最適化クイック・リファレンス・ガイド

# インテル® C++ および Fortran コンパイラー 15.0

IA-32 プロセッサー、インテル® 64 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:
http://www.xlsoft.com/jp/products/category_intel.html

## 内容

## アプリケーション・パフォーマンス

インテル® コンパイラーによるアプリケーション・チューニングの手順を示します。パフォーマンス・チューニングを始める前に **/Od (-O0)** なしでビルドを行い、アプリケーションの正当性 (正しく動作するか) を確認します。

1. 汎用最適化オプション (Windows* **/O1、/O2 もしくは /O3**; Linux* と OS X* **-O1、-O2 もしくは -O3**) を使用し、それぞれパフォーマンスを計測することで最適なオプションを調査します。ほとんどのユーザーは、高度な最適化を始める前にデフォルトの /O2 (-O2) オプションを使用する必要があります。
   ループ主体のアプリケーションでは、次に **/O3 (-O3)** オプションを試してください。**
2. パフォーマンスを細かくチューニングするには、プロセッサー固有オプションを使用します。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)** を、少なくともインテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3) 命令セットをサポートする非インテル製互換プロセッサー向けには、**/arch:SSE3 (-msse3)** を指定します。あるいは、コンパイルを行っているシステムのプロセッサーで利用可能な最上位の命令セットを利用する、**/QxHOST (-xhost)** を利用できます。**
   特定のプロセッサーまたは命令セット向けの最適化に関する詳しいオプションのリストは、推奨するプロセッサー固有最適化オプションをご覧ください。
3. プロシージャー間の最適化 (IPO) ― **/Qipo (-ipo)** ―、プロファイルに基づく最適化 (PGO) ― **/Qprof-gen と /Qprof-use (-prof-gen と -prof-use)** ―を追加し、どちらか一方もしくは両方のオプションからアプリケーションに利点があるかどうか検証します。
4. インテル® VTune™ Amplifier XE†† は、シリアルおよび並列パフォーマンスの "ホットスポット" を特定するのに役立ちます。これにより、開発者はチューニングの利点があるアプリケーションの特定の領域を知ることができます。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。
5. また、C/C++ 向けのインテル® Cilk™ Plus 言語拡張や、**/Qopenmp-simd (-qopenmp-simd)**† による OpenMP* 4.0 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。
6. 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)**; C/C++ 向けのインテル® Cilk™ Plus 言語拡張; **/Qopenmp (-qopenmp)**† オプションによる OpenMP* プラグマや宣言子; 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 XE†† 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。

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。インテル® コンパイラーの『ユーザー・リファレンス・ガイド』には、インテル® MIC アーキテクチャーとインテル® グラフィックス・テクノロジ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。

**これらのオプションはインテル製マイクロプロセッサーおよび互換マイクロプロセッサーで利用可能ですが、インテル製マイクロプロセッサーにおいてより多くの最適化が行われる場合があります。‡

†OpenMP* は、インテル® Parallel Studio XE に含まれるコンパイラーでサポートされますが、インテル® System Studio やインテル® INDE に含まれるコンパイラーではサポートされません。
††これらの製品は、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。

## 汎用最適化オプション

これらのオプションはインテル製マイクロプロセッサーおよび互換マイクロプロセッサーで利用可能ですが、インテル製マイクロプロセッサーにおいてより多くの最適化が行われる場合があります。

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/Od** | **-O0** | **最適化なし。**アプリケーション開発の初期段階やデバッグ時に使用。アプリケーションが正常に動作している場合、より高度な最適化オプションを使用します。 |
| **/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>これらのオプションは、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。ほとんどのケースで、**/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よく使用するため、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*** もしくはファイル ***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。***name*** 引数に指定できるキーワードには、以下のようなものがあります:<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>**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<br>**offload** – インテル® MIC アーキテクチャーもしくはインテル® グラフィックス・テクノロジー・デバイスへのデータと (もしくは) 実行のオフロード |
| **/Qopt-report-help** | **-qopt-report-help** | 上記の **/Qopt-report-phase (-qopt-report-phase)** の ***name*** で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) で指定されるファイル、関数、サブルーチンと (もしくは) 行番号の範囲をレポートしません。<br>例: “myfile、myfun、line1-line2” |

