

# インテル® Fortran コンパイラー・リリース ノート (インテル® oneAPI 2023 リリース)

本書は、英文「[Intel® Fortran Compiler Release Notes and New Features](#)」(英語)の日本語参考訳です。

バージョン: 2023.0  
2022 年 12 月 12 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述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、インテル® アーキテクチャーの CPU 向けに、Fortran 2018 を含む、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、ifort のすべての Fortran 2018 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 OpenMP\* 5.0/5.1 GPU オフロード機能は ifort では利用できません。

最新の Fortran コンパイラーであるインテル® Fortran コンパイラー (ifx) に、ifort と同等の言語機能 (Fortran 2018) が含まれました。ifx は、CPU とインテルの GPU に対応しています。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。このリリースで、ifx は次の更新を行いました。

- FORTRAN 77、Fortran90/95、Fortran 2003、Fortran 2008 および Fortran 2018 言語標準を完全に実装
- ほとんどの OpenMP\* 4.5、OpenMP\* 5.0/5.1 ディレクティブおよびオフロード機能をサポート
- OpenMP\* 6.0 の一部の機能を追加

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)。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 のパフォーマンスの改善は、2023 年の各リリースで引き続き行われる予定です。

## 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。[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。

### 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 2023.0 リリース)

#### ifx バージョン 2023.0.0

- 最新の LLVM テクノロジーに基づくインテル® Fortran コンパイラーは、Co-Array を含む Fortran 2018 までの言語標準を完全にサポートしました。MPI や OpenMP\* などの外部 API は不要になります。
- オフロード並列処理を簡単に表現できるように、DO CONCURRENT GPU オフロードをサポートしました。
- 第 4 世代インテル® Xeon® スケーラブル・プロセッサー (開発コード名 Sapphire Rapids) およびインテル® データセンター GPU マックス・シリーズ (開発コード名 Ponte Vecchio) のサポートが追加されました。
- OpenMP\* 5.0、5.1 への準拠が強化され、オフロードとパフォーマンスが向上しました。
- 開発者は、ソースレベルで最適化されたコードをデバッグできるようになりました。
- インテル® Fortran コンパイラー・クラシックと同等の機能が含まれました。
- 報告された問題を修正
- Fortran 2003、Fortran 2008、Fortran 2018、OpenMP\* 4.5、OpenMP\* 5.0/5.1、および一部の OpenMP\* 6.0 機能の実装の詳細は、[この記事](#)を参照してください。

#### ifort バージョン 2021.8.0

- 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。
- 報告された問題を修正

## 動作環境

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

## 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 \(英語\)](#)
- Fortran 2003、Fortran 2008、Fortran 2018、OpenMP\* 4.5、OpenMP\* 5.0/5.1、および一部の OpenMP\* 6.0 機能の実装の詳細は、[この記事](#)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。

## 新しいおよび変更されたコンパイラーの機能

### ifort

#### ifort の新しいコンパイラー・オプション

- 新しいオプション `qmk1-ilp64`、`Qmk1-ilp64`
  - インテル® oneAPI マス・カーネル・ライブラリー (インテル® oneMKL) の ILP64 固有のバージョンにリンクするようにコンパイラーに指示します。
- 新しい `assume` オプションの設定、`[no]old_e0g0_format`
  - G0.0、E0、または ESO 形式編集記述子を持つ浮動小数点のフォーマットするのに Fortran 2018 の規則を使用するかどうかを指定します。
- 新しい `fuse-ld` オプションの設定、`lld`
  - `lld` リンカーを使用するようにコンパイラーに指示します。

