

VERSION HISTORY／バージョン履歴

Version 1.34.01 – November 14, 2011／バージョン 1.34.01 – 2011 年 11 月 14 日

- FIXED: VarType6 function returned 36-vbUserDefinedType when passed an enum value.／VarType6 は列挙型を渡されると vbUserDefinedType を返します。
- FIXED: the code generator has been modified so that a For Each that iterates over a form now correctly iterates over the form's controls.／フォームに対して繰り返し実行する For Each がフォームのコントロールに対しても正常に繰り返し実行するようにコードジェネレータが変更されました。
- FIXED: references to controls in the form UserControl.Controlname were mistakenly considered as late-bound reference, hence their default property wasn't correctly resolved.／フォーム上のコントロール (UserControl.Controlname) に対する参照が誤って遅延バインディング参照と見なされていたため、デフォルトプロパティが正しく解決されませんでした。
- FIXED: the Data Source attribute wasn't correctly extracted from the connection string of an ADOConnection object. / Data Source 属性が ADOConnection オブジェクトの接続文字列から正しく抽出されませんでした。
- FIXED: removing an item with no associated key from the ListImages, Nodes, Panels, ListItems, ColumnHeaders and other similar collections might cause an exception. / ListImages や Nodes や Panels や ListItems や ColumnHeaders やその他の類似のコレクションからキーを伴わないアイテムを削除。
- FIXED: the Path and List properties of the DirListBox control returned incorrect values with network drives. / DirListBox コントロールの Path プロパティと List プロパティがネットワークドライブについて不正な値を返しました。
- FIXED: the DirListBox control showed the (unreachable) computer node and threw exception when the user clicked on it. / DirListBox コントロールが到達不可能なコンピュータノードを表示し、ユーザーがそれをクリックしたときにエラーとなりました。
- FIXED: the DirListBox control failed to display sub-directories that aren't accessible by the user, as well as all directories that follow it. / DirListBox コントロールがユーザーによってアクセスできないサブディレクトリを表示することに失敗しました。その下のすべてのディレクトリも同様でした。
- FIXED: when translating For Each loops that iterate over a generic form (as opposed to a specific form, eg Form1), the code generator failed to append the reference to Controls6 collection. / ジェネリックなフォーム (例えば、Form1 のような特別なフォームではないフォーム) に対して繰り返される For Each ループを変換する際、コードジェネレータが Controls6 コレクションに対する参照を追加することに失敗しました。
- FIXED: (ADOLibrary) The Connection.Execute and Connection.OpenSchema methods mistakenly returned a forwardonly-readonly recordset instead of a server-side static recordset (as ADODB does). Because ADOLibrary doesn't support server-side static recordsets (except for SQL Server) the OpenSchema method now returns a client-side static recordset, which supports navigation and bookmarks (as ADODB does). / (ADOLibrary の) Connection.Execute メソッドと Connection.OpenSchema メソッドが、(ADODB が返すような) サーバーサイドの静的レコードセットではなく、誤って forwardonly-readonly なレコードセットを返しました。ADOLibrary が (SQL Server を除いて) サーバーサイドの静的レコードセットに対応していないため、現在は OpenSchema メソッドがクライアントサイドの静的レコードセットを返します。ADOLibrary は (ADODB がそうであるように) navigation と bookmarks に対応しています。
- FIXED: (ADOLibrary) The Connection.OpenSchema method didn't work correctly when no field restrictions were specified. / フィールドの制約が指定されていなかった場合、(ADOLibrary の) Connection.OpenSchema メソッドは正常に動作しませんでした。

- FIXED: in some cases, Property Get and Property Let with different scope generated an invalid Property VB.NET block／異なるスコープの Property Get と Property Let が不正なプロパティブロックを生成することがありました。
- FIXED: in certain cases, the binding mechanism implemented via the VB6BindingCollection class threw an exception when the current record had been deleted.／カレントレコードが削除されたときに VB6BindingCollection クラス経由で実装されたバインディング機構がエラーを発生させることがありました。
- FIXED: event handlers of the ADODB.Recordset and ADODB.Connection objects were not correctly recognized as such (this bug was introduced in version 1.34)／ADODB.Recordset オブジェクトと ADODB.Connection オブジェクトのイベントハンドラーがそれらを正しく認識されませんでした。(このバグはバージョン 1.34 で発見されたものです。)
- FIXED: setting ScaleMode=vbHimetric now correctly raises error 380 (Invalid property valid), as in VB6.／ScaleMode=vbHimetric としたとき、現在は正常に(VB6 と同様の)380 エラー(無効なプロパティが有効である)が発生します。

Version 1.34 – October 6, 2011／バージョン 1.34.00 – 2011 年 10 月 6 日

- ADDED: CreateArray6, ClearArray6, Redim6 and RedimPreserve6 now support 4-dim arrays
- ADDED: the VB6Config.DBCSSupport flag enables special treatment for DBCS strings (Japanese versions)
- ADDED: the VB6Config.UseExCompositeStyle flag reduces flickering on forms that have many controls and a background picture. By default, .NET forms

refresh the background each time a new control is loaded; when this flag is True, the background is refreshed only once, after the last control has become visible.

