

インテル® Visual Fortran コンパイラー 11.1 Windows* 版 プロフェッショナル・エディション インストール・ガイドおよびリリースノート

資料番号: 321417-002JA
2010年3月24日

目次

1	概要.....	4
1.1	変更履歴.....	4
1.2	製品の内容.....	5
1.3	動作環境.....	5
1.4	ドキュメント.....	7
1.5	サンプル.....	7
1.6	日本語サポート.....	7
1.7	テクニカルサポート.....	8
2	インストール.....	8
2.1	インストール前の準備.....	8
2.1.1	64ビット・アプリケーション用の Visual Studio の設定.....	8
2.1.2	Microsoft Windows Vista または Microsoft Windows 7 でのインストール.....	8
2.2	インストール.....	9
2.2.1	IMSL Fortran 数値計算ライブラリーのインストール.....	9
2.2.2	Microsoft Visual Studio 2005 Premier Partner Edition はサポートされて いません.....	9
2.2.3	Microsoft Visual Studio 2005 の管理者権限に関するメッセージダイアログ.....	10
2.3	製品の変更、更新、削除.....	10
2.4	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	型バインド・プロシージャの誤った派生型レイアウト.....	12
3.1.2	BIND(C) プロシージャの不適切な隠し引数の削除.....	12
3.1.3	/Qopenmp-threadprivate:compat の実装に関する変更.....	12

3.2	新規および変更されたコンパイラー機能.....	13
3.2.1	Fortran 2003 の機能.....	13
3.2.2	その他の変更.....	13
3.3	新規および変更されたコンパイラー・オプション.....	14
3.3.1	/Od オプションの /Op の除外.....	14
3.3.2	/warn:interface は /gen_interface を含意.....	14
3.4	その他の変更.....	14
3.4.1	ビルド環境コマンドスクリプトの変更.....	14
3.4.2	デフォルトの命令セットがインテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) を必要とするものに変更.....	15
3.4.3	最適化レポートがデフォルトで無効に設定.....	15
3.4.4	I/O を制御する新しい環境変数.....	15
3.4.5	OpenMP ライブラリーのデフォルトが "compat" に変更.....	15
3.4.6	OpenMP ライブラリーのデフォルトがダイナミック・リンクに変更.....	16
3.4.7	サンプルの提供 (ZIP 形式).....	16
3.4.8	Fortran プロジェクト・ファイルの互換性.....	16
3.4.9	新しいライブラリー・ファイル (ifmodintr.lib).....	16
3.4.10	IFOPNGL モジュールの使用時には OpenGL* AUX ライブラリーはリンクされません.....	16
3.5	既知の問題.....	17
3.5.1	/debug:parallel オプションはサポートされていません.....	17
3.5.2	Visual Studio .NET 2003 のドキュメント表示エラー.....	17
3.5.3	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.....	17
3.5.4	空の派生型の制限付きサポート.....	17
3.6	Fortran 2003 機能の概要.....	18
4	インテル® マス・カーネル・ライブラリー.....	20
4.1	本バージョンでの変更.....	20
4.1.1	インテル® マス・カーネル・ライブラリー 10.2 Update 5.....	20
4.1.2	インテル® マス・カーネル・ライブラリー 10.2 Update 4.....	21
4.1.3	インテル® マス・カーネル・ライブラリー 10.2 Update 3.....	21
4.1.4	インテル® マス・カーネル・ライブラリー 10.2 Update 2.....	22
4.2	既知の問題.....	24
4.3	注意事項.....	24
4.4	権利の帰属.....	24

5	インテル® Parallel Debugger Extension.....	24
5.1	新機能	24
5.2	既知の制限事項.....	25
5.3	ドキュメント	25
6	著作権と商標について.....	26

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。

1.1 変更履歴

このセクションでは製品アップデートにおける重要な変更内容を説明します。報告されている問題の修正リストは、[インテル® コンパイラー 11.1 プロフェッショナル・エディション 修正リスト](#) (英語) を参照してください。

Update 6

- [インテル® マス・カーネル・ライブラリー](#)が 10.2 Update 5 に更新
- 報告されている問題の修正

Update 5 (1.1.060)

- [インテル® マス・カーネル・ライブラリー](#)が 10.2 Update 4 に更新
- 「互換性」セクションに [OpenMP* threadprivate 変数](#)に関する情報が追加されました。
- 報告されている問題の修正

Update 4 (11.1.054)

- [インテル® マス・カーネル・ライブラリー](#)が 10.2 Update 3 に更新
- Fortran プロジェクトのプロパティ・ページでインテル® Parallel Debugger Extension を有効にできるようになりました。
- 報告されている問題の修正

Update 3 (11.1.051)

- 報告されている問題の修正

Update 2 改訂版 (11.1.048)

- クロスコンパイラー (IA-32 からインテル® 64、IA-32 から IA-64) がリビルド。一部の Windows 7 システムおよび Windows Server* 2008 システムでコンパイラーが動作しない問題が修正されました。生成されたコードの正当性は問題ではありません。
- サポートされるオペレーティング・システムに Microsoft* Windows 7 が追加

Update 2 (11.1.046)

- [隠し引数は BIND\(C\) ルーチンで使用されなくなりました](#)。これにより、ソース変更が必要な場合があります。
- [新しいオプション /Qmkl、/Qimsl、/QxAVX](#) に関する注意事項の追加
- [/warn:interface の動作変更](#)に関する注意事項の追加
- [インテル® 64 アーキテクチャー上のコマンドライン・コンパイルでファイル名に日本語が含まれている場合の診断メッセージの問題](#)に関する注意事項の追加
- [インテル® Parallel Debugger Extension のサポートの追加](#)
- 報告されている問題の修正

Update 1 (11.1.038)

- 型バインド・プロシージャーを含む派生型を宣言または使用しているソースの[再コンパイルの必要性](#)
- [/Od](#) の動作変更に関する注意事項の追加
- [FORT_BLOCKSIZE 環境変数と FORT_BUFFERCOUNT 環境変数](#)の記載
- 報告されている問題の修正

製品リリース (11.1.035)

