

Linux* 版インテル® Fortran コンパイラ 8.0 リリースノート

目次

[概要](#)
[新機能](#)
[動作環境](#)
[インストール](#)
[既知の制限事項](#)
[テクニカル・サポート](#)
[ドキュメント](#)
[追加情報](#)
[著作権と商標について](#)

概要

Linux 版インテル® Fortran コンパイラは、すべてのインテル 32 ビット (IA-32) プロセッサおよびインテル® Itanium® プロセッサに対応し、これらのプロセッサ上できわめて高速に動作するソフトウェアを生成します。インテル® コンパイラは、インテル® Pentium® 4 プロセッサと Pentium M プロセッサのストリーミング SIMD 拡張命令 2 (SSE2)、HT テクノロジ インテル Pentium 4 プロセッサのストリーミング SIMD 拡張命令 3 (SSE3)、インテル Itanium 2 プロセッサのソフトウェアのパイプライン化など数々の最適化機能に対応しているほか、プロシージャ間の最適化 (IPO) やプロファイルに基づく最適化 (PGO) によってアプリケーションのパフォーマンスを大きく向上させます。また、OpenMP* に対応し、自動並列化機能も備えるなど、マルチスレッド・コードの開発もサポートしています。

インストール・パッケージには以下のコンポーネントが含まれています:

- IA-32 ベース・アプリケーション開発用インテル Fortran コンパイラ 8.0: ifort
- Itanium ベース・アプリケーション開発用インテル Fortran コンパイラ 8.0: ifort
- IA-32/Itanium ベース・アプリケーション共通インテル® デバッガ 7.3: idb
- インテル Itanium ベース・アプリケーション生成用インテル Itanium アセンブラ 7.0: ias
- インテル・コンパイラのコード・カバレッジ・ツール
- インテル・コンパイラのテスト・プライオリタイゼーション・ツール
- 各ドキュメントに簡単にアクセスできるように、[ドキュメント・インデックス](#) (<インストール先ディレクトリ>/doc/fcompindex.htm) が提供されています。

インテル・コンパイラを使用して、Pentium 4 プロセッサおよび Itanium プロセッサ用にアプリケーションを最適化する方法については、[『Optimizing Applications with the Intel C++ and Fortran Compilers for Windows* and Linux』](#) (英語) を参照してください。インテルのソフトウェア開発製品の詳しい情報については、<http://www.intel.com/jp/developer/software/products/> を参照してください。

製品の内容

IA-32 ベース・アプリケーション開発用インテル Fortran コンパイラ

IA-32 ベース・アプリケーション開発用インテル Fortran コンパイラは以下の内容で構成されています:

- IA-32 アプリケーション開発用 Linux 版 Intel Fortran コンパイラ 8.0
- IA-32 ベース・アプリケーション用 Intel デバッガ 7.3
- Intel コンパイラのコード・カバレッジ・ツール
- Intel コンパイラのテスト・プライオリタイゼーション・ツール
- バージョン 8.0 の製品ドキュメント
 - 各ドキュメントに簡単にアクセスできるように、[ドキュメント・インデックス](#) (<インストール先ディレクトリ>/doc/fcompindex.htm) が提供されています。
 - トレーニング用のチュートリアル『Enhancing Performance with Intel® Compilers』(英語) も含まれています。

Itanium ベース・アプリケーション開発用 Intel Fortran コンパイラ

Itanium ベース・アプリケーション開発用 Intel Fortran コンパイラは、以下の内容で構成されています:

- Itanium ベース・アプリケーション開発用ネイティブ Fortran コンパイラ 8.0
- Itanium ベース・アプリケーション用 Intel デバッガ 7.3
- Itanium ベース・アプリケーション生成用 Itanium アセンブラ 7.0
- Intel コンパイラのコード・カバレッジ・ツール
- Intel コンパイラのテスト・プライオリタイゼーション・ツール
- バージョン 8.0 の製品ドキュメント
 - 各ドキュメントに簡単にアクセスできるように、[ドキュメント・インデックス](#) (<インストール先ディレクトリ>/doc/fcompindex.htm) が提供されています。
 - トレーニング用のチュートリアル『Enhancing Performance with Intel® Compilers』(英語) も含まれています。

互換性

.mod ファイルを生成するバージョンを含む、以前のバージョンのコンパイラでコンパイルされたすべての Fortran ソースを再コンパイルする必要があります。サードパーティのライブラリを使用している場合、これらのライブラリもリビルドする必要があります。リビルドが必要なライブラリの詳細については、[Intel® プレミア・サポート](#) (英語) でご確認ください。他のバージョンの Fortran コンパイラとの互換性については、Intel プレミア・サポート、または <http://developer.intel.com/jp/developer/software/products/compilers/flin/> 以下の説明を参照してください。

新機能

Intel Fortran コンパイラ 8.0 は、Intel Fortran と Compaq* Visual Fortran の最高のテクノロジーを組み合わせた新世代の Intel Fortran コンパイラです。Fortran 言語機能を提供し、Fortran 特有の最適化を行う、コンパイラの "フロントエンド" には、Compaq Visual Fortran のテクノロジーが採用されています。Fortran ランタイム・ライブラリ (I/O および数値演算以外の組み込み関数) も、Compaq Visual Fortran のテクノロジーが採用されています。コンパイラの "バックエンド" には、Intel プロセッサ上で最高のパフォーマンスを実現するべく、Intel の最新のハイパフォーマンスなコード生成と最適化テクノロジーが採用されています。

HT テクノロジー Intel Pentium 4 プロセッサ (SSE3 対応) 向けの最適化のサポート

HT テクノロジー Intel Pentium 4 プロセッサ (SSE3 対応) は新世代の IA-32 プロセッサです。このプロセッサは、複素数データ型、浮動小数点から整数への変換、および水平加算を含むベクトル化ループのパフォーマンスを向上させる新しい "SSE3" 命令セットをサポートします。

