



インテル® Composer XE 2011 最適化クイック・リファレンス・ガイド

IA-32 プロセッサおよびインテル® 64 プロセッサ

XE

インテル® ソフトウェア開発製品

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。本バージョンのコンパイラーでは、すべての最適化レベルは、デフォルトで SSE2 命令セットのサポートを想定しています。SSE2 命令がサポートされないインテル® Pentium® III プロセッサなど、以前の IA-32 プロセッサで実行するには、**/arch:IA32 (Windows*)** または **-mia32 (Linux*)** を利用する必要があります。

1. 一般的な最適化オプション (Windows では **/O1**、**/O2**、**/O3**。Linux および Mac OS* 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。これらのオプションは、インテル® マイクロプロセッサおよびインテル製以外のマイクロプロセッサの両方で利用可能ですが、インテル® マイクロプロセッサ向けのほうが、より多くの最適化が適用される場合があります。
2. IA-32 およびインテル® 64 ベースのシステム向けに、プロセッサ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、インテル® Core™ i7 プロセッサなどのインテル® Core™ プロセッサ・ファミリー向けには **/QxSSE4.2 (-xsse4.2)** オプション、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。このオプションは、インテル® マイクロプロセッサおよびインテル製以外のマイクロプロセッサの両方で利用可能ですが、インテル® マイクロプロセッサ向けのほうが、より多くの最適化が適用される場合があります。特定のプロセッサや命令セット向けに最適化するオプションのリストについては、「推奨するプロセッサ専用の最適化オプション†」の表を参照してください。
3. プロシージャ間最適化 (IPO) 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. 新しいガイド付き自動並列化 (GAP) 機能の **/Qguide (-guide)**、インテル® Cilk™ Plus C++ による言語拡張、並列パフォーマンス・オプションの **/Qparallel (-parallel)** や **/Qopenmp (-openmp)** を使用したり、あるいは製品に含まれるインテル® パフォーマンス・ライブラリーを使って、マルチスレッド、マルチコア、マルチプロセッサ・システムでベクトル化や並列実行向けにアプリケーションを最適化します。このような最適化処理は、インテル® マイクロプロセッサおよびインテル製以外のマイクロプロセッサの両方で利用可能ですが、インテル® マイクロプロセッサのほうが、より高いパフォーマンスが得られる場合があります。
5. インテル® VTune™ Amplifier XE 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」を識別します。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。インテル® Inspector XE は、メモリーおよびスレッド化エラーの診断を行い、開発プロセスを効率化して、マルチスレッド・アプリケーションの開発期間を短縮します。両製品ともインテル製以外のマイクロプロセッサ上では使用できません。

詳細は、製品のドキュメントをご覧ください。

一般的な最適化オプション

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。

Windows	Linux Mac OS X	説明
/Od	-O0	最適化は行われません。 このオプションは、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。アプリケーションが正常に動作することを確認した後は、より高度なオプションを使用してください。
/O1	-O1	サイズを考慮した最適化を行います。 オブジェクトのサイズを増やす傾向がある最適化を省略します。多くの場合、最小限のサイズで最適化されたコードが作成されます。 コードサイズが大きいため、メモリーページングが問題になる巨大なサーバー / データベース・アプリケーションにおいて、このオプションは効果的です。
/O2	-O2	最速化します (デフォルト設定)。 ベクトル化と実行速度を改善する多くの最適化を有効にします。多くの場合、/O1 (-O1) よりも速いコードを作成します。
/O3	-O3	/O2 (-O2) の最適化に加えて、スカラー置換、ループアンロール、分岐を除去するコード反復、効率的にキャッシュを使用するループ・ブロッキング、データ・プリフェッチ機能など、強力なループ最適化およびメモリーアクセス最適化を行います。 /O3 (-O3) オプションは、特に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。この最適化は、場合によって /O2 (-O2) の最適化よりもアプリケーションの実行が遅くなることがあります。
/fast	-fast	プログラム全体の速度を最大限にします。次のオプションを設定します。 Mac OS X: -ipo、-mdynamic-no-pic、-O3、-no-prec-div、-xHost Windows: /O3、/Qipo、/Qprec-div-、/QxHost Linux: -ipo、-O3、-no-prec-div、-static、-xHost
/Qopt-report[:n]	-opt-report [n]	最適化レポートを作成し、stderr 出力します。n には、0 (レポートなし) から 3 (最大限の情報) の範囲で詳細レベルを指定します。デフォルトは 2 です。
/Qopt-report-phase:name	-opt-report-phase=name	最適化フェーズごとのレポートを生成します。name には次に示すフェーズを指定します。 all - 全フェーズの最適化レポート (デフォルト) ipo_inl - プロシージャー間の最適化におけるインライン展開レポート hlo - 高レベル・オブティマイザー・レポート (ループおよびメモリーの最適化を含む) hpo - ハイパフォーマンス最適化レポート (ベクトライザーおよびパラレイザーを含む) pgo - プロファイルに基づく最適化レポート
/Qopt-report-help	-opt-report-help	上記の /Qopt-report-phase (-opt-report-phase) で利用可能なすべてのフェーズオプションを表示します。コンパイルは行いません。
/Qopt-report-routine:string	-opt-report-routine=string	string で指定された文字列を含む関数またはサブルーチンに関するレポートを生成します。デフォルトでは、すべての関数およびサブルーチンのレポートが生成されます。
/Qdiag-enable:sc[n]	-diag-enable sc[n]	スタティック・セキュリティー解析を有効にします。n は、レポートする診断レベル (1= クリティカル・エラーのみ、2= すべてのエラー、3= すべてのエラーと警告) を指定します。解析結果の表示には、インテル® Inspector XE が必要です。

