Microsoft Visual Studio 2005 既知の問題

このドキュメントでは、Microsoft Visual Studio 2005 の使用時に発生する可能性のある問題について説明します。 インストール、アンインストール、修復、および他のセットアップ手順に関する問題については、readme.htm ファイルを参照してください。

Beta 2 と製品版の互換性に影響する変更点の一覧については、http://go.microsoft.com/fwlink/?LinkId=51223 を参照してください。

1. Visual Studio

1.1 Visual Studio 2005 Beta 2 の VC++ プロジェクト ファイルを開くことができない

Visual Studio 2005 の最終バージョンを使用して、現在のコンピュータ上の Visual Studio のインストール場所とは異なる場所でコンピュータ上に最後に読み込まれた Beta 2 の VC++ プロジェクトを開いた場合、またはプロジェクトのパスが変更されている場合に、エラーが発生することがあります。

この問題を解決するには

プロジェクト設定ファイルを編集します。ソリューション エクスプローラで淡色表示されたプロジェクト ノードを右クリックし、[<プロジェクト名>.vcproj の編集] をクリックすると、XML エディタにファイルが表示されます。InheritedPropertySheets タグの値を "$(VCInstallDir)VCProjectDefaults\UpgradeFromVC70.vsprops" に変更し、プロジェクトを再読み込みします。

1.2 Windows 2000 で [アプリケーション検証ツールを使って開始] を使用すると、"アプリケーション検証ツールは、*machine_name* で実行中のオペレーティング ではサポートされていません。Windows XP 以降にアップグレートしてください。" というメッセージが表示される

Windows 2000 では、アプリケーション検証ツールをサポートしていません。

この問題を解決するには

Windows XP 以降にアップグレードします。

1.3 [アプリケーション検証ツールを使って開始] を使用すると、"アプリケーション検証ツールは、アップデートされたシステム ファイルを必要としていますが、使用中のコンピュータ *machine_name* では見つかりませんでした。Windows ダウンロード センターからアップデート ファイルを取得しますか。" というダイアログ ボックスが表示される

Visual Studio のセットアップでは、アプリケーション検証ツールに必要なシステム ファイルはインストールされません。 必要なバージョンのファイルがコンピュータのオペレーティング システム上に存在しない場合は、アプリケーション検証ツールを使用する前に、ファイルを更新しておく必要があります。

この問題を解決するには

ダイアログ ボックスの [はい] をクリックしてインターネット ブラウザを開き、Microsoft ダウンロード センターのサイトの更新用リンクを表示します。必要な更新をインストールします。Microsoft ダウンロード センターには、次のリンクからアクセスすることもできます。
http://go.microsoft.com/fwlink/?LinkId=49500

1.4 分散システム デザイナのツールボックスをリセットするまで、ツールボックスに Windows のローカル言語設定が反映されない

ユーザーは [オプション] ダイアログ ボックスで、Visual Studio で Windows のローカル言語を使用するように設定できます。 ただし、この設定は分散システム デザイナのツールボックスをリセットするまで反映されません。

この問題を解決するには

1) すべての分散システム デザイナ (アプリケーション デザイナ、システム デザイナ、論理データセンター デザイナ、配置デザイナ) を閉じ

ます。
2) ツールボックス内で右クリックし、[ツールボックスのリセット] をクリックします。
3) 使用する分散システム デザイナを再度開きます。

1.5 Windows 98 および Windows Me 上のリモート デバッグ

問題 **1 :**

Visual Studio 2005 を使用して、Windows Server 2003 ベースのドメイン コントローラで、ドメインに含まれる Windows 98 コンピュータまたは Windows ME コンピュータに対して、既定のトランスポートを使用してリモート デバッグを行うと、Windows 98/ME コンピュータを起動してから約 10 分後に、次のエラー メッセージが表示される場合があります。

Microsoft Visual Studio

'<コンピュータ名>' という名前の Microsoft Visual Studio リモート デバッグ モニタに接続できません。アクセスが拒否されました。

原因 : Windows Server 2003 ベースのドメイン コントローラで、サーバー メッセージ ブロック (SMB: Server Message Block) 署名が有効かつ必須になっているためです。

問題 **2 :**

Windows XP 以降のコンピュータ上の Visual Studio 2005 を使用して、Windows 98 コンピュータまたは Windows ME コンピュータのリモート デバッグを行うと、次のエラー メッセージが表示される場合があります。

Microsoft Visual Studio

'<qualifier_name>' という名前の Microsoft Visual Studio リモート デバッグ モニタに接続できません。リモート コンピュータ上の Microsoft Visual Studio リモート デバッグ モニタは、ローカル コンピュータに接続できません。DCOM 通信を初期化できません。詳細については、ヘルプを参照してください。

原因 : Visual Studio 2005 を実行しているコンピュータで DCOM での匿名ログオンのリモート アクセスが無効になっているためです。

この問題を解決するには

問題 **1 :**

1) SMB 署名を有効にしても必須にはしないようにドメイン コントローラ ポリシーを設定します。
この方法については、http://support.microsoft.com/?kbid=887429 を参照してください。
2) ドメイン コントローラを再起動します。
3) Visual Studio 2005 がインストールされているコンピュータを再起動します。
4) Windows 98/ME を実行しているコンピュータを再起動します。

問題 **2 :**

Visual Studio 2005 を実行しているコンピュータで DCOM での匿名ログオンのリモート アクセスを許可するには、以下の手順を実行します。
1) コマンド プロンプトで「dcomcnfg」と入力し、Enter キーを押します。コンポーネント サービスが開きます。
2) コンポーネント サービスで、[コンポーネント サービス]、[コンピュータ]、[マイ コンピュータ] の順に展開します。
3) ツール バーの [マイ コンピュータの構成] をクリックします。[マイ コンピュータ] ダイアログ ボックスが表示されます。
4) [マイ コンピュータ] ダイアログ ボックスの [COM セキュリティ] タブをクリックします。
5) [アクセス許可] の [制限の編集] をクリックします。[アクセス許可] ダイアログ ボックスが表示されます。
6) [グループ名またはユーザー名] の [ANONYMOUS LOGON] をクリックします。
7) [ANONYMOUS LOGON のアクセス許可] の [リモート アクセス] チェック ボックスをオンにし、[OK] をクリックします。
8) コンピュータを再起動します。

1.6 Web サービス プロバイダ エンドポイントで F1 キーを押すと、"情報が見つかりません" と表示される

アプリケーション ダイアグラムで Web サービス プロバイダ エンドポイントが選択されているときに F1 キーを押しても、関連するヘルプトピックは見つかりません。

この問題を解決するには

ヘルプで「方法 : ASP.NET Web サービスの操作を定義する」を検索してください。

1.7 ユーザーの機密情報を、カスタム設定の中、またはセキュリティ保護された設定として SDM ドキュメントで定義さ

れていない設定の中に格納することは推奨されない

カスタム設定の中、または SDM (System Definition Model) ドキュメント (.sdm ファイルや .ad ファイルなど) でセキュリティ保護された設定として定義されていない設定の中に格納された機密情報は、プレーンテキスト形式でこれらのファイル内に置かれます。

この問題を解決するには

設定および制約エディタを使用する際、SDM ドキュメントでセキュリティ保護された設定として指定されている設定には機密情報を格納できますが、それ以外の設定には機密情報を格納しないでください。詳細については、ヘルプの「方法：アプリケーション、サーバー、エンドポイント、およびゾーンのカスタム設定を作成する」、および「アプリケーションの実装に関する考慮事項」を参照してください。

1.8 Itanium (IA64) ネイティブ ツールのコマンド プロンプトの起動

Itanium (IA64) にツールをインストールしたときに、Itanium (IA64) ネイティブ ツールのコマンド プロンプトを起動するためのショートカットは作成されません。

この問題を解決するには

Itanium (IA64) ネイティブ ツールのコマンド プロンプトを起動するには、以下の手順を実行します。
1) [スタート] メニューの [ファイル名を指定して実行] をクリックします。
2) **cmd** と入力します。
3) コマンド プロンプトで、「<Visual Studio 2005 のインストール ディレクトリ>\VC\bin\vcvars64.bat」と入力します。

1.9 ネットワーク共有のインストルメント化されたモジュールのプロファイリングと実行はサポートされていない

Visual Studio の既定の設定で、ネットワーク共有のインストルメント化されたバイナリを実行すると、未処理の例外が発生します。

この問題を解決するには

プロファイリングするプロジェクトまたはバイナリをローカル ハード ドライブにコピーまたは移動します。

1.10 Event Tracing for Windows (ETW) を使用した IIS 5.1 のイベント プロファイリング データの収集はサポートされていない

IIS 5.1 を対象とするアプリケーションのトレース プロファイリング、および ETW イベントの収集の有効化によって、UI で障害が発生し、コマンド ラインに誤った Error VSP1432 が報告されます。

1.11 IA64 プロファイリングはサポートされていない

IA64 プロファイリングはサポートされなくなりました。

1.12 スタンドアロンのプロファイラはコード カバレッジをサポートしていない

スタンドアロンのプロファイラのインストールでコード カバレッジを実行しようとすると、未処理の例外が発生します。

1.13 ASP.NET アプリケーションに対してコード カバレッジを実行すると、web.config が変更された状態のままになる場合がある

ASP.NET プロジェクトに対する単体テストの実行時に、テスト エンジンは web.config ファイルを変更する必要があります。場合によっては、これらのファイルがクリーンアップされず、以降のテスト実行時およびアプリケーション実行時に例外が発生する可能性があります。

この問題を解決するには

テスト実行時に作成されたバックアップから、元の web.config ファイルを復元します。

1.14 Windows 98 では、ReportViewer コントロールのサポートが制限されている

Windows 98 でアプリケーションを実行する場合は、Windows フォームの ReportViewer コントロールの機能が制限されます。このコントロールを使用して表示できるのは、SQL Server 2005 Reporting Services のレポート サーバー上で以前に発行されたレポートだけです。このコントロールを使用して、ローカル処理モードでクライアントのレポート定義 (.rdlc) ファイルを表示することはできません。

1.15 IIS ワーカー プロセスが SYSTEM として実行されている場合、ASP.NET のコード カバレッジは機能しない

IIS ワーカー プロセスが SYSTEM として実行されている場合、ASP.NET のコード カバレッジ データの収集は機能しません。 IIS ワーカー プロセスの名前は IIS のバージョンごとに異なるため、システム上のワーカー プロセスの名前を MSDN で確認してください。タスク マネージャを使用すると、ワーカー プロセスを実行しているユーザーを確認できます。
メモ：コード カバレッジが無効になっていれば、単体テストは正しく実行されます。

この問題を解決するには

SYSTEM として実行しないように IIS ワーカー プロセスを変更します。

1.16 フォーム認証を有効にした状態で ASP.NET アプリケーションをプロファイリングしようとするとエラーが発生する

フォーム認証を有効にした状態で ASP.NET アプリケーションをプロファイリングしようとすると、Web サイト構成エラーが発生し、アプリケーションを実行できなくなります。

この問題を解決するには

既知の解決策はありません。

1.17 コードの実行中に分散システム デザイナのダイアグラムでアクションを実行すると、Visual Studio が応答しなくなる

たとえば、デバッガの実行中にシステム デザイナでシステム ダイアグラムにアプリケーションを追加すると、Visual Studio が応答しなくなります。

この問題を解決するには

分散システム デザイナのダイアグラムに対してアクションを実行する前に、コードの実行を停止してください。

1.18 共有 dll に対してトレース プロファイリングを実行すると、その dll を使用するすべてのプロセスがプロファイリングされる

共有 dll があり、トレース プロファイリングする場合、その dll を使用しているすべてのプロセスをプロファイリングすることになります。

この問題を解決するには

1) vsperfcmd -proccessoff:PID を使用して、プロファイリングする必要のない各プロセスのプロファイリングを無効にします。
2) HKCU/software/microsoft/visual studio/8.0/VSPERF/Monitor/Settings/ProcessProfile=Off を設定してから、ターゲット プロセスを開始します。その後で vsperfcmd -processon:PID を使用し、プロファイリングの対象とするすべてのプロセスについてプロファイリングを有効にします。
メモ：プロファイリングを終了した後は、モニタをシャットダウンしてデータをレポート ファイルにフラッシュするために、該当する共有 dll を使用しているすべてのプロセスを終了する必要があります。

1.19 ネイティブに生成された (NGEN の) イメージをトレース プロファイリングする方法

NGEN のイメージをインストルメントしようとすると、誤った VSP1014 エラーが報告されます。
CLR は、既に NGEN であるインストルメント化されたマネージ バイナリを読み込みません。

この問題を解決するには

NGEN のイメージではなく、マネージ バイナリ自体をインストルメントしたら、次のいずれかを実行する必要があります。

1) CLR がインストルメント化されたバイナリを使用するように、NGEN のイメージを削除します。
または
2) ngen.exe を使用してバイナリを再生成します。

1.20 API をプロファイリングする MarkProfile の使用時に、プロファイラ モニタを実行していないとクラッシュする

vsperf.h で提供される MarkProfile、CommentMarkProfile、または CommentMarkAtProfile を使用する場合、プロファイラ モニタを実行していないとアプリケーションがクラッシュします。これは、API に対する呼び出しを追加し、プロファイラではなくデバッガでそれらの呼び出しを起動した場合に、最も顕著に発生します。

この問題を解決するには

"vsperfcmd -start:sample -output:foo.vsp" を使用してプロファイラ モニタを起動します。

1.21 Web プロジェクト : GAC にインストールされたアセンブリに定義された型は、クラス ダイアグラムに表示できない

次のすべての条件に当てはまる場合、クラス ダイアグラムに参照アセンブリ内の型を表示することはできません。
1. クラス ダイアグラムが Web プロジェクトに含まれている。
2. 参照アセンブリがグローバル アセンブリ キャッシュ (GAC: Global Assembly Cache) にインストールされている。
3. 参照アセンブリが Framework のルート ディレクトリ ("Microsoft.Net\Framework\<バージョン>\") にインストールされていない。

この問題を解決するには

Web プロジェクト以外のプロジェクトから参照型を表示します。

1.22 インストルメント化されたマネージ dll を Web サイト プロジェクトに読み込めない

Web サイト内でホストされたインストルメント化されたマネージ ユーザー コントロールは、.NET の既定のセキュリティ設定では読み込むことができません。

この問題を解決するには

ローカル Web サイトの [完全な信頼] を有効にして、インストルメント化された dll を読み込めるようにする必要があります。
caspol.exe -ag 1.2 -url http://localhost FullTrust

1.23 アプリケーション検証ツールが混合モードの C++ プロジェクトで有効にならない。代わりに、次のダイアログ ボックスが表示される。
"選択したデバッグ モードではアプリケーション検証ツールはサポートされていません。詳細についてはサポート ドキュメントを参照してください。検証ツールを使用せずにデバッグを継続する場合は [OK] をクリックしてください。"

アプリケーション検証ツールは、[デバッグ] メニューで選択できる場合でも、混合モードのアプリケーションでは有効になりません。

この問題を解決するには

アプリケーション検証ツールを使用せずにデバッグを続行します。

1.24 Windows 2000 でプロファイラ ドライバが正しく初期化されない

プロファイラ ドライバでは、初めて使用されるときにドライバ自体の初期化処理を実行する必要があります。この初期化処理がリモート デスクトップ セッションを通じて Windows Server 2000 上で行われた場合、ドライバの初期化はそれ以降のセッションには反映されますが、現在のセッションには反映されません。現在のセッションでモニタを起動しようとすると、Error VSP1398 が発生し、モニタを起動できません。

この問題を解決するには

次のいずれかの方法を実行します。

1) コンピュータを再起動します。
または

2) リモート デスクトップを使用して、同じコンピュータの別のセッションに接続します。

1.25 テスト実行ウィンドウ : "Admin users" グループに属さない "Controller users" グループのユーザーがコントローラに接続できない

ユーザーを "Controller users" グループに追加しても "Controller admin" グループには追加しなかった場合、その変更はコントローラ サービスを再起動するまで有効になりません。このため、そのユーザーはコントローラに接続できません。

この問題を解決するには

コントローラ サービスを再起動します。

1.26 エディタでファイルを再読み込みしたときに、エンコーディングの変更が表示されない場合がある

Visual Studio 2005 では、ファイルの再読み込み時にエンコーディングの変更は検出されません。現在のエディタ外部のファイルのエンコーディングを変更した場合、またはエディタで開いているファイルを変更するソース管理操作を実行した場合に、Visual Studio はファイルを自動的に再読み込みします。エディタでファイルの内容が再読み込みされた後、正しく表示されない場合があります。

この問題を解決するには

□□□ 変更を保存せずにファイルを閉じます。

□□□ [ファイル] メニューの [開く] をクリックし、[ファイル] をクリックします。

□□□ [ファイルを開く] ダイアログ ボックスで、[開く] の横の矢印をクリックし、[ファイルを開くアプリケーションの選択] をクリックします。

□□□ [ファイルを開くアプリケーションの選択] ダイアログ ボックスの一覧で、ファイルを開くエディタ (バイナリ エディタ、リソース エディタなど) を選択します。特定のエンコーディングでファイルを開くには、XML Editor with Encoding など、エンコーディングをサポートするエディタを選択します。

