



メモリー/スレッドデバッガー インテル® Inspector XE 2013

製品紹介

主な機能

- C、C++、C#、および Fortran アプリケーションのエラーを検出
- 特別なビルドは必要なし。通常のコンパイラーとビルドを使用可能。
- ソースが利用できない場合でもすべてのコードを検証
- 複数のソースコード位置のエラーと対応するコールスタックを表示
- 新機能! 複雑なエラーの診断を容易にするデバッガー・ブレイクポイント
- 新機能! ヒープメモリー使用量が増え続ける原因を特定するヒープ増加解析

「我々はクラッシュ問題に 1 週間かけて取り組み、メモリー破壊が原因であることは特定できましたが、ソースの解析は非常に困難でした。インテル® Inspector XE を実行したところ、実際にクラッシュが発生する場所よりもかなり前に境界外の配列があることがすぐに判明し、問題解決までの時間を大幅に短縮できました。」

Envivio
シニア・コーデック・アーキテクチャー・エンジニア
Mikael Le Guerroué 氏

インテル® Inspector XE は、以下のスイートに含まれています。

- インテル® Parallel Studio XE
- インテル® C++ Studio XE
- インテル® Fortran Studio XE
- インテル® Cluster Studio XE

サポートされるオペレーティング・システム:

- Windows*
- Linux*

より信頼性の高いアプリケーションを提供

インテル® Inspector XE 2013 は、Windows*/Linux* 向けの簡単に使用できるダイナミック・メモリー・エラーおよびスレッドエラー検出ツールです。生産性の向上、コストカット、製品開発期間の短縮を実現します。

開発サイクルの初期段階でメモリー/スレッドの不具合を発見

できるだけ早い段階でエラーを発見することで、エラーの修正にかかるコストを抑えられます。インテル® Inspector XE を使用すると、エラーを容易に発見し診断できます。

従来のリグレッション・テストやスタティック解析では検出できないエラーを発見

インテル® Inspector XE は、コード内のエラーを引き起こす箇所が実行されない場合でも、潜在的なエラー、間欠的なエラー、非決定性的エラーを発見します。

メモリーエラー	スレッドエラー
<ul style="list-style-type: none"> ▪ メモリーリーク ▪ メモリー破壊と不正なアクセス ▪ メモリーの割り当てと解放を行う API の不一致 ▪ 一貫していないメモリー API の使用 	<ul style="list-style-type: none"> ▪ データ競合 <ul style="list-style-type: none"> - ヒープ競合 - スタック競合 ▪ デッドロック

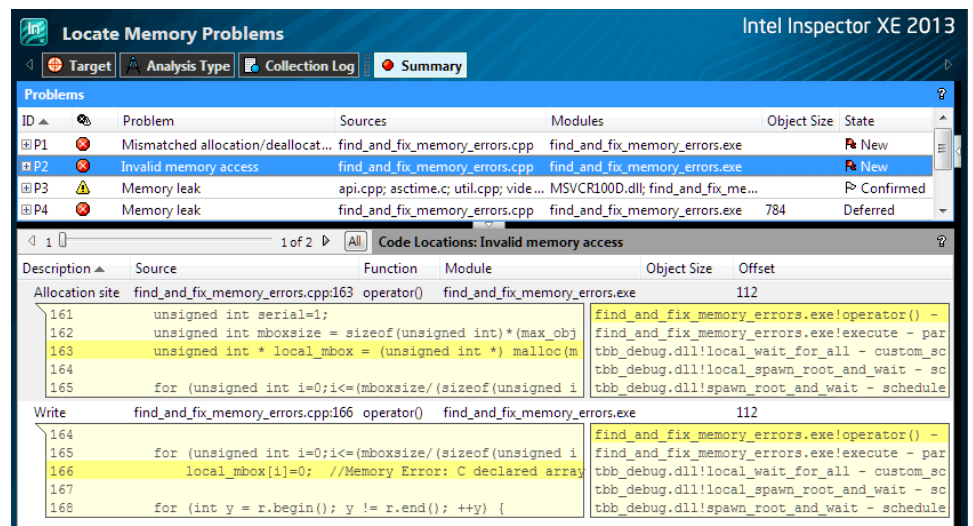
C、C++、C#、Fortran、またはこれらの言語が混在したアプリケーションに対応

C# の GUI とパフォーマンスの影響を受けやすい C++ のコードを混合していますか? Fortran ソースに古いコードが含まれていますか? ソースのないライブラリーを使用していますか? すべて問題ありません。動的インストルメンテーションにより、ソースのないサードパーティーのライブラリーを含む、すべてのコードを検査できます。