並列パフォーマンス

OpenMP* や自動並列化を使用するオプションは、インテル® マイクロプロセッサおよびインテル製以外のマイクロプロセッサの両方で利用可能ですが、一部の最適化は、インテル® マイクロプロセッサ向けにのみ適用される場合があります。

Windows	Linux Mac OS X	説明
/Qopenmp	-openmp	OpenMP* 指示句がある場合、その指示によるマルチスレッド・コードが生成されます。スタックのサイズを増やさなければならないことがあります。
/Qparallel	-parallel	自動パラライザは、安全に並列実行可能な構造のループ（インテル® Cilk™ Plus のアレイ・ノテーションによって暗黙的に定義されるループを含む）を検出し、そのループに対するマルチスレッド・コードを自動生成します。
/Qpar-report[:n]	-par-report[n]	自動並列化の診断レベルを制御します。n には、0（レポートなし）から 3（最大限の情報）の範囲で詳細レベルを指定します。デフォルトは 0 です。
/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 n=100。デフォルト：n=100）。 0 - 計算量にかかわらず並列化を行います。 100 - パフォーマンス向上が確実に見込める場合にのみ、ループを並列化します。 /Qparallel (-parallel) とともに使用する必要があります。
/Qguide[:n]	-guide[=n]	ガイド付き自動並列化。ループのベクトル化や自動並列化を促進するアドバイスを生成します。オブジェクト・ファイル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、-parallel（Linux または Mac OS X）または /Qparallel（Windows）オプションが同時に指定された場合に提供されます。 n には 1 から 4 までの値で、アドバイスのレベルを指定します。4 は最も詳細なレベルのアドバイスです。n が省略された場合、デフォルトは 4 です。
/Qopt-matmul[-]	[-no-]opt-matmul	コンパイラーが生成する行列乗算（matmul）ライブラリー呼び出しを有効 [無効] にします。行列乗算におけるループの入れ子（該当する場合）を特定し、ループを matmul ライブラリー呼び出しに置換して、パフォーマンスを向上します。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。また、このオプションは、/O2 (-O2) オプション以上が指定されない限り、効果はありません。
/Qcilk-serialize	-cilk-serialize	インテル® Cilk™ Plus 言語拡張で並列化されたコードをシリアル化します。つまり、コンパイラーは、シリアルプログラムを生成します。このオプションは、cilk_stub.h ヘッダーファイルのインクルードを強制します。このヘッダーファイルには、インテル® Cilk™ Plus キーワードを無効にするプリプロセッサ・マクロが含まれています。詳細は、『インテル® コンパイラー・ユーザー・リファレンス・ガイド』の「インテル® Cilk™ Plus の使用」を参照してください。
/Qcoarray:shared	-coarray=shared	共有メモリーシステムにおいて Fortran 2008 標準の Co-Array を有効にします（Fortran のみ）。その他のオプション、詳細情報については、『インテル® コンパイラー・ユーザー・リファレンス・ガイド』を参照してください。このオプションは、インテル® マイクロプロセッサおよびインテル製以外のマイクロプロセッサの両方で利用可能です。一部の最適化は、インテル® マイクロプロセッサのみに適用される場合があります。

