



メモリーエラーの排除と プログラムの安定性の向上



メモリーエラーの排除と プログラムの安定性の向上



1 つのツールを実行するだけで違いが出るのでしょうか？

はい。多くの場合、複雑なバグを発見して、アプリケーションの安定性を向上させることができます。このガイドでは、インテル® Parallel Inspector 分析ツールを使用して、コードの問題を排除する方法を説明します。これにより、コードの信頼性が向上し、開発コストを抑えることができます。次に、サンプル・アプリケーションの使用方法を示します。

パフォーマンス向上のための 3 つのステップ

ステップ 1. インテル® Parallel Studio のインストール

1. [インテル® Parallel Studio の評価版をダウンロード](#)します。
2. parallel_studio_setup.exe をクリックして、インテル® Parallel Studio をインストールします。

ステップ 2. Tachyon サンプル・アプリケーションのインストールと参照

サンプル・アプリケーションのインストール：

1. サンプルファイル [Tachyon_conf.zip](#) をダウンロードします。このサンプルは、Microsoft® Visual Studio® 2005 を使用して作成された C++ コンソール・アプリケーションです。
2. Tachyon_conf.zip ファイルをシステムの書き込み可能なフォルダーに展開します。

ステップ 3. インテル® Parallel Inspector を使用したメモリーエラーの検出

インテル® Parallel Inspector は、Microsoft Visual Studio C/C++ を利用する開発者向けのシングルおよびマルチスレッドのエラー検証ツールです。リークや破壊を含むメモリーエラーに加えて、データ競合やデッドロックのようなスレッドエラーも検出します。この統合された開発支援ツールは、エラーを正確に示し、アプリケーションの信頼性と品質を保証するためのガイダンスを提供します。

注：インテル® Parallel Studio は Microsoft Visual Studio 2005/2008 に統合されます。このチュートリアルでは、Microsoft Visual Studio 2005 開発環境 (IDE) を使用しています。メニュー項目の異なる IDE を使用する場合はご注意ください。

注：サンプルを実行した結果が常に同じになるとは限りません。実際の画面は、チュートリアルで示されている画面とは異なることがあります。

メモリーエラーの排除とプログラムの安定性の向上



メモリーエラーの特定、分析、解決

インテル® Parallel Inspector を使って、一連のステップを実行することで、直列または並列プログラムのメモリーエラーを特定、分析、解決できます。このチュートリアルでは、tachyon_conf という名前のサンプルプログラムを使用して、順に説明します。

ターゲットの選択

1. Microsoft Visual Studio でサンプルを開きます。[ファイル] > [開く] > [プロジェクト/ソリューション] を選択して、tachyon_conf\vc8\tachyon_conf.sln ファイルを開きます。

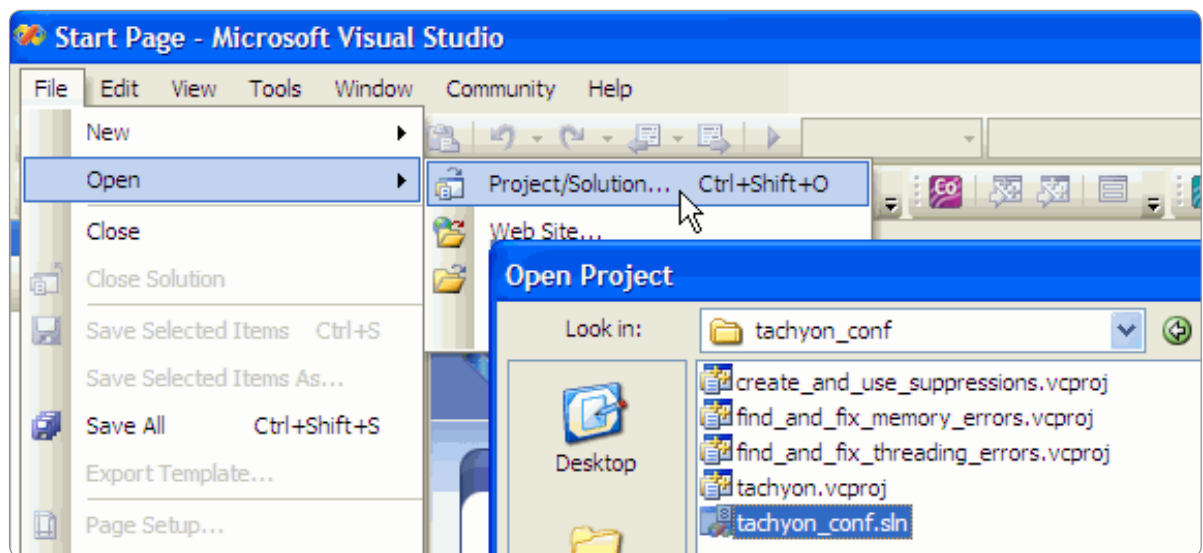


図 1

[ソリューション エクスプローラ] ペインに tachyon_conf ソリューションが表示されます。図 1

2. [ソリューション エクスプローラ] ペインで、find_and_fix_memory_errors プロジェクトを右クリックして、[スタートアップ プロジェクトに設定] を選択します。

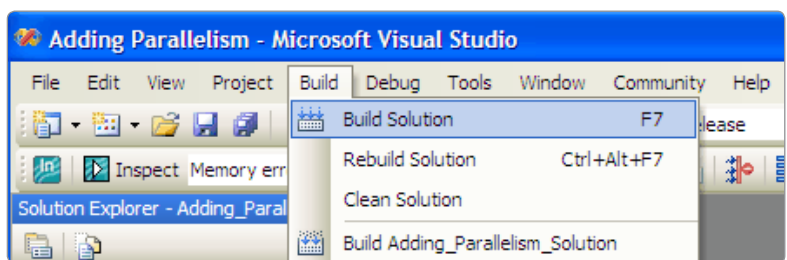


図 2

3. [ビルド] > [ソリューションのビルド] を選択してアプリケーションをビルドします。図 2

4. [デバッグ] > [デバッグなしで開始] をクリックしてアプリケーションを実行します。図 3

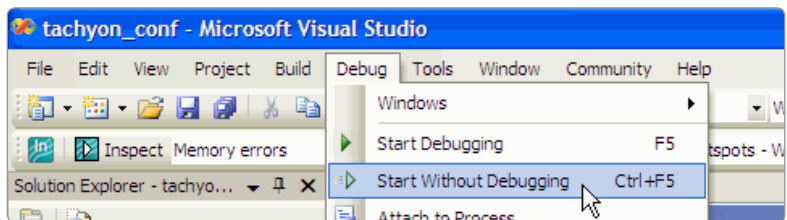


図 3

メモリーエラーの排除とプログラムの安定性の向上



ターゲットのビルド

Microsoft Visual Studio プロジェクトの設定を確認します。次に、プロジェクトをビルドして、インテル® Parallel Inspector がメモリーエラーをチェックできる実行ファイルを作成します。

