

インテル(R) C++ コンパイラー 10.0 日本語版 スペシャル・エディション 入門ガイド

目次

概要

インテル(R) C++ コンパイラーの設定

はじめに

検証用ソースファイル

適切なインストールの確認

コンパイラーの起動 (コマンドライン)

コンパイル (最適化オプションなし)

実行/プログラムの検証

コンパイル (最適化オプションあり)

実行/パフォーマンスの比較

コンパイラーの起動 (Microsoft* Visual Studio* .NET IDE)

コンパイル (最適化オプションなし)

実行/プログラムの検証

コンパイル (最適化オプションあり)


実行/パフォーマンスの比較

既存ソースのコンパイル

追加情報


概要

インテル(R) C++ コンパイラー 10.0 Windows* 版は、Windows ホストシステムでコマンドラインまたは Microsoft* Visual Studio* .NET 統合開発環境 (IDE) のいずれかを使用して C/C++ ソースファイルをコンパイルし、Windows プラットフォームで実行するアプリケーションを作成します。この製品、および他のインテル(R) ソフトウェア開発製品に関する詳細は、[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。

 **注:** C++ コンパイラーのデフォルトのインストール・パスは C:\Program Files\Intel\Compiler\C++\10.0.025 です。このガイドでは、パス C:\Program Files\Intel は <install-dir> と表記するため、デフォルトの IA-32 インストール・パスは <install-dir>\Compiler\C++\10.0.025\IA32 と表記されます。

インテル(R) C++ コンパイラーの設定

インテル C++ コンパイラー 10.0 Windows 版が正常にインストールされたら、環境変数を設定する必要があります。[「インストール・ガイド」](#)を参照して、インテル C++ コンパイラー 10.0 Windows 版が正しく設定されていることを確認してください。

 **注:** Microsoft Visual Studio 環境を先にインストールした場合、インテル C++ コンパイラー 10.0 Windows 版を Microsoft Visual Studio IDE 環境で使用するための追加の手順は必要ありません。インテル C++ コンパイラー 10.0 Windows 版を先にインストールした場合は、[「インストール・ガイド」](#)の手順に従ってください。

[スタート] > [プログラム] > [Intel(R) Software Development Tools (インテル(R) ソフトウェア開発ツール)] > [インテル(R) C++ コンパイラー 10.0.025] > [IA-32 対応アプリケーション用インテル(R) C++ コンパイラーのビルド環境] を選択して、コマンド・ウィンドウを開くと、環境が自動的に設定されます。バッチ・ファイル (IA-32 アーキテクチャーの場合、<install-dir>\Compiler\C++\10.0.025\IA32\bin\iclvars.bat) を実行して環境を設定することもできます。

Microsoft Visual Studio IDE を使用している場合、プロジェクトを開いた後で、[プロジェクト] > [インテル(R) C++ プロジェクトへ変換] を選択してください。IDE を使用してプロジェクトを設定する方法の詳細は、インストールされるインテル(R) C++ コンパイラーのドキュメントの「アプリケーションのビルド」を参照してください。

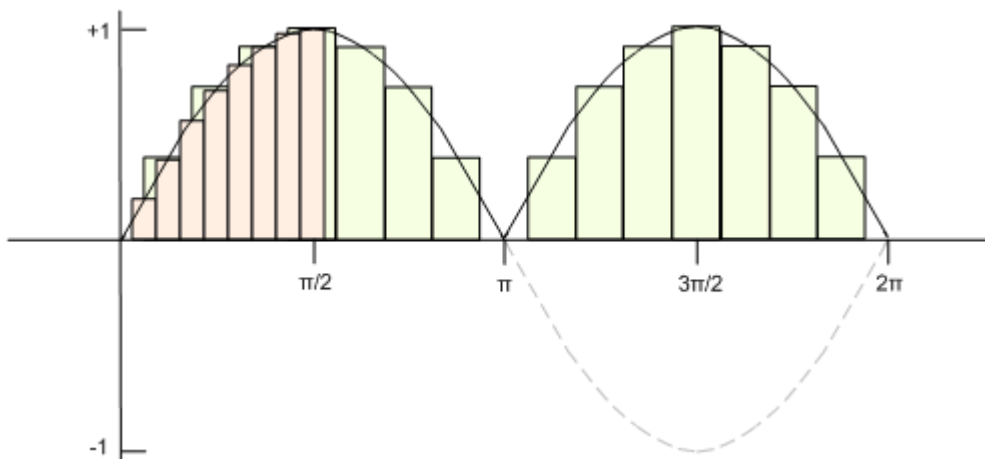
はじめに

インテル C++ コンパイラーのインストールが完了したら、適切なインストール、設定、およびコンパイラーの動作を確認する基本的な検証作業を実行してください。この検証には、コンパイラーのインストールの一部としてインストールされる検証用ソースファイル

`<install-dir>\Compiler\C++\10.0.025\samples\optimize\int_sin.c` を使用します。

検証用ソースファイル

検証用ソースファイルは、1 サイクル 2 パイラジアン（ 2π ）の正弦曲線の絶対値を統合する数値演算プログラムです。次の図は、計算に使用される方法を示しています。この方法は、曲線と上辺の中央部分が一致するように長方形を連続的に追加します。長方形の数が増えると（長方形の幅が狭くなると）、計算される領域は 4 (4.0) に近づきます。次の図は、 2^4 内点と 2^5 内点の最初の 8 片で何が計算されているかを示しています。



プログラムには、stdio、stdlib、時間ライブラリーに加えてインテルの数値演算ライブラリーが含まれているため、インテル・ライブラリーがインストールされているかどうかを確認します。プログラムのタイミン
グ関数は、プログラム実行の開始から終了までを測定したアプリケーション・クロックの数を返します。
この時間測定は不正確で、プロセッサとその作業量に依存して変わります。

適切なインストールの確認

検証用ソースファイルをコンパイルして実行し、出力が既知の正しい値である 4 に収斂するかどうか
をチェックすることで、コンパイラーが適切にインストールされたかどうかを確認できます。次の手順を
実行して、インストールを確認します。

1. この検証用ソースファイルは、

`<install-dir>\Compiler\C++\10.0.025\samples\optimize` ディレクトリにあり

ます。ディレクトリへの書き込み権限があることを確認してから、ディレクトリに移動し、次のように検証用ソースファイルをコンパイルします。

```
prompt> icl int_sin.c
```

2. <install-dir>\Compiler\C++\10.0.025\samples\optimize ディレクトリに、プログラム int_sin.exe が生成されます。次のようにプログラムを実行します。

```
prompt> int_sin
```

3. プログラムを起動したコマンド・プロンプト・ウィンドウに次のような出力が表示されます。

Number of Interior Points	Computed Integral
4	3.141593e+000
8	3.792238e+000
16	3.948463e+000
32	3.987141e+000
64	3.996787e+000
128	3.9910.0.02597e+000
256	3.999799e+000
512	3.999950e+000
1024	3.999987e+000
2048	3.999997e+000
4096	3.999999e+000
8192	4.000000e+000

```
-----  
16384      | 4.000000e+000 |  
-----  
32768      | 4.000000e+000 |  
-----  
65536      | 4.000000e+000 |  
-----  
131072     | 4.000000e+000 |  
-----  
262144     | 4.000000e+000 |  
-----  
524288     | 4.000000e+000 |  
-----  
1048576    | 4.000000e+000 |  
-----  
2097152    | 4.000000e+000 |  
-----  
4194304    | 4.000000e+000 |  
-----  
8388608    | 4.000000e+000 |  
-----  
16777216   | 4.000000e+000 |  
-----  
33554432   | 4.000000e+000 |  
-----  
67108864   | 4.000000e+000 |
```

