

# インテル® oneAPI ベース・ツールキット 2021.1 リリースノート

---

2020 年 12 月 7 日

インテル® oneAPI ベース・ツールキットは、ダイレクト・プログラミングと API プログラミングをサポートし、インテル® プロセッサおよび互換プロセッサ、第 9 世代および第 11 世代インテル® プロセッサ・グラフィックス、インテル® Iris® Xe MAX グラフィックス、インテル® Arria® 10 FPGA およびインテル® Stratix® 10 SX FPGA を含む多様なハードウェアにわたって、ネイティブコードを完全にサポートする統一された言語とライブラリーを提供します。ダイレクト・プログラミング・モデルと API ベースのプログラミング・モデルに対応しており、開発とパフォーマンス・チューニングを支援する解析およびデバッグツールも備えています。

## サポートされる主な機能

### 2021.1 の新機能

#### ツールキット・レベルの主な機能

- Linux\*、Windows\*、および macOS\* プラットフォームをサポート。ただし、各プラットフォームで利用可能な製品は異なります。
- 「[インテル® oneAPI ツールキット・インストール・ガイド](#)」(英語) に従って、使用するオペレーティング・システム向けの最新の GPU ドライバーをインストールしてください。
- デフォルトのインストール・パスは次のとおりです。本リリースをインストールする前に、すべてのベータ版をアンインストールすることを推奨します。
  - Linux\* または macOS\*: /opt/intel/oneapi
  - Windows\*: C:\Program Files (x86)\Intel\oneAPI
- ベータ版インテル® oneAPI ベース・ツールキット、インテル® Parallel Studio XE、またはインテル® System Studio を使用するアプリケーションをインテル® oneAPI ベース・ツールキット製品リリースへ移行するには、アプリケーション全体をリビルドしてください。
- Linux\* および Windows\* では、C/C++ コンパイラー (icx) と DPC++ コンパイラー (dpcpp) を含むインテル® oneAPI DPC++/C++ コンパイラーを利用できます。
- デバイス・オフロード・コードでは、レベルゼロ (Level-0) ランタイムがデフォルトのバックエンドです。必要に応じて、[こちらの手順](#) (英語) に従って、バックエンドを OpenCL\* に変更します。すべてのライブラリー API と製品が、レベルゼロと OpenCL\* バックエンドの両方をサポートしているわけではありません。詳細は、製品レベルのリリースノートとドキュメントを参照してください。
- インテル® oneAPI ベース・ツールキットは、Linux\*、Windows\*、および macOS\* でインテル® Parallel Studio XE またはインテル® System Studio と共存できます。
- 追加の配布チャネルとしてインテル® oneAPI ツールキットでは yum および APT を、パフォーマンス・ライブラリーでは conda、PIP、NuGet をサポートしました。

## インテル® oneAPI DPC++/C++ コンパイラー

- インテル® oneAPI DPC++ コンパイラー
  - DPC++ 1.0 仕様をサポート:  
<https://spec.oneapi.com/versions/latest/elements/dpcpp/source/index.html#onedpcpp-section> (英語)。
  - Ahead-Of-Time (AOT) コンパイルをサポート。
  - 明示的な SIMD プログラミングの試験的サポート。
  - Visual Studio\* 2017 および 2019 (Windows\*)、Eclipse\* (Linux\*) との統合。
  - 複数の FPGA ターゲット・プラットフォームをサポート。
- インテル® C++ コンパイラー
  - Clang および LLVM ベースのコンパイラー (icx)。
  - OpenMP\* 4.5、オフロードを含む OpenMP\* 5.0 のサブセットをサポート。
  - ベクトル化とループの最適化。

## インテル® oneAPI DPC++ ライブラリー (インテル® oneDPL)

- 並列アルゴリズム、DPC++ 実行ポリシー、特殊イテレーター、その他のユーティリティを含む [oneDPL 1.0 仕様](#) (英語) をサポート。
- oneDPL アルゴリズムは DPC++ バッファおよび統合共有メモリー (USM) 内のデータを操作可能。
- "<array>"、"<complex>"、"<functional>"、"<tuple>"、"<utility>"、その他の標準ライブラリー API を含む標準 C++ ライブラリーのサブセットを DPC++ カーネルでサポート。
- DPC++ カーネルで使用するための標準 C++ 乱数ジェネレーターと分布。