ネイティブコードを含むバイナリーのデバッグモードとリリースモードの両方でインテル® Parallel Inspector を使用できます。以下のオプションを使用してデバッグモードでターゲットをコンパイル / リンクした場合、最も正確な結果が得られます。 [図 4](#)

図 4

コンパイラ / リンカーのオプション	正しい設定	設定が正しくない場合の影響
デバッグ情報	有効 (/Zi または /ZI)	ファイル / 行情報不足
最適化	無効 (/Od)	ファイル / 行情報不正
ダイナミック・ランタイム・ライブラリー	選択 (/MD または /MDd)	誤診あるいは情報不足

ターゲットのビルド

デバッグモードに設定されていることを確認するには:

1. [ソリューション エクスプローラ] ペインで、find_and_fix_memory_errors プロジェクトを右クリックして、[プロパティ] を選択します。 [図 5](#)
2. [構成] ドロップダウン・リストが [Debug]、または [アクティブ (Debug)] に設定されていることを確認します。 [図 6](#)
3. 左ペインで、[構成プロパティ] > [C/C++] > [全般] を選択します。[デバッグ情報の形式] が [プログラム データベース (/Zi)] または [エディット コンティニュー用プログラム データベース (/ZI)] に設定されていることを確認します。

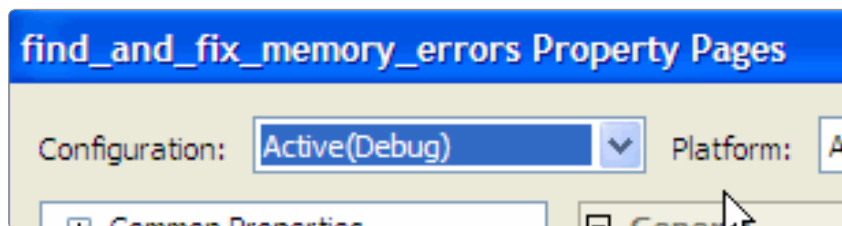


図 5

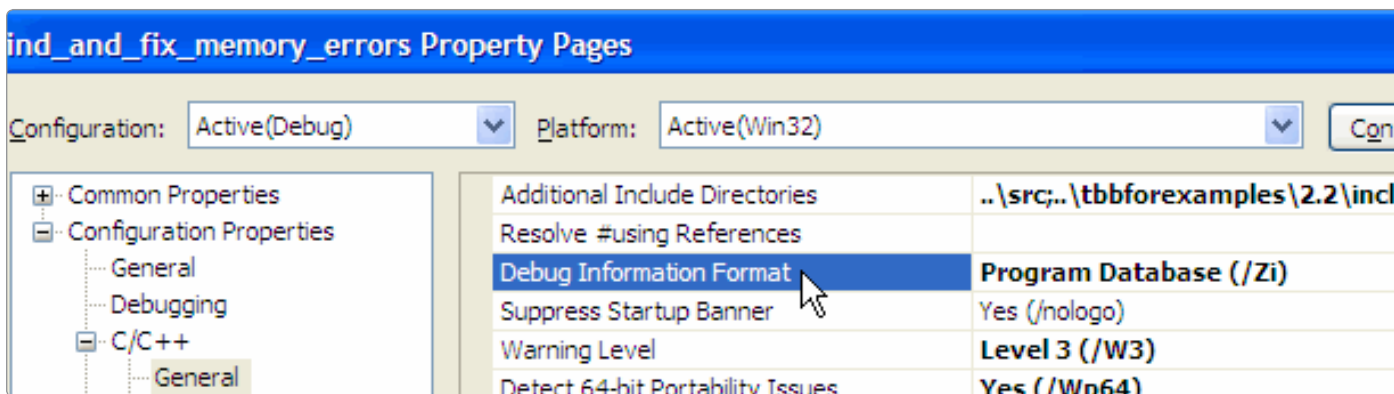


図 6

メモリーエラーの排除とプログラムの安定性の向上



図 7

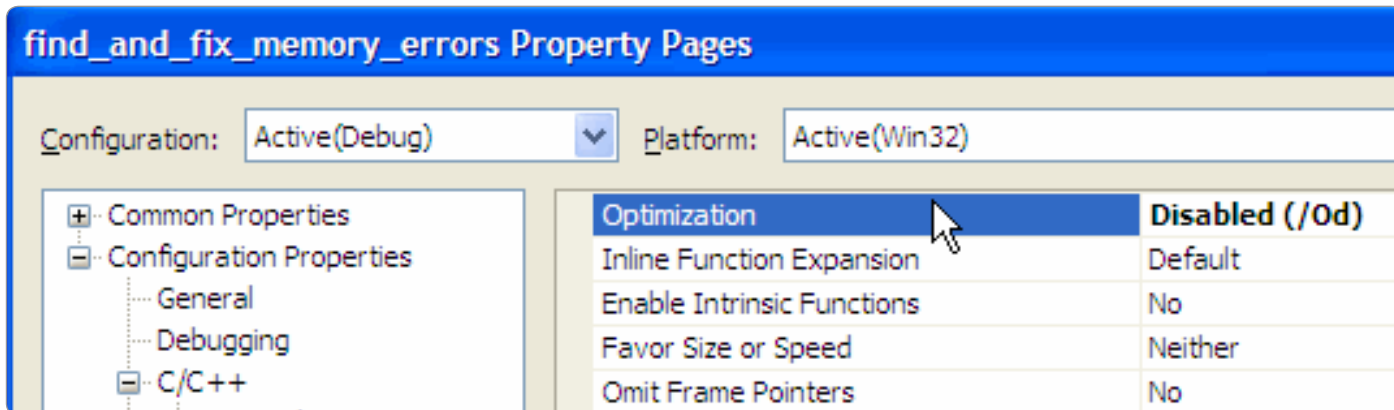
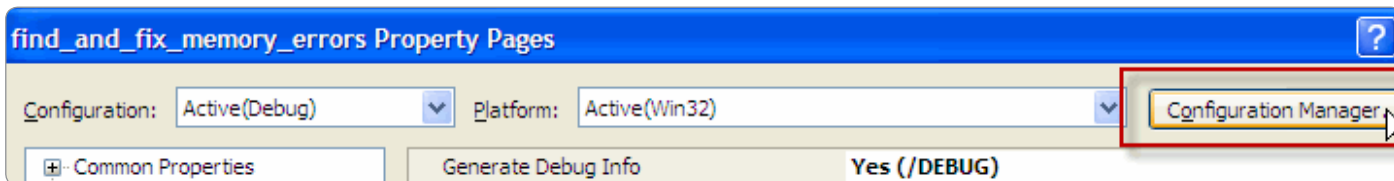