Linux 版インテル Fortran コンパイラ 8.0 には、この新しいプロセッサ向けに最適化する機能が追加されています。この機能を使用するには、`-xP` または `-axP` オプションを指定します。詳細は、『インテル® Fortran コンパイラ・ユーザーズ・ガイド』の「最適化」のセクションを参照してください。

IA-32 プロセッサの新しい最適化オプション

このリリースでは、2 つの新しいコード生成オプションが追加されました。`-xB` および `-axB` は、インテル Pentium M プロセッサで最高のパフォーマンスを引き出すコードを生成するようにコンパイラに指示します。新しい `-xN` および `-axN` オプションは、すべてのインテル Pentium 4 プロセッサ向けに最適化します。Pentium 4 プロセッサで最高のパフォーマンスを引き出すため、新しい `-xN` および `-axN` オプションを使用することを推奨します。Pentium M プロセッサの場合は、`-xB` または `-axB` オプションを使用してください。詳細は、『インテル® Fortran コンパイラ・ユーザーズ・ガイド』の「最適化」のセクションを参照してください。

デフォルトのリンク動作の変更

コンパイラの設定ファイル `<インストール先ディレクトリ>/bin/ifort.cfg` が変更され、リンカへの `RPATH` コマンドが削除されました。このコマンドは、以前のバージョンで、実行ファイル形式のインテル共用ライブラリの場所を設定していたものです。この `RPATH` ディレクティブを設定ファイルに置く短所として、“ifort” (コンパイラ) と入力するだけで、次のようなエラーを誘引していたという点があります：

```
/usr/lib/crt1.o: In function `_start': /usr/lib/crt1.o(.text+0x18): undefined
reference to `main'
```

このエラーの原因は、使用しているコンパイラやインストールに問題があるのではないかという混乱を招いたため、このバージョンでは `RPATH` ディレクティブが削除され、“ifort” と入力すると次のように詳細なエラー・メッセージが表示されるようになりました：

```
ifort: Command line error: no files specified; for help type "ifort -help"
```

ただし、実行ファイルにはインテル共用ライブラリの場所が含まれていないため、その場所を指定する必要があります。次のいずれかの方法で指定できます：

1. `LD_LIBRARY_PATH` 環境変数にインテル共用ライブラリの場所を含める。sh 型のシェルの場合は、“`export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<インストール先ディレクトリ>/lib`” と入力する。
2. `ldconfig` システム・ユーティリティを使用する。(注: ユーティリティの実行には、root ユーザ権限が必要です。)
3. コンパイラの設定ファイル `<インストール先ディレクトリ>/bin/ifort.cfg` を変更する。

次のようなエラー・メッセージが表示された場合は、上記の 3 つのいずれかの方法を使用してインテル共用ライブラリの場所を設定する必要があります：

```
error while loading shared libraries: libcprts.so.5: cannot open shared object file:
No such file or directory
```

インテル・コンパイラのコード・カバレッジ・ツール

インテル・コンパイラのコード・カバレッジ・ツールは、インテル・コンパイラのプロファイルに基づく最適化テクノロジーを利用して、開発者に特定のワークロードにおけるアプリケーション・コードのカバレッジの全体像を呈示します。ツールはアプリケーションのコード・カバレッジを把握するために、コンパイラにより生成された静的プロファイル情報と、ユーザのインストール済みバイナリを

ワークロードで実行した結果、生成される動的プロファイル情報を組み合わせます。カバレッジ・ツールは、この情報とアプリケーションのソースを併用して、コード・カバレッジが色付きの注釈で強調表示された HTML ページを作成します。また、アプリケーションのファイルと関数を並べ替えて、カバレッジの低いモジュールと関数を簡単に識別できるフレームを作成します。コード・カバレッジを表示するには、任意のブラウザを使用できます。

インテル・コンパイラのコード・カバレッジ・ツールをさまざまな方法で活用して、開発効率を改善し、問題を減少し、アプリケーションのパフォーマンスを向上させることができます。ツールをテストスペースのアプリケーション・プロファイルに適用すると、カバレッジ情報に基づいてテストの全体の質を測定することができます。同様に、パフォーマンス・ワークロードのプロファイルに適用すると、コード・カバレッジ情報により実行されたワークロードが、アプリケーションの重要なコードをどの程度カバーするかを把握できます。インテル・コンパイラが提供するプロファイルに基づく最適化の利点を享受するには、パフォーマンス・クリティカルなモジュールの高いカバレッジが必要です。また、このツールは、アプリケーションの各基本ブロックの動的実行カウントを表示して、カバレッジとパフォーマンス・チューニングの両方に有用な情報を示すオプションを提供します。カバレッジ・ツールは、アプリケーションの 2 つの異なる実行におけるプロファイルを比較できます。この機能は、アプリケーションがカスタムなどによってテストスペース外で実行される場合、アプリケーションのテストでカバーされないアプリケーションのコードの部分を検索するのに使用することができます。

インテル・コンパイラのコード・カバレッジ・ツールは、Windows/Linux 上の IA-32 プロセッサ・ファミリおよび Itanium プロセッサ・ファミリでサポートされます。また、C、C++、および Fortran をシームレスにサポートします。

インテル・コンパイラのテスト・プライオリタイゼーション・ツール

テスト・プライオリタイゼーション・ツールは、インテル・コンパイラのプロファイルに基づく最適化テクノロジーを利用して、以前のアプリケーションの実行プロファイルを基に、アプリケーションのテストの選択と重要度付けを行います。ユーザは、このツールを使用して、アプリケーションのコードのサブセットに対してより適切なテストを選択し、重要度付けを行うことができます。アプリケーションの特定のモジュールが変更されると、テスト・プライオリタイゼーション・ツールはそのセットの変更によって最も影響を受けるテストを示唆します。また、以前に実行されたアプリケーションのプロファイル・データを分析して、アプリケーションのコンポーネントとテストの依存性を確認し、この情報を基にテストのプロセスをガイドします。このツールは、アプリケーションのコード・カバレッジを基に、効率的なテスト階層を作成するのに使用できます。例えば、テストの全体をカバーするアプリケーション・テストの最小サブセットを定義します。また、テストのターンアラウンド時間を大幅に短縮します。長い時間を費やして多くの失敗を見つける代わりに、ユーザは、セットの変更によって問題が発生する少数のテストをすばやく見つけられます。このツールはまた、テストがボトルネックとなる大規模なアプリケーションのテストと開発に要する時間を大幅に短縮するのに役立ちます。アプリケーションの任意のサブセットで全体をカバーするために必要なテストの数を最小限にし、さらに、テストの実行時間が決まっている場合は、特定のレベルのコード・カバレッジを最小限の時間で達成するテストを選択して重要度付けを行うことができます。

