# インテル® C++ および Fortran コンパイラー 18.0 最適化クイック・リファレンス・ガイド

## およびインテル® VTune™ Amplifier の概要

IA-32 プロセッサー、インテル® 64 プロセッサー、インテル® Xeon Phi™ プロセッサーおよび互換プロセッサー (非インテル製プロセッサー) 向け

製品と購入情報については、インテル® ソフトウェア開発ツールのサイトをご覧ください:
https://www.xlsoft.com/jp/products/category_intel.html

# 目次



## 製品と購入情報については、インテル® ソフトウェア開発ツールのサイトをご覧ください:
https://www.xlsoft.com/jp/products/category_intel.html

§ インテル® Xeon Phi™ プロセッサーは、インテル® Parallel Studio XE に含まれるコンパイラーではサポートされますが、インテル® System Studio に含まれるコンパイラーではサポートされません。

** これらのオプションは、インテル® マイクロプロセッサーおよびインテル製以外のマイクロプロセッサーの両方で利用可能ですが、インテル® マイクロプロセッサー向けのほうが、より多くの最適化が適用される場合があります。‡

## アプリケーション・パフォーマンス

### インテル® コンパイラーでアプリケーションをチューニングする手順

パフォーマンス・チューニングを開始する前に、**/Od (-O0)** を使用して最適化を行わずにアプリケーションをビルドし、正常に動作することを確認してください。

1. 一般的な最適化オプション (Windows* では **/O1**、**/O2**、**/O3**。Linux* および macOS* では **-O1**、**-O2**、**-O3**) を使用してパフォーマンスを測定し、アプリケーションにとって最適なオプションを判断します。通常は、最初に **/O2 (-O2)** (デフォルト) を試してから、より高度な最適化を行うと効果的です。次に、ループを多用するアプリケーションに対しては **/O3 (-O3)** を試します。**

2. **/Qx (-x)** や **/arch (-m)** のようなプロセッサー専用のオプションを使用してきめ細かな最適化を行います。例えば、第 4 世代インテル® Core™ プロセッサー・ファミリー向けには **/QxCORE-AVX2 (-xcore-avx2)** オプション、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3) 以上の命令セットをサポートするインテル製以外の互換プロセッサー向けには **/arch:SSE3 (-msse3)** オプションがあります。また、**/QxHOST (-xhost)** オプションを指定し、コンパイルするホストマシンに搭載されるプロセッサーで利用可能な最上位の命令セットを使用して最適化することもできます。**

3. プロシージャー間の最適化 (IPO) を行う **/Qipo (-ipo)** やプロファイルに基づく最適化 (PGO) を行う **/Qprof-gen** および **/Qprof-use** (**-prof-gen** と **-prof-use**) オプションを追加して、パフォーマンスを測定し、これらの最適化がアプリケーションにとって効果的かどうかを確認します。

4. マルチスレッド、マルチコア、そしてマルチプロセッサー・システム上で並列実行を最適化するため、自動並列化オプション **/Qparallel (-parallel)**、**/Qopenmp (-qopenmp)** オプションによる OpenMP* プラグマやディレクティブ、または、製品に含まれるインテル® パフォーマンス・ライブラリーを使用します。**インテル® Inspector を使用して、メモリーとスレッドのエラーを診断し、開発工程を早めてマルチスレッド・アプリケーションの市場投入までの時間を短縮します。

5. インテル® Advisor とインテル® VTune™ Amplifier†† を使用して、シリアルあるいは並列処理におけるパフォーマンスの「hotspot」を識別し、アプリケーション・コードでさらにチューニングが必要な部分を特定することができます。コンパイラーの最適化レポート **/Qopt-report (-qopt-report)** オプションは、個々の最適化の可能性を特定するのを支援します。

6. また、**/Qopenmp-simd (-qopenmp-simd)** による OpenMP* の SIMD 機能を使用して、明示的なベクトル・プログラミングを行い、アプリケーションを最適化します。**

7. インテル® Xeon Phi™ x200 製品ファミリーでは、ステップ 5 の結果に基づいてデータ・プリフェッチを追加することを検討してください。

詳細については、製品ドキュメントのページをご覧ください: https://software.intel.com/intel-software-technical-documentation (英語)。インテル® コンパイラーの『デベロッパー・ガイドおよびリファレンス』には、インテル® MIC アーキテクチャー向けのアプリケーションのコンパイルに関する節が含まれています。
エクセルソフト社のドキュメント・ページ (https://www.xlsoft.com/jp/products/intel/tech/documents.html?tab=1) では、日本語化されたドキュメントをご案内しております。

** これらのオプションは、インテル® マイクロプロセッサーおよびインテル製以外のマイクロプロセッサーの両方で利用可能ですが、インテル® マイクロプロセッサー向けのほうが、より多くの最適化が適用される場合があります。‡

††製品の一部の機能は、非インテル製マイクロプロセッサー上では利用できません。

### インテル® コンパイラー『ユーザー・リファレンス・ガイド』日本語版ダウンロード：

インテル® コンパイラーのユーザー・リファレンス・ガイド日本語版は、無償評価版登録ページ: https://www.xlsoft.com/jp/products/download/intelj.html より、「インテル® Parallel Studio XE Cluster Edition」の評価版をお申し込みいただくことで入手いただけます。