## インテル® DPC++ 互換性ツール

- CUDA\* カーネル、ホストおよびデバイス API 呼び出し (メモリー管理、イベント、数学など)、ライブラリー呼び出し (cuBLAS、cuSPARSE、cuSolver、cuRand、NVIDIA\* Thrust) の移行をサポート。通常、ツールにより 80%-90% の CUDA\* コードを DPC++ コードへ移行できます。
- コードを手動で移行する必要がある場合に作業を支援するため、警告メッセージをコマンドラインに出力するとともに、生成されたコードにもインラインで表示。
- Visual Studio\* 2017 および 2019 (Windows\*)、Eclipse\* (Linux\*) との統合により移行の利便性を強化。

## インテル® oneAPI マス・カーネル・ライブラリー (インテル® oneMKL)

- インテル® マス・カーネル・ライブラリー (インテル® MKL) からインテル® oneAPI マス・カーネル・ライブラリー (インテル® oneMKL) に名称を変更。
- 新たに次のプログラミング・モデルをサポート: CPU およびインテル® GPU 向けのプログラミング、インテル® GPU 向けの C/Fortran OpenMP\* オフロード・インターフェイスを含むデータ並列 C++ (DPC++) API。

- インテル® プロセッサー・グラフィックスおよび X<sup>e</sup> アーキテクチャー・ベースのグラフィックスで統合共有メモリー (USM) をサポート。

### 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)

- 下位互換性に影響する変更
  - 利便性を向上し、ライブラリーを簡素化するためコードベースを改良。詳細は、「[TBB の改良: 背景、変更点、および現代化](#)」(英語) を参照してください。このバージョンは、以前のバージョンと下位互換性がありません。
- 新機能
  - 順序付きコンカレント・コンテナー、NUMA 向け `task_arena` インターフェイス拡張、機能ノードと再開可能なタスクの相対的優先順位をサポートするフローグラフ API を完全にサポート。
  - タスク領域の優先順位を指定する `task_arena` インターフェイス拡張を実装。

### GDB 向けインテル® ディストリビューション

- CPU、GPU、および FPGA エミュレート・デバイスへオフロードされたカーネルのデバッグをサポート。
- GPU デバイスへ自動的にアタッチしてデバッグイベントをリスニング。
- デバッグする JIT コンパイルされた (動的にロードされた) カーネルコードを自動検出。
- DPC++、C++ OpenMP\* オフロードデバッグ、および OpenCL\* をサポート。
- アクティブな SIMD レーンをリストして、スレッドごとに現在の SIMD レーンのコンテキストを切り替える機能を提供。

### 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 (インテル® IPP)

- CPU のみサポート。
- 第 10 世代インテル® Core™ プロセッサー・ファミリーで、インテル® IPP Cryptography の AES、RSA サポート向けの最適化を拡張。
- CRC8、CRC16、CRC24、CRC32 チェックサムを計算する新しいユニバーサル CRC 関数を追加。

### インテル® oneAPI コレクティブ・コミュニケーション・ライブラリー (インテル® oneCCL)

- ディープラーニングのトレーニングに使用される処理 (`allgather`、`allreduce`、`alltoall` (v)、`broadcast`、`reduce`、`reduce_scatter`) を効率良く実装。
- C++ API、DPC++ との相互運用性を提供。
- 以下のディープラーニングの最適化を含みます。
  - 計算と通信をオーバーラップする非同期進行。
  - ネットワークを最適に利用するための専用コア。
  - メッセージの優先順位付け、永続性、およびアウトオブ実行。

