

- [sort by category／カテゴリー別](#)
- [sort by date／日付順](#)

## sort by category／カテゴリー別

---

### General／共通

- **[INFO]** VB Migration Partner setup fails if VB6 isn't installed／VB6 がインストールされていない場合、VB Migration Partner のセットアップが失敗します。
- **[INFO]** Two VB.NET project can't coexist in the same directory／2 つの VB.NET プロジェクトが同じフォルダに共存することはできません。
- **[INFO]** Migrated VB.NET code might not work in partially-trusted applications／信頼性が中途半端なアプリケーションでは、変換された VB.NET コードは動作しない可能性があります。
- **[INFO]** Using the VBMigrationPartner\_Support module／VBMigrationPartner\_Support モジュールの使用
- **[INFO]** Multithreaded VB6 applications don't retain their multithreading behavior／マルチスレッドの VB6 アプリケーションはマルチスレッド動作を保持しません。
- **[INFO]** Converted VB.NET applications target 32-bit CPUs by default／デフォルトでは、移行されたアプリケーションは 32bit をターゲットとします。
- **[INFO]** VB Migration Partner generates signed interop assemblies／VB Migration Partner は署名された相互運用アセンブリを生成します。
- **[HOWTO]** Generate Visual Studio 2008 projects／Visual Studio 2008 プロジェクトを生成する
- **[HOWTO]** Show UPGRADE\_ISSUE comments in the Task List window／タスクリストウインドウに UPGRADE\_ISSUE を表示する
- **[HOWTO]** Exclude portions of VB6 code from the migration process and replace it with custom VB.NET statements／変換プロセスから VB6 コードの一部を除外して任意の VB.NET ステートメントに置き換える
- **[HOWTO]** Determine if unsupported members throw an exception at runtime／実行時にサポートされないメンバが例外を発生させた場合に決定する
- **[HOWTO]** Ignore fatal runtime errors / 致命的な実行時エラーを無視する
- **[HOWTO]** Using the NGEN tool to speed up the startup phase of a migrated .NET application／変換されたアプリケーションのスタートアップ処理を高速化する NGEN ツールを使用する
- **[HOWTO]** Using migrated .NET DLLs from VB6, ASP, and other COM clients／VB6、ASP、その他の COM クライアントから .NET Dll に変換する
- **[HOWTO]** Create references to .NET assemblies with absolute path／.NET アセンブリへの参照を絶対パスで作成する
- **[HOWTO]** Create an empty directory under the application's folder／アプリケーションフォルダに空のディレクトリを作成する

- **[HOWTO]** Create a report with all the migration issues and warnings／すべての移行課題と警告のレポートを作成する
- **[HOWTO]** Modify project-level options in VB.NET programs／VB.NET プログラムのプロジェクトレベルオプションを変更する
- **[HOWTO]** Enforce project-level Option Strict Off setting in VB.NET programs／VB.NET プログラムのプロジェクトレベルオプションを強制的に Strict Off 設定にする
- **[PRB]** VB Migration Partner is unable to run Visual Studio／VB Migration Partner が Visual Studio を起動することができない。
- **[HOWTO]** Terminate the migration when a warning message is emitted／警告メッセージが発行されたときに変換を中止する
- **[HOWTO]** Execute multiple instances of VB Migration Partner／VB Migration Partner をマルチインスタンスで実行する方法
- **[HOWTO]** Convert VB6 files that are shared among projects／複数のプロジェクトで共有される VB6 ファイルを変換する方法
- **[HOWTO]** Incrementally migrate a large VB6 project and avoid the “Maximum number of errors has been exceeded” message／巨大な VB6 プロジェクトを段階的に変換して「エラーの最大数を越えた」エラーを回避する
- **[HOWTO]** Skipping one or more methods when applying the PreProcess or PostProcess pragmas／PreProcess プラグマや PostProcess プラグマを適用するときの一つ以上のメソッドをスキップする
- **[HOWTO]** Limit the scope of PreProcess and PostProcess pragmas／PreProcess プラグマや PostProcess プラグマの範囲を制限する
- **[INFO]** VB Migration Partner doesn't support references to external EXE projects／VB Migration Partner は外部の EXE プロジェクトに対する参照をサポートしません。
- **[PRB]** Conversion fails with an "Unable to find ResToResx.exe tool" error／「ResToResx.exe が見つからない」エラーで変換が失敗する
- **[PRB]** Conversion fails with a "Project item name not found" error／「プロジェクトのアイテム名が見つからない」エラーで変換が失敗する
- **[PRB]** A locked EXE or DLL file causes the migration process or compilation command to fail／ロックされた EXE や DLL ファイルが変換プロセスやコンパイルコマンドを失敗させる
- **[PRB]** Errors when doing unit testing or using reflection to explore the support library／単体テストをしているときやサポートライブラリを探して反映させるときにエラーが発生する
- **[HOWTO]** Temporarily disable pragmas for debugging purposes／デバッグのため一時的にプラグマを無効にする
- **[PRB]** PreProcess or PostProcess pragma can make VB Migration Partner to hung／PreProcess プラグマや PostProcess プラグマが VB Migration Partner をハングさせる可能性がある
- **[PRB]** VB.NET applications crash with “Unable to create ActiveX object” error／VB.NET アプリケーションが「ActiveX オブジェクトを作成できない」エラーでクラッシュする
- **[PRB]** Compatibility problems with MS Build／MS Build との互換性の問題
- **[PRB]** VB Migration Partner may stop when converting VB6 apps that reference a .NET DLL／.NET DLL を参照する VB6 アプリケーションを変換するとき VB Migration Partner が停止する可能性がある
- **[PRB]** Out of memory exceptions when converting very large projects／非常に大きなプロジェクトを変換したときにメモリ不足例外が発生する

- [\[PRB\] VB Migration Partner fails to recognize a reference to a VB6 project](#) / VB Migration Partner が VB6 プロジェクトに対する参照の認識に失敗する
- [\[PRB\] VB Migration Partner generates incomplete classes](#) / VB Migration Partner が不完全なクラスを生成する
- [\[HOWTO\] Modify migrated projects to use CodeArchitects.VBPowerPack.dll](#) / [CodeArchitects.VBPowerPack.dll を使用するために変換されたプロジェクトを変更する](#)

## Language / 言語

- [\[PRB\] GoSub statements inside For loops and With blocks cause a compilation error](#) / For Loop と With ブロック内部の GoSub 文がコンパイルエラーを起こす
- [\[PRB\] Structures with same name may be generated in the same VB.NET namespace](#) / 同じ名前の構造体が VB.NET の同じ名前空間に生成されるかもしれない
- [\[PRB\] Limitations in the conversion of #If blocks](#) / #If ブロックの変換上の制限
- [\[INFO\] VB.NET property values passed to a ByRef parameter can be modified](#) / ByRef パラメータに渡される VB.NET プロパティの値は変更できる
- [\[INFO\] The End keyword is converted as ApplicationExit6](#) / END キーワードは ApplicationExit6 に変換される
- [\[INFO\] VB.NET floating point divisions never raise Division-by-zero errors](#) / VB.NET の浮動小数点の割り算は 0 で割ってもエラーにならない
- [\[PRB\] Math operations on Variants cause an Overflow exception](#) / バリエーションの数値演算がオーバーフロー例外を起こす
- [\[PRB\] VB.NET collections can't be modified while inside a For Each loop](#) / VB.NET のコレクションは For Each Loop 内部では変更することができない。
- [\[PRB\] VB.NET arrays are copied using "shallow" semantics](#) / VB.NET の配列は「shallow」の動作を使用して複製される
- [\[HOWTO\] Avoid Option Explicit Off in generated VB.NET files](#) / 生成された VB.NET ファイルの Option Explicit Off を避ける
- [\[INFO\] UDT fields accessed via a VB6Variant variable are readonly](#) / VB6Variant 変数経由でアクセスされるユーザー定義型は読み取り専用です。
- [\[INFO\] How VB Migration Partner deals with late-bound calls](#) / VB Migration Partner が遅延バインディング呼び出しを取り扱う方法
- [\[HOWTO\] Improve code generation with Variant variables and parameters](#) / [Variant 変数とパラメータを伴うコード生成を改善する](#)
- [\[HOWTO\] Replace Collection with the VB6CollectionVariant type](#) / VB6CollectionVariant 型でコレクションを置換する
- [\[PRB\] Late-bound code might throw an exception or not work as expected](#) / 遅延バインドコードが例外を起こしたり予想通りに動作しないかもしれない
- [\[HOWTO\] Reduce runtime errors caused by null strings](#) / NULL 文字による実行時エラーを減らす
- [\[HOWTO\] Reduce compilation errors caused by late binding or undetected conversions](#) / 遅延バインドや検知されない移行によるコンパイルエラーを減らす
- [\[HOWTO\] Deal with Null values in If statements](#) / If 文で NULL を取り扱う

- **[HOWTO]** Speed up string concatenation／文字連結を高速化する
- **[PRB]** Changing the type of a variable causes a Type Mismatch compilation error／変数の型を変更すると型不一致エラーが発生する
- **[HOWTO]** Solve issues related to Null values in database fields／データベース列の NULL 値に関わる問題を解決する
- **[HOWTO]** Modify the behavior of methods and classes in VB.NET library and VB Migration Partner support library／VB.NET ライブラリと VB Migration Partner サポートライブラリのメソッドとクラスの動作を変更する
- **[INFO]** VB.NET doesn't support methods with same name as events／VB.NET はイベントと同じ名前のメソッドをサポートしない
- **[HOWTO]** Handle compilation errors caused by name collisions／名前の衝突によるコンパイルエラーを取り扱う
- **[PRB]** Compilation error when implementing an interface that defines a collection class／コレクションクラスを定義するインターフェースを実装するときコンパイルエラーが発生する
- **[PRB]** Pragmas related to arrays might not work if applied to a ReDim statement／ReDim ステートメントに適用される場合、配列に関するプラグマが動作しないかもしれない
- **[HOWTO]** Apply ByVal keyword to unassigned by-reference parameters／署名されていない参照渡しのパラメータに ByVal キーワードを適用する
- **[HOWTO]** Implement functions that return arrays with nonzero LBound／0 でない配列を返すファンクションを実装する
- **[HOWTO]** Assign a specific data type when the As clause is missing／As 節がないとき、特定のデータ型を割り当てる
- **[HOWTO]** Convert byte-oriented string functions／バイト指向型文字列ファンクションを変換する
- **[PRB]** Default member isn't resolved for arguments of IIF and Choose methods／デフォルトメンバが IIF の引数と Choose メソッドに対して解決されない
- **[HOWTO]** Move the declaration of a variable into a For or For Each loop／For や For Each Loop の中に変数の宣言を移動する
- **[HOWTO]** Transform "Not x Is y" expressions into "x IsNot y" expressions／「Not x Is y」式を「x IsNot y」式に変換する
- **[HOWTO]** Replace App6 properties with native VB.NET members／App6 プロパティをネイティブの VB.NET のメンバで置き換える
- **[HOWTO]** Get rid of warnings related to Screen.MousePointer property／Screen.MousePointer プロパティに関する警告を取り除く
- **[PRB]** The VB.NET application doesn't compile because of a "Method XYZ not found" compilation error／「メソッド XYZ が見つからない」というコンパイルエラーのため VB.NET アプリケーションがコンパイルできない
- **[HOWTO]** Apply pragmas only to variables of specific types／特定の型の変数にだけプラグマを適用する
- **[HOWTO]** Using the SetType pragma with optional parameters／オプションパラメータを伴う SetType プラグマを使用する
- **[HOWTO]** Convert While loops into Do loops／While loop を Do loop に変換する
- **[HOWTO]** Convert weak object references and the ObjPtr keyword to VB.NET／曖昧なオブジェクト参照と ObjPtr キーワードを VB.NET に変換する

- **[HOWTO]** Generate auto-implemented properties for Visual Basic 2010 / Visual Basic 2010 用の自動実装プロパティを生成する
- **[HOWTO]** Optimize Boolean expressions using PostProcess pragmas / PostProcess プラグマを使ってブーリアン式を最適化する
- **[PRB]** Compilation errors with classes or usercontrols that expose events with same name as other members and that are used to define an interface / 他のメンバと同じ名前のイベントを公開するクラスやユーザーコントロール、またはインターフェースを定義するために使用されているクラスやユーザーコントロールでコンパイルエラーになる
- **[HOWTO]** Fix ReDim statements that change the rank of an array / 配列の rank を変更する ReDim ステートメントを修正する
- **[PRB]** VB6 and VB.NET store arrays into Collection objects in different ways / VB6 と VB.NET が異なる方法で配列をコレクションに格納する
- **[PRB]** The VB.NET application doesn't compile because of a "Number of indices exceeds the number of dimensions of the indexed array" compilation error / 「索引数が索引化された配列の次元数を超過しました」というコンパイルエラーのため VB.NET アプリケーションがコンパイルできない
- **[PRB]** The migrated VB.NET application can't compile because of the error "'XYZ' is already declared in this enum" / 「XYZ はすでにこの列挙型に宣言されています」というエラーのため変換された VB.NET アプリケーションがコンパイルできない
- **[INFO]** The Erase statement doesn't work well with VB6 static arrays / Erase ステートメントは VB6 の静的配列と相性が悪い
- **[PRB]** Arrays held in Variant and Object variables are converted in a special way / バリエントに保持されている配列とオブジェクト変数は特別な方法で変換される
- **[PRB]** The IsArray method returns False with uninitialized array stored in Object variables / オブジェクト変数に格納された初期化されていない配列を伴う IsArray メソッドは False を返す
- **[INFO]** Default methods with arguments are converted to read-only properties / 引数を伴うデフォルトメソッドは読み取り専用プロパティに変換される
- **[PRB]** The VB.NET application can't compile because of an "'AddressOf' expression cannot be converted to 'Integer' because 'Integer' is not a delegate type" error / 「Integer はデリゲート型ではない」というエラーのために AddressOf 式を Integer に変換できないため、VB.NET アプリケーションがコンパイルできない
- **[PRB]** The VB.NET application can't compile because a constant in a Structure is ambiguous between two modules / 構造体の定数がふたつのモジュールの間で曖昧なので VB.NET アプリケーションがコンパイルできない
- **[PRB]** VB6 applications that use window subclassing or other API callback methods throw a CallbackOnCollectedDelegate excetion / サブクラスウインドウや他の API コールバックメソッドを使う VB6 アプリケーションが CallbackOnCollectedDelegate 例外を起こす
- **[PRB]** A call to a Windows API Declare returns a string filled with spaces / Windows API の呼び出しがスペースのみの文字列を返す
- **[PRB]** Calls to GetOpenFileName API method returns trimmed string / GetOpenFileName API メソッドの呼び出しが切り取られた文字列を返す

- **[PRB]** A FileGet6 or FilePut6 method throws an exception while attempting to read or write a User-Defined Type variable / ユーザー定義型変数の読み書きをしようとすると FileGet6 や FilePut6 というメソッドが例外を起こす
- **[PRB]** Most VB.NET methods don't work well with strings that contain ASCII 0 characters / ほとんどの VB.NET のメソッドが ASCII 0 の文字を含む文字列との相性が悪い
- **[PRB]** The FileLineInput6 statement uses CR-LF pairs as delimiters / FileLineInput6 は区切り文字として CRLF を使用する
- **[PRB]** The vbDefaultButton4 constant might not be converted correctly / vbDefaultButton4 定数は正しく変換されない可能性がある
- **[PRB]** VB.NET doesn't support implicit conversions in Const statements / VB.NET は Const の暗黙的な変換をサポートしていない
- **[PRB]** Using an array of UDTs causes a compilation error or doesn't work correctly at runtime / ユーザー定義型の配列を使用するとコンパイルエラーを引き起こすか、実行時に正常に動作しなくなる
- **[PRB]** Passing an enumerated value to a Boolean argument throws an exception when the method returns / メソッドが値を戻すときに列挙型からブーリアン引数に値が渡されるとエラーが発生する
- **[PRB]** Get# and Put# statements might not behave as expected / Get# と Put# は想定取りの動作を行わない可能性がある
- **[PRB]** Structures can't have the same name as other members in VB.NET / VB.NET では構造体は他のメンバと同じ名称を持つことができない
- **[PRB]** Structures containing DBCS strings may cause exceptions when passed to an external (Declare) method / 外部の (Declare) メソッドに渡された場合、2 バイト文字を含む構造体はエラーの原因となる可能性がある
- **[PRB]** The Len6 method can return invalid values when applied to a structure containing fixed-length strings / 固定長文字列を含む構造体に使用された場合、Len6 メソッドは無効な値を戻すことがある
- **[PRB]** The migrated application takes too long at loading its first form / 変換後のアプリケーションが最初のフォームを呼び出すとき時間がかかりすぎる
- **[PRB]** The system hangs when using SendKeys to simulate key presses in an external application / 外部アプリケーションのキープレスをシミュレートする SendKeys を使用する場合、システムがハングアップする
- **[INFO]** Declare statements can generate methods with different names / Declare 文は異なる名称のメソッドを生成することがある
- **[HOWTO]** Migrate VB6 applications that use window subclassing / サブクラスウインドウを使用する VB6 アプリケーションを変換する
- **[INFO]** OLE\_COLOR type is converted to System.Drawing.Color / OLE\_COLOR は System.Drawing.Color に変換される
- **[INFO]** DIB images are copied and pasted as regular images / DIB イメージは標準イメージとしてコピー&ペーストされる
- **[PRB]** Output from the Print statement appears slightly different from the original VB6 application / Print 文の出力結果が元の VB6 アプリケーションとは若干異なっているように見える
- **[BUG]** The VB6Printer object doesn't print in landscape mode / VB6Printer オブジェクトがランドスケープモードで印刷しない
- **[PRB]** The Line property of output TextStream always returns 1 / TextStream 出力の Line プロパティが常に 1 を返してしまう

