

インテル® Fortran コンパイラー & インテル® Fortran コンパイラー・クラシック・リリースノート

本書は、英文「[Intel® Fortran Compiler Release Notes and New Features](#)」(英語)の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

バージョン: 2024.0.0

2023 年 11 月 16 日

このドキュメントでは、インテルの Fortran コンパイラーの新機能および変更された機能について説明します。また、製品ドキュメントに記述されていない機能と問題についても説明します。

インテル® HPC ツールキット・パッケージは 2 つの Fortran コンパイラーを提供します。

- インテル® Fortran コンパイラー (ifx) は、ifort のすべての Fortran 機能に加えて、インテルの GPU へオフロードする OpenMP* ディレクティブの機能をサポートします。ifx のインテルの GPU をターゲットとする OpenMP* 5.0、5.1、5.2 および 6.0 の target 機能は ifort では利用できません。
- ifort は従来のインテル® コンパイラーのテクノロジーを使用しており、ifx へ移行するユーザーをサポートするために提供されています。ifort は、この移行期間中に ifx とともに提供されます。ifort は、Windows* 32 ビット・ライブラリーおよびアプリケーション向けのバイナリーを作成する機能を提供します (ifx は 64 ビット・バイナリーのみ作成します)。

インテル® Fortran コンパイラー (ifx) は、インテルのフラッグシップ Fortran コンパイラーです。ifx は、ifort のフロントエンドとランタイム・ライブラリーをベースに、新しい LLVM バックエンド・コンパイラー・テクノロジーを使用しています。ifx は、従来の DEC 拡張機能、優れた Fortran 標準規格のサポート、強力なコンパイラー・ディレクティブなど、馴染みのある ifort と同じ Fortran 機能を提供します。ifx では、さらにパフォーマンスを向上する OpenMP* TARGET やインテルの GPU への自動 DO CONCURRENT オフロードを含む、インテルの最新の CPU と GPU 向けの新しい機能が追加されます。

ifort と ifx はどちらも、Fortran 2018 までの Fortran 標準規格を完全にサポートしており、一部の Fortran 2023 機能もサポートします。

ifx および ifort はバイナリー (.o / .obj) とモジュール (.mod) ファイルの互換性があります。ifort で生成されたバイナリーやライブラリーは、ifx でビルドされたバイナリーやライブラリーとリンクできます。また、一方のコンパイラーで生成された .mod ファイルは、他方のコンパイラーで使用できます (64 ビット・ターゲットのみ、-ipo を指定してコンパイルしている場合を除く)。どちらのコンパイラーも同じランタイム・ライブラリーを使用します。複数のユーザーから、ifx でコンパイルされたアプリケーションのパフォーマンスは ifort でコンパイルされた同じアプリケーションと同等であるとの報告を受けています。

注: インテル® Fortran コンパイラー・クラシック (ifort) は非推奨になりました。2024 年後半のリリースで削除される予定です。Windows* および Linux* の継続的なサポート、新しい言語のサポート、新しい言語機能、最適化のサポートが提供される、LLVM ベースのインテル® Fortran コンパイラー (ifx) へ移行することを推奨します。ifx の詳細は、『[インテル® Fortran コンパイラー・デベロッパー・ガイドおよびリファレンス](#)』(英語) および『[ifort ユーザー向け ifx 移行ガイド](#)』を参照してください。

注: macOS* のサポートは、インテル® Fortran コンパイラー・クラシック 2024.0 で削除されました。

動作環境

「[Intel® Fortran コンパイラーの動作環境](#)」を参照してください。

リリースの入手方法

[こちらのサイト](#) (英語) から手順に従って Intel® HPC ツールキットをダウンロードし、インストール手順に従ってインストールします。すべての機能を使用するには、最初に Intel® oneAPI ベース・ツールキットをインストールしてください。[スタンドアロン・コンポーネント・ページ](#) (英語) から Fortran コンパイラーのみダウンロードすることもできます。

変更履歴

このセクションではマイナーリリースにおける重要な変更内容を説明します。

Intel® Fortran コンパイラーおよび Intel® Fortran コンパイラー・クラシック 2024.0.0 リリースの新機能

Intel® Fortran コンパイラー 2024.0.0

- Intel® Fortran コンパイラーに、Fortran、OpenMP*、OpenMP* オフロードアドレス、メモリーリーク、初期化されていないメモリー、スレッドデータ競合、デッドロック、CPU 上の未定義の動作を開発者が検出できる LLVM サニタイザーが追加されました。生産性を大幅に向上させます。
- Intel® Fortran コンパイラーに初期 Fortran 2023 標準規格のサポートが追加され、OpenMP* 5.0、5.1 および 5.2 の対応が強化されました。また、初期 OpenMP* 6.0 標準規格に準拠しました。
- 報告された問題を修正しました。
- 機能やセキュリティ・アップデートなど、サードパーティー・コンポーネントの最新バージョンが含まれました。

Intel® Fortran コンパイラー・クラシック 2021.11.0

- 報告された問題を修正しました。
- 機能やセキュリティ・アップデートなど、サードパーティー・コンポーネントの最新バージョンが含まれました。

ディレクトリー・レイアウトの変更

インストールとセットアップを合理化するため、すべての製品のディレクトリー・レイアウトを変更しました。

統合ディレクトリー・レイアウトは 2024.0 で実装されました。ツールキットの複数のバージョンをインストールしている場合、統合レイアウトにより、インストールしているツールキットのバージョンに対応する正しいコンポーネントのバージョンが開発環境に含まれるようになります。2024.0 以前に使用されていたコンポーネント・ディレクトリー・レイアウトは、新規および既存のインストールで引き続きサポートします。

環境の初期化方法や統合レイアウトの利点など、統合レイアウトの詳細は、「[Linux* での setvars および oneapi-vars スクリプトの使用](#)」(英語) および「[Windows* での setvars および oneapi-vars スクリプトの使用](#)」(英語) を参照してください。