### ifx

#### 言語および OpenMP\*

#### Fortran 2018

- 実装完了

## OpenMP\*

- 次の OpenMP\* 5.1 の機能が ifx で利用できるようになりました。
  - DEPOBJ
  - DISPATCH の INTEROP 節
  - SCAN ディレクティブ
  - DEPEND タイプ mutexinoutset, inoutset, および depobj
  - 可変長配列の DECLARE MAPPER
  - DECLARE VARIANT の APPEND\_ARGS 節の PREFER\_TYPE
  - SCOPE 構文
- !\$OMP PREFETCH DATA
  - メモリー・レイテンシーの影響を最小限に抑えるため、データをキャッシュに事前にロードするようにコンパイラーに指示します。
- ほとんどの OpenMP\* 5.0/5.1 の機能と一部の OpenMP\* 6.0 の機能が実装されました。詳細は、『[Intel® Fortran コンパイラー・クラシックおよび Intel® 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](#)を参照してください。

## ifx の新しいコンパイラー・オプション

- ifort の次のコンパイラー・オプションが ifx で利用できるようになりました。
  - -align
  - -falign-functions
  - -check:pointers
- -fprofile や -flto など、いくつかの Clang コンパイラー・オプションが ifx コンパイラーでサポートされました。これらのオプションについては、このドキュメントで説明していません。Clang オプションの詳細は、[Clang のドキュメント](#)（英語）を参照してください。
  - Clang の -fprofile オプションは ifort でのみサポートされている [Q]prof オプションの機能を置き換えることに注意してください。
  - -flto は ifort でサポートされている [Q]ipo オプションの機能を置き換えます。このオプションは、プログラム全体でリンク時最適化 (LTO) を有効にします。
- 新しいオプション qmkl-ilp64、Qmkl-ilp64
  - Intel® oneAPI マス・カーネル・ライブラリー (Intel® oneMKL) の ILP64 固有のバージョンにリンクするようにコンパイラーに指示します。
- 新しい assume オプションの設定、[no]old\_e0g0\_format
  - G0.0、E0、または ESO 形式編集記述子を持つ浮動小数点のフォーマットするのに Fortran 2018 の規則を使用するかどうかを指定します。
- 新しい fuse-ld オプションの設定、lld
  - lld リンカーを使用するようにコンパイラーに指示します。

## 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](#)を参照してください。

## 終了予定のサポート

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。

### 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) は古い機能 (非推奨) で、2023 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® 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。

