

インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* インストール・ガイドおよび リリースノート

2015 年 4 月 1 日

目次

1 概要.....	4
1.1 Update 3.....	4
1.2 Update 2.....	4
1.3 Update 1.....	4
1.4 インテル® Visual Fortran Composer XE 2013 SP1 以降 (インテル® Parallel Studio XE 2015 Composer Edition での変更).....	4
1.5 製品の内容.....	5
1.6 動作環境.....	5
1.6.1 Visual Studio* 2008 はサポートされていません.....	7
1.6.2 Windows* XP はサポートされていません.....	7
1.7 ドキュメント.....	7
1.7.1 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.....	7
1.7.2 Windows Server* 2012 の Microsoft* Internet Explorer* 10 でドキュメントが 表示されない問題.....	7
1.8 最適化に関する注意事項.....	8
1.9 サンプル.....	8
1.10 日本語サポート.....	8
1.11 テクニカルサポート.....	8
2 インストール.....	9
2.1 インストール前の準備.....	9
2.1.1 インストールに必要なソフトウェア.....	9
2.2 インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS) のインストール.....	9
2.3 オンライン・インストーラー.....	9
2.3.1 オンライン・インストーラーによりダウンロードされるコンテンツの格納... 10	10

2.4	インストール	10
2.4.1	インストール後の再起動を推奨	10
2.4.2	クラスターでのインストール	10
2.4.3	ライセンスサーバーの使用	10
2.4.4	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	11
2.5	インテル® Software Manager	11
2.6	製品の変更、更新、削除	11
2.7	サイレント・インストール/アンインストール	12
2.7.1	非インタラクティブ・カスタム・インストールのサポート	12
2.8	インストール先フォルダー	12
2.9	既知の問題と変更点	13
3	インテル® Visual Fortran コンパイラー	14
3.1	互換性	14
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)	14
3.1.2	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	15
3.1.3	Fortran Expression Evaluator	15
3.2	割り当て多相コンポーネントの代入に関する新しいライブラリー・サポート (15.0.2)	15
3.3	新規および変更されたコンパイラー機能	15
3.3.1	Fortran 2003 の機能	15
3.3.2	Fortran 2008 の機能	15
3.3.3	OpenMP* 機能	16
3.3.4	新しい宣言子と追加された宣言子	17
3.3.5	その他の機能	17
3.3.6	Co-Array (13.0)	17
3.3.7	派生型のコンポーネントでの ATTRIBUTES ALIGN 宣言子の指定 (13.0.1)	18
3.3.8	ファイル・バッファリング動作の変更 (13.1)	18
3.3.9	スタティック解析は非推奨 (廃止予定)	19
3.3.10	Fortran ライブラリー・バージョンを取得するための新しいランタイムルーチン	19
3.3.11	IA-32 およびインテル® 64 アーキテクチャー向けインテル® 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) 命令セットをサポート (インテル® コンパイラー 15.0.1)	19
3.3.12	SIMD ループ宣言子で MIN/MAX リダクションをサポート	19
3.4	新規および変更されたコンパイラー・オプション	20

3.4.1	インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* の 新規および変更されたコンパイラ・オプション	20
3.5	Visual Studio* 統合の変更点	21
3.5.1	新しいプロジェクトでは DLL ライブラリーがデフォルト (14.0)	21
3.5.2	並列ビルドオプション (13.1)	22
3.5.3	Microsoft* Visual Studio* IDE でソースコードのナビゲーションが向上	22
3.5.4	Microsoft* Visual Studio* IDE での最適化レポートオプションの変更	22
3.5.5	Microsoft* Visual Studio* のオンラインヘルプ形式の変更	22
3.5.6	[ツール] > [オプション] および [プロジェクト] メニューの項目名の変更	22
3.5.7	既存のコードからインテル® Fortran プロジェクトを新規作成	23
3.6	既知の問題	23
3.6.1	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	23
3.6.2	Microsoft* Visual Studio* 2012/2013 のみの環境でデバッグに失敗する問題	23
3.6.3	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	23
3.6.4	Windows* 7 のアップデートにより Visual Studio* 2010 プロジェクトのビル ド時に LNK1123 エラーが発生する	24
3.6.5	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まだ完全に実装さ れていない	24
3.7	Microsoft* Visual Studio* 2010/2012/2013 に関する注意事項	24
3.7.1	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	24
3.7.2	プロジェクトの依存関係の調整	26
3.7.3	Windows Server* 2012 と Windows Server* 2008 で Visual Studio* 2012 のド キュメントを表示できない場合	26
3.8	Fortran 2003 および Fortran 2008 機能の概要	27
4	インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャー向けイ ンテル® Debugger Extension	27
4.1	機能	28
4.2	インテル® Debugger Extension の使用	28
4.3	ドキュメント	28
4.4	既知の問題	28
5	インテル® MKL	29
5.1	インテル® MKL 11.2 Update 3 の新機能	29
5.2	インテル® MKL 11.2 Update 2 の新機能	30
5.3	インテル® MKL 11.2 Update 1 の新機能	31
5.4	インテル® MKL 11.2 の新機能	32
5.5	注意事項	35

5.6 既知の問題.....	35
5.7 権利の帰属.....	35
6 著作権と商標について	36

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。リリースノートの最新アップデートについては、インテル® ソフトウェア開発製品レジストレーション・センターにリストされているリリースノートを参照してください。

インテル® Parallel Studio XE Composer Edition は統合的なソフトウェア開発ツールであり、各コンポーネントは異なるライセンスの下で提供されます。詳細は、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](#)」を参照してください。

1.1 Update 3

- インテル® Visual Fortran コンパイラーが 15.0.3 にアップデート
- [インテル® マス・カーネル・ライブラリー\(インテル® MKL\) 11.2 Update 3](#)
- 報告された問題の修正

1.2 Update 2

- [割り当て多相コンポーネントの代入に関する新しいライブラリー・サポート](#)
- Microsoft* Visual Studio* Community 2013 をサポート
- インテル® Visual Fortran コンパイラーが 15.0.2 にアップデート
- [インテル® MKL 11.2 Update 2](#)
- 報告された問題の修正

1.3 Update 1

- [IA-32 およびインテル® 64 アーキテクチャー向けインテル® アドバンスド・ベクトル・エクステンション 512 \(インテル® AVX-512\) 命令セットをサポート \(インテル® コンパイラー 15.0.1\)](#)
- [\[ツール\]>\[オプション\] および \[プロジェクト\] メニューの項目名の変更](#)
- 日本語版を含む最初のアップデート
- [既存のコードからインテル® Fortran プロジェクトを新規作成](#)
- [SIMD ループ宣言子で MIN/MAX リダクションをサポート](#)
- インテル® Visual Fortran コンパイラーが 15.0.1 にアップデート
- [インテル® MKL 11.2 Update 1](#)
- 報告された問題の修正

1.4 インテル® Visual Fortran Composer XE 2013 SP1 以降 (インテル® Parallel Studio XE 2015 Composer Edition での変更)

- インテル® Visual Fortran コンパイラーが [バージョン 15.0 にアップデート](#)
 - [新しい最適化レポートのインターフェイス、構造、オプション](#) (既存の /Qopt-report、/Qvec-report、/Qopenmp-report、および /Qpar-report オプションを使用しているユーザーは、インテル® C++ コンパイラーのユーザー・リファレンス・ガイドで詳細を確認することを強く推奨します。)

- レポート情報をソースに統合し関連領域へのハイパーリンクとともに表示する最適化レポートの新しい IDE 統合。詳細は、ユーザー・リファレンス・ガイドを参照してください。
- [OpenMP* 4.0 の機能を追加サポート](#)
- [オンライン・インストーラーでのカスタム・インストール設定](#)
- [PGO によるスレッドセーフなプロファイル生成が可能](#)
- [PGO .dyn ファイル名にカスタム・プリフィックスを追加する新しい INTEL_PROF_DYN_PREFIX 環境変数](#)
- [スタティック解析は非推奨 \(廃止予定\)](#)
- Windows* XP はサポートされていません
- Microsoft* Visual Studio* 2008 はサポートされていません
- [インテル® MIC アーキテクチャー向けインテル® Debugger Extension がバージョン 7.7-8.0 にアップデート](#)
- [インテル® MKL がバージョン 11.2 にアップデート](#)
- 報告された問題の修正