- 低精度データ型 (`int[8, 6, 32, 64]`、`fp[32, 64]`、`bf16`) の集合操作。
- Linux\* のみサポート。

### インテル® oneAPI データ・アナリティクス・ライブラリー (インテル® oneDAL)

- インテル® データ・アナリティクス・アクセラレーション・ライブラリー (インテル® DAAL) からインテル® oneAPI データ・アナリティクス・ライブラリーに名称を変更し、パッケージ名も変更。
- 32 ビット・バージョンのライブラリーの提供終了。
- OpenCL\* およびレベルゼロ・バックエンドの両方でインテル® GPU をサポート。
- CPU/GPU の両方で次のアルゴリズムを oneDAL 1.0 仕様に準拠: K 平均法、PCA、ランダムフォレスト分類と回帰、KNN、SVM。
- GPU で新しいインテル® DAAL および `daal4py` 機能をサポート
  - バッチ・アルゴリズム: K 平均法、共分散、PCA、ロジスティック回帰、線形回帰、ランダムフォレスト分類と回帰、勾配ブースティング分類と回帰、KNN、SVM、DBSCAN、低次モーメント。
  - オンライン・アルゴリズム: 共分散、PCA、線形回帰、低次モーメント。
  - DPC++ API をサポートするデータ管理機能を追加: SYCL\* ベースの数値テーブル (`SyclNumericTable`) を表現するための新しいテーブルタイプと最適化された CSV データソース。
  - CPU のグラフ・アナリティクスにテクニカル・プレビュー機能を追加 - ジャカード係数。

### インテル® oneAPI ディープ・ニューラル・ネットワーク・ライブラリー (インテル® oneDNN)

- oneAPI 1.0 仕様に準拠した SYCL\* API 拡張の追加。
- インテル® DPC++ コンパイラーとレベルゼロランタイムのサポート。
- インテル® プロセッサ・グラフィックスおよび X<sup>e</sup> アーキテクチャー・ベースのグラフィックスで統合共有メモリー (USM) をサポート。

### インテル® oneAPI ビデオ・プロセッシング・ライブラリー (インテル® oneVPL)

- AVC/H.264、HEVC/H.265、MJPEG、AV1 ソフトウェア・デコードおよびエンコード。
- ビデオ処理 (サイズ変更、カラー変換、トリミング)。
- ユーザー・インターフェイスと内部割り当てバッファーを備えたフレームメモリー管理。
- DPC++ カーネル統合。

### Python\* 向けインテル® ディストリビューション

- **マシンラーニング**: XGBoost 1.2 (新しい CPU 最適化を含む)、ランダムフォレスト分類/回帰、KNN、スパース K 平均、DBSCAN、SVM、SVC、ランダムフォレスト、ロジスティック回帰などを含む新しい `scikit-learn` および `daal4py` の最適化。
- **GPU の初期サポート**: GPU 対応のデータ並列 NumPy\* (`dpnp`)、`dpnp`、`scikit-learn`、`daal4py`、Numba で DPCTL (デバイス、キュー、USM データ管理のための新しい Python\* パッケージ) を初期

サポート。GPU 向けの daal4py 最適化。DBSCAN、K 平均法、線形回帰、ロジスティック回帰で GPU をサポート。

- ベータ版 Intel® Scalable Dataframe Compiler (Intel® SDC) - pandas\* を高速化する Numba 拡張。

## Intel® Advisor

- **オフロード・アドバイザー:** ハードウェアが手元になくても GPU へ効率良くオフロードするコードを作成できます。オフロードの候補を特定し、潜在的なスピードアップを数値化し、ボトルネックを特定し、データ転送コストを予測し、最適化の方法に関するガイダンスを得ることができます。
- **GPU 向け自動ルーブリック解析:** ハードウェアによって課せられるパフォーマンスの上限に対して GPU カーネルの実際のパフォーマンスを可視化して、メモリーと計算を効率良く最適化するための推奨事項が得られます。
- **メモリーレベルのルーブリック解析:** メモリー階層のボトルネック (L1、L2、L3、または DRAM) をピンポイントで特定します。
- **フローグラフ・アナライザーの DPC++ サポート:** 非同期タスクグラフを可視化し、パフォーマンスの問題を診断し、修正方法の推奨事項を得られます。
- **直感的なユーザー・インターフェイス:** 新しいインターフェイスのワークフローとツールバーに GPU ルーブリック解析とオフロード・アナライザーが追加されました。
- **Intel® Iris® X® MAX グラフィックス・サポート:** ルーブリック解析とオフロード・アドバイザーで Intel® Iris® X® MAX グラフィックスがサポートされました。