- **[INFO]** Unsupported ActiveX controls / サポートされていない ActiveX コントロール
- **[INFO]** [Controls in the support library can't be dropped on a standard VB.NET form](#) / サポートライブラリのコントロールは標準の VB.NET フォームに置くことができない
- **[PRB]** Control events don't fire / コントロールのイベントが動作しない
- **[PRB]** A migrated form doesn't fire the Paint event / 変換後のフォームで Paint イベントが発生しない
- **[HOWTO]** [Use controls from VB Migration Partner's support library on a standard .NET form](#) / 標準の .NET フォーム上で VB Migration Partner のサポートライブラリのコントロールを使用する
- **[HOWTO]** Reduce flickering when loading or resizing a form with many controls / 多数のコントロールを持つフォームがロードされたり、リサイズされるときにチラツキを抑える
- **[INFO]** How VB Migration Partner implements drag-and-drop features / VB Migration Partner がドラッグアンドドロップ機能を実装する方法
- **[HOWTO]** Reduce flickering in graphic operations / 画像処理のチラツキを抑える
- **[HOWTO]** Permanently display label and button accelerators / ラベルとボタンのアクセラレータを恒久的に表示する
- **[PRB]** The VB.NET application ends with a fatal exception when the splash screen closes / スプラッシュスクリーンを閉じたときに VB.NET アプリケーションが致命的エラーで終了してしまう
- **[HOWTO]** Define a more reliable Ambient.UserMode property / より信頼性の高い Ambient.UserMode プロパティを定義する
- **[PRB]** ObjectDisposed exception where re-opening a closed form / 閉じたフォームを再オープンするときの ObjectDisposed エラー
- **[PRB]** Form-level variables in default forms don't retain their values between calls / デフォルトフォームのフォームレベル変数が呼び出しと呼び出しの間で保持されていない
- **[PRB]** The VB.NET application hangs on the splash screen or terminates earlier than expected / スプラッシュスクリーンで VB.NET アプリケーションがハングアップしたり、予想よりも早く終了してしまう
- **[HOWTO]** Determine the font used for forms and controls / フォームとコントロールで使用されるフォントを決める
- **[PRB]** A reference to the Controls collection isn't converted to Controls6 / コントロールコレクションに対する参照が Controls6 へ変換されていない
- **[PRB]** Setting the Font property at runtime causes an unexpected Paint event which may freeze the application / 実行時のフォントプロパティの設定がアプリケーションをフリーズさせる可能性のある予期せぬ Paint イベントの原因となってしまう。
- **[PRB]** Setting the Font property inside a Paint event causes an endless loop / Paint イベント内部のフォントプロパティの設定が終わりなきループ処理の原因となってしまう
- **[PRB]** A nonexisting property is generated for a control in the \*.Designer.vb file / \*.Designer.vb ファイルのコントロールに対して存在しないプロパティが生成されてしまう
- **[PRB]** VB.NET controls are taller than the original VB6 control / VB.NET コントロールの高さが元の VB6 コントロールよりも大きくなってしまう。
- **[PRB]** A form loaded with Load method might cause errors in its Unload event handler / Load メソッドでロードされたフォームが自身の Unload イベントハンドラーのエラーの原因になってしまうかもしれない

- **[HOWTO]** Speed up For Each loops that iterate over the Controls collection / コントロールコレクションに対して繰り返し用いられる For Each ループを高速化する
- **[HOWTO]** Iterate over all forms of current application / アプリケーションのフォームすべてに対して繰り返し操作を行う
- **[PRB]** ScaleHeight and ScaleWidth properties aren't correctly serialized at design-time / ScaleHeight プロパティと ScaleWidth プロパティがデザイナー上では正しい順番で並ばない
- **[HOWTO]** Change the value of a given property of all controls of specified type or name / 特定の型や名前のコントロールのプロパティの値を変更する
- **[PRB]** Multiple or missing Paint events / 多重 Paint イベントと Paint イベントの喪失
- **[PRB]** The CLS method fails to refresh child controls on the form / CLS メソッドがフォーム上のコントロールのリフレッシュに失敗してしまう
- **[PRB]** Drag-and-drop operations inside a ListBox control may deliver fake results / ListBox コントロール内でのドラッグ&ドロップ操作が見せかけの動作をする可能性がある
- **[PRB]** Elements of control arrays can't be referenced before being created / コントロール配列の要素が作成される前に参照されない可能性がある
- **[PRB]** PictureBox, Frame and other container controls don't correctly process the Tab key if TabStop=False / PictureBox コントロールや、Frame コントロール、そしてその他のコンテナコントロールが TabStop=False の場合に Tab キーを正常に処理しない
- **[PRB]** Forms with same name as one of their controls can't be opened in Visual Studio designer / フォーム上のコントロールと同じ名称を持つフォームを Visual Studio のデザイナーで表示できない
- **[INFO]** Arrays of menu controls can't contain separators / メニューコントロールは分離記号を含められない
- **[INFO]** All ListBox and Frame controls in an array must have same style / 配列内の ListBox コントロールと Frame コントロールは同じスタイルでなければならない
- **[PRB]** Can't mix VB graphic properties and GDI native calls / VB の画像プロパティと GDI 固有の呼び出しは混在させることができない
- **[PRB]** NullReference exception in Form\_Initialize / Form\_Initialize における NullReference エラー
- **[PRB]** A form behaves differently the second time it is opened / 2 回目に表示する際にフォームが異なる動作をしてしまう
- **[PRB]** The Load event of a form doesn't fire when you access its property or control from another form / フォームのプロパティや他のフォームのコントロールにアクセスするときにフォームの Load イベントが動作しない
- **[PRB]** Graphic output is cleared when the form becomes visible / フォームが可視化されたときに画像出力がクリアされてしまう
- **[INFO]** Negative values for ScaleWidth and ScaleHeight properties cause incorrect output in Microsoft Vista / Windows Vista では ScaleWidth プロパティと ScaleHeight プロパティに設定した負の値が異常な出力の原因となってしまう
- **[PRB]** Closing a form causes the ObjectDisposed exception / フォームを閉じると ObjectDisposed エラーが発生してしまう
- **[PRB]** An MDI Child form can mistakenly display a menu bar / MDI 子フォームがメニューバーを誤って表示してしまう可能性がある



- **[PRB]** Menus on MDI child forms doesn't replace the toplevel menu of the MDI parent form / MDI 子フォーム上のメニューが MDI 親フォームトップレベルのメニューを置き換えない
- **[PRB]** VB.NET forms don't display the size grip handle / VB.NET フォームがサイズグリップを表示しない
- **[PRB]** MDI child forms don't become visible / MDI 子フォームが可視化しない
- **[PRB]** Assignments to Font properties generate runtime errors in late-bound mode / 遅延バインディングではフォントプロパティの代入が実行時エラーを発生させる
- **[PRB]** The font of a control changes unexpectedly at runtime / コントロールのフォントが実行時に勝手に変わってしまう
- **[PRB]** Arrays of menu items can throw an exception if items are destroyed too early / メニューアイテムの破棄が早すぎると、配列がエラーを起こしてしまう
- **[PRB]** Invoking the Show method of MDI child forms containing one or more ActiveX controls might not fire the Load event / ActiveX コントロールを含む MDI 子フォーム Show メソッドを呼び出しても Load イベントが発生しない可能性がある
- **[PRB]** ActiveX controls aren't migrated correctly / ActiveX コントロールが正常に変換されない
- **[PRB]** Assignments to VB6 Form's Left and Top properties are ignored / VB6 Form の Left プロパティと Top プロパティへの代入が無視される
- **[PRB]** Using the hDC property causes an InvalidOperationException error / hDC プロパティを使用すると InvalidOperationException エラーが発生してしまう
- **[PRB]** Setting the TabIndex property in VB.NET doesn't affect the TabIndex of other controls / VB.NET では TabIndex プロパティを設定しても他のコントロールの TabIndex が変更されない
- **[PRB]** Controls in migrated VB.NET forms may be hidden by other controls / 変換後の VB.NET フォーム上のコントロールが他のコントロールによって隠れてしまう可能性がある
- **[PRB]** Disabled controls and read-only TextBox controls appears with grayed background / 無効なコントロールと読み取り専用の TextBox コントロールの背景が灰色となってしまう
- **[PRB]** Selecting the contents of a TextBox when the control gets the input focus doesn't work in VB.NET / コントロールが入力状態となったとき、TextBox の内容の選択が VB.NET では行われない
- **[PRB]** Setting the Visible=False for a Frame may cause spurious GotFocus, LostFocus, and Validate events / Frame に対する Visible=False 設定によって GotFocus イベントや LostFocus イベントや Validate イベントが発生してしまう
- **[PRB]** Shape and Line controls aren't updated correctly / Shape コントロールと Line コントロールが正常に更新されない
- **[INFO]** Forms with Line and Shape controls may exhibit a lot of flickering / Line コントロールと Shape コントロールを持つフォームに多数のチラツキが見られる可能性がある
- **[PRB]** PictureBox controls can be cleared unexpectedly or receive spurious Paint events / PictureBox コントロールが勝手にクリアされたり、誤った Paint イベントを受け取ってしまう可能性がある
- **[PRB]** The MsgBox, InputBox, and common dialogs cause spurious LostFocus and GotFocus events / MsgBox や InputBox やコモンダイアログが誤った LostFocus イベントや GotFocus イベントの原因となってしまう
- **[HOWTO]** Convert from OLE types to .NET types / OLE 型から .NET 型へ変換する
- **[HOWTO]** Leverage the features of native .NET controls / .NET コントロール固有の機能を利用する
- **[HOWTO]** [Leverage .NET TextAlign property with command, checkbox, and option button controls / Command や CheckBox や OptionButton に .NET の TextAlign プロパティを利用する](#)

- [\[INFO\] Minor differences between CommonDialog, Printer, Line, and Shape VB6 classes and their implementation in CodeArchitects.VBPowerPack.dll/VB6 のコモンダイアログクラスや Printer クラスや Line クラスや Shape クラスと CodeArchitects.VBPowerPack.dll に実装されたクラスの違い](#)
- [\[INFO\] VB Migration Partner uses the Line, Shape, and Printer classes in Microsoft VB PowerPacks/VB Migration Partner は Microsoft VB PowerPacks の Line クラスと Shape クラスと Printer クラスを使用する](#)
- [\[HOWTO\] Speed up execution by removing unnecessary Refresh methods/不要な Refresh メソッドを削除して高速化する](#)
- [\[INFO\] VB.NET automatically sets the Checked property of the first control in a group of Option buttons/VB.NET はオプションボタングループの最初のコントロールの Checked プロパティを自動的に設定する](#)
- [\[PRB\] Image controls whose Picture property is Nothing aren't transparent/Picture プロパティがない Image コントロールは透明ではない](#)
- [\[PRB\] Transparent VB6 Label controls aren't transparent in the VB.NET application/透明な VB6 ラベルコントロールは VB.NET アプリケーションでは透明ではない](#)
- [\[PRB\] Mouse event aren't received by transparent Image and Label controls/透明な Image コントロールと Label コントロールは Mouse イベントを受け取らない](#)
- [\[PRB\] Label and Image controls appear in front of other controls/Label コントロールと Image コントロールは他のコントロールの前に表示される](#)
- [\[INFO\] DoEvents6 calls can cause weird user interface behaviors/DoEvents6 はユーザーインターフェースの動きを奇妙なものにすることがある](#)
- [\[INFO\] Minor differences between VB6 and VB.NET implementation of DDE/DDE の実装における VB6 と VB.NET との間の相違](#)
- [\[INFO\] Timers are disabled inside VB6 IDE when a MsgBox or InputBox statement is active/MsgBox 文や InputBox 文がアクティブなとき VB6 の IDE ではタイマーは無効である](#)
- [\[INFO\] Double-clicking on a control array element at design-time creates a brand new event handler/デザイナー上のコントロール配列の要素をダブルクリックすると新しいイベントハンドラが作成される](#)
- [\[PRB\] Replacing a VB6Placeholder control with another control causes an error in the VB.NET designer/VB.NET のデザイナー上で VB6Placeholder コントロールを他のコントロールに置き換えるとエラーが発生する](#)
- [\[PRB\] The VB6 form has border, but the migrated VB.NET form hasn't/VB6 フォームは境界線を持っているが、変換された VB.NET フォームは持っていない](#)
- [\[PRB\] Scrollbars in MDI forms are always visible/MDI フォームのスクロールバーは常に可視化されている](#)
- [\[PRB\] "Classic" \(VB3-style\) drag-and-drop stops working/「古い」\(VB3 スタイルの\) ドラッグ&ドロップは動作を停止させる](#)
- [\[INFO\] Minor differences among different versions of Microsoft Windows Common Controls library/Microsoft Windows コモンコントロールライブラリのバージョン間の相違](#)
- [\[INFO\] User-defined coordinate systems can have unexpected results/ユーザーによってコーディネートされたシステムは予期しない結果をもたらすことがある](#)
- [\[INFO\] Controlling the LostFocus and Validate event sequence/LostFocus イベントと Validate イベントの順番を制御する](#)
- [\[INFO\] Controlling the GotFocus and MouseDown event sequence/GotFocus イベントと MouseDown イベントの順番を制御する](#)

- **[INFO]** Minor differences between VB6 and VB.NET TreeView controls/VB6 と .NET の TreeView コントロールの相違
- **[INFO]** HitTest method of the TreeView control can return different values in VB.NET/TreeView コントロールの HitTest メソッドは VB.NET では異なる値を返す可能性がある
- **[PRB]** Images displayed in TreeView, ListView, and other controls aren't transparent/TreeView コントロールや ListView コントロールなどで表示されるイメージは透明ではない
- **[INFO]** Minor differences between VB6 and VB.NET StatusBar controls/VB6 と VB.NET の StatusBar コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET ImageCombo controls/VB6 と VB.NET の ImageCombo コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET DTPicker controls/VB6 と VB.NET の DTPicker コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET MonthView controls/VB6 と VB.NET の MonthView コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET UpDown controls/VB6 と VB.NET の UpDown コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET SSTab controls/VB6 と VB.NET の SSTab コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET ListView controls/VB6 と VB.NET の ListView コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET Toolbar controls/VB6 と VB.NET の Toolbar コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET MSChart controls/VB6 と VB.NET の MSChart コントロールの相違
- **[INFO]** [How to migrate the PictureDn, PictureUp, and PictureDisabled properties of SSCommand and SSRibbon controls/SSCommand コントロールと SSRibbon コントロールの PictureDn プロパティと PictureUp プロパティと PictureDisabled プロパティを変換する方法](#)
- **[PRB]** Top property of controls hosted inside an SSTab control is relative to TabPage border/SSTab コントロール内部に配置されたコントロールの Yop プロパティが TabPage の境界と関連付けられてしまう
- **[PRB]** [Forms containing an ImageList control don't display at design-time after updating the CodeArchitects.VBLibrary DLL./CodeArchitects.VBLibrary DLL を更新した後、ImageList コントロールを含むフォームがデザイナーに表示されない](#)
- **[PRB]** Setting the current tab in a SSTab control brings the input focus to the control/SSTab コントロールのカレントタブの設定によってコントロールが入力可能状態になってしまう
- **[PRB]** Hotkeys in SSTab caption work only if the form's KeyPreview property is set to True/フォームの KeyPreview プロパティが True の場合にしか SSTab の見出しのホットキーが動作しない
- **[INFO]** Minor differences between VB6 and VB.NET ImageList controls/VB6 と VB.NET の ImageList コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET Command controls/VB6 と VB.NET の Command コントロールの相違
- **[INFO]** Binary properties of a few controls aren't migrated correctly/2、3のコントロールのバイナリープロパティは正常に変換されない

- [\[PRB\] Assignments to the ListView.SelectedItem property are ignored if the control isn't visible](#) / コントロールが非表示の場合、`ListView.SelectedItem` プロパティに対する代入は無視されてしまう
- [\[INFO\] Minor differences between VB6 and VB.NET DriveListBox, DirListBox, and FileListBox controls](#) / VB6 と VB.NET の DriveListBox コントロールと DirListBox コントロールと FileListBox コントロールの相違
- [\[PRB\] The ListView's background image doesn't migrate correctly](#) / ListView の背景イメージは正常に変換されない
- [\[PRB\] Deleting a Toolbar, ListView, or StatusBar control prevents the form from loading in Visual Studio designer](#) / Toolbar コントロールや ListView コントロールや StatusBar コントロールを削除するとフォームを Visual Studio のデザイナーに表示することができなくなる
- [\[INFO\] The .NET Toolbar control can't contain standard controls](#) / .NET の Toolbar コントロールは標準コントロールを含めることができない
- [\[HOWTO\] Move a control over a toolbar at runtime](#) / 実行時にツールバー上にコントロールを移動する
- [\[PRB\] The UpDown control can affect the wrong property of its buddy control](#) / UpDown コントロールは対応するコントロールのプロパティに誤った値をセットすることがある
- [\[PRB\] Removing an item from a multi-selectable ListBox resets ListIndex property](#) / 複数選択可能なリストボックスからアイテムを削除すると ListIndex プロパティがリセットされてしまう
- [\[HOWTO\] Check an item of a ListBox with Style=CheckBoxes with a single mouse click](#) / Style=CheckBoxes のリストボックスのアイテムをシングルクリックで確認する
- [\[PRB\] Removing an item from a single-selectable ListBox selects the previous item](#) / 単一選択可能なリストボックスからアイテムを削除するとひとつ前のアイテムが選択されてしまう
- [\[PRB\] SmallChange property can't be higher than LargeChange](#) / SmallChange プロパティが LargeChange より大きくならない
- [\[PRB\] ScrollBar controls ignore the LargeChange property](#) / ScrollBar コントロールが LargeChange プロパティを無視してしまう
- [\[PRB\] DTPicker controls display their content using the LongDate format](#) / DTPicker コントロールがその内容を LongDate 書式で表示してしまう
- [\[PRB\] The DTPicker control doesn't accept dates earlier than 1/1/1753](#) / DTPicker コントロールが 1753/1/1 より前の日付を受け付けない
- [\[INFO\] Minor differences between VB6 and VB.NET TabStrip controls](#) / VB6 と VB.NET の TabStrip コントロールの相違
- [\[INFO\] Minor differences between VB6 and VB.NET RichTextBox controls](#) / VB6 と VB.NET の RichTextBox コントロールの相違
- [\[PRB\] SelFontName, SelFontSize, SelBold, SelItalic, SelUnderline, and SelStrikeThru properties of RichTextBox control return Nothing instead of Null](#) / RichTextBox コントロールの SelFontName プロパティと SelFontSize プロパティと SelBold プロパティと SelItalic プロパティと SelUnderline プロパティと SelStrikeThru プロパティが Null ではなく Nothing を返してしまう。
- [\[PRB\] Invalid assignments to the Rtf property might not throw an exception](#) / Rtf プロパティに無効な代入を行ってもエラーが発生しない可能性がある
- [\[INFO\] Fields are restored in a different way if an error occurs with a Data control](#) / データコントロールにエラーが発生した場合、フィールドが異なる方法で回復される