1.5 製品の内容

インテル® Parallel Studio XE 2015 Update 3 Composer Edition for Fortran Windows* は、次のコンポーネントで構成されています。

- インテル® Visual Fortran コンパイラー XE 15.0.3。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® MKL 11.2 Update 3
- インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャー向けインテル® Debugger Extension
- 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* 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

1.6 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 IA-32 またはインテル® 64 アーキテクチャー・プロセッサをベースとするコンピューター (インテル® Pentium® 4 プロセッサ以降、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に、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™ コプロセッサ

- [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 または 2013 が必要
- Microsoft* Windows* 7、Microsoft* Windows* 8、Microsoft* Windows* 8.1、Microsoft* Windows Server* 2012、Microsoft* Windows Server* 2008 SP2 (IA-32 アーキテクチャーのみ)、Microsoft* Windows Server* 2008 (R2 SP1)、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* 2013、Visual Studio* 2012、Visual Studio* 2010、または Visual Studio* 2010 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* 2013 Professional Edition 以上
 - Microsoft* Visual Studio* Community 2013 以上
 - Microsoft* Visual Studio* 2012 Professional Edition 以上
 - Microsoft* Visual Studio* 2010 Professional Edition 以上 (C++ コンポーネントがインストールされていること)
 - Microsoft* Visual Studio* 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 [1]
- 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 Studio* Express 2013 for Windows Desktop
 - Microsoft* Visual Studio* Express 2012 for Windows Desktop
 - Microsoft* Visual C++* 2010 Express Edition [2]
- 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 Studio* Express 2013 for Windows Desktop
 - Microsoft* Visual Studio* Express 2012 for Windows Desktop
 - Microsoft* Windows* Software Development Kit for Windows* 8.1
 - 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* 2010 Shell ベースのインテル® Visual Fortran 開発環境は、インテル® Parallel Studio XE 2015 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
2. Microsoft* Visual C++* 2010 Express Edition と Intel® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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。
 3. Intel® Visual Fortran コンパイラーは、デフォルトで、Intel® 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・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・プロセッサ上で動作するコードを生成できます。ただし、Intel® MKL を呼び出すアプリケーションでは Intel® 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、Intel 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。

1.6.1 Visual Studio* 2008 はサポートされていません

Visual Studio* 2008 のサポートを終了しました。新しいバージョンの Visual Studio* に移行してください。

1.6.2 Windows* XP はサポートされていません

Windows* XP のサポートを終了しました。新しいバージョンの Windows* オペレーティング・システムに移行してください。

1.7 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。

1.7.1 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。「[Using Intel® Visual Fortran to Create and Build Windows*-based Applications](#)」(PDF) を参照してください。

1.7.2 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 の 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、信頼済みサイトのリストに "about:internet" を追加します。オプションで、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" を削除できます。

1.8 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向けには、インテル製マイクロプロセッサ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固有の最適化は、インテル製マイクロプロセッサ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

1.9 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。

1.10 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。

日本語版は、インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 初期リリースの後の Update で提供されます。

日本語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・システムで英語版を使用する場合は、「[Changing Language Setting to see English on a Japanese OS Environment or Vice Versa on Windows](#)」(英語)の説明を参照してください。

1.11 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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 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* 8.1、Windows* 8、Windows* 7 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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ています。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してください。

2.2 インテル® メニーコア・プラットフォーム・ソフトウェア・スタック (インテル® MPSS) のインストール

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ビルドする場合のみインストールする必要があります。インテル® MPSS は、インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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。

2.3 オンライン・インストーラー

デフォルトのダウンロード版インストール・パッケージが、サイズの小さいオンライン・インストーラーになりました。オンライン・インストーラーは、選択したパッケージを動的にダウンロードし、インストールします。このパッケージを使用するには、インターネット接続が必要です。インターネット・プロキシを使用している場合は、プロキシの設定が必要になることがあります。インターネット接続が利用できない環境でインストールする場合は、このオンライン・インストール・パッケージではなく、フルパッケージを利用してくだ

さい。オンライン・インストーラーをダウンロードして実行ファイルとして保存し、コマンドラインから起動することもできます。

2.3.1 オンライン・インストーラーによりダウンロードされるコンテンツの格納

オンライン・インストーラーは、ほかのシステムにコピーしてオフラインで使用できるように、ダウンロードしたコンテンツを標準インストール・パッケージ形式で格納します。デフォルトのダウンロード・ディレクトリーは <ProgramFiles>\intel\downloads です。この場所は、オンライン・インストーラーの "--download-dir [FOLDER]" コマンドライン・オプションで変更できます。オンライン・インストーラーには、インストールしないでパッケージを作成できるダウンロード専用モードも用意されています。このモードは、"--download-only" コマンドライン・オプションで有効になります。

2.4 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。

製品をダウンロードした後、実行ファイル (.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。

2.4.1 インストール後の再起動を推奨

インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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。

2.4.2 クラスタでのインストール

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。

2.4.3 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、「[Licensing: Setting Up the Client for a Floating License](#)」(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。

2.4.4 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* のアップデートをインストールするとき、「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/ja-jp/subscriptions/dd264831\(v=vs.100\).aspx](http://msdn.microsoft.com/ja-jp/subscriptions/dd264831(v=vs.100).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。

2.5 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、「[Intel® Software Improvement Program](#)」(英語)を参照してください。

2.6 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows*
- インテル® Parallel Studio XE 2015 Composer Edition for Windows*
- インテル® Parallel Studio XE 2015 Professional Edition for Windows*
- インテル® Parallel Studio XE 2015 Professional Edition for Fortran Windows*

コンパイラ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® 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 Compilers and Tools (インテル(R) コンパイラおよびツール)] > [Visual Fortran (インテル(R) Visual Fortran)] > [Compiler (コンパイラ)] ダイアログから、使用するコンパイラのバージョンを選択できます。バージョン 13.0 (インテル® Visual Fortran Composer XE 2011) よりも古いコンパイラは、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。

2.7 サイレント・インストール/アンインストール

コンパイラの自動インストール/アンインストールについては、「[Intel® Compilers for Windows* Silent Installation Guide](#)」(英語)を参照してください。

2.7.1 非インタラクティブ・カスタム・インストールのサポート

インテル® Parallel Studio XE 2015 は、「インタラクティブ」インストール中のユーザーの選択肢を(サイレント・インストールに使用できる)設定ファイルに保存する機能をサポートしています。この設定ファイルは、コマンドライン・インストールで次のオプションを使用すると作成されます。

- `--duplicate=config_file_name`: 設定ファイルの名前を指定します。フルパスのファイル名が指定された場合、"`--download-dir`"は無視され、設定ファイルがあるディレクトリにインストール・パッケージが作成されます。
- `--download-dir=dir_name`: 設定ファイルを作成する場所を指定します(オプション)。このオプションを指定しない場合、インストール・パッケージおよび設定ファイルはデフォルトのダウンロード・ディレクトリに作成されます。

```
%Program Files%\Intel\Download\<package_id>
```

次に例を示します。

```
w_fcompxe_online_2015.0.0XX.exe --duplicate=ic15_install_config.ini &
--download-dir "C:\temp\custom_pkg_ic15"
```

設定ファイルおよびインストール・パッケージが"C:\temp\custom_pkg_ic15"に作成されます。

2.8 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_COMPILER15` 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® Parallel Studio XE 2015 Composer Edition for Fortran Windows* を検出できます。

- C:\Program Files\Intel\Composer XE 2015
 - bin
 - ia32
 - ia32_intel64
 - intel64
 - intel64_mic

- sourcechecker
- compiler
 - include
 - ia32
 - intel64
 - mic
 - lib
 - ia32
 - intel64
 - mic
- debugger
- Documentation
- 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 上のコンパイラーでインテル®64 上で動作するアプリケーションをビルドする場合に使用するファイル