## 一般的な最適化オプション**

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/Od** | **-O0** | **最適化なし**。アプリケーション開発の初期段階およびデバッグ時に使用します。 |
| **/Os**<br>**/O1** | **-Os**<br>**-O1** | **コードサイズの最小化**。オブジェクト・サイズを増加させる最適化を無効にします。多くの場合に、最も小さな最適化されたコードを生成します。<br>このオプションは、大きなコードサイズに起因するメモリーページングが問題となる大規模なサーバー/データベース・アプリケーションで有効です。 |
| **/O2** | **-O2** | **実行速度の最大化**。デフォルト設定。ベクトル化を含む多くの最適化を有効にします。多くの場合に、**/O1 (-O1)** よりも高速なコードを生成します。 |
| **/O3** | **-O3** | **/O2 (-O2)** で提供される最適化に加え、キャッシュとデータ・プリフェッチをより効率良く使用するため、スカラー置換、ループアンロール、分岐を排除するコードの複製、ループ・ブロッキングなど、より積極的なループとメモリーの最適化を有効にします。<br>**/O3 (-O3)** オプションは、浮動小数点演算を多用するループや大きなデータセットを処理するループを含むアプリケーションに推奨します。これらの積極的な最適化は、アプリケーションの種類により **/O2 (-O2)** と比べて遅くなることがあります。 |
| **/Qopt-report [:*n*]** | **-qopt-report [*n*]** | 最適化レポートを生成します。デフォルトでは、レポートは *.optrpt* 拡張子を持つファイルに出力されます。***n*** には、***0*** (レポートなし) から ***5*** (最も詳しい) の詳細レベルを指定します。デフォルトは ***2*** です。 |
| **/Qopt-report-file:*name*** | **-qopt-report-file=*name*** | 最適化レポートを ***stderr***、***stdout***、またはファイル ***name*** に出力します。 |
| **/Qopt-report-phase:*name1*, *name2*, …** | **-qopt-report-phase=*name1*, *name2*, …** | 最適化フェーズ ***name1***、***name2*** 固有の最適化レポートを生成します。<br>フェーズには、以下のキーワードを指定できます。<br>***all*** – すべてのフェーズのすべての最適化レポート (デフォルト)<br>***loop*** – ループの入れ子とメモリーの最適化<br>***vec*** – 自動ベクトル化と明示的なベクトル・プログラミング<br>***par*** – 自動並列化<br>***openmp*** – OpenMP* によるスレッド化<br>***cg*** – コード生成<br>***ipo*** – インライン展開を含むプロシージャー間の最適化<br>***pgo*** – プロファイルに基づく最適化 |
| **/Qopt-report-help** | **-qopt-report-help** | 上記の **/Qopt-report-phase (-qopt-report-phase)** の ***name*** に指定可能なすべてのキーワードを表示します。コンパイルは実行されません。 |
| **/Qopt-report-routine: *substring*** | **-qopt-report-routine= *substring*** | ***substring*** (部分文字列) を含む関数やサブルーチンのみをレポートします。デフォルトでは、すべての関数とサブルーチンがレポートされます。 |
| **/Qopt-report-filter:*"string"*** | **-qopt-report-filter=*"string"*** | ***"string"*** (文字列) で指定したファイル、関数、サブルーチン、行番号の範囲のみレポートします。<br>例: "myfile、myfun、line1-line2" |
| **/Qopt-report-annotate:*fmt*** | **-qopt-report-annotate=*fmt*** | 最適化情報でソースリストにアノテーションを追加します (デフォルトはオフ)。<br>***fmt*** には、***text*** (デフォルト) または ***html*** を指定できます。 |
| **/Qstd:*cval*** | **-std=*cval*** | より新しい C++ または C 言語標準の機能をサポートします。<br>***cval*** には、c++17、c++14、c++11、c11、c99 などを指定できます。 |