Application Clocks = 5.500000e+003

4. コンパイラーの動作が異なる場合、またはコンパイルしたプログラムの出力が正しくない場合は、インストールと設定を確認してください。必要であれば、再インストールを行ってください。問題がなければ、インテル C++ コンパイラーは適切にインストールされ、正しく動作しています。

コンパイラーの起動 (コマンドライン)

インテル C++ コンパイラー 10.0 Windows 版は、`icl` を使用してコマンドラインから起動できます。ここでは、前述した `int_sin.c` 検証用ソースファイルを使用します。作業の大部分をコマンドラインからではなく、IDE インターフェイスを使用している場合は、このセクションをスキップして、[「コンパイラーの起動 \(Microsoft* Visual Studio* .NET IDE\)」](#)セクションに進んでください。

`int_sin.c` 検証用ソースファイルは、
`<install-dir>%Compiler%C++%10.0.025%samples%optimize` ディレクトリにあります。

このセクションを開始する前に、Windows の [スタート] メニューから [プログラム] > [Intel(R) Software Development Tools (インテル(R) ソフトウェア開発ツール)] > [インテル(R) C++ コンパイラー 10.0.025] > [IA-32 対応アプリケーション用インテル(R) C++ コンパイラーのビルド環境] を開き、次のようにディレクトリを移動してください。

```
prompt> cd <install-dir>%Compiler%C++%10.0.025%samples%optimize
```

コンパイル (最適化オプションなし)

最初に、最適化オプションを使用しないでコンパイルし、パフォーマンスの基準を確立します。次のようにインテル C++ コンパイラーを起動します。

```
prompt> icl int_sin.c /Od
```

`/zi` デバッグ・オプション (デフォルトの最適化が `/O2` から `/Od` に変更される) を使用してもかまいません。

実行/プログラムの検証

プログラムは、ソースと同じディレクトリに `int_sin.exe` という名前で生成されます。次のようにプログラムを実行します。

```
prompt> int_sin
```

各計算で消費される実行時間 (プロセッサ・クロック・サイクルの数) は、内点の数が増えると、計算された整数値 4.0 に近く (または等しく) なります。次のような出力が表示されます。

```
Number of      | Computed Integral |  
Interior Points |                    |  
-----
```

4		3.141593e+000	

8		3.792238e+000	

16		3.948463e+000	

32		3.987141e+000	

64		3.996787e+000	

128		3.9910.0.02597e+000	

256		3.999799e+000	

512		3.999950e+000	

1024		3.999987e+000	

2048		3.999997e+000	

4096		3.999999e+000	

8192		4.000000e+000	

16384		4.000000e+000	

32768		4.000000e+000	

65536		4.000000e+000	

131072		4.000000e+000	

262144		4.000000e+000	

524288		4.000000e+000	

1048576		4.000000e+000	

2097152		4.000000e+000	

```

-----
4194304      | 4.000000e+000 |
-----
8388608      | 4.000000e+000 |
-----
16777216     | 4.000000e+000 |
-----
33554432     | 4.000000e+000 |
-----
67108864     | 4.000000e+000 |

```

Application Clocks = 1.184300e+004

コンパイル (最適化オプションあり)

インテル C++ コンパイラーのいくつかの最適化オプションを使用して実現されるパフォーマンス向上は非常に重要です。他のオプションは、異なる領域で演算やパフォーマンスを向上します。次のように (デフォルトの最適化を使用して) コンパイラーを起動します。

```
prompt> icl int_sin.c
```

デフォルトでは、コンパイラーはコードの実行速度を優先するレベル 2 の最適化 (/O2) を行いません。

実行/パフォーマンスの比較

次のように int_sin プログラムの最適化バージョンを実行します。

```
prompt> int_sin
```

最適化を行わなかった場合と、アプリケーション・クロックの数を比較します。実際の時間の差は使用するアーキテクチャーに依存します。次の出力は、一般的な IA-32 システムのものです。

```

Number of      | Computed Integral |
Interior Points |                    |
-----
4              | 3.141593e+000     |
-----
8              | 3.792238e+000     |

```


16		3.948463e+000	
32		3.987141e+000	
64		3.996787e+000	
128		3.9910.0.02597e+000	
256		3.999799e+000	
512		3.999950e+000	
1024		3.999987e+000	
2048		3.999997e+000	
4096		3.999999e+000	
8192		4.000000e+000	
16384		4.000000e+000	
32768		4.000000e+000	
65536		4.000000e+000	
131072		4.000000e+000	
262144		4.000000e+000	
524288		4.000000e+000	
1048576		4.000000e+000	
2097152		4.000000e+000	
4194304		4.000000e+000	

```
8388608      | 4.000000e+000 |
-----
16777216     | 4.000000e+000 |
-----
33554432     | 4.000000e+000 |
-----
67108864     | 4.000000e+000 |
```