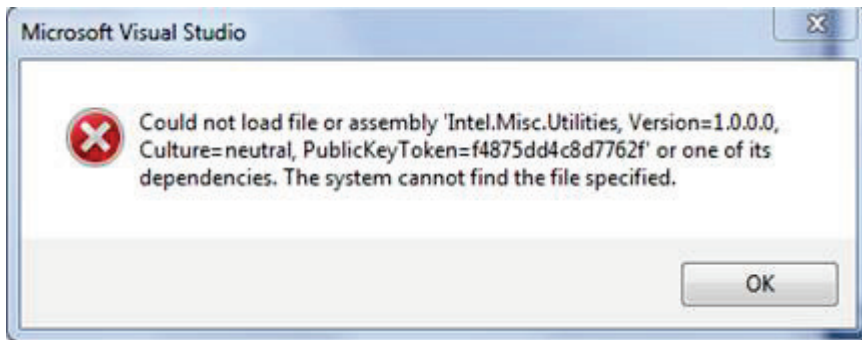
英語以外の Windows* システムにインストールする場合、Program Files フォルダー名が異なる場合があります。インテル® 64 アーキテクチャー・システムでは、フォルダー名は Program Files (X86) またはそれに相当する名前です。

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 2015.nnn (nnn はアップデート番号) に変更されます。

2.9 既知の問題と変更点

- アンロックコードを使用したオフラインのリモート・アクティベーションは削除されました。代わりに、ライセンスファイルまたはライセンス・マネージャーを使用してください。
- インテル® C++ Composer XE 2013 SP1 Update 2 がインストールされているシステムにインテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* をインストールする場合、次の Visual Studio* エラーが表示されることがあります:"

ファイルまたはアセンブリ 'Intel.Misc.Utilities' またはその依存関係の 1 つが読み込めませんでした。指定されたファイルが見つかりません。"



これはインテル® C++ Composer XE 2013 SP1 Update 2 の問題で、以降のリリースでは修正されています。インテル® C++ Composer XE 2013 SP1 Update 3 またはインテル® Parallel Studio XE 2015 Composer Edition for C++ Windows* をインストールすることで解決します。

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 15.0 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ラー間の最適化 (/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

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、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、「[Redistributable Libraries for Intel® Visual Fortran Composer XE](#)」(英語)を参照してください。

3.1.3 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。

3.2 割り当て多相コンポーネントの代入に関する新しいライブラリー・サポート (15.0.2)

割り当て多相コンポーネントを含む派生型の明示的/暗示的な代入時に不正な動作が発生する問題を解決するため、コード生成時に呼び出される新しいライブラリー・ルーチンがインテル® コンパイラー 15.0.2 でサポートされました。つまり、該当する代入を含む Fortran ソースをインテル® コンパイラー 15.0.2 以降でコンパイルする場合は、インテル® Fortran コンパイラー 15.0.2 以降のライブラリーとリンクする必要があります。そうでない場合、_alloc_assign_v2 ルーチンがないためリンクエラーになります。一般に、使用するコンパイラーよりも古いバージョンのライブラリーとリンクすべきではありません。

3.3 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](#) (PDF (英語)) および [Fortran 2008 規格](#) (PDF (英語)) を参照してください。

3.3.1 Fortran 2003 の機能

- パラメーター化された派生型

3.3.2 Fortran 2008 の機能

- BLOCK 構造
- EXECUTE_COMMAND_LINE 組込みサブルーチン

3.3.3 OpenMP* 機能

[OpenMP* 4.0](#) の次の宣言子、節、およびプロシージャがコンパイラーでサポートされます。これらの機能の一部は、暫定仕様に基づきインテル® Visual Fortran Composer XE 2013 Update 3 でサポートされました。また、以前サポートされていたいくつかのキーワード (DECLARE TARGET MIRROR、DECLARE TARGET LINKABLE、MAPTO、MAPFROM、SCRATCH) はサポートされなくなりました。さらに、一部の構文は以前の仕様から変更されています。

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。

SIMD 宣言子:

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

コプロセッサ宣言子:

- OMP TARGET DATA
- OMP TARGET
- OMP TARGET UPDATE
- OMP DECLARE TARGET

その他の宣言子:

- OMP PARALLEL PROC_BIND
- OMP TASKGROUP
- OMP CANCEL
- OMP CANCELLATION POINT

節:

- MAP
- DEPEND

プロシージャ:

- OMP_GET_DEVICE_NUM
- OMP_GET_PROC_BIND
- OMP_SET_DEVICE_NUM

3.3.3.1 KMP_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設定やプロセス・アフィニティ・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。

3.3.3.2 KMP_DYNAMIC_MODE 環境変数による "asat" サポートの廃止

KMP_DYNAMIC_MODE 環境変数による "asat" (自動自己割り当てスレッド) のサポートが廃止されました。将来のリリースで削除される予定です。

3.3.4 新しい宣言子と追加された宣言子

インテル® Parallel Studio XE 2015 Composer Edition 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 OPTIMIZATION_PARAMETER INLINE-MAX-TOTAL-SIZE=N
- ATTRIBUTES OPTIMIZATION_PARAMETER INLINE-MAX-PER-ROUTINE=N

3.3.4.1 BIND(C) と ATTRIBUTES STDCALL を一緒に使用可能

コンパイラー 15.0 では、互換性のあるプロシージャ (宣言に BIND(C) 言語バインド属性を含むプロシージャ) で ATTRIBUTES STDCALL 宣言子を指定することができます。この組み合わせにより、IA-32 アーキテクチャー向け Windows* アプリケーションでは次のような効果が得られます。

- 呼び出しメカニズムが STDCALL に変更されます。プロシージャ終了時のスタックのクリーンアップ方法に影響します。
- BIND 属性の外部サフィックス "@n" が追加されます。n はリターン時にスタックから削除されるバイト数です。

STDCALL によるその他の影響 (値渡しなど) はありません。必要に応じて、(ATTRIBUTES VALUE ではなく) Fortran 標準の VALUE 属性を利用できます。その他のプラットフォームでは、STDCALL と BIND(C) を一緒に指定しても効果はありません。

3.3.5 その他の機能

これらの機能に関する詳細は、コンパイラー・ドキュメントを参照してください。

- 新しい環境変数 INTEL_PROF_DYN_PREFIX。異なる実行で生成される PGO の ".dyn" ファイルを簡単に区別できるように、任意のプリフィックスを追加できます。インストルメントされたアプリケーションを開始する前に、この環境変数に任意の文字列を設定すると、.dyn ファイル名に指定した文字列がプリフィックスとして追加されます。
- SIMD ベクトル内の "レーン ID" を示す新しい `__intel_simd_lane()` 組み込み関数。この組み込み関数は、ショートベクトル・ハイパーオブジェクトのレデューサー実装の記述をサポートします。また、SIMD 対応関数内でリダクション操作の実行を可能にします。

3.3.6 Co-Array (13.0)

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® Parallel Studio XE Cluster Edition 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、「[Windows* 環境での分散 Co-Array アプリケーションのビルドと実行](#)」を参照してください。

現在、インテル® MPI ライブラリー以外の 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。

3.3.6.1 Co-Array 使用する利点とインテル® MPI ライブラリーとの互換性

インテル® Fortran コンパイラ 14 の Co-Array は、インテル® MPI ライブラリー 5.0 と互換性がありません。Co-Array を使用する場合は、インテル® Fortran コンパイラ 15 以上を使用していることを確認してください。そうでない場合は、インテル® MPI ライブラリー 4.x を使用してください。

3.3.7 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 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で使用できません。

3.3.8 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 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ように大きな内部バッファを割り当てます。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際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の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際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 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ようになりました。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 でこのような 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のと同じ方法で、入力のパフォーマンス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

これらの手法は、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。これらの手法を使用すると、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。

3.3.9 スタティック解析は非推奨 (廃止予定)

