフォーマンスが大幅に向上
- LAPACK:
  - ?(OR/UN)G(LQ/QL/QR/RQ) 関数の並列バージョンのパフォーマンスが大幅に向上
- ScaLAPACK:
  - バージョン 2.0.2 にアップデート。新しい関数を追加:
    - P?HSEQR: 非対称固有値問題
    - P?SYEVR/P?HEEVR: MRRR (Multiple Relatively Robust Representations) アルゴリズム
- クラスター FFT:
  - FFTW2 ラッパーを含む多次元クラスター FFT での転置順を実装
- VSL:
  - VSL 対数正規乱数ジェネレーター (RNG) で累積分布逆関数 (ICDF) メソッドをサポート
  - データ・フィッティングおよび VSL 要約統計量関数の宣言に const 指定子を追加
- データ・フィッティング:
  - インテル® Xeon® プロセッサー X5570 およびインテル® Xeon® プロセッサー E5-2600 におけるスカラーケース (補間位置の数が 1) とベクトルケースのデフォルト/準一様分割、ソートされた補間位置用の df[d/s]Interpolate1D、df[d/s]InterpolateEx1D、df[d/s]SearchCells1D、df[d/s]SearchCellsEx1D ルーチンのパフォーマンスが向上
  - データ・フィッティング・ルーチンのパラメーターの正当性確認を無効にすることにより、補間位置の数が少ない (12 個未満) 場合のパフォーマンスを向上させる DF_DISABLE_CHECK_FLAG パラメーターを dfiEditVal エディターでサポート
- 転置:
  - アウトオブプレース行列転置 (mkl_?omatcopy2) を並列化。パフォーマンスが大幅に向上
- サービス関数:
  - インテル® MKL メモリー・アロケーターで使用されるピークメモリーの量に関する情報を提供する mkl_peak_mem_usage 関数を追加

    - インテル® MKL メモリー・アロケーターの機能を標準 C ライブラリーのメモリー割り当て API に拡張する mkl_calloc 関数と mkl_realloc 関数を追加
- SMP LINPACK を拡張 (残差チェック):
    - 失敗が検出されるとエラーコード 1 を返し、生成された残差が精度チェックをパスするかどうか結論を出力します。条件付き数値再現性モードがオフの場合、同じ行列で実行ごとに残差が多少異なることがあるので、注意してください。残差チェックは結果が信頼できることを保証します。

4.3 インテル® MKL 11.0 Update 1 の新機能

- スパース BLAS
    - 複素共役転置とエルミートの CSRMV 機能を最適化
- PARDISO
    - エルミート行列の対数値の虚部を無視
- クラスター FFT
    - ハイブリッド・クラスター FFT (MPI + OpenMP*) のパフォーマンスが向上 (最大 2 倍)
    - よりコミュニケーションの少ない新しいクラスター FFT アルゴリズムを 1D FFT 用に実装 - 有効にするには環境変数 MKL_CFFT_SOI_ENABLE を YES または 1 に設定 (詳細はインテル® MKL のドキュメントを参照)
- VSL
    - ユーザーが定義した平均で記述統計の推定値を計算する VSL_SS_METHOD_FAST_USER_MEAN メソッドのサポートを追加
    - インテル® Xeon® プロセッサー E5-2690 において記述統計の推定値を計算する VSL_SS_METHOD_FAST メソッドのパフォーマンスが向上
- 転置
    - 第 2 世代インテル® Core™ マイクロアーキテクチャーにおけるアウトオブプレース転置のパフォーマンスが向上 (最大 7 倍)
- サービス関数
    - 古い名前の 7 つのサービス関数を削除 (詳細は http://intel.ly/OqbZEL (英語) を参照)

4.4 インテル® MKL 11.0 の新機能

- 条件付きビット単位演算の再現性 (CBWR): コード分岐の選択により柔軟性を持たせ、アルゴリズムに決定性があることを保証して、パフォーマンスと再現性のある結果のバランスを取るインテル® MKL の新しい機能です。詳細は、『インテル® MKL ユーザーガイド』を参照してください。また、CBWR ナレッジベースの記事 (http://intel.ly/P4yRXR) (英語) も参照してください。
- インテル® MKL は、新しい FMA3 命令を含む、新しいインテル®アドバンスト・ベクトル・エクステンション 2 (インテル® AVX2) を使用した最適化もサポートします。詳細は、インテル® AVX2 のサポートに関するナレッジベースの記事 (http://intel.ly/PVmq3h) (英語) を参照してください。
- BLAS:
    - インテル® アドバンスト・ベクトル・エクステンション (インテル® AVX) 対応の 64 ビット・プログラムで DSYRK/SSYRK のパフォーマンスが向上
- LAPACK:
    - [S/D]GETRF、[S/D]POTRF、[S/D]GEQRF、[S/D]GELQF、[S/D]GEQLF、[S/D]GERQF をインテル® MIC アーキテクチャーのネイティブ実行用に最適化
    - LAPACK バージョン 3.4.1 のサポートを追加
- FFT:
    - 記述子あたりのスレッド数を制限する構成パラメーター DFTI_THREAD_LIMIT を追加
    - 64 ビット整数で指定されたサイズの 1D 実数-複素数変換のサポートを追加