図 8



4. [構成プロパティ] > [C/C++] > [最適化] を選択します。[最適化] フィールドが [無効 (/Od)] に設定されていることを確認します。



5. [構成プロパティ] > [C/C++] > [コード生成] を選択します。[ランタイム ライブラリ] フィールドが [マルチスレッド DLL (/MD)] または [マルチスレッド デバッグ DLL (/MDd)] に設定されていることを確認します。

6. [構成プロパティ] > [リンカ] > [デバッグ] を選択します。[デバッグ情報の生成] フィールドが [はい (/DEBUG)] に設定されていることを確認します。

ターゲットがデバッグモードでビルドするように設定されていることを確認するには：

1. [プロパティ] ダイアログボックスで [構成マネージャ] ボタンをクリックします。
2. [アクティブ ソリューション構成] ドロップダウン・リストが [Debug] に設定されていることを確認します。
3. [閉じる] ボタンをクリックして [構成マネージャ] ダイアログボックスを閉じます。
4. [OK] ボタンをクリックして [プロパティ ページ] ダイアログボックスを閉じます。

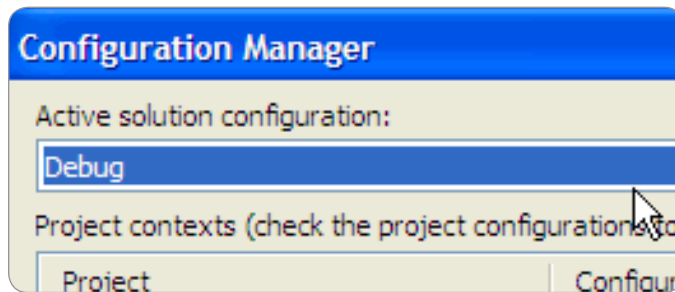


図 9

メモリーエラーの排除とプログラムの安定性の向上

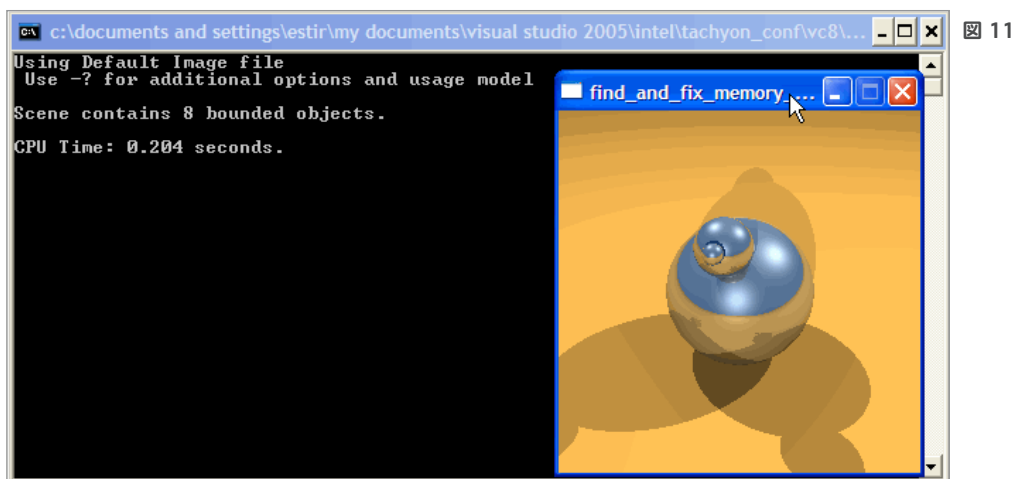
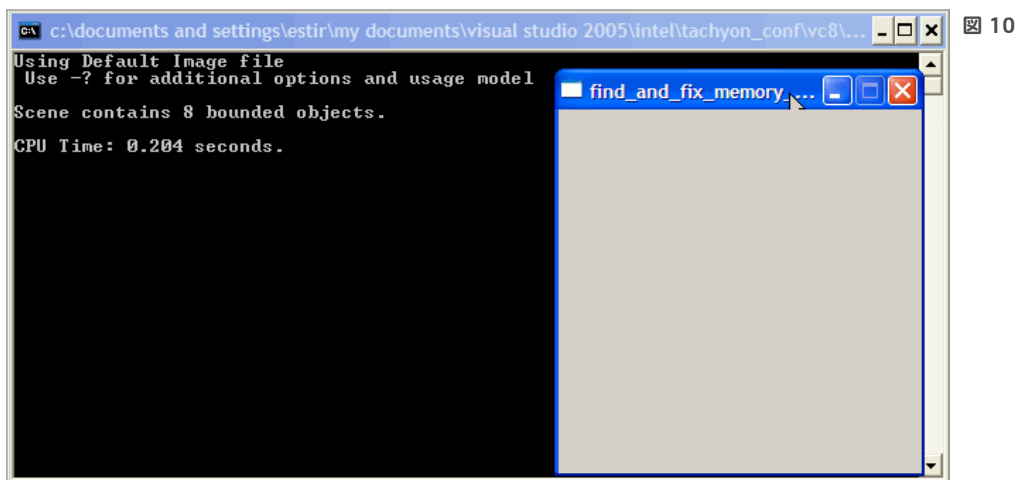


ターゲットのビルド

1. [デバッグ] > [デバッグなしで開始] を選択します。アプリケーションを開始すると、以下のよう
な画面が表示されます。

イメージはレンダリングされていません。 [図 10](#)