スタティック解析は非推奨 (廃止予定) の機能です。将来のリリースでは削除される予定です。ご意見やお問い合わせは、[こちら](#)までお寄せください。

3.3.10 Fortran ライブラリー・バージョンを取得するための新しいランタイムルーチン

- `FOR_IFCORE_VERSION` は、Fortran ランタイム・ライブラリー (ifcore) のバージョンを返します。
- `FOR_IFPORT_VERSION` は、Fortran 移植ライブラリー (ifport) のバージョンを返します。

3.3.11 IA-32 および Intel® 64 アーキテクチャー向け Intel® アドバンスド・ベクトル・エクステンション 512 (Intel® AVX-512) 命令セットをサポート (Intel® コンパイラー 15.0.1)

Intel® コンパイラー 15.0.1 では、現在の Intel® メニー・インテグレートッド・コア (Intel® MIC) アーキテクチャー向け Intel® AVX-512 命令のサポートに加えて、Intel® AVX-512 命令対応の IA-32 および Intel® 64 アーキテクチャー・ベースのプロセッサで Intel® AVX-512 命令がサポートされるようになりました。

Intel® AVX-512 命令は、インライン・アセンブリ、`/Q[a]xCORE-AVX512 (Windows*)` または `-[a]xCORE-AVX512 (Linux*/OS X*)` コンパイラー・オプションによりサポートされません。

3.3.12 SIMD ループ宣言子で MIN/MAX リダクションをサポート

Intel® コンパイラー 15.0 では、SIMD ループ宣言子で MIN/MAX リダクションをサポートしました。

```
!DIR$ SIMD REDUCTION (MAX:SIMDMAX)
  DO I = 1, SIZE
    IF (X(I) > SIMDMAX) SIMDMAX = X(I)
  END DO
```

```
!DIR$ SIMD REDUCTION (MIN:SIMDMIN)
  DO I = 1, SIZE
    IF (X(I) < SIMDMIN) SIMDMIN = X(I)
  END DO
```

```
!DIR$ SIMD REDUCTION (MAX:XMAX)
  DO I = 1, SIZE
    XMAX = MAX (XMAX, X(I))
  END DO
```

```
!DIR$ SIMD REDUCTION (MIN:XMIN)
  DO I = 1, SIZE
    XMIN = MIN (XMIN, X(I))
  END DO
```

3.4 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。

3.4.1 インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* の新規および変更されたコンパイラー・オプション

- [/assume:\[no\]std_value](#)
- [/assume:ieee_fpe_flags](#)
- [/fast](#)
- [/Qeliminate-unused-debug-types\[-\]](#)
- [/Qinit:snan](#)
- [/Qopt-dynamic-align\[-\]](#)
- [/Qopt-report](#)
- [/Qprof-gen:\[no\]threadsafe](#)

廃止予定のコンパイラー・オプションのリストは、『インテル® コンパイラー・ユーザー・リファレンス・ガイド』の「コンパイラー・オプション」を参照してください。

3.4.1.1 /assume:std_value をデフォルトに変更

コンパイラー 15.0 では、互換性のないプロシージャー (宣言に BIND(C) 言語バインド属性を含まないプロシージャー) の仮引数に (ATTRIBUTES VALUE ではなく) Fortran 標準の VALUE 属性を指定した場合、デフォルトで Fortran 標準のセマンティクスが適用され、デフォルトの引き渡しメカニズムにより実引数の再定義可能な一時コピーが渡されます。以前のバージョンでは、VALUE は常に実引数を値渡ししていました。コンパイラー 14.0 で /assume:std_value は、標準に準拠したセマンティクスを指定し、/standard-semantics が指定された場合に有効になりました。

3.4.1.2 /standard-semantics と /fp:strict または /fp:except により /assume:ieee_fpe_flags が有効になる

コンパイラー 15.0 では、/standard-semantics と /fp:strict または /fp:except のいずれかが指定されると、/assume:ieee_fpe_flags も有効になります。このオプションは、プロシージャーの開始時に浮動小数点例外状態を保存し、終了時に復元します。保存/復元操作はパフォーマンスを大幅に低下させるため、このオプションは浮動小数点例外を操作または照会するアプリケーションでのみ利用すべきです。Fortran 標準の IEEE_ARITHMETIC、IEEE_EXCEPTIONS、IEEE_FEATURES 組み込みモジュールを使用する場合、インテル® Fortran コンパイラーでは /fp:strict を指定する必要があります。

3.4.1.3 /fast オプションの変更

/fast オプションに /fp:fast=2 が追加されました。このオプションは、パフォーマンスのチューニングを容易にします。

3.4.1.4 新しい /Qinit:snan コンパイラー・オプション

浮動小数点変数をシグナル型 NaN に初期化して、その値が設定される前にフェッチされたらトラップすることで、実行時に初期化されていない変数を探すのに役立つ新しいコマンドライン・オプションです。

3.4.1.5 新しい/Qopt-dynamic-align[-] コンパイラー・オプション

このオプションを指定すると、コンパイラーはベクトル化されたコード、特に反復回数の多いループのパフォーマンスを最大限に引き出すため、入力データの動的アライメントに基づき条件付きの最適化を実装します。ただし、この最適化により、同じ値のアライメントされたデータとアライメントされていないデータで、ビット単位の結果が異なることがあります。このオプションを指定しない場合、コンパイラーはこれらの最適化を行わず、ビット単位の再現性が保持されます。

3.4.1.6 新しい最適化レポートのインターフェイス、構造、オプション(インテル® Parallel Studio XE 2015 Composer Edition)

インテル® Parallel Studio XE 2015 Composer Edition で、4 種類の最適化レポート (/Qopt-report、/Qvec-report、/Qopenmp-report、/Qpar-report) が 1 つの /Qopt-report インターフェイスに統合されました。情報の表示方法、内容、精度が見直され、どの最適化がコンパイラーにより行われたか、最適なパフォーマンスを達成するにはどのようなチューニングを行えばよいか、ユーザーが理解しやすいように変更されました。

並列ビルドの問題により、このレポートはデフォルトで stderr に出力されません。代わりに、各ソース・ファイルごとにレポートを含む出力ファイル (拡張子 .oprpt) が、コンパイラーの出力ディレクトリー (オブジェクト・ファイルが生成されるディレクトリー) に生成されます。この動作を変更するには、/Qopt-report-file オプション (例: /Qopt-report-file:stderr) を使用します。

/Qvec-report、/Qopenmp-report、/Qpar-report オプションは廃止予定ですが、現在は /Qopt-report オプションの対応する値にマップされます。レポートの内容および形式、デフォルトの出力先は新しい opt-report と同じになります。

変更の詳細についてドキュメントを参照することを強く推奨します。詳細は、『インテル® コンパイラー・ユーザー・リファレンス・ガイド』の「コンパイラー・リファレンス」 > 「コンパイラー・オプションのカテゴリーと説明」 > 「最適化レポートオプション」を参照してください。

3.4.1.7 新しい PGO インストルメンテーション・モード /Qprof-gen:[no]threadsafe

PGO インストルメンテーションに、OpenMP* 3.1 などの高度な並列化を含むアプリケーションで PGO データ収集を可能にするモードが追加されました。これにより、IA-32 およびインテル® 64 アーキテクチャーで PGO が強化され、インテル® MIC アーキテクチャーのネイティブ・プログラミング・モデルで PGO がサポートされます。

3.5 Visual Studio* 統合の変更点

3.5.1 新しいプロジェクトでは DLL ライブラリーがデフォルト (14.0)

インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* をインストールした後に Fortran プロジェクトを新規作成すると、DLL 形式のランタイム・ライブラリーを使用するようにプロジェクト・プロパティーが設定されます。これは、Microsoft* Visual C++* の動作と同じですが、インテル® Visual Fortran コンパイラーの以前のバージョンの動作とは異なります。スタティック・ライブラリーを使用する場合は、プロジェクト・プロパティーの [Fortran] > [Libraries (ライブラリー)] > [Runtime Library (ランタイム・ライブラリー)] で変更します。OpenMP* ライブラリー libiomp5md.dll は DLL 形式でのみ提供され、アプリケーションで OpenMP* を使用する場合は、どちらの設定を選択してもこの DLL が使用されます。