## 並列パフォーマンス

OpenMP* や自動並列化オプションは、インテル製マイクロプロセッサーおよび互換マイクロプロセッサーの両方で利用可能ですが、これらのオプションは非インテル製マイクロプロセッサーでは最適化の効果がなく、インテル製マイクロプロセッサーでのみ効果を発揮することがあります。

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/Qopenmp** | **-qopenmp** | OpenMP* プラグマ/宣言子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、ローカル配列がオートマチックになり、スタックサイズの増加が必要なことがあります。OpenMP* API の仕様については、http://www.openmp.org をご覧ください。 |
| **/Qparallel** | **-parallel** | 自動並列化は、インテル® Cilk™ Plus の配列表記による暗黙のループを含む、安全に並列実行できる構造化ループを検出し、自動的にループのマルチスレッド・コードを生成します。 |
| **/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>0 – 計算量にかかわらずループを並列化します。<br>100 – 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。/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* そして *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『ユーザー・リファレンス・ガイド』をご覧ください。 |
| **/Qguide[:*n*]** | **-guide[=*n*]** | ガイド付き自動並列化ループをベクトル化もしくは並列化する方法を提案するようにコンパイラーに指示します。この場合、オブジェクトや実行ファイルは生成されません。自動並列化のアドバイスは、**/Qparallel (-parallel)** オプションが指定されている場合にのみ生成されます。<br>*n* には 1 から 4 の値を指定でき、大きくなるほど詳細レベルのアドバイスが生成されます。レベル 4 は最も高度で積極的なアドバイスが提供されます。*n* が省略された場合、デフォルトは 4 です。 |
| **/Qopt- matmul[-]** | **-q[no-]opt-matmul** | このオプションは、コンパイラーが行列乗算ループの入れ子を識別して、パフォーマンス向上のため matmul ライブラリーを呼び出しに置き換えることを有効 [無効] にします。このオプションは、**/O3 (-O3)** と **/Qparallel (-parallel)** が指定され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)** 以上のオプションが指定されない限り、効果はありません。 |
| **/Qcilk-serialize** | **-cilk-serialize** | このオプションは、ヘッダーファイル cilk_stubs.h のインクルードを強制することで、コンパイラーにインテル® Cilk™ Plus のスレッド化キーワードを無視させ、シリアル実行コードを生成することを指示します。(C/C++ のみ)。詳細は、インテル® コンパイラーの『ユーザー・リファレンス・ガイド』の「インテル® Cilk™ Plus プログラムのビルド、実行、デバッグ」をご覧ください。 |
| **/Qcoarray: shared** | **-coarray= shared** | 共有メモリーシステム上で Fortran 2008 の Co-Array 機能を有効にします (Fortran のみ)。Co-Array オプションと詳細については、インテル® コンパイラーの『ユーザー・リファレンス・ガイド』をご覧ください。 |
| **/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) とのリンクを要求します。デフォルトではオフです。*name* に指定可能なキーワードは以下です:<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 |

# インテル® Cilk™ Plus を使用した並列パフォーマンス

| スレッド化のキーワード | 説明 (C/C++ のみ) |
|---|---|
| **cilk_spawn** | インテル® Cilk™ Plus のランタイムによって動的にスケジュールされ、呼び出し元と並列にスポーンされた関数を実行することを許可します (しかし必須ではありません)。 |
| **cilk_sync** | バリアを定義: スポーンされたすべての子が完了するまで関数は待機します。 |
| **cilk_for** | ループ反復の並列実行を許可 (必須ではありません) する for ループを定義します。 |

