

インテル® C++ コンパイラー 11.1 Windows* OS 版 プロフェッショナル・エディション 入門ガイド

本ガイドでは、インテル® C++ コンパイラー・プロフェッショナル・エディションを起動する方法、プロジェクトでインテル® ライブラリーを使用する方法、インテル® Parallel Debugger Extension を使用してコードをデバッグする方法を説明します。

本ガイドは、開発アプリケーションのパフォーマンス向上目標に最も適したスレッド化手法の選択に役立つ一般的なシナリオを紹介します。また、より多くのリファレンス情報を入手するための手段も提供します。

目次

1	インテル® C++ コンパイラー・プロフェッショナル・エディションの起動.....	1
2	インテル® パフォーマンス・ライブラリーの使用.....	3
3	スレッド化手法の選択.....	5
4	インテル® Parallel Debugger Extension によるコードのデバッグ.....	6
5	ユーザー・リファレンス・ドキュメント.....	9
	著作権と商標について.....	11


1 インテル® C++ コンパイラー・プロフェッショナル・エディションの起動

インテル® C++ コンパイラー 11.1 Windows 版プロフェッショナル・エディションは、以下のバージョンの Microsoft* Visual Studio* IDE に統合します。

- Microsoft Visual Studio 2008
- Microsoft Visual Studio 2005
- Microsoft Visual Studio .NET 2003



Microsoft Visual Studio IDE からインテル® コンパイラー・プロフェッショナル・エディションを起動するには、次の手順に従います。

1. Microsoft Visual Studio を起動します。
2. **[ソリューション エクスプローラ]** ペインで Visual Studio ソリューションを開くか、または作成します。
3. **[プロジェクト]** メニューまたはプロジェクトのコンテキスト・メニューから、**[Intel C++ Compiler Pro (インテル(R) C++ コンパイラー・プロフェッショナル)]** > **[Use Intel C++ (インテル(R) C++ を使用)]** を選択します。
4. **[Confirmation (確認)]** ダイアログボックスで、**[Yes (はい)]** をクリックします。これにより、ソリューションでインテル® C++ コンパイラーが使用されるようになります (ツールバーの  をクリックして設定することもできます)。
5. Visual Studio の **[ビルド]** メニューから **[ソリューションのリビルド]** を選択します。

コンパイル結果が **[出力]** ウィンドウに表示されます。

アプリケーションのビルドについての詳細は、インテル® C++ コンパイラー・プロフェッショナル・エディションのオンラインヘルプを参照してください。「[ユーザー・リファレンス・ドキュメント](#)」で追加情報やヘルプが得られます。

コマンドラインからのコンパイラーの起動

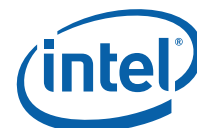
Windows のコマンドラインからコンパイラーを起動するには、Windows のプログラムメニューから次のように選択します。

[Intel(R) Software Development Tools (インテル(R) ソフトウェア開発ツール)] >
[Intel(R) C++ Compiler Professional 11.1.xxx (インテル(R) C++ コンパイラー 11.1.xxx プロフェッショナル)] >
[Build Environment for <プラットフォーム名> (<プラットフォーム名> 用ビルド環境)]
を選択します。

コマンドプロンプトが表示され、インテル® コンパイラーの環境変数が設定されます。C++ 言語ソースファイル (例: my_source_file.cpp) をコンパイルするには、次のコマンドを実行します。

```
icl my_source_file.cpp
```