3.5.2 並列ビルドオプション (13.1)

Visual Studio* ビルド環境に、マルチコアまたはマルチプロセッサ・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・ページを開いて、[Fortran] > [General (全般)] > [Multi-processor Compilation (マルチプロセッサのコンパイル)] で [Yes (はい)] を選択します。

3.5.3 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。

3.5.4 Microsoft* Visual Studio* IDE での最適化レポートオプションの変更

インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* では、Visual Studio* の [構成プロパティ] > [Fortran] > [Diagnostics (診断)] にある [Optimization Diagnostics Level (最適化診断レベル)]、[Optimization Diagnostics Phase (最適化診断フェーズ)]、[Vectorizer Diagnostics Level (ベクトル化診断レベル)]、[OpenMP Diagnostic Level (OpenMP 診断レベル)]、[Auto-Parallelizer Diagnostic Level (自動並列化診断レベル)] プロジェクト・プロパティの値が更新されました。これらのプロパティを使用している場合は、プロジェクトの [プロパティ ページ] ダイアログから値の更新が必要になることがあります。また、更新後、以前のバージョンのコンパイラーを使用するように設定を変更した場合、再度プロパティの値の更新が必要になることがあります。

3.5.5 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、いくつかの小さな問題が発生することがあります。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*: 目次が表示されなくなることがあります。検索の大文字と小文字は区別されません。
- Safari*: Windows* の反応が遅くなります。

3.5.6 [ツール] > [オプション] および [プロジェクト] メニューの項目名の変更

インテル® Parallel Studio XE 2015 Update 1 から、インテル® コンパイラー関連の一部の項目名が変更されました。

- [ツール] > [オプション] の左ペインにある [Intel Composer XE (インテル(R) Composer XE)] が [Intel Compilers and Tools (インテル(R) コンパイラーおよびツール)] になりました。利用可能な設定 (インクルード・ディレクトリー、コードカバ

レジの設定、パフォーマンス・ライブラリーの設定、その他)は変更されていません。

- [プロジェクト]メニューまたはプロジェクトを右クリックして表示されるコンテキスト・メニューの [Intel Composer XE (インテル(R) Composer XE)] が [Intel Compiler (インテル(R) コンパイラー)] になりました。

3.5.7 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。

3.6 既知の問題

3.6.1 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。

3.6.2 Microsoft* Visual Studio* 2012/2013 のみの環境でデバッグに失敗する問題

Microsoft* Visual Studio* 2012/2013 のみがインストールされている Microsoft* Windows* システムでは、Fortran アプリケーションのデバッグに失敗することがあります。ウォッチ (式の評価) や条件付きブレークポイントなどに失敗します。

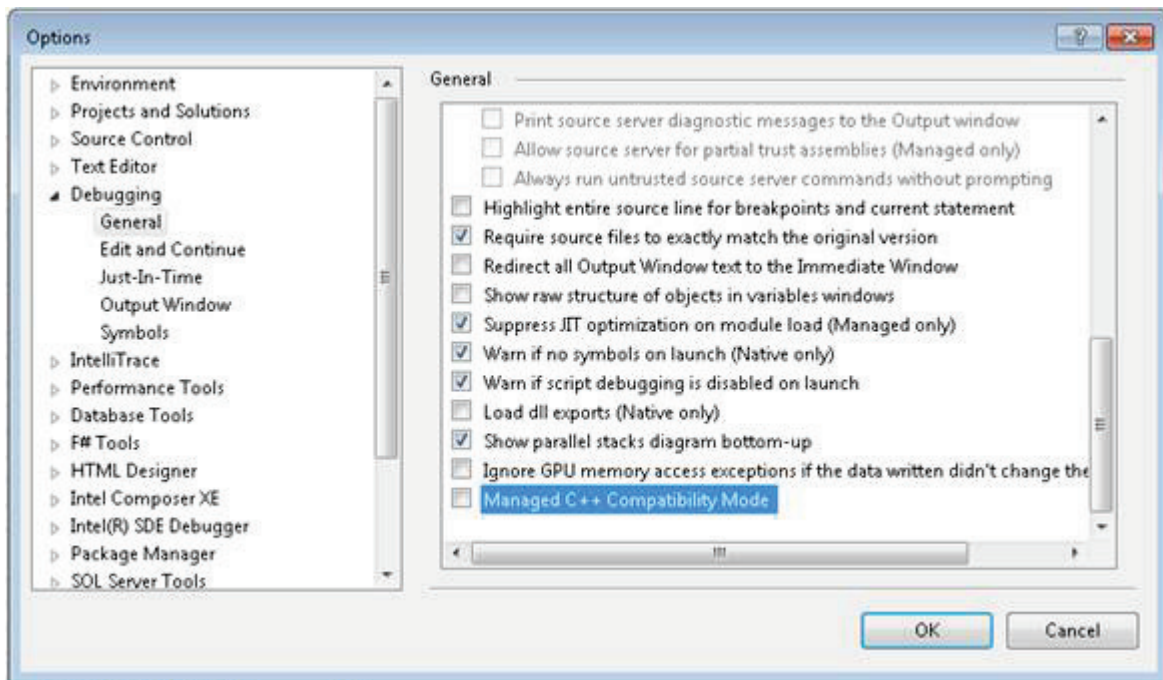
インテル® Parallel Studio XE 2015 Composer Edition for Fortran Windows* は、Fortran アプリケーションをデバッグできるようにするため、Fortran Expression Evaluator (FEE) と呼ばれるデバッガー拡張を提供しています。一部の FEE 機能には、Microsoft* Visual Studio* 2010 ライブラリーが必要です。

1 つの方法として、Microsoft* Visual Studio* 2012/2013 に加えて、Microsoft* Visual Studio* 2010 をインストールすることができます。別の方法として、[こちらから](#) Microsoft* Visual C++* 2010 SP1 再頒布可能パッケージ (x86) をダウンロードしてインストールできます。

3.6.3 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、次の設定を無効します。

[ツール] > [オプション] > [デバッグ] > [全般] > [Managed C++ 互換モード] チェックボックス



マネージ・コード・アプリケーションのプロジェクト・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]

3.6.4 Windows* 7 のアップデートにより Visual Studio* 2010 プロジェクトのビルド時に LNK1123 エラーが発生する

.NET Framework 4.5.1 用の Windows Update または Visual Studio* 2012 をインストールすると、リンク処理で使用される DLL のバージョンが新しくなり、Visual Studio* 2010 リンカーが "LNK1123: COFF への変換中に障害が発生しました: ファイルが無効であるか、または壊れています。" エラーを出力します。この問題を解決するには、[こちら](#)から Microsoft* Visual Studio* 2010 Service Pack 1* をインストールします。

3.6.5 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まだ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まだ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
- パラメーター化された拡張型の表示で FEE 構文エラーが発生

3.7 Microsoft* Visual Studio* 2010/2012/2013 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/ 2013 にも適用されます。

3.7.1 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 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_COMPILER15)\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_COMPILER15)\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* XP では、通常以下の場所にあります。

```
C:\Documents and Settings\\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\\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。

3.7.2 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* 2010/2012/2013 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* 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・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。

3.7.3 Windows Server* 2012 と Windows Server* 2008 で Visual Studio* 2012 のドキュメントを表示できない場合

Windows Server* 2012 と Windows Server* 2008 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] > [セキュリティー] を選択して、[インターネット] ゾーンで [MIME スニфф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。

3.8 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 標準のすべての機能と Fortran 2008 標準の多く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
- 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。
- BLOCK 構造
- EXECUTE_COMMAND_LINE 組込みサブルーチン

4 インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャー向けインテル® Debugger Extension

このセクションでは、インテル® Debugger Extension の変更点、新機能、カスタマイズ、および既知の問題をまとめています。インテル® Debugger Extension は、インテル® メニー・

インテグレートド・コア (インテル® MIC) アーキテクチャー向けのコードのみサポートします。

4.1 機能

- オフロード拡張を使用して、コプロセッサのネイティブ・アプリケーションとホスト・アプリケーションの両方をサポート
- 同時に複数のコプロセッサ・カードをデバッグ (オフロード拡張を使用)

4.2 インテル® Debugger Extension の使用

インテル® Debugger Extension は Microsoft* Visual Studio* IDE のプラグインです。Microsoft* Visual Studio* IDE で定義されたプロジェクトのデバッグを可能にします。インテル® Xeon Phi™ コプロセッサ向けアプリケーションは、ロードして実行することも、アタッチすることもできます。

インテル® Debugger Extension の使用方法は、「[ドキュメント](#)」を参照してください。

4.3 ドキュメント

インテル® デバッガのドキュメントは、以下の場所にあります。

