



Visual Studio 2010 による迅速なバグ修正

2010 年 3 月

コードのデバッグには膨大な時間が費やされます。単純な問題でも解決するには時間がかかります。Visual Studio 2010 Ultimate の新しいデバッグ ツールを使用すると、開発者はどのようなデバッグでも非常に効率よく作業をすることができます。さらに、必要とする時間は従来に比べて大幅に削減できます。捉えどころのない再現しないエラーや、規則性がなく発生するエラーを迅速に取り除くことで、時間とコストを抑え、顧客満足度を高めることができます。

を高めることができます。

要する時間は従来に比べて大幅に削減できます。捉えどころのない再現しないエラーや、規則性がなく発生するエラーを迅速に取り除くことで、時間とコストを抑え、顧客満足度を高めることができます。

要する時間は従来に比べて大幅に削減できます。捉えどころのない再現しないエラーや、規則性がなく発生するエラーを迅速に取り除くことで、時間とコストを抑え、顧客満足度を高めることができます。

ソフトウェア ビジネスで最も重要な考慮事項は、品質です。昔から言われているように、早い段階で欠陥を発見できなければ後で莫大な損害が生じます。たとえばアジャイル コミュニティでは、"早めにエラーを発生させること" が重要なスローガンになっています。そのため、継続的インテグレーションとテストが基本的な手法として行われています。Visual Studio 2010 は、テストと品質のための革新的技術を積極的に提供し、ソフトウェア チームの優れた成果に貢献します。ホワイトペーパー『高品質なアプリケーションの構築』も参照してください。

Visual Studio 2010 リリースの重要なテーマの 1 つは "再現不可能なエラーをなくす" ことです。つまり、テスト担当者によって発見されたバグは、バグの修正を担当する開発者によって必ず再現できなければなりません。バグを再現できないことがこのように重大な問題となるのはなぜでしょうか。それは、バグの修正にコストと時間がかかるからです。最悪の場合には、開発者の制御下にある環境内でバグを再現できないためにバグを修正できないということにもなりかねません。この問題の原因は、多くの場合、バグ報告の仕方にあります。たとえば、バグが発見されたことが開発者に電話で伝えられる場合、どんな状況で発生したのか、あいまいな情報しか伝わらない可能性があります。また、あるコンピュータではバグが発生し、別のコンピュータでは発生しない場合もあります。よくある "このマシンでは発生した" という現象です。Visual Studio 2010 は、こうした問題の根本的な解決に向けて飛躍的に進歩しています。

具体性のあるバグ

再現性の問題を解決する方法にはいくつかのアプローチがありますが、まずは、実行可能な何らかの手段によって確実に再現できるようにバグをシステムに登録することが重要です。開発者には、具体性があり実行可能な情報を何らかの形で正確に提供する必要があります。バグが発生した状況を漠然と説明するだけでは不十分です。Microsoft Visual Studio Test Professional 2010 では、Microsoft Test Manager (Visual Studio 2010 Ultimate にも同梱されています) の導入によって、このアプローチを実現できます。Microsoft テスト ランナーを使用して手動でテストを実行する場合、また自動テストや機能テストを実行する場合に、テスト担当者は実行するテストのビデオを録画することができます。テストが失敗した場合、開発者にそのビデオを提供して、バグが発生するまでの状況や、その時点でユーザーが実行していた操作を、正確に示すことができます。この情報はバグを修正するうえで非常に重要であり、データ コレクターを使用して取得します。

データ コレクターを使用すると、開発者の作業を支援するために必要なデータが、テスト中のシステムからテスト インフラストラクチャによって取得されます。ビデオ録画機能、テスト影響分析コレクターなど、いくつかの組み込みのデータ コレクターのほかに、システム情報コレクターや IntelliTrace™ コレクターなど、よりわかりやすいツールもあります。これらの標準ツールで、これまで開発者からは利用できなかったデータを提供できるようになりますが、それだけでなく、コレク

ターは拡張が可能です。開発者個人またはチームで独自のコレクターを作成して必要な情報を収集できます。これにより、問題の原因を把握するための情報収集に、無限の可能性がもたらされます。

Visual Studio 2010 からテストを実行する場合も、長時間のパフォーマンス テストや、チーム ビルドから起動するテストで、データ コレクターを使用できます。データ コレクターの使用は Test Manager に限定されません。