推奨するプロセッサ専用の最適化オプション[†]

Windows	Linux Mac OS X	説明
/Qxtarget	-xtarget	<p>target で指定された命令セットをサポートするインテル® プロセッサ向けの専用コードを生成します。実行ファイルは、インテル製以外のプロセッサ、またはより低い命令セットのみをサポートするインテル® プロセッサでは実行できません。target の値は次のとおりです (命令セットの高い順) :</p> <p>AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2</p> <p>注: Mac OS X では、SSE3 および SSE2 オプションはサポートされません。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。</p>
/arch:target	-mtarget	<p>target で指定された命令セットをサポートするインテル® プロセッサまたはインテル製以外の互換プロセッサ向けの専用コードを生成します。実行ファイルを指定された命令セットをサポートしないインテル® プロセッサまたはインテル製以外の互換プロセッサで実行させると、ランタイムエラーを引き起こす可能性があります。</p> <p>target の値は次のとおりです : SSE4.1、SSSE3、SSE3、SSE2、IA32</p> <p>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 アーキテクチャでのみサポートされます。Mac OS X では、SSE3、SSE2、IA32 オプションはサポートされません。</p>
/QxHOST	-xhost	<p>コンパイルを行うホスト・プロセッサで利用可能な最上位の命令セットを利用したコードを生成します。インテル® プロセッサ上では、これは、最適な /Qx (-x) オプションに対応します。インテル製以外の互換プロセッサ上では、/arch (-m) オプションの IA32、SSE2 または SSE3 に対応します。このオプションによる一部の最適化は、インテル® マイクロプロセッサ向けにのみ適用される場合があります。[†]</p>
/Qaxtarget	-axtarget	<p>target で指定された命令セットをサポートするインテル® プロセッサ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 の値は次のとおりです : AVX、SSE4.2、SSE4.1、SSSE3、SSE3、SSE2</p> <p>カンマで区切って複数の値を指定し、単一の実行ファイルで複数のプロセッサ向けのチューニングを行うことができます。例 : /QaxSSE4.2,SSE3。デフォルトのコードは、SSE2 以上をサポートするインテル® プロセッサまたはインテル製以外の互換プロセッサで動作しますが、/Qx (-x) または /arch (-m) オプションを利用して変更することができます。</p> <p>例えば、インテル® Core™ プロセッサ・ファミリー向けに最適化された専用コードと、SSE3 以上をサポートするインテル® プロセッサまたはインテル製以外の互換プロセッサ向けのコードを生成するには、/QaxSSE4.2 /arch:SSE3 (Linux では -axsse4.2 -msse3) を指定します。</p> <p>アプリケーションはインテル® プロセッサ上で実行されているかどうかランタイムで自動検出し、インテル® プロセッサ上で実行されている場合は、最適なコードを選択します。インテル® プロセッサが検出されなかった場合は、デフォルトコードを選択します。</p> <p>注: Mac OS X では、sse3 および sse2 オプションはサポートされません。このオプションによる一部の最適化は、インテル® マイクロプロセッサでのみ適用される場合があります。[†]</p>

オンライン記事「Intel® compiler options for SSE generation and processor-specific optimizations (SSE 命令の生成とプロセッサ固有の最適化のためのインテル® コンパイラ・オプション)」(英語)で、プロセッサ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の詳細は、『インテル® コンパイラ・ユーザー・リファレンス・ガイド』で説明されています。