```
<install-dir>\Documentation\en_US|ja_JP\debugger\ ↵  
mic\gdb_quickstart_win.pdf
```

4.4 既知の問題

- オフロードデバッグは Microsoft* Visual Studio* 2012 および Microsoft* Visual Studio* 2013 でのみサポートされています。
- オフロードセクションでは [逆アセンブル] ウィンドウで開始アドレスから 1024 バイトを超える範囲にスクロールすることはできません。
- インテル® MIC アーキテクチャー・アプリケーションの例外処理はサポートされていません。
- アプリケーション実行中のブレークポイントの変更は正しく動作しません。変更されたかのように見えますが、変更が適用されません。
- インテル® MIC アーキテクチャーのネイティブ・アプリケーションの開始はサポートされていません。現在実行中のアプリケーションにアタッチすることはできません。
- Microsoft* Visual Studio* の [スレッド] ウィンドウには、スレッドの凍結、凍結解除、名前変更を行うコンテキスト・メニューがあります。これらのコンテキスト・メニューは、コプロセッサ上のスレッドでは正しく動作しません。
- オフロードセクションの直前にブレークポイントを設定すると、オフロードセクションの最初の文にブレークポイントが設定されます。この動作は、設定したブレークポイントとオフロードセクションの間にホスト用の文がない場合のみ起こります。これは Microsoft* Visual Studio* ブレークポイントの通常の動作ですが、ホストとコプロセッサのコードが混在表示されることがあります。オフロードセクションの不要なブレークポイントは、必要に応じて、手動で無効に (または削除) することができます。
- オフロードセクションを含むインテル® 64 対応アプリケーションのみ、インテル® メニー・インテグレートド・コア・アーキテクチャー向けインテル® Debugger Extension を使用してデバッグすることができます。
- オフロードセクションをステップアウトすると、ホストコードにステップバックせず、(別のイベントが発生しない限り) 停止することなく実行が継続されます。これは意図された動作です。
- 「次のステートメントの設定」機能は、オフロードセクション内では動作しません。

- ブレークポイントがプロジェクトのオフロードセクションにすでに設定されている場合、デバッガーを開始するとアドレスのない境界ブレークポイントが表示されることがありますが、動作には影響しません。
- オフロードセクションでは、次のヒットカウンター条件を含むブレークポイントは動作しません: 「ヒット カウント数が次の数と等しいときに中断」 および 「ヒット カウントが次の数の倍数になったときに中断」
- [逆アセンブル] ウィンドウの次のオプションはオフロードセクション内では動作しません: 「行番号を表示」、「シンボル名の表示」、「ソースコードの表示」
- オフロードセクションの外部で宣言された変数を評価すると誤った値が表示されます。
- 詳細は情報は、出力 (デバッグ) ウィンドウを参照してください。実装されていない機能がリストされるか (上記を参照)、デバッグセッションの設定問題に必要な追加情報が提供されます。このウィンドウを開くには、Microsoft* Visual Studio* で [デバッグ] > [ウィンドウ] > [出力] を選択します。
- オフロードが有効なアプリケーションをデバッグする場合、デバッガーは書き込む前にそのメモリー位置を読み取る代入 (例えば、 $x=x+1$) を含む式を評価しません。([イミディエイト] ウィンドウ や [ウォッチ] ウィンドウなどで) 式を評価する場合、そのような代入は使用しないでください。
- オフロードセクションで条件付きブレークポイントを使用すると、デバッガーがストールすることがあります。条件付きブレークポイントをオフロードセクション内に作成した場合、条件を評価するときにデバッガーがハングアップすることがあります。この問題は現在調査中で、将来のリリースで修正される予定です。
- インテルから提供されているデバッガー拡張の動作 (例えば、実行制御) や出力 (例えば、逆アセンブリー) は Microsoft* Visual Studio* のデバッガーと異なる場合があります。これは、それぞれ実装しているデバッグ手法が異なるためです。デバッグに大きな影響はありません。

5 インテル® MKL

このセクションでは、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。問題の修正については、[こちら](#)を参照してください。

5.1 インテル® MKL 11.2 Update 3 の新機能

- 大規模な SMP システムでスケーリングが向上するように、インテル® MKL のメモリー・マネージャーが拡張されました。
- インテル® MKL のインテル® Xeon Phi™ コプロセッサ・ベースのシステムへの自動オフロードをより細かく制御するため、次の新しいサービス関数が追加されました: `mkl_mic_get_meminfo`、`mkl_mic_get_cpuinfo`、`mkl_mic_set_flags`、`mkl_mic_get_flags`、`mkl_mic_clear_status`、`mkl_mic_get_status`。
- BLAS:
 - すべてのインテル® Xeon® プロセッサで (D/S)SYMV の並列パフォーマンスを向上
 - 64 ビットのインテル® MKL でインテル® アドバンスト・ベクトル・エクステンション (インテル® AVX) の (C/D/S/Z/DZ/SC)ROT パフォーマンスを向上
 - 64 ビットのインテル® MKL でインテル® アドバンスト・ベクトル・エクステンション 2 (インテル® AVX2) の (C/Z)ROT パフォーマンスを向上
 - 64 ビットのインテル® MKL でインテル® AVX2 の k が大きな ?SYRK/?HERK、?SYR2K/?HER2K、?GEMM の並列パフォーマンスを向上

- インテル® Xeon Phi™ コプロセッサで ?SYRK/?HERK、?SYR2K/?HER2K パフォーマンスを向上
- LAPACK:
 - インテル® AVX またはインテル® AVX2 ベースのマルチソケット・システムで特異ベクトル計算の SVD パフォーマンスを向上
 - ピボット演算を使用しない新しい不完全 LU 分解を追加

5.2 インテル® MKL 11.2 Update 2 の新機能

- BLAS:
 - インテル® Xeon Phi™ コプロセッサで $k \gg m$ 、 $k \gg n$ の場合の ?GEMM パフォーマンスを向上
 - 64 ビットのインテル® MKL でインテル® AVX2 の ?HEMM/?SYMM の並列パフォーマンスとシリアルパフォーマンスを向上
 - インテル® AVX2 の ?HERK/?SYRK および ?HER2K/?SYR2K の並列パフォーマンスとシリアルパフォーマンスを向上
 - CBLAS インターフェイスと ?GEMM3M ルーチンに MKL_DIRECT_CALL サポートを追加
 - インテル® AVX-512 の CGEMM パフォーマンスを向上
 - AMD* Opteron* 6000 シリーズで SGEMM および ZGEMM パフォーマンスを向上
 - 64 ビットのインテル® MKL でインテル® AVX2 の CGEMM および ZGEMM のパフォーマンスをわずかに向上
- LAPACK:
 - 固有ベクトルが不要な場合に対称固有値ソルバーのパフォーマンスを最大 3 倍向上
 - 特異ベクトルが必要な場合に ?GESVD パフォーマンスを 2 ~ 3 倍向上
 - 非正方行列でインテル® AVX2 の ?GETRF パフォーマンスを最大 14 倍向上
 - CNR (条件付き数値再現性) が有効な場合と無効な場合の ?GETRF パフォーマンスのギャップを 5% 以下に減少
 - インテル® AVX2 向けのインテル® Optimized LINPACK Benchmark の共有メモリー (SMP) 実装のパフォーマンスを最大 40% 向上
- クラスタ用並列直接法スパースソルバー:
 - 分散 CSR 形式の解による右辺ベクトルの上書き機能を追加
 - 分散 CSR 形式ですべての計算ノードの方程式の解を集約する機能を追加
- インテル® MKL PARDISO:
 - インテル® Xeon Phi™ コプロセッサで全体的なスケーラビリティを大幅に向上
 - インテル® Xeon® プロセッサで解の算出ステップのスケーラビリティを向上
 - アウトオブコア・モードでメモリー・フットプリントを軽減
 - 因数分解ステップ後に入力行列で使用されたメモリーを解放する機能を追加。これにより、反復改善が不要でユーザーによって無効にされている場合、メモリー消費量が軽減されます。
- 拡張固有値ソルバー:
 - インテル® Xeon® プロセッサでパフォーマンスを向上

- VSL:
 - サマリー統計:
 - タスクの次元と観測数がほぼ同じ場合に分散/共分散行列計算および相関行列計算ルーチンのパフォーマンスを向上
 - RNG:
 - インテル® Xeon® プロセッサで Sobol および Niederreiter 準乱数ジェネレーター (RNG) のパフォーマンスを向上
- 畳み込みおよび相関:
 - 3D 畳み込みのパフォーマンスを向上

5.3 インテル® MKL 11.2 Update 1 の新機能

- インテル® MKL for Windows* および Linux* は、現在のインテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャー向けインテル® AVX-512 命令のサポートに加えて、次世代のインテル® マイクロアーキテクチャー Skylake (開発コード名) でインテル® 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) 命令をサポートします。
- BLAS:
 - インテル® マイクロアーキテクチャー Skylake (開発コード名) において次の関数を最適化
 - (D/Z)AXPY、(S/D/C/Z)COPY、DTRMM (三角行列が右辺にあり行列の転置がない場合)
 - IA-32 アーキテクチャーとインテル® 64 アーキテクチャーの両方でインテル® AVX2 の次のレベル 1 BLAS 関数を最適化
 - (S/D)DOT、(S/D)SCAL、(S/D)ROT、(S/D)ROTM、(S/D/C/Z)SWAP、(S/D/SC/DZ)ASUM
 - インテル® AVX2 において ?GEMM のパフォーマンス (シリアルおよびマルチスレッド) が向上 (IA-32 アーキテクチャー)
 - インテル® AVX およびインテル® AVX2 において beta=0 の場合の ?GEMM のパフォーマンスが向上 (インテル® 64 アーキテクチャー)
 - インテル® AVX において DGEMM のパフォーマンス (シリアルおよびマルチスレッド) が向上 (インテル® 64 アーキテクチャー)
- LAPACK:
 - LAPACK バージョン 3.5 をサポート。このバージョンでは次の新機能を追加。
 - rook ピボット・アルゴリズムを含む対称/エルミート LDLT 因数分解ルーチン
 - 直交列を含む縦長行列と横長行列の 2×1 CSD
 - M>N で特異ベクトルが必要ないときの (C/Z)GE(SVD/SDD) のパフォーマンスが向上
- FFT:
 - インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャーにおいて、1D バッチ FFT に自動オフロードモードを追加
 - ハイブリッド (OpenMP*+MPI) クラスタ FFT のパフォーマンスが向上
 - 大きな 1D 実数-複素数変換の精度が向上
- クラスタ用並列直接法スパースソルバー:
 - 同じ並べ替えの多くの因数分解ステップをサポート (maxfct > 1)
- インテル® MKL PARDISO:
 - シュール補行列をサポート (明示的なシュール補行列を得ることおよびシュール補行列により式を解くことを含む)

- スパース BLAS:
 - インテル® マイクロアーキテクチャ Skylake (開発コード名) において SpMV を最適化
 - 行列構造およびインデックスの検証を簡素化する疎行列チェッカー機能をスタンドアロン API として追加 (詳細は、『[インテル® マス・カーネル・ライブラリー \(インテル® MKL\) リファレンス・マニュアル](#)』の「Sparse Matrix Checker Routines」を参照)
 - C/C++ 用スパース BLAS API は定数引数に const 修飾子を使用
- VML:
 - 精度動作を制御する新しい環境変数 MKL_VML_MODE を追加。この環境変数は、VML 関数の動作を制御するために使用可能 (vmlSetMode() 関数のアナログ)

5.4 インテル® MKL 11.2 の新機能

- インテル® ストリーミング SIMD 拡張命令 4.1 (インテル® SSE4.1) およびインテル® ストリーミング SIMD 拡張命令 4.2 (インテル® SSE4.2) 命令セット対応のすべてのインテル® Atom™ プロセッサ向けの最適化を提供
- 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) 命令セットをサポート (BLAS、DFT、VML の最適化は制限あり)
- BLAS および LAPACK ドメインで verbose モードをサポート (インテル® MKL 関数呼び出しの入力引数をキャプチャー可能)
- インテル® MPI ライブラリー 5.0 をサポート
- インテル® MKL を使用して特定の複雑な問題を解く方法を説明する新しいドキュメント、インテル® MKL クックブック (http://software.intel.com/en-us/mkl_cookbook (英語)) を提供
- すべてのプロセッサにおいて小行列の ?GEMM パフォーマンスを向上する MKL_DIRECT_CALL または MKL_DIRECT_CALL_SEQ コンパイル機能を追加 (詳細は、『インテル® マス・カーネル・ライブラリー (インテル® MKL) ユーザーズガイド』を参照)
- インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャにおいて、シングル・ダイナミック・ライブラリー (mkl_rt) をリンクする機能を追加
- カスタマイズ可能なエラーハンドラーを追加。詳細は、『インテル® マス・カーネル・ライブラリー (インテル® MKL) リファレンス・マニュアル』の「mkl_set_exit_handler()」の説明を参照
- リソース共有メカニズムによりインテル® Xeon Phi™ コプロセッサの自動オフロード機能を拡張 (詳細は、『インテル® マス・カーネル・ライブラリー (インテル® MKL) リファレンス・マニュアル』の mkl_mic_set_resource_limit() 関数および MKL_MIC_RESOURCE_LIMIT 環境変数の説明を参照)
- クラスタ用並列直接法スパースソルバー:
 - インテル® MKL PARDISO 直接法スパースソルバーの分散メモリーバージョンである、クラスタ用並列直接法スパースソルバーを追加
 - 分散行列の行列集約ステップのパフォーマンスが向上
 - 複数の因数分解ステップにおける並べ替え情報の再利用が可能に
 - 分散 CSR 形式、分散行列、RHS、分散ソリューシヨンのサポートを追加
 - 複数の右辺が含まれる式の解の算出をサポート
 - 因数分解および解の算出ステップのクラスタサポートを追加
 - ピュア MPI モードのサポートおよびハイブリッド構成での単一 OpenMP* スレッドのサポートを追加

- BLAS:
 - 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) 対応の 64 ビット・プロセッサにおいて ?GEMM のスレッド・パフォーマンスが向上
 - インテル® AVX-512 命令セット用の ?GEMM、?TRSM、DTRMM を最適化
 - インテル® MIC アーキテクチャーにおいて、外積 [large m, large n, small k] および Tall Skinny 型行列 [large m, medium n, small k] の ?GEMM のパフォーマンスが向上
 - インテル® MIC アーキテクチャーにおいて自動オフロードモードの ?TRSM および ?SYMM のパフォーマンスが向上
 - インテル® AVX2 対応の 64 ビット・プロセッサにおいてレベル 3 BLAS 関数のパフォーマンスが向上
 - コンパイル中に MKL_DIRECT_CALL または MKL_DIRECT_CALL_SEQ が定義されている場合、すべてのプロセッサにおいて小行列の ?GEMM パフォーマンスが向上 (詳細は、『インテル® マス・カーネル・ライブラリー (インテル® MKL) ユーザーズガイド』を参照)
 - インテル® SSE4.2、インテル® アドバンスド・ベクトル・エクステンション (インテル® AVX)、およびインテル® AVX2 命令セット対応の 64 ビット・プロセッサにおいて、beta=1、k=1 の場合の DGER および DGEMM のパフォーマンスが向上
 - インテル® AVX-512 命令セット用の (D/Z)AXPY を最適化
 - インテル® AVX2 およびインテル® AVX-512 命令セット用の ?COPY を最適化
 - インテル® AVX-512 命令セット用の DGEMV を最適化
 - インテル® AVX およびインテル® AVX2 対応の 64 ビット・プロセッサにおいて SSYR2K のパフォーマンスが向上
 - すべてのインテル® プロセッサ用の ?AXPBY のスレッド・パフォーマンスが向上
 - インテル® AVX-512 において side=R、uplo={U,L}、transa=N、diag={N,U} の場合の DTRMM のパフォーマンスが向上