1.2 製品の内容

インテル® Visual Fortran コンパイラー 11.1 Windows 版プロフェッショナル・エディションには、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。IA-32、インテル® 64、および IA-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。
- IA-64 対応アプリケーション開発用インテル® アセンブラー
- インテル® マス・カーネル・ライブラリー 10.2 Update 5
- Microsoft 開発環境への統合
- Microsoft Visual Studio* 2005/2008 用インテル® Parallel Debugger Extension
- Microsoft Visual Studio 2008 Shell とライブラリー (学生ライセンス、評価版ライセンス、コンパイラー・スイート製品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

インテル® Visual Fortran プロフェッショナル・エディション IMSL* 同梱版には、上記のほか、Visual Numerics* 社の IMSL Fortran 数値計算ライブラリーが含まれています。

1.3 動作環境

アーキテクチャー名についての説明は、<http://software.intel.com/en-us/articles/intel-architecture-platform-terminology/> (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) または IA-64 アーキテクチャー (インテル® Itanium® プロセッサ) プロセッサをベースとするコンピューター
 - 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・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、Microsoft Windows Vista*、Microsoft Windows 7、Microsoft Windows Server* 2003、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 2008 SP1 あるいは 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。下記にリストされている Visual Studio のその他のバージョンは Windows Server 2008 または Windows HPC Server 2008 で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 2008 Standard Edition またはそれ以上の Edition (C++ 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 2005* Standard Edition またはそれ以上の Edition (C++ 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 Shell (インテル® Fortran コンパイラ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]
- IA-32 対応アプリケーションのビルドに、Microsoft Visual Studio 開発環境またはコマンドライン・ツールを使用する場合は、次のいずれか:
 - Microsoft Visual Studio .NET 2003 (C++ コンポーネントがインストールされていること) [3]
 - Microsoft Visual C++ .NET 2003* [3]
- IA-64 対応アプリケーションのビルドに、Microsoft Visual Studio 開発環境またはコマンドライン・ツールを使用する場合は、次のいずれか:
 - Microsoft Visual Studio 2008 Team System Edition (C++ コンポーネントと [Itanium 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) [4] さらに、Windows 2008 および .NET Framework 3.5 用 Microsoft Windows SDK
 - Microsoft Visual Studio 2005 Team System Edition (C++ コンポーネントと [Itanium 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) [4]
- 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++ 2008 Express Edition [5]
 - Microsoft Visual C++ 2005 Express Edition と Windows Server 2008 および .NET Framework 3.5 用 Microsoft Windows SDK
- 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は、次のいずれか:
 - Windows Vista 用 Microsoft Windows Software Development Kit Update
 - Windows Server 2008 および .NET Framework 3.5 用 Microsoft Windows SDK
- IA-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は:
 - Windows Server 2008 および .NET Framework 3.5 用 Microsoft Windows SDK
- ドキュメントの参照用に Adobe* Reader* 7.0 以降

注:

1. Microsoft Visual Studio 2005/2008 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。本バージョンでは、Microsoft Visual Studio 2010 はサポートされていません。
2. Microsoft Visual Studio 2008 Shell ベースのインテル® Visual Fortran 開発環境は、インテル® Visual Fortran コンパイラ・プロフェッショナル・エディションのアカデミック・ライセンスと商用ライセンスに含まれています。評価ライセンスまたは学生ライセンスには含まれていません。また、インテル® C++ コンパイラが含まれている「コンパイラ・スイート」製品にも含まれていません。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 Studio .NET 2003 は、Microsoft Windows Vista または Microsoft Windows 7 ではサポートされていません。インテル® Parallel Debugger Extension は Microsoft Visual Studio .NET 2003 ではサポートされていません。本製品の将来のバージョンでは、Microsoft Visual Studio .NET 2003 はサポートされなくなる予定です。
 4. IA-64 システムでは、Microsoft Visual Studio はサポートされていません。
 5. Microsoft Visual Studio Shell をインストールし、また Microsoft Visual C++ 2008 Express Edition (Microsoft C++ コンパイラーへの別アクセス) も使用する場合は、インテル® Visual Fortran コンパイラーと Visual Studio Shell をインストールする前に Visual C++ 2008 Express Edition をアンインストールしてください。Fortran のインストールが完了したら、必要に応じて Visual C++ 2008 Express Edition をインストールします。Fortran と C++ コンパイラー環境は個別で、混合はされません。
 6. IA-64 アーキテクチャー・システムでの開発は、IA-64 アーキテクチャー・アプリケーションのビルドのみがサポートされています。
 7. インテル® Visual Fortran コンパイラーは、デフォルトで、インテル® SSE2 命令対応のプロセッサが必要な IA-32 アーキテクチャー・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。
 8. 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 Win32 API ルーチンを使用している可能性があります。アプリケーションの互換性テストをご自身の責任で行ってください。アプリケーションを実行するには、特定のランタイム DLL をターゲットシステムにコピーしなければならないことがあります。

1.4 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 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。

1.6 日本語サポート

インテル® コンパイラーは、日本語ユーザー向けのサポートを提供しています。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。

日本語の言語サポート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・システムで英語の言語サポートを使用する場合は、<http://software.intel.com/en-us/articles/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。

1.7 テクニカルサポート

インストール時にコンパイラー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 64ビット・アプリケーション用の Visual Studio の設定

Microsoft Visual Studio 2005 または 2008 を使用し、64ビット・アプリケーション(インテル® 64 または IA-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 2005/2008 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 2005 (または 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 2005/2008 Team System Edition を使用して、IA-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するには、上記の手順に従い、[Itanium コンパイラおよびツール] ボックスがオンになっていることを確認してください。

2.1.2 Microsoft Windows Vista または Microsoft Windows 7 でのインストール

Microsoft Windows Vista または Microsoft Windows 7 では、Microsoft Visual Studio .NET 2003 はサポートされていません。Microsoft Visual Studio 2005 ユーザーは、*Visual Studio 2005 Service Pack 1 (VS 2005 SP1)* と *Visual Studio 2005 Service Pack 1 Update for Windows Vista (VS 2005 SP1 ページからリンクが提供)* をインストールしてください。これらのアップデートをインストールした後に、管理者権限で Visual Studio が実行できることを確認してください。実行できない場合、インテル® コンパイラーを使用できません。詳細は、Microsoft の「Visual Studio on Windows Vista」ページ (<http://msdn2.microsoft.com/en-us/vstudio/aa948853.aspx> (英語)) および関連ドキュメントを参照してください。