コンパイラーの機能の変更点

新機能

- インテル® Fortran コンパイラー (ifx) は Microsoft* Visual Studio* (Windows*) を使用している場合のデフォルトの Fortran コンパイラーです。

ifx 2024.0.0 および ifort 2021.11.0

初期 Fortran 2023 の機能

- DO CONCURRENT の REDUCE 局所性指定子
- INTEGER および REAL 代入文の RHS の BOZ 定数
- PARAMETER 文の INTEGER および REAL 値としての BOZ 定数
- 配列コンストラクター値としての INTEGER または REAL としての BOZ 定数
- ENUMERATION 文の INTEGER 値としての BOZ 定数
- 度数を返す組込み三角関数 ACOSD(X)、ASIND(X)、ATAN2D(Y, X)、COSD(X)、SIND(X)、および TAND(X)

新しい OpenMP* の機能

- ORDERED ディレクティブで、DOACROSS 節は DEPEND 節を置換します。
- OpenMP* 5.2 は一部のディレクティブを PURE として分類します。Fortran PURE プロシージャでは PURE ディレクティブのみ許可されます。コンパイラーは、PURE プロシージャで非 PURE ディレクティブを検出するとメッセージを出力します。

ifx 2024.0.0

新しい OpenMP* の機能

- MATCH 節が DECLARE VARIANT ディレクティブで許可されるようになりました (OpenMP* 5.0)。
- DEPOBJ 構造の DESTROY 節に destroy-var 修飾子が必要になりました (OpenMP 5.1)。
- TASKLOOP ディレクティブの GRAINSIZE 節および NUM_TASK 節に STRICT 修飾子を指定できるようになりました (OpenMP* 5.1)。
- オプションの ORDER 節の REPRODUCIBLE および UNCONSTRAINED 修飾子を CONCURRENT 引数に指定してループの反復の順序を指定できます (OpenMP* 5.2)。
- ORDER 節を DISTRIBUTE 構造で指定できるようになりました (OpenMP* 5.1)。
- GROUPPRIVATE ディレクティブはコンテキスト・グループのスレッドで共有される変数を指定しますが、各コンテキスト・グループはプライベート・コピーを保持します (OpenMP* 6.0)。
- DEVICE_TYPE 節を DECLARE TARGET ディレクティブで指定できるようになりました。DEVICE_TYPE(HOST) はホストデバイスのみの構造の本体をコンパイルし、DEVICE_TYPE(NOHOST) はターゲットデバイスの本体をコンパイルし、DEVICE_TYPE(ANY) はホストとターゲットの両方の本体をコンパイルします (OpenMP* 5.0)。

新しいコンパイラー・オプション

- `-fopenmp-device-code-split`(Linux*)および `/Qopenmp-device-code-split` (Windows*)は、OpenMP* オフロード Ahead-Of-Time コンパイルで SPIR-V* カーネルの並列コンパイルを有効にします。
- `-fopenmp-max-parallel-link-jobs`(Linux*)および `/Qopenmp-max-parallel-link-jobs` (Windows*)は、適用可能な場合、デバイスのリンク手順中に実行される並列アクションの最大数を決定します。
- `-fsanitize=address`(Linux*)および `/fsanitize:address`(Windows*)は、LLVM メモリー・サニタイザー (MSAM) メモリーエラー検出機能を有効にします。
- `-fsanitize=threads` は、LLVM メモリー・サニタイザー (MSAM) を有効にして、OpenMP* およびスレッドコードのデータ競合を検出できるようにします (Linux* のみ)。
- `-fprofile-ml-use` は、事前トレーニング済みのマシンラーニング・モデルを使用して、プロファイルに基づく最適化を推進する分岐実行確率を予測できるようにします。
- `-mno-gather`(Linux*)および `/Qgather-`(Windows*)は、自動ベクトル化でのギャザー命令の生成を無効にします。
- `-mno-scatter`(Linux*)および `/Qscatter-`(Windows*)は、自動ベクトル化でのスキッター命令の生成を無効にします。
- `-mauto-arch`(Linux*)および `/Qauto-arch`(Windows*)は、パフォーマンス上の利点がある場合、x86 アーキテクチャー・プロセッサ向けに複数の機能固有の自動ディスパッチ・コードを生成するようにコンパイラーに指示します。
- `-flink-huge-device-code` は、リンクされたバイナリーの後半にデバイスコードを配置するようにコンパイラーに指示します。これは、デバイスコードが 2GB を超えるときに、周囲の実行可能およびリンク可能形式 (ELF) セクション間での 32 ビット PC 相対再配置を防ぐためです。以前のオプション `-fsycl-link-huge-device-code` は非推奨になりました。将来のリリースで削除される予定です。

新しいデバイス階層

2024.0 リリース以降、新しいデバイス階層モデルが導入され、インテル® データセンター GPU マックス・シリーズの各 GPU タイルを独立した GPU デバイスとして公開するデフォルトモードが導入されます。この変更は、最新のアーキテクチャーの GPU のパフォーマンスを引き出す最適な方法を見つけるためにインテルとユーザーが協力して行った作業に基づいて行われたものです。詳細は、『[oneAPI GPU 最適化ガイド](#)』および「[GPU タイル階層を使用するためのオプション](#)」(英語)を参照してください。

参考情報

インテル® Fortran コンパイラー (ifx) の Fortran および OpenMP* 標準の実装の詳細は、[こちらの記事](#)を参照してください。

終了予定のサポート

インテル® Fortran コンパイラー・クラシック (ifort) は非推奨になりました。2024 年後半のリリースで削除される予定です。Windows* および Linux* の継続的なサポート、新しい言語のサポート、新しい言語機能、最適化のサポートが提供される、LLVM ベースのインテル® Fortran コンパイラー (ifx) へ移行することを推奨します。ifx の詳細は、『[インテル® Fortran コンパイラー・デベロッパー・ガイドおよびリファレンス](#)』（英語）および『[ifort ユーザー向け ifx 移行ガイド](#)』を参照してください。