□□□ [OK] をクリックします。

□□□ [エンコード] ダイアログ ボックスで、[エンコード] ドロップダウン リストから適切なエンコーディングを選択します。

□□□ [OK] をクリックします。

1.27 セットアップ プロジェクトのインストール時に製品の修復が発生する

Visual Studio 2005 の後に Visual Studio .NET 2003 をインストールした場合、いずれかのバージョンの Visual Studio からセットアップ プロジェクトをビルドすると製品の修復が発生します。製品の修復は、Visual Studio .NET 2003 をインストールしたユーザー以外のユーザーに対してのみ発生します。

この問題を解決するには

製品の修復が発生したとき、そのまま続行して完了させます。ただし、修復が正常に終了するには、Visual Studio .NET 2003 のインストールを実行したユーザー以外の、管理者特権を持つユーザー アカウントで実行する必要があります。

1.28 Visual Studio 2005 Tools for Microsoft Office プロジェクトのインストール後に Office 製品を実行できない

Reg-Free COM コンポーネントを含む Visual Studio 2005 Tools for Microsoft Office アドイン プロジェクトをインストールすると、アドインの対象となる Office 製品を実行できなくなります。

この問題を解決するには

Reg-Free COM は、Visual Studio 2005 Tools for Microsoft Office アドイン プロジェクトではサポートされていません。

1.29 Visual Studio .NET 2003 からインポートしたセットアップ プロジェクトと配置プロジェクトにビルド時に署名できない

署名が有効になっている Visual Studio .NET 2003 からインポートしたセットアップ プロジェクトと配置プロジェクトは、Visual Studio 2005 でのビルド時に署名されません。ビルド時に、"ファイル ' <filename> ' は署名されていません。配置プロジェクトには使用できないプロパティの署名が含まれています。詳細については、ヘルプを参照してください。" というメッセージが [エラー一覧] ウィンドウに表示されます。

この問題を解決するには

署名を有効にするには、セットアップ/配置プロジェクトのビルド後の手順のビルド出力で、ソフトウェア開発キット ツールの SignTool を起動します。これを実行する方法の詳細については、上記のビルド メッセージに関連するヘルプ トピックを参照してください。ヘルプにアクセスするには、[エラー一覧] ウィンドウでメッセージを強調表示し、F1 キーを押します。

プロジェクトの不要になった署名プロパティを削除するには、以下の手順を実行します。
1. [エラー一覧] ウィンドウでメッセージをダブルクリックします。
2. 表示されるダイアログ ボックスの [はい] をクリックします。これで、プロジェクトの不要になった署名プロパティがプロジェクト ファイルから削除されます。

1.30 Visual Studio 2005 で使用できる一部のブートストラップ パッケージは、32 ビット プラットフォームだけを

対象としている

Visual Studio 2005 で使用できる .NET Framework 2.0、SQL Server 2005 Express、Microsoft Visual J# 再頒布可能パッケージ、Windows Installer 2.0、および Windows Installer 3.1 用のブートストラップ パッケージは、32 ビット プラットフォームだけを対象としています。これらのパッケージに組み込まれたブートストラップを 64 ビット プラットフォームで実行した場合、インストールがブロックされることがあります。

この問題を解決するには

64 ビット バージョンの .NET Framework 2.0 および SQL Server 2005 Express 用の再頒布可能なブートストラップ パッケージは、Microsoft ダウンロード センターで入手できます。

1.31 [設定としてユーザー定義された型] を使用すると、設定デザイナで予期しない動作の原因となる場合がある

[設定としてユーザー定義された型] を使用すると、UDT が含まれるアセンブリを使用するプロジェクトを開いているときにそのアセンブリを更新した場合、問題が発生することがあります。このシナリオが必要な場合は、UDT アセンブリでインクリメンタル バージョン セマンティクスを使用することで、この問題を回避できます。

この問題を解決するには

設定として使用するユーザー定義型がクラス ライブラリに存在する場合、ライブラリをインクリメンタル方式でバージョン化して問題を軽減します。ユーザー定義型が EXE 内に存在する場合は、IDE を閉じ、再起動して UDT への変更を有効にする必要があります。

1.32 ユーザー コントロール プロジェクト内の "My" にフォーム プロパティがない

My.Forms は、ユーザー コントロール プロジェクトでは使用できません。

この問題を解決するには

既知の解決策はありません。

1.33 リモート デバッグ時に Sleep 呼び出しで中断した場合、オブジェクトのメソッドを呼び出せない

リモート コンピュータで次のコードを実行しているとき、実行中のコードにリモート デバッグで接続し、Sleep() 呼び出しの内部で実行を中断したとすると、変数 c のメンバを評価することはできません。

```
c = New c1      'c1 は有効なクラスであることが前提
While True
Threading.Thread.Sleep(1000)
End While
```

この問題を解決するには

Sleep() からステップ アウトすると、通常どおりオブジェクトを評価できるようになります。

1.34 Visual Studio 2005 Team Edition for Software Testers : ロード テストのエラー テーブルに、"しきい値の規則を適用するために必要とされる依存カウンタが見つかりませんでした" というメッセージが表示される

ロード テストの結果ビューのエラー テーブルに、"しきい値の規則を適用するために必要とされる依存カウンタが見つかりませんでした" というメッセージと共にエラーが表示されます。ロード テストには、あるパフォーマンス カウンタを別のパフォーマンス カウンタと比較するしきい値の規則が含まれています。このエラーは、サンプリング間隔中に比較パフォーマンス カウンタによってインスタンスが生成されない場合に発生します。

この問題を解決するには

関連付けられたしきい値の規則を変更して、別のパフォーマンス カウンタではなく定数と比較するようにします。これは、指定したしきい値の規則を実行できない点を除き、ロード テストの結果に影響を及ぼすことはなく、安全に無視できます。

カウンタを変更するには、以下の手順を実行します。
1. ロード テストを編集し、[カウンタ セット]、[ロード テスト]、[カウンタ カテゴリ]、[LoadTest:Request]、[カウンタ]、[Avg. Connection Wait Time]、[しきい値規則] の順にクリックします。
2. カウンタの比較規則を削除します。
3. 右クリックしてカウンタの比較規則を追加します。
4. この規則で、[しきい値を超えたときに警告] を true に設定し、警告を .01 (10 ミリ秒)、重大なしきい値を .02 (20 ミリ秒) に設定します。

1.35 Visual Studio 2005 Team Edition for Software Testers : Web テストを実行すると、"オブジェクトのインスタンスに設定されていないオブジェクトです。" というエラーによってテストが失敗する

未使用のデータ ソースが含まれたコード化された Web テストを実行すると、"オブジェクトのインスタンスに設定されていないオブジェクトです。" というエラーが発生し、テストが失敗します。このエラーが発生するのは、コード化された Web テスト内でデータ ソースが定義されているにもかかわらず、テストでデータ ソースがどのアイテムにもバインドされないためです。

この問題を解決するには

コード化された Web テストから未使用のデータ ソースを削除するか、DataBinding 属性をテスト クラスに追加することにより、テストでデータ ソースのフィールドをアイテムにバインドします。

1.36 Visual Studio 2005 Team Edition for Software Testers : SoapHttpClientProtocol 実装の Proxy プロパティを明示的に設定することが必要

Web サービスを呼び出す単体テストを含んだロード テストを実行する場合、単体テストのコードでは、System.Web.Services.Protocols.SoapHttpClientProtocol を実装する Web サービス プロキシ クラスの Proxy プロパティを明示的に設定する必要があります。プロキシを自動的に検出する必要がある場合はパフォーマンスのボトルネックが生じる可能性がありますが、Proxy を明示的に設定すると、この問題を防ぐことができます。

たとえば、次のような Web サービス プロキシ クラスがあるとします。

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Web.Services.WebServiceBindingAttribute(Name="InstantOrderSoap", Namespace="http://tempuri.org/")]
public partial class InstantOrder : System.Web.Services.Protocols.SoapHttpClientProtocol {

}
```

プロキシ オブジェクトを作成した後、そのオブジェクトを使用する前に、Proxy プロパティを次のように明示的に設定します。

```
InstantOrder orderCheck = new InstantOrder();
orderCheck.Proxy = new WebProxy("myproxy", true);
```

1.37 ActiveX EXE オブジェクトのバリアント プロパティに構造体を渡せない

Windows 98 では、ActiveX EXE オブジェクトのバリアント プロパティに構造体を渡すことはできません。クリーンな Windows 9X コンピュータでは、System DLL rpcrt4.dll が既定で登録されていないため、オペレーティング システムに登録キー [HKEY_CLASSES_ROOT\CLSID\{B5866878-BD99-11D0-B04B-00C04FD91550}] がないことが問題です。登録が見つからないため、呼び出しが失敗します。

この問題を解決するには

C:\Windows\System に移動し、RegSvr32.exe rpcrt4.dll を手動で呼び出します。

1.38 循環制約が指定された型パラメータで New を使用すると、VB コンパイラがハングし、メモリの割り当てが増加する

コンソール アプリケーションに以下のコードを貼り付けると、アプリケーションがハングします。

```
Class C1(Of U As U)     'この循環制約によって後で問題が生じる
End Class

Partial Class C1(Of U)
Dim x As New U          '問題 ? これが原因で無限ループが発生し、メモリを使用できなくなるまでメモリ使用量が増え続ける

End Class
```

この問題を解決するには

循環制約を削除します。

1.39 ClickOnce を使用してアプリケーションを起動した場合、StartupNextInstanceEvent ハンドラに対するパラメータにコマンド ライン引数が含まれない

ClickOnce を使用してアプリケーションを起動した場合、StartupNextInstanceEvent に対するパラメータに、アプリケーションの 2 つ目のインスタンスに対するコマンド ライン引数が含まれません。My.Application.Deployment.ActivationUri プロパティを使用して、コマンド ライン引数を取得する必要があります。

この問題を解決するには

次のように、ASP ランタイム クラスを使用してコマンド ライン引数を取得します。

```
Imports System.Web

Dim commandLineArgs as NameValueCollection = _
HttpUtility.ParseQuery(My.Application.Deployment.ActivationUri)
```

1.40 VB コンパイラが InternalsVisibleTo 属性をサポートしていない

```
<Assembly: InternalsVisibleTo("Foo.Dll, PublicKeyToken=a29c01bbd4e39ac5")>
```

この属性は Friend 型を他のアセンブリに公開します。この属性は、Visual Basic コンパイラではサポートされていません。

プライベート型をテストするためにこの属性に依存する一部の単体テスト テクノロジは、期待どおりに動作しません。

この問題を解決するには

既知の解決策はありません。

1.41 vsperfcmd.exe に関する一部のドキュメントがない

一般的なユーザー シナリオについて説明する vsperfcmd のドキュメントはありません。

vsperfcmd.exe と共に使用する一般的なコマンドの例を次に示します。

```
vsperfcmd -start:sample -output:foo.vsp
vsperfcmd -start:trace -output:foo.vsp -counter:g,1,0,InstructionsRetired
vsperfcmd -launch:foo.exe -args:"input arguments" -gc:lifetime
vsperfcmd -attach:foo.exe -pf:1
```

1.42 VB Web コントロール ライブラリ プロジェクトで My.Log を使用できない

VB Web コントロール ライブラリ プロジェクトでは、My.Log は使用できません。 My はプロジェクト単位であるため、My.Log は Web サイト プロジェクト内の WebUserControls では使用できますが、Web サイトで使用される Web コントロール ライブラリ プロジェクト内のクラスでは使用できません。

この問題を解決するには

My.Log のメソッドの代わりに Trace.Write を使用します。

1.43 ユーザーにファイル I/O アクセス許可がない場合、My.Application.Log.WriteEntry が例外をスローすることがある

1) 次のコードによって新しいコンソール アプリケーションを作成します。

```
My.Application.Log.DefaultFileLogWriter.CustomLocation = "D:\temp"
My.Application.Log.WriteEntry("Foo")
```

2) アプリケーションを実行します。

結果：
ログ ファイルは D:\temp に作成されますが、ディレクトリ D:\Documents and Settings\<ユーザー名>\Application Data\Microsoft\WindowsApplication1\1.0.0.0 が存在しない場合、このディレクトリも作成されます。ユーザーにこれを実行するためのアクセス許可がない場合、例外がスローされます。既定では、ASP.NET プロセスには Application Data ディレクトリへの書き込みアクセ

ス許可がないため、Web アプリケーションでクラス ライブラリを使用したときに、この例外が発生する可能性が高まります。

この問題を解決するには

1) app.config を使用して既定の FileLogTraceListener を削除し、別の TraceListener を使用するよう My.Application.Log を再構成します。
2) 未処理の例外イベントを待機し、イベントが発生したら続行します。
3) すべてのログ呼び出し (または、最低でも初めて実行されたと考えられるログ呼び出し) を試行/キャッチします。

1.44 64 ビット プラットフォーム上で実行している Visual Basic アプリケーションおよび C# アプリケーションで、32 ビット COM コンポーネントへの参照が機能しない場合がある

既存の COM コンポーネントのほとんどは 32 ビット プラットフォーム上でのみ使用でき、64 ビット プラットフォーム上の 64 ビット プロセスでは動作しません (ただし、64 ビット プラットフォーム上の 32 ビット プロセスでは正しく動作します)。このような 32 ビット COM コンポーネントを参照する Visual Basic アプリケーションおよび C# アプリケーションは、既定では 64 ビット プラットフォーム上では動作しません。これは、既定ではアプリケーションが 64 ビット プロセスとして起動するためです。

1 つ以上の COM 参照を含むプロジェクトが次のような場合にこの問題が発生します。
1. Visual Studio 2005 に移行され、64 ビット プラットフォーム上で実行された場合

または
2. 64 ビット プラットフォーム上で Visual Studio 2005 を使用して作成された場合

Visual Studio 2005 では、VB コンパイラと C# コンパイラは、プラットフォーム ターゲット プロパティを使用して、.exe または .dll を 32 ビットと 64 ビットのどちらの CPU アーキテクチャ モードで実行する必要があるかを判断します。Visual Studio 2005 では、このプロパティが既定で "AnyCPU" に設定されています。これは、ホスト プラットフォームに応じて、アプリケーションを 32 ビット モードと 64 ビット モードのどちらのモードででも実行できることを示しています。この場合、これらのアプリケーションをデバッグまたは実行しようとすると、"クラスをインスタンス化できません..." などのメッセージが表示される場合があります。

この問題を解決するには

COM コンポーネントを参照する Visual Basic プロジェクトまたは C# プロジェクトのプラットフォーム ターゲット プロパティに "X86" を設定します。

C# プロジェクトの場合：
1. ソリューション エクスプローラでプロジェクトを右クリックし、[プロパティ] を開きます。
2. [ビルド] タブをクリックします。
3. [プラットフォーム ターゲット] プロパティに「X86」を設定します。

VB プロジェクトの場合：
1. ソリューション エクスプローラでプロジェクトを右クリックし、[プロパティ] を開きます。
2. [コンパイル] タブをクリックします。
3. [詳細コンパイル オプション] をクリックします。
4. [ターゲット CPU] プロパティに「X86」を設定します。

Express Editions：
Visual Basic 2005 Express Edition および Visual C# 2005 Express Edition では、開発環境内でターゲット プロパティが公開されていません。プロジェクト ファイルを変更するには、テキスト エディタまたは XML エディタを使用して注意しながら変更する必要があります。
1. プロジェクトまたはソリューションを閉じます。
2. [ファイル] メニューの [ファイルを開く] をクリックします。
3. プロジェクト ディレクトリに移動し、プロジェクト ファイルを強調表示します。
4. [開く] をクリックし、XML エディタでプロジェクト ファイルを開きます。
5. 最初の <PropertyGroup> セクションを探し、次の行を追加します。
<PlatformTarget>x86</PlatformTarget>
1. プロジェクト ファイルを保存します。
2. [ファイル] メニューの [プロジェクトを開く] または [ソリューションを開く] を使用して、プロジェクトまたはソリューションを再度開きます。
3. 開発、デバッグ、テストを続行します。

また、アプリケーションが 64 ビット プラットフォームを対象としている場合は、アプリケーションに追加する COM コントロールの 64 ビット対応版を開発用コンピュータと配置用コンピュータに置くことができます。

1.45 Windows 移動プロファイルを使用すると、起動時に毎回、初めての起動であることを示すメッセージが表示される場合がある

Visual Studio 製品ファミリのいずれかを Windows 移動プロファイルと共に使用すると、セッションを起動するたびに、"Visual Studio 2005 は最初の使用のために環境を構成中です。数分間お待ちください。" という初めての起動であることを示すメッセージが表示される場合があります。これにより、起動時のパフォーマンスが不必要に低下する可能性があります。

この問題を解決するには

[ツール] の [オプション] をクリックします。[設定のインポートとエクスポート] をクリックし、[このファイルに自動的に設定を保存する:] に指定したパスを "My Documents" ディレクトリの下ではないパスに変更します。

1.46 Visual Studio 2005 の起動前に DTE.CommandBars にアクセスするアドインをインストールすると、アドインの読み込みエラーになる

Visual Studio 2005 をインストールした後に、OnConnection で DTE.CommandBars の取得を試みるアドインをインストールした場合、Visual Studio 2005 を初めて起動したときに次のメッセージが表示されます。

> アドイン <アドイン名> の読み込みに失敗したか、または例外が発生しました。
> このアドインを削除しますか?

この問題を解決するには

Visual Studio 2005 をインストールする場合、インストール後に Visual Studio を起動し、アドインをインストールする "前" に Visual Studio を再度シャットダウンします。
このアドインが既にインストールされているときに、上記のメッセージが表示された場合は、[いいえ] をクリックし、Visual Studio を閉じます。コマンド プロンプトで、<VSInstallDir>\Common7\IDE に移動し、次のコマンドを実行します。

