

# インテル® Fortran コンパイラー・リリース ノート(インテル® oneAPI 2022.1 リリース)

本書は、英文「[Intel® Fortran Compiler for oneAPI Release Notes, oneAPI 2022.1 Release](#)」(英語)の日本語参考訳です。

---

2021年12月6日

## はじめに

このドキュメントでは、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載されていない既知の問題について説明します。

インテル® oneAPI 製品パッケージは 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・クラシック(ifort)は、CPU 向けにクラス最高の Fortran 言語機能とパフォーマンスを提供します。インテル® Fortran コンパイラー(ifx)は、開発者にインテル® GPU への OpenMP\* オフロード機能を提供します。ifx の OpenMP\* 5.0/5.1 GPU オフロード機能は ifort では利用できません。ifort は、2022 年も引き続き GPU オフロードのサポートを必要としないユーザーにとってクラス最高の Fortran コンパイラーとして提供されます。Microsoft\* Visual Studio\* 環境向けのデフォルトのコンパイラーは ifort です。

インテル® Fortran コンパイラー(ifx)が製品版となり、CPU および GPU に対応しました。ifx は、インテル® Fortran コンパイラー・クラシック(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、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。この初期リリースで、ifx は、Fortran 77、Fortran90/95、Fortran 2003(パラメーター化された派生型を除く)、Fortran 2008(Co-Array を除く)言語標準、OpenMP\* 4.5 および OpenMP\* 5.0/5.1 のディレクティブとオフロード機能を実装しました。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)。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。ifx でコンパイルしたアプリケーションのパフォーマンスは、ifort でコンパイルしたアプリケーションのパフォーマンスに匹敵する場合も匹敵しない場合もあります。ifx のパフォーマンスと Fortran 標準言語の改善は、2022 年のアップデートリリースで行われる予定です。

このリリースの ifx のバージョン番号は 2022.0.0、ifort のバージョン番号は 2021.5.0 です。

## リリースの入手方法

[このサイト](#)(英語)から手順に従ってインテル® oneAPI 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ツールキットをインストールします。すべての機能を使用するには、最初にインテル® oneAPI ベース・ツールキットをインストールしてください。

## 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。

# インテル® Fortran コンパイラー・クラシックおよびインテル® Fortran コンパイラーの新機能(インテル® oneAPI 2022.1 リリース)

## ifort バージョン 2021.5.0

- Fortran 言語の修正およびセキュリティー・アップデート

## ifx バージョン 2022.0.0

- パラメーター化された派生型を除くすべての Fortran 2003 標準を実装
- Co-Array を除くすべての Fortran 2008 標準を実装
- 報告された問題を修正
- Fortran 2003、Fortran 2008、Fortran 2018、OpenMP\* 4.5 および OpenMP\* 5.0/5.1 機能の実装の詳細は、[この記事](#) (英語) を参照してください。

## 動作環境

[インテル® Fortran コンパイラーの動作環境](#) (英語) を参照してください。

Linux\* では、CMake\* 3.20 で、インテル® oneAPI HPC ツールキット 2021.2 以降のインテル® oneAPI の LLVM コンパイラー (ifx および icx) は、コンパイラー ID `IntelLLVM` でサポートされます。Windows\* では、CMake\* でまだ ifx はサポートされていません。インテル® oneAPI のクラシック・コンパイラー (icc、icpc、および ifort) は、コンパイラー ID `Intel` で引き続きサポートされます。

## 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 ツールキットの導入ガイドは、オンラインでも利用できます。詳細は、「[インテル® oneAPI ツールキット導入ガイド](#)」(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® oneAPI ツールキット \(Linux\\* 版\) 導入ガイド](#) (英語)
- [インテル® oneAPI 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 2003、Fortran 2008、OpenMP\* 4.5 および OpenMP\* 5.0/5.1 機能の実装の詳細は、[この記事](#) (英語) 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

サポートが必要な場合は、[インテル® Fortran コンパイラー・フォーラム](#) (英語) にアクセスしてください。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](#) (英語) してください。

## 新規および変更されたコンパイラー機能

### ifort

#### OpenMP\*

- OpenMP\* 5.0 標準に準拠するように、スケジュール種別が `dynamic` または `guided` の場合のワークシェアリング・ループ構文の OpenMP\* デフォルト・ループ・スケジュール修飾子を `nonmonotonic` に変更しました。`monotonic` 動作を想定しているユーザーコードは、この変更により正しく動作しなくなる可能性があります。以前のコードの動作を維持するには、`schedule` 節に `monotonic` スケジュール修飾子を追加してください。