コンパイルが正常に終了すると、実行可能ファイル my_source_file.exe が現在のディレクトリーに作成されます。

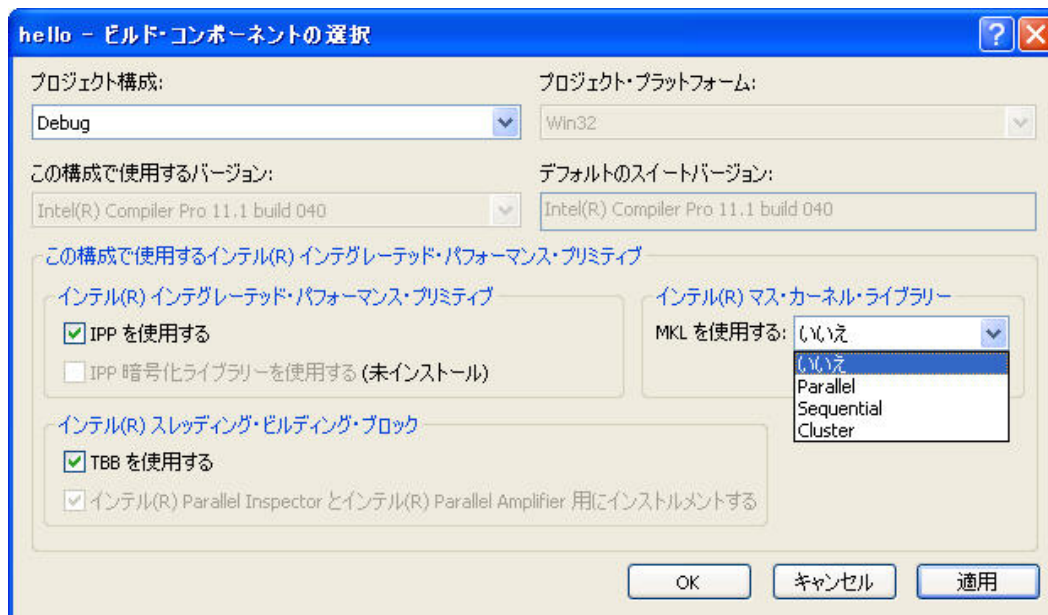


2 インテル® パフォーマンス・ライブラリーの 使用

インテル® C++ プロジェクトでインテル® 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)、インテル® スレディング・ビルディング・ブロック (インテル® TBB)、インテル® マス・カーネル・ライブラリー (インテル® MKL) を使用するには、次の操作を行います。

1. [ソリューション エクスプローラ] でプロジェクトを開きます。
2. メニューバーで [プロジェクト] > [Intel C++ Compiler Pro (インテル(R) C++ コンパイラ・プロフェッショナル)] > [Select Build Components (ビルド・コンポーネントの選択)] をクリックします (またはツールバーの をクリックします)。

[Select Build Components (ビルド・コンポーネントの選択)] ダイアログボックスが開きます。



ダイアログボックスで、インテル® IPP を有効にします。



操作	結果
[Use IPP (IPP を使用する)] チェックボックスをオンにする	インテル® IPP の一般的なライブラリーのみが有効になります。
[Use IPP Cryptography (IPP 暗号化ライブラリーを使用する)] チェックボックスをオンにする	インテル® IPP の暗号化ライブラリーとインテル® IPP の一般的なライブラリーの両方が有効になります。
[Use IPP (IPP を使用する)] チェックボックスをオフにする	インテル® IPP ライブラリーが無効になります。
[Use TBB (TBB を使用する)] チェックボックスをオンにする	インテル® TBB ライブラリーが有効になります。
[Use TBB (TBB を使用する)] チェックボックスをオフにする	インテル® TBB ライブラリーが無効になります。
[Use MKL (MKL を使用する)] チェックボックスをオンにする	インテル® MKL ライブラリーが有効になります。
[Parallel (並列)] を選択する	並列のインテル® MKL ライブラリーを使用します。
[Sequential (シーケンシャル)] を選択する	シーケンシャルなインテル® MKL ライブラリーを使用します。
[Cluster (クラスター)] を選択する	クラスターのインテル® MKL ライブラリーを使用します。
[No (いいえ)] を選択する	インテル® MKL ライブラリーがすべて無効になります。