Application Clocks = 5.500000e+003

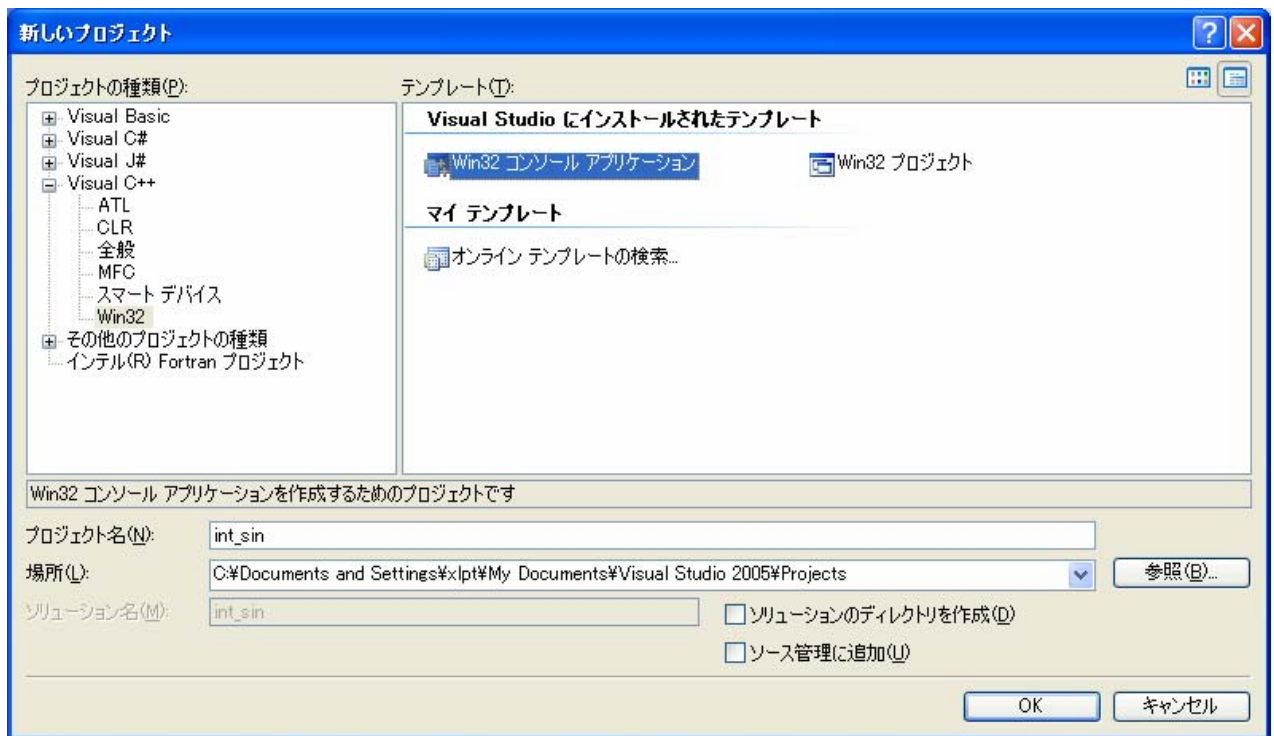


注: この例における (最適化なしから最適化ありにした場合の) 実行時間の大幅な向上はすべてのプログラムにあてはまるものではありませんが、通常は、適切な最適化を行うことで、インテル(R) プロセッサ上で実行するプログラムの実行時間を向上できます。インテル C++ コンパイラーは、デフォルトでは /O2 レベルでプログラムを最適化する点に注意してください。

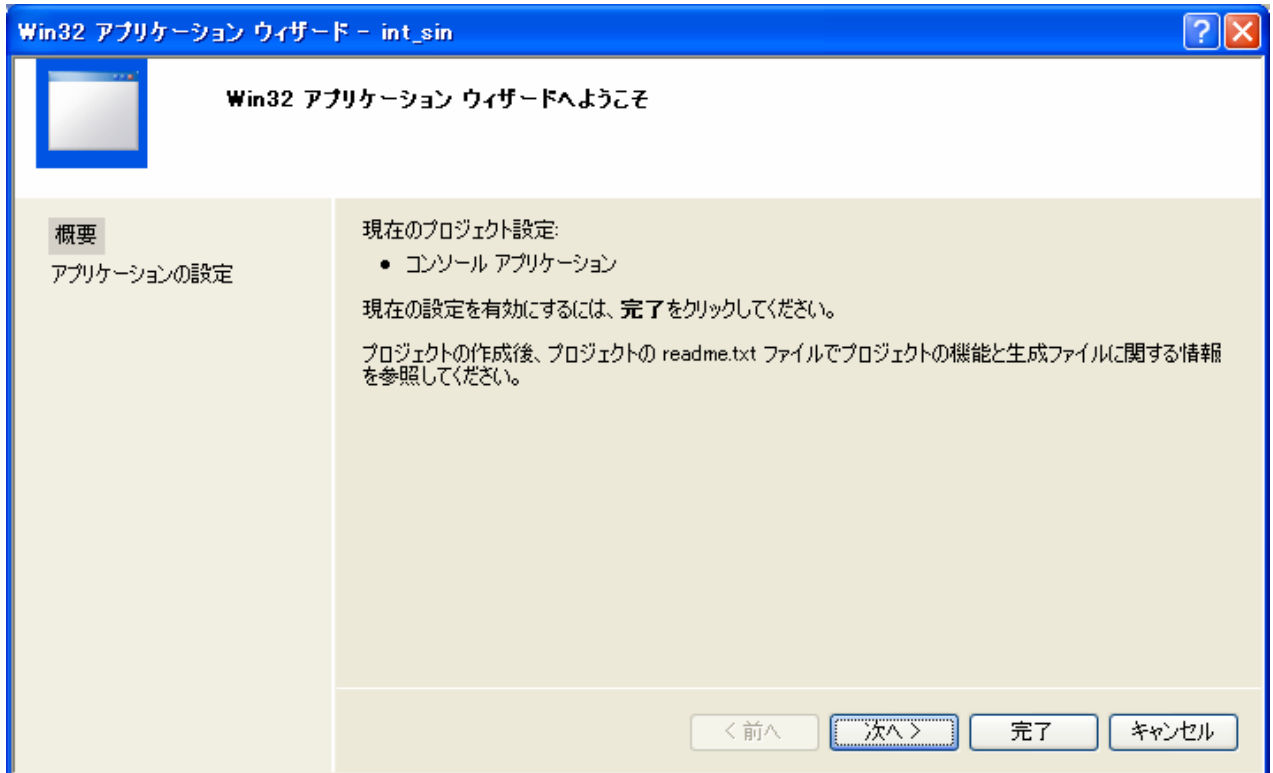
コンパイラーの起動 (Microsoft* Visual Studio* .NET IDE)

インテル C++ コンパイラーをインストールするシステムに Microsoft Visual Studio .NET 2003 または Microsoft Visual Studio 2005 がインストールされている場合、追加の手順なしで Microsoft Visual Studio .NET IDE 環境でインテル C++ コンパイラーを使用できます。Visual Studio .NET の実行中にインテル C++ コンパイラーをインストールした場合、Visual Studio を再起動すると、インテル C++ コンパイラーが選択できるようになります。インテル C++ コンパイラーをインストールしたら、次の手順を実行します。

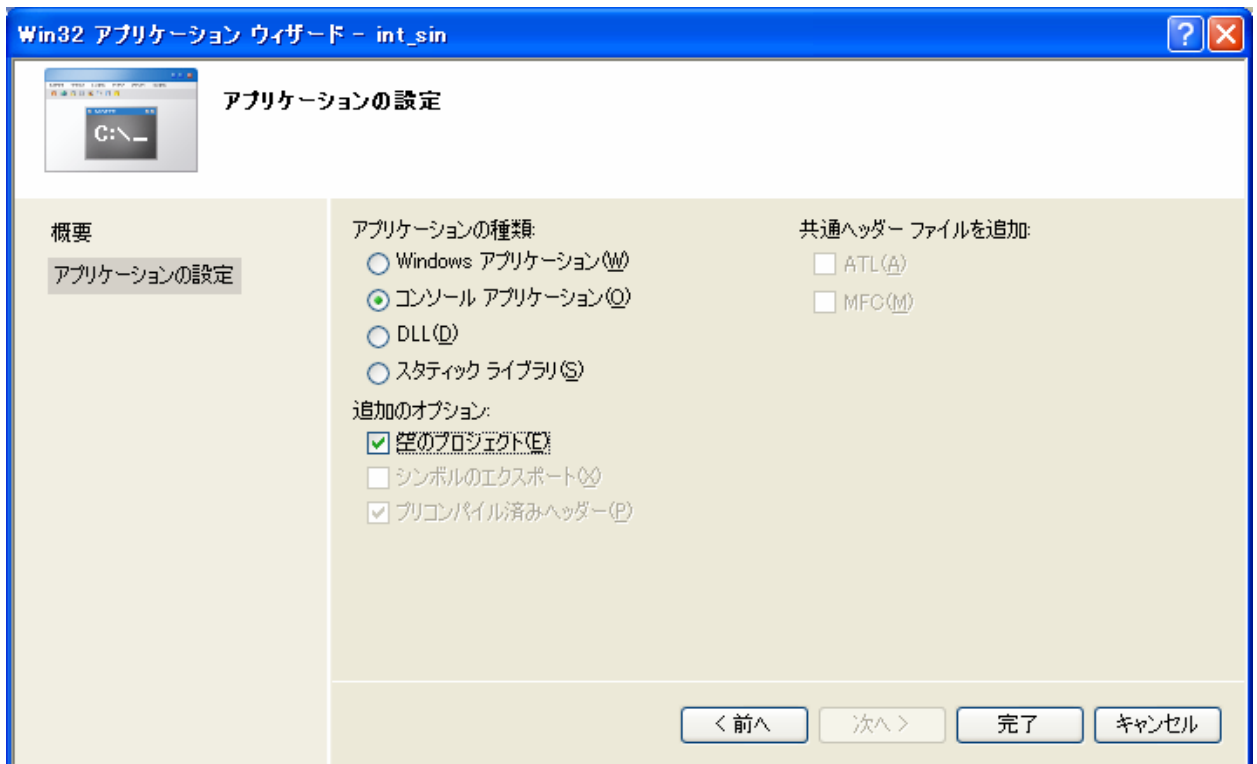
1. Visual Studio .NET で [ファイル] - [新規作成] - [プロジェクト] を選択して新規プロジェクトを作成します。[新しいプロジェクト] ダイアログ・ボックスで、次に示すように、**Win32 コンソール プロジェクト**を選択します。プロジェクト名として int_sin を使用します。