Microsoft Visual Studio 2008 は、アップデートを適用しなくても Windows Vista および Windows 7 で使用できますが、最新の Microsoft Service Pack を適用することを推奨します。

また、「[Microsoft Visual Studio 2005 の管理者権限に関するメッセージダイアログ](#)」も参照してください。

2.2 インストール

初めて製品をインストールする場合は、インストール中にシリアル番号の入力が求められますので、あらかじめご用意ください。製品のインストールと使用には、有効なライセンスが必要です。

インストールを開始するには、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。

2.2.1 IMSL Fortran 数値計算ライブラリーのインストール

インテル® Visual Fortran プロフェッショナル・エディション IMSL 同梱版のライセンスをお持ちの場合は、コンパイラとは別に IMSL をインストール (ダウンロードまたはディスクのいずれか) する必要があります。IMSL ライブラリーをインストールする前にコンパイラをインストールしてください。

2.2.2 Microsoft Visual Studio 2005 Premier Partner Edition はサポートされていません

インテル® Visual Fortran 10.0/10.1/11.0 で提供されていた Microsoft Visual Studio 2005 Premier Partner Edition (VSPPE) は 11.1 ではサポートされていません。VSPPE がインストールされていて、Visual Studio .NET 2003 や Visual Studio 2008 がインストールされていない場合は、製品のフルパッケージで Microsoft Visual Studio 2008 Shell がインストールされます。

また Visual Studio 2003 あるいは 2008 がインストールされている場合は、コンパイラのバージョン 11.1 をインストールすると、以前のコンパイラ統合が削除され、それと共に Visual Studio 2005 Premier Partner Edition の統合も削除されます。必要に応じて、以前のバージョンの VSPPE 統合をインストーラーを実行して ([変更] をクリックして、Visual Studio 2005 統合のみを選択)、再インストールすることができます。

2.2.3 Microsoft Visual Studio 2005 の管理者権限に関するメッセージダイアログ

Microsoft Visual Studio 2005 を使用している場合、Microsoft Windows Vista またはそれ以降の Microsoft Windows にインストール中、次のようなダイアログが表示されることがあります。



このダイアログが表示された場合は、[常にこのメッセージを使用する] ボックスをオンのままにして、[続行] ボタンをクリックします。[Visual Studio の終了] を選択したり、何もしない場合 (このメッセージは 2 分後にタイムアウトします)、コンパイラ統合のインストールは完了しません。

詳細は、「[Microsoft Windows Vista でのインストール](#)」を参照してください。

2.3 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] 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。

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。複数のバージョンのコンパイラをインストールし、その中から選択して使用することができます。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。

2.4 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。

- C:\Program Files\Intel\Compiler\11.1\xxx
 - bin
 - ia32
 - ia32_intel64
 - ia32_ia64
 - intel64
 - ia64
 - Documentation
 - include
 - ia32
 - intel64
 - ia64

- lib
 - ia32
 - intel64
 - ia64
- mkl
 - benchmarks
 - em64t
 - examples
 - ia32
 - ia64
 - include
 - interfaces
 - tests
 - tools
- Samples
- setup_f
- VSDebugExtension

xxx は 3 桁のビルド番号です。bin、include、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 および em64t: 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
- ia64: IA-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。
- ia32_ia64: IA-32 (またはインテル® 64) 上での実行用コンパイラー。IA-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 および IA-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。

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 11.1 でもそのまま使用できます。ただし、次の例外があります。

- BIND(C) 属性を含むオブジェクトまたは呼び出しプロシージャーでは、[ソースコードの変更と再コンパイルが必要な場合](#)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 10.0 よりも前のコンパイラーを使用してインテル® 64 または IA-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、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。

3.1.1 型バインド・プロシージャーの誤った派生型レイアウト

初期のバージョンの 11.1 コンパイラーでは未使用の空間が型バインド・プロシージャーを含む派生型に誤って追加されます。この問題は 11.1 Update 1 で修正されています。初期の 11.1 コンパイラーでコンパイルされたそのような型のオブジェクトを宣言または使用するすべてのソースは、バージョン 11.1 Update 1 以降を使用して再コンパイルする必要があります。

3.1.2 BIND(C) プロシージャーの不適切な隠し引数の削除

インテル® Fortran コンパイラーの古いバージョンでは、BIND(C) 属性でプロシージャーの隠し引数が不適切に渡されたり、使用されていました。例えば、Fortran ルーチンに CHARACTER 引数があり、BIND(C) が指定されている場合、コンパイラーはこの 2 つの隠し引数が関数のリターンアドレスと長さ用に渡されたものとみなし、さらに各文字引数の別の隠し引数とみなしていました。このような隠し引数は必要であり、非互換 (BIND(C) 属性を持たない) の Fortran プロシージャーでは正しいため、Fortran 規格ではこのような隠し引数の互換プロシージャーにおける使用を禁止しています。Fortran および C の引数リストは 1:1 で対応している必要があります。

この問題が発生する多くのケースでは、当該プロシージャーは互換プロシージャーの標準規則に合っておりません。例えば、引数が CHARACTER 型の場合、長さは 1 でなければなりません。そのような文字の 1 つの配列が引数である場合があります。どのような配列でも互換プロシージャーの関数の戻り値としては許可されていません。しかし、標準準拠の互換プロシージャーを作成して、コンパイラーが隠し引数を渡したり、想定することが可能でした。

11.1 Update 2 では、隠し引数の受け渡しや想定はしないように修正されています。隠し引数を渡す必要のあるような C コードを記述している場合は、記述を変更し、再コンパイルする必要があります。実引数として渡された結果変数で関数をサブルーチンに変換しなければならない場合もあります。ご不便をおかけして申し訳ありませんが、この修正は正当性と移植性における重要なものです。ご了承ください。

3.1.3 /Qopenmp-threadprivate:compat の実装に関する変更

/Qopenmp-threadprivate:compat オプションを使用してコンパイルしたすべての Fortran ソースは、11.1 Update 5 以降で再コンパイルする必要があります。このオプションは、OpenMP threadprivate 変数の Microsoft Visual C++* 実装と互換性のあるコードを生成しますが、以前のバージョンの Fortran ではリンク時にエラーとなったり、ランタイム時に正しく動作しない問題がありました。