インテル® Fortran コンパイラー (ifx)

- `-fsycl-link-huge-device-code` は古いオプション (非推奨) で、将来のリリースで削除される予定です。代わりに `-flink-huge-device-code` を使用してください。
- `-fopenmp` は古いオプション (非推奨) で、将来のリリースで削除される予定です。代わりに `-qopenmp` または `-fiopenmp` を使用してください。
- Linux* の `-mkl` コンパイラー・オプションは古いオプション (非推奨) で、将来のリリースで削除される予定です。2021.4.0 リリースから、代替オプションとして `-qmkl` が追加されました。このコンパイラー・オプションは、インテル® oneAPI マス・カーネル・ライブラリー (インテル® oneMKL) の必要なライブラリーにリンクするようにコンパイラーに指示します。

インテル® Fortran コンパイラー・クラシック (ifort)

- ディレクティブ SIMD (`!dir$ SIMD`) は古い機能 (非推奨) で、将来のリリースで削除される予定です。ユーザーは、ソースコードで、このディレクティブを OpenMP* SIMD ディレクティブ `!$omp simd` および関連する節に置換する必要があります。OpenMP* SIMD ディレクティブを使用するときは、`-qopenmp-simd` または `-qopenmp` コンパイラー・オプションを追加することを忘れないでください。`-O2` を使用してコンパイルすると、`-qopenmp-simd` が暗黙的に指定されます。
- Linux* の `-mkl` コンパイラー・オプションは古いオプション (非推奨) で、将来のリリースで削除される予定です。2021.4.0 リリースから、代替オプションとして `-qmkl` が追加されました。このコンパイラー・オプションは、インテル® oneAPI マス・カーネル・ライブラリー (インテル® oneMKL) の必要なライブラリーにリンクするようにコンパイラーに指示します。

削除されたサポート

2021.7 以降、macOS* のサポートはインテル® プロセッサーを搭載している Mac* コンピューターのみを対象としていることに注意してください。

macOS* のサポート

macOS* のサポートは、インテル® Fortran コンパイラー・クラシック 2024.0 で削除されました。

インテル® C++ コンパイラー・クラシック (icc/icpc)

インテル® C++ コンパイラー・クラシック (icc/icpc) は 2024.0 リリースで削除されました。Windows* および Linux* の継続的なサポート、新しい言語のサポート、新しい言語機能、最適化のサポートが提供される、LLVM ベースのインテル® oneAPI DPC++/C++ コンパイラー (icx/icpx) へ移行することを推奨します。icx/icpx の詳細は、『[インテル® oneAPI DPC++/C++ コンパイラー・デベロッパー・ガイドおよびリファレンス](#)』（英語）および『[icc ユーザー向け dpcpp/icx 移行ガイド](#)』を参照してください。

既知の問題

Windows* 上でのインストール

Windows* パフォーマンス・ツールキット (Win11SDK_WindowsPerformanceToolkit) がインストールされているオフラインシステム上の Microsoft* Visual Studio* 2022 (17.7 以降) に Intel® ソフトウェア開発ツール (Intel® oneAPI ベース・ツールキット、Intel® HPC ツールキット、またはそれらのコンポーネント) を統合する際に既知の問題があり、正しく統合されません。この問題を回避するには、Intel® ソフトウェア開発ツールのインストール中にインターネット接続を有効にするか、Intel® ソフトウェア開発ツールをインストールする前に Windows* パフォーマンス・ツールキットをアンインストールし、Intel® ソフトウェア開発ツールをインストールした後に Windows* パフォーマンス・ツールキットを再インストールしてください。

非 POSIX* シェル

環境起動スクリプトは POSIX* 準拠のスクリプトに移行されました。csh などの非 POSIX* シェルでは、次のコマンドラインを使用した回避方法があります。

コンポーネント・ディレクトリー・レイアウトの場合。

```
% bash -c 'source /opt/intel/oneapi/setvars.sh ; exec csh'
```

統合ディレクトリー・レイアウトの場合。

```
% bash -c 'source <install-dir>/<toolkit-version>/oneapi-vars.sh ; exec csh'
```

ifx および ifort

これらの説明は ifx および ifort コンパイラー・ドライバーの両方を使用した場合に適用されます。

コンパイル時の整数式の評価

- Intel® Fortran コンパイラー (ifx) バージョン 2023.2.0 および Intel® Fortran コンパイラー クラシック (ifort) バージョン 2021.10.0 以降
 - コンパイル時の整数式の評価でオーバーフローが発生した場合でも、ifort と ifx が警告を出力しない問題が修正されました。
 - コンパイル時の整数式の評価でオーバーフローが発生すべき場合であっても、ifort と ifx は代わりに値を暗黙的に大きな KIND である INTEGER(8) にプロモートして正しい数値結果を生成していました。この問題が部分的に修正され、ほとんどの式はオーバーフローするようになりました。数値に差が生じる可能性があります。オーバーフローの警告が出力されます。下位互換性のために、式のすべての値が明示的な KIND パラメーターのない整数リテラルである一部のケースでは、INTEGER(8) への自動プロモートが引き続き行われます。この場合も警告が出力されます。

COMMON ブロック

- Intel® Fortran コンパイラー (ifx) バージョン 2023.2.0 および Intel® Fortran コンパイラー クラシック (ifort) バージョン 2021.10.0 以降
 - OpenMP* データ共有節で個々の COMMON ブロック変数ではなく COMMON ブロックを宣言するプログラムは、ランタイムエラー (セグメンテーション・フォールトや不正な結果) を引き起こします。この問題を回避するには、個々の COMMON ブロック変数を宣言します。