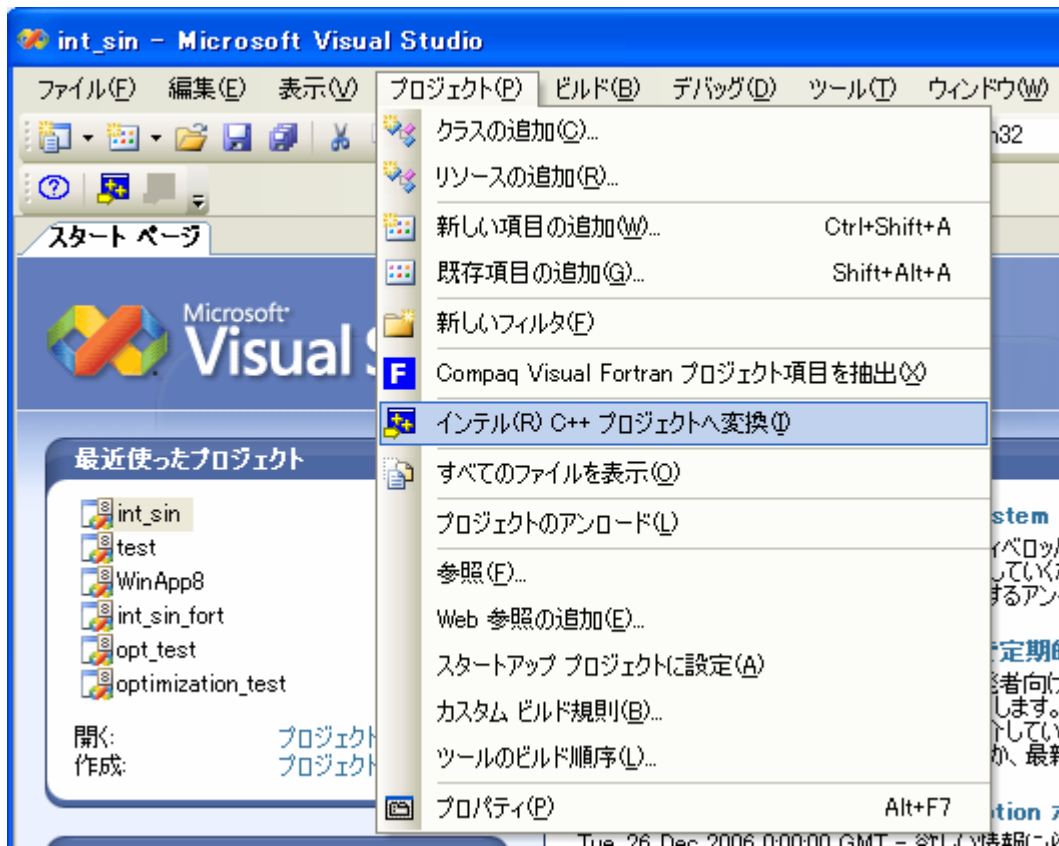
2. Win32 アプリケーション ウィザードが次のウィンドウを表示します。



3. [アプリケーションの設定] を選択して、[追加のオプション] で [空のプロジェクト] を選択します。[完了] をクリックして、Win32 コンソール・プロジェクトを作成します。



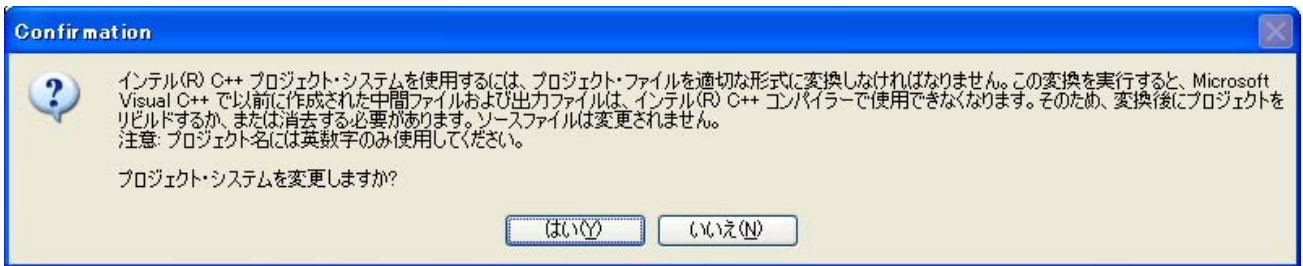
4. 次に、[プロジェクト] メニューから [インテル(R) C++ プロジェクトへ変換] を選択、またはインテル(R) C++ ツール バーの [インテル(R) C++ プロジェクトへ変換] ボタンをクリックします。Microsoft Visual C++ プロジェクトがインテル C++ プロジェクトに変換され、新しいインテル C++ プロジェクト・ファイル (.icproj) が作成されます。



または、

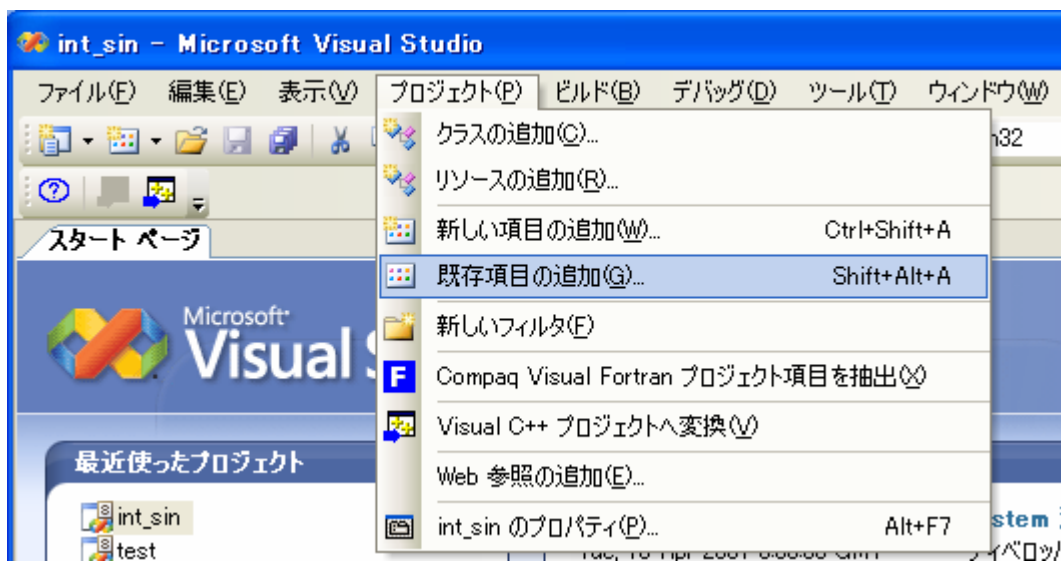


5. [Confirmation (確認)] ダイアログで [はい] をクリックします。変換が成功するとソリューション エクスプローラー内にインテル(R) C++ プロジェクトのロゴが表示されます。