- **[INFO]** Minor differences between VB6 and VB.NET DataList controls/VB6 と VB.NET の DataList コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET DataCombo controls/VB6 と VB.NET の DataCombo コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET RemoteData controls/VB6 と VB.NET の RemoteData コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET DataGrid controls/VB6 と VB.NET の DataGrid コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET controls belonging to the Threed (ActiveTreedPlus) control library/VB6 と VB.NET の Threed (ActiveTreedPlus) に属するコントロールの相違
- **[HOWTO]** Migrating VB6 reports to NET/VB6 のレポートを .NET に変換する
- **[PRB]** WLOption controls ignore the Group property/WLOption コントロールは Group プロパティを無視してしまう
- **[INFO]** The cdIOFNShareAware flag of the CommonDialog control isn't supported/コモンダイアログコントロールの cdIOFNShareAware フラグはサポートされない
- **[INFO]** TabStrip and SSTab controls don't display hotkeys in button captions/TabStrip コントロールと SSTab コントロールはボタンの見出しにホットキーを表示しない
- **[BUG]** The Scroll event of the VB6FlatScrollBar control doesn't fire repeatedly while dragging the indicator/マウスカーソルをドラッグしても VB6FlatScrollBar コントロールの Scroll イベントが連続的に発生しない
- **[BUG]** The MouseIcon isn't migrated for MMControl and MSHFlexGrid controls/MMControl コントロールと MSHFlexGrid コントロールについては MouseIcon は変換されない
- **[PRB]** Properties of the MSChart control might not be migrated correctly/MSChart コントロールのプロパティは正常に変換されない可能性がある
- **[INFO]** GotFocus and LostFocus events in UserControl/VB6 ユーザーコントロールの GotFocus イベントと LostFocus イベント
- **[PRB]** Serializable properties in UserControl using system objects may crash Visual Studio/システムオブジェクトを使用しているユーザーコントロールの直列化可能なプロパティのせいで Visual Studio を強制終了する可能性がある
- **[HOWTO]** Simplify code produced for UserControl classes/VB6 ユーザーコントロールクラス用に生成されたコードを簡素化する
- **[PRB]** Property references in a form might not match property names in a UserControl/VB6 フォーム上で参照を行うプロパティがユーザーコントロールのプロパティ名と一致しないことがある
- **[PRB]** Errors when re-opening modal, default forms that host an ActiveX control/VB6 ActiveX コントロールを含むモーダルのデフォルトフォームを再オープンするときのエラー
- **[PRB]** The Default and Cancel properties of a VB6UserControl object don't behave like in VB6/VB6 ユーザーコントロールオブジェクトの Default プロパティと Cancel プロパティが VB6 のように動作しない
- **[PRB]** Stand-alone UserControl, UserDocument, and PropertyPage keywords might not be converted correctly/VB6 独立した UserControl や UserDocument や PropertyPage といったキーワードは正常に変換されないことがある
- **[PRB]** Private UserControls aren't loaded with Controls.Add method/VB6 プライベート型のユーザーコントロールは Controls.Add method に読み込まれない

- **[PRB]** The application stops with the “ActiveX control is windowless. Windowless ActiveX controls aren't supported” error message／「ActiveX コントロールはウインドウを持っていません。ウインドウを持たない ActiveX コントロールはサポートされていません。」というエラーでアプリケーションが終了してしまう
- **[PRB]** Common problems when generating an ActiveX wrapper with AxWrapperGen／AxWrapperGen を使用して ActiveX のラッパーを生成したときの一般的な問題
- **[PRB]** UserControl classes don't raise the InitProperties, ReadProperties and WriteProperties events／ユーザーコントロールクラスで InitProperties イベントや ReadProperties イベントや WriteProperties イベントが起きない
- **[PRB]** Can't use Tab key to move the focus to a DataGrid ActiveX control／DataGrid ActiveX コントロールにフォーカス移動させるためにタブキーを使用することができない
- **[PRB]** Mouse custom cursor in drag-and-drop operations with ActiveX controls doesn't work correctly／ActiveX コントロールでのドラッグ&ドロップ操作においてマウスのカスタムカーソルは正常に動作しない
- **[HOWTO]** Write an extender DLL that can manipulate design-time properties of converted controls／変換されたコントロールのデザイン用プロパティを操作することができる拡張 DLL を記述する
- **[HOWTO]** Retrieving a reference to the actual ActiveX control wrapped in a class generated with AxWrapperGen／AxWrapperGen で生成されたクラスでラップされた ActiveX コントロールに対する参照を検索する
- **[INFO]** You need a runtime license for RDO and for ActiveX controls when running on a computer on which VB6 isn't installed／VB6 がインストールされていない PC で実行するには RDO と ActiveX コントロール用のランタイムライセンスが必要となる
- **[PRB]** The list area of a DataCombo or DataList control isn't automatically updated／DataCombo コントロールや DataList コントロールのリスト領域は自動的に更新されない

## Classes and COM objects／クラスと COM オブジェクト

- **[PRB]** Common issues when converting VB6 applications that use type libraries／タイプライブラリを使用する VB6 アプリケーションを変換する際の一般的な課題
- **[PRB]** A reference to a COM type library is missing／COM タイプライブラリに対する参照が失われてしまう
- **[PRB]** The VB.NET application throws a NullReference exception (or some other unexpected error) during the initialization phase／イニシャライゼーションの過程で Vb.NET アプリケーションが NullReference エラー（もしくは他の予期せぬエラー）を起こしてしまう
- **[HOWTO]** Get rid of the "Class" suffix used for COM classes／COM クラスに使用される接尾辞「Class」を取り除く
- **[HOWTO]** Use different versions of same COM type library or ActiveX control／同じ COM タイプライブラリまたは ActiveX コントロールの異なるバージョンを使用する
- **[PRB]** Members in COM libraries might be missing or raise unexpected errors／COM ライブラリのメンバが失われたり、予期せぬエラーを起こしたりする
- **[HOWTO]** Deal with Null, Empty, and missing values／Null と Empty と欠損値を取り扱う
- **[HOWTO]** Ensure that properties with both Property Let and Property Set are translated correctly／Property Let と Property Set の両方が正常に変換されていることを確認する

- **[INFO]** Properties may be migrated without any warning related to VB6's ByRef parameters being converted to VB.NET's ByVal parameters/VB.NET の ByVal パラメータに変換される VB6 の ByRef パラメータに関する警告なしでプロパティが変換される可能性がある
- **[HOWTO]** Change the base class of a VB.NET class/VB.NET クラスの基底クラスを変更する
- **[HOWTO]** Work with auto-instantiating arrays/自動インスタンス化配列を伴った動作をさせる
- **[PRB]** Objects returned from COM methods may cause an InvalidCastException error/COM メソッドが返すオブジェクトが InvalidCastException エラーの原因となる可能性がある
- **[PRB]** The ScriptControl can't interact with classes that aren't visible to COM/ScriptControl は COM からは見えないクラスとは相互作用することができない
- **[HOWTO]** Avoid compilation errors caused by unsupported designers/サポートされないデザイナーを使ってコンパイルエラーを回避する
- **[HOWTO]** Fix VB6 client applications that reference a COM DLL that has migrated to VB.NET using binary compatibility/バイナリ互換を使用して VB.NET に変換した COM DLL を参照する VB6 クライアントアプリケーションを修正する
- **[HOWTO]** Optimize startup time in N-tier applications/N 層アプリケーションで起動時間を最適化する
- **[HOWTO]** Avoid reentrancy problems in free-threaded .NET components/フリースレッドな .NET コンポーネントの再入可能な問題を回避する

## Database and data-binding/データベースとデータバインディング

- **[PRB]** Connection and Recordset objects should be closed explicitly/Connection オブジェクトと Recordset オブジェクトは明示的に閉じられる必要がある
- **[PRB]** The Execute method of the ADODB.Connection object can create a spurious Recordset object/ADODB.Connection オブジェクトの Execute メソッドは偽のレコードセットオブジェクトを作成する可能性があります。
- **[PRB]** The DAO RepairDatabase method causes a compilation error/DAO の RepairDatabase メソッドはコンパイルエラーの原因となる
- **[PRB]** DAO and RDO Data control may not update bound controls when record position changes/DAO と RDO のデータコントロールはレコードの場所が変わったときにバインドコントロールを更新しない可能性がある。
- **[PRB]** The migrated VB.NET has a "Name 'xyz' is not declared" compilation error/変換後の VB.NET に「『xyz』は宣言されていません」というコンパイルエラーが出ている。
- **[INFO]** ADO-related code in the migrated VB.NET application might not behave like the original VB6 code/変換後の VB.NET アプリケーションの ADO に関連するコードは元の VB6 のコードのように動作しない可能性がある
- **[INFO]** Classes and user controls generated by VB6 Data Object Wizard aren't supported/VB6 のデータオブジェクトウィザードによって生成されたクラスとユーザーコントロールはサポートされない
- **[BUG]** The Validate event may not fire for the Data and RDO Data control/データコントロールと RDO データコントロールについては Validate イベントは発生しない可能性がある
- **[PRB]** Assigning the ActiveConnection and Source properties of an ADODB.Recordset object may cause a runtime error/ADODB.Recordset オブジェクトの ActiveConnection プロパティと Source プロパティへの代入はランタイムエラーの原因となることがある

- **[HOWTO]** Solve syntax errors in ADO DC event handlers / ADO DC イベントハンドラの構文エラーを解決する
- **[PRB]** Passing an ADO read-only property to a ByRef argument causes a COMException / ByRef 引数に ADO の読み取り専用プロパティを渡すと COMException エラーが発生する
- **[PRB]** VB.NET doesn't allow to close an ADO Connection while a transaction is active / VB.NET ではトランザクション処理中は ADO 接続を閉じることはできない
- **[PRB]** Disappearing records in client-side ADO DB.Recordset with batch optimistic lock / 一括楽観的ロックを行うクライアントサイドの ADO DB.Recordset のレコードが消えてしまう
- **[PRB]** The data source of a bound control isn't updated if the control's value is modified via code / コントロールの値がコード上で変更された場合、バインドコントロールのデータソースが更新されない
- **[PRB]** The Data control may fail when accessing a non-Access data source / 非アクセスデータソースにアクセスしようとするデータコントロールが失敗する可能性がある
- **[PRB]** Passing ADOX properties to ByRef argument may cause unexpected runtime exceptions / ByRef 引数に対して ADOX プロパティを渡すと予期せぬランタイムエラーが発生する可能性がある
- **[PRB]** The DataBindings property of a bound user control isn't migrated / バインドされたユーザーコントロールの DataBindings プロパティが変換されない
- **[INFO]** adDBFileTime enum values are converted to ADO DB\_DataTypeEnum\_adDBFileTime constants / adDBFileTime 列挙型は ADO DB\_DataTypeEnum\_adDBFileTime 定数に変換される
- **[INFO]** Some properties of VB6DataGrid control can't be modified from inside Visual Studio's property window / VB6DataGrid コントロールのプロパティのいくつかは Visual Studio のプロパティウィンドウから変更することができない
- **[PRB]** Code that references a VB6DataGrid column by its string key may not work in the migrated VB.NET program / 文字キーによって VB6DataGrid のカラムを参照するコードは変換後の VB.NET プログラムで動作しない可能性がある
- **[INFO]** DBGrid control isn't supported / DBGrid コントロールはサポートされていません
- **[PRB]** DataGrid and MSHFlexGrid controls don't work as expected and/or throw exceptions / DataGrid と MSHFlexGrid コントロールは予想通りに動かなかったり、エラーを起こしたりする。
- **[PRB]** Changing properties of the data source associated with a DataGrid control can throw an exception / DataGrid コントロールに関連付くデータソースのプロパティを変更するとエラーが発生することがある
- **[PRB]** Assigning a StdDataFormat object to the DataFormat property of an ADO DB.Field causes runtime errors / ADO DB.Field の DataFormat プロパティに StdDataFormat オブジェクトを代入するとランタイムエラーが発生する。
- **[PRB]** Column formatting via StdDataFormat objects doesn't work for DataGrid controls / DataGrid コントロールに対して StdDataFormat オブジェクトを使ったカラムフォーマットはできない
- **[PRB]** The DataGrid's SelColChange event may fire unexpectedly / DataGrid の SelColChange イベントは勝手に発生することがある
- **[PRB]** Data classes might need manual adjustments / Data クラスは手作業で調整する必要がある



## sort by date／日付順

---

記事が翻訳されている場合、リンク先に翻訳記事が表示されます。

- [\[INFO\] VB Migration Partner uses the Line, Shape, and Printer classes in Microsoft VB PowerPacks／VB Migration Partner は Microsoft VB PowerPacks の Line クラスと Shape クラスと Printer クラスを使用する](#)
- [\[HOWTO\] Reduce flickering when loading or resizing a form with many controls／多数のコントロールを持つフォームをリサイズしたり読み込んだりするときのチラツキを抑える](#)
- [\[PRB\] Setting the current tab in a SSTab control brings the input focus to the control／SSTab コントロールのカレントタブの設定によってコントロールが入力可能状態になってしまう](#)
- [\[HOWTO\] Modify migrated projects to use CodeArchitects.VBPowerPack.dll／CodeArchitects.VBPowerPack.dll を使用するために変換されたプロジェクトを変更する](#)
- [\[HOWTO\] Leverage .NET TextAlign property with command, checkbox, and option button controls／コマンドボタンやチェックボックスやオプションボタンに.NET の TextAlign プロパティを利用する](#)
- [\[HOWTO\] Use controls from VB Migration Partner's support library on a standard .NET form／標準の.NET フォーム上で VB Migration Partner のサポートライブラリのコントロールを使用する](#)
- [\[PRB\] The application stops with the "ActiveX control is windowless. Windowless ActiveX controls aren't supported" error message／「ActiveX コントロールはウインドウを持っていません。ウインドウを持たない ActiveX コントロールはサポートされていません。」というエラーでアプリケーションが終了してしまう](#)
- [\[INFO\] Minor differences between CommonDialog, Printer, Line, and Shape VB6 classes and their implementation in CodeArchitects.VBPowerPack.dll／VB6 のコモンダイアログクラスや Printer クラスや Line クラスや Shape クラスと CodeArchitects.VBPowerPack.dll に実装されたクラスの違い](#)
- [\[INFO\] Migrated VB.NET code might not work in partially-trusted applications／信頼性が中途半端なアプリケーションでは、変換された VB.NET コードは動作しない可能性があります。](#)
- [\[PRB\] A migrated form doesn't fire the Paint event／変換後のフォームで Paint イベントが発生しない](#)
- [\[PRB\] Can't use Tab key to move the focus to a DataGrid ActiveX control／DataGrid ActiveX コントロールにフォーカス移動させるためにタブキーを使用することができない](#)
- [\[INFO\] Controls in the support library can't be dropped on a standard VB.NET form／サポートライブラリのコントロールは標準の VB.NET フォームに置くことができない](#)
- [\[PRB\] Errors when re-opening modal, default forms that host an ActiveX control／ActiveX コントロールを含むモーダルのデフォルトフォームを再オープンするときのエラー](#)
- [\[PRB\] Common problems when generating an ActiveX wrapper with AxWrapperGen／AxWrapperGen を使用して ActiveX のラッパーを生成したときの一般的な問題](#)
- [\[HOWTO\] Using migrated .NET DLLs from VB6, ASP, and other COM clients／VB6、ASP、その他の COM クライアントから.NET DLL に変換する](#)
- [\[PRB\] Drag-and-drop operations inside a ListBox control may deliver fake results／ListBox コントロール内でのドラッグ&ドロップ操作が見せかけの動作をする可能性がある](#)
- [\[PRB\] Compatibility problems with MS Build／MS Build との互換性の問題](#)
- [\[INFO\] Minor differences among different versions of Microsoft Windows Common Controls library／Microsoft Windows コモンコントロールライブラリのバージョン間の相違](#)