```
devenv /resetaddin *
```

1.47 クラス ライブラリのビルドを最後に実行したときに、Visual Basic または Visual J# のクラス ライブラリ内の Web サービス参照がリバース エンジニアリングされない場合がある

このシナリオでは、Visual Studio ソリューションに Web サービスへの Web 参照を含む Visual Basic または Visual J# のクラス ライブラリが含まれます。また、ソリューションには、このクラス ライブラリを参照する ASP.NET Web サイトも含まれます。これらのアイテムをソリューションに追加したら、構成ファイルの一部のエントリを、クラス ライブラリの App.config ファイルから Web サイトの Web.config ファイルにコピーする必要があります。構成ファイルのエントリをコピーした後、アプリケーション ダイアグラムを追加してこれらのアイテムをリバース エンジニアリングする前にクラス ライブラリをビルドした場合、ASP.NET Web サイトだけがダイアグラムでリバース エンジニアリングされる可能性があります。したがって、Web サービス参照は、ASP.NET アプリケーションの Web サービス コンシューマ エンドポイントとしてリバース エンジニアリングされず、参照先 Web サービスへの接続がリバース エンジニアリングされないことがあります。ダイアグラム上にまだ存在しない Web サービスの Web 参照の場合、参照先 Web サービスの外部 Web サービスがリバース エンジニアリングされない場合があります。

この問題を解決するには

1. アプリケーション ダイアグラムを追加した後で、クラス ライブラリ内の Web 参照を右クリックし、[Web 参照の更新] をクリックします。
2. クラス ライブラリを再度ビルドします。

1.48 テーブルの場所を変更すると接続できない場合がある

実行時にテーブルの場所を別のデータベースに変更する場合は、新しいデータベース内のテーブルと元のレポート内のテーブルを同じ名前にする必要があります。テーブル名が一致しない場合、接続は失敗します。

この問題を解決するには

新しいデータベースに元のレポート内のテーブルと同じ名前のテーブルが含まれていることを確認します。

1.49 null 文字列を使用してビューアの選択数式を設定することはできない

レポート内のすべてのレコードを表示するようにビューアの選択数式を設定する場合は、null 文字列ではなく空の文字列を使用する必要があります。null 文字列は、元のレポート選択数式をオーバーライドできません。

この問題を解決するには

空の文字列は "" または String.Empty で表します。null 文字列は C# では null、Visual Basic では Nothing で表します。

1.50 Shift-JIS でエンコードされたプロジェクトを移行すると文字が破損する場合がある

Visual Studio .NET 2002 または Visual Studio .NET 2003 から Visual Studio 2005 に移行した ASP.NET Web アプリケーションでは、Shift-JIS としてエンコードされた日本語は正しく表示されません。

この問題を解決するには

移行前に、すべての ASP ページを UTF-8 に変換します。

1.51 Crystal Reports 9 または Crystal Reports 10 からプロジェクトをインポートしたときに名前空間参照が失われる

Crystal Reports 9 または Crystal Reports 10 の Windows プロジェクトを Crystal Reports for Visual Studio 2005 に移行すると、CrystalDecisions.ReportSource、CrystalDecisions.Shared、CrystalDecisions.Windows.Forms の各アセンブリへの参照は失われます。

この問題を解決するには

移行後、アプリケーションをコンパイルする前に、参照を手動で追加します。

1.52 Windows プロジェクトと ASP.NET Web プロジェクトが同じクラス ライブラリを参照している場合、そのクラス ライブラリ内の Web サービス参照がリバース エンジニアリングされないことがある

このシナリオでは、Visual Studio ソリューションに、実装する Windows アプリケーションと ASP.NET アプリケーションを含むアプリケーション ダイアグラムが含まれます。また、ソリューションには、Web サービスへの Web 参照を含むクラス ライブラリも含まれます。Windows プロジェクトと ASP.NET Web プロジェクトは、共にこのクラス ライブラリを参照しています。クラス ライブラリをビルドし、必要なエントリをクラス ライブラリの App.config ファイルから両方のプロジェクトの構成ファイルにコピーした後、Windows アプリケーションまたは ASP.NET アプリケーションの Web サービス コンシューマ エンドポイントとして、Web 参照がリバース エンジニアリングされない場合があります。また、参照先 Web サービスへの接続もリバース エンジニアリングされません。ダイアグラム上にまだ存在しない Web サービスの Web 参照の場合、参照先 Web サービスの外部 Web サービスがリバース エンジニアリングされない場合があります。

この問題を解決するには

1. アプリケーション ダイアグラムを閉じます。
2. Windows プロジェクトと ASP.NET プロジェクトから共有クラス ライブラリへの参照を削除します。
3. クラス ライブラリ参照を ASP.NET プロジェクトに追加します。
4. アプリケーション ダイアグラムを開きます。
5. クラス ライブラリ参照を Windows プロジェクトに追加します。

1.53 FTP Web サイトのサーバー エクスプローラでデータ ファイルを展開すると、IDE がクラッシュする可能性がある

FTP Web サイトを開き、データ ファイル (MDB または MDF) を App_Data ディレクトリに追加した後に、[サーバー エクスプローラ] ウィンドウを開き、Data Connections ノードの
データ ファイルへの接続を展開した場合、IDE がクラッシュまたはハングする可能性があり、保存していないデータが失われることがあります。

この問題を解決するには

ファイル システムまたは IIS Web サイトを使用して、ローカル ファイルの場所から Web サイトを開発し、[Web サイトのコピー] コマンドを使用して、FTP パスとの間でファイルを転送します。

1.54 Firefox から出力またはエクスポートすると例外がスローされる場合がある

ユーザーが CrystalReportViewer コントロールからレポートをエクスポートまたは出力したときに、例外がスローされる場合があります。この問題は、Web ページに CrystalReportViewer コントロールと Microsoft Web コントロールが含まれており、Firefox Web ブラウザでページを表示したときに発生します。

この問題を解決するには

ASP ページで EnableEventValidation に False を設定して、ユーザーが Firefox でレポートを出力またはエクスポートできるようにします。

1.55 アプリケーション検証ツールが混合モードの C++ プロジェクトで有効にならない。代わりに次のメッセージが表示される

"選択したデバッグ モードではアプリケーション検証ツールはサポートされていません。詳細についてはサポート ドキュメントを参照してください。検証ツールを使用せずにデバッグを継続する場合は [OK] をクリックしてください。"

アプリケーション検証ツールは、[デバッグ] メニューで選択できる場合でも、混合モードのアプリケーションでは有効になりません。アプリケーション検証ツールが有効になっておらず、アプリケーション検証ツールを使用せずにデバッグを続行できることを示すダイアログ ボックスが表示されます。

この問題を解決するには

アプリケーション検証ツールを使用せずにデバッグを続行します。

1.56 Visual Studio 2005 Team Edition for Software Testers : 適切な構成であるにもかかわらず、ロード テストの結果が結果データベースに格納されない

ロード テストの構成が適切であるにもかかわらず、ローカル実行のロード テストの結果が結果データベースに格納されません。この状況が発生すると、次のメッセージが表示されます。
"ロード テストの結果データベースを開けませんでした。テスト コントローラ (またはローカル コンピュータ) の接続文字列で指定されたロード テストの結果データベースが、ロード テスト スキーマを含み、現在使用可能なデータベースを指定していることを確認してください。"
ローカル ロード テストの結果データベースは、最初のローカル ロード テスト実行の初期化中に作成されます。このエラーは、最初のロード テストの実行を開始するユーザーが、データベースを作成できるだけの特権を持っていない場合に発生します。

この問題を解決するには

管理者が最初のロード テストの実行を開始して、ロード テストの結果データベースを強制的に作成する必要があります。

1.57 Visual Studio 2005 Team Edition for Software Testers : SQL Server Express に格納されたロード テストの結果に、リモート コンピュータからアクセスできない場合がある

SQL Server Express の構成と Windows ファイアウォールによって、SQL Server Express に格納されたロード テストの結果へのリモート アクセスがブロックされる場合があります。SQL Server Express の既定のインストールでは、データベースへのリモート アクセスは無効になっています。Load Controller のセットアップによって、SQL Server Express の構成と Windows ファイアウォールの構成は SQL Server Express へのリモート アクセスを許可するように設定されます。ただし、セットアップで SQL Server Express をインストールした場合にも、これらの構成変更が行われます。Visual Studio のセットアップでも SQL Server Express をインストールできますが、リモート アクセスを許可するように SQL Server Express と Windows ファイアウォールの構成が自動的に変更されるわけではありません。

ロード テストの結果、ビューアが SQL Server Express に格納されたロード テストの結果にリモート アクセスできない場合、次のメッセージが表示されます。

"結果レポジトリを読み取れませんでした : ロード テストの結果レポジトリにアクセスできませんでした: サーバーへの接続を確立中にエラーが発生しました。SQL Server 2005 への接続時にこのエラーが発生するのは、既定の設定で SQL Server がリモート接続を許可していないことが原因の可能性があります。"

この問題を解決するには

リモート アクセスを許可するよう SQL Server Express と Windows ファイアウォールを手動で構成します。SQL Server Express へのリモート アクセスを有効にするには、以下の手順を実行します。

1. [スタート] メニューで [すべてのプログラム]、[Microsoft SQL Server 2005]、[構成ツール]、[SQL Server 構成マネージャ] の順にクリックして、SQL Server 構成マネージャを開きます。
2. SQL Server 構成マネージャの左ペインで、[SQL Server 2005 ネットワークの構成] のノードを展開し、[SQLEXPRESS のプロトコル] をクリックします。
3. 右ペインで [名前付きパイプ] を右クリックし、[有効化] をクリックします。
4. [TCIP/IP] を右クリックし、[有効化] をクリックします。
5. 左ペインで [SQL Server 2005 のサービス] をクリックします。
6. 右ペインで [SQL Server ブラウザ] を右クリックし、[プロパティ] をクリックします。
7. [プロパティ] ダイアログ ボックスの [サービス] タブをクリックします。
8. [サービス] ページで [開始モード] プロパティを [自動] に設定し、[OK] をクリックします。
9. [SQL Server ブラウザ] を右クリックし、[開始] をクリックします。
10. [SQL Server (SQLEXPRESS)] を右クリックし、[再起動] をクリックします。

Windows ファイアウォールを構成するには、以下の手順を実行します。

1. [Windows ファイアウォール] ダイアログ ボックスを開き、[例外] タブをクリックします。
2. [プログラムの追加] をクリックし、[参照] をクリックして sqlbrowser.exe を検索し、[OK] をクリックします。
3. [プログラムの追加] をクリックし、[参照] をクリックして sqlservr.exe を検索します。[OK] をクリックします。
4. [ポートの追加] をクリックし、[名前] に "SQL Service"、[ポート番号] に "1433" と入力し、[TCP] オプション ボタンをオンにします。
5. [Windows ファイアウォール] ダイアログ ボックスで [OK] をクリックします。

1.58 Long (VT_I8) の相互運用は Windows XP でしかサポートされていない

Long (VT_I8) を渡す Windows 2000 または Windows 9x での遅延バインド呼び出しは失敗します。OLE オートメーション経由の VT_I8 でサポートされているプラットフォームは Windows XP だけです。

この問題を解決するには

Long ではなく整数を使用します。

1.59 EnvDTE80.WindowKinds.vsWindowKindImmediate 定数の値が誤っている

vsWindowKindImmediate 定数は、イミディエイト ウィンドウの正しい GUID を返しません。

この問題を解決するには

vsWindowKindImmediate 定数の代わりに GUID として {ECB7191A-597B-41F5-9843-03A4CF275DDE} を使用します。
また、次のコードによってイミディエイト ウィンドウの正しい GUID を取得することもできます。