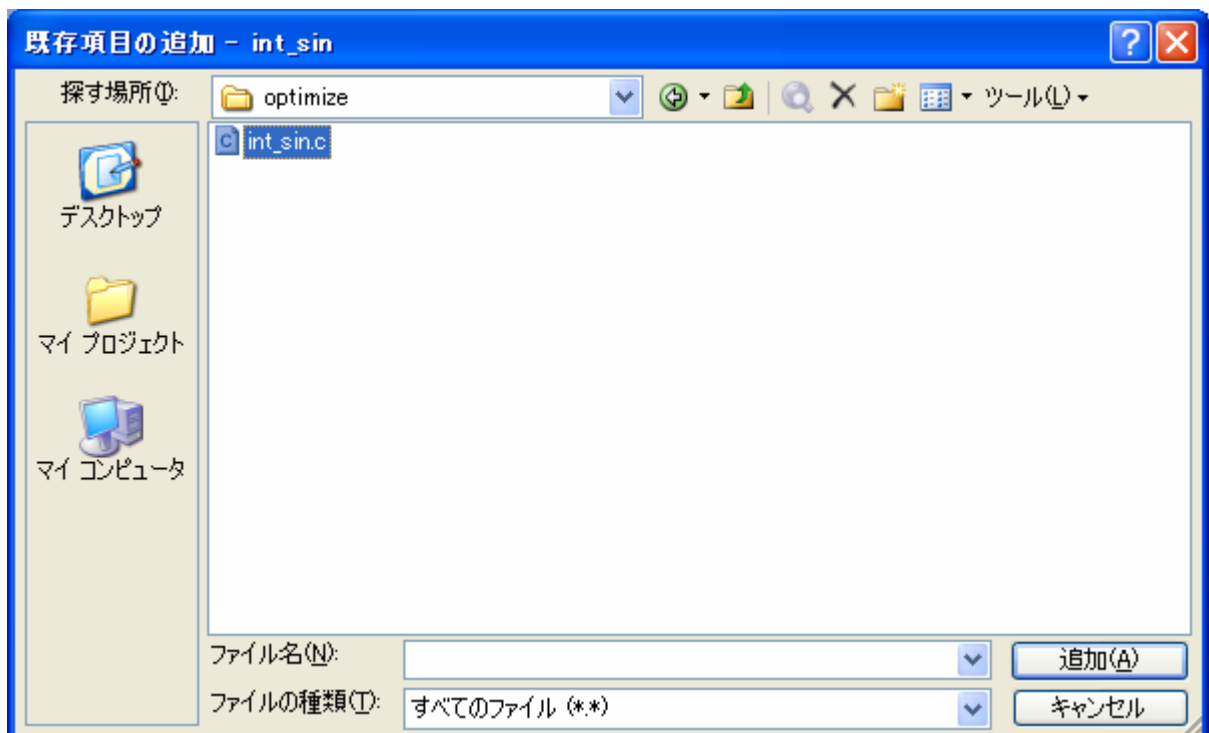
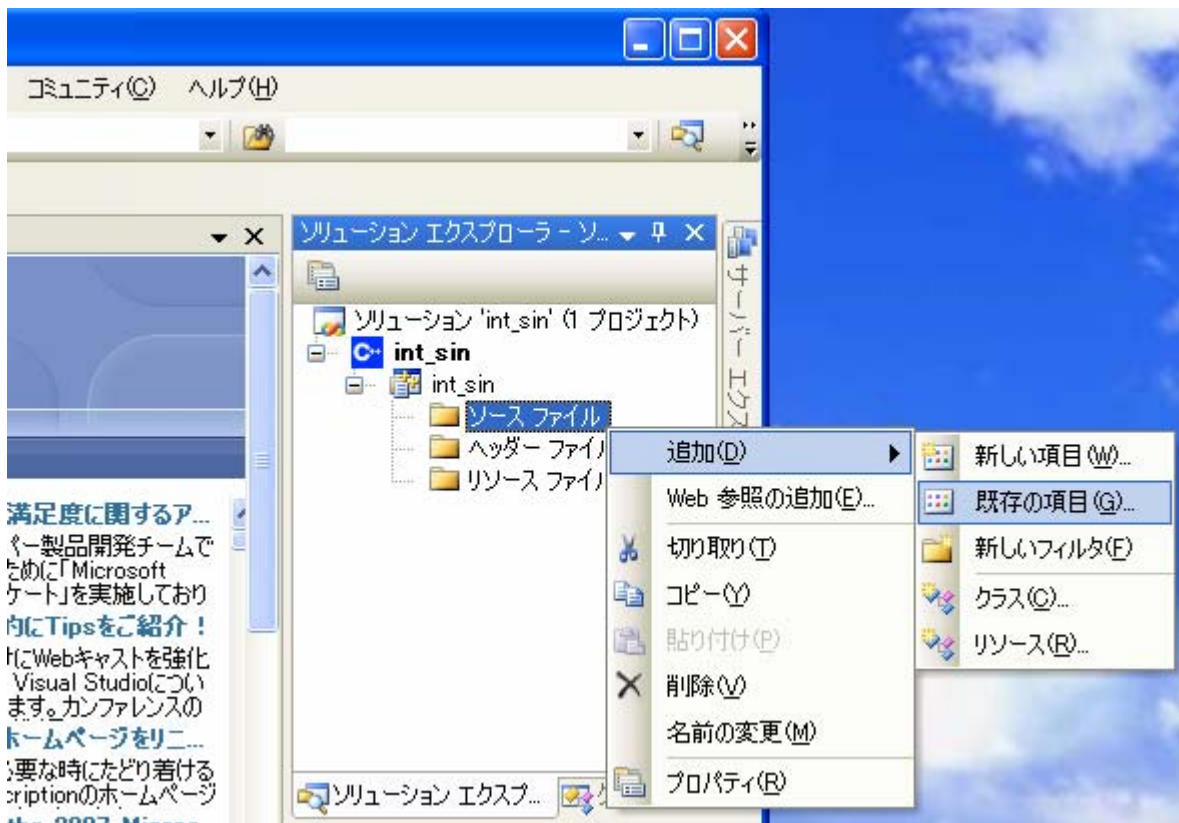


6. IDE で [プロジェクト] - [既存項目の追加...] を選択、またはソリューション エクスプローラからソースファイル を右クリックして表示されるメニューから [追加] - [既存の項目] を選択し、表示される「既存項目の追加」ダイアログで以下のファイルを選択して [追加] ボタンをクリックします。