インテル・コンパイラのテスト・プライオリタイゼーション・ツールは、Windows/Linux 上の IA-32 プロセッサ・ファミリおよび Itanium プロセッサ・ファミリでサポートされます。また、C、C++、および Fortran をシームレスにサポートします。

詳細に関しては、次のリンクを参照してください:

<http://www.intel.com/software/products/compilers/techtoc/pgt.htm> (英語)

インテル・デバッガのグラフィカル・ユーザ・インターフェイス

インテル・デバッガ idb に、オプションのグラフィカル・ユーザ・インターフェイスが用意されました。

既知の問題と制限を含む詳細は、インテル・デバッガのリリースノートを参照してください。

プロシージャ間の最適化 (IPO) におけるバージョン番号付きの中間ファイル (.il)

IPO で生成された各 .il ファイルには、バージョン番号が割り当てられます。コンパイラは、同じバージョンの .il ファイルのみを処理します。バージョン番号はビルド処理の一部として、自動的に生成され、更新されます。

追加された機能は次のとおりです:

- Compaq Visual Fortran 6.6 との完全な言語互換性
- 派生型拡張子の Fortran 2003 割付けコンポーネントのサポート
- ランタイム・エラーとソースを関連付けるトレースバック・オプションおよびアプリケーションからのトレースバックを要求する TRACEBACKQQ ルーチン
- ドキュメントの追加:
 - 『Intel® Fortran コンパイラ・ユーザズ・ガイド』では、アプリケーションのビルドおよびアプリケーションの最適化に関する情報が、Vol I、Vol II で説明されています。
 - 以前、『Intel® Fortran Programmer's Reference』で説明されていた Intel Fortran の言語情報 (組込みプロシージャやディレクティブなど) は、『Intel® Fortran Language Reference』(PDF ファイル、英語) で説明されています。

その他の変更:

- Intel Fortran のコマンド名が ifort に変更されました。Intel Fortran の以前のバージョンでは、コマンド名に ifc または efc が使用されていました。Intel Fortran 8.0 でも ifc と efc コマンド名を使用することができます。しかし、Intel Fortran の将来のバージョンでは、ifort コマンド名以外は使用できなくなる予定です。
- LOGICAL 表現における true と false の値が Intel Fortran の以前のバージョンで使用されていた値から変更されました。Intel Fortran の以前のバージョンでは、整数表現が LOGICAL として解釈された場合、ゼロは .FALSE.、ゼロ以外の値は .TRUE. として解釈されていました。Intel Fortran コンパイラ 8.0 では、偶数の整数値が .FALSE.、奇数の整数値が .TRUE. として解釈されるように変更されました。さらに、論理定数 .TRUE. の値が、Intel Fortran の以前のバージョンで使用されていた 1 ではなく -1 に変更されました。この変更は Compaq Visual Fortran の仕様と一致するように行われたものですが、-fpscomp logicals コンパイラ・オプションを使用して Intel Fortran の以前のバージョンの仕様に戻すこともできます。
- これまでライブラリ・ルーチンとして解釈されていたコマンドライン照会ルーチン GETARG、IARGC および NARGS は、組込みプロシージャとして解釈されるように変更されました。アプリケーションでこれらのルーチンを EXTERNAL として宣言している場合、またはこれらのルーチンへの明示的なインターフェイスを用意している場合、宣言およびインターフェイス部分を削除しないとビルドに失敗します。
- HP* OpenVMS* Fortran コンパイラとの互換性を指定する -vms コンパイラ・オプションの効果が変更されました。Intel Fortran の以前のバージョンでは、このオプションは文字リテラル中で %n を新しい行として認識できないようにして、書式なしファイルの RECL のデフォルトのサイズを 4 バイトに設定していました。バージョン 8.0 では、これらの処理に対して別々のスイッチ (-assume [no]bscc および -assume [no]byterecl) が追加されました。-vms オプションの効果の詳細は、『Intel® Fortran コンパイラ・ユーザズ・ガイド』の「-vms」の説明を参照してください。
- Intel Fortran の以前のバージョンでは、書式なしファイルのデフォルトの RECL= の単位はバイトでした。バージョン 8.0 では、Compaq Fortran のデフォルト (4 バイト) 単位を採用しました。コンパイラ・オプション -assume byterecl を追加すれば、1 バイト単位で指定することもできます。
- Intel Fortran の以前のバージョンでは、8 バイトの数値変数が自然境界でアライメントさ

れるように COMMON 中の変数間で 7 バイトまでのパディングを行っていました。しかし、これは Fortran 標準の規則には沿っていません。バージョン 8.0 では、デフォルトで COMMON ブロック中の変数間にパディングを挿入しないように変更されました。アライメントされていない数値変数が見つかった場合、コンパイラは警告メッセージを発行しません (-warn noalign が指定されている場合を除きます)。この変更は、アプリケーション名以外で COMMON 中の変数にアクセスするアプリケーション (例えば、COMMON が C extern で共用されている場合) を除けば、ほとんどのアプリケーションでは影響はありません。これまでと同じように、COMMON 中の変数がパディングされるようにするには、-align commons (4 バイト境界までアライメント) または -align dcommons (8 バイト境界までアライメント、バージョン 7.1 と同じ) を指定してください。