**レデューサー**は、合計の累積など、安全に並列実行できるリダクション操作を許可します。
例: **cilk::reducer< cilk::op_add<unsigned int>>** 符号なし整数への合計を行うレデューサーを定義。

**ホルダー: cilk::holder** テンプレート・クラスは、スレッドセーフで便利なタスク・ローカル・ストレージ方式を提供します。

**配列表記 (アレイ・ノーテーション):** 最適化レベル -O2 以上でベクトライザーによって SIMD 並列コードの生成を可能にする可読性がある明示的なデータ並列 C/C++ 言語拡張であり、依存関係がないことを明示します。
構文: **array[<lower bound>:<length>:<stride>]**
例: **bb[:][:]= 0** は、2 次元配列 bb 全体にゼロを代入します (サイズと型はコンパイラーが認識できなければなりません)。
**c[j:len] = sqrt(c[k:len:2])** は、c[k] から始まる 1 つ飛びの要素の len 個の平方根を求め、c[j] から始まる要素に格納します。これはベクトル化しても安全 ( j<k であれば) であることをコンパイラーに通知します。

**リダクション関数**が用意されています: **__sec_reduce_add(a[:])** など。これは、配列 a のすべての要素の sum を返します。

**SIMD 対応関数:** 関数がスカラーまたは SIMD モードで呼び出されても、その関数呼び出しを含むループを効率良くベクトル化することを可能にする言語拡張。コンパイラーは、1 つ以上のスカラー引数をベクトル操作で置き換えできる代替関数を生成します。

C/C++ 構文: **__declspec (vector(*clauses*)) func_name(arguments)** (または attribute)
Fortran 構文: **!DIR$ ATTRIBUTES VECTOR:(clauses) :: func_name**

オプションの clauses (節)には、*uniform*、*linear*、*mask*、*processor*、*vectorlength* そして *vectorlengthfor* を指定できます。

ベクトルバージョンの関数は、配列表記を使用して直接起動されるか、ループなどから間接的に起動されます。

```
a[:]= func_name(b[:],c[:],d,..);
for (int i=0; i<n; i++)       a[i] = func_name(b[i],c[i],d,...) ;
DO J=1,N;          A(J) = FUNC_NAME(B(J),C(J),D,...) ;ENDDO
```

同様の機能が、OpenMP* 4.0† の **DECLARE SIMD** でもサポートされます。SIMD 対応関数を含むループのベクトル化を確実にするため、SIMD プラグマや宣言子が必要になることがあります。

**SIMD プラグマ (C/C++) と宣言子 (Fortran) を使用した明示的なベクトル・プログラミング**
SIMD 命令を使用してベクトル化を行うようにコンパイラーに指示します。プログラマーは、正当性 (明示的にプライベート変数やリダクションを指定するなど) に関して責任があります。セマンティクスは、OpenMP* の #pragma omp parallel for (C/C++) と !$OMP PARALLEL DO (Fortran) 宣言と同じです。
コンパイラーは、OpenMP* 4.0† の SIMD 句で同様の機能をサポートします。

C/C++ 標準構文: **#pragma simd [節]**　　Fortran の構文: **!DIR$ SIMD [節]**

| 節 | 説明 |
|---|---|
| **private(*var1*,*var2*,...)** | ループの各反復間で競合を避けるためプライベートにする変数を指定します。 |
| **reduction(oper:*var1*,*var2*,...)** | 変数 *var1*、*var2*、... にオペレーター oper を使用したベクトル・リダクション操作を行うことをコンパイラーに指示します。 |
| **linear(var1:step1,...)** | スカラーループの各反復で、変数 vars1 は step1 ずつインクリメントされます。 |

そのほか、firstprivate、lastprivate、[no]assert、vectorlength、vectrolengthfor、vecremainder 節がサポートされます。
_Simd と _Reduction キーワードは、#pragma simd reduction(..) の代替手段を提供します。
詳細は、www.cilk.com とインテル® コンパイラーの『ユーザー・リファレンス・ガイド』をご覧ください。