```
DTE.Windows.Item("Immediate Window").ObjectKind
```

1.60 共有アドイン ウィザードのセットアップ プロジェクトは、Microsoft Office だけがインストールされたコンピュータでは動作しない

Office のアドインを実行するには、Extensibility.dll が必要です。これは、共有アドインのセットアップ プロジェクトに自動的に含まれるわけではありません。

この問題を解決するには

ソリューション エクスプローラで、アドイン セットアップ プロジェクトを右クリックし、[追加] - [アセンブリ] をクリックします。[.NET] タブで [機能拡張] をクリックし、[OK] をクリックします。アセンブリを追加したら、ソリューション エクスプローラでそのアセンブリを選択し、[登録] プロパティに「vsdraDoNotRegister」が設定されていることを確認します。

1.61 フォントと色のオプションがリセット時にすぐに反映されない

環境設定のリセットは、[ツール] の [設定のインポートとエクスポート] をクリックし、[すべての設定をリセット] を選択することによって実行できます。また、コマンド ウィンドウで "Tools.ImportAndExportSettings /reset" コマンドを実行することでリセットすることもできます。どちらの場合も、Visual Studio を再起動するまでフォントと色のオプションは反映されません。

この問題を解決するには

リセット操作を実行したら、Visual Studio を再起動します。

1.62 アプリケーション検証ツールは、DLL ベースのプロジェクト、ATL Web サービス プロジェクト、および VC スマート デバイス プロジェクトではサポートされていない

アプリケーション検証ツールは、次のシナリオではサポートされていません。
1. DLL プロジェクトで [アプリケーション検証ツールを使って開始] を使用し、ソリューションに含まれていない実行可能ファイルにプロジェクトを読み込む場合。
2. [プロジェクトのプロパティ] ? [デバッグ] の [コマンド] を現在読み込まれているプロジェクト ($TargetPath) 以外のものに変更する場合。これは、リモート デバッグにも当てはまります。
3. ATL Web サービス プロジェクトが別のプロセスで実行されているため、アプリケーション検証ツールがこれらのプロジェクトでサポートされていない場合。
4. VC スマート デバイス プロジェクトが別のアーキテクチャで実行されているため、これらのプロジェクトでアプリケーション検証ツールがサポートされていない場合。
アプリケーション検証ツールは、IDE に読み込まれた EXE プロジェクトでのみ有効になります。

この問題を解決するには

上記のシナリオ 1 と 2 の場合、IDE に対象の実行可能ファイルのプロジェクトを読み込み、アプリケーション検証ツールを使用します。上記のすべてのケースでアプリケーション検証ツールを使用した場合、通常のデバッグ セッション ([デバッグ] - [開始]) と同様に動作します。

1.63 Windows プロジェクトと ASP.NET Web プロジェクトが同じクラス ライブラリを参照している場合、そのクラス ライブラリ内の Web サービス参照がリバース エンジニアリングされないことがある

このシナリオでは、Visual Studio ソリューションに、実装する Windows アプリケーションと ASP.NET アプリケーションを含むアプリケーション ダイアグラムが含まれます。また、ソリューションには、Web サービスへの Web 参照を含むクラス ライブラリも含まれます。Windows プロジェクトと ASP.NET Web プロジェクトは、共にこのクラス ライブラリを参照しています。クラス ライブラリをビルドし、必要なエントリをクラス ライブラリの App.config ファイルから両方のプロジェクトの構成ファイルにコピーした後、Windows アプリケーションまたは ASP.NET アプリケーションの Web サービス コンシューマ エンドポイントとして、Web 参照がリバース エンジニアリングされない場合があります。また、参照先 Web サービスへの接続もリバース エンジニアリングされません。ダイアグラム上にまだ存在しない Web サービスの Web 参照の場合、参照先 Web サービスの外部 Web サービスがリバース エンジニアリングされない場合があります。

この問題を解決するには

1. アプリケーション ダイアグラムを閉じます。
2. Windows プロジェクトと ASP.NET プロジェクトから共有クラス ライブラリへの参照を削除します。
3. クラス ライブラリ参照を ASP.NET プロジェクトに追加します。
4. アプリケーション ダイアグラムを開きます。
5. クラス ライブラリ参照を Windows プロジェクトに追加します。

1.64 Visual Studio 2005 Team Edition for Software Testers : Web テストを実行すると、"オブジェクトのインスタンスに設定されていないオブジェクトです。" というエラーによってテストが失敗する

未使用のデータ ソースが含まれたコード化された Web テストを実行すると、"オブジェクトのインスタンスに設定されていないオブジェクトです。" というエラーが発生し、テストが失敗します。このエラーが発生するのは、コード化された Web テスト内でデータ ソースが定義されているにもかかわらず、テストでデータ ソースがどのアイテムにもバインドされないためです。

この問題を解決するには

コード化された Web テストから未使用のデータ ソースを削除するか、DataBinding 属性をテスト クラスに追加することにより、テストでデータ ソースのフィールドをアイテムにバインドします。

1.65 [データ ソース] ウィンドウからフィールドをコピーして貼り付けた後、コントロールの DataBindings プロパティが正しくない

Windows フォーム デザイナ上にコントロールを生成するために、データベース データ ソースからフィールドを (ドラッグ アンド ドロップではなく) コピーして貼り付けると、データ バインド関連のプロパティが、テーブル内のフィールドではなくテーブルに設定されます。オブジェクトと Web データ ソースはこの問題の影響を受けません。

この問題を解決するには

Windows フォーム デザイナに追加するそれぞれのコントロールのプロパティを編集して、(DataBindings) プロパティを正しいデータ ソースに変更します。

1.66 SQL Server 2005 と同じマシンに VS をインストールすると、VS で Business Intelligence プロファイルが使用されることがある

SQL Server 2005 と同じマシンに Visual Studio 2005 をインストールした場合、VS を初めて起動したときにプロファイルの選択が要求されず、VS プロファイルが Business Intelligence 開発用に設定されていることがあります。これは、Business Intelligence 開発をサポート (Business Intelligence Development Studio をサポート) するバージョンの Visual Studio が SQL Server 2005 によってインストールされ、この目的に応じて VS プロファイルが設定されているために発生します。

この問題を解決するには

VS の [ツール] メニューの [設定のインポートとエクスポート] をクリックすると起動される、設定のインポートとエクスポート ウィザードを使用すると、VS プロファイルをいつでも変更できます。

ウィザードの [すべての設定をリセット] をクリックし、[次へ >] をクリックします。次に、[はい、現在の設定を保存します] をクリック

し、[次へ >] をクリックします。次に、目的のプロファイル ([全般的な開発設定] など) をクリックし、[完了] をクリックします。

場合によっては、新しい設定をすべて有効にするために VS の再起動が必要になります。

1.67 更新プログラムをインストールしていないコンピュータで共有アドインが読み込まれない

アドインが実行されるコンピュータに更新プログラムをインストールしないと、Word または Excel に共有アドインが読み込まれない場合があります。

この問題を解決するには

詳細については、サポート技術情報 (http://go.microsoft.com/fwlink/?LinkId=52419) を参照してください。

1.68 C# Express をアンインストールすると、Visual Studio コンテンツ インストーラの実行に失敗する

C# Express をインストールし、次に Visual Studio の他の Edition をインストールした後、C# Express をアンインストールすると、Visual Studio コンテンツ インストーラの起動に失敗します。

この問題を解決するには

コントロール パネルの [プログラムの追加と削除] を選択し、Visual Studio の [修復] を実行します。

1.69 英語以外のバージョンの Windows オペレーティング システムに .addin ファイルをインストールするときに回避策が必要な場合がある

コンピュータで英語以外の Windows オペレーティング システムを実行している場合、"Documents and Settings\All Users\Application Data\Microsoft\MSEnvShared\Addins" フォルダのパスの一部がローカライズされていると、このパスがアドインで認識されません。

この問題を解決するには

1. Visual Studio 2005 で、[ツール] メニューの [オプション] をクリックし、[アドイン/マクロ セキュリティ] をクリックします。
2. [アドイン ファイル パス] セクションに、.addin ファイルへの完全パスを追加します。

または、.addin ファイルを [システム ドライブ]:\Documents and Settings\All Users\Application Data\Microsoft\MSEnvShared\Addins にインストールします。

1.70 リモート アクセス プロバイダがファイルを自動マージしたあと、警告ダイアログ″行の終わりの不整合″が表示される

Visual Source Safe のマージ機能は UTF-8 ファイルの最後が非ASCII テキストの場合、正確に行の最後を判断できません。
コマンドラインツールで″ssexp /merge″と起動した際にも同じ警告ダイアロが表示されます。コマンドラインツールで UI 操作オプションを付け ″ssexp /merge /a″ と起動した場合はこの警告ダイアロ表示されません。

この問題を解決するには

警告ダイアログ ″行の終わりの不整合″ 上で ″はい″ を選択します。

1.71 Visual Studio 2005 July CTP 日本語版で作成した Windows アプリケーションのプロジェクト ファイルを Visual Studio 2005 製品版で開くと″コンポーネント 'System' への参照は、プロジェクトに既に存在します。″ というメッセージが表示される

Visual Studio 2005 July CTP 日本語版がインストールされたマシンで Windows アプリケーションのプロジェクト ファイルを作成し、その後 Visual Studio 2005 July CTP 日本語版をアンインストールした後、Visual Studio 2005 製品版をインストールして起動した場合、「最近使ったプロジェクト」に July CTP 版で作成したプロジェクト ファイルが表示されます。そのプロジェクトファイルを選択して開こうとすると″コンポーネント 'System' への参照は、プロジェクトに既に存在します。″ というメッセージが表示される場合があります。

この問題を解決するには

このメッセージが表示された場合は一旦 Visual Studio 2005 製品版を終了し、再起動することでプロジェクト ファイルを使用することが可能になります。

2. .NET Framework

2.1 手動テスト (Word 形式) の名前に 2 バイト文字セット (DBCS: Double-Byte Character Set) が含まれていると、テストの実行中および実行後に正しく表示されない

手動テストの実行中には、手動テストの [実行中] ページが表示され、テストの実行が完了すると、手動テストの結果ページが表示されます。

手動テストの名前に DBCS 文字が含まれていると、この 2 つのページに手動テストが正しく表示されません。表示内容は、そのテストの HTML が未加工のまま表示されたように見えます。テストの本文を右クリックし、[最新の情報に更新] をクリックすると、テストが数分間ハングした後、空白になります。

この問題を解決するには

Microsoft Word 形式の手動テストの名前には、1 バイト文字セット (SBCS: Single-Byte Character Set) のみを使用します。テストの [説明] プロパティおよびテストの本文では、DBCS 文字を使用できます。

2.2 64 ビット コンピュータで 32 ビットの Visual Studio 2005 と SQL Server 2005 を実行している場合、誤ったバージョンの msvsmon.exe が起動される

Visual Studio 2005 (x86) および SQL Server 2005 (x86) がインストールされている 64 ビット コンピュータでは、SQLCLR デバッグが失敗します。次のメッセージが表示されます。

```
---------------------------
Microsoft Visual Studio
---------------------------
.NET コードをデバッグできません。'<Machine Name>' 上の SQL Server プロセスにアタッチできませんでした。64 ビット バージョンの Microsoft Visual Studio リモート デバッグ モニタ (MSVSMON.EXE) を使用して、32 ビット プロセスまたは 32 ビット ダンプをデバッグすることはできません。32 ビット バージョンを使用してください。
---------------------------
OK
---------------------------
```

コンピュータのオペレーティング システムが 64 ビットであるため、64 ビット バージョンの msvsmon.exe が自動的に起動されます。アプリケーションは共に 32 ビットであるため、32 ビット バージョンを起動する必要があります。これは、現時点では修正できないアーキテクチャ上の問題です。

この問題を解決するには

ローカル コンピュータで SQLCLR デバッグを行う場合は、ユーザーが 32 ビット バージョンの msvsmon.exe を手動で起動します。

32 ビット バージョンの msvsmon.exe を登録して自動的に起動するようにすると、他の 64 ビット デバッグのシナリオを実行できなくなるため、これはお勧めしません。

2.3 AuthorizationStoreRoleProvider : 承認マネージャから誤ったエラーまたは不明瞭なエラーが返される

AuthorizationStoreRoleProvider は承認マネージャに依存しています。承認マネージャから返されるすべてのエラー メッセージが問題の根本原因を示しているとは限りません。既知の誤ったエラー メッセージまたは不明瞭なエラー メッセージを以下に示します。

"COMException (0x8007052b): パスワードを更新できませんでした。現在のパスワードとして指定された値が間違っています。"
このエラーは、アクセス拒否エラーです。ASP.NET が IIS 5.0、Windows XP IIS 5.1、または Windows Server 2003 上の IIS 5.0 分離モード上に配置され、アプリケーションが統合 Windows 認証を使用するように設定され、かつ、ポリシー ファイルが現在の ASP.NET アプリケーションのディレクトリ構造内にあるという条件がすべて満たされた場合に、このエラーが発生する可能性があります。ASP.NET がローカル コンピュータ アカウントとして実行されているときに、リモートの AD インスタンスまたは ADAM インスタンス内のポリシー ストアにアクセスしようとした場合にも、このエラーが発生する可能性があります。

"追加データがあります"
このエラーは、ポリシー ストアが見つからなかったことを意味します。接続文字列が ADAM インスタンスを指しているが、存在しない ADAM パーティションを参照している場合、このエラーが発生する可能性があります。たとえば、接続文字列が "LDAP://localhost:4000/Cn=storename, DC=Partition1" の場合、ADAM インスタンスに Partition1 が存在しないと、このエラーが発生します。

"指定されたサーバーは、要求された操作を実行できません。"
このエラーは、指定されたサーバーが見つからなかったことを意味します。接続文字列が存在しないサーバーを指しているか、AD または ADAM が待機していないポート番号を使用している場合に、このエラーが発生する可能性があります。

"ArgumentException: パラメータが間違っています"
このエラーは、ユーザーがロールに属しているかどうかを確認するために使用する承認マネージャの LDAP クエリ グループが無効であることを示しています。

この問題を解決するには

上記の各エラー状況について、考えられる原因を検討します。アプリケーションの構成が原因の 1 つとして考えられる場合は、上記の情報を基にアプリケーションの構成を変更または修正します。

2.4 共有ホスト コンピュータでは、**SQL Server Express** のユーザー インスタンスを無効にする必要がある

ASP.NET 2.0 では、SQL Server 2005 Express と統合することにより、ASP.NET 2.0 の新しいアプリケーション サービスの多くで必要とされるデータベースを自動作成できるようになっています。この機能は、対話ユーザーの ID または ASP.NET をホストするワーカー プロセスの ID を使用して実行されるサーバー プロセスの生成を、SQL Server 2005 Express がサポートすることによって実現されています。共有ホスト サーバー上のような信頼性の低い環境では、ASP.NET アプリケーション間で意図しないデータ共有が発生する可能性があるため、SQL Server ワーカー プロセスの生成機能を有効にしないようにする必要があります。

この問題を解決するには

スタンドアロンのインストーラを使用して SQL Server 2005 Express をインストールする場合は、高度なセットアップ オプションを使用することで、ユーザー インスタンス機能のサポートを明示的に無効にできます。

次のように、既存の SQL Server 2005 Express インストールのユーザー インスタンス機能を手動で無効にすることもできます。
1. ローカル ボックス管理者としてログインし、cmd.exe を実行してコマンド ウィンドウを開きます。
2. PATH 環境変数に示されたディレクトリから osql.exe を使用できない場合は、ディレクトリを osql.exe を格納する SQL Server 2005 Express ディレクトリに変更します。
3. SQL Server の親インスタンスに接続します (osql -E -S .\sqlexpress)。
4. 次の SQL コマンドを実行します。

```
exec sp_configure 'show advanced option', '1'
go
reconfigure with override
go
exec sp_configure 'user instances enabled', 0
    go
reconfigure with override
go
```

2.5 フォーム認証の永続 **Cookie** の動作が、セッション ベースの **Cookie** と同じタイムアウト設定を使用するように変更されている

以前のバージョンの ASP.NET では、フォーム認証の永続 Cookie の有効期間は 50 年としてハードコーディングされていました。ASP.NET 2.0 では、フォーム認証の永続 Cookie の有効期限は、現在の日時に構成の "timeout" 属性の値を加えた値に設定されます。既定では、これは永続 Cookie の有効期間が 30 分であることを意味します。有効期間を延長する場合は、"timeout" 属性の値を増やす必要があります。ASP.NET 2.0 では、これは、セッション ベースの Cookie と永続 Cookie の両方に格納されたフォーム認証チケットで新しいタイムアウト値を使用するということです。

この問題を解決するには

永続 Cookie に別のタイムアウト値が必要な場合、開発者は FormsAuthentication.SetAuthCookie などのメソッドを使用して最初に Cookie を設定した後に、フォーム認証 Cookie への参照を取得できます。次に、Response.Cookies[FormsAuthentication.FormsCookieName] の呼び出しによって Cookie への参照を取得できます。開発者は、取得した HttpCookie で Expires プロパティに別の値を設定できます。

2.6 **AuthorizationStoreRoleProvider** の **DeleteRole** メソッドの動作が適切でない

存在しないロールを削除しようとした場合、プロバイダの DeleteRole メソッドは、"要素が見つかりません。" という例外を誤ってスローします。この場合、プロバイダは例外ではなく false を返す必要があります。また、既存のユーザーが属しているロールを削除しようとした場合には、プロバイダは例外をスローする代わりに false を返します。

この問題を解決するには

1. プロバイダの DeleteRole メソッドに対する呼び出しをすべて try-catch ブロック内にラップします。こうしておけば、1 つ以上のメンバを含んだロールを削除しようとしたとき例外がスローされるようにする修正が今後行われても、既存のコードには影響がありません。
2. ロールの削除を試みる前に、ロールが存在するかどうかを確認します。

2.7 **Xml** シリアル化と既定のアプリケーション **ID** 以外の **ID** を使用すると、**ASP.NET** プロファイル機能が失敗する可能性がある

ASP.NET プロファイル機能の既定のプロパティ シリアル化機構は Xml シリアル化です。ASP.NET プロセス ID が ASPNET (IIS 5.0 および IIS 5.1 の場合) または NETWORK SERVICE (IIS 6 の場合) 以外の ID である場合、"InvalidOperationException: 一時クラスを生成できませんでした。" という例外によって、Xml シリアル化プロセスは失敗します。アプリケーションの偽装を使用した場合にも、同じ問題が発生します。これらの例外が発生するのは、XmlSerializer が一時クラス ファイルを %windir%\temp ディレクトリに書き込むため

です。既定では、このディレクトリは、既定の ASP.NET アカウントにだけ "フォルダの一覧/データの読み取り" アクセス許可を付与します。既定の ASP.NET アカウント以外のアカウントも、このディレクトリに一時ファイルを書き込むことはできますが、その後に XmlSerializer が使用する一時クラスを検索するために必要な特権はありません。

この問題を解決するには

セキュリティ保護が必要な環境の場合、開発者は別のシリアル化機構 (バイナリ シリアル化または文字列シリアル化) を使用する必要があります。アプリケーション間で一時クラス ファイルを誤って共有する可能性の低いアプリケーションの場合は、ワーカー プロセス ID またはアプリケーションの偽装 ID に %windir%\temp ディレクトリに対する "フォルダの一覧/データの読み取り" 特権を付与できます。

2.8 SQL Server Express 使用時の ASP.NET のセッション状態の制限

SQL Server Express は、ASP.NET の SQL Server ベースのセッション状態を使用する開発環境でのみ使用してください。これは、SQL Server Express では SQL Server エージェント サービスがインストールされないためです。したがって、セッション状態のクリーンアップ ジョブが実行されることはないため、セッション状態データはデータベースに蓄積されることになります。有効期限の切れたセッションを手動でクリーンアップするには、SQL Server Express に接続し、"EXECUTE DeleteExpiredSessions" コマンドを実行します。

この問題を解決するには

実行用アプリケーションには、SQL Server 2005 の標準バージョンのいずれかを使用します。

2.9 AuthorizationStoreRoleProvider : ApplicationName の既定値が承認マネージャでエラーになる

承認マネージャでは、"/" 文字が含まれるアプリケーション名は許可されません。構成の applicationName が設定されていない場合、AuthorizationStoreRoleProvider は、他の ASP.NET プロバイダで使用されているロジックにしたがって applicationName の既定値を決定します。ASP.NET 内での実行時には、この値が "/" 文字で始まる値になります。

この問題を解決するには

AuthorizationStoreRoleProvider を ASP.NET アプリケーション内で使用するときは、必ず構成で applicationName 属性を設定します。

2.10 Windows 98 および Windows ME で、Web サービスを使用するアプリケーションをデバッグするとアプリケーションがハングする

Visual Studio 2005 では、Windows 98 または Windows ME で、Web サービスを使用するアプリケーションをデバッグするとアプリケーションがハングします。

この問題を解決するには

デバッガ コンポーネントの csm.dll を無効にし、ブレークポイントを使用して Web サービスを停止します。

csm.dll を無効にするには、以下の手順を実行します。

1) Windows の [スタート] メニューの [ファイル名を指定して実行] をクリックします。
2) [ファイル名を指定して実行] ダイアログ ボックスで「regedit.exe」と入力します。
3) HKEY_CLASSES_ROOT\CLSID\{12A5B9F0-7A1C-4FCB-8163-160A30F519B5} に移動します。
4) InprocServer32\(default) の値が "<ドライブ>:\Program Files\Common Files\Microsoft Shared\VS7Debug\csm.dll" であることをチェックし、正しいレジストリ キーに移動したことを確認します。
5) レジストリ キー (HKEY_CLASSES_ROOT\CLSID\{12A5B9F0-7A1C-4FCB-8163-160A30F519B5}) を削除します。

2.11 C# コード スニペットのショートカットに分音文字を含めることはできない

C# コード スニペットは、.snippet ファイルに XML で定義されます。スキーマで可能なタグの 1 つとして、<Shortcut></Shortcut> タグがあります。ショートカットを使用すると、スニペットをすばやく挿入できます。これを実行するには、ショートカット文字列を入力し、Tab キーを押す必要があります。

ショートカット文字列には多くの Unicode 文字を使用できますが、コンプレックス スクリプトの言語で使用される分音文字は、有効なショートカット文字として認識されません。つまり、コンプレックス スクリプトの言語で表記した文字列は、ショートカットとして使用できない場合があるということです。ショートカットに分音文字が含まれている場合、ショートカット文字列を入力し、Tab キーを押すと、タブ文字は挿入されますが、スニペット コードは挿入されません。
メモ : スニペットのショートカットは IntelliSense でも表示されます。

この問題を解決するには

1. 分音文字が含まれていない Unicode ショートカット名を選択します。

2. ショートカットを使用する代わりに、Snippet Picker の UI を使用してスニペットを挿入します。

2.12 System.Net では、DNS タイムアウトが DNS 解決を実行するために System.Net によって呼び出されるアンマネージ API のタイムアウトよりも短い場合に、DNS 解決をタイムアウトするロジックが削除されている

以前のバージョンの .NET Framework では、System.Net の DNS は、長時間のネイティブ タイムアウトを回避するために、DNS 型にタイムアウト ロジックを追加していました。正確な DNS タイムアウトがないと、他の Web 要求の正確なタイムアウトを設定できません。ただし、このタイムアウトを実装するために、DNS 解決は別のスレッドにオフロードされます。スレッド プールのレベルが低い場合、DNS 解決はタイムアウトするまで使用可能なスレッドを待機するため、例外がスローされる場合があります。この例外を防ぐために、DNS タイムアウト ロジックが削除されました。

この問題を解決するには

既知の解決策はありません。

2.13 SqlCacheDependency は、System.Data.SqlClient.SqlDependency.Start の呼び出しによって初期化する必要がある

SQL Server 2005 でのクエリ通知の動作が変更されたため、開発者は SqlCacheDependency を使用する前に、少なくとも 1 回は SqlDependency の Start メソッドを呼び出す必要があります。Start を呼び出すのに適した場所としては、Web アプリケーションの global.asax にある Application_Start メソッド内が挙げられます。