動作環境

IA-32 プロセッサの動作環境

- インテル Pentium II プロセッサ 450 MHz 版以降の IA-32 プロセッサを搭載したシステム (インテル Pentium 4 プロセッサ推奨)
- RAM 128 MB (256 MB 推奨)
- ディスク空き容量: 100 MB (インストール時、ファイルのダウンロードやテンポラリ・ファイル用に別途 200 MB の空き容量が必要)
- 次のディストリビューションに代表される glibc 2.2.5、2.2.93、または 2.3.2 およびカーネル 2.4.X を搭載した Linux システム。注: リストされているすべてのディストリビューションで動作が確認されているわけではありません。また、すべてのディストリビューションをリストしていません。
 - Red Hat* Linux 7.3
 - Red Hat Advanced Server 2.1
 - SUSE* Enterprise 8 / United Linux* 1.0
 - Red Hat Enterprise Linux 3.0
 - Red Hat Linux 9
 - Red Hat Linux 8.0

Itanium プロセッサの動作環境

- Itanium プロセッサまたは Itanium 2 プロセッサを搭載したコンピュータ
- RAM 512 MB (1 GB 推奨)
- ディスク空き容量: 150 MB (インストール時、ファイルのダウンロードやテンポラリ・ファイル用に別途 200 MB の空き容量が必要)
- 次のディストリビューションに代表される glibc 2.2.4、2.2.5、または 2.3.2 および Linux カーネル 2.4.X を搭載した Linux システム。注: リストされているすべてのディストリビューションで動作が確認されているわけではありません。また、すべてのディストリビューションをリストしていません。
 - Red Hat Advanced Server 2.1
 - SUSE* Enterprise 8 / United Linux* 1.0
 - Red Hat Linux 7.2
 - Red Hat Enterprise Linux 3.0

注: 非常に大きなソースファイル (数千行) を -O3、-ipo および -openmp などの高度な最適化でコンパイルする場合は、相当な量の RAM が必要になります。

インストールに関する注意事項

以前のバージョンのコンパイラを使用している場合は、インテル Fortran コンパイラ 8.0 をインストール

ールする前に、以前のコンパイラをアンインストールすることを推奨します。アンインストールに関しては、[「コンパイラとデバッガのアンインストール」](#)を参照してください。

インテル Fortran コンパイラのインストール・スクリプトは、システム・ユーティリティ RPM を使用してファイルをインストールします。RPM 4.0.2 および RPM 4.1 の両方に制限があるので注意してください。詳細は[「既知の制限事項」](#)を参照してください。

ライセンスのインストール

インテル Fortran コンパイラは、Macrovision 社の FLEXIm* 電子ライセンス・テクノロジーを使用しています。ライセンスの管理は透過的に行われます。インテル Fortran コンパイラ 8.0 のインストール・プログラムは、製品コンポーネントをインストールする前に有効なライセンスをチェックします。また、プログラムのコンパイル時とビルド時にもライセンス・ファイルのチェックを行います。

注: サポート・サービスの有効期間内であれば、Linux 版インテル Fortran コンパイラのバージョン 6.0.1 または 7.1 のライセンスを 8.0 でもそのまま使用できます。

インストール前に、以下の手順に従ってライセンス・ファイルをセットアップしてください。

- インテル Fortran コンパイラ 8.0 のダウンロード版をご利用の場合は、ライセンスは電子メールで送信されます。ライセンス・ファイルのインストール方法については、電子メールに記載の指示に従ってください。
- インテル Fortran コンパイラ 8.0 の CD-ROM 版パッケージをご利用の場合は、有効なライセンスは CD-ROM に入っており、インストール・プログラムがこれを自動的に検出します。しかし、テクニカル・サポートを受けたり、製品アップデートのダウンロードおよびインストールを行うには、**CD-ROM 版のユーザのみ**以下の手順が必要となります:
 1. **製品登録を行う:** 製品外箱の折り蓋に記載されているシリアル番号を確認してください。次に、<http://www.intel.com/software/products/registrationcenter/> (英語) にアクセスし、指示に従ってください。登録が完了すると、24 時間以内に電子メールで新しいライセンスが送られます。
 2. **新しいライセンスをインストールする:** 電子メールで送られた新しいライセンスは、1 年間のサポートサービスを保証するもので、この期間内に製品アップデートのダウンロードとインストール、およびフルのテクニカル・サポートを受けることができます。また、この電子メールにはライセンスのインストール方法についても記載されています。指示に従って新しいライセンスのインストールを完了してください。

注:

ライセンス・ファイルの拡張子は、".lic" です。

デフォルトのライセンス・ディレクトリは /opt/intel_fc_80/licenses/ です。

サポート・サービス・ライセンスの詳細は、

<http://www.intel.com/software/products/compilers/flin/pricelist.htm> (英語) を参照してください。

インテル Fortran コンパイラのインストール

次の手順で IA-32 コンパイラをインストールします。

1. コンパイラ・パッケージをダウンロードします。
2. 書き込み権限があるディレクトリでパッケージを解凍 (untar) します。

```
> untar -xvf I_fc_p[c]_8.0.xxx.tar
```

または

```
> untar -zxvf I_fc_p[c]_8.0.xxx.tar.gz
```
3. インストール・スクリプトを実行します。
rpm コマンドを実行するために、**root ユーザ** でログインします。そして、解凍されたファイル

が抽出されたディレクトリでインストール・スクリプトを実行します。

```
> source ./install.sh
```

root アカウントでログインできない場合は、rpm2cpio を使用して RPM ファイルを手動で解凍し、ifortvars.sh (.csh) ファイルを編集してコンパイラのインストール先ディレクトリを含めると、root 権限なしでもコンパイラをインストールできます。インストール・スクリプトでは、この手順を自動的に行います。