- **[PRB]** The FileLineInput6 statement uses CR-LF pairs as delimiters／FileLineInput6 は区切り文字として CRLF を使用する
- **[PRB]** SmallChange property can't be higher than LargeChange／SmallChange プロパティが LargeChange より大きくならない
- **[PRB]** A form loaded with Load method might cause errors in its Unload event handler／Load メソッドでロードされたフォームが自身の Unload イベントハンドラーのエラーの原因になってしまうかもしれない
- **[PRB]** Control events don't fire／コントロールのイベントが動作しない
- **[PRB]** VB Migration Partner generates incomplete classes／VB Migration Partner が不完全なクラスを生成する
- **[PRB]** [Forms containing an ImageList control don't display at design-time after updating the CodeArchitects.VBLibrary DLL.／CodeArchitects.VBLibrary DLL を更新した後、ImageList コントロールを含むフォームがデザイナーに表示されない](#)
- **[PRB]** Invoking the Show method of MDI child forms containing one or more ActiveX controls might not fire the Load event／ActiveX コントロールを含む MDI 子フォーム Show メソッドを呼び出しても Load イベントが発生しない可能性がある
- **[PRB]** ScaleHeight and ScaleWidth properties aren't correctly serialized at design-time／ScaleHeight プロパティと ScaleWidth プロパティがデザイナー上では正しい順番で並ばない
- **[PRB]** Setting the Visible=False for a Frame may cause spurious GotFocus, LostFocus, and Validate events／Frame に対する Visible=False 設定によって GotFocus イベントや LostFocus イベントや Validate イベントが発生してしまう
- **[PRB]** Top property of controls hosted inside an SSTab control is relative to TabPage border／SSTab コントロール内部に配置されたコントロールの Yop プロパティが TabPage の境界と関連付けられてしまう
- **[HOWTO]** Define a more reliable Ambient.UserMode property／より信頼性の高い Ambient.UserMode プロパティを定義する
- **[HOWTO]** Write an extender DLL that can manipulate design-time properties of converted controls／変換されたコントロールのデザイン用プロパティを操作することができる拡張 DLL を記述する
- **[HOWTO]** Check an item of a ListBox with Style=CheckBoxes with a single mouse click／Style=CheckBoxes のリストボックスのアイテムをシングルクリックで確認する
- **[PRB]** Setting the Font property at runtime causes an unexpected Paint event which may freeze the application／実行時のフォントプロパティの設定がアプリケーションをフリーズさせる可能性のある予期せぬ Paint イベントの原因となってしまう。
- **[PRB]** Setting the Font property inside a Paint event causes an endless loop／Paint イベント内部のフォントプロパティの設定が終わりなきループ処理の原因となってしまう
- **[HOWTO]** Skipping one or more methods when applying the PreProcess or PostProcess pragmas／PreProcess プラグマや PostProcess プラグマを適用するときの一つ以上のメソッドをスキップする
- **[INFO]** [How to migrate the PictureDn, PictureUp, and PictureDisabled properties of SSCommand and SSRibbon controls.／SSCommand コントロールと SSRibbon コントロールの PictureDn プロパティと PictureUp プロパティと PictureDisabled プロパティを変換する方法](#)
- **[PRB]** The MsgBox, InputBox, and common dialogs cause spurious LostFocus and GotFocus events／MsgBox や InputBox やコモンダイアログが誤った LostFocus イベントや GotFocus イベントの原因となってしまう
- **[HOWTO]** Change the value of a given property of all controls of specified type or name／特定の型や名前のコントロールのプロパティの値を変更する
- **[PRB]** Mouse custom cursor in drag-and-drop operations with ActiveX controls doesn't work correctly／ActiveX コントロールでのドラッグ & ドロップ操作においてマウスのカスタムカーソルは正常に動作しない

- **[PRB]** “Classic” (VB3-style) drag-and-drop stops working／「古い」(VB3 スタイルの)ドラッグ & ドロップは動作を停止させる
- **[HOWTO]** Speed up string concatenation／文字連結を高速化する
- **[INFO]** Minor differences between VB6 and VB.NET SStab controls／VB6 と VB.NET の SStab コントロールの相違
- **[PRB]** The Line property of output TextStream always returns 1／TextStream 出力 の Line プロパティが常に 1 を返してしまう
- **[PRB]** Removing an item from a single-selectable ListBox selects the previous item／単一選択可能なリストボックスからアイテムを削除するとひとつ前のアイテムが選択されてしまう
- **[INFO]** DoEvents6 calls can cause weird user interface behaviors／DoEvents6 はユーザーインターフェースの動きを奇妙なものにすることがある
- **[PRB]** Label and Image controls appear in front of other controls／Label コントロールと Image コントロールは他のコントロールの前に表示される
- **[PRB]** The migrated application takes too long at loading its first form／変換後のアプリケーションが最初のフォームを呼び出すとき時間がかかりすぎる
- **[PRB]** An MDI Child form can mistakenly display a menu bar／MDI 子フォームがメニューバーを誤って表示してしまう可能性がある
- **[PRB]** Can't mix VB graphic properties and GDI native calls／VB の画像プロパティと GDI 固有の呼び出しは混在させることができない
- **[PRB]** The CLS method fails to refresh child controls on the form／CLS メソッドがフォーム上のコントロールのリフレッシュに失敗してしまう
- **[PRB]** The VB.NET application hangs on the splash screen or terminates earlier than expected／スプラッシュスクリーンで VB.NET アプリケーションがハングアップしたり、予想よりも早く終了してしまう
- **[PRB]** Multiple or missing Paint events／多重 Paint イベントと Paint イベントの喪失
- **[PRB]** Assignments to Font properties generate runtime errors in late-bound mode／遅延バインディングではフォントプロパティの代入が実行時エラーを発生させる
- **[PRB]** Menus on MDI child forms doesn't replace the toplevel menu of the MDI parent form／MDI 子フォーム上のメニューが MDI 親フォームトップレベルのメニューを置き換えない
- **[HOWTO]** Modify the behavior of methods and classes in VB.NET library and VB Migration Partner support library／VB.NET ライブラリと VB Migration Partner サポートライブラリのメソッドとクラスの動作を変更する
- **[PRB]** VB Migration Partner fails to recognize a reference to a VB6 project／VB Migration Partner が VB6 プロジェクトに対する参照の認識に失敗する
- **[PRB]** Out of memory exceptions when converting very large projects／非常に大きなプロジェクトを変換したときにメモリ不足例外が発生する
- **[INFO]** Minor differences between VB6 and VB.NET controls belonging to the Threed (ActiveTreedPlus) control library／VB6 と VB.NET の Threed(ActiveTreedPlus)に属するコントロールの相違
- **[PRB]** The Data control may fail when accessing a non-Access data source／非アクセスデータソースにアクセスしようとするデータコントロールが失敗する可能性がある
- **[HOWTO]** Using the NGEN tool to speed up the startup phase of a migrated .NET application／変換されたアプリケーションのスタートアップ処理を高速化する NGEN ツールを使用する
- **[HOWTO]** Convert weak object references and the ObjPtr keyword to VB.NET／曖昧なオブジェクト参照と ObjPtr キーワードを VB.NET に変換する
- **[HOWTO]** Migrating VB6 reports to NET／VB6 のレポートを.NET に変換する

- [\[PRB\] Assigning a StdDataFormat object to the DataFormat property of an ADODB.Field causes runtime errors / ADODB.Field の DataFormat プロパティに StdDataFormat オブジェクトを代入するとランタイムエラーが発生する。](#)
- [\[PRB\] Column formatting via StdDataFormat objects doesn't work for DataGrid controls / DataGrid コントロールに対して StdDataFormat オブジェクトを使ったカラムフォーマットはできない](#)
- [\[PRB\] Form-level variables in default forms don't retain their values between calls / デフォルトフォームのフォームレベル変数が呼び出しと呼び出しの間で保持されていない](#)
- [\[PRB\] Structures containing DBCS strings may cause exceptions when passed to an external \(Declare\) method / 外部の \(Declare\) メソッドに渡された場合、2 バイト文字を含む構造体はエラーの原因となる可能性がある](#)
- [\[PRB\] Most VB.NET methods don't work well with strings that contain ASCII 0 characters / ほとんどの VB.NET のメソッドが ASCII 0 の文字を含む文字列との相性が悪い](#)
- [\[PRB\] Closing a form causes the ObjectDisposed exception / フォームを閉じると ObjectDisposed エラーが発生してしまう](#)
- [\[HOWTO\] Generate auto-implemented properties for Visual Basic 2010 / Visual Basic 2010 用の自動実装プロパティを生成する](#)
- [\[HOWTO\] Incrementally migrate a large VB6 project and avoid the "Maximum number of errors has been exceeded" message / 巨大な VB6 プロジェクトを段階的に変換して「エラーの最大数を越えた」エラーを回避する](#)
- [\[PRB\] The Len6 method can return invalid values when applied to a structure containing fixed-length strings / 固定長文字列を含む構造体で使用された場合、Len6 メソッドは無効な値を戻すことがある](#)
- [\[PRB\] Serializable properties in UserControls using system objects may crash Visual Studio / システムオブジェクトを使用しているユーザーコントロールの直列化可能なプロパティのせいで Visual Studio を強制終了する可能性がある](#)
- [\[HOWTO\] Create a report with all the migration issues and warnings / すべての移行課題と警告のレポートを作成する](#)
- [\[PRB\] ScrollBar controls ignore the LargeChange property / ScrollBar コントロールが LargeChange プロパティを無視してしまう](#)
- [\[PRB\] ObjectDisposed exception where re-opening a closed form / 閉じたフォームを再オープンするときの ObjectDisposed エラー](#)
- [\[PRB\] VB Migration Partner may stop when converting VB6 apps that reference a .NET DLL / .NET DLL を参照する VB6 アプリケーションを変換するとき VB Migration Partner が停止する可能性がある](#)
- [\[HOWTO\] Replace Collection with the VB6CollectionVariant type / VB6CollectionVariant 型でコレクションを置換する](#)
- [\[PRB\] VB6 and VB.NET store arrays into Collection objects in different ways / VB6 と VB.NET が異なる方法で配列をコレクションに格納する](#)
- [\[PRB\] The IsArray method returns False with uninitialized array stored in Object variables / オブジェクト変数に格納された初期化されていない配列を伴う IsArray メソッドは False を返す](#)
- [\[PRB\] NullReference exception in Form\\_Initialize / Form\\_Initialize における NullReference エラー](#)
- [\[PRB\] UserControl classes don't raise the InitProperties, ReadProperties and WriteProperties events / ユーザーコントロールクラスで InitProperties イベントや ReadProperties イベントや WriteProperties イベントが起きない](#)
- [\[HOWTO\] Reduce flickering in graphic operations / 画像処理のチラツキを抑える](#)
- [\[PRB\] Scrollbars in MDI forms are always visible / MDI フォームのスクロールバーは常に可視化されている](#)
- [\[PRB\] Elements of control arrays can't be referenced before being created / コントロール配列の要素が作成される前に参照されない可能性がある](#)
- [\[PRB\] Hotkeys in SSTab caption work only if the form's KeyPreview property is set to True / フォームの KeyPreview プロパティが True の場合にしか SSTab の見出しのホットキーが動作しない](#)

- **[HOWTO]** Fix ReDim statements that change the rank of an array／配列の rank を変更する ReDim ステートメントを修正する
- **[INFO]** Minor differences between VB6 and VB.NET implementation of DDE／DDE の実装における VB6 と VB.NET との間の相違
- **[PRB]** Compilation errors with classes or usercontrols that expose events with same name as other members and that are used to define an interface／他のメンバと同じ名前のイベントを公開するクラスやユーザーコントロール、またはインターフェースを定義するために使用されているクラスやユーザーコントロールでコンパイルエラーになる
- **[PRB]** Connection and Recordset objects should be closed explicitly／Connection オブジェクトと Recordset オブジェクトは明示的に閉じられる必要がある
- **[PRB]** The Execute method of the ADODB.Connection object can create a spurious Recordset object／ADODB.Connection オブジェクトの Execute メソッドは偽のレコードセットオブジェクトを作成する可能性があります。
- **[INFO]** The Erase statement doesn't work well with VB6 static arrays／Erase ステートメントは VB6 の静的配列と相性が悪い
- **[PRB]** Arrays held in Variant and Object variables are converted in a special way／バリエーションに保持されている配列とオブジェクト変数は特別な方法で変換される
- **[HOWTO]** Get rid of warnings related to Screen.MousePointer property／Screen.MousePointer プロパティに関する警告を取り除く
- **[HOWTO]** Determine if unsupported members throw an exception at runtime／実行時にサポートされないメンバが例外を発生させた場合に決定する
- **[PRB]** The list area of a DataCombo or DataList control isn't automatically updated／DataCombo コントロールや DataList コントロールのリスト領域は自動的に更新されない
- **[PRB]** The data source of a bound control isn't updated if the control's value is modified via code／コントロールの値がコード上で変更された場合、バインドコントロールのデータソースが更新されない
- **[INFO]** Minor differences between VB6 and VB.NET ImageList controls／VB6 と VB.NET の ImageList コントロールの相違
- **[INFO]** How VB Migration Partner implements drag-and-drop features／VB Migration Partner がドラッグアンドドロップ機能を実装する方法
- **[PRB]** Stand-alone UserControl, UserDocument, and PropertyPage keywords might not be converted correctly／独立した UserControl や UserDocument や PropertyPage といったキーワードは正常に変換されないことがある
- **[PRB]** Structures can't have the same name as other members in VB.NET／VB.NET では構造体は他のメンバと同じ名称を持つことができない
- **[PRB]** Compilation error when implementing an interface that defines a collection class／コレクションクラスを定義するインターフェースを実装するときコンパイルエラーが発生する
- **[INFO]** How VB Migration Partner deals with late-bound calls／VB Migration Partner が遅延バインディング呼び出しを取り扱う方法
- **[PRB]** Private UserControls aren't loaded with Controls.Add method／プライベート型のユーザーコントロールは Controls.Add method に読み込まれない
- **[PRB]** Disappearing records in client-side ADODB.Recordset with batch optimistic lock／一括楽観的ロックを行うクライアントサイドの ADODB.Recordset のレコードが消えてしまう
- **[INFO]** All ListBox and Frame controls in an array must have same style／配列内の ListBox コントロールと Frame コントロールは同じスタイルでなければならない
- **[PRB]** ActiveX controls aren't migrated correctly／ActiveX コントロールが正常に変換されない

- **[INFO]** UDT fields accessed via a VB6Variant variable are readonly/VB6Variant 変数経由でアクセスされるユーザー定義型は読み取り専用です。
- **[HOWTO]** Convert byte-oriented string functions/バイト指向型文字列ファンクションを変換する
- **[PRB]** Default member isn't resolved for arguments of IIF and Choose methods/デフォルトメンバが IIF の引数と Choose メソッドに対して解決されない
- **[HOWTO]** Ensure that properties with both Property Let and Property Set are translated correctly/Property Let と Property Set の両方が正常に変換されていることを確認する
- **[PRB]** VB.NET applications crash with "Unable to create ActiveX object" error/VB.NET アプリケーションが「ActiveX オブジェクトを作成できない」エラーでクラッシュする
- **[HOWTO]** Leverage the features of native .NET controls/.NET コントロール固有の機能を利用する
- **[HOWTO]** Retrieving a reference to the actual ActiveX control wrapped in a class generated with AxWrapperGen/AxWrapperGen で生成されたクラスでラップされた ActiveX コントロールに対する参照を検索する
- **[HOWTO]** Convert VB6 files that are shared among projects/複数のプロジェクトで共有される VB6 ファイルを変換する方法
- **[HOWTO]** [Improve code generation with Variant variables and parameters/Variant 変数とパラメータを伴うコード生成を改善する](#)
- **[PRB]** The VB.NET application can't compile because a constant in a Structure is ambiguous between two modules /構造体の定数がふたつのモジュールの間で曖昧なので VB.NET アプリケーションがコンパイルできない
- **[HOWTO]** Optimize startup time in N-tier applications/N 層アプリケーションで起動時間を最適化する
- **[PRB]** PictureBox, Frame and other container controls don't correctly process the Tab key if TabStop=False/ PictureBox コントロールや、Frame コントロール、そしてその他のコンテナコントロールが TabStop=False の場合に Tab キーを正常に処理しない
- **[PRB]** VB.NET controls are taller than the original VB6 control/VB.NET コントロールの高さが元の VB6 コントロールよりも大きくなってしまう。
- **[HOWTO]** Avoid reentrancy problems in free-threaded .NET components/.NET のフリースレッドコンポーネントの再入可能な問題を回避する
- **[HOWTO]** Apply pragmas only to variables of specific types/特定の型の変数にだけプラグマを適用する
- **[PRB]** The VB.NET application can't compile because of an "'AddressOf' expression cannot be converted to 'Integer' because 'Integer' is not a delegate type" error/「Integer はデリゲート型ではない」というエラーのために AddressOf 式を Integer に変換できないため、VB.NET アプリケーションがコンパイルできない
- **[PRB]** The Default and Cancel properties of a VB6UserControl object don't behave like in VB6/VB6UserControl オブジェクトの Default プロパティと Cancel プロパティが VB6 のように動作しない
- **[PRB]** The vbDefaultButton4 constant might not be converted correctly/vbDefaultButton4 定数は正しく変換されない可能性がある
- **[INFO]** Binary properties of a few controls aren't migrated correctly/いくつかのコントロールのバイナリープロパティは正常に変換されない
- **[INFO]** Minor differences between VB6 and VB.NET Command controls/VB6 と VB.NET の Command コントロールの相違
- **[PRB]** Members in COM libraries might be missing or raise unexpected errors/COM ライブラリのメンバが失われたり、予期せぬエラーを起こしたりする