バグの検出

開発者は、現象を理解できたら、その問題をデバッグし、コード内で修正が必要な箇所を具体的に見つけ出す必要があります。コードのデバッグは難しい作業になることがあります。見慣れない古いコードの場合はなおさらです。必然的に、デバッグには相当な時間が必要です。報告された問題を開発者が再現できないために、結局、問題が頻発して容易に追跡できるようになるまでバグが修正されず放置される場合も少なくありません。IntelliTrace™ は、開発者がセッションを "再生" するための新しいツールを提供します。再生するセッションは、直前のデバッグ セッションや、テスト担当者がコードに対して実行したテスト ケースの結果です。つまり、開発者はアプリケーションがセッション内のどの時点でどのように動作していたかを後から再生して確認することが可能です。開発者は、ソースを見ることなく、問題の詳細を特定できることもあるでしょう。

これにより、開発者の効率性が大きく向上します。エラーを再現するために、何度もアプリケーションを実行する必要がなくなります。エラーが発生したセッションを再生だけで済みます。さらに、開発者は新しいアプリケーション インスタンスを起動し、その実行中のアプリケーションに IntelliTrace™ ログをアタッチし、アプリケーションを 1 ステップずつ実行することで、記録されたセッションをデバッグすることもできます。図 1 は、IntelliTrace™ ログの要約ページです。このページを見ると、非常に多くの情報が収集され、開発者に提供されることがわかります。

IntelliTrace™ ログには実行された全スレッドが表示され、開発者は目的とするスレッドを選ぶことができます。テスト ランナーを使った手動テストのセッションからバグが報告されている場合、開発者は最初から 1 ステップずつ実行するのではなく、実行中だったコード セクションにジャンプし、そこから調査をすることができ

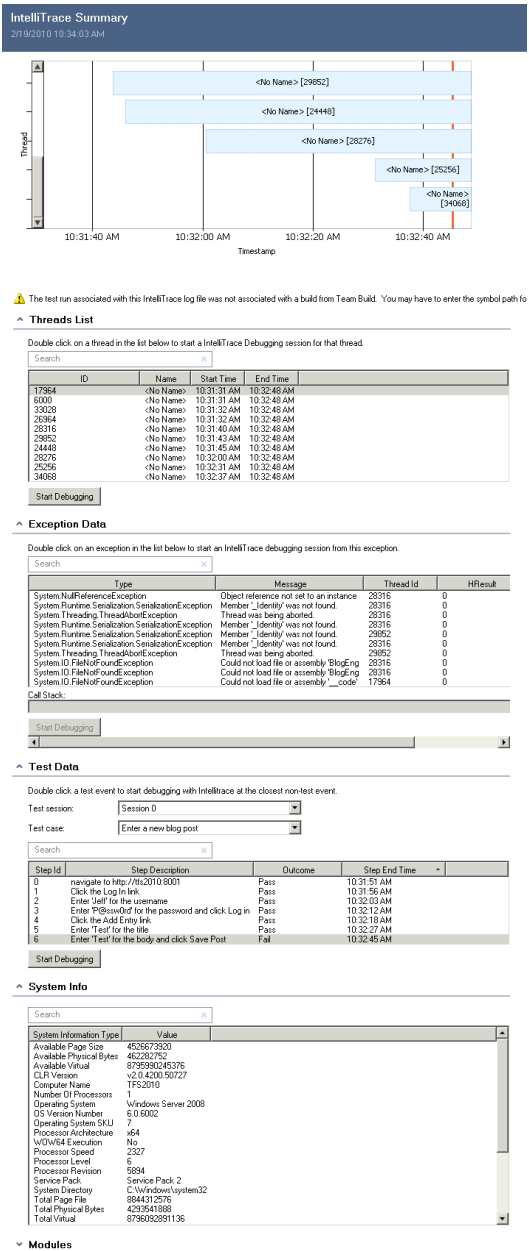


図 1 – IntelliTrace™ ログ

ます。IntelliTrace™ によってバグの検出に必要な時間が短縮されるだけでなく、この作業での担当者のフラストレーションも少なくなります。

バグの修正

Visual Studio 2010 Ultimate の IntelliTrace™ 機能は、豊富な情報をすばやく提供し、問題にかかわる開発者とテスト担当者の関係を円滑化します。1 人で作業する開発者なら、専属のテスト専門技術者が存在するかのように感じるかもしれません。図 2 は、IntelliTrace™ によるデバッグ セッションです。