4. ライセンス・ファイルのディレクトリを入力します。
これは上記でライセンス・ファイル (*.lic) を保存したディレクトリです。インストール・プログラムは、Linux コンポーネントのインテル Fortran コンパイラをインストールする前に、有効なライセンスをチェックします。
5. ライセンスのチェック終了後、インストール・プログラムは既にインストールされているインテルのソフトウェア製品と次のインストール・メニュー項目を表示します。
 - 32 ビット・アプリケーション用インテル・コンパイラ (IA-32 システムにインストールする場合) または
Itanium アーキテクチャ用インテル・コンパイラ (Itanium ベース・システムにインストールする場合)
 - 32 ビット・アプリケーション用 Linux Application Debugger または
Itanium ベース・アプリケーション用 Linux Application Debugger
6. インストールするパッケージを選択します。
推奨するインストールの順序は次のとおりです：
 - 最初にインテル Fortran コンパイラをインストールします。
 - 次に、Linux Application Debugger (インテル・デバッガ、idb) をインストールします。インストールを開始する前に、エンドユーザ・ライセンス契約 (EULA) が表示されます。"accept" と入力してライセンス契約に同意します。
すべての製品を使用するために必要なパッケージがインストールされます。RPM パッケージが既にインストールされている場合は、インストール・スクリプトはこれを報告し、インストールが完了しなかったことを示します。その後、次の RPM パッケージのインストールを続行します。
既存のファイルを更新するには、デフォルトの RPM オプション -U --replacefiles を使用することを推奨します。
デフォルトのインストール・ディレクトリは、インテル Fortran コンパイラでは /opt/intel_fc_80/、インテル・デバッガでは /opt/intel_idb_xx/ です (xx: は "idb -V" を実行すると表示される、idb のバージョン番号です)。
7. インストールが完了すると、インストールされたインテル・パッケージ、それからインストール・メニューが再度表示されます。'x' を入力してインストール・スクリプトを終了します。

環境変数

インテル Fortran コンパイラ 8.0 は、定義されている場合、次の環境変数をサポートします:

- FPATH - INCLUDE および USE (コンパイル済み .mod) ファイルを検索する追加ディレクトリを指定します。ドライバは、これらを -I オプションに変換します。
- LIBRARY_PATH - リンカが検索する追加ディレクトリを指定します。ドライバは、これらを -L オプションに変換します。

コンパイラ環境の設定

インストール・スクリプト (install.sh) は、PATH、LD_LIBRARY_PATH、および MANPATH のような環境変数を設定するコンパイラ環境スクリプト・ファイル (ifortvars.sh/ idbvars.sh) を作成します。これらのスクリプト・ファイルをログイン・スクリプト (.login ファイル) に追加することを強く推奨します。".login" ファイルで一旦、変数が設定されれば、セッションごとにスクリプト・ファイルを実行する必要がなくなります。

コンパイラ環境を設定するスクリプトの実行

- > source <install-dir>/bin/ifortvars.sh(.csh)
(IA32 システムまたは Itanium ベース・システムで ifort を使用する場合)
- > source <install-dir>/bin/idbvars.sh(.csh)
(IA32 システムまたは Itanium ベース・システムで idb を使用する場合)

また、インストール・プログラムにより、すべてのコンパイル処理に共通の設定を含むコンパイラ設定ファイル (<インストール先ディレクトリ>/bin/ifort.cfg) が作成されます。これらのファイルを編集して、デフォルトのオプションを追加することができます。注: コンパイラのアップデート・パッケージをインストールする場合は、変更した設定ファイルを別のファイル名で保存し、ファイルが上書きされないようにする必要があります。

製品のインストールが完了したら、サポートへ登録してください。登録方法については、[「テクニカル・サポート」](#)を参照してください。

コンパイラとデバッガのアンインストール

下記の手順に従ってインテル・コンパイラとインテル・デバッガをアンインストールします。

1. root ユーザ権限でログインします。
2. コンパイラをアンインストールする場合は次のファイルを使用します:
> source <compiler-install-dir>/bin/uninstall.sh または
> source /opt/intel_fc_80/bin/uninstall.sh (コンパイラをデフォルトのディレクトリにインストールした場合)
3. デバッガをアンインストールするには次のファイルを使用します:
> source <debugger-install-dir>/bin/uninstall.sh または
> source /opt/intel_idb_73/bin/uninstall.sh (コンパイラをデフォルトのディレクトリにインストールした場合)

既知の制限事項

RPM 4.0.2 および RPM 4.1 インストール時のエラーについて

RPM 4.0.2 はデフォルト以外のディレクトリにはインストールできません。この問題は RPM 4.0.3 で修正されています。また、RPM 4.1 もデフォルト以外のディレクトリにはインストールできません。この問題は RPM 4.11 から 4.2 で修正されています。

インテル・デバッガのバージョン 7.3 のインストールについて

インテル Fortran コンパイラ 8.0 パッケージから IA-32 または Itanium ベース・アプリケーション用インテル・デバッガのバージョン 7.3 をインストールする場合、既にインストールされているインテル・デバッガのバージョン 7.1 または 7.2 は、7.3 にアップグレードされます。

libpthreads への静的リンクによるセグメンテーション違反

libpthreads.a を静的にリンクして作成されたアプリケーション (-static が使用された場合、libpthreads.a がデフォルトで使用されます) で 2MB 以上のスタック領域を使用すると、Linux のバージョンによってはセグメンテーション違反が発生することがあります。これは Linuxthreads の既知の制限です。この問題が発生した場合、libpthreads を動的にリンクしてください。Red Hat Linux 9 および Red Hat Enterprise Linux 3.0 では、nptl-devel パッケージをインストールして ifort コマンドラインで "-I/usr/include/nptl -L/usr/lib/nptl" を指定する方法もあります。このコマンドは、スタックサイズの制限がない、nptl でのみ動作し静的にリンクされるバイナリを作成します。

FLEXlm ライセンス・デーモン

FLEXlm ライセンス・デーモンは、Red Hat Linux 9 が動作しているシステムでは現在サポートされていません。このデーモンはフローティング・ライセンスを使用している場合のみ必要で、この場合は、サポートされる他のオペレーティング・システムで動作するシステムにライセンス・サーバをインストールしなければなりません。サーバを必要としないシングル・ユーザ・ライセンスを使用している場合は、影響ありません。