```
void Application_Start(object sender, EventArgs e)
{
System.Data.SqlClient.SqlDependency.Start("接続文字列");
}
```

接続文字列は、SqlCacheDependency で使用するためのコマンドの実行時に使用するものと同じであることが必要です。

2.14 System.Net は、WebClient.UploadValues() の呼び出し時に CRLF 文字列を追加しないように変更されている

以前のバージョンの .NET Framework では、WebClient.UploadValues() の呼び出しによって、アップロードされたコンテンツに CRLF が追加されたため、サーバー アプリケーション エラーになりました。HTML 4.01 仕様に記述されているように、CRLF 文字は application/x-www-form-urlencoded コンテンツ タイプのエンコーディング方式に違反します。CRLF が追加されることはなくなりました。

この問題を解決するには

WebClient.UploadData() を使用して CRLF を追加し、アプリケーションを再コンパイルするか、CRLF 文字を要求しないようサーバー アプリケーションを修正します。

2.15 UriBuilder が、Fragment プロパティまたは Query プロパティに設定した値が後で消去されることがないよう変更されている

UriBuilder 型を使用すると、Uri インスタンスの内容を少しずつ設定できます。以前のバージョンの .NET Framework では、Query プロパティと Fragment プロパティの両方を設定することはできませんでした。Version 2.0 では、このエラーが修正され、Query プロパティまたは Fragment プロパティに値を設定することによって他のフィールドが不用意に消去されることはなくなりました。以前の動作に合わせてアプリケーションのロジックが作成されている場合は、誤った動作を防ぐために修正が必要になる可能性があります。

この問題を解決するには

既知の解決策はありません。

2.16 アプリケーション構成ファイルの System.Net 要素内に存在する値を適用する前に必要なアクセス許可が変更されている

アプリケーション構成ファイルの System.Net 要素内に存在する設定を適用するために必要なアクセス許可が、以前のバージョンの .NET Framework から変更されています。構成ファイルの値を適用するために必要なアクセス許可は、コード内の設定の変更時に、関連付けられたプロパティで必要とされるアクセス許可と同じになりました。

この問題を解決するには

既知の解決策はありません。

2.17 FTP 要求を送信した後に、同じディレクトリに対して SSL (FTPs) を使用して FTP 要求を送信すると、ファイルが見つからないという例外がスローされる

アプリケーションがサーバーに対して FTP 要求を送信した後に、SSL を有効にした状態で同じサーバーに対して同じパスを使用して別の FTP 要求を送信した場合、2 番目の要求は失敗します。この場合、ファイルが見つからないという例外がスローされます。ただし、2 番目の要求が同じパスに対してでなければ、その要求は成功します。

2.18 WebRequest.ServicePoint.Address は、当該のサービス ポイントがプロキシ サーバーの場合にだけ、制限されていない Web アクセス許可を要求する

この新しい要求によって、部分信頼で実行されているアプリケーションがネットワーク プロキシ アドレスを見つけることはできなくなります。ServicePoint.Address API を使用する部分的に信頼されたアプリケーションは、アドレスがプロキシ サーバーを指している場合に SecurityException を受け取る場合があります。

この問題を解決するには

HttpWebRequest.Address を使用します。

2.19 Visual Studio および .NET Framework の 32 ビット動作と 64 ビット動作間のパリティに対する例外

1. IA64 WOW64 対応の FrontPage Server Extensions

.NET Framework 1.1 を実行しているリモートの IA64 コンピュータ上で、.NET Framework 1.1 を使用して Web サイトを作成する場合に、FrontPage はサポートされません。一部の基本機能は、ファイル共有機構によって動作します。

2. J# は、ネイティブ 64 ビットでの実行をサポートしていません。J# コードは、64 ビット プラットフォーム上の WOW64 でのみ実行できます。

3. SQL Server Express は、.NET Framework 2.0 の 64 ビット版ではサポートされていません。

4. IA64 WOW64 の .NET Framework で実行している負荷の高い ASP.NET アプリケーションについては、パフォーマンスまたはスケーラビリティの保証はありません。

5. データ ブレークポイントは、IA64 の WOW64 上で実行している Visual Studio 2005 では機能しません。

6. Visual Studio 2005 では、エディット コンティニュ デバッガ機能はネイティブ 64 ビット環境では動作しません。ただし、64 ビット WOW 環境では動作します。

7. Visual Studio 2005 での VC ATL の例外は次のとおりです。

- ビルド システムは、64 ビットを対象とする dll を WOW64 に登録しません。

- 64 ビット コンピュータ上の既定の ATL サーバー プロジェクトのデバッグ時に、"%1 は有効な Win32 アプリケーションではありません。" というエラー メッセージが表示されます。

- デバッグする実行可能ファイルと URL は、64 ビット構成の ATL サーバー プロジェクト用に事前設定されていません。

- 64 ビット ATL サーバー プロジェクトのデバッグ時に、ユーザーが指定した .srf ファイルで Internet Explorer が起動しません。

- ATL Server の 64 ビット構成に対応するデバッガの既定の種類は、Web サービス デバッガではなく、ローカル Windows デバッガです。

- デバッグ プロパティは、プロジェクト内の新しい構成には反映されません。たとえば、x86 の ATL サーバー プロジェクトで開始してからこのプロジェクトの 64 ビット構成を作成した場合、デバッグ対象を Internet Explorer に変更しないとデバッグは機能しなくなります。

8. ネイティブ 64 ビット環境での相互運用機能デバッグ (マネージとアンマネージの混合モードのデバッグ) はサポートされていません。ただし、64 ビット WOW 環境では動作します。

9. 一部の MDA (Re-entrancy、LoaderLock、PInvokeStackImbalance など) は、64 ビットではサポートされていません。

10. MMX 組み込み命令は、IA64 および x64 の C++ コンパイラではサポートされていません。

11. インライン アセンブリは、IA64 および x64 の C++ コンパイラではサポートされていません。

12. 高水準言語のほとんどの構築要素は、x64 MASM ではサポートされていません。
MASM は IA64 をサポートしていませんが、マイクロソフトではインテルのアセンブラ (ias.exe) を出荷しています。

13. Visual Studio 2005 では、VC++ コンパイラ スイッチである /ARCH:SSE は、x64 および IA64 の VC++ コンパイラではサポートされていません。

14. System.Diagnostics.Process.MainModule API および System.Diagnostics.Process.Modules API は、64 ビット子プロセスの WOW64 で実行すると失敗します。

15. 同じ GUID と CLSID を持つ 32 ビットと 64 ビットの COM+ サービス コンポーネントを同時に登録することはできません。

16. 64 ビット版 SDK には、64 ビット ネイティブ DBGCLR は含まれていません。DBGCLR は WOW64 のみです。

17. 64 ビット版 SDK には、64 ビット ネイティブ DEXPLORE.EXE は含まれていません。DEXPLORE.EXE は WOW64 のみです。

18. x64 および IA64 対応のブートストラップ マニフェスト パッケージはありません。

ClickOnce やその他のアプリケーション用の 64 ビット ブートストラップはありません。.NET Framework がインストールされていない 64 ビット コンピュータで、ClickOnce アプリケーションをインストールしようとすると、"64 ビットのオペレーティング システムでは、このバージョンの .NET Framework 2.0 がサポートされません。アプリケーションの開発元に連絡してください。" というメッセージが表示され、インストールできません。

このエラーは、32 ビット版としてのみ作成されたアプリケーションを WOW64 で実行した場合にも発生します。

19. IA64 上には Visual Studio 2005 はインストールされません。

Visual Studio 2005 は、IA64 上にはインストールされません (IA64 のデザイン時のサポートはありません)。Visual C++ には、ネイティブ IA64 のコマンド ライン ツール インストーラがあります。

20. VC++ を使用して IA64 を対象とする 64 ビット ネイティブ アプリケーションの開発を行なう場合は、Visual Studio Team System をお使いいただくか、「Visual Studio 64-bit Hosted Visual C++ Tools 2005」 をお使いください。また、マネージ アプリケーションの開発を行なう場合は、/platform コンパイラ オプションに、対象とするプラットフォームに対応する引数を設定することで、x86、x64、IA64 のいずれか、またはそれら全てのプラットフォーム上で動作させることが可能です。

21. blittable 型を持つ P/Invoke シグネチャは、64 ビットでは扱いが異なります。これは、64 ビットでは P/Invoke の実装が異なるため、32 ビットで誤って動作した不正な P/Invoke シグネチャが 64 ビットで動作しない場合があるためです。

22. Visual Studio 2005 Tools for Office は、64 ビットではサポートされていません。

2.20 C# Web サイトでのリファクタリング操作の実行時に、実際にはエラーが存在しなくても、ビルド エラーが報告される場合がある

場合によっては、ビルド エラーが含まれていない C# Web サイトでリファクタリング コマンドを実行したときに、プロジェクトまたはその依存関係のいずれかが現在ビルドされておらず、参照を更新できない可能性があることを示す警告が表示されることがあります。
このダイアログは、[続行] または [プレビュー] をクリックすることによって安全に無視できます。リファクタリングは正常に実行され、すべての参照が更新されます。

この警告が表示される一般的なプロジェクト構成の例を次に示します。
- Web ディレクトリに global.asax ファイルが格納され、App_Code ディレクトリに global.asax.cs ファイルが格納された、以前のバージョンの Visual Studio から移行した Web プロジェクト。
- Web ページが Web ユーザー コントロールを参照し、Web ユーザー コントロールが App_Code ディレクトリ内で宣言された型を参照している Web プロジェクト。

この問題を解決するには

既知の解決策はありません。ただし、この警告は安全に無視できます。

2.21 System.Net で登録できるようになった既定の FtpWebRequest 実装によって、独自の FTP コンポーネントを使用するアプリケーションが破損する場合がある

.NET Framework 2.0 以前では、System.Net の拡張性のあるプラグ可能なプロトコル フレームワークをアプリケーションで使用することにより、FTP 要求を処理するコンポーネントを登録できました。さまざまな Web 要求を処理するためのコンポーネントは、コンポーネントを特定の URI プレフィックスに関連付けることによって登録されます。このプレフィックスと一致する Web 要求は、プレフィックスに関連付けられたコンポーネントによって処理されます。.NET Framework 2.0 では、System.Net は、既定で "ftp:" プレフィックスに対して登録される FtpWebRequest コンポーネントをサポートするようになりました。このリリース以前にこのプレフィックスに登録しているアプリケーションは、プレフィックス (FTP) が既に取得されているため、破損する可能性があります。

この問題を解決するには

独自の FTP コンポーネントを登録する前に、次のようにアプリケーション構成ファイルを更新して、既定の FTP プロトコル コンポーネントを削除します。

```
<system.net>
<webRequestModules>
<remove prefix = "ftp:" />
</webRequestModules>
</system.net>
```

2.22 .NET Framework 2.0 で、machine.config ファイルに空の System.Net タグが存在する場合に、GlobalProxySelection.Select の動作が .NET Framework 1.1 と異なる

.NET Framework 1.1 では、machine.config ファイルに空の System.Net 要素を指定すると、GlobalProxySelection.Select は、使用できるプロキシが存在しないことを示す空の Web プロキシ インスタンスを返します。.NET Framework 2.0 では、この場合にシステム プロキシ設定 (Internet Explorer で指定するものと同じ設定) が既定で使用されるようになっています。

この問題を解決するには

次のように、machine.config ファイルを変更して既定のプロキシを無効にします。