さまざまなワークフローに対応

特別なコンパイラーやビルドは必要ありません。通常のデバッグビルドまたはプロダクション・ビルドを使用可能です。デバッグしているコードも出荷するコードも検査できます。GUI または自動化されたリグレッション・テスト (コマンドライン) を使えます。ユーザー・インターフェイスは、スタンドアロン (Windows* および Linux*)、または Microsoft* Visual Studio* に統合して使用することができます。

追加機能: スイートを購入すると、スタティック解析とポインターチェッカーを利用できます。インテル® Inspector XE の使用にインテル® コンパイラーは必要ありませんが、スイートを購入するとインテル® コンパイラーの追加機能をご利用になれます。スタティック解析はセキュリティー問題を発見します。ポインターチェッカーは、割り当てられたメモリーアドレスを超えてアクセスするメモリーを検出するコンパイラーベースの診断機能です。



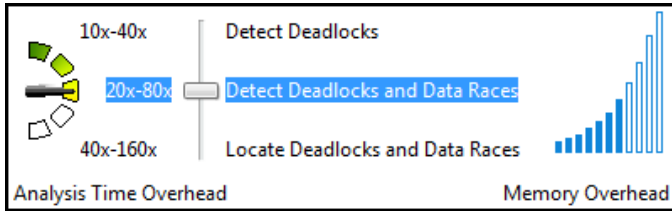
インテル® Inspector XE は、エラーの場所を特定できるように、スレッドエラーとメモリーエラーのソース場所とコールスタックを表示します。

主な機能

ID	Problem	Sources
P1	Mismatched allocation/	
P2	Invalid memory access	View Source Edit Source Copy to Clipboard Explain Problem Debug This Problem
P3	Memory leak	
P4	Memory leak	
P5	Memory leak	
P6	Memory growth	

レベルの選択 - 速度優先または検出優先

解析の最初のレベルは、オーバーヘッドが最も少ないレベルです。このレベルは高速であるため、開発中に使用します。2番目のレベル(左の画面)は、より時間がかかりますが、より多くの問題を検出します。新しい機能の確認によく使用します。3番目のレベルはリグレッション・テストやバグの検出に最適です。



診断を簡素化するデバッガー・ブレイクポイント

デバッガー・ブレイクポイントを使用すると、エラーが発生する直前にデバッガーに移るため、複雑なエラーの診断が容易になります。変数を調べてほかのスレッドを確認し、問題を診断します。Microsoft* Visual Studio*、GDB および IDB デバッガーがサポートされています。エラーを選択して、ポップアップ・メニューから [Debug This Problem (この問題のデバッグ)] を選択するだけです。

Diagnose Heap Growth

Reset Leak/Growth Detection

Show Leaks/Growth Now

ヒープ増加の解析と新しい“オンデマンド”リークレポート

アプリケーションの実行中、ヒープメモリ使用量が増え続ける原因を特定するのに役立ちます。ヒープ増加レポートにより、最後のベースライン以降に解放されていない割り当てがわかります。新機能として、最後のベースライン以降のリークを示すオンデマンドのレポートも利用できるようになりました。

一時停止/再開で解析をスピードアップ

```
__itt_suppress_push(__itt_suppress_threading_errors);
/* すべてのスレッドエラーを表示しません */
__itt_suppress_pop();
/* すべてのスレッドエラーを表示します */
```

一時停止/再開で解析をスピードアップ

範囲を限定して解析をスピードアップします。疑わしい箇所のみを解析の対象にできます。長いコードの無関係のセクションをスピードアップしてインストルメンテーションのオーバーヘッドを減らします。上手に使用することにより、エラーが発生するまで長い実行時間がかかる状況で役立ちます。

正確に非表示にすることで安全に誤検出を排除

```
Suppression = {
  Name = "Example";
  Type = { uninitialized_memory_access }
  Stacks = {
    {
      mod=a.out, func=update_x;
      func=main;
    }
  }
}
```

誤検出を非表示にしてチームで共有

誤検出を非表示リストに追加することにより、次回以降に調査する必要がなくなります。複数のリストをプロジェクトのほかのメンバーと共有できます。独自の非表示リストを作成して、コードに含まれないエラーをブロックします。モジュール全体を非表示にすると、収集時間が短縮されます。

ID	Problem	State
P1	Mismatched allocatio...	New
P2	Mismatched allocatio...	Not fixed
P3	Kernel resource leak	Confirmed
P4	Kernel resource leak	Not a problem
P5	Kernel resource leak	Fixed
P6	Invalid memory access	Fixed

チームの共同作業

レポートされるエラーには、エラーのソートや優先順位を設定できるように、エラーのステート情報 (New (新規)、Confirmed (確認済み)、Fixed (修正済み)、Not Fixed (未修正)、Regression (リグレッション)、Not a Problem (問題なし)、Deferred (遅延)) が含まれます。ステート情報は、複数の開発者間でマージおよび共有できます。オプションで、エラーに特定の状態がセットされた理由をコメントとして追加できます。