- LINPACK:
 - ヘテロジニアス Intel® Optimized MP LINPACK Benchmark for Clusters において行列生成のパフォーマンスが向上
 - Intel® Optimized MP LINPACK Benchmark パッケージのインテル® MIC アーキテクチャー用オフロード・オプションでインテル® AVX2 ホストをサポート
 - インテル® AVX2 対応の 64 ビット・プロセッサにおいて Intel® Optimized MP LINPACK Benchmark for Clusters パッケージのパフォーマンスが向上
- LAPACK:
 - ?(SY/HE)RDB のパフォーマンスが向上
 - 固有ベクトルが必要な場合の ?(SY/HE)EV のパフォーマンスが向上
 - 固有ベクトルが不要な場合の ?(SY/HE)(EV/EVR/EVD) のパフォーマンスが向上
 - 劣決定 (M が N 未満) の場合の ?GELQF、?GELS および ?GELSS のパフォーマンスが向上
 - ?GEHRD、?GEEV および ?GEES のパフォーマンスが向上
 - LAPACKE インターフェイスにおいて NaN チェッカーのパフォーマンスが向上
 - ?GELSX、?GGSPV のパフォーマンスが向上
 - 固有ベクトルが不要な場合の ?(SY/HE)(EV/EVR/EVD) のパフォーマンスが向上

- ?GETRF のパフォーマンスが向上
- $M > N$ で特異ベクトルが必要ないときの (S/D)GE(SVD/SDD) のパフォーマンスが向上
- インテル® MIC アーキテクチャーにおいて自動オフロードモードの ?POTRF UPLO=U のパフォーマンスが向上
- インテル® MIC アーキテクチャーにおいて ?SYRDB の自動オフロードを追加、固有ベクトルが不要な場合に ?SY(EV/EVD/EVR) がスピードアップ
- PBLAS および ScaLAPACK:
 - 大規模な分散ブロッキング係数の P?GEMM ルーチンで自動オフロードが可能に
- スパース BLAS:
 - インテル® AVX-512 命令セット用の SpMV カーネルを最適化
 - スパース BLAS で対角形式を使用する場合のリリースサンプルを追加
 - インテル® SSE4.2、インテル® AVX、およびインテル® AVX2 命令セット対応システムにおいてスパース BLAS レベル 2 およびレベル 3 のパフォーマンスが向上
- インテル® MKL PARDISO:
 - 任意のソルバーステージで後から使用できるようにインテル® MKL PARDISO ハンドルをディスクに格納する機能を追加
 - 非対称行列およびアウトオブコア・モードにピボット制御のサポートを追加
 - 非対称行列およびアウトオブコア・モードに対角抽出のサポートを追加
 - 非線型方程式の反復ソルバーとしてインテル® MKL PARDISO を使用するサンプルを追加
 - 反復改善が無効な場合、因数分解ステージ後にオリジナル行列で割り当てたメモリーを解放する機能を追加
 - 並べ替えアルゴリズムのアウトオブコア (OOC) 部分サイズのメモリー推定向上により、OOC モードの因数分解ステップのパフォーマンスが向上
 - インテル® MKL PARDISO の出力メッセージを変更
 - 構造対称の因数分解中のゼロピボットをサポート
- ポアソン・ライブラリー:
 - 線形方程式を解く前提条件としてインテル® MKL ポアソン・ライブラリーを使用するサンプルを追加
- 拡張固有値ソルバー:
 - 出力メッセージを変更
 - サンプルを変更
 - スパース問題を解くための事前定義インターフェイスに入力および出力 iparm パラメーターを追加
- FFT:
 - インテル® AVX-512 命令セット用の FFT を最適化
 - インテル® MIC アーキテクチャーにおいて 2 のべき乗でない長さのパフォーマンスが向上
- VML: 各ベクトル要素の小数部を計算する v[d]s]Frac 関数を追加
- VSL RNG:
 - 二項乱数ジェネレーターで ntrial=0 をサポート
 - インテル® MIC アーキテクチャーにおいて MRG32K3A および MT2203 BRNG のパフォーマンスが向上
 - インテル® AVX およびインテル® AVX2 命令セット対応のプロセッサにおいて MT2203 BRNG のパフォーマンスが向上

- VSL サマリー統計:
 - グループ化された/プールされた平均推定 (VSL_SS_GROUP_MEAN/VSL_SS_POOLED_MEAN) をサポート
- データ・フィッティング: ブレークポイント数が 2 または 3 の場合の自然 3 次スプライン構築関数の不正な動作を修正
- インテル® MKL 環境変数で指定したすべての設定を無視するインテル® MKL モードを追加
 - `mkl_set_env_mode()` ルーチン (インテル® MKL 固有のすべての環境設定を無視するようにインテル® MKL に指示) を呼び出してモードをセットアップすると、`MKL_NUM_THREADS`、`MKL_DYNAMIC`、`MKL_MIC_ENABLE` その他のすべてのインテル® MKL 関連の環境変数が無視される; 必要な引数は `mkl_set_num_threads()` や `mkl_mic_enable()` などのインテル® MKL サービスルーチンから設定可能

5.5 注意事項

- インテル® MKL では、インストールするコンポーネントを選択できるようになりました。PGI* コンパイラー、Compaq* Visual Fortran コンパイラー、SP2DP インターフェイス、BLAS95 および LAPACK95 インターフェイス、クラスターサポート (ScaLAPACK および Cluster DFT)、インテル® MIC アーキテクチャーのサポートに必要なコンポーネントは、インストール時に明示的に選択しない限りインストールされません。
- インテル® MKL クラスター・コンポーネント (ScaLAPACK および Cluster DFT) では、アライメントされていない CNR は利用できません。
- BOOST/uBLAS および Java* でのインテル® MKL の使用例は、製品パッケージからは削除され、以下の記事 (英語) からダウンロードすることができます。
 - [How to use Intel® MKL with Java*](#)
 - [How to use BOOST* uBLAS with Intel® MKL](#)
- API シンボル、引数の順序、リンク行はインテル® MKL 11.2 Beta Update 2 で変更されました。(詳細は、『インテル® マス・カーネル・ライブラリー (インテル® MKL) ユーザーズガイド』を参照してください)
- 廃止予定の項目は、[インテル® MKL 11.2 で廃止予定の項目](#) (英語) を参照してください。

5.6 既知の問題

- Windows* で大規模行列を自動オフロードするとデータ破損が発生したりクラッシュすることがあります。COI に問題があります:HSD4868293 (クリティカル)。COI は Windows* で 4GB を超えるバッファおよび 2M ページを割り当てることができません

回避方法: `MKL_MIC_MAX_MEMORY=3G` に設定します。

注: この問題はインテル® MPSS 3.3 で解決されます。

既知の制限事項の詳細なリストは、インテル® デベロッパー・ゾーンにある「[Intel® MKL Article List](#)」 (英語) を参照してください。

5.7 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。

6 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、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場合があります。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、インテルの Web サイト (<http://www.intel.com/design/literature.htm>) 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 コンパイラーおよびインテル® MKL は、インテルのエンド・ユーザー・ソフトウェア使用許諾契約書 (EULA) の下で提供されます。

GNU* プロジェクト・デバッガー (GDB) は、General GNU Public License GPL V3 の下で提供されます。

Intel、インテル、Intel ロゴ、Intel Atom、Intel Xeon Phi、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。

© 2015 Intel Corporation. 無断での引用、転載を禁じます。