このアプリケーションにエラーがない場合、出力は [図 11](#) のようになります。




メモリーエラーの排除とプログラムの安定性の向上

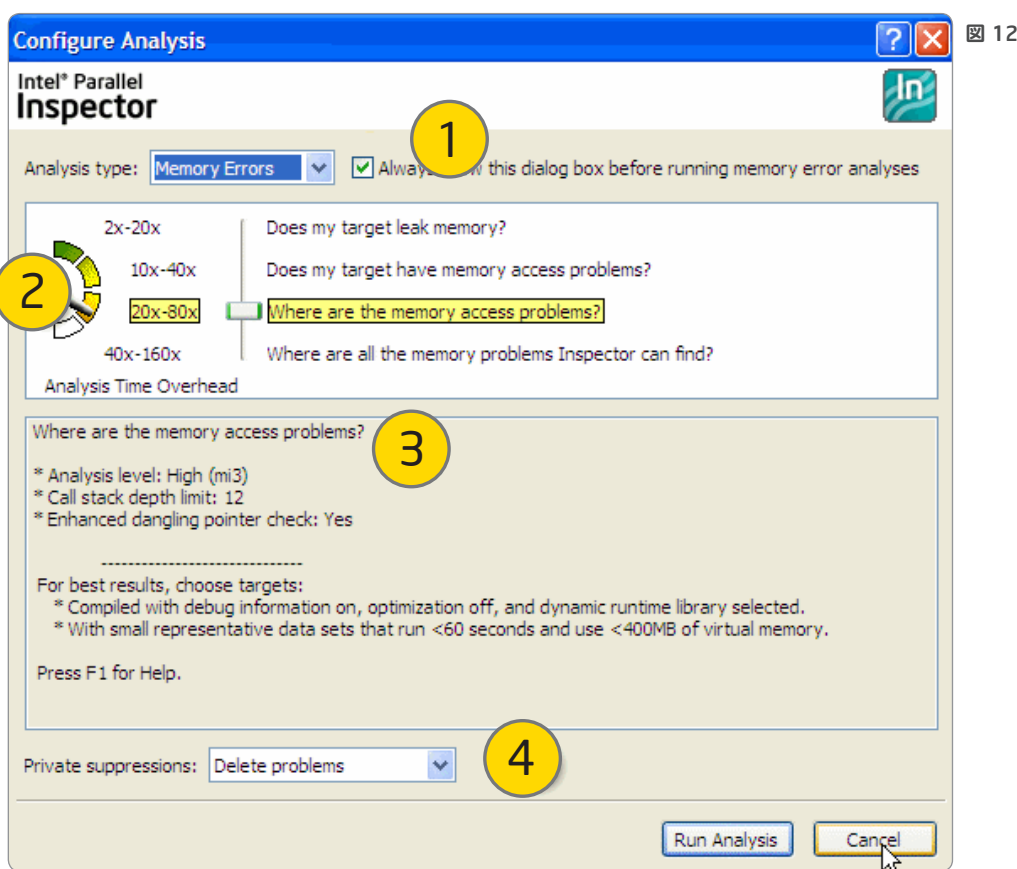


分析の設定

メモリーエラー分析の範囲と実行時間に合ったプリセット設定を選択します。

メモリーエラー分析を設定するには：

1. Microsoft Visual Studio で、[ツール] > [Intel Parallel Inspector (インテル (R) Parallel Inspector)] > [Inspect Memory Errors (メモリーエラーの検証)] を選択して、[Configure Analysis (分析の設定)] ダイアログボックスを表示します。
2. 設定スライダを [Where are the memory access problems? (メモリアクセスの問題が発生している場所は?)] プリセット設定に合わせます。  12



- 1 [Analysis type (分析の種類)] ドロップダウン・リストに、インテル® Parallel Inspector で利用可能な動的分析の種類 (メモリーエラー分析とスレッド化エラー分析) が表示されます。

このチュートリアルでは、メモリーエラー分析を説明します。メモリーエラー分析は、不正なメモリアクセス、メモリーリーク、割り当てと解放の不一致、不明な割り当て、初期化されていないメモリアクセスの検出に使用できます。

- 2 [Analysis Time Overhead (分析時間のオーバーヘッド)] ゲージは、それぞれのプリセット設定で結果の収集にかかる時間を示しています。時間は標準ターゲット実行時間との比較で表現されます。例えば、2x - 20x は標準ターゲット実行時間の 2 倍から 20 倍の長さになります。標準ターゲット実行時間が 5 秒の場合、推定収集時間は 10 秒から 100 秒になります。

デフォルトでは、設定スライダは最もオーバーヘッドの少ない [Does my target leak memory? (ターゲットはメモリーリークしているか?)] プリセット設定を指しています。


- 3 この設定詳細領域には、現在の設定についての詳細な情報が表示されます。設定スライダを移動すると、ゲージと詳細領域の内容が変更されます。

- 4 既知の問題を表示しないようにすると、生産性が大幅に向上します。詳細は、インテル® Parallel Inspector ヘルプの「suppression rule」(英語) を参照してください。

メモリーエラーの排除とプログラムの安定性の向上



分析の実行

メモリーエラー分析を実行して、解決しなければならないメモリーに関する問題を検出します。  13

メモリーエラー分析を実行するには：

[Configure Analysis (分析の設定)] ダイアログボックスの [Run Analysis (分析の実行)] ボタンをクリックします。

- find_and_fix_memory_errors.exe ターゲットを実行します。
- 問題を確認します。
- tachyon_conf/vc8/My Inspector Results フォルダーの結果を収集します。
- 結果を処理 (シンボル情報をファイル名と行番号に変換、重複結果の解決、問題セットの書式を設定) します。