Filter to only show errors from one source file

Filters	Sort
Investigated	1 item(s)
Not investigated	3 item(s)
Source	All
find_and_fix_memory_errors.cpp	4 item(s)
Module	
find_and_fix_memory_errors.exe	4 item(s)

フィルターでエラーリストを管理

特定のソースファイルのエラーのみを表示したい場合、クリックひとつで設定できます。新しいエラーの中で最も重要度の高いエラーも、クリックひとつで表示できます。フィルターは、重要度、問題タイプ、ステート、モジュールなどのカテゴリで利用できます。フィルターを使うことにより、特定のカテゴリの最も重要なエラーに注目することができます。

Pointer Checker**

診断が困難な範囲外アクセスを検出

ポインターチェッカー** は、割り当てられたメモリーアドレスを超えてアクセスするメモリーを検出するコンパイラベースの診断機能です。ダングリング・ポインターやバッファ・オーバーフローなどを発見します。例外が発生するとデバッガーが起動され、迅速な診断が可能です。

**インテル® Parallel Studio XE ファミリーのツールスイートを購入した場合に利用可能な追加機能です (詳細は、4 ページを参照してください)。

詳細

より少ない誤検出、より分かりやすいエラーメッセージ

インテル® Inspector XE は、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® OpenMP*、およびインテル® Cilk™ Plus の並列プログラミング・モデルのセマンティクスを解釈するため、短時間で解析が可能です。

- 1) ほかの製品よりも誤検出が少ない
- 2) エラーメッセージは内部のランタイムラベルではなく、ソースの用語を使用して表示される

動的インストルメンテーション: シンプルで、信頼性が高く、正確

ほかのメモリ/スレッド解析ツールと異なり、インテル® Inspector XE では解析のために特別な再コンパイルを行う必要がありません。通常のデバッグビルドやプロダクション・ビルドをそのまま使用できます。(ソースをマップできるようにシンボルを追加してください。) ワークフローを高速かつ容易にするだけでなく、信頼性と精度も向上します。一般的なスタティック・コンパイル・ベースのインストルメンテーション・テクニックを使用する製品では、動的に生成またはリンクされたコードの解析は困難です。インテル® Inspector XE は、ソースが利用できないサードパーティーのライブラリーを含む、すべてのコードを解析できます。

複数の OS (Windows* および Linux*) で同じユーザー・インターフェイス

Windows* と Linux* の両方で開発を行っている場合、両方の OS で同じ解析ツールを使用できると便利です。インテル® Inspector XE は、Windows* および Linux* で同じユーザー・インターフェイスを採用しています。Windows* では、スタンドアロンまたは Microsoft* Visual Studio* に統合して利用できます。

MPI アプリケーションのメモリエラーとスレッドエラーを解析

マルチコアおよびハイパースレディング対応のシステムの増加に伴い、MPI アプリケーションでもスレッドによる並列処理が実装される機会が増えています。インテル® Inspector XE は、MPI アプリケーションのメモリエラーとスレッドエラーを検出できます。単一共有メモリシステムで初期解析を実行してメモリエラーを特定した後、クラスターで追加の解析を実行します。結果はランクによりソートされます。

新機能

特徴	利点
デバッガー・ブレイクポイント	デバッガー・ブレイクポイントを使用すると、エラーが発生する直前にデバッガーに移るため、複雑なエラーの診断が容易になります。変数を調べてほかのスレッドを確認し、問題を診断します。Visual Studio*、GDB および IDB デバッガーがサポートされています。
ヒープ増加の解析	アプリケーションの実行中、ヒープメモリ使用量が増え続ける原因を特定するのに役立ちます。ヒープメモリ拡張解析は、GUI または API で指定された範囲で解放されないメモリ割り当てのリストを作成します。
オンデマンド・リーク・レポート★	プログラムの実行中にいつでも、素早くリークレポートを表示します。リーク解析のベースラインを設定/リセットして、ベースライン以降のリークのみを表示できるため、一定の時間内または一定のコード領域内のリークだけに集中することができます。GUI ボタンまたは API 呼び出しから利用できます。
一時停止/再開 API	範囲を限定して解析をスピードアップします。API を使用して、疑わしい箇所のみを解析の対象にできます。上手に使用することにより、エラーが発生するまで長い実行時間がかかる状況で役立ちます。
解析範囲の限定	API を使用して解析しないメモリ (例: 同期データ構造を含む箇所) をマークすることにより、誤検出を減らして解析をスピードアップします。
正確な非表示設定★	1 つまたは複数の特定のスタックフレームを非表示にすることができます。潜在的な本当のエラーを非表示にすることなく、誤検出を排除します。非表示リストはチームで共有できます。
互換性のある非表示設定★	インテル® Inspector XE は、Purify* や Valgrind* のような一般的なメモリデバッガーから非表示リストをインポートできるため、非表示リストを作成し直す必要がありません。インポートには、便利なテキスト形式を利用できます。
チームの共同作業の改善	チームメンバーが変更したエラーのステート情報 (確認済み、問題なし、修正済み、その他) をチームメンバー間でマージおよび共有できます。ステート情報にコメントを追加できるようになりました。
インテル® Xeon Phi™ コプロセッサ向けソフトウェアの検証	インテル® Inspector XE は、インテル® Xeon Phi™ コプロセッサ上で解析を実行しなくても、インテル® Xeon Phi™ コプロセッサ向けソフトウェアを解析できます。インテル® Inspector XE を使用してマルチコア・プロセッサ上で実行するアプリケーションを検証すると、インテル® Xeon Phi™ コプロセッサ上で実行したときに発生するメモリエラーとスレッドエラーを検出します。
MPI アプリケーションの解析	MPI アプリケーションのメモリエラーを検出します。MPI および OpenMP* を使って記述されたハイブリッド・アプリケーションのメモリエラーとスレッドエラーを発見します。クラスターにも簡単にインストール可能です。結果はランクによりソートされます。
OpenMP* 4.0★	インテル® Inspector XE は OpenMP* のセマンティクスを解釈するため、エラーメッセージを内部のランタイムラベルではなく、ソースの用語を使用して表示し、誤検出も抑えます。