## 推奨するプロセッサー固有の最適化オプション‡

これらのオプションの一部は、インテル製マイクロプロセッサーおよび互換マイクロプロセッサーで利用可能ですが、インテル製マイクロプロセッサーにおいてより多くの最適化が行われる場合があります。

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/Qx***target* | **-x***target* | *target* で指定された命令セットをサポートするすべてのインテル® プロセッサー向けの専用コードを生成します。実行可能ファイルは、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実行されません。*target* に指定可能な命令セット (上位から下位) : **CORE-AVX512、MIC-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 もしくは -m オプションでは有効とならないいくつかの最適化を有効にします。64 ビット OS X* では、**SSE3** と **SSE2** オプションはサポートされません。 |
| **/arch:***target* | **-m***target* | *target* で指定された命令セットをサポートするすべてのインテル® プロセッサーもしくは、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可能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>指定可能な target: **AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2、IA32**<br>**注:** IA-32 アーキテクチャーのみでサポートされます。OS X* 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)** オプションの **IA32、SSE2** もしくは **SSE3** の適切な値に相当します。このオプションは、非インテル製互換プロセッサーでは行われないインテル製マイクロプロセッサー向けの追加の最適化を有効にすることがあります。‡ |
| **Qax***target* | **-ax***target* | デフォルトのコードパスを生成しつつ、*target* で指定された命令セットをサポートするすべてのインテル製プロセッサーに特化したコードを生成することができます。<br>指定可能な target: **CORE-AVX512、MIC-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他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もしく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 (-axcore-avx2 -msse3** Linux* 向け) を使用します。<br>アプリケーションが実行時にインテル® プロセッサーで実行されているか自動検出し、最適なコードパスが選択されます。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。<br>**注:** 64 ビット OS X* では、**sse3** と **sse2** オプションはサポートされません。<br>このオプションは、非インテル製互換プロセッサーでは行われないインテル® マイクロプロセッサー向けの追加の最適化を有効にすることがあります。‡ |

最新のプロセッサー固有の最適化オプションについては、「インテル® SSE およびインテル® AVX 世代向けのインテル® コンパイラー・オプションとプロセッサー固有の最適化」の記事をご覧ください。
これらのオプションは、インテル® コンパイラーの『ユーザー・リファレンス・ガイド』で詳しく説明されています。

## オフロード向けコンパイル

| Windows* | Linux* | コメント |
|---|---|---|
| **/Qoffload[-]**<br>**/Qoffload[:*kywd*]** | **-q[no-]offload**<br>**-qoffload=*kywd*** | インテル® MIC アーキテクチャーやインテル® グラフィックス・テクノロジーへのオフロード向けのコンパイラーが持つ言語構文を制御します。***kywd*** には次の値が指定できます:<br>***none***: オフロード構文は無視され、すべてのコードはホスト上でのみ実行するようにコンパイルされます。**/Qoffload-** (**-qno-offload**) と等価です。<br>***mandatory:*** status 句が指定されておらずターゲットが利用できない場合、プログラムは失敗しオフロードはスキップされます。<br>***optional***: ターゲットが利用できない場合、すべてのコードはホストで実行されます。デフォルトは、**/Qoffload:mandatory** (**-qoffload=mandatory**) です。 |
| **/Qoffload-option,** ***target, tool, "option-list"*** | **-qoffload-option,** ***target, tool, "option-list"*** | ホスト用だけでなく、ターゲット向けのコンパイルに使用するオプションを指定します。***target*** は、***mic*** (インテル® MIC アーキテクチャー用) もしくは ***gfx*** (インテル® グラフィックス・テクノロジー用) です。***tool*** には、***compiler***、***ld***、***link*** または ***as*** を指定できます。 |
| **/Qoffload-attribute-target:*target-name*** | **-qoffload-attribute-target =*target-name*** | offload attribute target (***target-name***) と共に指定される各ファイルスコープ関数やデータ・オブジェクトのフラグ。***target-name*** には、***mic*** (インテル® MIC アーキテクチャー向け) もしくは ***gfx*** (インテル® グラフィックス・テクノロジー向け) を指定します。 |
| **/Qopt-report-phase:*offload*** | **-qopt-report-phase *offload*** | ホストとコプロセッサー、もしくはホストとプロセッサー・グラフィックス間でコピーされる変数のレポートをコンパイル時に生成します。 |
| **__INTEL_OFFLOAD** | **__INTEL_OFFLOAD** | ホスト上のオフロード・プログラミングで使用される事前定義マクロです。 |