### 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 `-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。

## 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'
```

## 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。

### 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。

### 機能およびセキュリティ・アップデート

インテル® Fortran コンパイラー・クラシック 2021.8.0 には、最新の機能とセキュリティ・アップデートのすべては含まれていません。2023 年 3 月にリリース予定のインテル® Fortran コンパイラーの新しいバージョンには、追加の機能とセキュリティ・アップデートが含まれる予定です。最新バージョンが入手可能になり次第、アップデートしてください。

## 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。

### 派生型のマッピング

ifx は、割り当て可能なコンポーネントを含む派生型のマッピングをサポートしていません。2023.0 以前は、割り当て可能な派生型コンポーネントに暗黙的にアクセスすると、予期しない結果を引き起こしていました。2023.0 では、ifx がこの機能をサポートしていないことを示す警告が出力されます。

### 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 構造と同じ REDUCTION 節がなく、`num_teams` 値が 1024 を超えている場合 (環境変数で明示的に設定した場合、または `thread_limit` が低くカーネルが値を超える場合)、OpenMP\* ディレクティブ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。

### 機能およびセキュリティ・アップデート

インテル® Fortran コンパイラー 2023.0.0 には、最新の機能とセキュリティ・アップデートのすべては含まれていません。2023 年 3 月にリリース予定のインテル® Fortran コンパイラーの新しいバージョンには、追加の機能とセキュリティ・アップデートが含まれる予定です。最新バージョンが入手可能になり次第、アップデートしてください。

### -ax の使用

`-fpic` オプションを指定してライブラリーをビルドする場合、`-ax` (自動 cpu ディスパッチ) オプションは現時点では使用できません。

### -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。

## 言語および OpenMP\*

### その他の既知の問題

- ifx は、OpenMP\* の COPYPRIVATE 節 (割り当て可能またはポインター配列) をサポートしていません [2023.0.0 で修正されました]。
- !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]VECREMAINDER、DISTRIBUTE POINT、NOFUSION、[NO]UNROLL、[NO]UNROLL\_AND\_JAM、VECTOR ALWAYS ASSERT 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。

### 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・リリースノート (以前のインテル® oneAPI リリース)

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

# インテル® Fortran コンパイラーの動作環境 (インテル® oneAPI 2023.0 リリース)

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

バージョン: 2022.3  
2022 年 12 月 12 日

## はじめに

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。

インテル® oneAPI 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® oneAPI 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。

## ハードウェア要件

- プロセッサー (CPU)
  - インテル® Core™ プロセッサー・ファミリー
  - インテル® Xeon® プロセッサー・ファミリー
  - インテル® Xeon® スケーラブル・プロセッサー・ファミリー
- アクセラレーター (ifx のみ)
  - 第 9 世代インテル® インテグレートド・グラフィックス以降
  - インテル® データセンター GPU マックス・シリーズ (PVC)
  - インテル® データセンター GPU フレックス・シリーズ (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® oneAPI 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。

これらの OS ディストリビューションはインテルによってテストされたもの、または動作が確認されているものです。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、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・コミュニティー・フォーラム](#) (英語) でサポートを受けることができます。

## Linux\*

- Red Hat\* Enterprise Linux\* 8.x、9.x
- Ubuntu\* 20.04 LTS、22.04
- Fedora\* 36、37
- SUSE\* Linux\* Enterprise Server 15 SP3、SP4
- Debian\* 9.0、10.0、11
- Amazon\* Linux\* 2、2022
- Rocky Linux\* 8、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 (PVC)
    - Red Hat\* Enterprise Linux\* 8.6
    - SUSE\* Linux\* Enterprise Server 15 SP4
    - Ubuntu\* 22.04
  - インテル® データセンター GPU フレックス・シリーズ (ATS-M)
    - Red Hat\* Enterprise Linux\* 8.6
    - 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 (iGPU)
    - SUSE\* Linux\* Enterprise Server 15 SP4
    - Ubuntu\* 20.04、22.04
  - OpenCL\* およびレベル 0 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) ドライバーは、[この記事](#) (英語) の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。

## 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。

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

## Windows\*

- 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 (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 (iGPU)
    - 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
  - OpenCL\* およびレベル 0 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。
    - 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 10 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  
<https://www.intel.com/content/www/us/en/download/19344/intel-graphics-windows-dch-drivers.html> (英語)
      - インテル® Arc™ A シリーズ・グラフィックス (DG2)  
<https://www.intel.com/content/www/us/en/download/726609/intel-arc-graphics-windows-dch-driver.html> (英語)
      - インテル® データセンター GPU フレックス・シリーズ (ATS-M)  
<https://registrationcenter.intel.com/en/products/subscription/956/> (英語)  
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](#)。

### 既知の問題

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 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、「[この記事](#)」(英語)を参照してください。

## macOS\*

### サポートしている macOS\* のバージョン#

- macOS\* 12.0 Monterey
- macOS\* 13.0 Ventura

# 注: インテル® プロセッサを搭載している Mac\* コンピューターの macOS\* をサポート

### Xcode\* との統合の削除

Fortran コンパイラーの Xcode\* との統合機能は削除されました。コマンドライン・インターフェイスは、macOS\* で引き続き動作します。

コマンドライン・コンパイラーには Xcode\* のコマンドライン・ツールが必要になるため、Xcode\* のインストールは引き続き推奨されていることに注意してください。ターミナルウィンドウから次のコマンドを使用して、Xcode\* コマンドライン・ツールのみインストールできます。