## Intel® VTune™ プロファイラー

- DPC++ と OpenMP\* の両方でオフロードコストをプロファイルして、パフォーマンスを低下させるメモリー転送を検出。
- 電力解析を使用して、スロットリングの問題をデバッグし、ワットあたりの FLOPS をチューニング。
- 遅い MMIO 書き込みが行われた場所を特定するように改善された I/O 分析を使用して、パフォーマンスを低下させる I/O 書き込みが発生するモジュールを発見。
- FPGA の各計算ユニットのストールとデータ転送に関する情報を収集して、より少ない推測で FPGA ソフトウェアのパフォーマンスを最適化。
- プロファイルの出発点としての新しいパフォーマンス・スナップショット。最初のプロファイル・ステップとして、パフォーマンス・スナップショットが追加され、最も大きな最適化の機会を特定するため詳細な解析 (メモリー、スレッドなど) を提案。

## oneAPI ベース・ツールキット用 Intel® FPGA アドオン (オプション)

- Linux\* パッケージ・マネージャー (yum、APT、および Zypper) を利用した oneAPI ベースキット用 Intel® FPGA アドオンのインストールをサポート。
- 4 つのアドオン・インストーラーと 3 つの FPGA ボード (Intel® PAC Intel® Arria® 10 GX FPGA 搭載版、Intel® FPGA PAC D5005、およびカスタム・プラットフォーム) をサポート。

## 動作環境

[「インテル® oneAPI ベース・ツールキットの動作環境」](#) (英語) を参照してください。

## 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」](#) (英語) を参照してください。

## ツールの使用方法

以下を参考にしてください。

- [インテル® oneAPI ツールキット入門 \(Linux\\* 版\)](#) (英語)
- [インテル® oneAPI ツールキット入門 \(Windows\\* 版\)](#) (英語)

## 既知の問題、制限事項、および回避方法

1. [DPC++ または OpenMP\\* オフロードを使用したヘテロジニアス・プログラムをデバッグする際の課題、ヒント、および既知の問題に関するホワイトペーパー](#) (英語) を参照してください。
2. **制限事項**
  1. 仮想マシンでの GPU コードの実行は、現在サポートされていません。
  2. オフラインの入門ガイドをダウンロードした場合、Chrome\* ではウィンドウサイズを変更するとテキストが消えることがあります。この問題を解決するには、ブラウザウィンドウのサイズを再度変更するか、別のブラウザを使用してください。
  3. Eclipse\* 4.12 では、Makefile から IDE プラグインによって作成されるサンプル・プロジェクトがビルドできません。これは、Eclipse\* 4.12 の既知の問題です。Eclipse\* 4.9、4.10、または 4.11 を使用してください。
3. **既知の問題 - 名前空間 "oneapi" が古いコンパイラーと競合 - error: reference to 'tbb' is ambiguous**
  1. この問題は、次のコンパイラーでのみ発生します。
    1. GNU\* gcc 7.x 以前
    2. LLVM Clang 3.7 以前
    3. インテル® C++ コンパイラー 19.0 以前
    4. Visual Studio\* 2017 バージョン 15.6 以前
  2. コードで以下の方法で名前空間を使用し、上記のいずれかのコンパイラーを使用する場合、「error: reference to 'tbb' is ambiguous (エラー: 'tbb' への参照が曖昧です)」のようなエラーが出力される場合があります。

oneDPL|oneDNN|oneTBB プログラムコード内の "using namespace oneapi;" ディレクティブは、上記のコンパイラーではコンパイルエラーになる場合があります。

test\_tbb.cpp:

```
namespace tbb { int bar(); }
namespace oneapi { namespace tbb = ::tbb; }

using namespace oneapi;
int zoo0() { return tbb::bar(); }
```

コンパイル出力:

```
test_tbb.cpp: In function 'int zoo0()':
test_tbb.cpp:5:21: error: reference to 'tbb' is ambiguous
int zoo0() { return tbb::bar(); }
```

**回避方法:**

"using namespace oneapi;" ディレクティブの代わりに、完全修飾名または名前空間エイリアスを使用してください。

test\_tbb\_workaround.cpp:

```
namespace tbb { int bar(); }
namespace oneapi { namespace tbb = ::tbb; }

// using namespace oneapi;
int zoo0() { return tbb::bar(); }
```