### ifx

#### 言語と OpenMP\*

#### Fortran 2003

- パラメーター化された派生型を除いて実装を完了しました。

#### Fortran 2008

- Co-Array を除いて実装を完了しました。

#### OpenMP\*

- OpenMP\* 5.0 標準に準拠するように、スケジュール種別が `dynamic` または `guided` の場合のワークシェアリング・ループ構文の OpenMP\* デフォルト・ループ・スケジュール修飾子を `nonmonotonic` に変更しました。`monotonic` 動作を想定しているユーザーコードは、この変更により正しく動作しなくなる可能性があります。以前のコードの動作を維持するには、`schedule` 節に `monotonic` スケジュール修飾子を追加してください。
- OpenMP\* 5.0/5.1 の多くの機能が実装されました。詳細は、『[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) および [この記事](#) (英語) を参照してください。

#### 参考情報

Fortran 2003、Fortran 2008、Fortran 2018、OpenMP\* 4.5 および OpenMP\* 5.0/5.1 機能の実装の詳細は、[この記事](#) (英語) を参照してください。

#### 既知の問題

- 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。

## 終了予定のサポート

### Microsoft\* Visual Studio\* 2017

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。

### 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。

### ifx

- -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 -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。

## 既知の問題

### 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。

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

### 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。

### Fortran ランタイム・ライブラリー

Windows\* では、インテル® oneAPI Fortran ランタイム・ライブラリーは新しいフォルダーにインストールされます。新しいフォルダーは %INTEL\_DEV\_REDIST%redist\intel64\_win\compiler です。インテル®

Parallel Studio XE の Fortran 再配布可能ファイルとインテル® oneAPI の Fortran ランタイム・ライブラリーをともにインストールすると、PATH 環境変数が正しく設定されません。

### Windows\* で致命的なエラー LNK1318: Unexpected PDB error; OK (0) が発生する

Windows\* で、高い最適化レベル(例えば、/O2)で /debug オプションを追加して実行ファイルをビルドすると、リンク時に致命的なエラーが表示されることがあります。

```
fatal error LNK1318: Unexpected PDB error; OK (0)
```

これは、Microsoft\* Visual Studio\* 2019 バージョン 16.11.2 に含まれているリンカーのバグが原因です。この問題を回避するには、'/inline:none' コマンドライン・オプションを使用して関数のインライン展開を無効にします。このオプションを使用すると、実行ファイルのパフォーマンスが低下する可能性があることに注意してください。

### Ubuntu\* 18.04.3 でコンパイラーを実行するとエラーになる

```
ifort: error #10105: ld: core dumped
```

Ubuntu\* 18.04 LTS の最新ビルドは 18.04.4 です。以前のビルド 18.04.3 には、インテル® Fortran コンパイラー・クラシックを実行するとエラーになる問題があります。18.04.2 および 18.04.1 などの以前のビルドでは、この問題は発生しません。

次のコマンドを実行して、ビルド番号を確認できます。

```
cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.4 LTS"
```

DISTRIB\_DESCRIPTION に注意してください。このシステムのビルドは、最新の 18.04.4 LTS です。

DISTRIB\_DESCRIPTION が 18.04.3 と表示された場合、ビルド 18.04.3 に含まれる glibc の問題により、インテル® コンパイラーを実行するとエラーになります。

次のエラーが表示されます。

```
ifort hello.f90
ifort: error #10105: ld: core dumped
ifort: warning #10102: unknown signal(-320620368)
ifort: error #10106: Fatal error in ld, terminated by unknown
```

最新のビルド 18.04.4 にアップグレードすると、修正された glibc を含む重要なアップデートがインストールされ、この問題は発生しなくなります。

```
sudo apt-get update
sudo apt update
```

### macOS\* 11 での IPO

macOS\* 11 で、インテル® oneAPI HPC ツールキットで利用可能なインテル® oneAPI Fortran コンパイラー・クラシック(ifort)の IPO(-ipo)を使用すると、リンク時にエラーが発生します。macOS\* 11 ではコンパイラー・オプションから -ipo を削除してください。

このリンクエラーは Linux\* または Windows\* では発生しません。また、macOS\* 10.x でも発生しません。macOS\* 11 Big Sur でのみ発生します。

macOS\* 11 Big Sur では IPO は動作しません。将来のリリースで削除される予定です。

## Fortran Co-Array アプリケーションが FAIL IMAGE の実行後にハングする

FAIL IMAGE 文を使用してイメージを失敗させる場合は、文または操作で許可している場合にイメージが失敗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・オプションを指定します。STAT= 指定子や STAT 引数を使用しない場合、またはいずれかのコンパイラー・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ません。イメージからの応答を待って失敗したイメージを調整しようとした場合、応答がないために、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ています。

```
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)      ! Would hang without STAT=
END SUBROUTINE FAIL_AND_SYNC
```

## クロスコンパイル(Microsoft\* Windows\* を実行している 64 ビット・マシンでの 32 ビット・モードでのコンパイル)

- Microsoft\* Windows\* で実行している
- クロスコンパイル(64 ビット・マシンでの 32 ビット・モードでのコンパイル)を行っている
- Microsoft\* Visual Studio\* 2019 を使用している
- コードで例外、特に浮動小数点例外を処理している
- プログラムで浮動小数点例外を取得すると、SEGV 例外が発生する