** これらのオプションは、インテル® マイクロプロセッサーおよびインテル製以外のマイクロプロセッサーの両方で利用可能ですが、インテル® マイクロプロセッサー向けのほうが、より多くの最適化が適用される場合があります。‡

## 並列パフォーマンス**

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/Qopenmp** | **-qopenmp** | OpenMP* プラグマ/ディレクティブが記述されている場合にマルチスレッド化されたコードが生成されます。Fortran では、ローカル配列が自動になり、スタックサイズの増加が必要なことがあります。 |
| **/Qopenmp-simd** | **-qopenmp-simd** | OpenMP* SIMD ディレクティブが記述されている場合に SIMD コードが生成されます。 |
| **/Qopenmp-stubs** | **-qopenmp-stubs** | OpenMP* ディレクティブを無視し、OpenMP* ランタイム・ライブラリー関数への参照をシングルスレッド処理と仮定してスタブ (ダミー) 関数にリンクします。 |
| **/Qparallel** | **-parallel** | 自動並列化は、安全に並列実行できる構造のループ (DO CONCURRENT 構文を含む) を検出し、ループのマルチスレッド・コードを自動生成します。 |
| **/Qpar-threshold[:*n*]** | **-par-threshold[*n*]** | パフォーマンス向上の可能性に基づいて、ループの自動並列化のしきい値を設定します。n=0 から 100 が指定でき、デフォルトは 100 です。<br>***n=0*** (計算量にかかわらずループを並列化します) から ***n=100*** (デフォルト、パフォーマンス上の利点があると思われる場合にのみループを並列化します)。<br>**/Qparallel** (**-parallel**) オプションを併用する必要があります。 |
| **/Qpar-affinity:*name*** | **-par-affinity=*name*** | OpenMP* や自動並列化アプリケーション向けにスレッドとプロセッサーのアフィニティーを指定します。***name*** に指定できる値は、***none*** (デフォルト)、***scatter***、**compact** などです。メインプログラムをコンパイルする場合にのみ効果があります。設定と詳細な情報については、インテル® コンパイラーの『デベロッパー・ガイドおよびリファレンス』をご覧ください。 |
| **/Qopt-matmul[-]** | **-q[no-]opt-matmul** | 入れ子の行列乗算ループの特定とコンパイラーが生成する matmul ライブラリー呼び出しとの置換を有効 [無効] にします。**/O3 (-O3)** と **/Qparallel (-parallel)** を指定すると、デフォルトで有効になります。このオプションは、**/O2 (-O2)** 以上を指定しない限り、効果はありません。 |
| **/Qcoarray [:*kywd*]** | **-coarray [=*kywd*]** | Fortran 2008 の Co-Array 機能を有効にします (Fortran のみ)。<br>***kywd*** には、***shared***、***distributed***、***coprocessor***、***single*** オプションを指定できます。詳細は、インテル® コンパイラーの『デベロッパー・ガイドおよびリファレンス』をご覧ください。 |
| **/Qcoarray-num-images:*n*** | **-coarray-num-images=*n*** | ***n*** には、Co-Array 実行ファイルを実行するイメージ数を指定します。<br>デフォルトはオフです (イメージ数は実行時に決定されます)。 |
| **/Qcoarray-config-file:*filename*** | **-coarray-config-file=*filename*** | ***filename*** には、MPI 構成ファイル (パスを含む) を指定します。<br>デフォルトはオフで、MPI のデフォルト設定が使用されます。 |
| **/Qmkl:*name*** | **-mkl=*name*** | インテル® マス・カーネル・ライブラリー (インテル® MKL) をリンクします。<br>デフォルトはオフです。***name*** に指定可能な値は以下のとおりです。<br>***parallel*** – スレッド化されたインテル® MKL をリンク (デフォルト)<br>***sequential*** – スレッド化されていないインテル® MKL をリンク<br>***cluster*** – MPI が実装されたシーケンシャルなインテル® MKL をリンク (クラスター・ライブラリーは macOS* では利用できません) |
| **/Qtbb** | **-tbb** | インテル® スレッディング・ビルディング・ブロック (インテル® TBB) をリンクします。C++ のみ。 |
| **/Qdaal[:*lib*]** | **-daal[=*lib*]** | 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) をリンクします。***parallel*** (***lib*** のデフォルト値) は、スレッド・ライブラリーにリンクします。***sequential*** は、スレッド化されていないライブラリーにリンクします。C++ のみ。 |

** これらのオプションは、インテル® マイクロプロセッサーおよびインテル製以外のマイクロプロセッサーの両方で利用可能ですが、インテル® マイクロプロセッサー向けのほうが、より多くの最適化が適用される場合があります。‡

## 推奨するプロセッサー固有の最適化オプション**

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/Qx***target* | **-x***target* | *target* で指定した命令セットをサポートするインテル® プロセッサー向けの専用コードを生成します。実行ファイルは、非インテル製プロセッサーや、下位の命令セットをサポートするインテル製プロセッサー上では動作しません。*target* に指定可能な命令セットは次のとおりです (上位から下位)。<br>**CORE-AVX512、MIC-AVX512、COMMON-AVX512、CORE-AVX2、AVX、SSE4.2、ATOM_SSE4.2、SSE4.1、ATOM_SSSE3、SSSE3、SSE3、SSE2**<br>**注:** このオプションは、/arch または -m オプションで有効にならない最適化を有効にします。64 ビット macOS* では、**SSE3** と **SSE2** はサポートされません。<br>**注: MIC-AVX512** (インテル® Xeon Phi™ x200 製品ファミリー向けの命令セット) は **CORE-AVX512** のサブセットではありません。**COMMON-AVX512** は両方に共通のサブセットです。 |
| **/arch:***target* | **-m***target* | *target* で指定した命令セットをサポートするすべてのインテル® プロセッサーまたは、非インテル製互換プロセッサー向けの専用コードを生成します。指定した命令セットをサポートしていないインテル製プロセッサーや、非インテル製互換プロセッサーで実行ファイルを実行すると、ランタイムエラーが発生する場合があります。<br>*target* に指定可能な値: **avx、sse4.2、sse4.1、ssse3、sse3、sse2、ia32**<br>**注: ia32** オプションは、専用ではない、x86/x87 汎用コードを生成します。これは、IA-32 アーキテクチャーでのみサポートされます。macOS* ではサポートされません。 |
| **/QxHOST** | **-xhost** | コンパイルを行うホストシステム上でサポートされる最上位の命令セットを利用するコードを生成します。インテル製プロセッサー上では **/Qx (-x)** オプションに相当します。非インテル製互換プロセッサー上では **/arch (-m)** オプションに相当します。 |
| **/Qax***target* | **-ax***target* | *target* で指定した命令セットをサポートするインテル® プロセッサー向けの専用コードとデフォルトコード (SSE2) を生成します。*target* に指定可能な値:<br>**CORE-AVX512、MIC-AVX512、COMMON-AVX512、CORE-AVX2、AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2**<br>カンマで区切った複数の値を指定することで、同じ実行ファイルにほかのインテル製プロセッサー向けにチューニングされたコードを含めることができます。例: **/QaxAVX、SSE4.2**。デフォルトのコードパスは、**SSE2** をサポートするすべてのインテル製または非インテル製互換プロセッサー上で実行できますが、**/Qx (-x)** や **/arch (-m)** オプションを使用して変更することができます。<br>例えば、第 4 世代インテル® Core™ プロセッサー・ファミリー向けに最適化されたコードパスと、**SSE3** をサポートするインテル製プロセッサーまたは非インテル製互換プロセッサー向けに最適化されたデフォルトのコードパスを生成するには、**/QaxCORE-AVX2 /arch:SSE3** (Linux* では **-axcore-avx2 -msse3**) を使用します。<br>アプリケーションは実行時にインテル® プロセッサーで実行されているか自動検出し、最適なコードパスを選択します。インテル製プロセッサーが検出されなかった場合、デフォルトのコードパスが選択されます。 |
| **/Qvecabi:cmdtarget** | **-vecabi=cmdtarget** | 上記の /Qx または /Qax (-x または -ax) オプションで指定されたターゲット向けに SIMD 関数のベクトルバージョンを作成します。 |
| **/Qopt-zmm-usage:*high*** | **-qopt-zmm-usage=*high*** | /QxCORE-AVX512 (Windows*) または -xcore-avx512 (Linux*) とともに使用すると、512 ビット SIMD 命令を積極的に生成します。 |
| **/tune:*proc*** | **-mtune=*proc*** | 専用のコードを生成しないでプロセッサー ***proc*** 向けに最適化します。<br>***proc*** に指定可能な値: ***skylake***、***knl***、***broadwell***、***haswell***、***ivybridge***、***sandybridge***、***corei7***、***silvermont***、***generic*** など。<br>実行ファイルはほかのプロセッサーでも動作します。 |