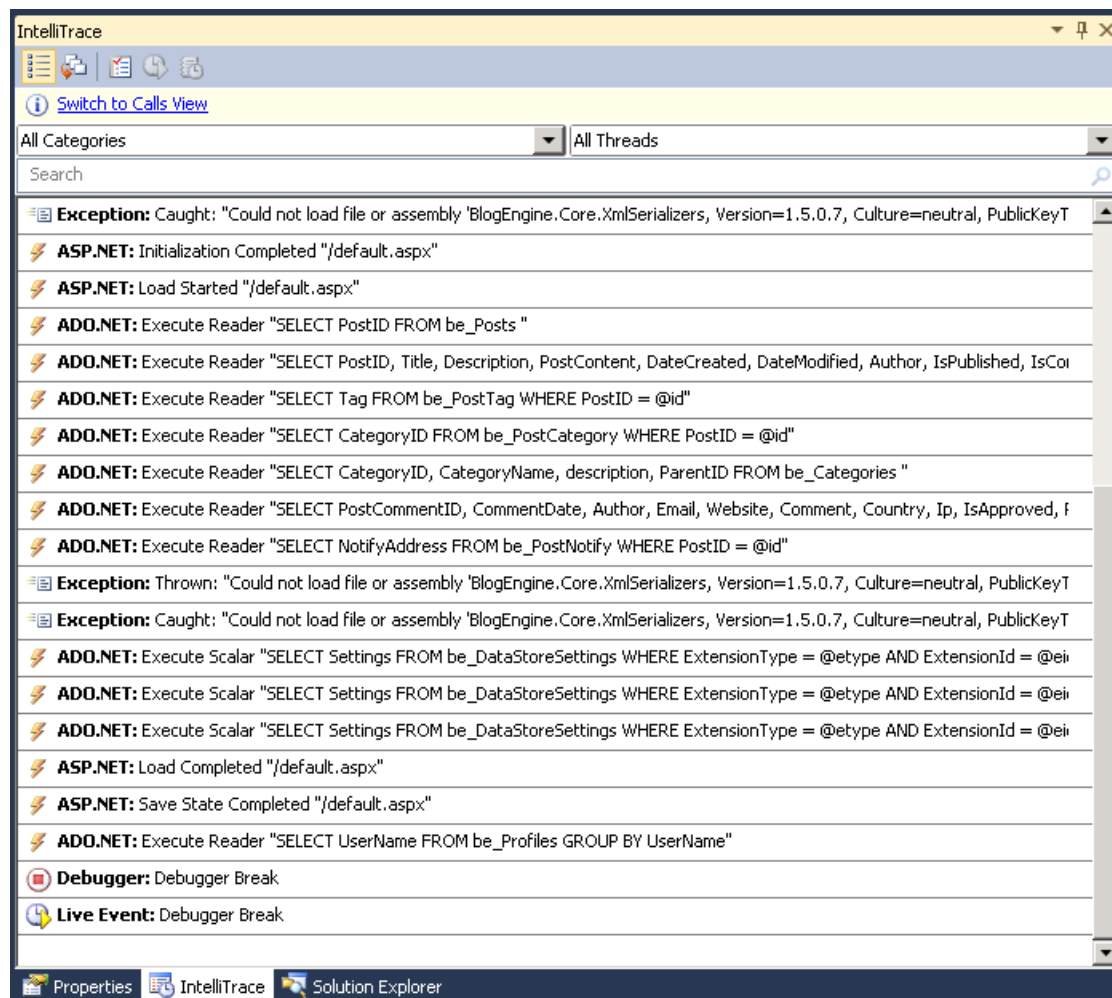


図 2 – IntelliTrace によるデバッグ セッション

テストまたはデバッグ セッション中に発生したあらゆる現象を確認できるということは、開発者がたとえば 50 個ものブレークポイントを設定して F5 (実行) キーを何度も押し、各ブレークポイントで提供されたデータを検証して、次のブレークポイントに移るという作業が不要になることを意味します。こうした手動でのプロセスは、コード内の停止すべき位置で開発者が誤って停止しなかった場合には最初からやり直しになるので、特に手間がかかります。

あらゆる情報を入手できる場合、その副次的効果として、他の方法では発見されないバグも発見できる可能性があります。図 2 では、通常のアプリケーション実行の途中で、いくつかの例外がスローされています。特にこの例外を探していたわけではないのですが、それでも発見されています。通常のデバッグ セッションでは、ブレークポイントを設定した位置よりも前で、何が起こったかを見ることはできません。このようなビューが得られるということは、以前は発見できなかったバグが発見できるようになることを意味します。