上記に該当する場合、プログラムは意図したとおりに例外を処理していません。

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。

## Red Hat\* 8 での Co-Array 機能の使用

Co-Array 機能を使用するアプリケーションを Red Hat\* 8.0 以降で実行するには、インテル® MPI ライブラリー 2019 Update 7 以降がインストールされている必要があります。2019 Update 7 より前のバージョンがインストールされている場合、アプリケーションは正常に完了せず、障害に関する情報はほとんど提供されません。

## 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。

## Fortran ランタイム・ライブラリー

Windows\* では、インテル® oneAPI Fortran ランタイム・ライブラリーは新しいフォルダーにインストールされます。新しいフォルダーは %INTEL\_DEV\_REDIST%redist\intel64\_win\compiler です。インテル® Parallel Studio XE の Fortran 再配布可能ファイルとインテル® oneAPI の Fortran ランタイム・ライブラリーをともにインストールすると、PATH 環境変数が正しく設定されません。

## Windows\* で致命的なエラー LNK1318: Unexpected PDB error; OK (0) が発生する

Windows\* で、高い最適化レベル(例えば、/O2)で /debug オプションを追加して実行ファイルをビルドすると、リンク時に致命的なエラーが表示されることがあります。

```
fatal error LNK1318: Unexpected PDB error; OK (0)
```

これは、Microsoft\* Visual Studio\* 2019 バージョン 16.11.2 に含まれているリンカーのバグが原因です。この問題を回避するには、'/inline:none' コマンドライン・オプションを使用して関数のインライン展開を無効にします。このオプションを使用すると、実行ファイルのパフォーマンスが低下する可能性があることに注意してください。

## 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 GPU 計算ワークロードを含むアプリケーションがある場合、ワークロードが終了するのを防ぐためハングチェック・タイムアウト期間を無効にする必要があります。詳細は、使用している OS の [インストール・ガイド](#) (英語) を参照してください。

## 言語と OpenMP\*

### その他の既知の問題

- !DIR\$ ディレクティブはフロントエンドで認識されますが、すべては実装されません。ifx がサポートする !DIR\$ ディレクティブは、IVDEP、VECTOR [NO]DYNAMIC\_ALIGN、VECTOR [NO]REMAINDER、DISTRIBUTE\_POINT、NOFUSION、[NO]UNROLL、[NO]UNROLL\_AND\_JAM です。
- 全体的なデバッグのパフォーマンスは、形状引き継ぎ配列を除いて大幅に向上しました。これは、ホストとターゲットの両方に当てはまります。

## スタティック・ライブラリーとターゲットオフロードの問題

バグにより、Linux\* 上の ifx コンパイラーは、ターゲット・オフロード・コード (GPU または FPGA 向けのオフロードコードなど) を含むライブラリーで -l オプションを使用したライブラリー・アーカイブのリンクをサポートしていません。問題と回避策の説明は、[この記事](#) (英語) を参照してください。この回避策は、インテルのほかの LLVM ベースのコンパイラー (icx、icpx および dpcpp) にも適用できます。

## 設定スクリプトのメッセージ: 「C から Fortran ライブラリーへのリンクに失敗」

### 症状

ifx を使用してコンパイルすると、GNU\* Autconf で生成された ./configure スクリプトで次のようなエラーメッセージが表示される。

```
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、次のエラーにより ./configure が終了していました。

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

## 問題

Fortran コードと C/C++ コードのリンクに必要なライブラリーを決定しようとする、GNU\* Autoconf 2.69 以前は、`-mllvm -loopopt=0` 詳細コンパイラー出力オプションの `-loopopt=0` をリンカーオプションとして誤って解釈し、リンカーに渡される FLIBS 変数に `-loopopt=0` を追加します。リンカーは存在しないライブラリーを探し、テストは失敗します。

## ソースコードをダウンロードす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問題を通知し、このリリースノートの情報を伝えてください。プロジェクトは `configure` スクリプトの処理方法が異なりますが、いくつかの一般的なルールが適用されます。パッケージメンテナーから連絡が届く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することも可能です。更新の難しさは、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、まず、下記の[パッケージメンテナー向けソリューション](#)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、次のコマンドを実行して、`configure.ac` ファイルと `configure` スクリプト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 で上記のエラーなしで完了するはずです。

プロジェクトのメンテナーが、tar または zip アーカイブのソース・ディストリビューションから `configure.ac` ファイルを削除するのは珍しいことではありません。その場合、ユーザーは通常、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ことで、ほとんどの場合は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` スクリプトを再構築する方法が記載されている場合は、その方法に従ってください。

## パッケージメンテナー向けソリューション

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、プロジェクト `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・リリースノート (以前のインテル® oneAPI リリース)

- [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ー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。

---

### 製品とパフォーマンス情報

<sup>1</sup> 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。