- **[HOWTO]** Fix VB6 client applications that reference a COM DLL that has migrated to VB.NET using binary compatibility／バイナリ互換を使用して VB.NET に変換した COM DLL を参照する VB6 クライアントアプリケーションを修正する
- **[HOWTO]** Execute multiple instances of VB Migration Partner／VB Migration Partner をマルチインスタンスで実行する方法
- **[HOWTO]** Terminate the migration when a warning message is emitted／警告メッセージが発行されたときに変換を中止する
- **[HOWTO]** Solve issues related to Null values in database fields／データベース列の NULL 値に関わる問題を解決する
- **[HOWTO]** Use different versions of same COM type library or ActiveX control／同じ COM タイプライブラリまたは ActiveX コントロールの異なるバージョンを使用する
- **[PRB]** A FileGet6 or FilePut6 method throws an exception while attempting to read or write a User-Defined Type variable／ユーザー定義型変数の読み書きをしようとすると FileGet6 や FilePut6 といったメソッドが例外を起こす
- **[HOWTO]** Get rid of the "Class" suffix used for COM classes／COM クラスに使用される接尾辞「Class」を取り除く
- **[PRB]** Errors when doing unit testing or using reflection to explore the support library／単体テストをしているときやサポートライブラリを探して反映させるときにエラーが発生する
- **[HOWTO]** Assign a specific data type when the As clause is missing／As 節がないとき、特定のデータ型を割り当てる
- **[HOWTO]** Permanently display label and button accelerators／ラベルとボタンのアクセラレータを恒久的に表示する
- **[PRB]** VB.NET arrays are copied using "shallow" semantics／VB.NET の配列は「shallow」の動作を使用してコピーされる
- **[PRB]** WLOption controls ignore the Group property／WLOption コントロールは Group プロパティを無視してしまう
- **[INFO]** The End keyword is converted as ApplicationExit6／END キーワードは ApplicationExit6 に変換される
- **[HOWTO]** Using the SetType pragma with optional parameters／オプションパラメータを伴う SetType プラグマを使用する
- **[INFO]** You need a runtime license for RDO and for ActiveX controls when running on a computer on which VB6 isn't installed／VB6 がインストールされていない PC で実行するには RDO と ActiveX コントロール用のランタイムライセンスが必要となる
- **[INFO]** VB.NET floating point divisions never raise Division-by-zero errors／VB.NET の浮動小数点の割り算は 0 で割ってもエラーにならない
- **[INFO]** VB Migration Partner generates signed interop assemblies／VB Migration Partner は署名された相互運用アセンブリを生成します。
- **[PRB]** The DAO RepairDatabase method causes a compilation error／DAO の RepairDatabase メソッドはコンパイルエラーの原因となる
- **[HOWTO]** Temporarily disable pragmas for debugging purposes／デバッグのため一時的にプラグマを無効にする
- **[PRB]** PreProcess or PostProcess pragma can make VB Migration Partner to hung／PreProcess プラグマや PostProcess プラグマが VB Migration Partner をハングさせる可能性がある
- **[HOWTO]** Convert from OLE types to .NET types／OLE 型から.NET 型へ変換する
- **[INFO]** VB Migration Partner doesn't support references to external EXE projects／VB Migration Partner は外部の EXE プロジェクトに対する参照をサポートしません。
- **[HOWTO]** Limit the scope of PreProcess and PostProcess pragmas／PreProcess プラグマや PostProcess プラグマの範囲を制限する
- **[HOWTO]** Avoid Option Explicit Off in generated VB.NET files／生成された VB.NET ファイルの Option Explicit Off を避ける

- **[HOWTO]** Optimize Boolean expressions using PostProcess pragmas / PostProcess プラグマを使ってブーリアン式を最適化する
- **[HOWTO]** Convert While loops into Do loops / While loop を Do loop に変換する
- **[PRB]** A nonexisting property is generated for a control in the \*.Designer.vb file / \*.Designer.vb ファイルのコントロールに対して存在しないプロパティが生成されてしまう
- **[PRB]** Changing the type of a variable causes a Type Mismatch compilation error / 変数の型を変更すると型不一致エラーが発生する
- **[PRB]** Pragmas related to arrays might not work if applied to a ReDim statement / ReDim ステートメントに適用される場合、配列に関するプラグマが動作しないかもしれない
- **[PRB]** A locked EXE or DLL file causes the migration process or compilation command to fail / ロックされた EXE や DLL ファイルが変換プロセスやコンパイルコマンドを失敗させる
- **[PRB]** The VB.NET application ends with a fatal exception when the splash screen closes / スプラッシュスクリーンを閉じたときに VB.NET アプリケーションが致命的エラーで終了してしまう
- **[PRB]** Conversion fails with a "Project item name not found" error / 「プロジェクトのアイテム名が見つからない」エラーで変換が失敗する
- **[INFO]** Minor differences between VB6 and VB.NET TabStrip controls / VB6 と VB.NET の TabStrip コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET MSChart controls / VB6 と VB.NET の MSChart コントロールの相違
- **[INFO]** Controlling the GotFocus and MouseDown event sequence / GotFocus イベントと MouseDown イベントの順番を制御する
- **[INFO]** Controlling the LostFocus and Validate event sequence / LostFocus イベントと Validate イベントの順番を制御する
- **[INFO]** GotFocus and LostFocus events in UserControl / ユーザーコントロールの GotFocus イベントと LostFocus イベント
- **[HOWTO]** Deal with Null, Empty, and missing values / Null と Empty と欠損値を取り扱う
- **[HOWTO]** Generate Visual Studio 2008 projects / Visual Studio 2008 プロジェクトを生成する
- **[INFO]** Minor differences between VB6 and VB.NET DataGrid controls / VB6 と VB.NET の DataGrid コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET DriveListBox, DirListBox, and FileListBox controls / VB6 と VB.NET の DriveListBox コントロールと DirListBox コントロールと FileListBox コントロールの相違
- **[HOWTO]** Change the base class of a VB.NET class / VB.NET クラスの基底クラスを変更する
- **[HOWTO]** Modify project-level options in VB.NET programs / VB.NET プログラムのプロジェクトレベルオプションを変更する
- **[HOWTO]** Enforce project-level Option Strict Off setting in VB.NET programs / VB.NET プログラムのプロジェクトレベルオプションを強制的に Strict Off 設定にする
- **[HOWTO]** Move a control over a toolbar at runtime / 実行時にツールバー上にコントロールを移動する
- **[INFO]** Minor differences between VB6 and VB.NET Toolbar controls / VB6 と VB.NET の Toolbar コントロールの相違
- **[PRB]** [Assignments to the ListView.SelectedItem property are ignored if the control isn't visible](#) / [コントロールが非表示の場合、ListView.SelectedItem プロパティに対する代入は無視されてしまう](#)
- **[PRB]** Selecting the contents of a TextBox when the control gets the input focus doesn't work in VB.NET / コントロールが入力状態となったとき、TextBox の内容の選択が VB.NET では行われない



- **[INFO]** Minor differences between VB6 and VB.NET RemoteData controls／VB6 と VB.NET の RemoteData コントロールの相違
- **[PRB]** The Load event of a form doesn't fire when you access its property or control from another form／フォームのプロパティや他のフォームのコントロールにアクセスするときにフォームの Load イベントが動作しない
- **[INFO]** Fields are restored in a different way if an error occurs with a Data control／データコントロールにエラーが発生した場合、フィールドが異なる方法で回復される
- **[INFO]** Minor differences between VB6 and VB.NET DataList controls／VB6 と VB.NET の DataList コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET RichTextBox controls／VB6 と VB.NET の RichTextBox コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET DTPicker controls／VB6 と VB.NET の DTPicker コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET ImageCombo controls／VB6 と VB.NET の ImageCombo コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET MonthView controls／VB6 と VB.NET の MonthView コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET UpDown controls／VB6 と VB.NET の UpDown コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET TreeView controls／VB6 と VB.NET の TreeView コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET StatusBar controls／VB6 と VB.NET の StatusBar コントロールの相違
- **[INFO]** Minor differences between VB6 and VB.NET DataCombo controls／VB6 と VB.NET の DataCombo コントロールの相違
- **[INFO]** VB Migration Partner setup fails if VB6 isn't installed／VB6 がインストールされていない場合、VB Migration Partner のセットアップが失敗します。
- **[PRB]** A reference to the Controls collection isn't converted to Controls6／コントロールコレクションに対する参照が Controls6 へ変換されていない
- **[INFO]** Default methods with arguments are converted to read-only properties／引数を伴うデフォルトメソッドは読み取り専用プロパティに変換される
- **[PRB]** Limitations in the conversion of #If blocks／#If ブロックの変換上の制限
- **[INFO]** Converted VB.NET applications target 32-bit CPUs by default／デフォルトでは、移行されたアプリケーションは 32bit をターゲットとします。
- **[INFO]** Minor differences between VB6 and VB.NET ListView controls／VB6 と VB.NET の ListView コントロールの相違
- **[HOWTO]** Create references to .NET assemblies with absolute path／.NET アセンブリへの参照を絶対パスで作成する
- **[PRB]** VB.NET doesn't allow to close an ADO Connection while a transaction is active／VB.NET ではトランザクション処理中は ADO 接続を閉じることはできない
- **[PRB]** VB.NET doesn't support implicit conversions in Const statements／VB.NET は Const の暗黙的な変換をサポートしていない
- **[INFO]** Two VB.NET project can't coexist in the same directory／2 つの VB.NET プロジェクトが同じフォルダに共存することはできません。
- **[INFO]** OLE\_COLOR type is converted to System.Drawing.Color／OLE\_COLOR は System.Drawing.Color に変換される

- **[INFO]** HitTest method of the TreeView control can return different values in VB.NET / TreeView コントロールの HitTest メソッドは VB.NET では異なる値を返す可能性がある
- **[HOWTO]** Transform “Not x Is y” expressions into “x IsNot y” expressions / 「Not x Is y」式を「x IsNot y」式に変換する
- **[PRB]** VB Migration Partner is unable to run Visual Studio / VB Migration Partner が Visual Studio を起動することができない。
- **[PRB]** Conversion fails with an “Unable to find ResToResx.exe tool” error / 「ResToResx.exe が見つからない」エラーで変換が失敗する
- **[INFO]** Negative values for ScaleWidth and ScaleHeight properties cause incorrect output in Microsoft Vista / Windows Vista では ScaleWidth プロパティと ScaleHeight プロパティに設定した負の値が異常な出力の原因となってしまう
- **[HOWTO]** Replace App6 properties with native VB.NET members / App6 プロパティをネイティブの VB.NET のメンバで置き換える
- **[PRB]** DTPicker controls display their content using the LongDate format / DTPicker コントロールがその内容を LongDate 書式で表示してしまう
- **[HOWTO]** Reduce runtime errors caused by null strings / NULL 文字による実行時エラーを減らす
- **[PRB]** Mouse event aren't received by transparent Image and Label controls / 透明な Image コントロールと Label コントロールは Mouse イベントを受け取らない
- **[PRB]** Calls to GetOpenFileName API method returns trimmed string / GetOpenFileName API メソッドの呼び出しが切り取られた文字列を返す
- **[INFO]** VB.NET property values passed to a ByRef parameter can be modified / ByRef パラメータに渡される VB.NET プロパティの値は変更できる
- **[PRB]** Image controls whose Picture property is Nothing aren't transparent / Picture プロパティがない Image コントロールは透明ではない
- **[PRB]** Common issues when converting VB6 applications that use type libraries / タイプライブラリを使用する VB6 アプリケーションを変換する際の共通的な課題
- **[PRB]** A reference to a COM type library is missing / COM タイプライブラリに対する参照が失われてしまう
- **[PRB]** Late-bound code might throw an exception or not work as expected / 遅延バインドコードが例外を起こしたり予想通りに動作しないかもしれない
- **[PRB]** DataGrid and MSFlexGrid controls don't work as expected and/or throw exceptions / DataGrid と MSFlexGrid コントロールは予想通りに動かなかったり、エラーを起こしたりする。
- **[PRB]** Changing properties of the data source associated with a DataGrid control can throw an exception / DataGrid コントロールに関連付くデータソースのプロパティを変更するとエラーが発生することがある
- **[PRB]** The VB.NET application throws a NullReference exception (or some other unexpected error) during the initialization phase / イニシャライゼーションの過程で Vb.NET アプリケーションが NullReference エラー（もしくは他の予期せぬエラー）を起こしてしまう
- **[PRB]** Code that references a VB6DataGrid column by its string key may not work in the migrated VB.NET program / 文字キーによって VB6DataGrid のカラムを参照するコードは変換後の VB.NET プログラムで動作しない可能性がある
- **[INFO]** Some properties of VB6DataGrid control can't be modified from inside Visual Studio's property window / VB6DataGrid コントロールのプロパティのいくつかは Visual Studio のプロパティウインドウから変更することができない
- **[INFO]** DBGrid control isn't supported / DBGrid コントロールはサポートされていません

- **[PRB]** VB.NET collections can't be modified while inside a For Each loop / VB.NET のコレクションは For Each Loop 内部では変更することができない。
- **[PRB]** Math operations on Variants cause an Overflow exception / バリエーションの数値演算がオーバーフロー例外を起こす
- **[HOWTO]** Move the declaration of a variable into a For or For Each loop / For や For Each Loop の中に変数の宣言を移動する
- **[HOWTO]** Implement functions that return arrays with nonzero LBound / 0 でない配列を返すファンクションを実装する
- **[PRB]** The DataGrid's SelColChange event may fire unexpectedly / DataGrid の SelColChange イベントは勝手に発生することがある
- **[PRB]** The VB.NET application doesn't compile because of a "Method XYZ not found" compilation error / 「メソッド XYZ が見つからない」というコンパイルエラーのため VB.NET アプリケーションがコンパイルできない
- **[PRB]** Data classes might need manual adjustments / Data クラスは手作業で調整する必要がある
- **[HOWTO]** Apply ByVal keyword to unassigned by-reference parameters / 署名されていない参照渡しのパラメータに ByVal キーワードを適用する
- **[HOWTO]** Reduce compilation errors caused by late binding or undetected conversions / 遅延バインドや検知されない移行によるコンパイルエラーを減らす
- **[INFO]** Using the VBMigrationPartner\_Support module / VBMigrationPartner\_Support モジュールの使用
- **[HOWTO]** Deal with Null values in If statements / If 文で NULL を取り扱う
- **[HOWTO]** Handle compilation errors caused by name collisions / 名前の衝突によるコンパイルエラーを取り扱う
- **[INFO]** VB.NET doesn't support methods with same name as events / VB.NET はイベントと同じ名前のメソッドをサポートしない
- **[PRB]** DAO and RDO Data control may not update bound controls when record position changes / DAO と RDO のデータコントロールはレコードの場所が変わったときにバインドコントロールを更新しない可能性がある。
- **[PRB]** The migrated VB.NET has a "Name 'xyz' is not declared" compilation error / 変換後の VB.NET に「『xyz』は宣言されていません」というコンパイルエラーが出ている。
- **[HOWTO]** Exclude portions of VB6 code from the migration process and replace it with custom VB.NET statements / 変換プロセスから VB6 コードの一部を除外して任意の VB.NET ステートメントに置き換える
- **[HOWTO]** Show UPGRADE\_ISSUE comments in the Task List window / タスクリストウインドウに UPGRADE\_ISSUE を表示する
- **[HOWTO]** Ignore fatal runtime errors / 致命的な実行時エラーを無視する
- **[INFO]** ADO-related code in the migrated VB.NET application might not behave like the original VB6 code / 変換後の VB.NET アプリケーションの ADO に関連するコードは元の VB6 のコードのように動作しない可能性がある
- **[BUG]** The Validate event may not fire for the Data and RDO Data control / データコントロールと RDO データコントロールについては Validate イベントは発生しない可能性がある
- **[PRB]** Assigning the ActiveConnection and Source properties of an ADODB.Recordset object may cause a runtime error / ADODB.Recordset オブジェクトの ActiveConnection プロパティと Source プロパティへの代入はランタイムエラーの原因となることがある
- **[INFO]** Classes and user controls generated by VB6 Data Object Wizard aren't supported / VB6 のデータオブジェクトウィザードによって生成されたクラスとユーザーコントロールはサポートされない
- **[PRB]** Passing an ADO read-only property to a ByRef argument causes a COMException / ByRef 引数に ADO の読み取り専用プロパティを渡すと COMException エラーが発生する
- **[HOWTO]** Solve syntax errors in ADODC event handlers / ADODC イベントハンドラの構文エラーを解決する

- **[INFO]** Multithreaded VB6 applications don't retain their multithreading behavior / マルチスレッドの VB6 アプリケーションはマルチスレッド動作を保持しません。
- **[PRB]** GoSub statements inside For loops and With blocks cause a compilation error / For Loop と With ブロック内部の GoSub 文がコンパイルエラーを起こす
- **[HOWTO]** Work with auto-instantiating arrays / 自動インスタンス化配列を伴った動作をさせる
- **[INFO]** Properties may be migrated without any warning related to VB6's ByRef parameters being converted to VB.NET's ByVal parameters / VB.NET の ByVal パラメータに変換される VB6 の ByRef パラメータに関する警告なしでプロパティが変換される可能性がある
- **[INFO]** adDBFileTime enum values are converted to ADODB\_DataTypeEnum\_adDBFileTime constants / adDBFileTime 列挙型は ADODB\_DataTypeEnum\_adDBFileTime 定数に変換される
- **[PRB]** Structures with same name may be generated in the same VB.NET namespace / 同じ名前の構造体が VB.NET の同じ名前空間に生成されるかもしれない
- **[PRB]** Objects returned from COM methods may cause an InvalidCastException error / COM メソッドが返すオブジェクトが InvalidCastException エラーの原因となる可能性がある
- **[PRB]** Passing ADOX properties to ByRef argument may cause unexpected runtime exceptions / ByRef 引数に対して ADOX プロパティを渡すと予期せぬランタイムエラーが発生する可能性がある
- **[HOWTO]** Create an empty directory under the application's folder / アプリケーションフォルダに空のディレクトリを作成する
- **[PRB]** The DataBindings property of a bound user control isn't migrated / バインドされたユーザーコントロールの DataBindings プロパティが変換されない
- **[PRB]** The ScriptControl can't interact with classes that aren't visible to COM / ScriptControl は COM からは見えないクラスとは相互作用することができない
- **[HOWTO]** Avoid compilation errors caused by unsupported designers / サポートされないデザイナーを使ってコンパイルエラーを回避する
- **[PRB]** The VB.NET application doesn't compile because of a "Number of indices exceeds the number of dimensions of the indexed array" compilation error / 「索引数が索引化された配列の次元数を超過しました」というコンパイルエラーのため VB.NET アプリケーションがコンパイルできない
- **[PRB]** The font of a control changes unexpectedly at runtime / コントロールのフォントが実行時に勝手に変わってしまう
- **[PRB]** MDI child forms don't become visible / MDI 子フォームが可視化しない
- **[PRB]** VB.NET forms don't display the size grip handle / VB.NET フォームがサイズグリップを表示しない
- **[PRB]** Arrays of menu items can throw an exception if items are destroyed too early / メニューアイテムの破棄が早すぎると、配列がエラーを起こしてしまう
- **[PRB]** Setting the TabIndex property in VB.NET doesn't affect the TabIndex of other controls / VB.NET では TabIndex プロパティを設定しても他のコントロールの TabIndex が変更されない
- **[PRB]** Using the hDC property causes an InvalidOperationException error / hDC プロパティを使用すると InvalidOperationException エラーが発生してしまう
- **[PRB]** Assignments to VB6 Form's Left and Top properties are ignored / VB6 Form の Left プロパティと Top プロパティへの代入が無視される
- **[HOWTO]** Iterate over all forms of current application / アプリケーションのフォームすべてに対して繰り返し操作を行う