開発者は IntelliTrace™ を使用して、テスト中に実行されたメソッドまたはクラスを検索することもできますが、単純に IntelliTrace™ ログを見て問題の発生箇所を特定することもできます。図 3 の場合、開発者はユーザー コントロールで誤った情報が読み込まれた原因を調べようとしています。

図 3 からわかるように、開発者はログを見るだけで、ユーザー コントロールが読み込まれた箇所を特定し、Cookie 情報を読み取るべきではないにもかかわらず読み取っていたことを突き止めています。

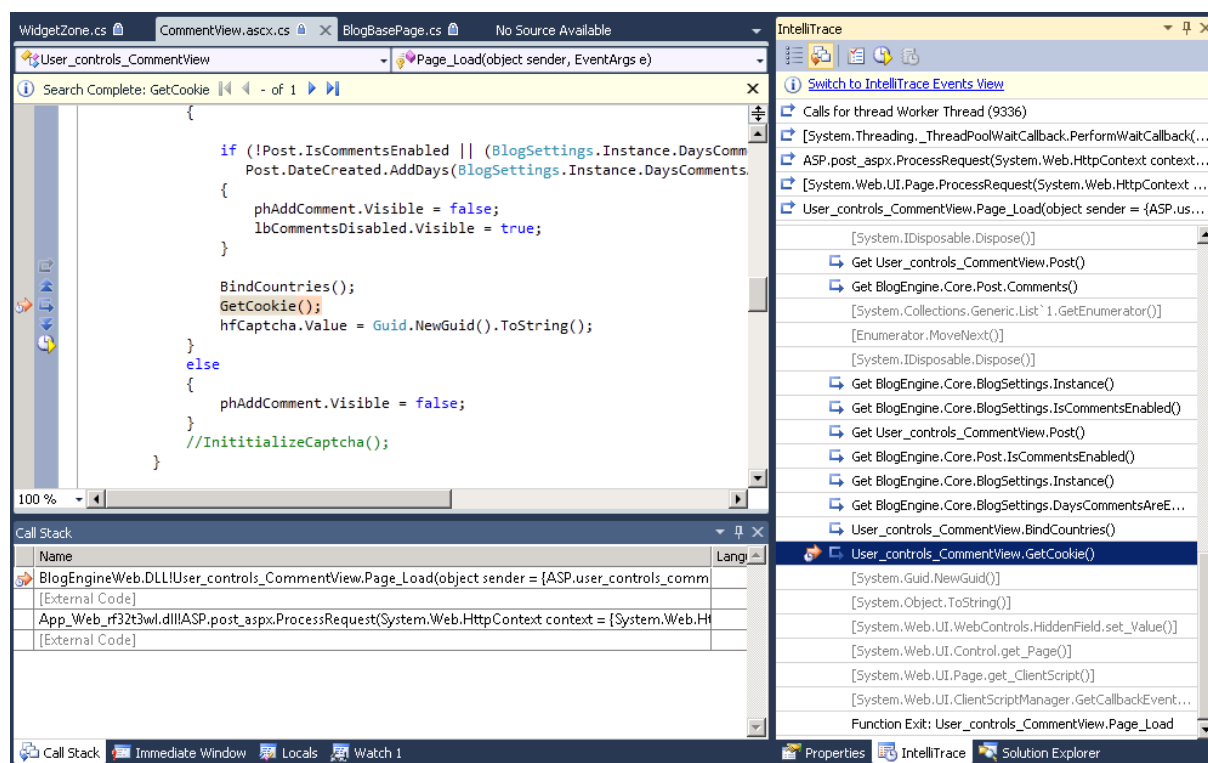


図 3 – IntelliTrace™ によるデバッグ

ここまでわかったら、開発者は問題のメソッドにステップ インし、どのような情報が読み込まれているか、その原因は何か、そしてデバッグ中に読み込まれた値は何かを確認します。

INTELLITRACE™ による時間の削減

コードのデバッグには膨大な時間が費やされます。単純な問題でも解決するには時間が必要です。Visual Studio 2010 Ultimate の新しいデバッグ ツールを使用すれば、開発者はどのようなデバッグ セッションでも優れた成果を上げることができます。さらに、必要とする時間は従来に比べて大幅に削減されます。捉えどころのない再現不可能なエラーや、散発的に発生するエラーを迅速に取り除くことで、テスト担当者と開発者の時間を節約できます。チーム リーダーは現場スタッフのツールキットに IntelliTrace があれば、デバッグから次の機能や次のリリースの作業にすばやく移れることがわかっているのです、安心できます。