新しいバージョンでは、threadprivate を作成する Fortran COMMON またはモジュール変数で初期化するか (COMMON の BLOCK DATA サブプログラムにある DATA またはモジュール変数の初期値)、あるいは対応する変数が __thread または __declspec(thread) 属性を持つ threadprivate な struct として宣言された C コードを使用してプログラムをリンクする必要があります。

3.2 新規および変更されたコンパイラー機能

いくつかの言語機能についての説明がコンパイラー・ドキュメントにまだ含まれていない可能性があります。必要に応じて、Fortran 2003 規格 (http://j3-fortran.org/doc/2003_Committee_Draft/04-007.pdf) を参照してください。

3.2.1 Fortran 2003 の機能

- オブジェクト指向の機能
 - CLASS 宣言
 - SELECT TYPE 構造
 - EXTENDS_TYPE_OF と SAME_TYPE_AS 組み込み関数
 - 多相型エンティティ
 - 継承と関連付け
 - 遅延バインディングと抽象型
 - 型問い合わせ組み込み関数
- 型バインド・プロシージャ
 - TYPE CONTAINS 宣言
 - ABSTRACT 属性
 - DEFERRED 属性
 - NON_OVERRIDABLE 属性

注: GENERIC 属性と型バインド操作はこのリリースではサポートされていません。
- 無指定文字長エンティティ
- PRIVATE コンポーネントの PUBLIC 型と PUBLIC コンポーネントの PRIVATE 型
- 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
- SYSTEM_CLOCK 組み込み関数の 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 でゼロを返します。このオプションが指定されない場合、Fortran 95 はプロセッサ依存の値を、インテル® Fortran はデフォルトで 1 を返します。
/assume:noold_maxminloc が指定されるとパフォーマンスが低下します。

3.2.2 その他の変更

- 文字列長チェックが有効 (/check:bounds) で文字オブジェクトが引数として渡されると、渡された長さの最小値と呼び出されたプロシージャで宣言された長さが上限として使用されます。
- リストで対応する変数が LOGICAL ではない場合、リスト指定またはネームリスト指定の入力で、LOGICAL 定数形式の入力値項目 (例: T または .F) は許可されなくなりました。同様に、I/O リスト変数が LOGICAL 型の場合、対応する入力値は、LOGICAL 定数の形式でなければなりません。新しい /assume:old_logical_ldio オプションを使用して古い動作に戻すことができます。
- 浮動小数点例外動作のコンパイルごとの制御 (/fpe=all)

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。

- /assume:[no]ieee_fp_flags
- /assume:[no]old_logical_ldio
- /assume:[no]old_maxminloc
- /fpe-all
- /hotpatch
- /Qdiag-enable:sc-include
- /Qdiag-enable:sc-parallel
- /Qimsl
- /Qmkl[:lib]
- /QxAVX

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。

3.3.1 /Od オプションの /Op の除外

バージョン 11.1 では、最適化を無効にする /Od オプションは、浮動小数点精度を最大にする /Op を含意しなくなりました。/Op スイッチは廃止予定です。そのため、浮動小数点精度の影響を受けやすいアプリケーションには、明示的に /fp オプションを指定することを推奨します。

3.3.2 /warn:interface は /gen_interface を含意

バージョン 11.1 で /warn:interface を指定すると、エラーチェック目的でインターフェイスを生成し使用できます。/warn:interface が有効な場合は /gen_interface を指定する必要はなくなりました。

3.4 その他の変更

3.4.1 ビルド環境コマンドスクリプトの変更

ビルド環境を構築するコマンド・ウィンドウ・スクリプトが使用する Microsoft Visual Studio バージョンを任意で指定できるよう 11.1 で変更されました。ビルド環境ウィンドウを開くのに、定義済みのスタート・メニュー・ショートカットを使用していない場合は、次のコマンドを使用して適切な環境を構築してください。

```
"C:\Program Files\Intel\Compiler\11.1\xxx\Bin\ifortvars.bat" arch [vs]
```

xxx は、リビジョン番号です。arch は、ia32、ia32_intel64、intel64、ia32_ia64、ia64 のいずれかです（「[インストール先フォルダー](#)」を参照）。vs は任意で指定します。vs2008 または vs2005 のいずれかです。vs が指定されていない場合は、コマンドライン統合用にインストール時に指定された Visual Studio のバージョンがデフォルトで使用されます。Microsoft Visual Studio .NET 2003 は vs 引数で指定できないことに注意してください。

コンパイラーを異なるパスにインストールしている場合は、適切なフォルダーを指定してください。

注: インストールされている Visual Studio のバージョンが Visual Studio 2008 Shell の場合は、vs 引数を指定しないでください。

3.4.2 デフォルトの命令セットがインテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) を必要とするものに変更

バージョン 11.0 以降、IA-32 アーキテクチャー向けのコンパイルでは、`/arch:SSE2` (旧:`/QxW`) がデフォルトになりました。`/arch:SSE2` でビルドされたプログラムは、インテル® Pentium® 4 プロセッサや特定のインテル以外のプロセッサなど、インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2) をサポートするプロセッサ上で実行する必要があります。互換性を保証するランタイムチェックは行われません。プログラムが命令をサポートしていないプロセッサで実行された場合は、無効な命令フォルトが発生する場合があります。これにより、インテル® SSE 命令が x87 命令の代わりに使用され、高い精度ではなく、宣言された精度で計算が行われることがあるため、浮動小数点結果が変更される可能性があることに注意してください。

すべてのインテル® 64 アーキテクチャー・プロセッサでインテル® SSE2 がサポートされています。

汎用 IA-32 の以前のデフォルトを使用する場合は、`/arch:IA32` を指定してください。

3.4.3 最適化レポートがデフォルトで無効に設定

バージョン 11.1 以降、コンパイラーは、ベクトル化、自動並列化、OpenMP スレッド化ループに関する最適化レポートメッセージをデフォルトで表示しないようになりました。これらのメッセージを表示するには、`/Qdiag-enable:vec`、`/Qdiag-enable:par`、`/Qdiag-enable:openmp` を指定するか、`/Qvec-report`、`/Qpar-report`、`/Qopenmp-report` を使用する必要があります。