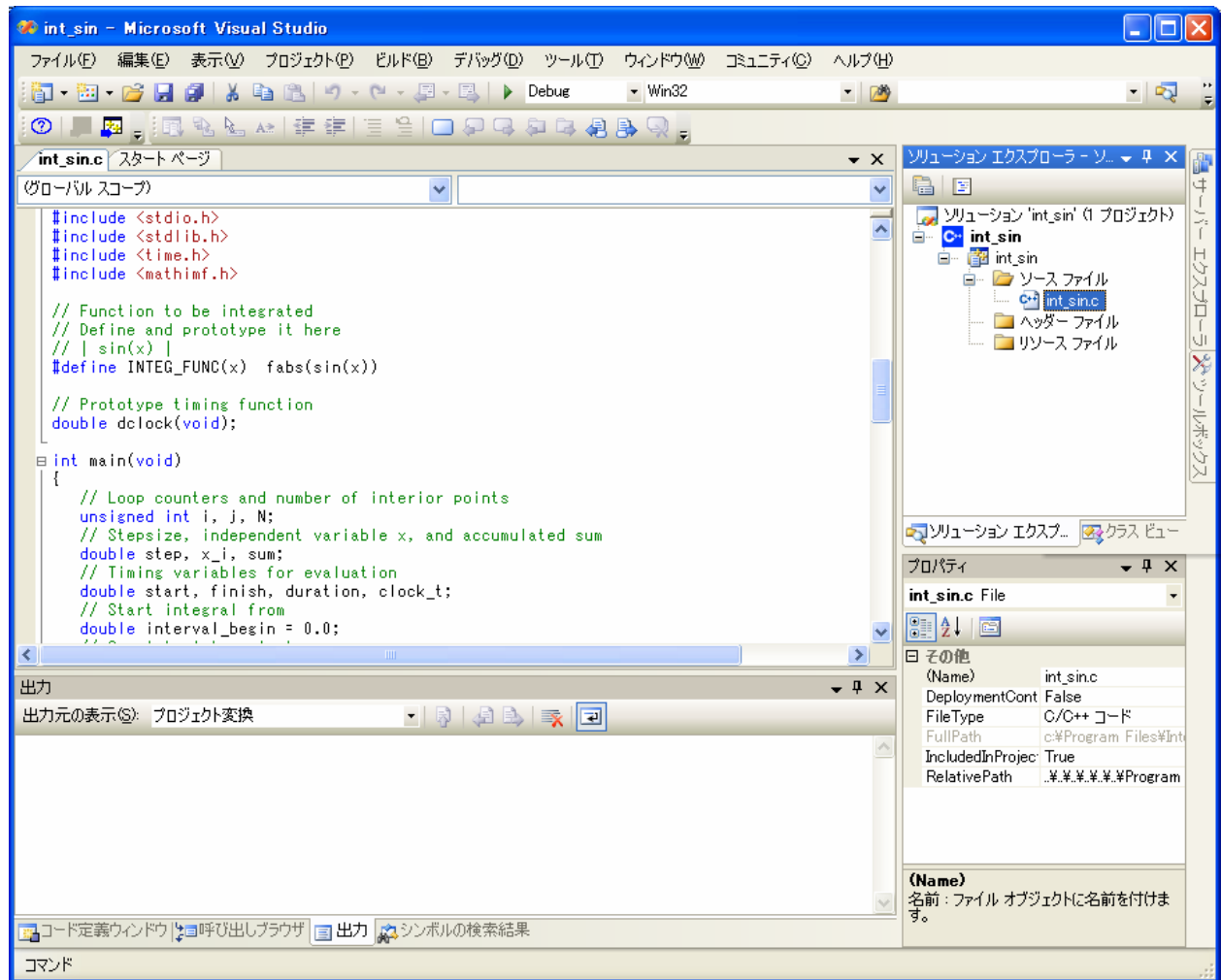
<install-dir>%Compiler%¥C++¥10.0.025¥samples¥optimize¥int_sin.c



または、



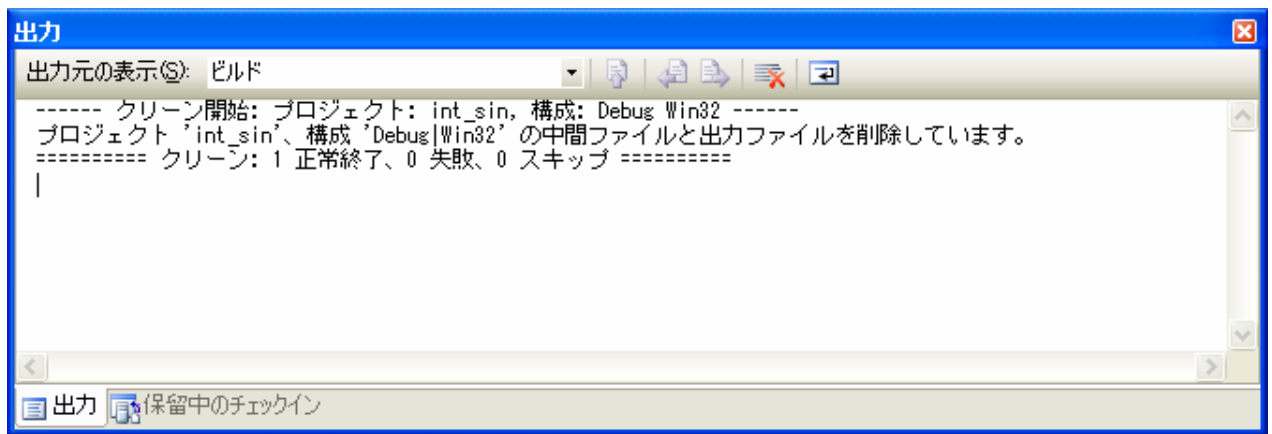
7. 新しいプロジェクト int_sin に、サンプルのソースファイル int_sin.c が追加されます。