- **[HOWTO]** Speed up For Each loops that iterate over the Controls collection／コントロールコレクションに対して繰り返し用いられる For Each ループを高速化する
- **[HOWTO]** Determine the font used for forms and controls／フォームとコントロールで使用されるフォントを決める
- **[PRB]** Forms with same name as one of their controls can't be opened in Visual Studio designer／フォーム上のコントロールと同じ名称を持つフォームを Visual Studio のデザイナーで表示できない
- **[PRB]** Graphic output is cleared when the form becomes visible／フォームが可視化されたときに画像出力がクリアされてしまう
- **[PRB]** A form behaves differently the second time it is opened／2 回目に表示する際にフォームが異なる動作をしてしまう
- **[INFO]** Arrays of menu controls can't contain separators／メニューコントロールは分離記号を含められない
- **[PRB]** The VB6 form has border, but the migrated VB.NET form hasn't／VB6 フォームは境界線を持っているが、変換された VB.NET フォームは持っていない
- **[INFO]** VB.NET automatically sets the Checked property of the first control in a group of Option buttons／VB.NET はオプションボタングループの最初のコントロールの Checked プロパティを自動的に設定する
- **[HOWTO]** Speed up execution by removing unnecessary Refresh methods／不要な Refresh メソッドを削除して高速化する
- **[PRB]** Replacing a VB6Placeholder control with another control causes an error in the VB.NET designer／VB.NET のデザイナー上で VB6Placeholder コントロールを他のコントロールに置き換えるとエラーが発生する
- **[PRB]** Transparent VB6 Label controls aren't transparent in the VB.NET application／透明な VB6 ラベルコントロールは VB.NET アプリケーションでは透明ではない
- **[INFO]** Timers are disabled inside VB6 IDE when a MsgBox or InputBox statement is active／MsgBox 文や InputBox 文がアクティブなとき VB6 の IDE ではタイマーは無効である
- **[INFO]** Double-clicking on a control array element at design-time creates a brand new event handler／デザイナー上のコントロール配列の要素をダブルクリックすると新しいイベントハンドラが作成される
- **[PRB]** Disabled controls and read-only TextBox controls appears with grayed background／無効なコントロールと読み取り専用の TextBox コントロールの背景が灰色となってしまう
- **[PRB]** Images displayed in TreeView, ListView, and other controls aren't transparent／TreeView コントロールや ListView コントロールなどで表示されるイメージは透明ではない
- **[PRB]** Controls in migrated VB.NET forms may be hidden by other controls／変換後の VB.NET フォーム上のコントロールが他のコントロールによって隠れてしまう可能性がある
- **[PRB]** Shape and Line controls aren't updated correctly／Shape コントロールと Line コントロールが正常に更新されない
- **[PRB]** PictureBox controls can be cleared unexpectedly or receive spurious Paint events／PictureBox コントロールが勝手にクリアされたり、誤った Paint イベントを受け取ってしまう可能性がある
- **[INFO]** User-defined coordinate systems can have unexpected results／ユーザーによってコーディネートされたシステムは予期しない結果をもたらすことがある
- **[INFO]** Forms with Line and Shape controls may exhibit a lot of flickering／Line コントロールと Shape コントロールを持つフォームに多数のチラツキが見られる可能性がある
- **[PRB]** A call to a Windows API Declare returns a string filled with spaces／Windows API の呼び出しがスペースのみの文字列を返す

- **[PRB]** VB6 applications that use window subclassing or other API callback methods throw a `CallbackOnCollectedDelegate` exception / サブクラスウインドウや他の API コールバックメソッドを使う VB6 アプリケーションが `CallbackOnCollectedDelegate` 例外を起こす
- **[INFO]** The `cdIOFNShareAware` flag of the `CommonDialog` control isn't supported / コモンダイアログコントロールの `cdIOFNShareAware` フラグはサポートされない
- **[PRB]** Using an array of UDTs causes a compilation error or doesn't work correctly at runtime / ユーザー定義型の配列を使用するとコンパイルエラーを引き起こすか、実行時に正常に動作しなくなる
- **[PRB]** The system hangs when using `SendKeys` to simulate key presses in an external application / 外部アプリケーションのキープレスをシミュレートする `SendKeys` を使用する場合、システムがハングアップする
- **[PRB]** `Get#` and `Put#` statements might not behave as expected / `Get#` と `Put#` は想定取りの動作を行わない可能性がある
- **[PRB]** Passing an enumerated value to a Boolean argument throws an exception when the method returns / メソッドが値を戻すときに列挙型からブーリアン引数に値が渡されるとエラーが発生する
- **[PRB]** The migrated VB.NET application can't compile because of the error " 'XYZ' is already declared in this enum" / 「XYZ はすでにこの列挙型に宣言されています」というエラーのため変換された VB.NET アプリケーションがコンパイルできない
- **[HOWTO]** Simplify code produced for `UserControl` classes / ユーザーコントロールクラス用に生成されたコードを簡素化する
- **[PRB]** Property references in a form might not match property names in a `UserControl` / フォーム上で参照を行うプロパティがユーザーコントロールのプロパティ名と一致しないことがある
- **[PRB]** Properties of the `MSChart` control might not be migrated correctly / `MSChart` コントロールのプロパティは正常に変換されない可能性がある
- **[INFO]** `TabStrip` and `SSTab` controls don't display hotkeys in button captions / `TabStrip` コントロールと `SSTab` コントロールはボタンの見出しにホットキーを表示しない
- **[BUG]** The `Scroll` event of the `VB6FlatScrollBar` control doesn't fire repeatedly while dragging the indicator / マウスカーソルをドラッグしても `VB6FlatScrollBar` コントロールの `Scroll` イベントが連続的に発生しない
- **[BUG]** The `MouseIcon` isn't migrated for `MMControl` and `MSHFlexGrid` controls / `MMControl` コントロールと `MSHFlexGrid` コントロールについては `MouseIcon` は変換されない
- **[PRB]** The `UpDown` control can affect the wrong property of its buddy control / `UpDown` コントロールは対応するコントロールのプロパティに誤った値をセットすることがある
- **[PRB]** Removing an item from a multi-selectable `ListBox` resets `ListIndex` property / 複数選択可能なリストボックスからアイテムを削除すると `ListIndex` プロパティがリセットされてしまう
- **[BUG]** The `VB6Printer` object doesn't print in landscape mode / `VB6Printer` オブジェクトが横置き向きで印刷しない
- **[INFO]** Unsupported ActiveX controls / サポートされていない ActiveX コントロール
- **[PRB]** The `ListView`'s background image doesn't migrate correctly / `ListView` の背景イメージは正常に変換されない
- **[PRB]** Deleting a `ToolBar`, `ListView`, or `StatusBar` control prevents the form from loading in Visual Studio designer / `ToolBar` コントロールや `ListView` コントロールや `StatusBar` コントロールを削除するとフォームを Visual Studio のデザイナーに表示することができなくなる
- **[INFO]** The `.NET` `ToolBar` control can't contain standard controls / `.NET` の `ToolBar` コントロールは標準コントロールを含めることができない
- **[INFO]** Declare statements can generate methods with different names / `Declare` 文は異なる名称のメソッドを生成することがある

- **[PRB]** SelFontName, SelFontSize, SelBold, SelItalic, SelUnderline, and SelStrikeThru properties of RichTextBox control return Nothing instead of Null / RichTextBox コントロールの SelFontName プロパティと SelFontSize プロパティと SelBold プロパティと SelItalic プロパティと SelUnderline プロパティと SelStrikeThru プロパティが Null ではなく Nothing を返してしまう。
- **[PRB]** Invalid assignments to the Rtf property might not throw an exception / Rtf プロパティに無効な代入を行ってもエラーが発生しない可能性がある
- **[HOWTO]** Migrate VB6 applications that use window subclassing / サブクラスウインドウを使用する VB6 アプリケーションを変換する
- **[PRB]** Output from the Print statement appears slightly different from the original VB6 application / Print 文の出力結果が元の VB6 アプリケーションとは若干異なっているように見える
- **[INFO]** DIB images are copied and pasted as regular images / DIB イメージは標準イメージとしてコピー & ペーストされる
- **[PRB]** The DTPicker control doesn't accept dates earlier than 1/1/1753 / DTPicker コントロールが 1753 年 1 月 1 日より前の日付を受け付けない

## [HOWTO] Modify migrated projects to use CodeArchitects.VBPowerPack.dll CodeArchitects.VBPowerPack.dll を使用するように変換済みプロジェクトを修正する

Starting with version 1.34, VB Migration Partner can migrate the Printer, CommonDialog, Line and Shape classes by means of the new CodeArchitects.VBPowerPack.dll library, which has the advantage of not depending on any COM DLL.

バージョン 1.34 以降の VBMP は新しい CodeArchitects.VBPowerPack.dll ライブラリを使用して Printer クラス、CommonDialog クラス、Line クラス、そして Shape クラスを変換できます。新しいライブラリには COM DLL に依存しないという利点があります。

This article explains how you can modify projects that are already migrated to leverage the new DLL and therefore avoid (or at least, reduce) dependencies from COM. Basically, this process has the following steps.

[この記事では新しい DLL を使用するように変換済みプロジェクトを修正し、COM 依存を回避する(少なくとも減らす)方法を説明します。この工程は以下のようになります。

1. Add a reference to the CodeArchitects.VBPowerPack.dll and Microsoft.VisualBasic.PowerPacks.Vs.dll files.

CodeArchitects.VBPowerPack.dll ファイルと Microsoft.VisualBasic.PowerPacks.Vs.dll ファイルに対する参照を追加してください

2. add a project-level import for the **CodeArchitects.VBPowerPack** namespace.

プロジェクトに **CodeArchitects.VBPowerPack** namespace に対する import を追加してください

3. modify the VisualBasic6\_Support.vb file (located in the My Project and hidden by default) so that all reference to VB6Printer and VB6Printers classes is changed into **VB6PrinterPP** and **VB6PrintersPP**, respectively.

(My Project にあり、デフォルトでは非表示となっている) VisualBasic6\_Support.vb ファイルを修正して、VB6Printer クラスと VB6Printers クラスに対する参照をすべて **VB6PrinterPP** クラスと **VB6PrintersPP** クラスに変更してください。

4. check the Error List window and look whether your code uses one or more members that have been obsoleted. In this case, your code is guaranteed not to work as it did before the changes.

エラーリストウインドウを確認し、ソースコードが一つ以上の古いメンバーを使用していないかどうか確認してください。この場合、ソースコードが変更される前のように動作するという確証はありません。

5. Carefully read [this article](#) to check whether other incompatibilities or important differences can affect the way your code behaves.

非互換性や重要な差異がソースコードの動きに影響を与えるかどうかを確認するために[こちらの記事](#)を注意深く読んでください。



## Improve code generation with Variant variables and parameters / バリエーション変数とパラメータを使用するコード生成を改善する

A VB6 Variant can contain either a scalar value (strings, numbers, dates, etc.) or an object. When converting variant expressions to VB.NET, VB Migration Partner takes a prudent approach and assumes that the variant might contain an object. This detail often causes the generation of unnecessarily verbose VB.NET code if a `DefaultMemberSupport` pragma has been emitted, or prevents VB Migration Partner from correctly generating a reference to the default property of an object.

VB6 のバリエーション変数はスカラー値 (文字列、数値、日付など) またはオブジェクトを保持することができます。バリエーション式を VB.NET に変換する際、VB Migration Partner はバリエーション変数がオブジェクトを保持する可能性があるとして仮定し、慎重なアプローチを採用します。これによって、`DefaultMemberSupport` が発行されていた場合、不要に冗長な VB.NET コードが生成されたり、VB Migration Partner がオブジェクトのデフォルトプロパティに対する参照を正しく生成できないことがあります。

Consider the following example:

下記のコードについて考察してください。

```
' ## DefaultMemberSupport True
Sub Test (ByVal tb As TextBox, ByVal name As Variant)
    name = tb
End Sub
```

Here's the converted VB.NET code:

下記のコードが変換後の VB.NET コードです。

```
Sub Test (ByVal tb As VB6TextBox, ByVal name As Variant)
    SetDefaultMember6 (name, tb.Text)
End Sub
```

The above code perfectly reproduces the VB6 behavior, but it is unnecessarily verbose. In fact, it is clear that the VB6 developer meant `name` to be a scalar value (a string, to be precise), but VB Migration Partner can't infer this fact. You can generate simpler code by letting VB Migration Partner that the Variant parameter is meant to contain a scalar value, as follows:

上記のコードは VB6 の動作を完璧に再現していますが、不要に冗長的です。実際に、VB6 コードの作成者が `name` をスカラー値 (正確には文字) として扱っていたことは明白なのですが、VB Migration Partner はこの事実を予想することができません。ユーザーは下記のようにして、VB Migration Partner にバリエーション変数がスカラー値を保持するということを考慮させることによって簡潔なコードを生成することができます。

```
' ## DefaultMemberSupport True
Sub Test (ByVal tb As TextBox, ByVal name As Variant)
' ## name.AssumeType String
    name = tb
End Sub
```

which causes this code to be generated:

上記のようにすることで、下記のコードが生成されます。

```
Sub Test(ByVal tb As VB6TextBox, ByVal name As Variant)
    name = tb.Text
End Sub
```

The AssumeType pragma is also useful to force VB Migration Partner to generate a default property when calling a method. Unlike direct assignments – where the Set keyword is used to tell whether you are assigning a scalar value or an object – the action of passing a value to a method parameter is quite ambiguous under VB6. Consider this code:

メソッドを呼び出している場合、AssumeType プラグマも VB Migration Partner にデフォルトプロパティを生成させるのに役に立ちます。直接的な代入（ユーザーがスカラー値を代入しているのかオブジェクトを代入しているのか見分けるために Set キーワードが使用される箇所）と違って、メソッドパラメータに値を渡すという行為は VB6 においては非常に曖昧です。下記のコードについて考察してください。

```
Sub Test(ByVal rs As ADODB.Recordset)
    ShowFieldValue rs("FirstName")
End Sub
```

```
Sub ShowFieldValue(ByVal fieldValue As Variant)
    MsgBox fieldValue
End Sub
```

This is how the code is translated to VB.NET:

上記のコードは次のように VB.NET に変換されます。

```
Public Sub Test(ByVal rs As ADODB.Recordset)
    ShowFieldValue(rs.Fields("FirstName"))
End Sub
```

```
Public Sub ShowFieldValue(ByVal fieldValue As Object)
    MsgBox6(fieldValue)
End Sub
```

Here's the problem: VB Migration Partner fails to append the name of the default member of the ADODB.Field object (i.e. Value) in the call to the ShowFieldValue method. The reason is that the fieldValue parameter is defined as a Variant, therefore it might receive either a scalar or an object. You can help VB Migration Partner to generate better code by specifying that the parameter is always a scalar, as follows:

これには次のような問題があります。VB Migration Partner は ShowFieldValue メソッドを呼び出す際に、ADODB.Field オブジェクトのデフォルトメンバー（例えば、Value）の名前を追加することができません。その理由は、fieldValue パラメータがバリエーションとして定義されているため、スカラー値でもオブジェクトでも受け取ることができるからです。ユーザーは次のように、パラメータが常にスカラー値であると指定することによって、より適切なコードを VB Migration Partner に生成させることができます。

```
Sub ShowFieldValue(ByVal fieldValue As Variant)
    ## fieldValue.AssumeType Single
```

```
MsgBox fieldValue  
End Sub
```

Notice that the pragma argument is Single, but actually any scalar data type works. However Single works better than other types because it never forces conversions (as it happens with Double values when they are assigned a Date, or viceversa).

プラグマの引数は Single ですが、実際にはどのようなスカラーのデータ型であろうと動作することを覚えてください。しかし、Single は (Double 値が Date 値を代入されたり、その逆の場合に起こるような) コンバージョンを強制しないので、他のデータ型と比べて適切に動作します。

## [INFO] Controls in the support library can't be dropped on a standard VB.NET form / サポートライブラリのコントロールは VB.NET 標準フォームに配置できない

The controls in VB Migration Partner's support libraries are meant to ease the migration of existing VB6 forms and can operate correctly only if placed on the surface of a **VB6Form** object. Any attempt to drop such controls onto a regular VB.NET form causes either a design-time or a runtime error. This behavior is by default and shouldn't be considered as a VB Migration Partner's limitation.