## インテル® グラフィックス・テクノロジー向けのコンパイル§§

| Windows*<br>(32 ビットおよび 64 ビット) | Linux*<br>(64 ビットのみ) | コメント |
|---|---|---|
| **/Qgpu-arch:*arch*** | **-mgpu-arch=*arch*** | コンパイラーは、***arch*** で指定されるインテル® マイクロアーキテクチャー開発コード名 ***ivybridge*** もしくは ***haswell*** 上のプロセッサー・グラフィックス向けのネイティブ命令を生成します。デフォルトは、JIT エンジンで翻訳される仮想命令です。 |
| **__GFX__** | **__GFX__** | インテル® グラフィックス・テクノロジー向けのプログラミングで利用できる事前定義済みマクロです。 |

## インテル® グラフィックス・テクノロジー向けの環境変数§§

| 変数 | コメント |
|---|---|
| **GFX_CPU_BACKUP=1** | ターゲットが利用できない場合、オフロードコードはホストで実行されます。***0*** に設定されている場合、ターゲットが利用できないとアプリケーションは失敗します。 |
| **GFX_MAX_THREAD_COUNT** | ループの入れ子を並列化する際のターゲットスレッドの最大数を制御します。デフォルトは ***-1*** です (システムのデフォルト)。 |
| **GFX_OFFLOAD_TIMEOUT=*n*** | オフロードタスクは、***n*** 秒でタイムアウトします (デフォルトは、***n*** = ***60***)。このオプションを有効にするには、システム回復タイムアウトを無効にするか、増やす必要があるかもしれません。 |
| **GFX_SHOW_TIME=1** | 実行の最後にオフロードのタイミング情報を出力します。デフォルトは、***0*** (出力しない) です。 |
| **GFX_LOG_OFFLOAD=*n*** | オフロードのログを生成します。***n*** には、***0*** (ログなし) から ***3*** (最も詳しい) の詳細レベルを指定します。デフォルトは ***0*** です。 |

§§コンパイラーのインテル® グラフィックス・テクノロジーのサポートは、オペレーティング・システムのサポートに依存します。

詳細については、https://software.intel.com/articles/getting-started-with-compute-offload-to-intelr-graphics-technology にある導入ガイドや、インテル® コンパイラーのユーザー・リファレンス・ガイドをご覧ください。