**追加情報:**

"using namespace oneapi;" ディレクティブは、現在、oneMKL| oneDAL| oneCCL とほかの oneAPI ライブラリーと一緒に使用した場合にコンパイルエラーとなることがあるため推奨しません。この問題には 2 つの回避方法があります。

1. 上記のような完全修飾名前空間を使用します。
2. 以下に示すような、oneMKL| oneDAL| oneCCL の名前空間エイリアスを使用します。

```
namespace one[mkl|dal|ccl] = ::oneapi::[mkl|dal|ccl];
onemkl::blas::dgemm( ... ); | onedal::train(); |
onccl::allgathersv();
```

**既知の問題 - 特定の環境でのみ発生する Windows\* のインストール・エラー「LoadLibrary failed with error 126: the specified module could not be found」**

- **エラーが発生する環境:** AMD\* グラフィックスを搭載した Windows\* システム。
- **詳細:**

AMD\* グラフィックス・カードまたは AMD Radeon\* Vega グラフィックス・ユニットを搭載した Windows\* システムでは、インテル® oneAPI ツールキットのインストーラーがエラー「LoadLibrary failed with error 126: the specified module could not be found (LoadLibrary が エラー 126 で失敗しました: 指定されたモジュールが見つかりません)」を出力する場合があります。この問題はすでに報告済みで、現在調査中です。本リリースでは、次の回避方法を使用してください。

- **回避方法:**

次の手順に従って、インテル® oneAPI ツールキットをインストールする間、インテル® HD グラフィックスを一時的に無効にします。

[デバイス マネージャー] > [ディスプレイ アダプター] を開きます。リストされているディスプレイを右クリックして、**[無効]** を選択します。

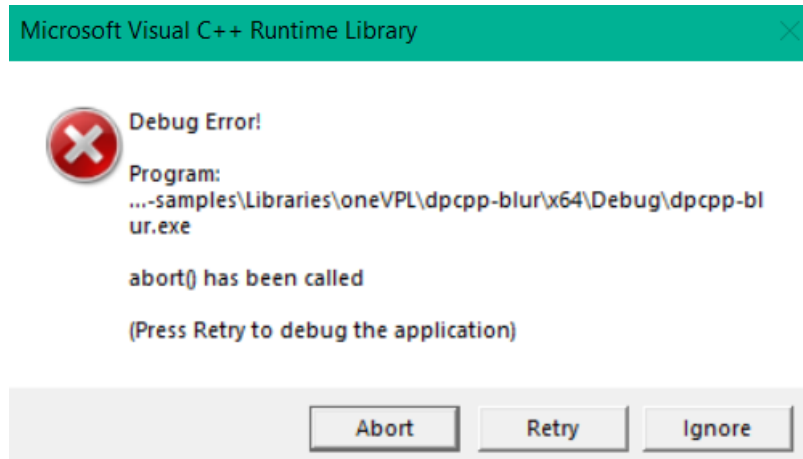
## 既知の問題 - Microsoft\* Visual C++\* ランタイム ライブラリーからのデバッグエラー

- エラーが発生する環境: Windows\* で DPC++ と oneAPI ライブラリー (oneTBB を除く) を併用する「Debug」ビルドのみ。

- 詳細:

このエラーは、oneVPL などの動的・デバッグ・ライブラリーを持たない oneAPI ライブラリーを使用して、DPC++ プログラムを「Debug」構成でビルドした場合に発生することがあります。

次のようなエラーが出力されます。

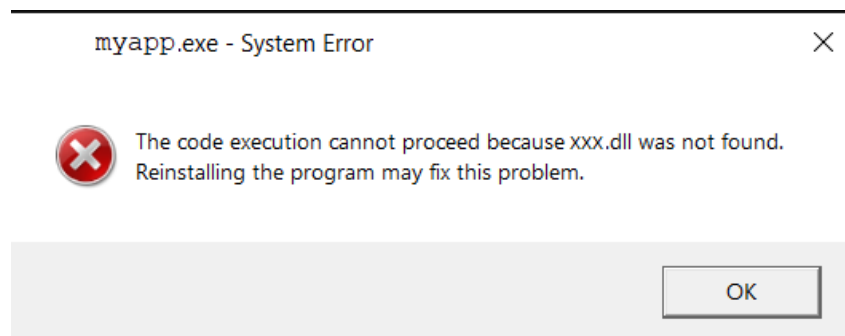


- 回避方法:

1. 「Release」構成でプログラムをビルドしてください。

## Windows\* のその他の制限事項

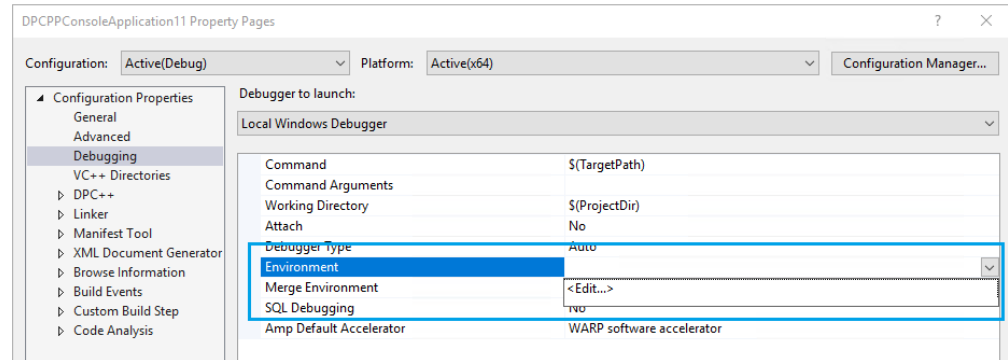
- Visual Studio\* 2017 または 2019 がインストールされているマシンでは、**インテル® oneAPI ベース・ツールキットの IDE 統合のインストールに長い時間を要します**。場合によっては、IDE 統合のインストールに 30 分以上かかることがあります。インストールが完了するまでお待ちください。
- Visual Studio\* でプログラムを実行すると、「... .. sycl.dll was not found. ... .. (sycl.dll が見つかりません)」や次のようなランタイムエラーが出力される場合があります。