- VML /VSL:
  - MT19937、MT2203、MRG32k3a BRNG、離散一様分布および幾何分布 RNG をインテル® MIC アーキテクチャーのネイティブ実行用に最適化
  - インテル® アドバンスト・ベクトル・エクステンション (インテル® AVX) における viRngGeometric のパフォーマンスが向上
  - データ適合 Integrate1d 関数にスレッド化を実装
- 転置:パフォーマンスを大幅に向上させるために単精度と倍精度のリーディング・ディメンジョンが行列サイズより大きな正方行列のインプレース転置を並列化
- スレッド制御の柔軟性を高めるローカルスレッド制御関数 (mkl_set_num_threads_local) を実装
- リンク・ライン・アドバイザー:
  - アーキテクチャー (IA-32/インテル® 64) とインターフェイス・レイヤー (LP64/ILP64) を選択するアドバイザー機能を追加

4.5 推奨されていない (古い) 機能と削除された機能

詳細は、インテル® MKL の機能に関する記事 (http://intel.ly/LkZKGL (英語)) を参照してください。

- インテル® SSE2 拡張命令セットをサポートしないプロセッサー (インテル® Pentium® III 以前) でのインテル® MKL のサポートを終了
- Windows* の PATH 環境変数をインストール中に設定しないように変更。インテル® MKL の DLL を使用するアプリケーションを実行する場合は、インテル® MKL の DLL が含まれているフォルダーが PATH 環境変数に含まれていることを確認する必要があります。
- インテル® MKL GNU Multiple Precision* (GMP) 関数インターフェイスを削除
- タイミング関数 mkl_set_cpu_frequency() を無効化 - インテル® MKL リファレンス・マニュアルで説明されているように、mkl_get_max_cpu_frequency()、mkl_get_clocks_frequency()、mkl_get_cpu_frequency() を使用してください。
- MKL_PARDISO 定数を削除 - mkl_domain_set_num_threads() 関数で PARDISO ドメインを指定する場合は MKL_DOMAIN_PARDISO を使用してください。
- インテル® MKL 10.2 Update 4 の畳み込み関数と相関関数の特別な後方互換関数を削除
- Windows* 版のインテル® MKL およびインテル® コンパイラーから OpenMP* スタティック・ランタイム・ライブラリーを削除
- ドキュメント:
  - HTML 形式のインテル® MKL リファレンス・マニュアルの提供を終了

4.6 既知の問題

本リリースにおける既知の制限事項の詳細なリストは、http://intel.ly/ptEfAP (英語) を参照してください。

4.7 権利の帰属

エンド・ユーザー・ソフトウェア使用許諾契約書 (End User License Agreement) で言及されているように、製品のドキュメントおよび Web サイトの両方で完全なインテル製品名の表示 (例えば、"インテル® マス・カーネル・ライブラリー") とインテル® MKL ホームページ (www.intel.com/software/products/mkl (英語)) へのリンク/URL の提供を正確に行うことが最低限必要です。

インテル® MKL の一部の基となった BLAS の原版は http://www.netlib.org/blas/index.html (英語) から、LAPACK の原版は http://www.netlib.org/lapack/index.html (英語) から入手できます。LAPACK の開発は、E. Anderson、Z. Bai、C. Bischof、S. Blackford、J. Demmel、J. Dongarra、J. Du Croz、A. Greenbaum、S. Hammarling、A. McKenney、D. Sorensen らによって行われました。

LAPACK 用 FORTRAN 90/95 インターフェイスは、http://www.netlib.org/lapack95/index.html (英語) にある LAPACK95 パッケージと類似しています。すべてのインターフェイスは、純粋なプロシージャー用に提供されています。

インテル® MKL クラスター・エディションの一部の基となった ScaLAPACK の原版は http://www.netlib.org/scalapack/index.html (英語) から入手できます。ScaLAPACK の開発 は、L. S. Blackford、J. Choi、A. Cleary、E. D'Azevedo、J. Demmel、I. Dhillon、J. Dongarra、S. Hammarling、G. Henry、A. Petitet、K. Stanley、D. Walker、R. C. Whaley らによって行われました。

インテル® MKL の PARDISO は、バーゼル大学 (University of Basel) から無償で提供されている PARDISO 3.2 (http://www.pardiso-project.org (英語)) と互換性があります。

本リリースのインテル® MKL の一部の FFT 関数は、カーネギーメロン大学からライセンスを受けて、SPIRAL ソフトウェア生成システム (http://www.spiral.net/ (英語)) によって生成されました。SPIRAL の開発は、Markus Püschel、José Moura、Jeremy Johnson、David Padua、Manuela Veloso、Bryan Singer、Jianxin Xiong、Franz Franchetti、Aca Gacic、Yevgen Voronenko、Kang Chen、Robert W. Johnson、Nick Rizzolo らによって行われました。

インテル® MKL Extended Eigensolver の機能は、Feast Eigenvalue Solver 2.0 (http://www.ecs.umass.edu/~polizzi/feast/) をベースにしています。

5　著作権と商標について

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を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、その欠陥や故障によって人身事故が発生するようなアプリケーションでの使用を想定した設計は行われていません。

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ないでください。

本書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。

本書で紹介されている注文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、インテルの Web サイトを参照してください。
http://www.intel.com/design/literature.htm

インテル・プロセッサー・ナンバーはパフォーマンスの指標ではありません。プロセッサー・ナンバーは同一プロセッサー・ファミリー内の製品の機能を区別します。異なるプロセッサー・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。

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

* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。