オンライン記事「インテル® SSE およびインテル® AVX 世代 (SSE2、SSE3、SSSE3、ATOM_SSSE3、SSE4.1、SSE4.2、ATOM_SSE4.2、AVX、AVX2 ) 向けのインテル® コンパイラー・オプションとプロセッサー固有の最適化 (https://www.isus.jp/products/c-compilers/performance-tools-for-software-developers-intel-compiler-options/)」で、プロセッサー専用の推奨最適化オプションを参照してください。

** これらのオプションは、インテル® マイクロプロセッサーおよびインテル製以外のマイクロプロセッサーの両方で利用可能ですが、インテル® マイクロプロセッサー向けのほうが、より多くの最適化が適用される場合があります。‡

## インテル® Xeon Phi™ x200 製品ファミリー向けの最適化

| Windows* | Linux* | 説明 |
|---|---|---|
| **/Qopt-threads-per-core:*n*** | **-qopt-threads-per-core=*n*** | 物理コアあたり ***n*** スレッドで最適化するようにコンパイラーに指示します。***n=1***、***2***、***3*** または ***4***。 |
| **/Qopt-prefetch:*n*** | **-qopt-prefetch=*n*** | ソフトウェア・プリフェッチのレベルを ***n=0*** から ***5*** で指定します。<br>インテル® Xeon Phi™ x100 製品ファミリーでは、最適化レベル **-O2** 以上のデフォルトは ***n=3*** です。 |
| **/Qopt-prefetch-distance=*n1[,n2]*** | **-qopt-prefetch-distance=*n1[,n2]*** | データ・プリフェッチ前のベクトル化されるループ反復の数を指定します。<br>***n1*** は L2 キャッシュ、***n2*** (≤***n1***) は L1 キャッシュ向けです。デフォルトはオフです (コンパイラーが選択します)。 |
| **/Qimf-domain-exclusion:*n*** | **-fimf-domain-exclusion=*n*** | 数学関数が IEEE 標準に準拠する必要がない、特殊なケースの引数を指定します。n のビットは以下に相当します。***0*** – 極値 (非常に大きい、非常に小さい、特異値に近いなど)、***1*** – NaN、***2*** – 無限大、***3*** – デノーマル値、***4*** – ゼロ。 |
| **/align: array64byte** | **-align array64byte** | アライメントされたロードとベクトル化を支援するため、64 で割り切れるメモリーアドレスに配列の先頭を配置するように指示します。(Fortran のみ)。 |
| **/Qopt-assume-safe-padding** | **-qopt-assume-safe-padding** | ユーザープログラムと同様に、コンパイラーが配列や動的に割り当てられたオブジェクトの終端から最大 64 バイトを超えて安全にアクセスできることを示します。パディングの追加はユーザーの責任です。デフォルトはオフです。 |

詳細は、https://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-code-named-knights-landing-application-readiness (英語)、https://www.isus.jp/products/psxe/low-precision-optimizations/ およびインテル® コンパイラーの『デベロッパー・ガイドおよびリファレンス』(https://software.intel.com/en-us/cpp-compiler-18.0-developer-guide-and-reference (C++、英語) および https://software.intel.com/en-us/fortran-compiler-18.0-developer-guide-and-reference (Fortran、英語) をご覧ください。

§ インテル® Xeon Phi™ プロセッサーは、インテル® Parallel Studio XE に含まれるコンパイラーではサポートされますが、インテル® System Studio に含まれるコンパイラーではサポートされません。