```
<system.net>
<defaultProxy enabled="false" />
</system.net>
```

2.23 System.Uri でホストと共に IPv6 スコープ ID が含まれていない

以前のバージョンの .NET Framework では、IPv6 アドレスを使用して Uri インスタンスを作成すると、ホスト アドレスと共にスコープ ID が必ず含まれます。スコープ ID は、ローカル ネットワーク アダプタのインデックスを参照するため、リモート ホストにとっては意味がありません。また、スコープ ID では構文に '%' 文字を使用するため、Uri エスケープ形式の '%hh' と衝突します。ホストにスコープ ID が含まれていると、他の URI パーサーやリゾルバと相互運用するための System.Uri の機能が損なわれます。Version 2.0 の System.Uri では、Uri インスタンスに IPv6 ホストと共にスコープ ID が含まれることはなくなりました。また、System.Uri には、DnsSafeHost という新しいプロパティも追加されています。このプロパティは、IPv6 ホスト アドレスと共にスコープ ID を返します。

この問題を解決するには

IPv6 ホスト アドレスと共にスコープ ID を返す Uri.DnsSafeHost を使用します。

2.24 System.Transactions の構成を使用して、リモート プロキシとしてクラスタを指定すると、クラスタのフェールオーバー時に TransactionException がスローされる場合がある

クラスタに DistributedTransactionManagerName を設定することにより、構成内でクラスタをリモート プロキシとして指定した場合、クラスタのフェールオーバー時に TransactionException がスローされる場合があります。

この問題を解決するには

クラスタをリモート プロキシとして指定するには、コンポーネント サービス MMC を使用して、クラスタをリモート MSDTC として使用するよう MSDTC を構成します。

2.25 IDE でさまざまなアクションを実行しているときに、"Microsoft Visual C# 2005 IntelliSense でエラーが発生しました。" というメッセージが表示される場合がある

場合によっては、エディタでさまざまなアクションを実行しているときに、"Microsoft Visual C# 2005 IntelliSense でエラーが発生しました。ご不便をおかけして申し訳ありません。" というメッセージが表示されることがあります。ソース データの取得時にエラーが発生したこ

とが原因の場合もあります。

このメッセージが表示される原因となる可能性のあるアクションの例を次に示します。
- 外部エイリアスの名前を変更した場合。
- 'runat="server"' 属性が設定されていない Web プロジェクトで "すべての参照の検索" を実行した場合。

これは致命的な状況ではありません。レポート送信ボタンをクリックすることにより、ユーザーは必要な作業を安全に続行でき、データが失われることもありません。

この問題を解決するには

C# 言語サービスの致命的ではないエラー メッセージの表示をすべて無効にするには、2 つの方法があります。これを実行するには、次のレジストリ パスのレジストリを変更します。

[HKEY_CURRENT_USER\Software\Microsoft\VisualStudio\8.0\CSharp\Options\Editor]

(1) "Watson_ReportExceptions"=dword:00000001
指定した値を使用してこのレジストリ キーを追加すると、エラー報告機能が完全に無効になります。

メモ：このレジストリ キーを使用して C# 言語サービスの致命的ではないすべてのエラー メッセージを無効にした場合、このエラーに関連しないシナリオのフィードバックを収集できなくなる可能性があります。

(2) "Watson_DeferSendingUntilLater"=dword:00000000
指定した値を使用してこのレジストリ キーを追加すると、メッセージの表示は無効になりますが、マイクロソフトへのフィードバックの送信は引き続き行うことができます。情報を収集して送信するときに、ごく短時間ですが IDE が応答しなくなります。

2.26 トランザクションの上位変換時に、トランザクションの DistributedIdentifier を取得しようとすると NullReferenceException がスローされる

トランザクションの上位変換時に、そのトランザクションの DistributedIdentifier を取得しようとすると、NullReferenceException がスローされる場合があります。

この問題を解決するには

この問題を回避するには、2 つの方法があります。

トランザクションへのアクセスが同期されていることを確認します。

または

DistributedIdentifier プロパティをチェックする前に、トランザクションの上位変換が完了していることを確認します。

2.27 [Web 参照の追加] を使用して生成されたすべての ASP.NET Web サービス プロキシは、サービスがさまざまな URL で複数のエンドポイントを持つ場合でも、構成ファイルから取得した同じ URL を使用する

Visual Studio 2005 で Web 参照を追加すると、サービスの URL は構成ファイルの AppSettings セクションから自動的に取得されます。Web サービスがさまざまな URL で複数のエンドポイントを持つ場合、複数のプロキシ型が生成されますが、これらはすべて構成ファイルから取得した同じ URL 値を使用します。

この問題を解決するには

次の 2 つの方法が可能です。
1. WSDL ドキュメントを編集し、各サービスを個別のドキュメントに分割します。

または
2. 構成ファイルを編集して Web サービスごとにキーを追加し、System.Configuration.ConfigurationSettings.AppSettings を使用してコード内の値を読み取ります。これらの値を使用してプロキシ インスタンスの URL プロパティを設定します。

2.28 生成される ASP.NET Web サービス プロキシにおける既定の Nullable<T> のサポートによって、プロキシの再生成時に既存のコードが破損する可能性がある

ASP.NET Web サービスの nullable="true" 属性が含まれたスキーマをインポートすると、生成されるプロキシに Nullable<T> 型が含まれるため、以前の属性は無視されます。この変更により、デザイン時または実行時にアプリケーションが破損する可能性があります。

この問題を解決するには

可能な回避策は、プロキシ型を再生成しないようにすること、新しい Nullable<T> パターンを使用してコードを再コンパイルすること、または nullable="true" 属性を使用しないようにスキーマを変更することです。

2.29 Web サービス プロキシ インスタンスで Proxy プロパティに null を設定しても、要求がサーバーに直接送信されない

Web サービス プロキシ インスタンスで Proxy プロパティに null を設定すると、要求はすべての Web プロキシ サーバー設定を迂回し、サーバーに直接送信するよう強制されます。ただし、この機能は現在動作していません。null 値は無視され、代わりに構成済みの Web プロキシ サーバー設定が使用されます。

この問題を解決するには

次のように、プロキシ型を派生させ、GetWebRequest メソッドをオーバーライドすることにより、基になる WebRequest の Proxy プロパティを直接設定できます。

```
protected override WebRequest GetWebRequest(Uri uri)
{
WebRequest req = base.GetWebRequest(uri);
req.Proxy = this.proxy;
return req;
}
```

2.30 .NET Framework の異なるバージョン間で .NET Remoting およびランタイム シリアル化を使用する場合の注意事項

(.NET Remoting またはランタイム シリアル化を使用して) バージョンの異なる .NET Framework で実行しているアプリケーション間でデータを交換するときに、例外が発生する場合があります。これらの例外は、特定の型に Framework の異なるバージョン間での互換性がないこと、つまり、Framework の異なるバージョン間でその型を送信できないことを示します。

.NET Framework 2.0 には、この問題を解消するための Version Tolerant Serialization と呼ばれるテクノロジが含まれています。ただし、以前のバージョンの Framework では、やはりこの問題が発生します。そのため、一部の Version Tolerant Serialization 機能を .NET Framework 1.1 に追加する更新プログラムを提供する予定です。この更新プログラムには Service Pack 1 が必要となります。更新プログラムをインストールすると、更新プログラムを適用した Framework で実行しているアプリケーションは、このバージョン管理の問題に遭遇することなく、.NET Framework 2.0 で実行しているアプリケーションとやりとりできるようになります。.NET Framework 1.0 に対しては、このような更新プログラムを提供する予定はありません。

直接または .NET Remoting を通じてバイナリ シリアル化を使用する際に、バージョン管理に対処するのは Version Tolerant Serialization だけであるという点に注意してください。Framework のさまざまなバージョン間で SOAP シリアル化を使用する場合 (SoapFormatter クラスから直接使用するか、.NET Remoting を通じて使用します) には、前述のバージョン管理の問題が発生する可能性があります。

この問題を解決するには

この更新プログラムを入手するには、サポート技術情報の文書 (http://support.microsoft.com/?kbid=907262) を参照してください。

2.31 System.Transactions のトレースに既定以外のトレース リスナを指定すると部分信頼で動作しない

構成で System.Transactions のトレースに特定のトレース リスナを指定することは部分信頼でサポートされていないため、これを指定すると例外がスローされます。

この問題を解決するには

既定のトレース リスナは、部分信頼でサポートされています。そのため、構成でリスナが指定されていない場合、トレースは既定のリスナによってキャッチされ、debug.outputstring に出力されます。これらのトレースは、DBMon.exe を使用することで部分信頼で表示できます。

2.32 .NET Framework 2.0 に移行後、Web サービスまたは XML シリアル化のシナリオでの日付と時刻の計算が間違っている場合がある

.NET Framework 2.0 では、XML に対する日付と時刻の書き込みと読み取りの方法が変更されています。この変更の影響を受けるのは、主に次のシナリオです。
- タイム ゾーンが指定されていない時刻、または UTC タイム ゾーンの時刻を使用する場合
- タイム ゾーンまたは UTC タイム ゾーンを指定せずに XML に時刻値を書き込むサードパーティ製ソフトウェアとの相互運用性のシナリオ

この変更によって、一部の日付と時刻の計算および比較に誤りが生じる場合があります。これらのケースでは、以前の日付/時刻の動作に戻すことが必要な場合があります。

この問題を解決するには

以前の日付/時刻の動作に戻すには、次の構成変更を適用します。

```
<system.xml.serialization>
<dateTimeSerialization mode="Local" />
</system.xml.serialization>
```

2.33 SQL Server 2005 Express と IIS を使用した ASP.NET データベース自動作成に対応する Windows ユーザー プロファイル

SQL Server Express (SSE) と IIS を使用した ASP.NET の SQL Server プロバイダ用 MDF 自動作成プロセスは、IIS が ASP.NET のローカル コンピュータ アカウントまたは NT AUTHORITY\NETWORK SERVICE として実行されている場合にのみ機能します。この自動作成プロセスは、Visual Web Developer Web Server を使用して対話的に開発している場合 (ファイル ベースの Web サイト) にも機能します。ただし、別のローカル コンピュータ アカウントまたはドメイン ユーザー アカウントを使用して、IIS に対して開発作業を行う場合、データベース自動作成プロセスは失敗します。これは、これらのアカウントには Web サーバーで使用できる Windows ユーザー プロファイルがないためです。

この問題を解決するには

この問題の解決策はありません。IIS ベースの Web サイトのための SSE を使用したデータベース自動作成が機能するのは、ASPNET アカウントおよび NETWORK SERVICE アカウントを使用している場合だけです。

2.34 SQL Server 7.0 または 2000 を使用する ASP.NET プロバイダにおける Unicode サロゲートの動作

SQL Server を使用する ASP.NET プロバイダは、SQL Server 7.0 および SQL Server 2000 で提供されている Unicode サロゲートのサポートのレベルによって制約を受けます。SQL Server 7.0 および 2000 は、情報を損なうことなくサロゲート ペアを格納および取得します。ただし、サロゲートの言語比較は行いません。これらのバージョンの SQL Server を使用している場合、比較演算の実行時および一意性の検証時にサロゲート文字は無視されます。たとえば、これは SqlMembershipProvider がサロゲート文字だけが含まれた 2 つのユーザー名を同一と見なすということです。サロゲート文字だけが既存のユーザーとは異なるユーザー名で 2 つ目のユーザーを作成しようとした場合に、この動作は一意性の制約違反の原因となります。

この問題を解決するには

これらのバージョンの SQL Server に関する既知の解決策はありません。

2.35 FileWebRequest.PreAuthenticate が常に false を返す

WebRequest から継承した型は、PreAuthenticate プロパティを継承します。このプロパティは、サーバーからのチャレンジを待機せずに、認証情報を要求と共に送信できるようにするために使用します。FileWebRequest は事前認証をサポートしていないため、PreAuthenticate プロパティは常に false を返します。以前のバージョンの .NET Framework では、アプリケーションによって PreAuthenticate プロパティに true が設定された場合、後で値が照会されたときに、このプロパティは true を返していました。今後、このようなことはなくなります。FileWebRequest の PreAuthenticate プロパティは常に false を返します。

この問題を解決するには

FileWebRequest の PreAuthenticate プロパティを使用しないようにします。

2.36 .NET Remoting と SOAP シリアル化でのジェネリック型の使用

.NET Remoting テクノロジでは、バイナリ シリアル化と SOAP シリアル化の両方をサポートしています。つまり、リモート処理メッセージは、Microsoft 固有のバイナリ形式または SOAP XML で表すことができます。ただし、ジェネリック型 (.NET Framework 2.0 の新機能) を使用できるのはバイナリ シリアル化だけです。.NET Remoting を通じて行うか、SoapFormatter クラスを使用して直接行うかを問わず、SOAP シリアル化でのジェネリックの使用はサポートされていません。

この問題を解決するには

この問題の既知の解決策はありません。

2.37 C# Web サイトでのリファクタリング操作の実行時に、実際にはエラーが存在しなくても、ビルド エラーが報告される場合がある

場合によっては、ビルド エラーが含まれていない C# Web サイトでリファクタリング コマンドを実行したときに、プロジェクトまたはその依存関係のいずれかが現在ビルドされておらず、参照を更新できない可能性があることを示す警告が表示されることがあります。
このダイアログは、[続行] または [プレビュー] をクリックすることによって安全に無視できます。リファクタリングは正常に実行され、すべての参照が更新されます。

この警告が表示される一般的なプロジェクト構成の例を次に示します。
- Web ディレクトリに global.asax ファイルが格納され、App_Code ディレクトリに global.asax.cs ファイルが格納された、以前のバージョンの Visual Studio から移行した Web プロジェクト。
- Web ページが Web ユーザー コントロールを参照し、Web ユーザー コントロールが App_Code ディレクトリ内で宣言された型を参照している Web プロジェクト。

この問題を解決するには

既知の回避策はありませんが、この警告は安全に無視できます。

2.38 .NET Remoting またはバイナリ シリアル化でジェネリック型を使用すると、バージョン管理機能が機能しないことがある

.NET Remoting とバイナリ シリアル化は、疎結合モード (FormatterAssemblyStyle を Simple、includeVersions を False、strictBinding を False に設定して選択) で機能できます。このモードでは、シリアル化されたアセンブリのバージョンが、シリアル化を解除して読み込まれるアセンブリのバージョンと一致する必要はありません。この機能は、簡単にバージョン管理を行えるようにする目的で用意されています。たとえば、.NET Remoting サーバーを更新しても、クライアントは更新する必要がありません。

ただし、このバージョン管理が機能しない場合があります。具体的には、ジェネリック型を使用している場合、ジェネリック パラメータ型のアセンブリのバージョンを変更すると、疎結合モードでもシリアル化の解除に失敗することがあります。たとえば、MyType2 型の型パラメータを使用してジェネリック型 MyType1 のシリアル化を解除する場合、MyType2 が含まれたアセンブリをシリアル化したときのバージョンと、そのシリアル化を解除するときのバージョンが異なると、シリアル化の解除に失敗することがあります。

この問題を解決するには

この制限を回避するには、バージョン管理が重要な状況では .NET Remoting およびバイナリ シリアル化でのジェネリック型の使用を避けます。それ以外の場合は、.NET Framework のアセンブリ バインド リダイレクト機能を使って、存在しないアセンブリのバージョンに対する読み込み要求を、シリアル化の解除を実行するコンピュータ上に実際に存在するバージョンにリダイレクトします。その他の回避策として、AssemblyResolve イベントを使用してアセンブリの読み込みエラーを検出し、その名前の一部のみを使用してアセンブリの読み込みを再試行する方法もあります。

2.39 FtpWebRequest で KeepAlive が有効になっている場合、ファイルを取得するコマンドの後にディレクトリを一覧表示するコマンドを発行すると、要求と共に送信されるファイル パスの内容が変更される

FtpWebRequest で KeepAlive プロパティが true に設定されている場合、ファイルを取得するコマンドの後に、ディレクトリを一覧表示するコマンドを発行すると、送信されるファイル パスの内容が変更されます。目的のファイルを示すパスの後に、余分なパス区切り文字が追加されます。この動作によって悪影響を受けるのは一部の FTP サーバーのみですが、該当するサーバーでは、ファイルが見つからないというエラーが返される原因になります。

この問題を解決するには

ディレクトリを一覧表示するコマンドを発行する要求で KeepAlive プロパティを false に設定し、その後の取得では KeepAlive を true に戻します。

3. スマート デバイスのプログラマビリティ

3.1 F11 キーによる DLL 関数へのステップ インは、X86 スマート デバイス プロジェクトでは機能しない

1. x86 プラットフォームを対象としたエクスポートされた DLL 関数 (DllFunc) が含まれた、Win32 スマート デバイス プロジェクト DLL を作成します。
2. この DLL を LoadLibrary してその DLL 関数 (DllFunc) を呼び出す Win32 スマート デバイス プロジェクト EXE を作成します。また、このプロジェクトがスタートアップ プロジェクトとして設定されていることを確認します。
3. ソリューションのコンパイル、配置、およびデバッグを行います。
4. DLL 関数 (DllFunc) にステップ イン (F11 キー) してみます。x86 プラットフォーム上でのみ、GetProcAddress から取得したこの DllFunc() に対して [F11] キーでステップ インせず、[F10] キーと同様にステップ オーバーしてしまいます。

この問題を解決するには

デバッガ ウィンドウでこの DLL 関数 (DllFunc) を表示して 16 進数アドレスを取得し、逆アセンブリでブレークポイントを設定します。

3.2 ネイティブ プロセスへのマネージ アタッチでエラーが返されない

1. ネイティブのコンソールの SmartDevice プロジェクトを作成します。
2. Ctrl キーを押しながら [F5] キーを押します。
3. [ツール] - [プロセスにアタッチ] をクリックします。
4. [スマート デバイス] トランスポートとターゲット デバイスを選択します。
5. デバッグ エンジン選択ダイアログでマネージ デバッガを選択します。
6. ネイティブ プロセスを選択し、アタッチします。
7. エラー メッセージが表示されません。

この問題を解決するには

既知の解決策はありません。

3.3 中断モードで既存のブレークポイントにフィルタを適用すると、ブレークポイントが無視される

1) 次のコードのネイティブ アプリケーションを作成します。

```
void foo(void)
{
}

int _tmain(int argc, _TCHAR* argv[])
{
   int i = 0;

   foo();

   return 0;
}
```

2) "return 0;" の位置にブレークポイントを設定します。
3) デザイン時に、foo() 関数にブレークポイントを設定します。
4) [F10] キーを押して、アプリケーションを中断モードにします。
5) ブレークポイント ウィンドウを開き、関数ブレークポイントのフィルタ プロパティを編集して、フィルタとして ProcessName または ThreadName を含めます。
6) 実行します ([F5] キー)。
7) 関数ブレークポイントではヒットせず、代わりに "return 0" ステートメントで中断します。

上記のシナリオは、中断モードでフィルタ プロパティを編集してデザイン時のブレークポイントを変更すると、ブレークポイントが無視されることを示しています。これは、すべての種類のブレークポイントに当てはまります。

この問題を解決するには

ブレークポイントを設定し、デザイン時にフィルタを適用します。適用できるのは ProcessName フィルタのみです。ThreadName フィルタはデザイン時には ThreadId と ProcessId が判別できないため適用できません。

3.4 ネイティブ デバッグで、**while** ループの記述がその構成要素と共に **1** 行のソース行でまとめて記述されている場合、その **while** ループおよび構成要素での **[F10]** キーまたは **[F11]** キーの動作はステップ オーバーになる

以下の例で説明するシナリオでは、[F10] キーまたは [F11] キーは適切に動作しません。

コード スニペットの例