収集中、インテル® Parallel Inspector は下記のような [Event Log (イベントログ)] ウィンドウを表示します。

分析完了後に結果データの管理を開始するには：

[Interpret Result (結果の解釈)] ボタンをクリックして、[Overview (概要)] ウィンドウを表示します。

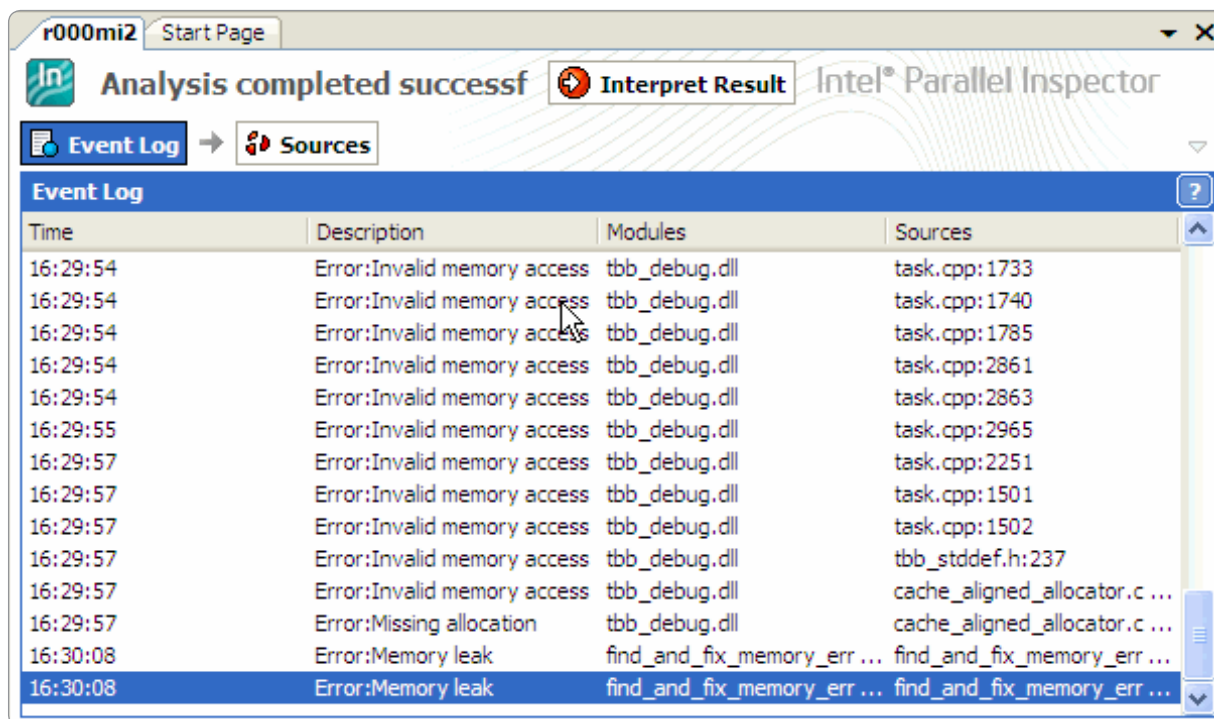


図 13

メモリーエラーの排除とプログラムの安定性の向上



問題セットの選択

[Overview (概要)] ウィンドウで検出したメモリー問題を調査する問題セットを選択します。 [図 14](#)

問題セットを選択するには：

1. [Problem Sets (問題セット)] ペインの [Sources (ソース)] 列ヘッダーをクリックして、ソースファイルの位置で問題セットをソートします。必要な場合、ペインの先頭にスクロールして、`find_and_fix_memory_error.cpp` ファイルの問題セットを検索するウィンドウを表示します。
2. `find_and_fix_memory_errors.cpp` ソースファイルの [Mismatched allocation/deallocation (割り当てと解除の不一致)] 問題セットのデータ行をダブルクリックして、[Sources (ソース)] ウィンドウを表示します。重点観測および関連観測のソースコードが表示されます。

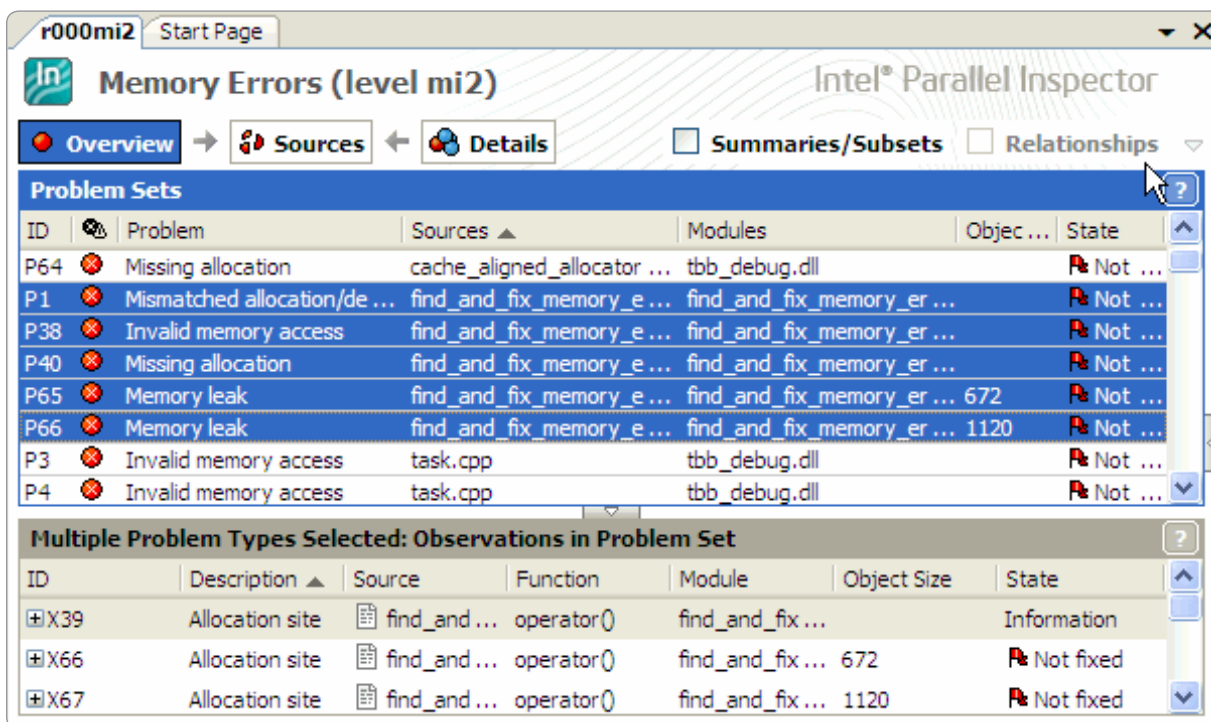


図 14

メモリーエラーの排除とプログラムの安定性の向上



find_and_fix_memory_errors.cpp ソースファイルで、[Allocation site (割り当てサイト)] と 1 つの [Mismatched deallocation site (解除の不一致サイト)] の観測を含む [Mismatched allocation/deallocation (割り当てと解除の不一致)] 問題セットの調査が開始されました。図 15