また、バージョン 11.1 以降、最適化レポートメッセージは `stdout` ではなく、`stderr` に送られます。

3.4.4 I/O を制御する新しい環境変数

バージョン 11.1 では、アプリケーションの実行時に I/O 動作に影響する 2 つの環境変数が追加されています。

`FORT_BLOCKSIZE` は、OPEN 文で `BLOCKSIZE=` が省略されたときに使用されるデフォルトの `BLOCKSIZE` 値を指定できます。有効なサイズは 0 から 2147467264 までの値です。サイズは最も近い 512 バイト境界に丸められます。デフォルトの `BLOCKSIZE` 値は 128KB です。

`FORT_BUFFERCOUNT` は、OPEN 文で `BUFFERCOUNT=` が省略されたときに使用されるデフォルトの `BUFFERCOUNT` 値を指定できます。有効な値は 0 から 127 までの値です。0 が指定された場合、デフォルト値の 1 が使用されます。

3.4.5 OpenMP ライブラリーのデフォルトが "compat" に変更

バージョン 10.1 では、新しい OpenMP ライブラリー・セットが追加され、アプリケーションは、インテル® コンパイラーと Microsoft コンパイラーの両方からの OpenMP コードを使用することが可能でした。この "互換" ライブラリーは古い "レガシー" ライブラリーよりも高いパフォーマンスを提供します。バージョン 11.x では、互換ライブラリーが OpenMP アプリケーションのデフォルト・ライブラリーとして使用されるようになりました。

`/Qopenmp-lib:compat` と等価です。古いライブラリーを使用する場合は、`/Qopenmp-lib:legacy` を指定してください。

"レガシー" ライブラリー (`libguide.lib`、`libguide40.lib` など) は、インテル® コンパイラーの将来のリリースからは削除される予定です。

3.4.6 OpenMP ライブラリーのデフォルトがダイナミック・リンクに変更

バージョン 11.0 より、デフォルトで OpenMP アプリケーションはダイナミック OpenMP ライブラリーにリンクされます。OpenMP ライブラリーのスタティック・リンクを指定するには、/Qopenmp-link:static を指定します。

3.4.7 サンプルの提供 (ZIP 形式)

バージョン 11.1 以降、コンパイラー・プログラミング・サンプルは zip アーカイブで提供されます。各 ZIP アーカイブを書き込み可能なフォルダーに展開してください。すべてのサンプルは Visual Studio ソリューションとして提供され、コマンドライン・ビルドの説明も提供されます。詳細は、samples.htm ファイルを参照してください。

3.4.8 Fortran プロジェクト・ファイルの互換性

Fortran プロジェクト・ファイル (.vfproj) の形式がバージョン 11.0 で変更されました。インテル® Visual Fortran の古いバージョンで作成されたプロジェクトを開くと、プロジェクトの変換が必要である旨のメッセージが表示されます。古いプロジェクトは、ファイルの種類が vfproj.old として保存されます。バージョン 11.x のプロジェクトを古いバージョンのインテル® Fortran 統合で使用することはできません (ただし、古いバージョンのコンパイラーは、[Tools (ツール)] > [Options (オプション)] > [Intel C++ (インテル(R) Fortran)] > [Compilers (コンパイラー)] から使用できます)。

3.4.9 新しいライブラリー・ファイル (ifmodintr.lib)

バージョン 11.1 では、コンパイル・コード・サポートの新しいライブラリー・ファイル、ifmodintr.lib が導入されています。このライブラリーは、ISO_C_BINDING や IEEE_ARITHMETIC など、プロシージャーやデータソースを含む組み込みモジュールの使用時に自動で参照されます。リンカーオプションを設定して、デフォルトのライブラリー宣言子を無視する場合は、このライブラリーをビルドオプションに追加しなければならない場合があります。DLL サポート・ライブラリーに対してビルドする場合でもこのスタティック・ライブラリーが使用されることに注意してください。

3.4.10 IFOPNGL モジュールの使用時には OpenGL* AUX ライブラリーはリンクされません

インテル® Visual Fortran コンパイラーには、OpenGL* ルーチン、型、定数の宣言子を提供する IFOPNGL モジュールが付属しています。以前、Microsoft では glaux.lib で OpenGL AUX ライブラリーをサポートしていました。Microsoft Visual Studio 2008 以降、このライブラリーは提供されなくなりました。

IFOPNGL モジュールには、AUX ルーチンの宣言子が含まれていますが、glaux.lib には自動でリンクされません。アプリケーションで AUX ルーチンが使用され、Visual Studio 2005 以前のバージョンを使用している場合は、glaux.lib と advapi32.lib の両方をアプリケーションで参照されているライブラリーのリストに追加する必要があります。プロジェクトのプロパティーから、[Linker (リンカー)] > [Input (入力)] > [Additional Dependencies (追加の依存ファイル)] で指定できます。

Visual Studio 2008 を使用している場合は、glaux.lib を Visual Studio 2005 インストールからコピーするか、または AUX ルーチンを使用しないようにプログラムを記述しなします。インテル® Visual Fortran コンパイラーでは、AUX ルーチンの使用に代わるコーディング例を示す OpenGL サンプルの AnimateGL と Rings が提供されています。

3.5 既知の問題

3.5.1 /debug:parallel オプションはサポートされていません

/debug:parallel オプションについては、ドキュメントに記述されていますが、まだサポートされていません。後続の製品アップデートでサポートされる予定で、リリースノートにも説明が記述される予定です。

3.5.2 Visual Studio .NET 2003 のドキュメント表示エラー

Microsoft Visual Studio .NET 2003 で Microsoft MSDN ライブラリー機能をインストールしていない場合は、インテル® Visual Fortran コンパイラー・ヘルプまたは F1 状況依存ヘルプのメニュー項目を使用する際、「Visual Studio 用にヘルプがインストールされていません」という旨のメッセージが表示されます。この問題を回避するには、[ヘルプ]>[目次]をクリックした後に表示されるエラーメッセージで [OK] をクリックしてください。[目次] ペインから製品のヘルプにアクセスできるようになります。この問題は、Microsoft Visual Studio 2005 または 2008 では発生しません。

3.5.3 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。

3.5.4 空の派生型の制限付きサポート

Fortran 2003 では、派生型をデータ・コンポーネントなしで宣言する機能が追加されています。インテル® コンパイラーの現在のリリースでは、このサポートには制限があります。将来のリリースでこの制限は解除される予定です。制限事項は次のとおりです。