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 Mac OS X	説明
/Qip	-ip	単一ファイルの最適化を行います。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プロシージャー間の最適化です。
/Qipo[n]	-ipo[n]	インライン展開およびその他のプロシージャー間の最適化を複数のソースファイルに対して行います。オプションの <i>n</i> 引数には、コンパイル時に生成するオブジェクト・ファイルの最大数を指定します。デフォルトの <i>n</i> は 0 です (コンパイラーが最適なファイル数を自動選択)。 警告：条件によってはコンパイル時間とコードサイズが大幅に増加する場合があります。
/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、同時に実行するジョブ数を指定します。デフォルトは 1 ジョブです。
/Ob2	-finline-functions -finline-level=2	インライン展開をコンパイラーの判断に任せます。このオプションは、/O2 および /O3 (-O2 および -O3) を指定すると有効になります。 警告：大きなファイルでは、コンパイル時間とコードサイズが大幅に増加する場合があります。/Ob0 (Linux および Mac OS X では -fno-inline-functions) を指定すると無効になります。
/Qinline-factor= <i>n</i>	-finline-factor= <i>n</i>	インライン化される関数の合計サイズと最大サイズを指定します。 <i>n</i> のデフォルト値は 100 (100%、スケール係数 1) です。
/Qprof-gen	-prof-gen	プロファイル最適化で参照する動的なパフォーマンス・データを生成するため、プログラムにインストルメント・コードを埋め込みます。
/Qprof-use	-prof-use	最適化中に prof-gen オプションで生成した実行ファイルのプロファイリング情報を参照します。
/Qprof-dir <i>dir</i>	-prof-dir <i>dir</i>	プロファイル出力ファイル *.dyn および *.dpi を格納するディレクトリーを指定します。
/Qprofile-functions	-profile-functions	関数をインストルメントし、各関数で消費された実行時間のプロファイルを生成します。
/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・ユーザー・リファレンス・ガイド』の「関数またはループの実行時間のプロファイル」を参照してください。

浮動小数点演算オプション

Windows	Linux Mac OS X	説明
/fp:name	-fp-model name	<p>浮動小数点演算における演算モデルを制御します。特定の最適化を制限して浮動小数点結果の一貫性を強化します。name の値は次のとおりです。</p> <p>fast=[1]2 - 精度や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）。一部の最適化は、インテル® マイクロプロセッサのみに適用される場合があります。</p> <p>precise - 精度に影響しない最適化のみ有効にします。</p> <p>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。変更されない限り、precise も適用されます。インテル® Fortran コンパイラーでは、double オプションおよび extended オプションは利用できません。</p> <p>except - 浮動小数点例外セマンティクスを使用します。</p> <p>strict - precise オプションと except オプションの両方を有効にし、デフォルトの浮動小数点環境を想定しません。</p> <p>推奨：浮動小数点演算の一貫性や再現性が重要な状況では、/fp:precise /fp:source (-fp-model precise -fp-model source) を推奨します。</p>
/Qftz[-]	-ftz[-]	<p>メインプログラムまたはメインの DLL をこのオプションでコンパイルすると、プログラムや DLL 全体で SSE 命令によるデノーマル結果をゼロにフラッシュします。/Od (-O0) を除き、デフォルトはオンです。</p>
/Qimf-precision:name	-fimf-precision:name	<p>このオプションは、算術ライブラリー関数の精度を制御します。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）。name の値は、high、medium、low のいずれかです。精度を下げるとパフォーマンスの向上につながります。また、その逆についても同じことが言えます。算術ライブラリーのルーチンの多くは、インテル製マイクロプロセッサ向けにより高度に最適化されています。</p>
/Qimf-arch-consistency:true	-fimf-arch-consistency=true	<p>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® プロセッサまたはインテル製以外の互換プロセッサにおいて一貫した結果を生成します。ランタイム・パフォーマンスが低下することがあります。デフォルトは“false”（オフ）です。</p>
/Qprec-div[-]	[-no-]prec-div	<p>浮動小数点除算の精度を上げます [下げます]。精度を上げるとパフォーマンスに多少影響します。</p>
/Qprec-sqrt[-]	[-no-]prec-sqrt	<p>平方根計算の精度を上げます [下げます]。精度を上げるとパフォーマンスに多少影響します。</p>

きめ細かなチューニング (すべてのプロセッサ)