```
1 int i = 0;
2 while (i < 1000) { i++; }
3 return 0;
```

手順：
- 1 行目で F9 キーを押し、この行でブレークポイントにヒットしたら F10 キーを押します。

- これで制御が 2 行目に移ります。
- F10 キーをもう一度押します。

上記の手順を実行した後に生じる問題は、デバッガが 2 行目ではなく 3 行目で停止または中断することです。つまり、F10 キーを繰り返し押しても、2 行目のブレークポイントは無視されます。
これは、ループのすべての構成要素が 1 行のソース行にまとめて記述されているため発生します。

この問題を解決するには

while ループが含まれたソース行 (2 行目) で、F9 キーを押してブレークポイントを設定します。これで、while ループをステップ スルーする間 (i が 1000 までインクリメントされるまで)、2 行目で F10 キーを押すたびにデバッガが中断するようになります。

または

コードを編集できる場合は、前述のコードを次のように変更することにより、期待するように動作させることはできます。

```
1 int i = 0;
2 while (i < 1000)
3 { i++; }
4 return 0;
```

3.5 スマート デバイス アプリケーションのネイティブなアサート ダイアログによって、画面にボタンが表示されなくなる場合がある

Pocket PC または Smartphone で、Visual Studio 2005 の C++ スマート デバイス アプリケーションによってアサート ダイアログがスローされたときに、ダイアログ ボックスのボタンが画面に表示されなくなる場合があります。

この問題を解決するには

Pocket PC または Smartphone のスクロール ボタンまたはスクロール キーを使用して、ボタンまで手動でスクロールします。

3.6 スマート デバイス プロジェクトの [新しい場合はコピーする] プロパティの意味

スマート デバイス プロジェクトでは、dll ファイルまたは exe ファイルの "新しさ" (更新による違い) は、Win32 バージョンでの比較によって判断されます。
- デバイス側ファイルのバージョンがデスクトップ側ファイルのバージョンよりも古い場合、ファイルはコピーされます。
- デバイス側のファイルのバージョンがデスクトップ側ファイルのバージョンよりも新しい場合、ファイルはコピーされません。
- チェックサムが異なる場合、ファイルはコピーされます。

dll および exe 以外のファイルについては、"新しさ" はチェックサムのみに基づいて判断されます。
タイムスタンプやサイズでは判断されません。

この問題を解決するには

既知の解決策はありません。

3.7 TCP/IP デバイスへの接続時にエラーが発生した場合、Visual Studio 2005 は ActiveSync デバイスに接続する

あるデバイス (例 : PPC) が ActiveSync 経由で接続されており、TCP/IP を使用して WinCE デバイス (例 : CEPC) に接続したときに、WinCE デバイスへの TCP/IP 接続が何らかの理由で失敗した場合、Visual Studio 2005 は ActiveSync 経由で接続されたデバイスに接続します。

この問題を解決するには

別のデバイスへの TCP/IP 接続を試みる前に、ActiveSync 経由で接続されているデバイスの接続を解除します。

3.8 オートメーションをサポートし、MFC ライブラリに動的にリンクされている MFC DLL を配置すると、登録時にエラー 0x8007007e が発生して登録できない

オートメーションをサポートし、MFC ライブラリに動的にリンクされている MFC DLL プロジェクトを配置するときに、Visual Studio 2005 では、プロジェクト フォルダのデバイスに必要な依存 DLL と共にこの DLL を配置します。DLL の登録に使用する LoadLibrary() は、conman コンポーネントが配置されたディレクトリで依存ライブラリを検索し、次に \Windows ディレクトリで検索します。このため、DLL の登録は失敗します。

この問題を解決するには

プロジェクト フォルダからデバイス上の \Windows ディレクトリに、必要な依存 DLL を手動で移動またはコピーします。

3.9 短いリース期間で構成された **DHCP** サーバーを持つワイヤレス ネットワークの使用時に、デバイス エミュレータへの接続性が不安定になる

ワイヤレス ネットワーク上では、仮想スイッチ ドライバを使用したデバイス エミュレータへの TCP/IP 接続が不安定になる場合があります。これは、DHCP サーバーの通信の実装方法、仮想スイッチ ドライバ、および 802.11 仕様が原因です。

この問題を解決するには

デバイス エミュレータへの接続性を安定させるには、次のトランスポートのいずれかを使用します。
1. DMA トランスポート

または
2. ループバック アダプタ上の TCP/IP トランスポート

3.10 MUI : ローカライズ版のリモート ツールを言語が一致しないシステム ロケールで実行すると、デバイス エミュレータに接続できない

Visual Studio 2005 のローカライズ版 (日本語版など) を MUI OS の英語版にインストールすると、リモート ツールはデバイス エミュレータに接続できなくなります。これは、リモート ツールは常にシステム ロケールの設定で動作するためです。同じ理由から、Visual Studio 2005 の 2 種類の言語版 (日本語版と英語版など) を MUI OS の英語版に同時にインストールした場合、リモート ツールは常に英語版のデバイス エミュレータに接続します。

この問題を解決するには

使用したい Visual Studio 2005 の言語に一致するようにシステム ロケールの設定を変更します。

3.11 VC++ のスマート デバイス アプリケーションの既定のプロジェクト設定では **/Os** コンパイラ オプションが設定されているため、デバッグを適切に行うことができない場合がある

VC++ のスマート デバイス アプリケーションの既定のプロジェクト設定では、コンパイラ最適化オプションである /Os が設定されています。/Os (コード サイズを最小にする) は、速度よりもサイズを優先するようコンパイラに指示することにより、出力される実行可能ファイル、DLL のサイズを最小限に抑えます。このオプションを使用すると、この設定用に生成されるコードが原因でデバッグを適切に行うことができなくなる場合があります。たとえば、ステップ実行時に、デバッガが if ステートメントと else ステートメントの両方にステップする可能性があります。

この問題を解決するには

コンパイラ最適化設定で /Os オプションの [速度またはサイズを優先] プロパティを [なし] に変更します。
手順
1) プロジェクトの [プロパティ ページ] ダイアログ ボックスを開きます。
2) C/C++ フォルダをクリックします。
3) [最適化] プロパティ ページをクリックします。
4) [速度またはサイズを優先] プロパティを [なし] に変更します。

これで、正しくデバッグを実行できるようになります。ただし、実行可能ファイルのサイズが若干増えます。

3.12 ローカリゼーション用にマークされたフォームを備えた **.NET Compact Framework 1.0** アプリケーションのコンパイル時のビルド エラー

ローカライズを必要とする .NET Compact Framework 1.0 を対象としたアプリケーションを開発する場合、SDK が見つからないことに関するエラーが発生します。

メッセージ :
".NET Compact Framework 1.0 を対象にするプロジェクトのローカリゼーションに必要な .NET Framework SDK Version 1.1 が見つかりません。"

この問題を解決するには

.NET Framework SDK Version 1.1 は、Visual Studio 2005 製品には同梱されていません。SDK をダウンロードしてインストールする必要があります。

次のリンクから SDK をダウンロードしてください。
http://www.microsoft.com/japan/msdn/netframework/downloads/

3.13 SqlCEResultSet DataSource をドラッグ アンド ドロップしても、デザイナ コード ファイルにコントロールのコードが生成されない

デバイスの Windows フォーム デザイナで、DataSource ウィンドウから SqlCEResultSet DataSource をドラッグ アンド ドロップしても、デザイナ コード ファイル (例：Form1.designer.cs) にコントロールのコードは生成されません。
これは、次のような場合に発生します。
1. コントロール生成オプションが [DataGrid] に設定されている場合：DataGrid を作成し初期化するためのコードは、デザイナ コード ファイルに生成されません。
2. コントロール生成オプションが [詳細] に設定されている場合：詳細ビューでコントロールを作成し初期化するためのコードは、デザイナ コード ファイルに生成されません。

この問題を解決するには

次のいずれか 1 つを実行します。
1. ドラッグ アンド ドロップ前：[プロパティ] ウィンドウまたはデザイン サーフェイスのコンテキスト メニューから、フォームのプロパティを編集します。
例：デザイン サーフェイスで右クリックし、[スキンの表示] をオフにします。[スキンの表示] をオンに戻すには、この手順を繰り返します。

または
2. ドラッグ アンド ドロップ前：ツールボックスからコントロールをフォームに追加します。

または
3. ドラッグ アンド ドロップ後：[プロパティ] ウィンドウまたはデザイン サーフェイスのコンテキスト メニューから、フォームのプロパティを編集します。

または
4. ドラッグ アンド ドロップ後：ツールボックスからコントロールをフォームに追加します。

3.14 MFC ライブラリに動的にリンクされた MFC ActiveX コントロールを配置すると、登録時にエラー 0x8007007e が発生して登録できない

ActiveX コントロールの登録に使用する LoadLibrary() API では、依存ライブラリを探す際に、まず conman コンポーネントが配置されたディレクトリ内を検索し、次に \Windows ディレクトリ内を検索します。このため、ActiveX コントロールの登録は失敗します。

この問題を解決するには

必要な MFC DLL をデバイスの \Windows ディレクトリにコピーします。たとえば、Visual Studio 2005 が C: ドライブにインストールされている場合、armv4 プロセッサの MFC DLL は C:\Program Files\Microsoft Visual Studio 8\VC\ce\Dll\armv4 にあります。

3.15 デバイス プロジェクト内の 1 つ以上の静的テキスト コントロールが実行時に表示されない

ダイアログ ベースの MFC アプリケーションをスマート デバイス上で実行したときに、デバイス上で静的テキスト コントロールのいずれかが実行時に表示されません。考えられる原因は次のいずれかです。

1. スマート デバイス用の MFC ベースのダイアログ アプリケーションを作成するためのコードがウィザードで生成されたコードであり、ダイアログに開発者が追加した複数の静的テキスト コントロールが含まれている。
または
2. スマート デバイスを対象とする VC++ アプリケーション用に開発されたカスタム コードが "解像度対応" をサポートしている。

この問題を解決するには

スマート デバイスを対象としたダイアログ ベースの MFC アプリケーションのコードとして、ウィザードで生成されたコードを使用する場合、既定ではウィザードは "解像度対応" アプリケーションのコードを生成します。解像度対応アプリケーションでは、DRA (Device Resolution Aware) API を使用します。横向きモードと縦向きモードをサポートするために、生成されたコードは API DRA::RelayoutDialog() を呼び出します。

API DRA::RelayoutDilog() では、各静的テキスト コントロールに一意の ID を必要とします。ただし、既定では、開発者がフォーム上でドラッグ アンド ドロップによって静的テキスト コントロールを追加すると、静的テキスト コントロールのすべての ID は IDC_STATIC として生成されます。

実行時にすべてのコントロールを表示するには、これらの静的テキスト コントロールの ID を IDC_STATIC_1、IDC_STATIC_2、IDC_STATIC_3 のように変更し、ID を一意にします。これにより、"解像度対応" アプリケーションの実行時に、すべての静的テキスト コントロールが確実に表示されるようになります。

静的テキスト コントロールの ID を変更するには、以下の手順を実行します。

1. フォーム上のコントロールを選択します。
2. 右クリックし、[プロパティ] をクリックします。
3. [ID] プロパティを選択します。
4. すべての静的テキスト コントロールの IDC_STATIC を、IDC_STATIC_2、IDC_STATIC_3 のように一意の識別子に変更します。

3.16 デバイス上で実行しているスマート デバイスの WinSock サーバーが OnConnect コールバック通知を受け取る

スマート デバイスのソケット サーバー実装には、WinCE を対象とする CSocket または CAsyncSocket を使用して実装される OnConnect() メソッドと OnAccept() メソッドがあります。クライアントがデバイスでホストされたこのサーバーへの接続を確立すると、サーバーは OnAccept() 通知と OnConnect() 通知の両方を受け取ります。

この問題を解決するには

サーバーで OnConnect() 通知を受け取らないようにするには、以下の手順を実行します。

サーバー ソケットでの Accept() の呼び出し前

1. サーバー ソケットのサブスクライブ先のイベントを使用して、AsyncSelect メソッドを呼び出します。"socket" がサーバー ソケットのオブジェクトの場合、次のようになります。
```
socket.AsyncSelect(FD_READ | FD_WRITE | FD_OOB | FD_ACCEPT | FD_CLOSE);
```
サーバーで Connect イベントを明示的にマスクするために、FD_CONNECT 以外のすべてのイベントを使用します。

2. サーバー ソケットで他のイベントをマスクする必要がある場合は、サーバー ソケットが関係しているイベントのリストを使用します。たとえば、サーバー ソケットで読み取り操作に関するイベントを受け取る必要がない場合は、次のように、AsynchSelect メソッド呼び出しのイベントのリストから FD_READ も削除します。

```
socket.AsyncSelect(FD_WRITE | FD_OOB | FD_ACCEPT | FD_CLOSE);
```

これで、FD_CONNECT イベントと FD_READ イベントがマスクされます。

ASyncSelect メソッドの動作の詳細については、
http://msdn2.microsoft.com/zh-tw/library/5yhyb5ef.aspx
を参照してください。

3.17 .vcw ファイルをダブルクリックしても、Visual Studio 2005 で eMbedded Visual C++ 4.0 (eVC4) プロジェクトを開くことができない

開発者が eVC4 を使用してデバイス プロジェクトを作成しました。 Visual Studio 2005 のインストール後、eVC4 ワークスペース ファイル "project.vcw" をダブルクリックして、Visual Studio 2005 で eVC4 プロジェクトを開こうとしています。

この問題を解決するには

標準の eVC4 プロジェクトは、Visual Studio 2005 プロジェクトとして登録されません。この標準の eVC4 プロジェクトを Visual Studio 2005 と互換性のあるソリューション ファイルに変換するには、以下の手順を実行します。

1. Visual Studio 2005 を開きます。
2. [ファイル] - [開く] の [プロジェクト/ソリューション] をクリックします。
3. [プロジェクトを開く] ダイアログ ボックスで、vcw ファイルに移動し、ファイルを選択して [開く] をクリックします。
4. 変換について [はい] をクリックし、このプロジェクトの質問を開きます。

3.18 構造体に参照型が含まれている場合、デバイス上で構造体をマーシャリングすると "NotSupportedException" が返される

デバイス プロジェクトでは、次のコードの SizeOf ステートメントは NotSupportedException をスローします。

```
Imports System.Runtime.InteropServices.MarShal
Module Module1
Structure S1
Public RefType As String
End Structure
Sub Main()
Dim sz As Integer
Dim struct As New S1
struct.RefType = "strings are reference types"
sz = SizeOf(struct)
End Sub
End Module
```

この問題を解決するには

参照型に属性を適用することにより、適切にマーシャリングされます。次に例を示します。

```
<MarshalAs(UnmanagedType.ByValTStr, SizeConst:=3)> Public x As String
```

詳細については、"MarshalAsAttribute" に関するドキュメントを参照してください。

3.19 x86 プラットフォームまたは SH4 プラットフォーム上で、[名前を付けて保存] パス選択ダイアログ ボックスが表示されない

ネイティブ MFC プロジェクトで、アプリケーション メニューの、ウィザードで生成された [名前を付けて保存] オプション ([ファイル] の [名前を付けて保存]) をクリックしたときに、[名前を付けて保存] ダイアログ ボックスが表示されません。

この問題を解決するには

コモン ダイアログ ボックス ライブラリの "GetSaveFileName" 関数を使用します。
たとえば、CWinApp 派生クラスのファイル名を作成するには、次の作業を行う必要があります。

メッセージ マップ エントリを作成し、ID_FILE_SAVE_AS コマンドのハンドラを登録します。
これは、次の方法で行うことができます。
- IDE の使用：
1. [名前を付けて保存] メニュー項目を右クリックします。[イベント ハンドラの追加] オプションをクリックします。
2. ハンドラの名前と説明を入力します。ハンドラを関連付けるクラスを選択します。
3. [追加して編集] オプションを選択して適切なエントリを作成します。ハンドラ内に記述する必要のあるコードについては
以下で説明しています。

- ソース ファイルの編集
1. BEGIN_MESSAGE_MAP マクロと END_MESSAGE_MAP マクロの間に、ON_COMMAND(ID_FILE_SAVE_AS, your_handler) 型のエントリを追加します。
2. 適切なクラスのヘッダー ファイルで、ハンドラを "afx_msg void your_handler();" として追加します。
3. 以下の説明を参考にして、ソースでこのクラスにこの関数をメンバ関数として実装します。

- ハンドラのコードでは、以下をこの順序で実行する必要があります。
1. OPENFILENAME 構造体を宣言し、適切なフィールドを初期化します (OPENFILENAME に関するドキュメントを参照)。
2. この構造体への参照を使用して GetSaveFileName 関数を呼び出します。
これにより、ファイル名選択ダイアログが作成されます。
3. 構造体の "lpstrFile" フィールドに、パス選択ダイアログ ボックスを使用して選択されたファイル名を格納します。

-ハンドラはこの選択された名前を使用することもできます。

3.20 ネイティブ デバッグで、x86 のクラス コンストラクタに対して [F11] キーが機能しない

CEPC x86 デバイスを対象としているときに、ネイティブ デバッグでコンストラクタ呼び出しにステップ インしても動作しません。

例：