VB Migration Partner のサポートライブラリのコントロールは既存の VB6 フォームの移行を簡素化するために作られており、**VB6Form** オブジェクト上に配置された場合でのみ正常に動作することができます。標準の VB.NET フォームにこれらのコントロールを配置しようとするのはデザイナーのエラーやランタイムエラーの原因となります。この動作はデフォルトであり、VB Migration Partner の制約とは見なされません。

Notice that the opposite isn't true: you can always place standard .NET controls onto a VB6Form, even though some limitations might apply. For example, these controls can't be part of "old-styled" control arrays or be instantiated dynamically with the Controls.Add method.

しかし、逆はそうではありません。ユーザーはいつでも VB6Form の上に .NET 標準コントロールを置くことができます。たとえ、いくつかの制約があったとしてもです。例えば、これらのコントロールは「古い形式の」コントロール配列になることができませんし、Controls.Add メソッドを使って動的にインスタンス化することもできません。

We recommend that you don't add new VB6Form instances to the project because such freshly-created forms don't contain all the support code that VB Migration Partner creates when migrating a VB6 form. If you absolutely need to define a new form that uses one or more controls defined in the support library, the only viable approach is adding the form to the VB6 application and then migrate it again to VB.NET.

弊社は、VB6 フォームを変換する際、プロジェクトに新しい VB6 フォームインスタンスを追加しないことを推奨しています。理由は、そのような新しく作成されたフォームが VB Migration Partner によって作成されたコードを全く含まないためです。万が一、サポートライブラリに定義されたコントロールを使用する新しいフォームを定義する必要がある場合の唯一の方法は、VB6 アプリケーションにフォームを追加し、VB.NET に再度変換することです。

More in general, we recommend that you never write code or new forms that use the objects and the controls defined in the CodeArchitects.VBLibrary. Using these "legacy" objects deprives you of all the benefits of the .NET Framework platform.

普通は、CodeArchitects.VBLibrary に定義されたコントロールやオブジェクトを使用する新しいフォームやコードを作成することは決してお勧めいたしません。これらの「レガシーな」オブジェクトを使うことは .NET Framework というプラットフォームの利点を失うこととなります。

**UPDATE:** starting with version 1.34, it is possible to drop controls contained in the VBLibrary onto a standard .NET form, albeit with a few limitations. For more information, read [this article](#).

**更新情報:** バージョン 1.34 から、いくつかの制約はあるものの、.NET 標準フォームに VBLibrary のコントロールを置くことが可能になりました。詳しい情報は[こちらの記事](#)をご覧ください。

## [HOWTO] Use controls from VB Migration Partner's support library on a standard .NET form /.NET 標準フォーム上で VB Migration Partner のサポートライブラリのコントロールを使用する

Starting with version 1.34 it is possible to drop a control from the VB Migration Partner library - e.g. VB6TextBox or VB6ListBox - on a standard .NET form, that is a form that inherits directly from the System.Windows.Forms.Form class.

バージョン 1.34 から、System.Windows.Forms.Form クラスを直接継承している .NET 標準フォームの上に VB Migration Partner ライブラリのコントロール—例 VB6TextBox、VB6ListBox—を配置することができるようになりました。

Such new capability, however, isn't officially supported and we recommend that you adopt it only if strictly necessary. More specifically, not all the features of the original VB6xxxx class are supported, for example data binding and graphic methods.

しかし、この新機能は公式にサポートされたわけではなく、弊社としてはどうしてもそうする必要のある場合に限ってのみ、そうすることをお勧めしています。さらに厳密に言えば、VB6xxx クラスの機能のすべてがサポートされているというわけでもありません。例えば、データバインディングやグラフィックメソッドはサポートされません。

If you are aware of this limitations and still proceed with using one of the VB Migration Partner controls on a standard .NET form, the Code Architects support team will help you in solving issues that should arise, however we can't guarantee that all these issues can be successfully solved.

この制約を考慮しつつ .NET 標準フォームの上で VB Migration Partner のコントロールを使うのであれば、コードアーキテツ社のサポートチームは問題解決のお手伝いをすることができますが、すべての問題がうまく解決することを保証することはできません。

**An important warning:** If you are dropping a VB Migration Partner control on the startup form of your application, it is mandatory that you initialize the support library before the control is used and the form is instantiated. This is achieved by adding a call to the InitializeLibrary method from inside the form's static constructor:

**重要な警告:** アプリケーションのスタートアップフォームに VB Migration Partner のコントロールを配置する場合、コントロールが使用される前やインスタンス化される前にサポートライブラリをイニシャライズする必要があります。これはフォームの静的コンストラクタの内部で InitializeLibrary メソッドを呼び出すようにすることで可能となります。

```
Shared Sub New()  
    EnsureVB6LibraryInitialization()  
End Sub
```

## [HOWTO] Leverage .NET TextAlign property with command, checkbox, and option button controls ／Command コントロール、CheckBox コントロール、OptionButton コントロールで.NET の TextAlign プロパティを利用する

By default, the VB6 Command, CheckBox and OptionButton controls offer limited control on how the text can be aligned. On the other hand, the corresponding .NET controls expose the **TextAlign** property, which allows you to precisely define where the text is displayed (top, middle, bottom, left, center, right, and all possible combinations of these values).

デフォルトでは、VB6 の Command コントロール、CheckBox コントロール、OptionButton コントロールは、テキストのアライメント方法が制限されたコントロールです。一方、対応する.NET コントロールは **TextAlign** プロパティを公開しており、テキストの表示位置 (top、middle、bottom、left、center、right とこれらの組み合わせ) を正確に定義することができます。

By default, the controls that come with VB Migration Partner perfectly mimic the VB6 behavior, which means that – apparently – you can't take advantage of the more flexible options that are available under .NET Framework.

デフォルトでは、VB Migration Partner のコントロールは VB6 の動作完全に複製しています。しかし、これは—外見上—.NET Framework において使用可能なより柔軟性の高いオプションを利用することができないということを意味します。

Starting with version 1.34, the VB6Command, VB6CheckBox and VB6OptionButton controls (as well as their windowless counterparts: VB6WLCommand, VB6WLCheck, and VB6WLOption) expose a new property, named **UseTextAlignment**.

バージョン 1.34 から、VB6Command コントロール、VB6CheckBox コントロール、VB6OptionButton コントロール (ウインドウを持たない VB6WLCommand、VB6WLCheck、VB6WLOption も同様です) は **UseTextAlignment** という新しいプロパティを公開しました。

You can set this property to True in the Properties window to override the default behavior of these control, and then set the TextAlign property to move the control's Text exactly where you want it to appear.

ユーザーはプロパティウインドウのこのプロパティを True にすることで、これらのコントロールのデフォルト動作を無効にして、TextAlign プロパティを設定することでコントロールのテキストを正確に自分の表示したい場所に移動することができます。

## [INFO] Minor differences between CommonDialog, Printer, Line, and Shape VB6 classes and their implementation in CodeArchitects.VBPowerPack.dll / CommonDialog、Printer、Line、そして Shape といった VB6 クラスと CodeArchitects.VBPowerPack.dll が提供するクラスの小さな違い

Starting with version 1.34, VB Migration Partner migrates the CommonDialog, Printer, Line, and Shapes classes to VB6CommonDialogPP, VB6PrinterPP, VB6LinePP, VB6ShapePP, and VB6ShapeOvalPP, all of which are defined in the CodeArchitects.VBPowerPack.dll. This article outlines the minor differences between these classes and the original VB6 objects.

バージョン 1.34 から VBMP は CommonDialog クラス、Printer クラス、Line クラス、そして Shapes クラスを VB6CommonDialogPP クラス、VB6PrinterPP クラス、VB6LinePP クラス、VB6ShapePP クラス、そして VB6ShapeOvalPP クラスに変換します。これらすべては CodeArchitects.VBPowerPack.dll に定義されています。この記事ではこれらのクラスと元の VB6 オブジェクトの小さな違いの要点が記述されています。

### VBPrinterPP

- the DrawMode property always returns the value 13-vbCopyPen  
DrawMode プロパティは常に 13-vbCopyPen を戻り値とします
- the DrawStyle property doesn't support the values 5-vbInvisible and 6-vbSolidInside  
DrawStyle プロパティは 5-vbInvisible と 6-vbSolidInside をサポートしません
- the DriverName and Port properties always return an empty string  
DriverName と Port プロパティは常に空文字を戻り値とします
- the hDC property always throw an exception  
hDC プロパティは常に例外を発生させます
- the TrackDefault property is False by default (it is True under VB6); assigning True to this property works as expected, but adds a noticeable overhead.  
TrackDefault プロパティの初期値は (VB6 では True ですが) False です。すなわち、プロパティに本来あるべき動作をさせるには True を割り当ててののですが、著しいオーバーヘッドが発生します。
- the Zoom property always returns the value 0  
Zoom プロパティは常に 0 を戻り値とします
- the PaintPicture method only supports the value vbSrcCopy for the *opcode* argument  
PaintPicture メソッドは *opcode* 変数に対しては vbSrcCopy のみサポートします。

Assigning an unsupported value to the DrawMode, Zoom, DriverName, or Port properties throws an exception (if VB6Config.ThrowOnUnsupportedMember is true) or is ignored.

DrawMode や Zoom や DriverName や Port プロパティにサポートされない値を割り当てると例外が発生するか (VB6Config.ThrowOnUnsupportedMember が True の場合は) 無視されます。

### VB6CommonDialogPP (when displaying the Print dialog / プリントダイアログを表示する場合)

- the HDC property always returns the value 0  
HDC プロパティは常に 0 を戻り値とします

## VB6LinePP

- DrawMode property only supports the value 1-vbBlackness  
DrawMode プロパティは 1-vbBlackness のみサポートします

## VB6ShapePP and VB6ShapeOvalPP / VB6ShapePP と VB6ShapeOvalPP

Square and rectangular shapes map to the VB6ShapePP class, whereas circular and oval shapes map to the VB6ShapePP classes. This arrangement has two implications:

正方形と長方形が VB6ShapePP クラスに割り当てられたのに対して、円と楕円は VB6ShapeOvalPP クラスに割り当てられました。この変更は二つの意味を持っています。すなわち、

- You can't change the Action property in a way that would require a change of the mapped class. For example, changing a square shape into an oval shape (or vice versa) isn't supported. However, changing from square to rectangular (or vice versa) and changing from circular to oval (and vice versa) is supported.

割り当てられたクラスの変更が必要とされるという意味で、Action プロパティを変更することはできません。例えば、正方形を楕円に変更することはサポートされていません。しかし、正方形を長方形に（またはその逆）と円から楕円（またはその反対）はサポートされます。

- You can create a control array containing both square/rectangular and circular/oval shapes at the same time.

正方形と長方形、円と楕円を同時に含むコントロール配列を作成することができます。



[INFO] VB Migration Partner uses the Line, Shape, and Printer classes in Microsoft VB PowerPacks/VB Migration Partner は Microsoft VB PowerPacks の Line クラスと Shape クラスと Printer クラスを使用します

All versions of VB Migration Partner up to 1.33 implement the Printer, Line, and Shape VB6 classes by means of custom classes defined in the support library. More precisely:

バージョン 1.33 以下の VB Migration Partner はサポートライブラリに定義されたカスタムクラスを使って Printer クラスと Line クラスと Shape クラスを実装します。より正確に言えば、

- the Printer class is implemented by means of the VB6Printer class, which is a wrapper for a COM object exposed by VBSuportLib.dll - this DLL is written in VB6 and therefore requires the VB6 runtime  
Printer クラスは VB6Printer クラスによって実装されます。VB6Printer クラスは VBSuportLib.dll によって公開されている COM オブジェクトのラッパーです。—この DLL は VB6 で作成されているので VB6 ランタイムを必要とします。
- the CommonDialog control is implemented by means of a class named VB6CommonDialog; when the ShowPrint method is invoked, this class instantiates a COM object, also exposed by VBSupport.dll.  
CommonDialog コントロールは VB6CommonDialog という名前のクラスによって実装されます。すなわち、ShowPrint メソッドが呼び出されると、このクラスは、同様に VBSupport.dll によって公開されている COM オブジェクトをインスタンス化します。
- the Line and Shape controls are implemented by means of the VB6Line and VB6Shape classes. At design time, these classes appear as components in the form's tray area, which makes it harder to move them after the migration has completed.  
Line コントロールと Shape コントロールは VB6Line クラスと VB6Shape によって実装されます。デザイナーではこれらのクラスはフォームトレイにコンポーネントとして表示されておりますが、変換終了後にそれらを移動するのは困難です。

Starting with version 1.34, the implementation of these three classes has been moved to another support library, named CodeArchitects.VBPowerPack.dll:

バージョン 1.34 から、これら 3 つのクラスの実装は CodeArchitects.VBPowerPack.dll という別のサポートライブラリに移されました。

- the Printer class is implemented by means of the **VB6PrinterPP** class, which in turn is a wrapper for the Printer class defined in Microsoft.VisualBasic.PowerPack.Vs.dll.  
Printer クラスは **VB6PrinterPP** クラスによって実装されます。また、VB6PrinterPP クラスは Microsoft.VisualBasic.PowerPack.Vs.dll に定義されている Printer クラスのラッパーです。
- the CommonDialog control is implemented by means of a class named **VB6CommonDialogPP**, which exclusively use standard .NET objects to implement all dialogs, including the Print dialog  
CommonDialog コントロールは **VB6CommonDialogPP** というクラスによって実装されます。  
VB6CommonDialogPP クラスはプリントダイアログを含むすべてのダイアログすべてを実装するための .NET 標準オブジェクトだけを使用します。
- the Line and Shape controls are implemented by means of the **VB6LinePP**, **VB6ShapePP** and **VB6ShapeOvalPP** classes, which work as wrappers for classes defined

in Microsoft.VisualBasic.PowerPack.Vs.dll.

Line コントロールと Shape コントロールは **VB6LinePP** クラスと **VB6ShapePP** クラスと **VB6ShapeOvalPP** クラスによって実装されます。これら (VB6LinePP、VB6ShapePP、VB6ShapeOvalPP) のクラスは Microsoft.VisualBasic.PowerPack.Vs.dll に定義されているクラスのラッパーとして動作します。

The new CodeArchitects.VBPowerPack.dll ensures that your migrated apps never depend on COM objects or the VB6 runtime to implement printing capabilities.

新しい CodeArchitects.VBPowerPack.dll によって、変換後のアプリケーションは COM オブジェクトや印刷機能を実装するための VB6 ランタイムに依存しなくなります。

However, it should be noted that – unlike the classes defined in VBSupportLib.dll – the classes provided in CodeArchitects.VBPowerPack.dll might not behave exactly as in VB6. For example, the distance between lines that are printed by the VB6PrinterPP object is slightly larger than the line distance produced by the VB6Printer. Also, the VB6PrinterPP object doesn't support a few Printer members, including DrawMode, DriverName, hDC, Port, and Zoom.

しかし、特筆すべきことは、—VBSupportLib.dll に定義されたクラスとは異なり—CodeArchitects.VBPowerPack.dll によって提供されるクラスは VB6 と正確に同じ動きをするとは限らないということです。例えば、VB6PrinterPP オブジェクトによって印刷された線と線と間の距離は VB6Printer を使った場合と比べると若干大きいのです。また、VB6PrinterPP はいくつかの Printer メンバ (DrawMode、DriverName、hDC、Port、Zoom) に対応していません。

For backward compatibility, therefore, VB Migration Partner continues to convert these classes as it did with previous versions. The new features must be explicitly enabled by the developer, in one of these two ways:

そのため、下位互換性を維持しつつ、これらのクラスを変換するためには以前のバージョンの VB Migration Partner で変換する必要があります。この新機能を使うには以下の二つの方法で明示的に有効化する必要があります。

- a. Use the **AddLibraryPath** pragma to point to the CodeArchitects.VBPowerPack.dll file, which by default is deployed in the C:\ProgramFiles\Code Architects\Code Architects VB Migration Partner\PowerPack folder. This technique allows you to decide whether to use the powerpack on a project-by-project basis. CodeArchitects.VBPowerPack.dll ファイルを指定する **AddLibraryPath** プラグマを使ってください。このファイルはデフォルトでは C:\ProgramFiles\Code Architects\Code Architects VB Migration Partner\PowerPack フォルダに格納されています。この方法を使えば、プロジェクトごとに **PowerPack** を使うかどうかを決めることができます。
- b. Copy the CodeArchitects.VBPowerPack.dll file from the PowerPack subfolder to the main VB Migration Partner's setup folder. In this way, all migrated projects will use the new features. PowerPack サブフォルダから VB Migration Partner のメインセットアップフォルダに CodeArchitects.VBPowerPack.dll ファイルをコピーしてください。この方法では、すべての変換済みプロジェクトに新機能が使用されます。

**Important note:** the CodeArchitects.VBPowerPack DLL has a reference to the Microsoft VB PowerPack library version 9.0. This DLL can be found in the PowerPack folder and must be distributed with all migrated projects that use any of the \*\*\*PP classes.