モジュール(.mod)ファイル

インテル® Fortran コンパイラー (ifx)バージョン 2023.1.0 およびインテル® Fortran コンパイラー・クラシック (ifort)バージョン 2021.9.0 以降、モジュール (.mod) ファイルは次のバージョンで作成されたモジュールファイルとのみ互換性があります。

- インテル® Fortran コンパイラー [Composer | Professional] Edition (Linux* 版) バージョン 11.0 以降
- インテル® Visual Fortran コンパイラー [Composer | Professional] Edition (Windows* 版) バージョン 11.0 以降
- インテル® Fortran コンパイラー [Composer | Professional] Edition (macOS* 版) バージョン 11.0 以降

バージョン 11.0 より前のインテル® Fortran コンパイラーで作成されたモジュールファイルとは互換性がありません。

GDB 10.0 およびそれ以前のバージョン

インテル® oneAPI 2023.0 とインテル® oneAPI 2023.1 の間の変更により、GDB 10.0 およびそれ以前のバージョンでは、インテル® C コンパイラー、インテル® C++ コンパイラー、インテル® Fortran コンパイラーで生成された SYCL* および OpenMP* CPU オフロードコードを適切にデバッグできなくなりました。これらの古いバージョンの GDB は、Red Hat* EL8、Ubuntu* 20.04、および Rocky Linux* 8 に含まれています。

回避方法

1. インテル® ディストリビューションの GDB を使用します。
2. バージョン 10.0 以降のオープンソース・バージョンの GDB をダウンロードします。

FAIL IMAGE を実行した後に Fortran Co-Array アプリケーションがハングする

FAIL IMAGE ステートメントを使用してイメージを失敗させる場合、ステートメントまたは操作で許可されている場合に失敗したイメージに遭遇する可能性のあるすべての Co-Array 操作で STAT= 指定子または STAT 引数を使用するか、assume failed_images または standard-semantics コンパイラー・オプションを指定する必要があります。STAT= 指定子、STAT 引数を使用しない場合、またはこれらのコンパイラー・オプションを指定しない場合、これらの操作は失敗したイメージをチェックしないで、失敗したイメージからの応答を待ちますが、応答が行われなため、アプリケーションがハングします。

次の例は、STAT= 指定子を使用してアプリケーションのハングを防ぐ方法を示します。

```
SUBROUTINE FAIL_AND_SYNC ( THIS_ONE )
  INTEGER THIS_ONE
  INTEGER MY_STAT
  IF (THIS_IMAGE() .EQ. THIS_ONE) THEN
    FAIL_IMAGE
  END IF
  SYNC ALL (STAT=MY_STAT) ! STAT= がないとハング
END SUBROUTINE FAIL_AND_SYNC
```

Red Hat* 8 での Co-Array 機能の使用

Co-Array 機能を使用しているアプリケーションを Red Hat* 8.0 以降で実行する場合、インテル® MPI ライブラリー 2019 Update 7 以降をインストールする必要があります。Update 7 よりも前のバージョンでは、アプリケーションが正常に完了せず、失敗に関する情報がほとんど得られません。

ifort

これらの説明は ifort コンパイラー・ドライバーを使用した場合に適用されます。

クロスコンパイル(Microsoft* Windows* 64 ビット環境での 32 ビット・モードでのコンパイル)

Microsoft* Windows* 64 ビット環境でクロスコンパイル(32 ビット・モードでのコンパイル)を行い Microsoft* Visual Studio* 2019 を使用してコードで例外(特に浮動小数点例外)処理を行っている場合、浮動小数点例外が発生したときに SEGV 例外が表示され、意図したとおりに例外が処理されません。

この問題を解決するには、リンクコマンドに `legacy_x86_flt_exceptions.lib` を追加します。

ifx

これらの説明は ifx コンパイラー・ドライバーを使用した場合に適用されます。

opt-reports

「schedule(dynamic)」を含む OpenMP* ループ構造を含むプログラムで `opt-report` を使用すると、コンパイラーがエラーを出力する既知の問題があります。この場合、`-qopt-report` を削除してコンパイルすることを推奨します。

特定の OpenMP* オフロード・ディレクティブを使用すると正しい結果が得られない可能性

TEAM 構造内部の入れ子の並列ループが REDUCTION 節で変数を使用していて、TEAM 構造に同じ REDUCTION 節がない場合、OpenMP* プラグマを使用してインテル® GPU にオフロードすると、正しい結果が得られない可能性があります。

正しくない結果を回避するには、`-mllvm -vpo-paropt-atomic-free-reduction-slm=true` を指定してコンパイルし、グローバル・メモリー・バッファを無効にします。

-ipo オプションと ifx/ifort オブジェクト・ファイルの使用

ifx はバイナリー(.o/.obj)とモジュール(.mod)ファイルの互換性があります。ifort で生成されたバイナリーやライブラリーは、ifx でビルドされたバイナリーやライブラリーとリンクできます。また、一方のコンパイラーで生成された .mod ファイルは、他方のコンパイラーで使用できます(64 ビット・ターゲットのみ)。`-ipo` オプションを指定してコンパイルすると、ifort と ifx 間の互換性はなくなります。

GPU ドライバーのハング問題

ネイティブ環境で実行時間の長い CPU 計算ワークロードを持つアプリケーションがある場合、ワークロードが終了されないようにハングチェック・タイムアウト期間を無効にする必要があります。詳細は、各 OS の「[インストールガイド](#)」(英語)を参照してください。