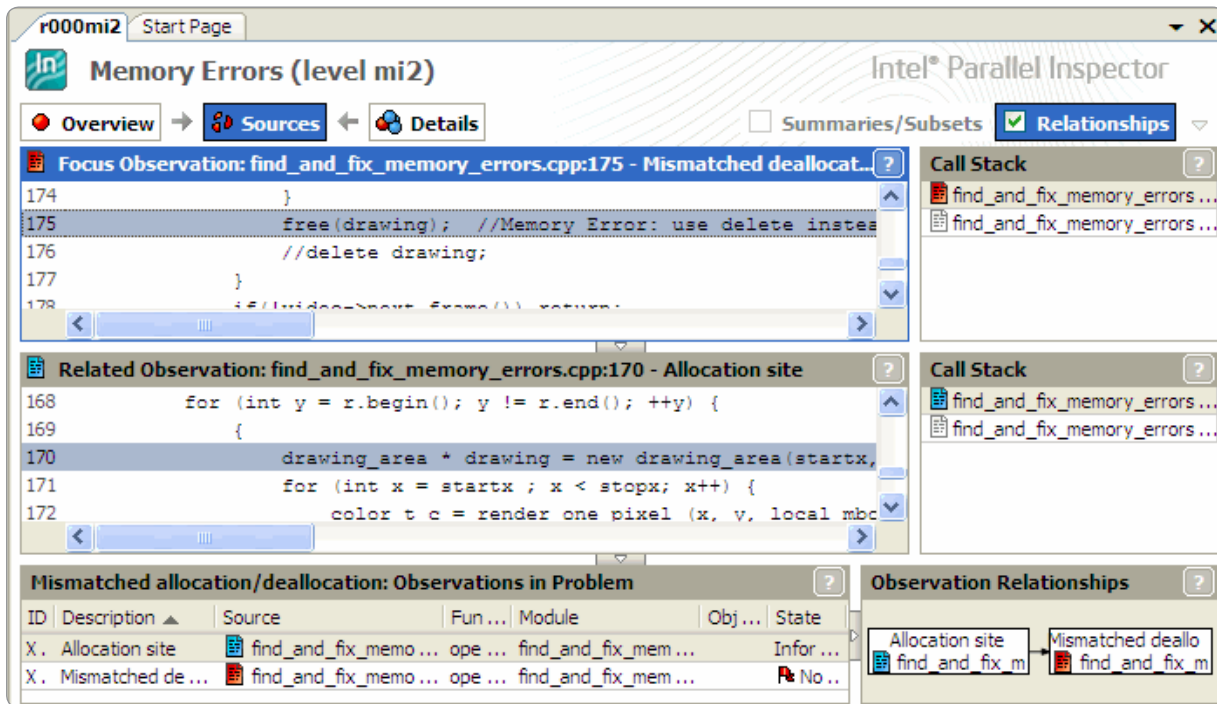


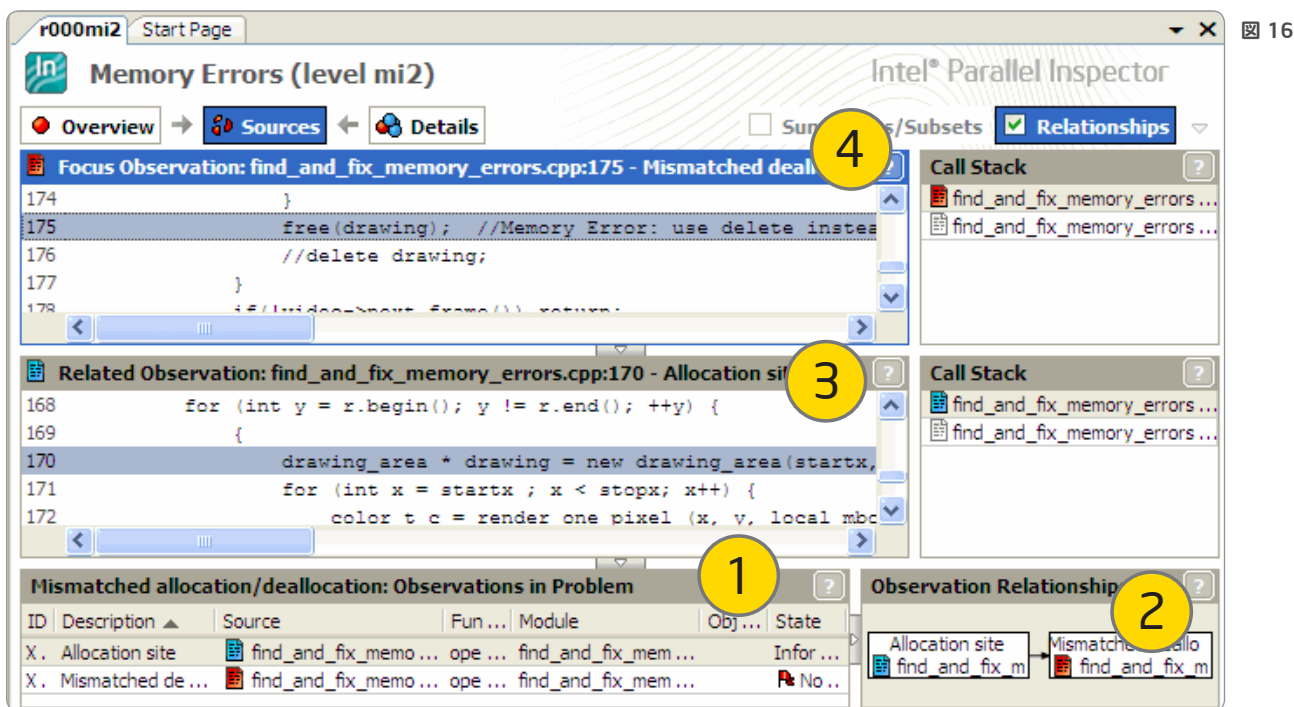
図 15

メモリーエラーの排除とプログラムの安定性の向上



結果データの解釈

[Sources (ソース)] ウィンドウのデータを解釈して、検出されたメモリー問題の原因を特定します。図 16



1 **[Observations in Problem Sets (問題セットの観測結果)]** ペインには、**[Mismatched allocation/deallocation (割り当てと解除の不一致)]** 問題セットのすべての問題の観測結果が表示されます。

[Allocation site (割り当てサイト)] の観測結果は、メモリーブロックが割り当てられた場所と関連コールスタックを表します。**[Mismatched deallocation site (解除の不一致サイト)]** の観測結果は、解除を試みた場所と関連コールスタックを表します。