[Select Build Components (ビルド・コンポーネントの選択)] ダイアログボックスで、[OK] または [Apply (適用)] をクリックすると、選択されたアーキテクチャーに適切なインテル® IPP、インテル® TBB、インテル® MKL の include パスと link ライブラリーがプロジェクトのプロパティー設定に追加されます。

ターゲット・プラットフォームが IA-64 またはインテル® 64 の場合、[Use ILP64 interfaces (ILP64 インターフェイスを使用する)] ボックスが表示されます。オンの場合、対応する ilp MKL ライブラリーがリンカーのコマンドラインに追加され、MKL_ILP64 プリプロセッサ定義がコンパイラーのコマンドラインに追加されます。オフの場合は、lp MKL ライブラリーがデフォルトで使用されます。

注: LP 版と ILP64 版のライブラリーの違いについては、『インテル® MKL ユーザーズ・ガイド』を参照してください。



3 スレッド化手法の選択

並列化手法	利点	考慮事項
インテル® IPP	高いパフォーマンス、特定のマルチメディア・アプリケーション・ドメインでスケーラブル、コードの並列性を公開しない。	暗号化、画像、音声、ビデオ、圧縮など。マルチメディアに最適です。
自動並列化	自動の並列化。コードの明示的な並列構文を公開しない。	適格なループに限定されます。
新しい並列拡張	ラピッド・プロトタイピングに最適。簡単な構文、OpenMP 3.0 ランタイムにマップ。	OpenMP やインテル® TBB のような shared/private 変数の制御はありません。
OpenMP 3.0	良く知られた仕様。効率的なランタイムで、スケーラブル。task 構造をサポートする OpenMP 3.0 により、関数レベルの並列化を支援。インテルの OpenMP サポートは Microsoft 実装との互換性を確保。	一部の C++ 開発者にとってはプラグマの使用は一般的ではありません。C++ オブジェクト/データに対しては難しい可能性があります。Microsoft Visual Studio 2008 は、OpenMP 2.5 をサポートしています。
インテル® TBB	自然な C++ ソリューション、効率的、スケーラブルなランタイム。	C++ オブジェクト指向のプログラミング (テンプレートとコンテナ) の理解が必要。
インテル® MKL	極めて高いパフォーマンスが求められる、科学、工学、金融系アプリケーション向けに高度に最適化され、広範囲にスレッド化された演算ルーチン。コードの並列性を意識する必要がない。	広範囲にスレッド化された算術関数: BLAS、LAPACK、スパースソルバー、高速フーリエ変換、ベクトルマシ、ベクトル統計。



4 インテル® Parallel Debugger Extension によるコードのデバッグ

サンプルを開く

インテル® C++ コンパイラー・プロフェッショナル・エディションでは、以下の場所にサンプル・アプリケーションが用意されています。

<install-dir>\Samples\<<locale>\C++\<<solution name>.zip

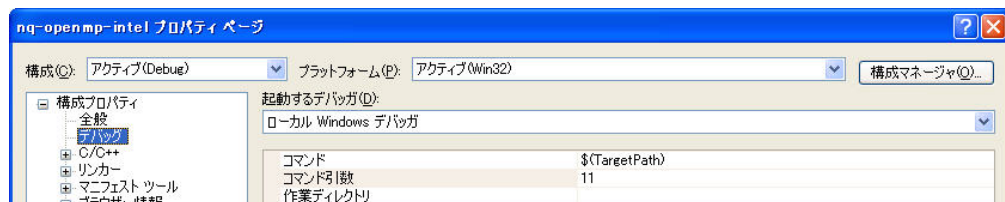
1. <solution name>.zip を任意のディレクトリーに解凍します。
2. .sln ファイルをダブルクリックするか、または Microsoft Visual Studio 開発環境からソリューション・ファイルを開きます。

NQueens.zip サンプルを使用して、下記の手順を行ってください。

注: 製品のインストール時にデフォルトのインストール場所を指定した場合、<install-dir> は C:\Program Files\Intel\Compiler\11.1\<<build_number> です。