## プロシージャー間の最適化 (IPO) オプションと プロファイルに基づく最適化 (PGO) オプション

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/Qip** | **-ip** | 現在のソースファイルを対象にしたインライン展開を含む、単一ファイルのプロシージャー間の最適化を行います。 |
| **/Qipo[*n*]** | **-ipo[*n*]** | 複数のソースファイルにまたがるインライン展開やその他のプロシージャー間の最適化を許可します。オプションの引数 ***n*** は、リンク時コンパイルの最大数 (オブジェクト・ファイル数) を制御します。デフォルトは、***n=0*** です (コンパイラーが判断します)。<br>**注:** 状況によりコンパイル時間とコードサイズが大幅に増えることがあります。 |
| **/Qipo-jobs[*n*]** | **-ipo-jobs[*n*]** | プロシージャー間の最適化 (IPO) のリンクフェーズで、同時に実行するコマンド (ジョブ) の数 ***n*** を指定します。デフォルトは ***1*** です。 |
| **/Ob2** | **-finline-functions -finline-level=2** | コンパイラーの判断により、現在のソースファイル内で関数のインライン展開を有効にします。**/O2** と **/O3** (**-O2** と **-O3**) ではデフォルトで有効になります。<br>**注:** ファイルサイズが大きい場合、コンパイル時間とコードサイズが大幅に増えることがあります。**/Ob0** (Linux* と macOS* では **-fno-inline-functions**) で無効にできます。 |
| **/Qinline-factor:*n*** | **-finline-factor=*n*** | インライン展開できる関数の合計サイズと最大サイズの係数を設定します。デフォルトは ***n=100*** で、100% またはスケーリング係数 1 を示します。 |
| **/Qprof-gen [:*kywd*]** | **-prof-gen [=*kywd*]** | プロファイル情報を生成するためインストルメントを行います。***kywd=threadsafe*** は、スレッド化されたアプリケーションのプロファイル生成を可能にします。***kywd=srcpos*** と ***globdata*** は、関数やデータの並び替えに役立つ追加情報を収集します。 |
| **/Qprof-use** | **-prof-use** | 最適化でプロファイル情報を使用するようにします。 |
| **/Qprof-dir *dir*** | **-prof-dir *dir*** | プロファイル結果の出力ファイル (*.dyn および *.dpi) を格納するディレクトリーを指定します。(あるいは、**PROF_DIR** 環境変数を使用してディレクトリーを指定します。) |
| **/Qprof-gen-sampling** | **-prof-gen-sampling** | インテル® VTune™ Amplifier を使用したハードウェア・イベント・ベースのプロファイル生成で使用する追加のデバッグ情報を生成します。 |
| **/Qprof-use-sampling:*file*** | **-prof-use-sampling=*file*** | 最適化で ***file*** のハードウェア・イベント・ベースのプロファイル情報を使用するようにします。 |
| **/Qprofile-functions** | **-profile-functions** | 各関数の実行時間を生成するようにインストルメント関数を追加します。 |
| **/Qprofile-loops** | **-profile-loops** | 関数をインストルメントし、各ループまたはループの入れ子のシリアルコードのプロファイルを生成します。プロファイルの表示方法やその他の詳細については、インテル® コンパイラーの『デベロッパー・ガイドおよびリファレンス』で、「関数またはループの実行時間のプロファイル」をご覧ください。 |

## きめ細かなチューニング (すべてのプロセッサー)**

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/Qunroll[*n*]** | **-unroll[*n*]** | ループアンロール回数の上限を設定します。***n=0*** はループアンロールを無効にします。デフォルトは、デフォルトのヒューリスティックを使用する **/Qunroll (-unroll)** です。 |
| **/Qopt-prefetch:*n*** | **-qopt-prefetch=*n*** | さまざまなレベルのソフトウェア・プリフェッチを有効にします。***n=0*** (デフォルト、プリフェッチなし) から ***n=5*** (最大限のプリフェッチ) で指定します。<br>警告: 過度のプリフェッチは、リソースの競合によりパフォーマンスが低下することがあります。 |
| **/Qopt-prefetch-issue-excl-hint** | **-qopt-prefetch-issue-excl-hint** | prefetchw 命令をサポートするプロセッサーで過度のプリフェッチ (書き込みの予測) の生成を有効にします。 |
| **/Qopt-block-factor:*n*** | **-qopt-block-factor=*n*** | デフォルトのヒューリスティックをオーバーライドするブロッキング係数 ***n*** (ブロック内のループ反復回数) を指定します。ループ・ブロッキングは、**/O3** (**-O3**) で有効になり、キャッシュのデータを再利用するように設計されています。 |
| **/Qopt-streaming-stores:*mode*** | **-qopt-streaming-stores *mode*** | **always** - アプリケーションのメモリー再利用が少ないと想定し、キャッシュをバイパスするストリーミング・ストアの生成を促進します。<br>**never** - ストリーミング・ストアの生成を無効にします。<br>**auto** - デフォルトのコンパイラーのヒューリスティックを使用します。 |
| **/Qrestrict[-]** | **-[no]restrict** | **restrict** キーワードによるポインターの一義化を有効 [無効] にします。デフォルトはオフです。(C/C++ のみ)。 |
| **/Oa** | **-fno-alias** | プログラムにエイリアシングがないことを想定します。デフォルトはオフです。 |
| **/Ow** | **-fno-fnalias** | 関数内にエイリアシングがないことを想定します。デフォルトはオフです。 |
| **/Qalias-args[-]** | **-fargument-[no]alias** | 関数の引数のエイリアス化を有効 [無効] にします。デフォルトはオンです。(C/C++ のみ)。**-fargument-noalias** は、配列引数を持つ関数呼び出しを含む C または C++ のループのベクトル化に役立ちます。 |
| **/Qansi-alias[-]** | **-[no-]ansi-alias** | ANSI および ISO C 標準の別名規則を有効 [無効] にします。<br>デフォルト: Windows* では無効、Linux* と macOS* では有効。 |
| **/Qopt-class-analysis[-]** | **-q[no-]opt-class-analysis** | C++ クラス階層情報を使用して、コンパイル時に C++ 仮想関数の呼び出しを解決します。C++ アプリケーションに標準的ではない C++ 構造 (ポインターのダウンキャスティングなど) が含まれている場合、動作が異なることがあります。デフォルトではオフですが、**/Qipo** (Windows*) や **-ipo** (Linux* と macOS*) オプションを指定すると、C++ の最適化を高めるためオンになります。(C++ のみ)。 |
| **/Qvec-threshold:0** | **-vec-threshold=0** | パフォーマンスの向上が見込めない場合でもループを自動ベクトル化するようにコンパイラーに指示します。 |
| **/Qvec[-]** | **-[no-]vec** | ベクトル化を有効 [無効] にします。デフォルトは、**/O2** (**-O2**) で有効です。 |
| **/Qstringop-strategy:*alg*** | **-mstringop-strategy=*alg*** | *memcpy* や *memset* のようなバッファー操作関数で使用するアルゴリズム ***alg*** を設定します。<br>***libcall***: ライブラリー呼び出しを行うようにコンパイラーに指示します。<br>***rep***: ***rep movs*** または同様のシーケンスを使用してインライン展開します。<br>***const_size_loop*** (デフォルト): コンパイル時にサイズが判明していて [**/Q** \| **-m**]**stringop-inline-threshold** 未満の場合、ループにインライン展開します。 |