```
xcode-select --install
```

### Xcode\* のコマンドライン・ツール

- Xcode\* 11 は、コマンドライン・ツールのみ使用できます。Xcode\* との統合は削除されました。
- Xcode\* 12 および Xcode\* 13 は、コマンドライン・ツールのみ使用できます。
  - Xcode\* との統合は削除されました。
  - Xcode\* 12.0、Xcode\* 12.1、または Xcode\* 12.2 とリンクした場合に、一部のライブラリーが見つかりません。

#### 回避方法

Xcode\* をインストールした後、次のコマンドを実行します。

```
sudo xcode-select -s /Applications/Xcode.app/Contents/Developer
```

次のコマンドを実行して、ライセンスに同意します (まだ同意していない場合のみ)。

```
sudo xcodebuild -license
```

### その他の macOS\* の情報

- インテル® oneAPI 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® oneAPI 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。
- macOS\* では ifx は利用できないため、OpenMP\* TARGET はサポートしていません。

### 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。zsh 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 zsh'
```

## 注

- 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。
- -O3、-ipo、-qopenmp などの高度な最適化を有効にして非常に大きなソースファイルをコンパイルすると、大量のメモリーが必要になります。
- 一部の最適化オプションは、アプリケーションを実行するプロセッサに関する制限があります。詳細は、オプションのドキュメントを参照してください。

## 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® oneAPI の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

これらのオペレーティング・システムはインテル® oneAPI 2023.0 では非推奨で、将来のリリースではサポートされなくなる予定です。

- Rocky Linux\* 8
- Debian\* 9、10
- Amazon\* Linux\* 2
- macOS\* 12

## 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 2023.0 でサポートされなくなりました。

- Windows Server\* 2016
- Red Hat\* Enterprise Linux\* 7
- SUSE\* Linux\* Enterprise Server 15 SP2
- Ubuntu\* 18.04 LTS
- CentOS\* 7
- Fedora\* 34
- Fedora\* 35
- macOS\* 11

# 開発環境

## 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せん。
  - インテル® oneAPI 製品の検出チェックは、インテル® oneAPI 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。

すべての VS Code 拡張で、CPU および GPU コマンドにツールヒントとコード補完が追加されました。

- インテル® oneAPI ツールキット向けサンプルブラウザー
  - 新機能
    - コマンドパレット形式でサンプルを閲覧できるようにしました。
    - UI/UX の改善 - 選択したサンプル用に新しいフォルダーを自動生成します。
    - UI/UX の改善 - 設定の更新がリアルタイムに反映されるようになりました。
- インテル® oneAPI ツールキット向け環境コンフィグレーター
  - 新機能
    - Windows\* では、管理者権限がなくてもインテル® oneAPI 環境を初期化できるようになりました。
- インテル® oneAPI ツールキット向け解析コンフィグレーター
  - 新機能
    - FPGA 属性にコード補完スニペットとホバーが追加されました。
    - インテル® VTune™ プロファイラーとインテル® Advisor がインストールされている場合、自動検出されるようになりました。
- インテル® oneAPI ツールキット向け GDB GPU サポート
  - 新機能
    - アプリケーションのデバッグプロセスでスレッド名にアクセスできるようになりました。

- インテル® oneAPI ツールキット向けインテル® DevCloud コネクター
  - 新機能
    - 初期環境設定プロセスが改善されました。
    - Cygwin との接続のセットアップ・プロセスが自動化されました。
    - プロキシ経由の接続で、ssh 構成ファイルのセットアップが自動化されました。
    - ssh フィンガープリント (known\_hosts ファイル) のセットアップが自動化されました。
    - エラー処理が大幅に改善されました。
    - インテル® DevCloud との接続向けに Remote-SSH ウィンドウの自動実行が追加されました。

#### 関連情報:

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

## 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)。
  - Microsoft\* Visual Studio\* 2017 Professional エディション (またはそれ以上) または Microsoft\* Visual Studio\* Community 2017。「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。
  - Microsoft\* Build Tools for Visual Studio 2022
  - Microsoft\* Build Tools for Visual Studio 2019
  - Microsoft\* Build Tools for Visual Studio 2017

#### 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® oneAPI ツールキット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](#) (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 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

<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。