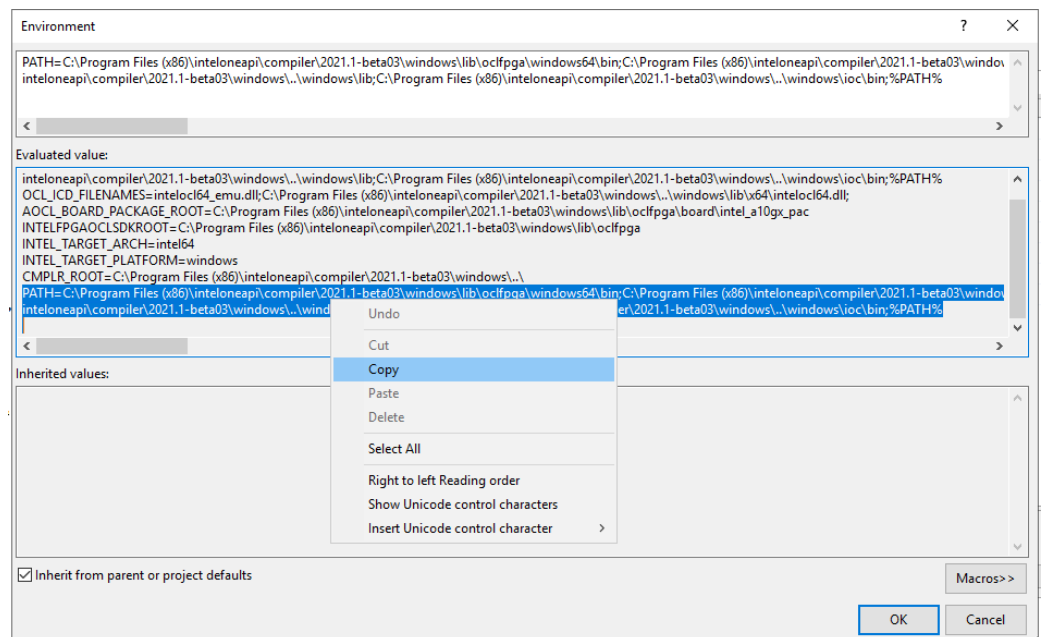


プログラムを実行するには、次の手順に従って [プロジェクト プロパティ] の [デバッグ] > [環境] を更新してください。

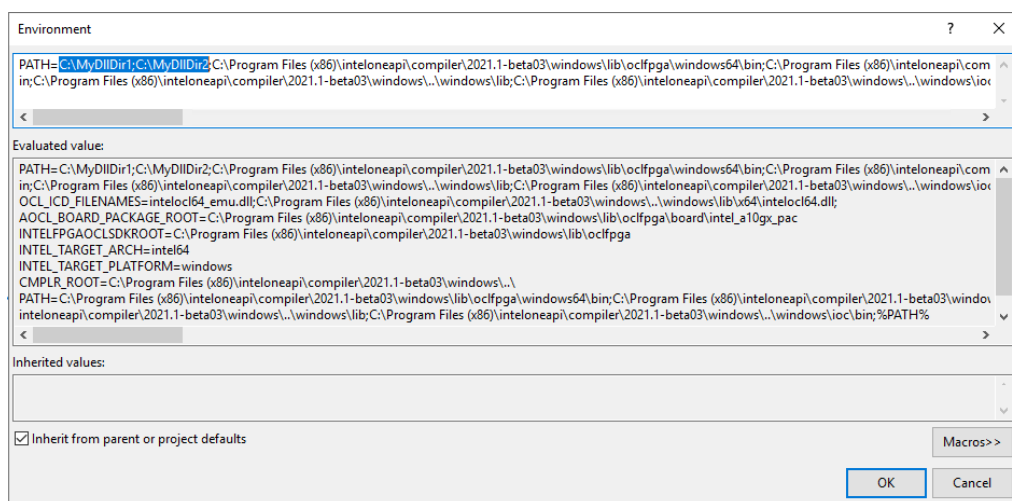
1. [プロジェクト プロパティ] の [デバッグ] > [環境] でドロップダウンから [編集] を選択します。



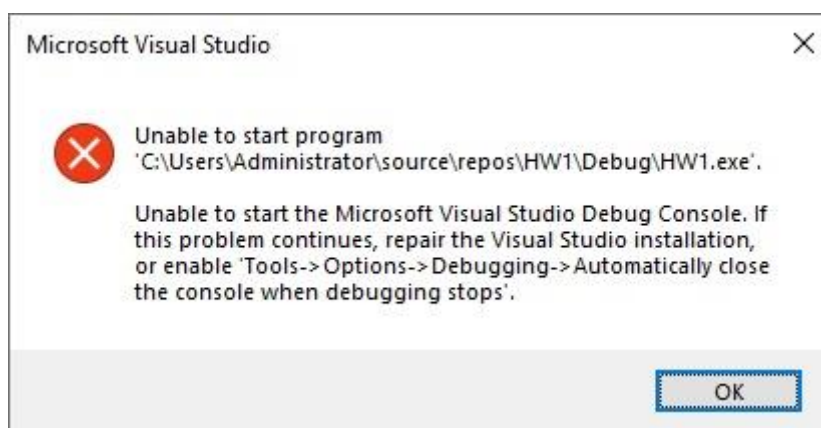
2. デフォルトの PATH 環境変数の値を下のセクションから上のセクションへコピー & ペーストします。Visual Studio\* 2017 以降で PATH 環境変数の追加のディレクトリが正しく処理されるようにするには、この手順が重要です。