| | | |
|---|---|---|
| **/align:array*nn*byte** | **-align array*nn*byte** | アライメントされたロードとベクトル化を支援するため、***nn*** で割り切れるメモリーアドレスに配列の先頭を配置するように指示します。(Fortran のみ)。 |
| **/assume [no] buffered_io** | **-assume [no] buffered_io** | I/O 効率を向上するため、連続するシーケンシャルな読み書きデータをバッファリングします。デフォルトでは、バッファリングしません。(Fortran のみ)。 |
| **/assume:contiguous_assumed_shape** | **-assume contiguous_assumed_shape** | 形状引き継ぎ仮引数が常に連続している (ユニットストライドできる) と仮定します。(Fortran のみ)。 |
| **/assume: contiguous_pointer** | **-assume contiguous_pointer** | ポインター仮引数が常に連続している (ユニットストライドできる) と仮定します。(Fortran のみ)。 |
| | **-f[no-]exceptions** | C++ の場合、**-fexceptions** がデフォルトで、例外処理テーブルの生成を有効にします。ベクトル化が適切に行われないことがあります。**-fno-exceptions** は例外指定を解析しますが無視されます。呼び出しに **-fno-exceptions** でコンパイルされた関数が含まれている場合、try ブロックや throw 文はエラーになります。 |
| **/Qfnsplit:*n*** | **-fnsplit=*n*** | 到達する確率が ***n*%** 未満の条件付きコードブロックを異なるコードセグメントに配置します。 |

** これらのオプションは、インテル® マイクロプロセッサーおよびインテル製以外のマイクロプロセッサーの両方で利用可能ですが、インテル® マイクロプロセッサー向けのほうが、より多くの最適化が適用される場合があります。‡

## 浮動小数点演算オプション

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/fp:*name*** | **-fp-model *name*** | 特定の最適化を制限することで、浮動小数点演算のパフォーマンス、精度、再現性を制御します。<br>***name*** に指定可能な値は以下のとおりです。<br>***fast[=1\|=2]*** – 精度と一貫性を多少低くすることにより、さらに強力な最適化が可能になります。(デフォルトは ***fast=1***)。<br>***precise*** – FMA (Fused Multiplty Add) 命令を生成する場合を除いて、異なる浮動小数点結果になるコンパイラーの最適化を無効にします。<br>***double/extended/source*** – 中間結果をそれぞれ倍精度、拡張精度、ソースの精度で丸めます。オーバーライドされない限り ***precise*** も適用されます。インテル® Fortran コンパイラーでは、***double*** と ***extended*** はサポートされません。***precise***、***strict***、***consistent*** が指定された場合、Intel64 のデフォルトは ***source*** です。<br>***except*** – 浮動小数点例外セマンティクスを強制します。<br>***strict*** – ***precise*** と ***except*** オプションを有効にし、デフォルトの浮動小数点環境を仮定しません。コンパイラーが、FMA (Fused Multiplty Add) 命令を生成しないように強制します。<br>***consistent*** – 異なる最適化レベルや同じアーキテクチャーの異なるプロセッサーで、一貫した再現性のある結果を有効にします。 |
| **/Qopt-dynamic-align[-]** | **-q[no-]opt-dynamic-align** | 実行時のデータ・アライメントに依存し、同じシリアル・アプリケーションを同じ入力データで繰り返し実行した際に、わずかな浮動小数点結果の違いを引き起こす可能性のある最適化を有効 [無効] にします。**/fp:precise** または **/fp:consistent** (**-fp-model precise** または **-fp-model consistent**) が指定されない限り、デフォルトで有効です。 |