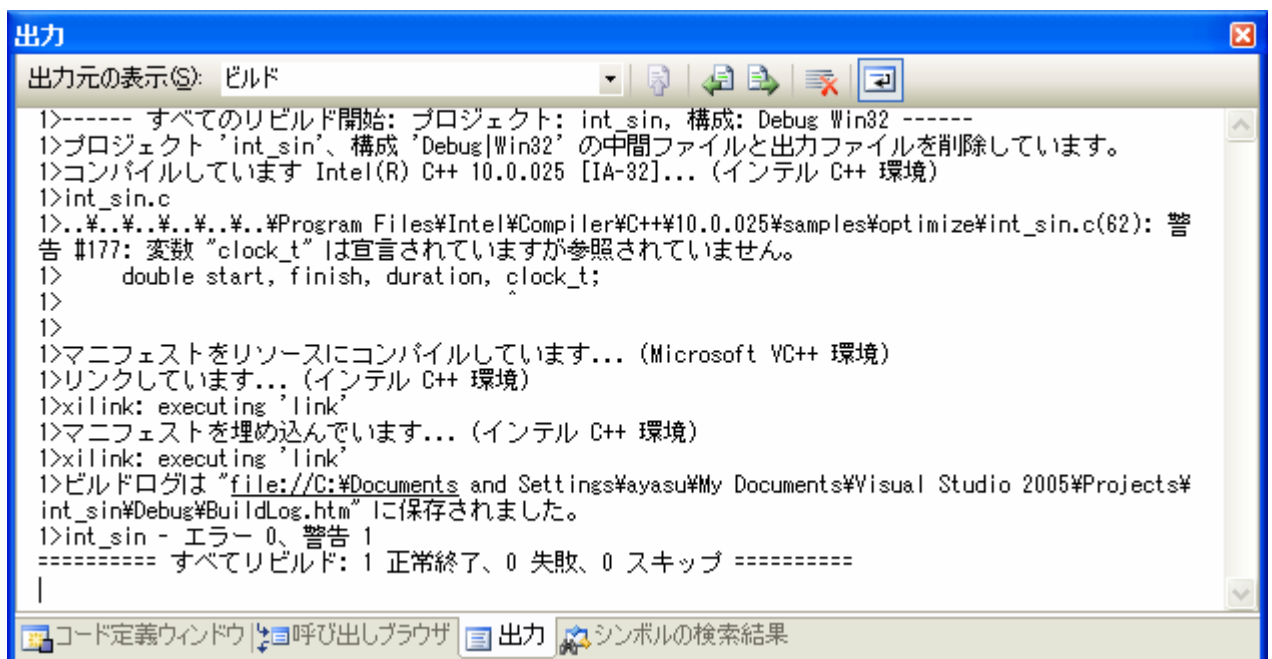
コンパイル（最適化オプションなし）

次の手順を実行します。

1. VS .NET でソリューションをビルドする前に、ソリューションを消去します。[ビルド] - [ソリューションのクリーン] を選択します。次のレポートが表示されます。



- 次に、[ビルド] - [ソリューションのビルド] を選択します。次のレポートが表示されます。



デフォルトのプロジェクト構成は Debug 構成なので、ソリューションは最適化なし、シンボリック・デバッグ情報付きでビルドされる点に注意してください。

これは、コマンドラインで `icl /Od /Zi int_sin.c` と入力した場合と同じです。

実行/プログラムの検証

- [デバッグ] - [デバッグなしで開始] を選択します。コマンド・ウィンドウに次の出力が表示されます。

1024	3.999987e+000
2048	3.999997e+000
4096	3.999999e+000
8192	4.000000e+000
16384	4.000000e+000
32768	4.000000e+000
65536	4.000000e+000
131072	4.000000e+000
262144	4.000000e+000
524288	4.000000e+000
1048576	4.000000e+000
2097152	4.000000e+000
4194304	4.000000e+000
8388608	4.000000e+000
16777216	4.000000e+000
33554432	4.000000e+000
67108864	4.000000e+000

Application Clocks = 1.090600e+004

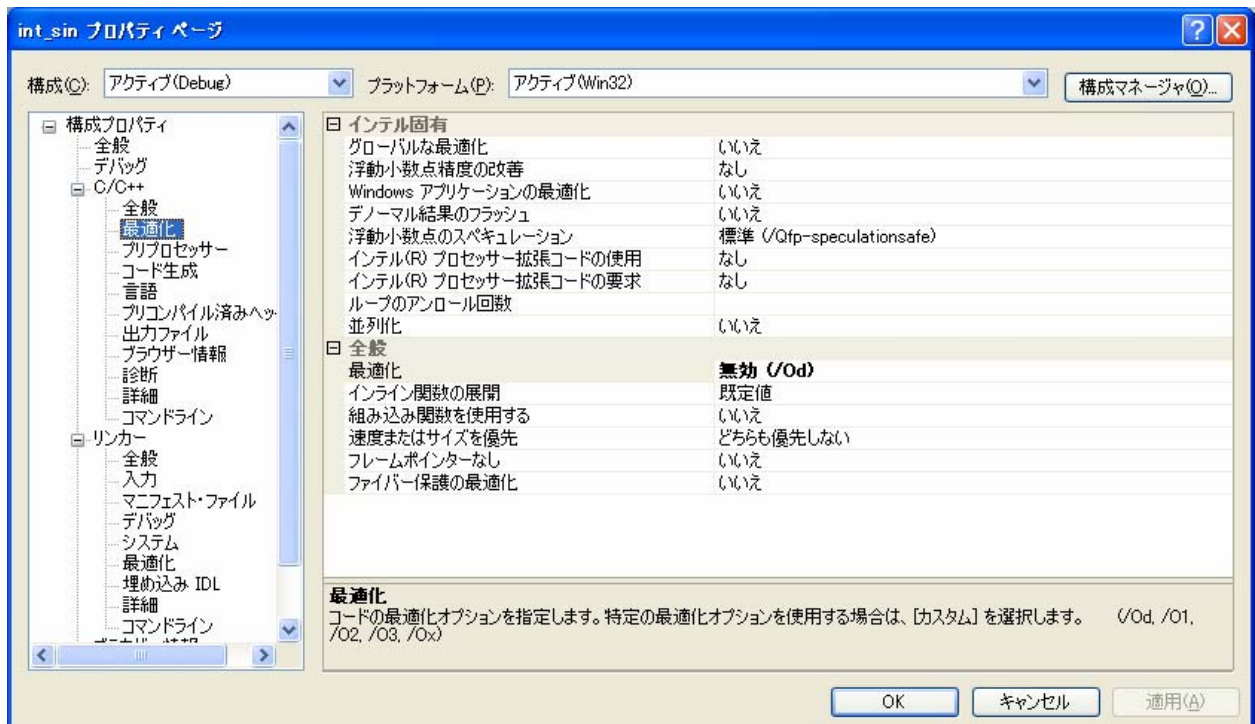
続行するには何かキーを押してください . . .