**重要事項:** CodeArchitects.VBPowerPack DLL は Microsoft VB PowerPack library version 9.0 を参照します。この DLL は PowerPack フォルダに入っており、xxxPP クラスを使用する変換後のプロジェクトすべてに配布される必要があります。

## It's impossible to set the initial tab in an SSTab control / SSTab コントロールにイニシャルタブをセットできない

Unlike the SSTab control, the .NET TabControl component doesn't support the ability to set at design time the page to be displayed when the form loads at runtime. For this reason, migrated .NET applications always display the first tab when a form appears, and it is necessary to manually assign the desired tab in the Form\_Load event handler:

SSTab コントロールと違って、.NET の TabControl コンポーネントはデザイナーで実行時にフォームロードが行われる際に表示されるページを設定することができません。このため、変換後の .NET アプリケーションは常にフォームが表示される際には最初のページが表示されます。また、表示したいタブがある場合は、Form\_Load イベントハンドラに記述する必要があります。

```
Private Sub Form_Load() Handles MyBase.Load
    ' display the second tab when the form appears
    SSTab1.Tab = 1
End Sub
```

A consequence of this limitation is that inserting the following statement in the Form\_Load method

この制約のため、Form\_Load メソッドに次のステートメントを記述します。

```
SSTab1.Tab = 0
```

fires neither the Click click nor .NET events such as OnSelecting or OnSelectedIndexChanged.

このコードによっては、Click イベントも OnSelecting や OnSelectedIndexChanged のような .NET のイベントも発生しません。

If your form relies on the correct firing of one of these events, you should change the current tab twice, as in:

もし、これらのイベントの一つを発生させなければならないフォームの場合、タブを 2 度変更するべきです。

```
SSTab1.Tab = 1
SSTab1.Tab = 0
```

However, it is crucial that these statements be located in the handler for the Activate event, rather than Form\_Load method, and that they be executed only the very first time the form is displayed. You can put the IsFirstActivateEvent boolean property to good use:

しかし、Form\_Load ではなく、Activate イベントのハンドラーにこれらのステートメントを記述することや、フォームが表示されるよりも早く実行されることは、重大な変更となります。そのような場合には IsFirstActivateEvent プロパティが役に立ちます。

```
Private Sub Form_Activate() Handles MyBase.Activate
    If IsFirstActivateEvent Then
        SSTab1.Tab = 1
        SSTab1.Tab = 0
    End If
```

' ...  
End Sub

[PRB] Forms containing an ImageList control don't display at design-time after updating the CodeArchitects.VBLibrary DLL. / CodeArchitects.VBLibrary DLL を更新した後、ImageList コントロールを含むフォームがデザイナーに表示されない

In all VB Migration Partner versions up to and including 1.34 ImageList controls were converted into VB6ImageList controls (defined in VB Migration Partner's support DLL), which in turn inherited from System.Windows.Forms.ImageList control. We later found that this arrangement prevented users from adding new images to the ImageList control at design-time, therefore in version 1.34 we changed the way this control is converted.

1.34 を含むそれ以下のバージョンの VB Migration Partner では ImageList コントロールは System.Windows.Forms.ImageList コントロールから継承された (VB Migration Partner サポート DLL に定義されている) VB6ImageList コントロールに変換されました。その後、私たちはこの変更がデザイナーで ImageList コントロールに新しいイメージを追加することを邪魔していることに気付いたため、バージョン 1.34 ではこのコントロールの変換方法を変更しました。

In version 1.34 and later versions, the ImageList control is translated to **two** different controls: an instance of the VB6ImageList control and an instance of the standard .NET ImageList control. For example, a VB6 control originally named *MyImageList* control is converted into the following controls:

1.34 とその後のバージョンでは、ImageList コントロールは二つの異なるコントロールに変換されます。すなわち、すべての VB6ImageList コントロールのインスタンスと .NET 標準の ImageList コントロールのインスタンスです。例えば、*MyImageList* と名付けられた VB6 コントロールは次のコントロールに変換されます。

- A CodeArchitects.VBLibrary.VB6ImageList control named *MyImageList*  
*MyImageList* という名前の CodeArchitects.VBLibrary.VB6ImageList コントロール
- A System.Windows.Forms.ImageList control named *MyImageList\_Control*  
*MyImageList\_Control* という名前の System.Windows.Forms.ImageList コントロール

The relationship between the two controls is established by assigning the *MyImageList\_Control* object to the **ImageList** property of the *MyImageList* control. After the migration, users can edit all the images held in the native ImageList control from the Properties window, as they would do with a standard .NET form.

二つのコントロールの関係は *MyImageList* コントロールの **ImageList** プロパティに *MyImageList\_Control* オブジェクトを代入することで成立しています。移行後、ユーザーは .NET 標準フォームで行うように、プロパティウインドウからネイティブの ImageList コントロールに保持されているイメージを編集することができます。

VB Migration Partner 1.34 and later versions automatically generates all the necessary statements in the code-behind portions of form classes, therefore in most cases you don't need to be aware of these behind-the-scenes implementation details, with **important one exception**: if you migrated your forms with VB Migration Partner 1.33 (or earlier version) and then you replace the support library with the one that comes with VB Migration Partner 1.34 (or later version), then your .NET forms can't be edited at design-time from inside Visual Studio.

VB Migration Partner 1.34 とそれ以降のバージョンは自動的にフォームクラスに必要なコードビハインドステートメントを生成しますので、これらのコードビハインド部分の実装の詳細を気にする必要はほとんどありません。ただし、重要な例外がひとつあります。それは、VB Migration Partner 1.33 (またはそれ以前のバージョン) を使ってフォームを変換し、サポートライブラリを

1.34(またはそれ以降のバージョン)のライブラリと置き換えた場合、.NET フォームは Visual Studio のデザイナーで編集できなくなってしまうということです。

The only workaround for this issue is by manually modifying the code-behind portions of forms that contain ImageList controls.

この問題の唯一の回避策は ImageList コントロールを含んだフォームのコードビハインド部分を手作業で修正することです。

1. Enable the “Show All” button in Solution Explorer window if necessary

必要に応じて、ソリューションエクスプローラーの Show All ボタンを有効にします。

2. Inside the Solution Explorer, click on the “+” to the left of the form in question; this action reveals a file named yourformname.Designer.vb. Double click on it to edit the form’s code-behind portion

ソリューションエクスプローラー内の、問題のフォームの左側の「+」をクリックします。この操作は、フォーム名称.Designer.vb というファイルを表示するものです。フォームの code-behind 部分を編集するにはこれをダブルクリックします。

3. Search for the statement that declares the MyImageList control; inserts the declaration of the MyImageList\_Control variable immediately after it:

MyImageList コントロールを宣言しているステートメントを探して、その直後に MyImageList\_Control 変数の宣言を挿入します。

```
Public WithEvents MyImageList As CodeArchitects.VBLibrary.VB6ImageList
Public WithEvents MyImageList_Control As System.Windows.Forms.ImageList
```

4. Search for the statement that creates an instance of the VB6ImageList control (inside the InitializeComponent method); inserts the statements that instantiates the .NET ImageList control immediately after it:

(InitializeComponent メソッドの中にある) VB6ImageList コントロールのインスタンスを生成しているステートメントを探して、その直後に.NET の ImageList コントロールをインスタンス化するステートメントを挿入します。

5. 

```
Me.MyImageList = New CodeArchitects.VBLibrary.VB6ImageList()
Me.MyImageList_Control = New System.Windows.Forms.ImageList()
```

6. Scroll the code inside InitializeComponent method, until you find the statements that assign the ImageStream and Size properties of the MyImageList control and change the control reference so that the properties of the MyImageList\_Control object are assigned instead:

MyImageList コントロールの ImageStream プロパティと Size プロパティを代入しているステートメントが見つかるまで、InitializeComponent メソッド内部のコードをスクロールし、代わりに MyImageList\_Control オブジェクトのプロパティが代入されるようにコントロールの参照を変更します。

7. **’ WAS:**

```
Me.MyImageList.ImageStream =
CType(resources.GetObject(“MyImageList.ImageStream”), _
```

9. System.Windows.Forms.ImageListStreamer)

```
Me.MyImageList.ImageSize = New System.Drawing.Size(32, 32)
```

- 11.

12. **’ BECOMES:**

13. `Me.MyImageList.ImageList = Me.MyImageList_Control`
14. `Me.MyImageList_Control.ImageStream =`
15. `CType(resources.GetObject("MyImageList.ImageStream"), _`
16. `System.Windows.Forms.ImageListStreamer)`  
`Me.MyImageList_Control.ImageSize = New System.Drawing.Size(32, 32)`
  
17. Save the file, close the form, and reloads it inside Visual Studio. The form should now display correctly at design-time.

Visual Studio でファイルを保存し、フォームを閉じて、再度読み込んでください。これで、フォームはデザイン時に正常に表示されます。



Assignments to the ListView.SelectedItem property are ignored if the control isn't visible / コントロールが不可視状態の場合、ListView.SelectedItem プロパティへの代入が無視される

The .NET ListView control differs from the VB6 ListView control for an important detail: assignments to the ListView.SelectedItem property are ignored if the control isn't visible. For this reason, assignments to this property that are performed before the form becomes visible -- for example, inside the Form\_Load event -- have no effect and leave the property set to Nothing:

.NET の ListView コントロールは VB6 の ListView コントロールとは次の重要な点が異なります。コントロールが不可視の場合、ListView.SelectedItem プロパティへの代入は無視されます。そのため、フォームが可視化される前(例えば、Form\_Load イベントの内部)に実行されるプロパティへの代入は無効となり、プロパティには Nothing がセットされます。

```
Private Sub Form_Load() Handles MyBase.Load
    ListView1.SelectedItem = ListView1.Items(1) ' <== this causes an exception
End Sub
```

This behavior often causes a NullReference exception later in the execution, and it isn't always easy to understand what the real cause of the exception is.

この動作はしばしば実行後に NullReference エラーを引き起こします。そして、例外の本当の原因を理解するのは簡単ではありません。

The ListView control in VB Migration Partner's support library inherits this behavior from the .NET control. However, to help developers to correctly diagnose the cause of spurious NullReference exceptions, the ListView.SelectedItem property throw an exception if the control is currently invisible. The exception message is:

VB Migration Partner のサポートライブラリの ListView コントロールは .NET のコントロールの動作を継承しています。しかし、仮の NullReference エラーの原因が診断できるように、ListView.SelectedItem プロパティはコントロールが不可視の場合に下記のエラーを発生させます。

Can't set the SelectedItem property at this time.

The simplest solution to this problem is moving the code that assign the SelectedItem to an event handler that runs when the form (and the control) is visible, for example in the Form\_Activate event handler.

この問題の最も簡単な対策は SelectedItem を代入するコードをフォームが可視化されているときに実行するイベントハンドラ(例えば、Form\_Activate イベントハンドラの中)に移動することです。

[PRB] Assigning a StdDataFormat object to the DataFormat property of an ADODB.Field causes runtime errors / ADODB.Field の DataFormat プロパティに StdDataFormat オブジェクトを代入するとランタイムエラーの原因となる

VB Migration Partner support custom formatting in bound controls. References to StdDataFormat objects are converted to **VB6StdDataFormat** objects and migrated VB.NET applications works like the original VB6 app. VB Migration Partner even support custom formatting by means of the Format and Unformat events of the StdDataFormat class.

VB Migration Partner はバインドコントロールのカスタム書式設定に対応しています。StdDataFormat オブジェクトへの参照は **VB6StdDataFormat** オブジェクトに変換されることで、変換後の VB.NET アプリケーションは VB6 アプリケーションのように動作するようになります。VB Migration Partner は StdDataFormat クラスの Format イベントと Unformat イベントを使ってカスタム書式設定にも対応しています。

Everything works well when the VB6StdDataFormat object is assigned to the DataFormat property of a control. However, if the VB6StdDataFormat object is assigned to the DataFormat property of an ADODB.Field object you get a runtime exception, because the DataFormat property is expected to receive a COM StdDataFormat object, not the .NET VB6StdDataFormat object.

VB6StdDataFormat オブジェクトがコントロールの DataFormat プロパティに代入される場合はすべて正常に動作しますが、VB6StdDataFormat オブジェクトが ADODB.Field オブジェクトの DataFormat プロパティに代入される場合、ランタイムエラーが発生します。理由は、DataFormat プロパティが .NET の VB6StdDataFormat オブジェクトではなく、COM の StdDataFormat が渡されることを想定しているからです。

You can fix this problem by wrapping the StdDataFormat object in a **ToStdDataFormat6** method, which is defined in the VisualBasic6\_Support.bas module. Alternatively, you can use a **ReplaceStatement** pragma to insert the method call during the migration process, as in this example:

この問題は **ToStdDataFormat6** メソッド—これは VisualBasic6\_Support.bas モジュールに定義されています—の StdDataFormat オブジェクトをラッピングすることで解決することができます。代わりに、次の例のように、**ReplaceStatement** プラグマを使って変換時にメソッドの呼び出しを挿入することもできます。

```
Dim std As New StdDataFormat
sdf.Type = fmtCustom
sdf.Format = "MM. dd"
' ## ReplaceStatement rs("OrderDate").DataFormat = ToStdDataFormat6(std)
Set rs("OrderDate").DataFormat = std
```

Notice that the ToStdDataFormat6 method requires a reference to the Microsoft.StdFormat DLL. You can this reference manually after the migration or, better, automatically during the migration by means of an **AddReference** pragma.

注意すべき点は ToStdDataFormat6 メソッドが Microsoft.StdFormat DLL への参照を必要とする点です。変換後、手作業で参照設定を行うことができますが、むしろ、**AddReference** プラグマを使って変換中に自動的に行った方がよいでしょう。

[PRB] Column formatting via StdDataFormat objects doesn't work for DataGrid controls / StdDataFormat オブジェクト経由で行われる DataGrid コントロールの Column の書式設定が動作しない。

VB Migration Partner support custom formatting in bound controls. References to StdDataFormat objects are converted to VB6StdDataFormat objects and migrated VB.NET applications works like the original VB6 app. VB Migration Partner even support custom formatting by means of the Format and Unformat events of the StdDataFormat class.

VB Migration Partner はバインドコントロールのカスタム書式設定に対応しています。StdDataFormat オブジェクトに対する参照は VB6StdDataFormat オブジェクトに変換されることで、変換後の VB.NET アプリケーションは元の VB6 アプリケーションと同じように動作します。VB Migration Partner は StdDataFormat クラスの Format イベントと Unformat イベントを使ってカスタム書式設定にも対応しています。

However, the custom formatting mechanism works smoothly only for VB6 controls that are rendered as managed .NET controls, such as TextBox and Label. When the bound control is a wrapper around an ActiveX control, as is the case of the DataGrid control, you must edit either the original VB6 code or the migrated VB.NET code.

しかし、カスタム書式設定のメカニズムは.NET のマネージドコントロールとして動作する VB6 コントロール—例えば、TextBox や Label—に対してしか正常に動作しません。バインドコントロールが ActiveX コントロールのラッパーである場合、DataGrid コントロールの場合のように、元の VB6 コードを修正するか変換後の VB.NET コントロールを修正する必要があります。

For example, let's assume that the original VB6 application includes the following code:

例えば、VB6 アプリケーションが次のようなコードを含んでいると仮定します。

```
Dim sdf As New StdDataFormat
sdf.FalseValue = "Is False"
sdf.TrueValue = "Is True"
Set DataGrid1.Columns(1).DataFormat = sdf
```

The converted VB.NET code causes a compilation error because you can't assign a VB6StdDataFormat object (defined in the control support library) to the DataFormat property exposed by columns of the DataGrid ActiveX control:

変換後の VB.NET コードはコンパイルエラーを起こします。理由は DataGrid ActiveX コントロールのカラムによって公開されている DataFormat プロパティに(コントロールサポートライブラリに定義されている)VB6StdDataFormat オブジェクトを代入できないためです。

```
Dim sdf As New VB6StdDataFormat()
sdf.FalseValue = "Is False"
sdf.TrueValue = "Is True"
DataGrid1.Columns(1).DataFormat = sdf
```

You can fix this problem by passing the VB6StdDataFormat object in a ToStdDataFormat6 method:

ToStdDataFormat6 メソッドに VB6StdDataFormat オブジェクトを渡すことでこの問題は解決できます。

```
Dim sdf As New StdDataFormat
sdf.FalseValue = "Is False"
sdf.TrueValue = "Is True"
Set DataGrid1.Columns(1).DataFormat = ToStdDataFormat6(sdf)
```

The ToStdDataFormat6 is defined in the VBMigrationPartner\_Support module method and does nothing under VB6. When the code is migrated to VB.NET, however, the method returns an instance of a StdDataFormat COM instance that is equivalent to the VB6StdDataFormat object defined at the top of the code snippet. After this substitution, all works as expected, including Formant and Unformat events.

ToStdDataFormat6 は VBMigrationPartner\_Support メソッドに定義されており、VB6 では動作しません。しかし、このコードが VB.NET に変換されると、メソッドは、抜粋されたコードの 1 行目に定義されている VB6StdDataFormat オブジェクトと等しい StdDataFormat の COM インスタンスのインスタンスを返すようになります。この置き換えによって、Formant イベントと Unformat イベントを含むすべての動作が正常に行われるようになります。

Notice that the ToStdDataFormat6 method requires a reference to the Microsoft.StdFormat DLL. You can this reference manually after the migration or, better, automatically during the migration by means of an [AddReference](#) pragma.

注意すべきことは ToStdDataFormat6 メソッドが Microsoft.StdFormat DLL に対する参照を必要とすることです。変換後に手作業で参照設定することもできますが、むしろ [AddReference](#) プラグマを使用して変換時に自動的に行われるようにしたほうがよいでしょう。

**Update:** starting with VB Migration Partner 1.10, all references to the DataFormat property of DataColumn are migrated as references to the special DataFormat6 property, which takes and returns a VB6StdDataFormat object. For this reason, the fix described in this article is necessary only if the DataFormat property is referenced via late-binding.

**更新情報:** VB Migration Partner 1.10 以降、DataGrid の Column の DataFormat プロパティへの参照はすべて特別な DataFormat6 プロパティ—このプロパティは VB6StdDataFormat オブジェクトの受け渡しを行います—への参照に変換されます。このため、この記事に書かれた修正は DataFormat プロパティがデータバインディング経由で参照される場合にのみ必要となります。