Itanium システムのダイナミック共有オブジェクト

Itanium システムでは、ダイナミック共有オブジェクト (.so) の作成を指定する `-shared` スイッチが使用された場合、`ld` は診断メッセージ "`@gprel relocation against dynamic symbol`" を出力します。この問題を解決するには、.so のビルドに使用されているコマンドラインに `-i_dynamic` および `-nofor_main` スイッチを追加します。インテル Fortran の将来のバージョンでは、これらのオプションは `-shared` が指定されたときに自動的に設定される予定です。

OpenMP の制限事項

大きなサイズのスタックを必要とする POSIX スレッドのプログラムの場合、一部の Linux のバージョンでは正常に動作しないことがあります。これは、Linux の POSIX スレッド・ライブラリのバージョンによっては、ハードコード化されたスタックサイズに制限があるためです。インテル・コンパイラでは POSIX スレッド・ライブラリを使用して OpenMP ベースおよび自動生成による並列化を実装しているため、上記の制約は OpenMP プログラム (`-openmp`)、およびインテル・コンパイラで自動生成された並列プログラム (`-parallel`) にも適用されます。通常、スタックの容量制限を超過したスレッド・プログラムを実行すると、セグメンテーション違反またはアドレッシング・エラーが発生します。

この制約を回避するには、`FLOATING_STACKS` パラメータを定義した `glibc`、通常はバージョン 2.2.4 以降 (IA-32、Itanium プロセッサ・ファミリとも) を使用するようにしてください。そして、`ulimit -s` コマンドを実行して最大シェル・スタック・サイズを十分な大きさの値 (単位:KB) に設定し、さらに環境変数 `KMP_STACKSIZE` を必要なスレッド・スタック・サイズ (単位:バイト) に設定してください。なお、シェル・スタック・サイズの最大値を `unlimited` と指定すると固定値に設定されてしまうため、問題を回避することができません。bash シェルでは、`ulimit -s` コマンドを使用して最大スタックサイズを設定できるのは 1 回のみであることに注意してください。C シェル (`csh`) では、`ulimit -stacksize` を使用することで、最大スタックサイズを何回でもリセットできます。`KMP_STACKSIZE` のデフォルト値は IA-32 の場合は 2 MB、Itanium ベース・システムの場合は 4 MB にまで引き上げられています。

上記の回避方法は RedHat 7.2 Linux ディストリビューションの `glibc` バージョン 2.2.4-13 (IA-32 版) および `glibc` 2.2.4-19 (Itanium プロセッサ・ファミリ版) で検証済みです。IA-32 版の `glibc` 2.2.4-13 に関しては、POSIX スレッド・ライブラリの共有バージョンを必ず使用してください (コンパイラの `.cfg` ファイル、またはコマンド・ラインで `-static` フラグを使用しないでください)。

さらに、共有ブロックが OpenMP ディレクティブで "`THREADPRIVATE`" として宣言された場合、共有ブロックの長さはブロックが宣言されたすべてのソースファイルで同じでなければなりません。

OpenMP とプロファイルに基づく最適化をともに行うと、プロファイル (.dyn ファイル) の生成のために、実行時間が長くなることがあります。これは既知の問題です。このバージョンのインテル Fortran コンパイラでは、OpenMP と組み合わせてプロファイルに基づく最適化を行うことは推奨しません。

一部のスレッド化 (OpenMP を含む) アプリケーションでは、Red Hat Linux 9 で導入された新しい NPTL ベース `pthread` ライブラリにリンクすると、問題が生じる場合があります。これは、環境変数 `LD_ASSUME_KERNEL=2.4.19` を設定することにより、解決できることがあります。この設定は、NPTL を実装していない以前の `pthread` ライブラリ・バージョンを使用します。

Red Hat Linux 7.2 における FPSWA 1.12 の問題

Itanium ベース・システムでは、システム BIOS の一部である Floating Point Software Assistance ハンドラ (FPSWA) が NaN や無限大のような例外値を生成する浮動小数点演算を完了します。FPSWA のバージョン 1.12 には、Red Hat Linux 7.2 システムでアプリケーションの実行中に予期しないセグメンテーション違反が発生する問題があります。インテルでは、この問題が Itanium システム用 Linux の他のディストリビューションで再現するかどうかは確認しておりません。

この問題を解決するには、システム BIOS を FPSWA のバージョン 1.18 以降を含むものにアップグレードしてください。BIOS のアップデート情報については、ご使用の Linux システムの製造元にお問い合わせください。

削除された機能

インテル Fortran コンパイラ 7.1 の次の機能はインテル Fortran 8.0 では利用できません:

- ランタイムの正確さをチェックする `-CA`、`-CS`、`-CU` および `-CV` オプション (配列と文字列境界を確認する `-CB` は利用可能です)
- コンパイル時の配列境界警告を有効にする `-WB` オプション (オプションを指定することはできませんが無視されます)
- バージョン 8.0 で実装が予定されていたローカル変数をゼロに初期化する `-zero` オプションはまだ実装されていません。最新情報は、インテル・プレミア・サポート・アカウントにログインしてご確認ください。

その他の情報

最新バージョンのインテル® C++ コンパイラに関する既知の制限事項および注意事項については、以下のリンク先を参照してください。

- [IA-32 アプリケーション開発用インテル Fortran コンパイラ](#) (英語)。 (IA-32 アプリケーション開発用コンパイラがインストールされている場合に限り参照可)
- [Itanium ベース・アプリケーション開発用インテル Fortran コンパイラ](#) (英語)。 (Itanium ベース・アプリケーション開発用コンパイラがインストールされている場合に限り参照可)

テクニカル・サポート

インテルでは、お客様のフィードバックを非常に重視しています。本製品で提供されるツールについてのテクニカル・サポートおよび FAQ や製品のアップデート情報を含むテクニカル情報を受けるには、<https://premier.intel.com> (英語) でインテル® プレミア・サポート・アカウントに登録してください。<http://support.intel.com/support/performance-tools/support.htm> (英語) で [Registration Center] をクリックして登録します。

注:

- サポートの登録はリリース製品またはプレリリース製品 (アルファ版、ベータ版など) により異なります。リリースされた製品のみに関するサポート Web ページがあります (<http://support.intel.com>) (英語)。
- 登録できない場合やプレミア・サポート・アカウントにアクセスできない場合は、developer.support@intel.com までご連絡ください。セキュリティ上の理由により、developer.support@intel.com では技術的な問題についてはお答えできませんのでご了承ください。
- パスワードを忘れた場合は、お手数ですが quad.support@intel.com にパスワードをリクエスト

トしてください。セキュリティ上の理由により、技術的な問題についてはお答えできませんのでご了承ください。

インテル Fortran コンパイラのユーザ・フォーラム、FAQ、ヒント、およびその他のサポート情報は、<http://support.intel.com/support/performance/tools/fortran/linux/> (英語) を参照してください。サポート情報全般については、<http://www.intel.com/software/products/support/> (英語) を参照してください。

問題の送信方法

問題を送信する手順:

1. <https://premier.intel.com/> (英語) にアクセスします。
2. ログイン名とパスワードを入力します。どちらも大文字と小文字が区別されます。
3. [Submit] ボタンをクリックします。
4. 「Confidentiality Statement (機密保護について)」を読み、[I Accept] ボタンを押します。
5. [Product] ドロップダウン・リストの隣の [Go] ボタンをクリックします。
6. 左側のナビゲーション・バーの [Submit Issue] リンクをクリックします。
7. [Product Type] ドロップダウン・リストから [Development Environment (tools,SDV,EAP)] を選択します。
8. ソフトウェアまたはライセンスに関する問題の場合は、[Product Name] ドロップダウン・リストから [Intel(R) Fortran Compiler for Linux*] を選択します。
9. 質問を入力します。ウィンドウの残りのフィールドも記入して、問題箇所の送信を完了します。

問題の報告および製品に関するご意見を送信される際のガイドライン

1. 問題、その他ご意見を入力してください。
問題の報告の場合は、その問題を再現できるように、できるだけ具体的に説明してください。コンパイラの問題報告の場合は、できるだけコンパイラ・オプションと簡単なテストコードを追加してください。
2. システム構成情報を入力します。

次のコマンドを使用して glibc とカーネルのバージョンを調べます:

```
uname -a  
-qa | grep glibc
```

rpm をインストールしている場合は、下記のコマンドを実行します:

```
ls /lib/libc*
```

プレミア・サポートの適切なフィールドに、表示された情報をコピーします。

下記のコマンドで、インテル Fortran コンパイラのパッケージ ID を取得します:

```
> ifort -V
```

出力された [Package ID] (例: `l_fc_p[c]_8.0.xxx`) をプレミア・サポートの適切なフィールドにコピーします。また、問題の解決に役立つと思われる事項はすべてお伝えください。

3. コンパイラをインストールできない場合、またはパッケージ ID を取得できない場合は、ダウンロードしたファイル名をパッケージ ID として入力してください。

解決済みの問題

コンパイラの最新バージョンで修正された問題点については、インテル・プレミア・サポート (<https://premier.intel.com/>) (英語) からダウンロードできる <package ID>_README (例: `l_fc_p[c]_8.0.xxx_README`) を参照してください。

ドキュメント

インテル・コンパイラのドキュメントおよび HTML ベースの関連ドキュメントは、ナビゲーション、インデックス検索、およびハイパーテキスト機能を備えており、ご使用の Web ブラウザでご覧になれます。[Adobe Acrobat Reader*](#) を使用して簡単に印刷できる、PDF 版のドキュメントも提供しています。

ドキュメントは <インストール先ディレクトリ>/doc (デフォルト: /opt/intel_fc_80/doc) ディレクトリにインストールされます。HTML 版のドキュメント・インデックスは、<インストール先ディレクトリ>/doc/fcompindex.htm (デフォルト: /opt/intel_fc_80/doc/fcompindex.htm) です。インタラクティブ形式 (HTML ベース) のトレーニング・チュートリアル、『Enhancing Performance with Intel Compilers』 (英語) もドキュメント・インデックスのリンクからアクセスできます。『Intel® Debugger Manual』 (英語) は、インテル・デバグガの doc ディレクトリにあります (デフォルトのディレクトリは、/opt/intel_idb_xx/doc です。xx: は、"idb -v" を実行すると表示される、idb のバージョン番号です)。

『Intel® Fortran コンパイラ ユーザーズ・ガイド』は、ご使用の Web ブラウザでご覧になれる HTML ベースのドキュメントで、次のように複数のドキュメントで構成されています:

- オプション・クイック・リファレンス・ガイド
- ユーザーズ・ガイド Vol I: アプリケーションのビルド
- ユーザーズ・ガイド Vol II: アプリケーションの最適化

以下のドキュメントは、Adobe Acrobat* 形式 (PDF ファイル) で提供されています:

- 『Intel® Fortran Language Reference』 (英語) は、組み込み関数を含む、インテル Fortran プログラミング言語について説明します。このドキュメントでは、Fortran 95 標準に対するインテル Fortran 拡張をビジュアルに説明します。このドキュメントは、Linux* 版インテル Fortran と Windows* 版インテル Visual Fortran の両方で利用できます。このドキュメントは、印刷された形式で別々に利用できます。
- 『Intel® Fortran ライブラリ・リファレンス』は、さまざまなインテル Fortran ライブラリに含まれるライブラリ・プロシージャについて説明し、ライブラリ・ルーチンをアルファベット順にリストします。このドキュメントは、Linux 版インテル Fortran と Windows 版インテル Visual Fortran の両方で利用できます。
- 『Linux* 版インテル® Fortran コンパイラ・ユーザーズ・ガイド』は、アプリケーションのビルドおよびアプリケーションの最適化に関する情報を、Vol I、Vol II で説明します。オプション・クイック・リファレンス・ガイドも提供します。これらのドキュメントの PDF バージョンは、印刷目的にのみ提供されています。これらのドキュメントの HTML バージョンは、ブラウザで表示して使用します。
- 『Using the Intel® License Manager for FLEXlm*』は、カウント方式ライセンスを使用するシステムで、Intel® License Manager for FLEXlm* をインストールして使用方法の詳細について説明します。