デバッグ用サンプルコードの準備

1. NQueens.zip を任意のディレクトリーに解凍します。
2. NQueens.sln をダブルクリックして、Microsoft Visual Studio でサンプル・プロジェクトを開きます。
3. nq-openmp-intel プロジェクトで、nq-openmp.cpp をダブルクリックします。
4. #pragma omp atomic 行をコメントアウトして、データ共有違反を作成します。
5. nq-openmp プロジェクト・プロパティーで、コマンド引数の 11 がプログラムに渡されるよう指定されていることを確認します。






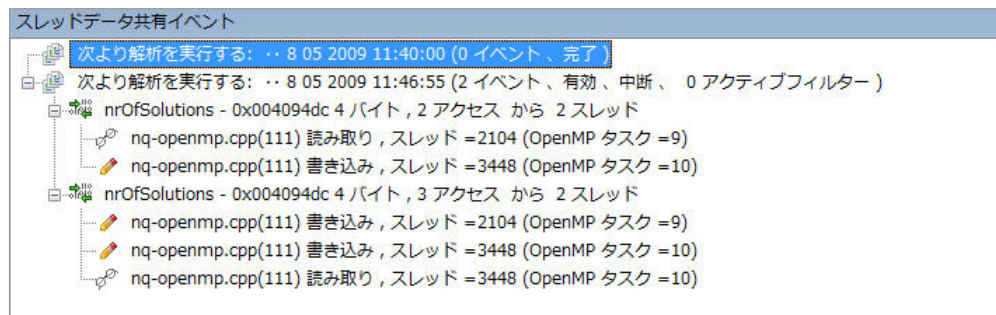


6. コンパイラーのコマンドライン・オプション /ZI、/debug:parallel と /Qopenmp がプロジェクトの [プロパティ ページ] で設定されていることをそれぞれ確認します:
 [構成プロパティ] > [C/C++] > [General (全般)] > [Debug Information Format (デバッグ情報の形式)],
 [構成プロパティ] > [C/C++] > [Debug (デバッグ)] > [Enable Parallel Debug Checks (並列デバッグチェックを有効にする)],
 [構成プロパティ] > [C/C++] > [Language (言語)] > [General (全般)] > [OpenMP Support (OpenMP サポート)].
7. リンカーのコマンドライン・オプション、/DEBUG が設定されていることを確認します:
 [構成プロパティ] > [Linker (リンカー)] > [Debugging (デバッグ)] > [Generate Debug Info (デバッグ情報の生成)].
8. プロジェクトをリビルドします。

並列コーディングの問題の特定

ランタイムデータの共有違反を引き起こしている可能性のあるスレッドデータ共有イベントを特定するには、次の操作を行います。

1. [デバッグ] > [Intel Parallel Debugger Extension (インテル(R) Parallel Debugger Extension)] > [Thread Data Sharing Detection (スレッドデータ共有検出)] メニュー (または  をクリック) で [Enable Detection (検出を有効にする)] を選択します。
2. [デバッグ] メニュー (または  をクリック) から [デバッグ開始] を選択して、デバッグセッションを開始します。スレッドデータ共有イベントが検出されるとすぐにプログラム実行が停止します。
3. [Thread Data Sharing Events (スレッドデータ共有イベント)] ボタン  をクリックして、実行停止のイベントログを参照します。








結果:

[Thread Data Sharing Events (スレッドデータ共有イベント)] ログから、2つの OpenMP スレッドが `nrOfSolutions` 変数にアクセスしようとしていることがわかります。スレッド 3956 が値を読み取り、スレッド 3756 が値を増分しています。これは、1つの変数に対して変更、依存する2つのスレッドがあるため、N-Queens アルゴリズムで不正な結果を起こすことのある深刻な問題です。#pragma omp atomic を使用して、`nrOfSolutions++` をインクリメント・スレッド・セーフにします。