## インテル® Xeon Phi™ コプロセッサー向けの最適化§

| Windows* | Linux* | コメント |
|---|---|---|
| **/Qmic** | **-mmic** | インテル® Xeon Phi™ コプロセッサーでネイティブに実行するアプリケーションをビルドします。(デフォルトではオフです)。 |
| **/Qopt-streaming-cache-evict:*n*** | **-qopt-streaming-cache-evict=*n*** | ストリーミング・ストアの後にキャッシュ・エビクション (追い出し) 命令を生成するかどうかを制御します。***n=0*** 追い出しなし; ***n=1*** L1 のみを追い出し; ***n=2*** L2 のみを追い出し (デフォルト); ***n=3*** L1 と L2 を追い出し。 |
| **/Qopt-assume-safe-padding** | **-qopt-assume-safe-padding** | ユーザープログラムによって、配列や動的に割り当てられたオブジェクトの終端を最大 64 バイト越えてコンパイラーが安全にアクセスできることを仮定します。ユーザーはパディングの責任を持ちます。デフォルトではオフです。 |
| **/Qopt-threads-per-core:*n*** | **-qopt-threads-per-core=*n*** | 物理コアあたりのスレッド数の最適化のためコンパイラーへヒントを与えます。***n*** は、***1***、***2***、***3*** または ***4*** です。 |
| **/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 ***4*** で指定します。最適化レベル /O2 (-O2) 以上で、デフォルトは ***n=3*** です。 |
| **/Qimf-domain-exclusion:*n*** | **-fimf-domain-exclusion=*n*** | IEEE 標準に準拠する必要がない、特殊なケースの数学関数を指定します。*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; ***4*** – ゼロ など。 |
| **/Qopt-gather-scatter-unroll** | **-qopt-gather-scatter-unroll** | 集約 (Gather) / 分散 (Scatter) ループの代替アンロールシーケンスを指定します。 |
| **/align: array64byte** | **-align array64byte** | アライメントされたロードとベクトル化の支援を可能にするため、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) |
| **__MIC__** | **__MIC__** | インテル® MIC アーキテクチャー向けのプログラミングで利用できる事前定義済みマクロです。 |

## インテル® Xeon Phi™ コプロセッサー向けの環境変数§

| 変数 | コメント |
|---|---|
| **OFFLOAD_REPORT=<*n*>** | オフロード・アプリケーションのランタイムレポートを生成します<br>***n***=1 ホストとコプロセッサー上の実行時間をレポートします<br>***n***=2 さらに、ホストとコプロセッサー間のデータ転送をレポートします<br>***n***=3 デバイスの初期化と個々の変数の転送を含む詳細をレポートします |
| **OFFLOAD_DEVICES=<*n1*,*n2*,...>** | ホスト上のプロセスが使用する物理コプロセッサーを ***n1***、***n2*** などに制限します (***n*** は 0 からの番号)。 |
| **MIC_STACKSIZE=<*n*>M** | オフロード・アプリケーション向けにコプロセッサー上の最大スタックサイズを設定します。この例では、***n*** はメガバイトです。 |
| **MIC_ENV_PREFIX=<name>** | オフロード・アプリケーション向けに、ホスト上の環境変数とコプロセッサーの環境変数を区別するためプリフィックスを指定します。<br>例えば、***name***=MIC であれば、MIC_OMP_NUM_THREADS がコプロセッサー上の OpenMP* スレッドの数を制御します。 |
| **MIC_USE_2MB_ BUFFERS=<*n*>M** | 実行時に ***n*** MB を越えるオフロードポインター変数は、2MB のラージページに割り当てられます。 |

§インテル® MIC アーキテクチャーとインテル® Xeon Phi™ コプロセッサーは、インテル® Parallel Studio XE に含まれるコンパイラーでサポートされますが、インテル® System Studio やインテル® INDE 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) とプロファイル・ガイドに基づく最適化 (PGO)

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/Qip** | **-ip** | 現在のソースファイル内の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。 |
| **/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ほか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。デフォルトでは、***n=0*** です (コンパイラーが判断します)。<br>注意: このオプションは、状況によってコンパイル時間とコードサイズを大幅に増やすことがあります。 |
| **/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) の数を指定します。デフォルトは 1 です。 |
| **/Ob2** | **-finline-functions**<br>**-finline-level=2** | コンパイラーが判断して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。このオプションは、**/O2** と **/O3** (**-O2** と **-O3**) で自動的に有効になります。<br>注意: ファイルサイズが大きな場合、このオプションはコンパイル時間とコードサイズを大幅に増やすことがあります。**/Ob0** (Linux* と OS X* 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。デフォルトで n は 100 です。これは、100% または 1 つのスケーリング要素を示します。 |
| **/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。 |
| **PROF_DIR** | **PROF_DIR** | プロファイル出力ファイルのディレクトリーを設定します (**/Qprof-use** や **-prof-use** で使用します)。 |
| **/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** | シリアルコードのそれぞれのループやループの入れ子のプロファイルを生成するようにインストルメント関数を追加します。詳細情報とプロファイルの表示方法については、インテル® コンパイラーの『ユーザー・リファレンス・ガイド』で、「関数のプロファイルやループの実行時間」をご覧ください。 |