- 派生型のオブジェクトが宣言される際、型には少なくとも 1 つのデータ・コンポーネントがなければなりません。空の型の拡張がサポートされています。次に例を示します。

```
type t
end type
```

```
type, extends (t) :: t1
end type
```

```
type, extends (t1) :: t2
  integer i
end type
```

```
type, extends (t2) :: t3
end type
```

```
type (t) :: rec1 ! Not supported, type t is empty
type (t1) :: rec2 ! Not supported, type t1 is empty
type (t2) :: rec3 ! Supported, type t2 is not empty
type (t3) :: rec4 ! Supported, type t3 is not empty
```

例外として、クラスオブジェクトの空の型での宣言はサポートされています。次に例を示します。

```
class(t1) :: rec5
```

サポートされていない空の型が見つかったと、次のメッセージが表示されます。

```
Declaring an object with no data component fields is not yet supported (データ・コンポーネント・フィールドのないオブジェクトの宣言はサポートされていません)
```

- 空の型のコンポーネントの参照はサポートされていません。例えば、上記の宣言を仮定します。

```
call sub(rec4%t3, rec4%t1, rec3%t)
print *, rec3%t1, rec4%t
call sub2(rec3%t2, rec4%t2)
```

rec4%t3、rec4%t1、rec4%t、rec3%t1、rec3%t への参照はサポートされていません。rec3%t2 と rec4%t2 の参照はサポートされます。サポートされていない参照が見つかったと、次のメッセージが表示されます。

```
Accessing an empty type is not yet supported (空の型へのアクセスはサポートされていません)
```

- 空の型の型コンストラクターはサポートされていません。上記の宣言を例にとると、型コンストラクター t() はサポートされていません。サポートされていないコンストラクターが見つかったと、次のメッセージが表示されます。

```
A type constructor for an empty type is not yet supported (空の型の型コンストラクターはサポートされていません)
```

3.6 Fortran 2003 機能の概要

インテル® Fortran コンパイラーは、最新の Fortran 規格である、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
- 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

- 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
- TYPE CONTAINS 宣言
- ABSTRACT 属性
- DEFERRED 属性
- NON_OVERRIDABLE 属性
- 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
- 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 ("`/assume:realloc_lhs`" オプションが必要)
- 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
- プロシージャ・ポインター
- ABSTRACT INTERFACE
- PASS 属性と NOPASS 属性
- SYSTEM_CLOCK 組み込み関数の COUNT_RATE 引数が任意の種類で 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 でゼロを返します。
- 型問い合わせ組み込み関数

- COMMAND_ARGUMENT_COUNT 組み込み関数
- EXTENDS_TYPE_OF と SAME_TYPE_AS 組み込み関数
- GET_COMMAND 組み込み関数
- GET_COMMAND_ARGUMENT 組み込み関数
- GET_ENVIRONMENT_VARIABLE 組み込み関数
- IS_IOSTAT_END 組み込み関数
- IS_IOSTAT_EOR 組み込み関数
- MAX/MIN/MAXVAL/MINVAL/MAXLOC/MINLOC 組み込み関数 (CHARACTER 引数)
- MOVE_ALLOC 組み込み関数
- NEW_LINE 組み込み関数
- SELECTED_CHAR_KIND 組み込み関数
- 次の組み込み関数はオプションの KIND= 引数を使用: ACHAR, COUNT, IACHAR, ICHAR, INDEX, LBOUND, LEN, LEN_TRIM, MAXLOC, MINLOC, SCAN, SHAPE, SIZE, UBOUND, VERIFY
- ISO_C_BINDING 組み込みモジュール
- IEEE_EXCEPTIONS、IEEE_ARITHMETIC、IEEE_FEATURES 組み込みモジュール
- ISO_FORTRAN_ENV 組み込みモジュール

サポートされていない Fortran 2003 機能には次の項目が含まれます。

- 型バインド・プロシージャの型バインド操作と GENERIC バインド
- 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

4 インテル® マス・カーネル・ライブラリー

このセクションでは、インテル® Visual Fortran コンパイラー・プロフェッショナル・エディションに同梱されているインテル® 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 本バージョンでの変更

本バージョンおよび以前のバージョンの変更についての詳細は、<http://software.intel.com/en-us/articles/new-in-intel-mkl-10-2/> (英語) を参照してください。

4.1.1 インテル® マス・カーネル・ライブラリー 10.2 Update 5

新機能

- LAPACK 3.2.1 アップデート (主に LAPACK 3.2 に関する修正) に対応

パフォーマンスの向上

- FFT
 - インテル® 64 アーキテクチャー上で 3 次元以上の複素数 FFT のパフォーマンスが向上
- VSL
 - 45nm インテル® Core™2 Duo プロセッサー以降の 64 ビット・ライブラリーで、MT19937 と MT2203 基本乱数ジェネレーター (BRNG) のパフォーマンスが向上

4.1.2 インテル® マス・カーネル・ライブラリー 10.2 Update 4

新機能

- 単精度複素数の絶対値を求める SCABS1 関数を追加
- 部分的なピボット演算を使用する一般的なソルバー (?GTSV) よりも 2 倍高速な対角優位の三重対角方程式用の ?DTSVB ソルバーを追加
- 対角優位の三重対角方程式用の因数分解ルーチン (?DTTRFB) と 前方/後方代入ルーチン (?DTTRSB) を追加

パフォーマンスの向上

- FFT
 - 8 または 13 の倍数の変換のパフォーマンスが向上
 - ベクトル長が 2 のべき乗でない 1D 複素数クラスター FFT を最適化
- VSL
 - デシメーションを必要とする畳み込み/相関演算のパフォーマンスが大幅に向上 (再リンクが必要。「[既知の問題](#)」を参照)