3. プログラムに必要なすべての DLL ファイルのディレクトリーをパスに追加します。

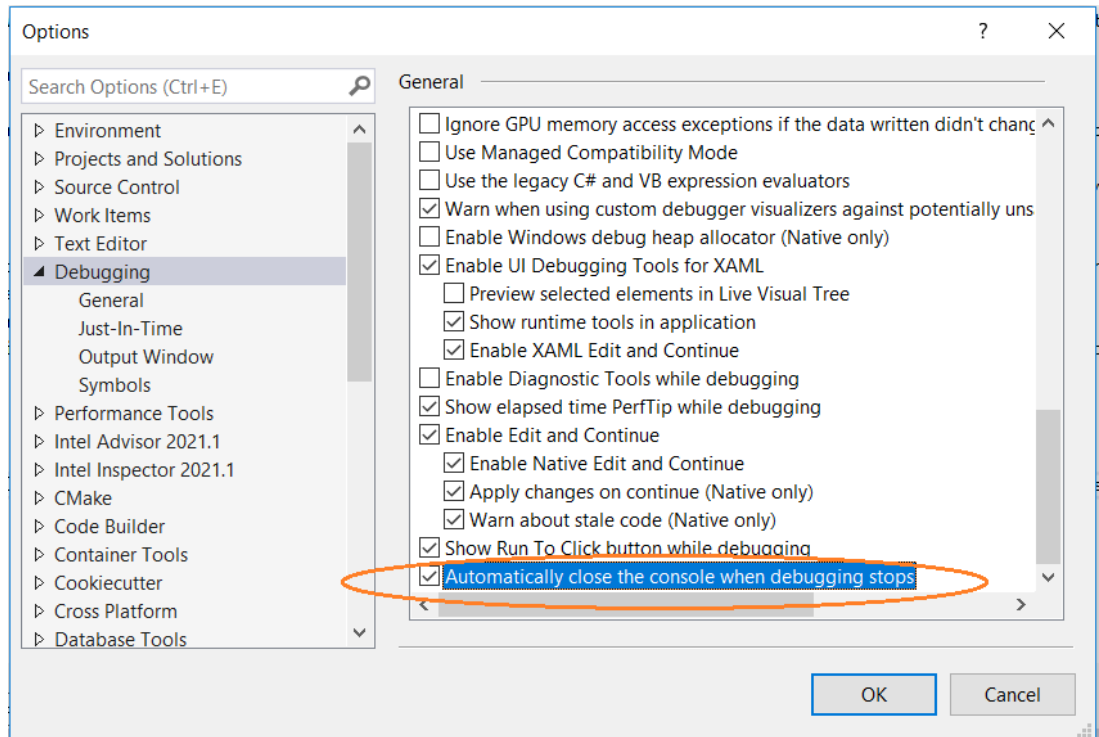


- Visual Studio\* でサンプルプログラムの実行時に「unable to start program 'xxx.exe' (プログラム 'xxx.exe' を開始できません)」エラーが発生することがあります。



この問題を回避するには、次の操作を行ってください。

1. [ツール] > [オプション] を選択して [デバッグ] タブで [デバッグの停止時に自動的にコンソールを閉じる] チェックボックスをオンにします。以下のスクリーンショットを参考にしてください。



# インテル® oneAPI ベース・ツールキットに含まれるツールのリリースノート

- [インテル® oneAPI DPC++/C++ コンパイラー・リリースノート](#)
- [インテル® DPC++ 互換性ツール・リリースノート \(英語\)](#)
- [インテル® oneAPI DPC++ ライブラリー・リリースノート \(英語\)](#)
- [oneAPI ベース・ツールキット用インテル® FPGA アドオン・リリースノート \(英語\)](#)
- [GDB\\* 向けインテル® ディストリビューション・リリースノート \(英語\)](#)
- [インテル® oneAPI マス・カーネル・ライブラリー・リリースノート \(英語\)](#)
- [インテル® oneAPI スレディング・ビルディング・ブロック・リリースノート \(英語\)](#)
- [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・リリースノート \(英語\)](#)
- [インテル® oneAPI データ・アナリティクス・ライブラリー・リリースノート \(英語\)](#)
- [Python\\* 向けインテル® ディストリビューション・リリースノート \(英語\)](#)
- [インテル® VTune™ プロファイラー・リリースノート \(英語\)](#)
- [インテル® Advisor リリースノート \(英語\)](#)
- [インテル® oneAPI ディープ・ニューラル・ネットワーク・ライブラリー・リリースノート \(英語\)](#)
- [インテル® oneAPI コレクティブ・コミュニケーション・ライブラリー・リリースノート \(英語\)](#)
- [インテル® oneAPI ビデオ・プロセッシング・ライブラリー・リリースノート \(英語\)](#)

## 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、Arria、Intel Core、Iris、Stratix、VTune は、アメリカ合衆国および / またはその他の国における 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。

インテル® コンパイラーでは、インテル® マイクロプロセッサに限定されない最適化に関して、他社製マイクロプロセッサ用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサ依存の最適化は、インテル® マイクロプロセッサでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサ用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

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、使用法、構成、およびその他の要因によって異なります。  
詳細については、[www.Intel.com/PerformanceIndex](http://www.Intel.com/PerformanceIndex) (英語) を参照してください。