『Enhancing Performance with Intel Compilers』は、IA-32 および Itanium ベース・システムにおけるアプリケーションの最適化に役立つコンパイラ・オプションと Itanium アセンブラについて説明するインタラクティブな HTML 形式のチュートリアルです。Itanium アセンブラについて説明するドキュメントも提供されます。

Manpage の表示

ifort(1) manpage では、コマンドライン・オプションおよび ifort コンパイラ・コマンドに関する情報のリストが提供されています。ifort(1) manpage を表示するには、<インストール先ディレクトリ>/bin/ifortvars.*sh ファイルを実行する source コマンドを使用して環境を設定してから、次のコマ

ンドを入力してください:

```
$ man ifort
```

man コマンドは、表示された内容のスクロール、文字列の検索、特定の場所へのジャンプ、その他の機能を実行する単一キーまたはキーの組み合わせを提供します。例えば、z を入力すると次の画面、w を入力すると前の画面を表示します。man コマンドのヘルプを表示するには、h キーを入力します。ヘルプを閉じるには、q キーを入力して、表示されている manpage に戻ります。検索するには、/ 文字を入力してから検索する文字列 (/string) を入力して、Enter を押します。man コマンドのテキストの表示後、q を押してシェル・コマンドのプロンプトに戻ります。

HTML ドキュメントの表示

HTML 形式のユーザズ・ガイドを表示する際、以前のように Java* や Javascript が有効であるブラウザを使用する必要がなくなりました。このドキュメント形式は、標準 Red Hat* ディストリビューションに付属している Web ブラウザで使用できることが確認されています。HTML 形式のユーザズ・ガイドは Java サポートを必要とせず、さらにブラウザに依存しないようにするため、検索機能が削除されました。ユーザズ・ガイドで検索を行う必要がある場合は、提供されている PDF 形式のファイル (HTML 形式と同じ内容) を使用してください。

PDF ドキュメントの表示

PDF 形式のファイルを表示するには、xpdf ユーティリティ (検索機能を含む) を使用するか、[Adobe Acrobat Reader](#) をインストールします。Adobe Acrobat Reader を使用する場合は、より強力なナビゲーション機能を持つ Netscape から Acrobat を起動して PDF ドキュメントを表示することを推奨します。Netscape から Acrobat Reader を起動できるようにするには、環境変数 PATH で指定したディレクトリに Acrobat Reader をインストールした上で、ブラウザの設定を変更する必要があります。注: Acrobat Reader の設定が正しく行われていないと、PDF ファイルが上書きされ、再インストールが必要となる場合があります。設定に不備がある場合、ブラウザで [ファイル名をつけて保存] のダイアログが表示されることがあります。ここで、[OK] をクリックしてしまうと PDF ドキュメント・ファイルが上書きされます。

Acrobat Reader の設定を更新するには、次の手順を行ってください。

```
[編集] - [設定] - [Navigator] - [アプリケーション] - [新規]
説明: Portable Document Format
MIME タイプ: application/pdf
拡張子: pdf
アプリケーション: acroread %s
```

または、ホーム・ディレクトリの .mailcap ファイルに次の行を追加して、acroread を設定することもできます。

```
application/pdf; acroread %s
```

Netscape ブラウザのバージョンによっては、[自動的に画像を読み込む] ([編集] - [設定] - [詳細] - [自動的に画像を読み込む]) のオプションを無効にしておかないと、HTML ドキュメント・ファイルを開く際にブラウザがフリーズすることがあります。この場合、ドキュメントの画像をブラウザで表示するには各画像をクリックする必要があります。

追加情報

関連製品とサービス

インテルのソフトウェア開発製品の詳しい情報については、<http://www.intel.com/jp/developer/software/products/> を参照してください。関連製品の一部を次にリストします:

- [インテル® ソフトウェア・カレッジ](#) (英語) では、最先端のソフトウェア開発テクノロジーに関するトレーニングを提供します。トレーニングには、インテルのすべてのアーキテクチャ、プラットフォーム、およびテクノロジーに関するトピックが含まれ、オンライン・コースやインストラクタが教えるコースなどがあります。
- インテル Fortran コンパイラのユーザ・フォーラム、FAQ、ヒント、およびその他のサポート情報は、<http://support.intel.com/support/performance-tools/fortran/linux/> (英語) をご覧ください。サポート情報全般については、<http://www.intel.com/software/products/support/> (英語) を参照してください。
- [インテル® VTune™ パフォーマンス・アナライザ](#)は、アプリケーションがどのように CPU を利用するかを評価し、アプリケーションのパフォーマンスの向上に必要な修正部分を特定するのに役立ちます。
- [インテル C++ コンパイラおよびインテル Fortran コンパイラ](#)は、ソフトウェアの実行の高速化に重要な役割を果たします。また、最新のインテル IA-32 プロセッサおよびインテル Itanium プロセッサをサポートしています。
- [インテル® パフォーマンス・ライブラリ](#)には、各種のインテル・プロセッサ向けに最適化されたルーチン群が収められています。[インテル® マス・カーネル・ライブラリ](#)は、数学、科学、工学などのソフトウェア開発に役立つ、最新のインテル Pentium プロセッサおよびインテル Itanium プロセッサ向けに最適化された線形代数、高速フーリエ変換、およびベクトル演算関数を提供します。[インテル® インテグレートッド・パフォーマンス・プリミティブ](#)は、複数のインテル・アーキテクチャおよびオペレーティング・システム用のハイ・パフォーマンスなソフトウェアをビルドするクロス・プラットフォーム・ツールで構成されています。

著作権と商標について

Intel、インテル、Itanium、Pentium、VTune は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

* 一般にブランド名または商品名は各社の商標または登録商標です。

(C) 2001 - 2004, Intel Corporation. 無断での引用、転載を禁じます。