4.1.3 インテル® マス・カーネル・ライブラリー 10.2 Update 3

パフォーマンスの向上

- BLAS
 - インテル® Core™ i7 プロセッサおよびインテル® Xeon® プロセッサ 5300 番台、5400 番台、5500 番台で次の BLAS レベル 1、2 関数の 32 ビット OS バージョンがスレッド化: (D,S,C,Z)COPY、(D,S,C,Z)SWAP、(D,S,C,Z)AXPY、(S,C)ROT、(S,C)DOT、CDOTC、(D,S,C,Z)GEMV、(D,S,C,Z)TRMV、(S,C)SYMV、(S,C)SYR、(S,C)SYR2
 - インテル® Xeon® プロセッサ 5300、5400、5500 で次の BLAS レベル 1 関数の 32 ビットおよび 64 ビット OS バージョンが向上: ZAXPY、ZSCAL、ZDOT(U,C)、(D,S)ROT
 - インテル® Xeon® プロセッサ 5300 で列より非常に多い行を持つ行列の DGEMM スレッド化効率が向上
- LAPACK
 - 次の LAPACK 関数のスケーラビリティが向上: ?POTRF、?GEBRD、?SYTRD、?HETRD、?STEDC 分割統治固有ソルバー
- FFT
 - 下層のカーネルが更新され、FFT において幅広くパフォーマンスが向上
 - 1 つの関数呼び出しで少数の変換が計算される場合に 3D FFT のスレッド化が向上
 - スレッド化を小規模の多次元変換に拡張
- VML
 - インテル® Xeon® プロセッサ 5500 番台の VML 関数がさらに最適化: v(s,d)Asin、v(s,d)Acos、v(s,d)Ln、v(s,d)Log10、vsLog1p、v[s/d]Hypot
- VSL
 - viRngPoisson および viRngPoissonV 乱数ジェネレーターのパフォーマンスが向上

ユーザービリティとインターフェイスの向上

- uBLAS、Java、FFTW3、LAPACK95、BLAS95 のサンプルプログラムの向上
- examples ディレクトリーとリファレンス・マニュアルのソースコードにある同一のサンプルは、リファレンス・マニュアルでは削除
- クラスター FFT の新しい 64 ビット整数 (ILP64) fftw_mpi インターフェイス

4.1.4 インテル® マス・カーネル・ライブラリー 10.2 Update 2

新機能

- LAPACK 3.2
 - 238 個の新しい LAPACK 関数
 - 超精密反復法の改良
 - ハウスホルダー QR 因数分解の非負対角
 - 低プロファイル行列でのハイパフォーマンス QR とハウスホルダー反射
 - 高速で正確な新しいヤコビ法 SVD
 - 矩形フル圧縮形式のルーチン
 - ピボットコレスキー
 - 混合精度反復法の改良 (コレスキー)
 - より安定した DQDS アルゴリズム
- DZGEMM 拡張 BLAS 関数の実装 (<http://www.netlib.org/blas/blast-forum/> (英語) の説明を参照)。リファレンス・マニュアルの BLAS セクションの *gemm 関数ファミリーの説明を参照してください。
- PARDISO で実数、複素数、単精度データをサポート

ユーザービリティ/インターフェイスの向上

- スパース行列形式変換ルーチン:
 - CSR (3-配列バリエーション) ↔ CSC (3-配列バリエーション)
 - CSR (3-配列バリエーション) ↔ 対角形式
 - CSR (3-配列バリエーション) ↔ スカイライン
- Fortran95 BLAS と LAPACK のコンパイル・モジュール・ファイル (.mod) を追加
 - モジュールは、インテル® Fortran コンパイラーで事前にビルドされており、インクルード・ディレクトリーにあります (フルパス情報については、インテル® MKL ユーザーズ・ガイドを参照してください)。
 - ほかのコンパイラー用のソースも提供されています。
 - インターフェイスについてのドキュメントは、インテル® MKL ユーザーズ・ガイドを参照してください。
- FFTW3 インターフェイスを直接メイン・ライブラリーに統合
 - デフォルトのインテル® Fortran コンパイラー規則と名前修飾で互換性のないコンパイラーでラッパーを作成するためのソースコードも提供されています。
 - 詳細は、リファレンス・マニュアルの付録 G を参照してください。
- DFTI_DESCRIPTOR_HANDLE を型の名前に追加。ユーザープログラムで型として参照できます。
- ユーザーデータへのアクセスを可能にするために最適化ソルバードメインのヤコビ行列計算ルーチンにパラメーターを追加 (詳細は、リファレンス・マニュアルの djacobix 関数の説明を参照してください)
- 64 ビット・アーキテクチャーでインテル® MKL の単精度 BLAS 関数 (頭文字 "s" または "c" の関数) から 64 ビット浮動小数点精度関数へのインターフェイス・マッピング呼び出しを追加 (詳細は、インテル® MKL ユーザーズ・ガイドの「sp2dp」を参照してください)
- 互換ライブラリー (「ダミーライブラリー」) を削除

パフォーマンスの向上

- インテル® 64 アーキテクチャー用にさらにスレッド化された BLAS レベル 1、2 関数
 - レベル 1 関数 (ベクトル-ベクトル): (C,S,Z,D)ROT、(C,Z,S,D)COPY、(C,Z,S,D)SWAP
 - キャッシュのデータ位置に応じて、4 コアのインテル® Core™ i7 プロセッサ上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 1.7 ~ 4.7 倍向上
 - キャッシュのデータ位置に応じて、24 コアのインテル® Xeon® プロセッサ 7400 番台システム上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 14 ~ 130 倍向上
 - レベル 2 関数 (行列-ベクトル): (C,Z,S,D)TRMV、(S,D)SYMV、(S,D)SYR、(S,D)SYR2
 - キャッシュのデータ位置に応じて、4 コアのインテル® Core™ i7 プロセッサ上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 1.9 ~ 2.9 倍向上
 - キャッシュのデータ位置に応じて、24 コアのインテル® Xeon® プロセッサ 7400 番台システム上でバージョン 10.1 Update 1 に対してパフォーマンスが最大 16 ~ 40 倍向上
- インテル® Core™ i7 プロセッサ、インテル® Xeon® プロセッサ (5300 番台、5400 番台、7400 番台) で、DSYRK の 32 ビット逐次バージョンに導入された再帰アルゴリズムのパフォーマンスが最大 20% 向上。
- インテル® Xeon® 7460 プロセッサで、大規模な問題の LU 因数分解 (DGETRF) がバージョン 10.1 Update 1 に対して 25% 向上。また小規模な問題でも劇的に向上。
- BLAS *TBMV/*TBSV 関数でレベル 1 BLAS 関数を使用。インテル® Core™ i7 プロセッサ上で最大 3%、インテル® Core™2 プロセッサ 5300 番台と 5400 番台で最大 10% のパフォーマンスが向上。
- DGEMM パフォーマンスを強化するスレッド化アルゴリズムの向上
 - 8 スレッドで最大 7% の向上、3、5、7 スレッドで最大 50% の向上 (インテル® Core™ i7 プロセッサ)
 - 3 スレッドで最大 50% の向上 (インテル® Xeon® プロセッサ 7400 番台)
- 非素数サイズのスレッド化 1D 複素数-複素数 FFT
- 3D 複素数-複素数変換の新しいアルゴリズムにより 1 スレッドまたは 2 スレッドで小さな問題サイズ (最大 64x64x64) についてより優れたパフォーマンスを提供
- 対称正定行列の演算時におけるアウトオブコア (OOC) PARDISO のハイレベルな並列化実装
- すべての行列の型でインコアとアウトオブコアの両方で PARDISO のメモリー使用量が減少
- 実対称行列、複素エルミート行列、複素対称行列に対し PARDISO OOC で使用されるメモリーがインテル® MKL 10.1 で使用されていたメモリーの半以下まで減少
- PARDISO/DSS における順序付けの並列化とシンボリック因子分解
- インテル® Core® i7 プロセッサとインテル® Core™2 プロセッサで次の VML 関数において最大 2 倍のパフォーマンスの向上 (平均で 30% の向上): v(s,d)Round、v(s,d)Inv、v(s,d)Div、v(s,d)Sqrt、v(s,d)Exp、v(s,d)Ln、v(s,d)Atan、v(s,d)Atan2
- インテル® Advanced Vector Extension (インテル® AVX) で次の関数の最適化バージョンが利用可能
 - BLAS: DGEMM
 - FFT
 - VML: exp、log、pow
 - 上記の関数にアクセスする `mkl_enable_instructions()` 関数に関する重要な情報については、インテル® MKL ユーザーズ・ガイドを参照してください。