```
class Foo
{
private:
int i;
```

```
public:
Foo()
{
   i = 0;
}
};

int _tmain(int argc, _TCHAR* argv[])
{
   Foo *f = new Foo(); // このステートメントで [F11] キーはコンストラクタにステップ インする必要があるが、代わりに "ソースがありません。逆アセンブルを表示しますか。" というウィンドウが表示される。
   return 0;
}
```

この問題は、x86 アーキテクチャにだけ当てはまることに注意してください。

この問題を解決するには

コンストラクタにブレークポイントを直接設定し、[F5] キーを押します。 コンストラクタのコードは、x86 デバイス上でデバッグする必要があります。

3.21 wceload.exe のないデバイスまたはエミュレータに CAB ファイルを配置する際に間違ったメッセージが表示される

wceload.exe は、Pocket PC および Smartphone の .cab ファイルのインストール タスクを実行します。
wceload.exe のないデバイスまたはエミュレータに .cab ファイルを配置すると、"指定されたファイルが見つかりません。" というメッセージが表示され、配置は失敗します。

この問題を解決するには

合致する wceload.exe を \Windows ディレクトリのデバイスに手動でコピーします。

3.22 仮想スイッチ ドライバ上の TCP トランスポートを使用して、デバイス エミュレータ上でアプリケーションを配置またはデバッグしているときに、断続的に接続性が失われる

これは、仮想スイッチ ドライバの問題によって発生します。

この問題を解決するには

デバイス エミュレータへの接続性を安定させるには、DMA トランスポートを使用します。

3.23 プロセス境界を越えたステップ実行ができない

再現手順：
1. VC++ -> スマート デバイス -> Win32 スマート デバイス プロジェクトを作成します。
2. プロジェクト作成ウィザードで、SDK として [Pocked PC 2003]、アプリケーションの種類として [Windows アプリケーション] を選択します。
3. プロジェクトをビルドします。
4. WndProc (return 0;) の最後のステートメントにブレークポイントを設定します。
5. [F5] キーを押します。
6. ブレークポイントに到達したら、呼び出し履歴を確認します。
7. [F10] キーを 2 回押して、関数から抜けます。
8. WndProc からステップアウトすると、呼び出し側が別のプロセスにあるため、"ステップできません" というエラーが返されます。

この問題を解決するには

プロセス間にまたがるステップ実行はサポートされていません。最終行では F10 キーではなく F5 キーを押せば、デバッガによる実行が継続され、このエラーの発生を回避できます。

3.24 生成されたデータ フォームの EditViewDialog の ComboBox コントロールで、誤った LookUpBinding 動作が発生する

生成されたデータ フォームの EditViewDialog の ComboBox コントロールで、誤った LookUpBinding 動作が発生します。

たとえば、この動作は、次の手順を実行すると発生する可能性があります。
1. PPC Magneto Device VB Project を作成します。
2. サーバー エクスプローラで、Northwind SDF Database への接続を作成します。
3. [データ ソース] ウィンドウで、Northwind SDF Database のすべてのテーブルを含むデータ ソースを作成します。
4. Orders テーブルをデータ グリッドとしてドラッグ アンド ドロップします。
5. [データ ソース] ウィンドウで、Orders.[Customer ID] のコントロールの種類を comboBox に変更します。
6. DataGrid スマート タスクを使用して、Orders のデータ フォームを作成します。
7. Form1.vb コード ファイル内に、次の関数があります。
   - NewMenuItemMenuItem_Click
   - OrdersDataGrid_Click
   これらの関数に対し、最後のステートメントとして次の行を追加します。
   "OrdersTableAdapter.Update(Me.NorthwindDataSet.Orders)"
8. EditViewDialog フォームを開きます。
9. Customer_IDComboBox プロパティを編集し、次のように設定します。
   DataSource の値：CustomersBindingSource
   ValueMember の値 - Customer ID
   DisplayMember の値 - Company Name
   データ バインディングの値 :: 選択された値：OrdersBindingSource - Customer ID
10. ビルドしてエミュレータに配置します。
11. エミュレータで、Order の SummaryView を開くレコードをクリックします。
12. [編集] をクリックして、Order の EditView を開きます。

期待される結果：
   comboBox には、選択した Order の顧客 ID に対応する会社名が表示されます。

実際の結果：
- comboBox には、会社名の一覧の最初の会社名が表示されます。
- [OK] をクリックすると、最初の会社名に対応する顧客 ID を使用して Order が変更されます。

この問題を解決するには

OrdersEditViewDialog コンストラクタ内で InitializeComponent 関数を呼び出している箇所の直後に、OrdersEditViewDialog_Load 関数の内容を切り取って貼り付けます。

4. ソース コード管理の統合

4.1 FrontPage ソース コード管理下にあるフォルダの名前を変更すると、その下にあるファイルが管理対象ではなくなる

FrontPage のバージョン管理を使用するリモート Web サイトでの作業中に、ソース管理されたアイテムを格納したフォルダの名前を変更した場合、Web サイトを閉じて再度開くまで、そのフォルダ内のすべてのアイテムは管理対象外として表示されます。

この問題を解決するには

フォルダの名前を変更したら Web サイトを閉じ、再度開きます。

4.2 ソース管理から複数レベルの IIS Web サイトを含むソリューションを開いた場合、デバッグするには IIS の読み取りアクセス許可を Web サイト内の親フォルダに追加することが必要な場合がある

ソース管理から IIS Web サイトを含むソリューションを開いた場合、Visual Studio は URL のターゲット パスへの IIS 読み取りアクセスだけを付与します。ただし、IIS Web サイトをデバッグするには、IIS は URL の仮想パスの各フォルダへの読み取りアクセスを必要とします。そのため、複数レベル (例：http://localhost/MyWebs/Web1) の Web サイトを含むソリューションを開いた場合、Web サイトをデバッグするには、親 Web (例：http://localhost/MyWebs) がどのフォルダを参照している場合でも、IIS への読み取りアクセスを付与する必要があります。

この問題を解決するには

Windows エクスプローラで、URL の親フォルダに ASP.NET ユーザー アカウントの "読み取り" アクセス許可を追加します。また、Windows Server 2003 を使用している場合は、NETWORK SERVICE アカウントにもこのフォルダへの読み取りアクセス許可が必要です。たとえば、ソース管理から "http://localhost/MyWebs/Web1" を含むソリューションを開いた場合、Windows エクスプローラで "http://localhost/MyWebs" の参照先ディレクトリに移動し、このフォルダにアクセス許可を追加します。または、Users グループに読み取り許可があるコンピュータ上の別の場所にソリューションを開くこともできます (既定では、My Documents ディレクトリとそのサブフォルダには、Users グループによる読み取りアクセス許可が付与されていません)。

5. Crystal Reports

Visual Studio 2005 の Crystal Reports に関する追加情報については、http://www.businessobjects.com/products/reporting/crystalreports/net/vsnet.asp (英語) を参照してください。

6. Tools for the Microsoft Office System

6.1 Visual Studio Tools for Office は、マルチパートの XML スキーマをサポートしていない

Microsoft Office Excel が主要スキーマとオプションのサポート スキーマで構成できる XmlMaps をサポートしている場合でも、Visual Studio Tools for Office がサポートするのは、単一の主要スキーマに基づく XmlMaps だけです。

この問題を解決するには

既知の解決策はありません。

6.2 WordDynamicControls のサンプルを再度開いたときに、コントロールを読み込まない

WordDynamicControls のサンプルの使用時に、コントロールを追加してから、ドキュメントを閉じて再度開くと、追加したコントロールが保存されていない場合があります。

この問題を解決するには

既知の解決策はありません。

6.3 デザイナでワークブックを開くと、VBA コードと XLM (Excel 4.0 Macro) コードの両方が実行される

ワークブックのイベント ハンドラに Visual Basic for Applications (VBA) コードまたは Excel 4.0 Macro (XLM) コードが含まれている場合、デザイナでワークブックを開くとコードが実行されます。

この問題を解決するには

Visual Studio Tools for Office を使用してワークブックをカスタマイズするときに、VBA コードまたは XLM コードが実行されないようにするには、ワークブックに基づくプロジェクトを作成する前に、ワークブックに関連付けられた VBA コードまたは XLM コードを削除する必要があります。

6.4 Microsoft Office XP だけがインストールされたコンピュータに Visual Studio Team System をインストールすると、Word または Excel の起動時にセキュリティ ダイアログが表示される

Office XP だけがインストールされたコンピュータに Visual Studio Team System をインストールした後、Word または Excel を起動すると、次のような Office セキュリティ ダイアログが表示されます。

VSTOWordAdaptor.dll (または VSTOExcelAdaptor.dll)
以下の作成者によるマクロが含まれています。
Microsoft Corporation

マクロにはウイルスが含まれている可能性があります。マクロを常に無効にすると安全ですが、正当なマクロの場合、一部の機能が失われることがあります。

セキュリティ レベルは高に設定されています。そのため、信頼していないソースのマクロを有効にすることはできません。
[] [この作成者のマクロを常に信頼する]

[マクロを無効にする] [マクロを有効にする] [詳しい説明]

この警告が表示されるのは、VSTOWordAdaptor.dll および VSTOExcelAdaptor.dll の署名に使用される Microsoft 署名が Office XP の信頼された署名の一覧に含まれていないためです。Office 2003 の信頼された署名の一覧には、Microsoft 署名が含まれています。

この問題を解決するには

Microsoft Corporation からのマクロを信頼します。 Visual Studio 2005 Tools for Office を使用することを計画している場合は、サポート対象のバージョンの Office 2003 をインストールする必要があります。Office 2003 をインストールすると、この Microsoft 署名が信頼された署名の一覧に追加されます。

6.5 テーブルへのバインディングが存在するときに、そのテーブルをデータセットから削除すると、アプリケーションのデバッグ時にコンポーネント トレイのコンポーネントが削除される場合がある

この問題は、データ バインド ホスト コントロール (リスト オブジェクトなど) をデザイン サーフェイスから削除し、編集機能またはデータ ソース構成ウィザードを使用して、データ ソースからバインド先のテーブルを削除した場合に発生します。新しいホスト コントロールを追加し、アプリケーションをデバッグした場合、例外が発生することがあります。デバッグを中止すると、デザイン サーフェイスはアンロードされ、コンポーネント トレイからデータ コンポーネントが削除されます。

この問題を解決するには

デザイン サーフェイスからホスト コントロールを削除し、データ ソースからバインド先のテーブルを削除した場合、アプリケーションをビルドする前に、テーブルのバインド ソースを手動で削除する必要があります。

6.6 Windows のユーザー ロケールが英語以外の言語に設定されている場合、Excel 英語版で Windows フォーム コントロールのエクステンダ プロパティが表示されない

この問題は、Windows のユーザー ロケールが英語以外の言語に変更されている場合に、Excel の英語版で発生します。高さや幅などの一部のプロパティは、Excel のデザイン サーフェイス上でホストされた Windows フォーム コントロールで使用できません。

この問題を解決するには

Office Multilingual User Interface (MUI) Pack をシステムにインストールします。Excel のユーザー インターフェイス言語を変更する必要はありません。

6.7 HTTP を使用してドキュメントを開くときに、Windows エクスプローラの [同じウィンドウで開く] がオンになっていない場合、Word は Visual Studio Tools for Office のカスタマイズを読み込まない

[同じウィンドウで開く] プロパティは、ドキュメントをブラウザと別のウィンドウのどちらでホストするかを判断するために、Internet Explorer で使用されます。ドキュメントを別のウィンドウでホストする場合、カスタマイズは読み込まれず、実行されません。

この問題を解決するには

Windows エクスプローラで [同じウィンドウで開く] を選択します。
[同じウィンドウで開く] プロパティの場所を見つけるには、以下の手順を実行します。
1) Windows エクスプローラを開きます。
2) [ツール] メニューをクリックします。
3) [フォルダ オプション] をクリックします。
4) [ファイルの種類] タブをクリックします。
5) [登録されているファイルの種類] の一覧の [DOC] を選択します。
6) [詳細設定] をクリックします。

6.8 Visual Studio Tools for Office ドキュメントをチェックアウトしても、ソース コード管理から必要な関連ファイルがチェックアウトされない

ソース コード管理から Visual Studio Tools for Office ドキュメントをチェックアウトした場合、必要なすべてのファイルが自動的にチェックアウトされるわけではありません。これは、ドキュメントのユーザー インターフェイス (UI: User Interface) コントロールとデータ コンポーネントの各プロパティとは別のファイルに、生成されたコードが格納されるためです。

依存ファイルがチェックアウトされないとファイルを更新できないため、ドキュメントをチェックインしたときに、依存ファイルに間接的に加えられた変更は失われます。

ドキュメントだけをチェックアウトした場合、関連ファイルの更新バージョンがサーバー上に存在しても、最新のプロパティを確認できません。関連ファイルをドキュメントと共にチェックインすると、サーバー上の関連ファイル内のプロパティは上書きされます。

この問題を解決するには

チェックインするドキュメントを変更する前に、すべての関連ファイルを必ずチェックアウトします。ドキュメントをチェックインするときには、すべての関連ファイルを忘れずにチェックインしてください。ソリューション エクスプローラの [すべてのファイルを表示] を選択すると、これらのファイルがドキュメントの下にインデントされて表示されます。

6.9 印刷レイアウト表示ではなく、標準表示で Word テンプレートを開くと、インラインではないコントロールは表示されない

テキストが含まれたインラインではない Word 内のオブジェクトが表示されるのは、印刷レイアウト表示だけです。既定では、Visual

Studio Tools for Office ドキュメントに追加される UI コントロールはテキストを含むインラインです。レイアウトの種類を変更するには、コントロールを右クリックし、[コントロールの書式設定]、[レイアウト] の順にクリックします。

既定では、Visual Studio Tools for Office は標準表示で Word テンプレートを開くため、インラインではないコントロールは表示されません。

この問題を解決するには

標準表示から印刷レイアウト表示に切り替えて、コントロールを表示します。

6.10 デザイン時にアクティブでない Excel シートからコントロールを削除すると、自動生成された分離コードが残される場合がある

各ワークシートには、必要に応じて読み込まれる独自のデザイン サーフェイスがあります。コントロールの追加、削除、変更など、ワークシートに変更を加えると、デザイン サーフェイスによって変更が検出され、生成済みのコードが更新されます。
デザイン時にアクティブでないシートからコントロールを削除した場合、デザイン サーフェイスがインスタンス化されていないため、生成済みのコードは部分的にしか更新できません。これは、少なくとも 2 つのケースで発生する可能性があります。1 つは、[名前付き範囲の管理] ダイアログを使用して、アクティブでないシートから NamedRange コントロールを削除した場合です。もう 1 つは、Excel ドキュメントを Visual Studio 外部で開いているときにコントロールを削除し、その後、Visual Studio 内でそのドキュメントを再度開いた場合です。

この問題を解決するには

[名前付き範囲の管理] ダイアログを使用して NamedRange コントロールを削除する前に、ワークシートをアクティブにします。
ビルド エラーを修復するには、プロパティの編集、サーフェイスへのコントロールの追加、またはサーフェイスからのコントロールの削除を実行することにより、影響を受ける各シートをダーティにします。その後、プロジェクトを再度ビルドします。

6.11 Visual Studio Tools for Office が、64 ビット アプリケーションをネイティブでサポートしていない

既定では、Visual Studio Tools for Office と他の種類のプロジェクトは、"Any CPU" 型に生成されます。ただし、Visual Studio Tools for Office ソリューションが動作するのは 32 ビット環境だけです。

この問題を解決するには

64 ビット コンピュータで Visual Studio Tools for Office ソリューションを実行するには、プロジェクト プロパティのプラットフォーム ターゲットを "Any CPU" から x86 に変更する必要があります。64 ビットの Windows オペレーティング システム上にアプリケーションを配置するにもかかわらず x86 を指定することに注意してください。また、コンソール アプリケーションや Windows フォーム アプリケーションなど、ServerDocument クラスを使用する他の種類のアプリケーションに対してもこの変更を行う必要があります。

6.12 Visual Studio Team System の単体テストが、Visual Studio Tools for Office プロジェクトではサポートされていない

Visual Studio Team System 製品に含まれる単体テスト機能は、Visual Studio Tools for Office ではサポートされていません。

この問題を解決するには

既知の解決策はありません。