関数の再入可能性イベントを特定するには、次の操作を行います。

1. [Enable Detection (検出を有効にする)] ボタン  をクリックして、データ共有イベント検出を無効  に設定します。
2. [デバッグ] > [Intel Parallel Debugger Extension (インテル(R) Parallel Debugger Extension)] > [Break on Re-Entrant Call (再入可能呼び出しでブレーク)] を選択 (または  をクリック) します。
3. 表示されたポップアップ・ウィンドウに「{„nq-openmp.exe}setQueen」と入力します。「setQueen」と入力してもかまいません。
4. [Continue Debug (デバッグを続行)] をクリックします。
5. 関数の再入可能性を検出すると、[Re-entrant Call Detected (再入可能呼び出しの検出)] ウィンドウが開き、スレッド ID に関する情報が表示されます。
6. [OK] をクリックします。

結果:

再入可能性が検出された関数のエントリーポイントにデバッガー命令ポインターが待機しています。

```
void setQueen(int queens[], int row, int col, int id) {  
    for(int i=0; i<row; i++) {  
        // vertical attacks  
        if (queens[i]==col) {  
            return;  
        }  
    }  
}
```

OpenMP タスク 20 と OpenMP タスク 19 の両方とも、`setQueen()` 関数に入っています。両タスクとも値を変更し、この関数内のデータを使用しています。そのため、互いのデータを上書きしないようにしなければなりません。`setQueen()` 関数でランタイム問題が見られるとき、`setQueen()` がスレッドセーフでない場合、このポインターは、スレッドセーフな実装が必要であることを示す非常に重要なポインターとなります。

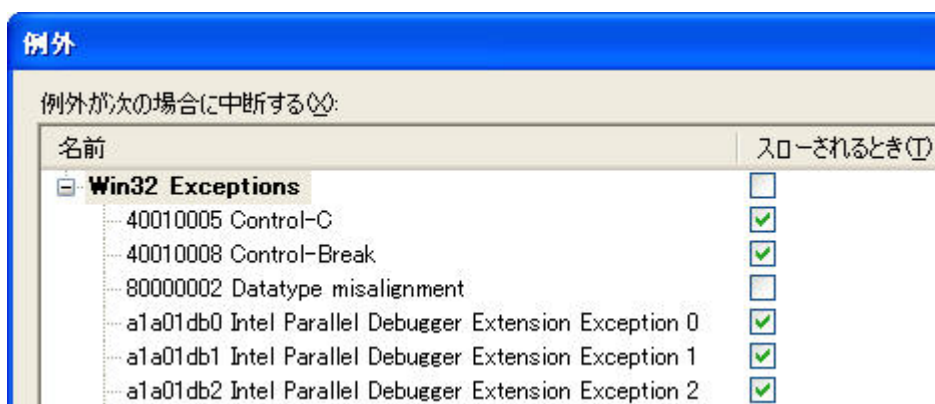


デバッグに関するその他の考慮事項

Microsoft Visual Studio 2005 を使用している場合、最初にインテル® Parallel Debugger の例外が Microsoft Visual Studio デバッガーで有効に設定されていることを確認してください。

次の操作を行って、例外が有効に設定されていることを確認します。

1. **[デバッグ]** プルダウンメニューから **[例外...]** を選択します。
2. **[Win32 Exceptions (Win 32 の例外)]** ツリービューで **[Intel Parallel Debugger Extension Exception (インテル(R) Parallel Debugger Extension の例外)]** が有効であることを確認します。




3. **[ファイル]** プルダウンメニューの **[すべてを保存]** を選択します。

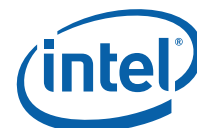
5 ユーザー・リファレンス・ドキュメント

このガイドは、インテル® C++ コンパイラー・プロフェッショナル・エディションの基本機能を紹介しています。詳細な機能説明については、次の資料を参照してください。