CMake* の情報

- CMake* 3.25 以降では、CMake* がリンカーフラグを追跡する方法の制限により、CMake* プロジェクトに LLVM ベースのコンパイラー (icx または ifx) とクラシックバージョンのコンパイラー (icc または ifort) の C/C++ コードと Fortran コードが混在していると、無効なリンクフラグが生成されます。例えば、CMake* は、C/C++ コンパイラーとして「icx」を使用し、Fortran コンパイラーとして「ifort」を使用したプロジェクトをビルドできません。
 - この問題を回避するには、混在言語アプリケーションを作成するときに LLVM ベースのコンパイラーのみを使用します。
- CMake* 3.27 以降、インテル® oneAPI DPC++/C++ コンパイラーとインテル® Fortran コンパイラーのサポートが更新され、リンカーの代わりにコンパイラー・ドライバーをリンクに使用できるようになりました。この変更により、SYCL* アプリケーションとライブラリーの作成のユースケースが有効になり、Windows* 上でプロシージャ間の最適化 (IPO) が有効になります。

設定スクリプトのレポート: 「linking to Fortran libraries from C fails (C から Fortran ライブラリーのリンクに失敗)」

症状

ifx でコンパイルしたとき、GNU* Autconf により生成された ./configure スクリプトで次のようなエラーメッセージがレポートされる。

```
checking for Fortran 77 libraries of ifx... -loopopt=0 -L/lib/./lib64 -L/lib/./lib64/ -L/usr/lib/./lib64 -L/usr/lib/./lib64/ -L/lib64 -L/lib/ -L/usr/lib64 -L/usr/lib -lifport -lifcoremt -limf -lsvml -lm -lipgo -lirc -lpthread -lirc_s -ldl
configure: WARNING: FLIBS does not work
checking for ifx flag to add single underscore to external names... none
checking for dummy main to link with Fortran 77 libraries... unknown
configure: error: in `/path/to/build/dir':
configure: error: linking to Fortran libraries from C fails
See `config.log' for more details
make: *** [build/config.status] Error 1
```

config.log ファイルを調べると、次のエラーが原因で ./configure が終了したことが分かります。

```
ld: cannot find -loopopt=0
```

問題

Fortran コードを C または C++ コードとリンクするために必要なライブラリーを決定する際に、GNU* Autoconf 2.69 以前は、詳細なコンパイラー出力の `-mllvm -loopopt=0` の `-loopopt=0` を誤ってリンカーフラグとして解釈します。GNU* Autoconf は `-loopopt=0` を FLIBS 変数に追加して、リンカーに渡します。リンカーは、存在しないライブラリーを探して、リンクに失敗します。

ソースコードをダウンロードしているユーザー向けのソリューション

このエラーの影響を受けるパッケージのメンテナーに、このリリースノートを参照するように伝えてください。configure スクリプトの処理方法はプロジェクトによって異なりますが、いくつかの一般的な規則が適用されます。パッケージメンテナーからの応答の前に問題を解決する必要があるユーザーは、configure スクリプトを自分で更新してください。更新の難易度は、パッケージの配布方法によって異なります。

更新するには、[パッケージメンテナー向けのソリューション](#)で説明されているように、GNU* Autoconf-2.70 以降をインストールします。

更新された autoconf をユーザーのパスにインストールした後、configure スクリプトと configure.ac ファイルを配布するパッケージを更新します。

```
autoreconf -if
```

これで、configure スクリプトが FLIBS エラーなしで完了するようになります。

プロジェクト・メンテナーが、tar や zip アーカイブでのソース・ディストリビューションから configure.ac ファイルを削除することは珍しくありません。その場合、ユーザーは、コード・リポジトリからプロジェクトをダウンロードして、プロジェクトの指示に従ってビルドする必要があります。configure スクリプトのバージョンがバージョン・コントロールにコミットされ、自動的に再生成されないことがあります。その場合、上記のように autoreconf -if を実行するか、configure スクリプトを削除すると、スクリプトが再生成されます。プロジェクトのドキュメントに configure スクリプトを再生成する方法が記載されている場合は、その指示に従ってください。

パッケージメンテナー向けのソリューション

ユーザーにエラーが表示されないようにするには、GNU* Autoconf をバージョン 2.70 以降に更新して、プロジェクトの configure スクリプトを再生成します。GNU* Autoconf 2.70 は 2020 年 12 月 8 日に[リリース](#) (英語)されました。ソースコードは `git clone http://git.sv.gnu.org/r/autoconf.git` で入手できます。GNU Autoconf のドキュメントは、[GNU Autoconf プロジェクト・ページ](#) (英語)から入手できます。

インテル® Fortran コンパイラーのインストール方法

インテル® oneAPI ツールキットのインストール・ガイドは、オンラインでも利用できます。詳細は、「[インテル® oneAPI ツールキットのインストール・ガイド](#)」(英語)を参照してください。

インテル® Fortran コンパイラーの使用方法

関連情報

- [インテル® HPC ツールキット\(Linux* 版\)導入ガイド](#) (英語)
- [インテル® HPC ツールキット\(Windows* 版\)導入ガイド](#) (英語)
- [インテル® oneAPI DPC/C++ コンパイラーおよびインテル® Fortran コンパイラー向けの GPU への OpenMP* オフロード導入ガイド](#) (英語)

ドキュメント

製品のドキュメントはオンラインで提供されています。

- [インテル® Fortran コンパイラー・クラシックおよびインテル® Fortran コンパイラー・デベロッパー・ガイドおよびリファレンス](#) (英語)
- ifx の詳細と Fortran および OpenMP* 標準の実装の詳細は、[こちらの記事](#)を参照してください。
- セマンティック・バージョンングに基づく [oneAPI バージョン管理スキーマ](#) (英語)

テクニカルサポート

商用サポートを利用可能な場合は、[サポートチケット](#)を作成してください。利用できない場合は、[インテル® Fortran コンパイラー・フォーラム](#) (英語)でサポートを受けることができます。

インテル® Fortran コンパイラー・リリースノート (以前のメジャーリリース)

- [インテル® oneAPI 2023](#)
- [インテル® oneAPI 2022](#)
- [インテル® oneAPI 2021](#) (英語)