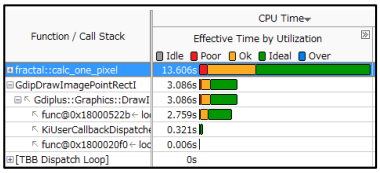
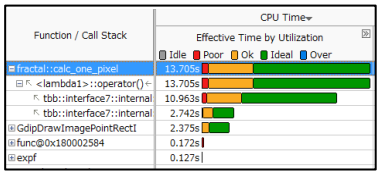
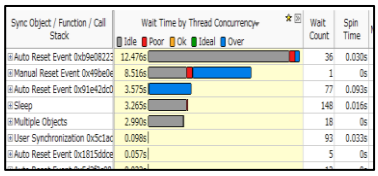
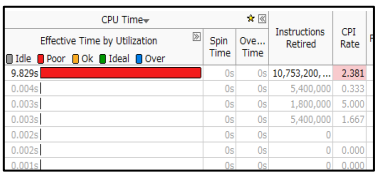
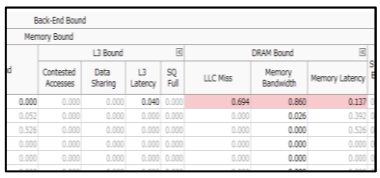
| | | |
|---|---|---|
| **/Qftz[-]** | **-[no-]ftz** | main プログラムや dll の main をこのオプションでコンパイルすると、プログラム (dll) 全体 で実行時にインテル® SSE やインテル® AVX 命令によるデノーマル結果がゼロにフラッシュされます。**/Od (-O0)** が指定されない限り、デフォルトで有効です。 |
| **/Qimf-precision:** ***name*** | **-fimf-precision:** ***name*** | 算術ライブラリー関数の精度を設定します。デフォルトはオフです (コンパイラーは、デフォルトのヒューリスティックを使用します)。***name*** には、***high***、***medium***、***low*** を指定できます。精度を下げると、特にベクトル化されたコードのパフォーマンスが向上する可能性がありますが、その逆もあり得ます。 |
| **/Qimf-arch-consistency:*true*** | **-fimf-arch-consistency=*true*** | 算術ライブラリー関数が、同じアーキテクチャーの異なるインテル製プロセッサーまたは非インテル製互換プロセッサーにおいて一貫した結果を生成するようにします。これにより、ランタイム・パフォーマンスが低下することがあります。**/fp:consistent** (**-fp-model consistent**) が指定されない限り、デフォルトは ***false*** (オフ) です。 |
| **/Qprec-div[-]** | **-[no-]prec-div** | 浮動小数点除算の精度を上げます [下げます]。パフォーマンスが低下 [向上] することがあります。 |
| **/Qprec-sqrt[-]** | **-[no-]prec-sqrt** | 平方根計算の精度を上げ [下げ] ます。パフォーマンスが低下 [向上] することがあります。 |
| **/Qprotect-parens[-]** | **-f[no-]protect-parens** | 括弧で指定した順序で式が評価されます。**/fp:precise** または **/fp:consistent** (**-fp-model precise** または **-fp-model consistent**) が指定されない限り、デフォルトはオフです。 |
| **/Qfma[-]** | **-[no]fma** | コンパイラーが、FMA (Fused Multiplty Add) 命令を生成しないように強制します (ただし、ランタイム・ライブラリーでは FMA 命令が使用される可能性があります)。 |
| **/Qimf-use-svml** | **-fimf-use-svml** | インテルの数学ライブラリー (LIBM) の代わりに、SVML (Short Vector Math Library) を使用してスカラー数学関数を実装するようにコンパイラーに指示します。 |
| **/Qimf-force-dynamic-target** | **-fimf-force-dynamic-target** | プロセッサーの種類に基づいて実行時に算術ライブラリー関数のコードパスを選択します。デフォルトはオフです。 |
| **/Qfp-speculation safe** | **-fp-speculation safe** | 浮動小数点例外が起こる可能性がある場合は、特定の最適化を無効にするようにコンパイラーに指示します。浮動小数点例外がデバッグでマスク解除されている場合に設定すると便利です。 |

https://www.isus.jp/products/c-compilers/consistency-of-floating-point-results/ もご覧ください。

# インテル® VTune™ Amplifier を使用したパフォーマンス解析

**Hotspot:** コードの中で CPU を使用してる時間が最も多い関数またはループの箇所です。
**ボトルネック:** コードの中で全体の最大の実行速度に影響する、最も遅い箇所です。
**ボトルネックの原因:** ジャンプするメモリー参照や、分岐予測ミスなどの処理が遅延の原因になります。