★ - バージョン 2013 初期リリース以降の新機能。新しい機能は継続的に製品アップデートで提供されます。有効なサポートサービスをお持ちの場合は、製品アップデートをご利用いただけます。

購入方法:言語別のスイート

アプリケーションをビルド、検証、チューニングする複数のツールが組み合わされた次のスイートがご利用になれます。本資料で説明している製品は青でハイライトされています。ライセンスは、シングルユーザー・ライセンス、フローティング・ライセンス、アカデミック・ライセンスが用意されています。

スイート>>	インテル® Cluster Studio XE	インテル® Parallel Studio XE	インテル® C++ Studio XE	インテル® Fortran Studio XE	インテル® Composer XE	インテル® C++ Composer XE	インテル® Fortran Composer XE
インテル® C/C++ コンパイラー	●	●	●		●	●	
インテル® Fortran コンパイラー	●	●		●	●		●
インテル® IPP	●	●	●		●	●	
インテル® MKL	●	●	●	●	●	●	●
インテル® Cilk™ Plus	●	●	●		●	●	
インテル® TBB	●	●	●		●	●	
インテル® Inspector XE	●	●	●	●			
インテル® VTune™ Amplifier XE	●	●	●	●			
インテル® Advisor XE	●	●	●	●			
スタティック解析	●	●	●	●			
インテル® MPI ライブラリー	●						
インテル® Trace Analyzer & Collector	●						
Rogue Wave IMSL* ライブラリー ²							●
オペレーティング・システム ¹	W、L	W、L	W、L	W、L	W、L	W、L、O	W、L、O

注: ¹ オペレーティング・システム: W=Windows、L=Linux、O=OS X* ² インテル® Visual Fortran Composer XE Windows* 版 IMSL* 同梱で利用可能

技術仕様

概要	
プロセッサのサポート	インテルの命令セットを含むアプリケーションの解析には、インテル® プロセッサおよび互換プロセッサがサポートされます。
オペレーティング・システム	Windows* および Linux*
開発ツールと環境	プラットフォームの標準に準拠するコンパイラー (Microsoft*, GCC*, インテル® コンパイラーなど) を使用できます。Microsoft* Visual Studio* 2008、2010、2012 に統合できます。詳細は、 http://www.intel.com/software/products/systemrequirements をご覧ください。
プログラミング言語	C、C++、C#、Fortran
システム要件	詳細は、 http://www.intel.com/software/products/systemrequirements をご覧ください。
サポート	すべての製品アップデート、インテル® プレミアサポート、インテル® サポートフォーラムを利用可能な1年間のサポートが含まれます。インテル® プレミアサポートは、セキュアな Web ベースで技術者からのサポートを受けられます。
コミュニティー	インテル® サポートフォーラム・コミュニティーでは、役立つ情報を取得、交換、参照することができます。 http://software.intel.com/en-us/forums



インテル® Inspector XE の詳細:

- 以下の Web サイトをご覧ください。
<http://intel.ly/inspector-xe>
- あるいは、左の QR コードをスキャンしてください。



30 日間の評価版:

- <http://intel.ly/sw-tools-eval> の Web サイトで、「Memory and Thread Debuggers」をクリックしてください。

最適化に関する注意事項

改訂 #20110804

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に関する詳細は、該当製品のユーザーズガイドまたはリファレンス・ガイドを参照してください。