Windows	Linux Mac OS X	説明
/Qunroll[n]	-unroll[n]	ループをアンロールする最大回数を設定します。 /Qunroll0 (-unroll0) はループアンロールを無効にします。デフォルトは、 /Qunroll (-unroll) で、コンパイラーの判断に任せます。
/Qopt-prefetch:n	-opt-prefetch=n	ソフトウェア・プリフェッチのレベルを制御します。 n には 0 (プリフェッチなし) から 4 (強力なプリフェッチ) までの値です。高レベルの最適化が有効な場合、デフォルトは 2 です。 警告 : 過度のプリフェッチはリソースの競合を引き起こし、パフォーマンスの低下につながります。
/Qopt-block-factor:n	-opt-block-factor=n	ループのブロッキング最適化におけるブロッキング係数を指定します。ブロッキング係数は、1 ブロックのループ反復数です。ループ・ブロッキングは /O3 (-O3) で有効になり、キャッシュ中のデータ利用率を高めるように設計されています。
/Qopt-streaming-stores:mode	-opt-streaming-stores mode	ストリーミング・ストアを生成するかどうかを指定します。 mode の値は次のとおりです。 always : アプリケーションがメモリー上のデータ再利用がほとんどないことを仮定して、キャッシュをバイパスするストリーミング・ストアの生成をコンパイラーに指示します。 never : ストリーミング・ストアの生成を無効にします。 auto : ストリーミング・ストアの生成をコンパイラーの判断に任せます。
/Qrestrict[-]	-[no]restrict	restrict キーワードとともに指定すると、ポインターの一義化が有効 [無効] になります。デフォルトではオフです (C/C++)。
/Oa	-fno-alias	プログラムでエイリアシングしないことを前提に最適化します。デフォルトではオフです。
/Ow	-fno-fnalias	関数内でエイリアシングしないことを前提に最適化します。デフォルトではオフです。
/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。デフォルトではオンです (C/C++)。 -fargument-noalias は、コンパイラーが関数の配列引数を含むループをベクトル化するのに役立ちます。
/Qopt-class-analysis[-]	-[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トなど) 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。ただし、 /Qipo (Windows) または -ipo (Linux および Mac OS X) コンパイラー・オプションが指定されるとデフォルトでオンになり、C++ 最適化を有効にします (C++ のみ)。
	-f[no-]exceptions	-f-exceptions : 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。 -fno-exceptions : C、Fortran のデフォルトです。コードサイズが小さくなります。C++ では、例外指定は解析されますが、無視されます。構造化例外処理 (try ブロックや throw 文) を使用していると、一連の呼び出し中の関数が -fno-exceptions でコンパイルされている場合、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可能性に基づいて、ループのベクトル化のしきい値 n を設定します (n=0 から n=100 。デフォルト: n=100)。 0 - 計算量にかかわらずベクトル化されます。 100 - パフォーマンス向上が確実に見込める場合にのみループはベクトル化されます。
/Qvec-report:n	-vec-report n	ベクトライザーの診断レベルを制御します。 n には、0 (レポートなし) から 3 (最大限の情報) の範囲で詳細レベルを指定します。デフォルトは 0 です。

デバッグオプション

Windows	Linux Mac OS X	説明
/Zi	-g	一般的な開発環境のデバッガーで利用できるデバッグ情報を生成します。このオプションは、/O2 (-O2) (または別の O オプション) が指定されない限り、/O2 (-O2) をオフにして /Od (-O0) をデフォルトにします。
-debug[:keyword]	-debug [keyword]	デバッグ情報のレベルを keyword で指定します。 keyword には次のキーワードを指定します。 none - デバッグ情報を生成しません (デフォルト)。 full (または all) - 最適化されていないコードのシンボル情報を生成します。 -g (/Zi) またはキーワードなしの -debug (/debug) と同じです。 extended - 最適化されたコードの拡張シンボル情報を生成します (Linux および Mac OS X のみ)。デバッグシンボルは通常、オブジェクト・モジュールのサイズを増やし、最適化されたコードのパフォーマンスを多少低下させることがあります。 -debug full も含まれます。 parallel - スレッドコードをデバッグするための追加情報およびインストルメンテーションを生成します。 -debug full は含まれません。

†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向けには、インテル製マイクロプロセッサ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® ストリーミング SIMD 拡張命令 2 (インテル® SSE2)、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3)、ストリーミング SIMD 拡張命令 3 補足命令 (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固有の最適化は、インテル製マイクロプロセッサ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の詳細は、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

製品情報および購入情報は、インテル® ソフトウェア
開発製品 Web サイトを参照してください。

<http://www.intel.com/cd/software/products/ijkk/jpn/compilers/>

Intel、インテル、Intel ロゴ、Cilk、Intel Core、Pentium、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 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。

© 2012 Intel Corporation. 無断での引用、転載を禁じます。

JPN/1205/8K/XL/SSG/KS

325020-001JA