インテル® VTune™ Amplifier を用いてパフォーマンス解析を行うことで、
1. Hotspot を発見し、
2. その箇所がボトルネックであるかを検出し、
3. ボトルネックの原因を特定して修正することで、パフォーマンスを改善させることができます。

| 主要な解析タイプ | 説明 | |
|---|---|---|
| **Basic Hotspots (コマンドライン解析タイプ: hotspots)** | Hotspot (CPU 時間を要しているソースコードの箇所) を特定します。<br>実際の CPU 時間とともに、表の一番上に最も CPU 時間を要している Hotspot が表示されます。 |  |
| **Concurrency (コマンドライン解析タイプ: concurrency)** | プログラムのマルチスレッド動作を解析します。<br>実時間 (Elapsed Time) の内、プロセッサーのすべてのコアを同時に利用できていた時間が長いほど、Ideal (望ましい) と分類されます。 |  |
| **Locks & Waits (コマンドライン解析タイプ: locksandwaits)** | 待ち状態による CPU 時間の浪費状況を解析します。<br>マルチスレッドにおけるクリティカルセクションでの同期や、入出力待ちなどが性能に影響しているかどうかを確認できます。 |  |
| **Advanced Hotspots (コマンドライン解析タイプ: advanced-hotspots)** | プログラムの各場所での CPI 値を求めます。<br>CPI とは、実行に要した CPU サイクル数と、実行された CPU 命令数の比であり、プログラムの実行効率を表す基本的な評価基準です。 |  |
| **General Exploration (コマンドライン解析タイプ: general-exploration)** | キャッシュミスや分岐予測ミスなど、CPU の動作に関連する詳細な性能情報やボトルネックを取得します。<br>解析結果において性能劣化の原因と見られるボトルネックを特定します。ボトルネックの原因となる項目や値はピンク色で強調表示されます。 |  |

**本製品の詳細は、製品サイトをご覧ください:** https://www.xlsoft.com/jp/products/intel/vtune/index.html

## デバッグオプション

| Windows* | Linux*/macOS* | 説明 |
|---|---|---|
| **/Zi**<br>**/debug**<br>**/debug:full**<br>**/debug:all** | **-g**<br>**-debug**<br>**-debug full**<br>**-debug all** | 最適化されていないコードの完全なシンボリック・デバッグのため、デバッグ情報を生成します。最適化オプション **/O (-O)** を指定した場合に有効になります。デバッグシンボルの生成は、一般的にオブジェクト・モジュールのサイズを増加させ、最適化されたコードのパフォーマンスをわずかに低下させることがあります。 |
| **/debug:none** | **-debug none** | デバッグ情報は生成されません。(デフォルト)。 |
| **/debug:minimal** | **-debug minimal** | ローカルシンボルではなく、デバッグ用の行番号情報を生成します。 |
| **/debug:inline-debug-info** | **-debug inline-debug-info** | このオプションを指定すると、インライン展開される関数のシンボルは、呼び出し元ではなく、呼び出される関数のソースに関連付けられます。<br>**-O2** を指定しない限り、**/debug:full** (**-debug full**) だけでは有効になりません。 |
| | **-debug extended** | 最適化されたコードのシンボリック・デバッグを改善するため、追加情報を生成します。(Linux* のみ。**debug full** では有効になりません。) |
| | **-debug parallel** | スレッド化されたコードのデバッグのため、追加のシンボルとインストルメント・コードを生成します。(Linux* のみ。**debug full** では有効になりません。) |
| **/Qsox[-]** | **-[no-]sox**<br>(Linux* のみ) | オブジェクト・ファイル (Windows* と Linux*) と実行ファイル (Linux*) に、コンパイラーのバージョンとオプションを文字列で埋め込みます。<br>デフォルトはオフです。 |
| **/Qtraceback** | **-traceback** | ランタイム時に致命的なエラーが発生したとき、ソースファイルのトレースバック情報を表示できるように、オブジェクト・ファイル内に補足情報を生成します。最適化されたコードで利用します。(Fortran アプリケーションのみ)。 |
| **/Qinit:snan, arrays** | **-init=snan, arrays** | 実行時に初期化されていない浮動小数点変数を検出しやすくするため、特定の浮動小数点変数をシグナル型 NaN に初期化します。(Fortran のみ)。<br>浮動小数点例外のマスクを解除するには、/fpe:0 (-fpe0) を指定します。 |
| **/Qinit:huge** | **-init=huge** | 特定の浮動小数点変数および整数変数を表示可能な最大値に初期化します。 |
| **/Qcheck-pointers:*kywd*** | **-check-pointers=*kywd*** | ポインターによるメモリーアクセスの範囲チェックを有効にします。<br>***kywd*** には、***none*** (デフォルト)、***write*** または ***rw*** (読み取りと書き込み) を指定できます。(C/C++ のみ)。 |
| **/Qcheck-pointers-mpx:*kywd*** | **-check-pointers-mpx=*kywd*** | ポインターによるメモリーアクセスのハードウェア支援範囲チェックを有効にします。***kywd*** には、***none*** (デフォルト)、***write*** (書き込みのみチェック) または ***rw*** (読み取りと書き込みをチェック) を指定できます。(C/C++ のみ)。 |
| **/Qcheck** | **-check** | 配列境界、スタックフレーム、初期化されていないポインター、連続した形状引き継ぎ仮引数またはポインター仮引数、割付けられていない割付け配列などのランタイムチェックを有効にします。(Fortran のみ)。 |
| **/Qfp-trap:common**<br>**/fpe:0** | **-fp-trap=common**<br>**-fpe0** | オーバーフロー、無効な浮動小数点例外およびゼロ除算浮動小数点例外をマスク解除します。(C/C++ メインのみ)。<br>(Fortran メインのみ)。 |

**製品と購入情報については、インテル® ソフトウェア開発ツールのサイトをご覧ください:**
https://www.xlsoft.com/jp/products/category_intel.html

## ‡ 最適化に関する注意事項

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ー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ー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ーに関して、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ー依存の最適化は、インテル® マイクロプロセッサー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ー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。

注意事項の改訂 #20110804