## 浮動小数点演算オプション

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/fp:*name*** | **-fp-model *name*** | 特定の最適化を制限することで、浮動小数点演算の結果の一貫性を高めます。***name*** に指定可能な値は以下です:<br>**fast=[1\|2]** – 精度と一貫性を少し犠牲にして、より積極的な最適化を可能にします。(デフォルトは、**fast=1** です)。これは、非インテル製互換マイクロプロセッサーでは実行されませんが、インテル製マイクロプロセッサーで実行されるいくつかの最適化が含まれています。<br>**precise** – 浮動小数点コードで精度を優先する最適化のみを許可します。<br>**double/extended/source** – 中間結果が、倍精度、拡張精度、ソースの精度で計算されます。オーバーライドされない限り **precise** が採用されます。インテル® Fortran コンパイラーでは、**double** と **extended** オプションはサポートされません。<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ますが、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>**推奨事項:** 浮動小数点の一貫性と再現性が必要とされる多くの状況では、**/fp:precise /fp:source (-fp-model precise -fp- model source)** が推奨されます。 |
| **/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。デフォルトでは、/fp:precise (-fp-model precise) が指定されない限り有効です。 |
| **/Qftz[-]** | **-ftz[-]** | main プログラムや dll の main がこのオプションでコンパイルされた場合、デノーマル (インテル® SSE やインテル® AVX 命令の結果) はプログラム全体で実行時にゼロへフラッシュされます (dll)。デフォルトは、**/Od (-O0)** を除いてオンです。 |
| **/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 OFF です (コンパイラーはデフォルトのヒューリスティックに従います)。指定可能な ***name*** の値は、***high***、***medium*** そして ***low*** です。精度を低下させると、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。算術ライブラリーの多くのルーチンは、互換マイクロプロセッサーよりもインテル製マイクロプロセッサー向けにより高度に最適化されます。 |
| **/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インテル製プロセッサーと、非インテル製互換プロセッサーで一貫した結果を生成するようにします。これによってランタイム・パフォーマンスが向上します。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 [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。 |

こちらをご覧ください: http://software.intel.com/articles/consistency-of-floating-point-results-using-the-intel-compiler