2 **[Observation Relationships (観測結果の関係)]** ペインには、**[Allocation site (割り当てサイト)]** と **[Mismatched deallocation site (解除の不一致サイト)]** の観測結果の関係が表示されます。関係ダイアグラムには、次の項目があります。

- ダイアグラムの各ボックスは、問題セットの 1 つの問題の観測結果を表しています。
- 1 つのボックスしかないダイアグラムは、関連する観測結果のない小さな問題です。
- 縦に積み重ねられたボックスは、同時発生の観測結果を示しています。
- 左右に並べられ、矢印で結合されたボックスは、時間順を示しています。**[Allocation site (割り当てサイト)]** の観測結果は、**[Mismatched deallocation site (解除の不一致サイト)]** の観測結果の前に発生しています。
- 線で結合されたボックスは、関係を示しています。

3 **[Related Observation Code (関連観測結果コード)]** ペインには、**[Allocation site (割り当てサイト)]** の観測結果を含む `find_and_fix_memory_errors.cpp` ソースファイルのソースコードが表示されます。**[Allocation site (割り当てサイト)]** の観測結果に対応するソースコードはハイライトされています。

4 **[Focus Observation Code (重点観測結果コード)]** ペインには、**[Mismatched deallocation site (解除の不一致サイト)]** の観測結果を含む `find_and_fix_memory_errors.cpp` ソースファイルのソースコードが表示されます。**[Mismatched deallocation site (解除の不一致サイト)]** の観測結果に対応するソースコードはハイライトされています。

メモリーエラーの排除とプログラムの安定性の向上



問題の解釈と解決のためにより多くの情報にアクセスするには：

1. **[Observations in Problem Set (問題セットの観察)]** ペインで、**[Mismatched deallocation site (解除の不一致サイト)]** の観察を右クリックします。
2. **[Explain Problem (問題の説明)]** を選択します。**Mismatched allocation/deallocation (割り当てと解除の不一致)** 問題に関するインテル® Parallel Inspector ヘルプ情報が表示されます。

結果データを解釈するには：

[Focus Observation Code (重点観測結果コード)] ペインと **[Related Observation Code (関連観測結果コード)]** ペインでコードを確認します。

[Related Observation Code (関連観測結果コード)] ペインの **[Allocation site (割り当てサイト)]** 観測のコードには new アロケータが含まれ、**[Focus Observation Code (重点観測結果コード)]** ペインの **[Mismatched deallocation site (解除の不一致サイト)]** 観測には free() デアロケータが含まれています。

問題の解決

Microsoft Visual Studio エディターにアクセスしてメモリー問題を解決します。

問題を解決するには：

1. **[Sources (ソース)]** ウィンドウの **[Focus Observation Code (重点観測結果コード)]** ペインでハイライトされているコードをダブルクリックします。別のタブに find_and_fix_memory_errors.cpp ソースファイルが表示されます。Microsoft Visual Studio エディタを使用してコードを編集できます。 [図 17](#)
2. free(drawing); をコメントにして、//delete drawing; を生かします。

[Mismatched allocation/deallocation (割り当てと解除の不一致)]

問題は、アロケータによる割り当てが論理的に行われていない関数で解放を行うと発生します。C++ プログラミング言語では、以下の関数がペアになっています。

- > new と delete
- > new[] と delete[]
- > malloc() と free()

割り当てと一致する解放を行った場合のみ、対応するメモリー割り当てや割り当てで使用した内部データ格納領域が正しく処理されます。割り当てと一致しない解放を行うと、多くの場合メモリーの再利用で問題が発生します。

注：割り当てと解放の不一致によって必ずしもアプリケーションがクラッシュするとは限りません。ただし、無関係な場所で後からクラッシュする可能性があります。

find_and_fix_memory_errors.cpp ソースファイルの new と free が一致していないことが **[Mismatched allocation/deallocation (割り当てと解除の不一致)]** 問題セットの原因であることがわかりました。

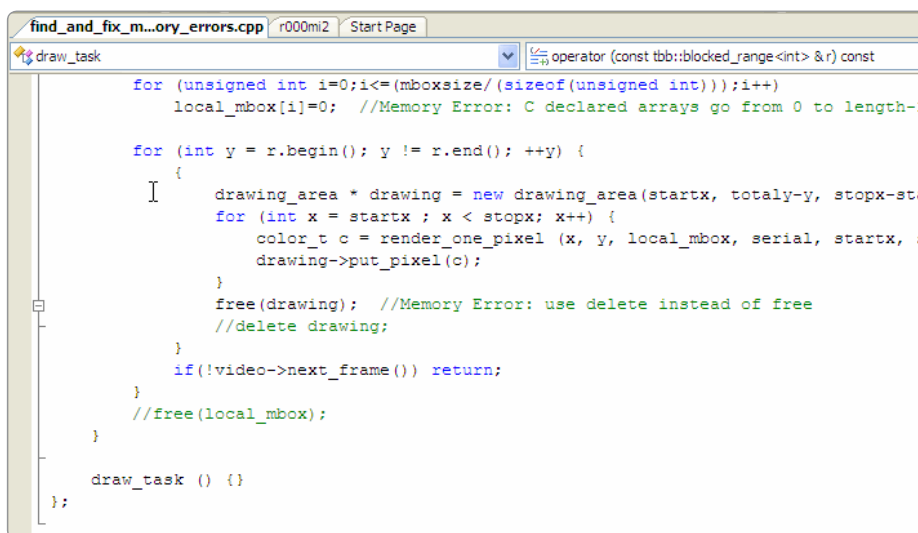


図 17

メモリーエラーの排除とプログラムの安定性の向上



リビルドと分析の再実行

ソースコードを変更してターゲットをリビルドした後、再度メモリーエラー分析を実行して、変更によってメモリーエラー問題が解決したかどうかを確認します。

ターゲットをリビルドするには：

[ソリューション エクスプローラ] ペインで、find_and_fix_memory_errors プロジェクトを右クリックして、ポップアップ・メニューから [ビルド] を選択します。

最後に実行した分析と同じ種類の設定を再実行するには：

[ツール] > [Intel Parallel Inspector (インテル (R) Parallel Inspector)] > [Re-inspect (再検証)] または [Inspect Memory Errors (メモリーエラーの検証)] を選択して、find_and_fix_memory_errors.exe ターゲットを実行します。以下のように表示されます。 [図 18](#)