インテル® Fortran コンパイラーの動作環境

本書は、英文「[Intel® Fortran Compiler for oneAPI System Requirements](#)」(英語)の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

バージョン: 2024.0.0

2023 年 11 月 16 日

このドキュメントでは、インテル® Fortran コンパイラー (ifx) およびインテル® Fortran コンパイラー・クラシック (ifort) のハードウェア、オペレーティング・システム、ソフトウェア要件を説明します。開発環境のオプションに関する情報も含まれています。

インテル® ソフトウェア開発ツールのすべての機能を利用して最適なパフォーマンスが得られるように、インテル® oneAPI ベース・ツールキットとインテル® HPC ツールキットの両方をインストールすることを推奨します。利用可能なすべてのインテル® パフォーマンス・ライブラリーを入手するには、両方のツールキットをダウンロードしてインストールする必要があります。ツールキットの入手方法およびインストール方法は、「[インテル® oneAPI ツールキットのインストール・ガイド](#)」(英語)を参照してください。

詳細は、「[インテル® Fortran コンパイラー・リリースノート](#)」を参照してください。

ディレクトリー・レイアウトの変更

インストールとセットアップを合理化するため、すべての製品のディレクトリー・レイアウトを変更しました。

統合ディレクトリー・レイアウトは 2024.0 で実装されました。ツールキットの複数のバージョンをインストールしている場合、統合レイアウトにより、インストールしているツールキットのバージョンに対応する正しいコンポーネントのバージョンが開発環境に含まれるようになります。

2024.0 以前に使用されていたコンポーネント・ディレクトリー・レイアウトは、新規および既存のインストールで引き続きサポートします。

環境の初期化方法や統合レイアウトの利点など、統合レイアウトの詳細は、「[Linux* での setvars および oneapi-vars スクリプトの使用](#)」(英語)および「[Windows* での setvars および oneapi-vars スクリプトの使用](#)」(英語)を参照してください。

ハードウェア要件

- プロセッサ (CPU)
 - インテル® Core™ プロセッサ・ファミリー
 - インテル® Xeon® プロセッサ・ファミリー
 - インテル® Xeon® スケーラブル・プロセッサ・ファミリー
- アクセラレーター (インテル® Fortran コンパイラー (ifx) のみ)
 - インテル® インテグレートッド・グラフィックス (第 11 世代以降)
 - インテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio、略称 PVC)

- インテル® データセンター GPU フレックス・シリーズ(開発コード名 Arctic Sound-M、略称 ATS-M)
- インテル® Arc™ グラフィックス(dGPU)
- インテル® Iris® Xe MAX グラフィックス(iGPU)
- メモリー(RAM):8GB(推奨)
- 14GB のディスク空き容量

ソフトウェア要件

インテル® Fortran コンパイラーはインテル® HPC ツールキットに含まれています。

すべてのソフトウェア・ツールを利用できるように、[インテル® oneAPI ベース・ツールキット](#)もインストールすることを強く推奨します。

Linux*

oneAPI 2024.0

oneAPI 2024.0 でサポートされている Linux* のディストリビューション

- Red Hat* Enterprise Linux* 8.x、9.x
- Ubuntu* 20.04、22.04
- Fedora* 37、38
- SUSE* Linux* Enterprise Server 15 SP3、SP4、SP5
- Debian* 11
- Amazon* Linux* 2022
- Rocky Linux* 9
- WSL 2
- **インテルのアクセラレーター(GPU)を使用した OpenMP* TARGET (ifx のみ)**
 - インテル® データセンター GPU マックス・シリーズ(開発コード名 Ponte Vecchio、略称 PVC)
 - Red Hat* Enterprise Linux* 9.2
 - SUSE* Linux* Enterprise Server 15 SP5
 - Ubuntu* 22.04
 - インテル® データセンター GPU フレックス・シリーズ(開発コード名 Arctic Sound-M、略称 ATS-M)
 - Red Hat* Enterprise Linux* 8.6、9.2
 - Ubuntu* 22.04
 - インテル® Arc™ グラフィックス(dGPU)
 - Ubuntu* 22.04
 - インテル® Iris® Xe グラフィックス(iGfX)
 - Ubuntu* 22.04

- レベルゼロおよび OpenCL* グラフィックス・ドライバーをインストールする必要があります。
 - Linux* 汎用インテル® GPU (GPGPU) ドライバーをインストールするには、[こちらの記事](#) (英語) のインストール手順に従ってください。
- gcc、g++ および関連ツールを含む Linux* デベロッパー・ツール・コンポーネントがインストールされていること。
- インテル® コンパイラーは、さまざまなバージョンの gcc を使用して、さまざまな Linux* ディストリビューションでテストされています。使用する glibc のバージョンは、使用している gcc のバージョンと一致している必要があります。その他のディストリビューションは、動作する場合としない場合があります、推奨されません。質問がある場合、商用サポートを利用可能な場合は、[サポートチケット](#)を作成してください。利用できない場合は、[インテル・コミュニティ・フォーラム](#) (英語) でサポートを受けることができます。

既知の問題

- 環境起動スクリプトは POSIX* 準拠のスクリプトに移行されました。csh などの非 POSIX* シェルでは、次のコマンドラインを使用した回避方法があります。
コンポーネント・ディレクトリー・レイアウトの場合：


```
% bash -c 'source /opt/intel/oneapi/setvars.sh ; exec csh'
```

 統合ディレクトリー・レイアウトの場合：


```
% bash -c 'source <install-dir>/<toolkit-version>/oneapi-vars.sh ; exec csh'
```