お客様からの声



ソフトウェア開発会社: 統合型ツール セットでコストを削減し、開発作業を効率化

統計ソフトウェアの開発会社 Minitab では、開発作業の簡易化と迅速化を希望していました。同社は Microsoft Visual Studio 2010 Ultimate、Visual Studio Team Foundation Server 2010、および Visual Studio Team Lab Management 2010 を含む統合ソリューションを導入しました。Minitab は現在、IntelliTrace や Microsoft Visual F# などの機能を利用して、生産性の向上、開発およびライセンス コストの削減、新しい製品やサービスの商品化に要する時間の短縮を実現しています。

"IntelliTrace は当社にとって重要なデバッグ ツールになるでしょう。このツールを使ってテスト ワークフローを記録し、バグ報告にそのワークフローを添付できるからです。それだけでなく、Visual Studio 2010 の IntelliSense が提供する、エラーの報告、解析、バックグラウンド処理の新機能によって、開発作業は確実にスピードアップします。当社では Visual Studio Team Foundation Server 2010 を使用して、ソフトウェアのチェック インを管理したり、IntelliTrace ログ ファイルに記録されるテストワークフローを共有したりすることも可能になりました。"- Minitab データベース開発マネージャー、Matt Kowalski 氏



コンサルティング会社: 新しいツールによってデバッグが 25% 以上スピードアップする見込み

ソフトウェア コンサルティング会社 Wintellect は、デバッグに革新的なプロセスを導入することで、バグの解消を迅速化し、コード品質を向上させることを望んでいました。同社が選んだのは、新しい IntelliTrace デバッグ機能を含む Microsoft Visual Studio 2010 Ultimate 開発システムでした。同社ではこのツールによって、アプリケーションのデバッグが従来より 25% 速くなると同時に、コラボレーションの円滑化、コストの削減が実現されるものと見込んでいます。

"私の知る限り、Visual Studio 2010 Ultimate に含まれている IntelliTrace のようなデバッグ ツールは、他にはありませんでした。"- Wintellect 共同設立者、John Robbins 氏

"一般的なデバッグ ツールでこの問題を解決しようとしたら、おそらく 2 時間はかかります。しかし IntelliTrace を使ったおかげで、約 20 分で解決できました。IntelliTrace によるデバッグ モードでは、アプリケーションの動作がやや遅くなりますが、この程度のパフォーマンス低下は全体から見ればまったく問題ありません。IntelliTrace を使用することで、アプリケーションのデバッグを 25% 以上スピードアップできると予測しています。"- Wintellect 共同設立者、John Robbins 氏

"IntelliTrace やプロファイル機能の「アプリケーションの動作を視覚化」オプションを使用すると、ソース コードをプリントアウトするだけの場合より、はるかに多くの情報を得ることができます。アプリケーションの状態を表示し、変数を変更し、コードを実行し、何が起こったかをログ ファイルで見ることができます。2 回のプログラム実行を比較することもできます。これはプログラムのチューニングにも役立ちます。改善すべき部分が簡単にわかるからです。"- Wintellect 共同設立者、John Robbins 氏

"IntelliTrace を使用すれば、他では再現しないが問題が存在するという場合でも、他のスタッフを苦労して説得する必要はありません。Visual Studio Team Foundation Server 2010 に保存した IntelliTrace ファイルを他のスタッフに見せるだけでいいからです。データや証拠があれば、テスト担当者と開発者の関係は良い方向へ大きく変化するでしょう。"- Wintellect 共同設立者、John Robbins 氏

"一般的に、デバッグの時間は、難しい問題、つまり再現するのが難しい問題の解決に費やされています。私が IntelliTrace を歓迎している理由は、まさにそれです。この 20 年間で初めて、デバッグという技術に本当の変化をもたらすツールであり、これは私たちが普段デバッグに費やしている時間を短縮してくれます。"- Wintellect 共同設立者、John Robbins 氏

"実稼働アプリケーションにたった 1 つバグがあるだけでも、企業は 100,000 ドル以上の損害を被ったり、評判を落としたりすることになりかねません。今日のビジネスの状況で、企業がこういったリスクや出費を避けるには、IntelliTrace のようなツールを使用することです。私にとっては、Visual Studio 2010 の中で絶対に必要な機能は IntelliTrace です。他の機能よりも、まずは IntelliTrace です。"- Wintellect 共同設立者、John Robbins 氏