4.2 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://software.intel.com/en-us/articles/intel-math-kernel-library-support-resources/> (英語) を参照してください。

4.3 注意事項

インテル® MKL の将来のバージョンでは以下の変更が予定されています。「[テクニカルサポート](#)」を参照してください。

- ファイル名に `solver` を含むライブラリーの内容をコア・ライブラリーに移動する予定です。これらの `solver` ライブラリーはその後削除される予定です。

4.4 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・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. Petitot、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。本リリースのインテル® MKL の一部の FFT 関数は、ヒューストン大学からライセンスを受けて、UHFFT ソフトウェア生成システム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。

5 インテル® Parallel Debugger Extension

このセクションでは、インテル® Visual Fortran コンパイラー・プロフェッショナル・エディションのインテル® Parallel Debugger Extension の変更点、新機能、および最新情報をまとめています。

5.1 新機能

インテル® Visual Fortran コンパイラー 11.1 Update 2 で、インテル® Parallel Debugger Extension による Fortran プログラムの並列デバッグサポートが追加されました。

インテル® Visual Fortran コンパイラー 11.1 Windows* 版プロフェッショナル・エディション インストール・ガイドおよびリリースノート

5.2 既知の制限事項

- Fortran の多次元配列が適切に表示されません。また、フィルター式でも受け付けません。
- Fortran の複素数型が適切に表示されません。
- カスタムの配列境界による Fortran 配列にフィルターを設定できません。
- Microsoft Visual Studio 2005 を使用している場合、3つのインテル固有の例外を手動で有効に設定する必要があります。[デバッグ] > [例外] を選択し、[Win32 Exceptions] ツリーを展開して、以下の項目を有効にします。

```
ala0ldb0 Intel Parallel Debugger Extension Exception 0
ala0ldb1 Intel Parallel Debugger Extension Exception 1
ala0ldb2 Intel Parallel Debugger Extension Exception 2
```

これは、プロジェクトごとに1回設定します。

- デバッグセッション中にインテルのデバッグ例外を無効にすると、Visual Studio (Visual Studio 2008 SP1 まで) がハングアップすることがあります。
- インテル® Parallel Debugger Extension を使用するには、OpenMP ライブラリーが動的にリンクされている必要があります (デフォルト)。インテル® Parallel Debugger Extension を使用する場合、OpenMP ライブラリーのスタティック・リンクを指定する `/Qopenmp-link:static` を使用しないでください。
- Microsoft Visual Studio 2008 を使用し、64ビット・アプリケーションのデバッグを行う場合、Visual Studio 2008 Service Pack 1 がインストールされている必要があります。
 - サービスパックがインストールされていない場合、Visual Studio 2005 および 2008 での 64ビット・アプリケーションのデバッグは、低メモリー領域にリンクされる場合のみ行うことができます。低メモリー領域にリンクされない場合、デバッグ対象が終了するまでイベントは表示されません。終了後、すべてのイベントがイベントウィンドウに表示されます。64ビット・アプリケーションを適切にデバッグするには、[プロジェクト] > [プロパティ] > [Linker (リンカー)] > [Advanced (詳細)] でベースアドレスを `0x10000` に設定します。
- [Data Sharing Events (データ共有イベント)] ウィンドウでローカル変数が「???'と表示されます。
- SSE レジスターウィンドウが 64ビット・アプリケーションで動作しません。ウィンドウに「???'と表示されます。
- スタティック・ローカル変数のフィルターがコンテキスト・メニューから正しく設定されません。
- 逆アセンブルビューで再入可能な呼び出しの検出が停止します。
- デバッガー拡張ウィンドウの配置が "docked" から "floating" に変更されるとウィンドウは空のままです。この問題を回避するには、"docked" のままにしておくか、または配置の変更後にデバッグセッションを再起動します。
- デバッガー拡張では、Visual Studio からアプリケーションを開始する必要があります。既存のプロセスへアタッチしている場合は動作しません。
- ウィンドウが非表示、あるいは閉じられた後に再度開かれた場合は、デフォルト (16進) 設定に戻ります。

5.3 ドキュメント

インテル® Parallel Debugger Extension のドキュメントは、Microsoft Visual Studio の [ヘルプ] メニュー、または特定のダイアログボックスで [ヘルプ] ボタンをクリックして表示することができます。

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ます。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。

Intel、インテル、Intel ロゴ、Intel Core、Itanium、Pentium、Xeon は、アメリカ合衆国および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。

© 2010 Intel Corporation. 無断での引用、転載を禁じます。