Windows*

oneAPI 2024.0

oneAPI 2024.0 でサポートされている Windows* のバージョン

- Microsoft* Windows* 10 (Pro および Enterprise)
- Microsoft* Windows* 11 (Pro および Enterprise)
- Microsoft* Windows Server* 2019
- Microsoft* Windows Server* 2022
- **インテルのアクセラレーター (GPU) を使用した OpenMP* TARGET (ifx のみ)**
 - インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)
 - Microsoft* Windows* 10 (Pro および Enterprise) 64 ビット版
 - Microsoft* Windows* 11 (Pro および Enterprise) 64 ビット版
 - Microsoft* Windows Server* 2019
 - Microsoft* Windows Server* 2022
 - インテル® Arc™ グラフィックス (dGPU)
 - Microsoft* Windows* 10 (Pro および Enterprise) 64 ビット版
 - Microsoft* Windows* 11 (Pro および Enterprise) 64 ビット版

- インテル® Iris® Xe グラフィックス (iGFX)
 - Microsoft* Windows* 10 (Pro および Enterprise) 64 ビット版
 - Microsoft* Windows* 11 (Pro および Enterprise) 64 ビット版
- レベルゼロおよび OpenCL* グラフィックス・ドライバーが必要です。
 - ドライバーをインストールするには、次の手順に従ってください。
 - インテル® Iris® Xe MAX グラフィックス (DG1) および第 11 世代から第 13 世代インテル® Core™ プロセッサ・グラフィックスの[手順](#) (英語)
 - インテル® データセンター GPU フレックス・シリーズ (開発コード名 Arctic Sound-M、略称 ATS-M)。インテル® レジストレーション・センターのアクセス方法は、インテル製品の担当者までお問い合わせください。

Windows* でインテル® Fortran コンパイラーを使用するには、Microsoft* Visual Studio* をインストールする必要があります。[詳細はこちら](#)。

既知の問題

- Windows* パフォーマンス・ツールキット (Win11SDK_WindowsPerformanceToolkit) がインストールされているオフラインシステム上の Microsoft* Visual Studio* 2022 (17.7 以降) にインテル® ソフトウェア開発ツール (インテル® oneAPI ベース・ツールキット、インテル® HPC ツールキット、またはそれらのコンポーネント) を統合する際に既知の問題があり、正しく統合されません。この問題を回避するには、インテル® ソフトウェア開発ツールのインストール中にインターネット接続を有効にするか、インテル® ソフトウェア開発ツールをインストールする前に Windows* パフォーマンス・ツールキットをアンインストールし、インテル® ソフトウェア開発ツールをインストールした後に Windows* パフォーマンス・ツールキットを再インストールしてください。
- インテル® oneAPI ツールキット 2022.1.3 以前およびインテル® Parallel Studio XE (すべてのバージョン) は Microsoft* Visual Studio* 2022 をサポートしていません。Microsoft* Visual Studio* 2022 がインストールされているシステムでインテル® oneAPI ツールキット 2022.1.3 以前およびインテル® Parallel Studio XE のインストーラーを実行すると、インストール、アップグレード、変更、アンインストールに失敗します。詳細は、[こちらの記事](#) (英語) を参照してください。

macOS*

注: インテル® HPC ツールキット (macOS* x86 版) は 2024.0 リリースから含まれなくなりました。インテル® Fortran コンパイラー・クラシックの macOS* 版も含まれなくなりました。いくつかのインテル主導のオープンソース開発者ツール・プロジェクトは、インテル® oneAPI スレッディング・ビルディング・ブロック (インテル® oneTBB) およびインテル® インプリシット SPMD プログラム・コンパイラー (インテル® ISPC) を含む Apple シリコン上での macOS* のサポートを継続します。インテルは、将来的に追加ツールのサポートを拡大するために貢献者と協力する機会を歓迎します。

サポート終了予定

インテグレートッド・グラフィックス・プロセッサ

第 6 世代から第 10 世代インテル® Core™ プロセッサ、および同世代の Intel Atom® プロセッサ、インテル® Pentium® プロセッサ、インテル® Celeron® プロセッサに内蔵されているグラフィックス・プロセッサ向け Windows* ドライバーのサポートはメンテナンス・モードに移行しました。セキュリティ問題および重大な問題の修正のみ行われます。

前述のプロセッサの既存の統合グラフィックス・プロセッサ機能を使用するインテル® ソフトウェア開発ツールは引き続き動作する可能性はありますが、サポートされなくなります。これらのプロセッサの CPU 機能は引き続きサポートされます。

オペレーティング・システム

2024.0 リリースでは、次の Linux* ディストリビューションが非推奨になりました。2024 年秋の 2025.0 リリースでサポートを終了する予定です。

CPU

- SUSE* Linux* Enterprise Server (SLES*) 15 SP3
- Ubuntu* 20.04
- Fedora* 37
- Debian* 11
- Amazon* Linux* 2022

インテルの GPU

- Red Hat* Enterprise Linux* 8.6
- SUSE* Linux* Enterprise Server (SLES*) 15 SP3

サポート終了

これらのオペレーティング・システムはインテル® oneAPI 2024.0 でサポートされなくなりました。

- Rocky Linux* 8
- Debian* 9 および 10
- Amazon* Linux* 2
- macOS* 12

インテル® ソフトウェア開発ツールは、第 11 世代以降のインテル® Core™ プロセッサに内蔵されているグラフィックス・プロセッサをサポートします。第 6 世代から第 10 世代インテル® Core™ プロセッサ、および同世代の Intel Atom® プロセッサ、インテル® Pentium® プロセッサ、インテル® Celeron® プロセッサに内蔵されているグラフィックス・プロセッサのサポートは 2023.1 リリースで終了しました。

- これらのプロセッサの CPU 機能は引き続きサポートされます。

macOS* のサポート

macOS* のサポートは、インテル® Fortran コンパイラー・クラシック 2024.0 で削除されました。

インテル® oneAPI ベース・ツールキット(macOS* x86 版)は 2024.0 リリースから含まれなくなりました。いくつかのインテル主導のオープンソース開発者ツール・プロジェクトは、インテル® oneAPI スレッディング・ビルディング・ブロック(インテル® oneTBB)およびインテル® インプリシット SPMD プログラム・コンパイラー(インテル® ISPC)を含む Apple シリコン上での macOS* のサポートを継続します。インテルは、将来的に追加ツールのサポートを拡大するために貢献者と協力する機会を歓迎します。