## 細かなチューニング (すべてのプロセッサー向け)

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/Qunroll[n]** | **-unroll[n]** | ループアンロール回数の上限を設定します。**/Qunroll0 (-unroll0)** 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*** は、***0*** (プリフェッチなし) から ***4*** (最大限のプリフェッチ) の任意の値です。**/O3 (-O3)** が指定される場合、***n*** のデフォルトは ***2*** です。<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-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** | ストリーミング・ストアの生成を有効/無効にします。***mode*** に指定できる値は以下です:<br>**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<br>**Never** - ストリーミング・ストアの生成を無効にする<br>**auto** - ストリーミング・ストアの生成に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する |
| **/Qrestrict[-]** | **-[no-]restrict** | **restrict** キーワードによるポーンターの一義化を有効 [無効] にします。デフォルトではオフです。(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前提にコンパイルします。デフォルトではオフです。 |
| **/Ow** | **-fno-fnalias** | 関数内でのエイリアシングがない事を想定します。デフォルトではオフです。 |
| **/Qalias-args[-]** | **-fargument-[no]alias** | 関数の引数がアライメントされている [アライメントされていない] ことを暗示。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 OS X* では有効。 |
| **/Qopt-class-analysis[-]** | **-q[no-]opt-class-analysis** | コンパイル時に C++ 仮想関数の呼び出しを解析し解決するのに、C++ クラス階層情報を使用しません。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、アプリケーションの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 OS X*) オプションが指定された場合、C++ の最適化を高めるためオンになります。(C++ のみ) |
| | **-f[no-]exceptions** | **-fexceptions** – C++ のデフォルトです。例外処理テーブルの生成を有効にします。<br>**-fno-exceptions** – C と Fortran のデフォルトです。コードサイズを最小化します。C++ では、例外指定は解析されますが無視されます。関数の呼び出しチェーンにある関数が **-fno-exceptions** でコンパイルされている場合、構造化例外処理 (try ブロックや throw 文) を使用した場合、エラーが発生します。 |
| **/Qvec-threshold:n** | **-vec-threshold=n** | パフォーマンス向上の効果がある可能性に基づいて、ループの自動ベクトル化の閾値 **n** を設定します。**0** ≤ ***n*** ≤ **100**、デフォルトは ***n*=100**。<br>**0** – 計算量にかかわらずループをベクトル化します。<br>**100** – パフォーマンス上の利点が確実である場合にのみ、ループをベクトル化します。 |
| **/Qvec[-]** | **-[no-]vec** | 自動ベクトル化を有効/無効にします。デフォルトは、**/O2 (-O2)** で有効です。 |

## デバッグオプション

| Windows* | Linux*、OS X* | コメント |
|---|---|---|
| **/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、一般的な開発環境のデバッガー向けにデバッグ情報を生成します。最適化オプション **/O2 (-O2)** 以上と **/Od (-O0)** を有効にしてコンパイルすると有効になります。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**<br>**-debug-info** | **-debug**<br>**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の代わりに、呼び出される関数のソースに関連付けられます。<br>**-O2** が指定されないと、**/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。**/debug:full** (**-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*) と実行可能ファイル (Linux*) 中に、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。デフォルトではオフです。 |
| **/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するようにコンパイラーに指示します。最適化されたコードで利用されます。(Fortran アプリケーションのみ) |

## その他の情報

インテル® コンパイラーの『ユーザー・リファレンス・ガイド』:
https://software.intel.com/compiler_15.0_ug_c および https://software.intel.com/compiler_15.0_ug_f
『ユーザー・リファレンス・ガイド』の日本語版は、 下記ページからインテル® Parallel Studio XE の無償評価版にご登録いただくことで、ご覧いただけます。
https://www.xlsoft.com/jp/products/download/intelj.html

インテル® MIC アーキテクチャー向けの最適化情報:
https://software.intel.com/articles/advanced-optimizations-for-intel-mic-architecture
http://software.intel.com/mic-developer
https://software.intel.com/articles/programming-and-compiling-for-intel-many-integrated-core-architecture

製品と購入情報については、エクセルソフト株式会社のインテル® ソフトウェア開発製品のサイトをご覧ください:
http://www.xlsoft.com/jp/products/category_intel.html

### ‡最適化に関する注意事項

インテル® コンパイラーは、互換マイクロプロセッサー向けには、インテル製マイクロプロセッサー向けと同等レベルの最適化が行われない可能性があります。これには、インテル® SSE2、インテル® SSE3、インテル® SSSE3 命令セットに関連する最適化およびその他の最適化が含まれます。インテルでは、インテル製ではないマイクロプロセッサーに対して、最適化の提供、機能、効果を保証していません。本製品のマイクロプロセッサー固有の最適化は、インテル製マイクロプロセッサーでの使用を目的としています。インテル® マイクロアーキテクチャーに非固有の特定の最適化は、インテル製マイクロプロセッサー向けに予約されています。この注意事項の適用対象である特定の命令セットに関する詳細は、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。

改訂 #20110804