リソース	説明
インテル® C++ コンパイラー・プロフェッショナル・エディションのドキュメント	<p>この HTML ページでは、下記のインテル® C++ コンパイラー 11.1 プロフェッショナル・エディションのドキュメントへのリンクが含まれています。</p> <ul style="list-style-type: none"> • インテル® C++ コンパイラー • インテル® スレディング・ビルディング・ブロック • インテル® マス・カーネル・ライブラリー • 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® Parallel Debugger Extension



	Windows の [スタート] メニューから、 [Intel(R) Software Development Tools (インテル(R) ソフトウェア開発ツール)] > [Intel(R) C++ Compiler Professional 11.1.xxx (インテル(R) C++ コンパイラー 11.1.xxx プロフェッショナル)] > [Documentation (ドキュメント)] をクリックして表示します。
サンプルコード	さまざまなスレッド化手法を学習できるサンプルファイルが <install_dir>\Samples\<locale>\C++\ に zip 形式で提供されています。samples.htm ファイルは、C++ サンプルの概要を説明します。
統合ユーザー・リファレンス・ヘルプ	Microsoft Visual Studio の [ヘルプ] メニューから統合ユーザー・リファレンス・ヘルプを開きます。[ヘルプ] > [Intel C++ Compiler Pro (インテル(R) C++ コンパイラー・プロフェッショナル)] > [Intel C++ Compiler Help (インテル(R) C++ コンパイラー・ヘルプ)] を選択してください。 インテル® C++ コンパイラーのクイック・スタート・ヘルプを有効にするには、ツールバーの  をクリックします。 フィルタリングは、目次とキーワード情報で利用できます。 [フィルタ条件] ドロップダウン・リストから [Intel (インテル)] を選択し、インテル® C++ コンパイラー・プロフェッショナル・ドキュメントのみを表示します。
製品情報とテクニカルサポート	インテル® ソフトウェア製品およびサポート情報については、 http://www.intel.co.jp/jp/software/products/ を参照してください。このサイトでは、以下の包括的な製品情報を提供しています。 <ul style="list-style-type: none">• ホワイトペーパーや技術資料などの技術的な情報を入手することができる、各製品へのリンク• ユーザーフォーラムへのリンク• ニュースおよびイベントへのリンク テクニカルサポート情報の検索、製品の登録、インテルへの問い合わせは、 http://www.intel.com/software/products/support/ (英語) にアクセスしてください。サポート情報についての詳細は、『リリースノート』の「テクニカルサポート」を参照してください。 ライセンスおよび登録方法についての詳細は、Web サイト https://registrationcenter.intel.com/ を参照してください。



著作権と商標について

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。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を許諾するためのものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商適格性、あらゆる特許権、著作権、その他知的財産権の非侵害性への保証を含む) に関していかなる責任も負いません。

インテルによる書面での合意がない限り、インテル製品は、その欠陥や故障によって人身事故が発生するようなアプリケーションでの使用を想定した設計は行われていません。

インテル製品は、予告なく仕様や説明が変更されることがあります。機能または命令の一覧で「留保」または「未定義」と記されているものがありますが、その「機能が存在しない」あるいは「性質が留保付である」という状態を設計の前提にしないでください。これらの項目は、インテルが将来のために留保しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一切責任を負いません。この情報は予告なく変更されることがあります。この情報だけに基づいて設計を最終的なものとししないでください。

本書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。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本書で紹介されている注文番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725 (アメリカ合衆国) までご連絡いただくか、インテルの Web サイトを参照してください。

インテル・プロセッサ・ナンバーはパフォーマンスの指標ではありません。プロセッサ・ナンバーは同一プロセッサ・ファミリー内の製品の機能を区別します。異なるプロセッサ・ファミリー間の機能の区別には用いられません。詳細については、http://www.intel.co.jp/jp/products/processor_number/ を参照してください。

Intel、インテル、Intel ロゴは、アメリカ合衆国および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。

© 2009 Intel Corporation. 無断での引用、転載を禁じます。

Microsoft 製品のスクリーンショットは、Microsoft Corporation の許可を得て使用しています。