インテル® HPC ツールキット(macOS* x86 版)は 2024.0 リリースから含まれなくなりました。いくつかのインテル主導のオープンソース開発者ツール・プロジェクトは、インテル® oneAPI スレッディング・ビルディング・ブロック(インテル® oneTBB)およびインテル® インプリシット SPMD プログラム・コンパイラー(インテル® ISPC)を含む Apple シリコン上での macOS* のサポートを継続します。インテルは、将来的に追加ツールのサポートを拡大するために貢献者と協力する機会を歓迎します。

開発環境

導入ガイド

- [インテル® HPC ツールキット\(Linux* 版\)導入ガイド\(英語\)](#)
- [インテル® HPC ツールキット\(Windows* 版\)導入ガイド\(英語\)](#)
- [インテル® oneAPI ベース & HPC ツールキット\(macOS* 版\)導入ガイド\(英語\)](#)

インテル® oneAPI ツールキット向け診断ユーティリティ

トップレベルの `sys_check` スクリプトが削除され、より高度なインテル® oneAPI ツールキット向け診断ユーティリティを使用してシステムチェックを実行できるようになりました。詳細は、「[インテル® oneAPI ツールキット向け診断ユーティリティ・ユーザーガイド](#)」(英語)を参照してください。

- 既知の問題と制限事項
 - 診断ユーティリティは、`.sh` および `.bat` 形式でのチェッカーの実行のみサポートしています。
 - インテル® ソフトウェア開発ツールの検出チェックは、インテル® ソフトウェア開発ツールのインストーラーのキャッシュが削除されている場合、製品を見つけることができません。

インテル® oneAPI ツールキット向け Visual Studio* Code(VS Code)拡張

[インテル® oneAPI ツールキット向け VS Code 拡張\(英語\)](#)は、oneAPI アプリケーションを作成、デバッグ、およびプロファイルする開発者を支援します。詳細は、「[Visual Studio* Code とインテル® oneAPI ツールキットの使用ユーザーガイド](#)」(英語)を参照してください。

[VS Code Marketplace\(英語\)](#)から以下の VS Code 拡張を利用できます。

これらの VS Code 拡張は、CPU コマンドと GPU コマンドに利用できます。

- インテル® oneAPI ツールキット向けサンプルブラウザー
- インテル® oneAPI ツールキット向け環境コンフィグレーター
- インテル® oneAPI ツールキット向け解析コンフィグレーター
- インテル® oneAPI ツールキット向け GDB GPU サポート
- インテル® oneAPI ツールキット向けインテル® DevCloud コネクター

Microsoft* Visual Studio*(Windows*)

- Microsoft* Visual Studio* 開発環境またはコマンドライン・ツールを使用して IA-32 またはインテル® 64 アーキテクチャー・アプリケーションをビルドするには、次のいずれかが必要です。
 - Microsoft* Visual Studio* 2022 の Community、Enterprise、および Professional エディション(「C++ によるデスクトップ開発」コンポーネントがインストールされていること)。インテル® Inspector とインテル® Advisor を除きます。
 - Microsoft* Visual Studio* 2019 の Community、Enterprise、および Professional エディション(「C++ によるデスクトップ開発」コンポーネントがインストールされていること)。
 - 詳細は、[インテル® コンパイラーと Microsoft* Visual Studio* および Xcode* の互換性](#)(英語)を参照してください。
 - [インストールと統合に関するヒント](#)(英語)
- コマンドライン・ツールのみを使用してインテル® 64 アーキテクチャー・アプリケーションをビルドする場合は次のいずれかが必要です。
 - Visual Studio* 2022 Build Tools
 - Visual Studio* 2019 Build Tools

既知の問題

Microsoft* Visual Studio* 2022 バージョン 17.2 をインストールすると、インテル® Fortran コンパイラーと Visual Studio* の統合が動作しなくなります。最新の情報は、[こちらの記事](#)(英語)を参照してください。

サポート終了予定

Microsoft* Visual Studio* 2017 とインテル® コンパイラーの統合は古い機能(非推奨)で、将来のリリースで削除される予定です。Visual Studio* との統合のサポートは、Visual Studio* の最新のメジャーバージョンおよび 1 つ前のメジャーバージョンでのみ利用できます。

Microsoft* Windows* Subsystem for Linux*

Windows* 10 と Windows* 11 で Microsoft* Windows* Subsystem for Linux* 2 (WSL2) を使用すると、CPU および GPU ワークフロー向けにインテル® ソフトウェア開発ツールとライブラリーのネイティブ Linux* ディストリビューションを Windows* 上にインストールできます。[詳細はこちら](#)(英語)。

インテル® Fortran コンパイラーの動作環境 (以前のインテル® oneAPI リリース)

- [インテル® oneAPI 2023](#)(英語)
- [インテル® oneAPI 2022](#)(英語)
- [インテル® oneAPI 2021](#)(英語)

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。

© Intel Corporation. Intel、インテル、Intel ロゴ、その他のインテルの名称やロゴは、Intel Corporation またはその子会社の商標です。

* 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。

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。

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。

インテルは、明示されているか否かにかかわらず、いかなる保証もいたしません。ここにいう保証には、商品適格性、特定目的への適合性、および非侵害性の黙示の保証、ならびに履行の過程、取引の過程、または取引での使用から生じるあらゆる保証を含みますが、これらに限定されるわけではありません。

製品および性能に関する情報

¹ 性能は、使用状況、構成、その他の要因によって異なります。詳細については、<http://www.intel.com/PerformanceIndex/> (英語) を参照してください。