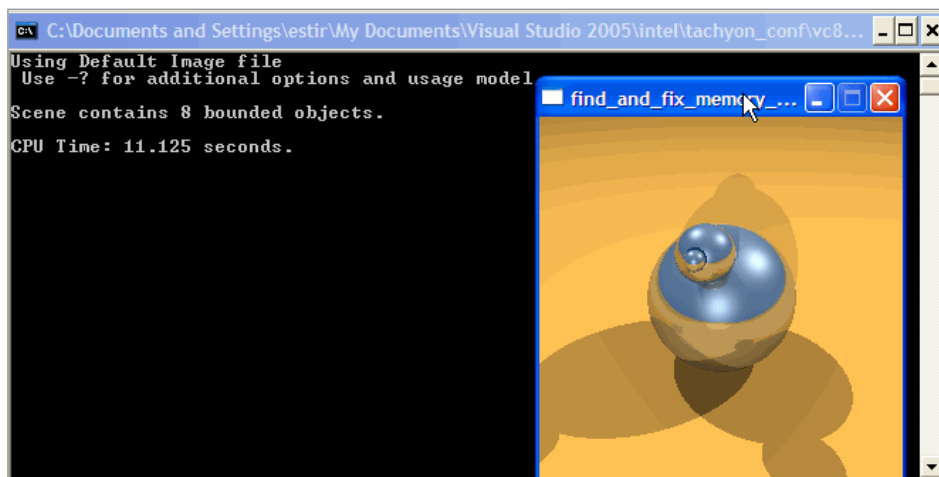


図 18: イメージが正しく表示されるようになりました。

結果

この例では、プログラムにバグがあり正しく動作しておらず、イメージはレンダリングされませんでした。インテル® Parallel Inspector を実行し、バグを修正した後、イメージは正しくレンダリングされるようになっています。ほとんどの場合、インテル® Parallel Inspector を実行するだけでこのような成果が得られます。

ただし、分析を細かく行うために非常に時間がかかることがあります。次のセクションでは、インテル® Parallel Inspector で大規模なアプリケーションを実行する場合のヒントを紹介します。

プログラムをマルチスレッド化している場合、インテル® Parallel Inspector が隠れたデータ競合やデッドロックなどのスレッドエラーも検出できることに驚かれることでしょう。

インテル® Parallel Inspector には、複数のワークロードやテストケースでアプリケーションのテストを自動化できるコマンドライン・インターフェイスも用意されています。バッチモードで一晩中実行したり、リグレッション・テスト (回帰テスト) の一部として実行できます。

メモリーエラーの排除とプログラムの安定性の向上



大規模なアプリケーション / 複雑なアプリケーションの場合のヒント

重要な概念：小さく典型的なデータセットを選択する

分析を実行するとき、インテル® Parallel Inspector はデータセットに応じてターゲットを実行します。データセットのサイズはターゲットの実行時間と分析速度に直接影響します。

例えば、1000x1000 ピクセルのイメージのほうが、100x100 ピクセルのイメージよりも処理は長くなります。大きなイメージではループで 1...1000 の反復空間が必要になるのに対して、小さなイメージでは 1...100 でかまわないことも理由の 1 つです。完全に同じコードパスを両方のケースで実行します。違いは、これらのコードパスを繰り返す回数だけです。

ターゲットから冗長な処理を省くことで、完全性を損なうことなく、分析時間を制御できます。


大きな繰り返し型のデータセットの代わりに、小さく典型的なデータセットを選択してください。数秒で実行できるデータセットが理想的です。すべてのコードが確実に検査されるように、他のデータセットを作成することもできます。

スレッド化エラーの管理

インテル® Parallel Inspector は、並列プログラムの隠れたデータ競合やデッドロックのようなスレッドエラーも特定、分析、解決します。発見、再現、修正が非常に困難な、再現性がなく、異なる結果となるようなエラーも検出できます。

コマンドラインを使用したテストの自動化

インテル® Parallel Inspector は、エラーを検出するためにコードパスを実行する必要があるため、異なるコードパスや異なるワークロードを考慮して、コードを複数回実行することになります。このため、コード検査ツールが十分な時間をかけてテストできるように、これらのテストを一晩中、あるいはリグレーション・テストの一部として実行し、コンピューターに作業させるほうがより効率的です。翌朝に複数のテストの結果を確認するだけで済みます。

インテル® Parallel Inspector のコマンドライン・バージョン (insp-cl) は、コマンドウィンドウから使用します ([スタート] > [ファイル名を指定して実行] を選択し、cmd と入力して [OK] をクリックした後、インテル® Parallel Inspector をインストールしたフォルダーのパスを入力して起動します)。  19

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\Program Files\Intel\Parallel Studio\Inspector\bin32>insp-cl --help
Usage: insp-cl [-r result-dir] [-s suppression-file] [-c config] [options] [--]
<app-name> [app options]]

-?, -h, -help           Displays brief tool description and usage.
-U, -version            Displays version information.
-q, -quiet              Display error and warning messages only.
-v, -verbose            Display info messages as well as errors and
                        warnings.
-debug                 Write debug messages to log file.
-r, -result-dir=<string> Specify directory in which to store the results
                        file.
-config-list            Display available configuration file names with
                        descriptions.
-c, -config=<string>   Specify configuration file name.
-option-file=<string>  Specify file containing list of tool options.
-command-list          Display available collector control commands.
    
```

insp-cl のヘルプを表示するには、--help と入力します。

```

> c:\Program Files\Intel\
Parallel Studio\Inspector\
bin32\insp-cl --help
    
```

図 19

メモリーエラーの排除と プログラムの安定性の向上



並列処理に関する情報

インテルでは、開発者が現在および将来のプロセッサ処理能力を活用する、正当で高性能なコードを記述できるように、並列処理に関するさまざまな情報を提供しています。インテル® Parallel Studio およびその他の関連項目についてインテル社のエキスパートが提供している情報をご活用ください。

関連リンク (英語)

[インテル® ソフトウェア・ネットワーク・フォーラム](#)

[インテル® ソフトウェア開発製品ナレッジベース](#)

[インテル® ソフトウェア・ネットワーク・ブログ](#)

[インテル® Parallel Studio Web サイト](#)

[インテル® TBB Web サイト](#)

[Go Parallel – 並列化に関するブログ、記事、ビデオ](#)

[開発者向け無料 Web セミナー \(オンデマンド\)](#)

その他の導入ガイド

[hotspot の特定と最適化](#)

[並列化による既存プログラムの最適化](#)