2. 最適化を行わなかった場合のアプリケーション・クロックの数をメモします。

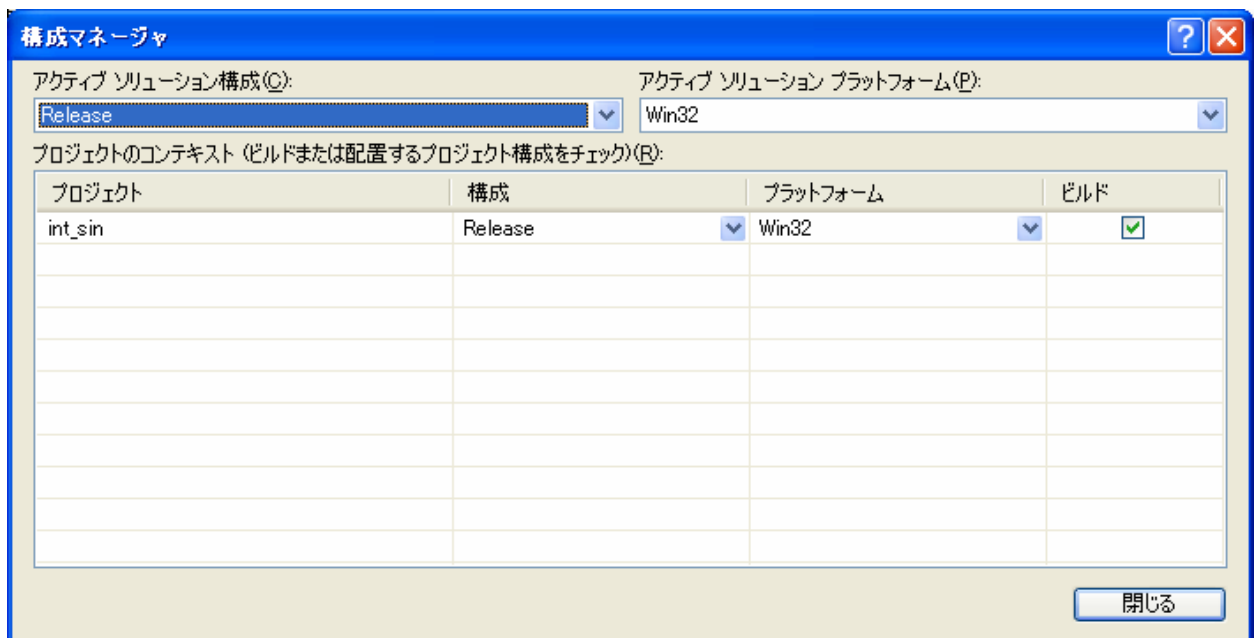
コンパイル（最適化オプションあり）

最適化を行って再コンパイルします。次の手順を実行します。

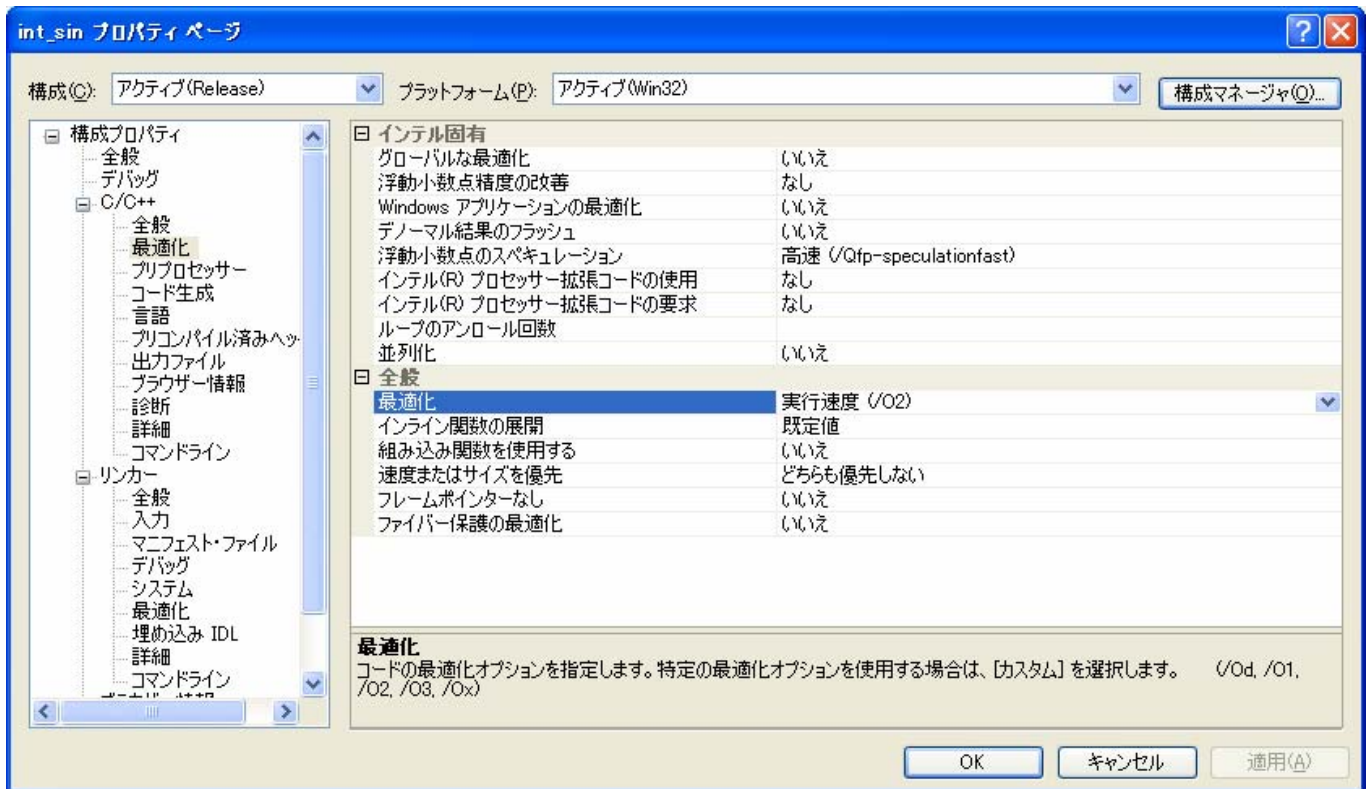
1. [プロジェクト] - [int_sin のプロパティ] を選択します。次のダイアログ・ボックスが表示されます。



2. [構成マネージャ...] をクリックして、[アクティブ ソリューション構成] を **Release** に変更します。[閉じる] をクリックします。



3. [プロパティ ページ] ダイアログ・ボックスに戻り、[構成プロパティ] - [C/C++] - [最適化] を選択して、[最適化] が**実行速度 (/O2)** に設定されていることを確認します。[コマンドライン] を使用してオプションを定義することもできます。




4. [ビルド] - [ソリューションの消去] を選択してから、[ビルド] - [ソリューションのビルド] を選択して、最適化ありで int_sin プロジェクトをビルドします。

実行/パフォーマンスの比較

1. [デバッグ] - [デバッグなしで開始] を選択します。コマンド・ウィンドウに次の出力が表示されます。

```
C:\ * C:\Documents and Settings\xlpt\My Documents\Visual Studio 2005\Projec... - [ ] X
-----|-----|
2048   | 3.999997e+000 |
-----|-----|
4096   | 3.999999e+000 |
-----|-----|
8192   | 4.000000e+000 |
-----|-----|
16384  | 4.000000e+000 |
-----|-----|
32768  | 4.000000e+000 |
-----|-----|
65536  | 4.000000e+000 |
-----|-----|
131072 | 4.000000e+000 |
-----|-----|
262144 | 4.000000e+000 |
-----|-----|
524288 | 4.000000e+000 |
-----|-----|
1048576| 4.000000e+000 |
-----|-----|
2097152| 4.000000e+000 |
-----|-----|
4194304| 4.000000e+000 |
-----|-----|
8388608| 4.000000e+000 |
-----|-----|
16777216| 4.000000e+000 |
-----|-----|
33554432| 4.000000e+000 |
-----|-----|
67108864| 4.000000e+000 |
-----|-----|
Application Clocks = 6.562000e+003
続行するには何かキーを押してください . . .
```

2. 最適化を行った場合のアプリケーション・クロックの数をメモして、最適化を行わなかった場合と比較します。

 **注:** この例における（最適化なしから最適化ありにした場合の）実行時間の大幅な向上はすべてのプログラムにあてはまるものではありませんが、通常は、適切な最適化を行うことで、インテル・プロセッサ上で実行するプログラムの実行時間を向上できます。インテル C++ コンパイラは、デフォルトでは /O2 レベルでプログラムを最適化する点に注意してください。

最適化設定とその他のオプションを変更することで、プログラムの実行速度、コンパイル時間、コードサイズに与える影響を判断できます。

既存ソースのコンパイル

このガイドでは、コンパイル、最適化の適用/削除、引数の指定、およびプログラム出力をモニタする手順を説明しました。既存の C/C++ ソース・プログラムがある場合、例の代わりにそれらのソースファイルをコンパイルして実行してみてください。

追加情報

このガイドで提供した例は、インテル C++ コンパイラーの機能の一部を紹介しただけに過ぎません。

その他の機能は、インストールされるドキュメント・インデックス `doc_index.htm` から参照できます。このインデックスには、本製品に含まれているすべてのドキュメントへのリンクが含まれています。

インテル C++ コンパイラー 10.0 Windows 版の詳細は、インストールされるインテル(R) C++ コンパイラーのドキュメントを参照してください。

テクニカル・サポートや制限事項については、製品の[リリースノート](#)を参照してください。

インテル・ソフトウェア開発製品の詳しい情報については、次のインテル Web サイトを参照してください。

<http://www.intel.co.jp/jp/software/products/index.htm>