- ADDED: For uniformity's sake, CByte6, CInt6, CLng6, CSng6 have been added to the library, even though VB Migration Partner never generates calls to them.
- ADDED: new migration INFO remark is generated for each variable/member whose type has been inferred
- ADDED: the VB6Config.UseVBPowerPack flag indicates whether the PrintForm method uses the VB PowerPacks library.
- ADDED: the VB6CommonDialog class exposes the following new properties: OpenFileDialog, SaveFileDialog, FontDialog, ColorDialog, PrintDialog. If these properties are set before invoking the ShowXxxx method, then the specified control is used instead of a temporary control, so that additional properties can be written to or read from.
- ADDED: the new FileGetArrayHeader6 and FilePutArrayHeader6 methods read/write SafeArray headers.
- ADDED: the VB6Form, VB6PictureBox, and VB6UserControl classes now expose the DoubleBuffered property, which permits to perform groups of complex graphic methods and make their outcome appear immediately as if it were a single operation.
- ADDED: the VB6CheckBox, VB6OptionButton, VB6CommandButton, VB6WLCheck, VB6WLOption, and VB6WLCommand controls now expose the UseTextAlignment boolean property (also available in the property window at design time). If this property is true, then these controls honor the .NET TextAlignment property instead of mimicking the VB6 behavior.
- ADDED: the new CodeArchitects.VBPowerPack.Dll adds support for the following classes, that use features from Microsoft VB PowerPacks VB6PrinterPP, VB6PrintersPP, VB6LinePP, VB6ShapePP and VB6CommonDialogPP. Thanks to this new feature, the VBSupportLib.dll is never required for these controls.
- CHANGED: the VB6ImageList control now generates a companion ImageList .NET control, which actually contains the images. The user should add or modify images at design-time by means of this companion ImageList control. This

change can cause an error if you install a new version of the CodeArchitects.VBLibrary DLL for apps migrated by earlier versions of VB Migration Partner. More info: <http://www.vbmigration.com/detknowledgebase.aspx?id=676>

- CHANGED: each exe/dll has its own instance of the App6, Screen6, and Forms6 object (in previous versions all executables shared the same global object). This change may affect behavior of migrated .NET apps.
- CHANGED: Chr/ChrW calls with a constant numeric values are converted to ControlChars.NNN constants (if possible, ie Chr(13)); if not, they are converted to standard Chr/ChrW VB.NET method (instead of Chr6).
- CHANGED: forms that have no menu, no caption and ControlBox=False were displayed as borderless forms, regardless of the value of the BorderStyle property. Starting with this version, these forms are now shown as borderless forms only if BorderStyle is equal to 1-vbFixedSingle or 4-vbFixedToolWindow.
- CHANGED: the Count property of following collections now returns a 32-bit integer (instead of a 16-bit integer)
GROUP A) StatusBar.Panels, TapStrip.Tabs, ToolBar.Buttons, ToolBar.ButtonMenus, TreeView.Nodes, ImageList.ListImages, GROUP B) ListView.ListItems, ListView.ListSubItems, ListView.ColumnHeaders, ImageCombo.ComboItems please notice, however, that in VB6 the Count properties in collections belonging to group A are 16-bit values. Treating them as 32-bit integers should not introduce any issue, though.
- CHANGED: the VB6Binding and VB6Bindingcollection classes have been moved to CodeArchitects.VBLibraryAdodb.dll.
- CHANGED: the App6, Screen6, Forms6, Printer6, and Printers6 global objects are implemented as properties (instead of fields) in the VisualBasic6.Support.vb file, and the corresponding object is instantiated only on-demand (lazy instantiation)
- CHANGED: no 0354 warning is emitted for statements that read the Value property of a DAO/ADODB field
- CHANGED: when saving a converted VB.NET project, the output path defined in Options dialog can be now provided
- CHANGED: support for Line, Shape, CommonDialog, Printer, and Printers objects is implemented by means of features exposed by CodeArchitects.VBPowerPack.Dll, which in turn uses Microsoft VB PowerPacks. For this reason, the Microsoft.VisualBasic.VBPowerPacks.Vs.dll assembly must be installed on both the computer when the migration is performed and on the end user's computer. (You can disable this new feature by removing the CodeArchitects.VBPowerPack.dll from VB Migration Partner's setup folder)
- CHANGED: because the Printer object is now implemented by means of the Printer class found in Microsoft.VisualBasic.PowerPacks.Vs, the following properties aren't supported any longer: DrawMode, hDC, DevideName, Port, Zoom
- CHANGED: because Shape controls are now by means of ShapeRectangle and ShapeOval classes found in Microsoft.VisualBasic.PowerPacks.Vs, it isn't possible to define an array of Shape controls that mix rectangular and oval shapes. Likewise, it isn't possible to change the Shape property at runtime in such a way that a rectangular shape is transformed into an circle/oval shape, and vice versa.
- CHANGED: AxWrapperGen now generates x86 projects (instead of "AnyCPU")
- FIXED: the AutoDispose pragma wasn't correctly applied to UDTs
- FIXED: in some cases a reference to a property named after a form class was renamed as "Me"
- FIXED: the design-time value of the Tag property of the ListView, Toolbar, and StatusBar controls wasn't migrated properly and was set to Nothing in the migrated VB.NET code.
- FIXED: a bug introduced in v.1.33 prevented the warning 0328 from being emitted
- FIXED: floating point constants with missing "As" clause (e.g. "Const x = 123.45") were considered of type Single instead of Double

- FIXED: under some circumstances, the MergeIfs pragma caused a #0338 warning (“Unable to translate next statement”)
- FIXED: the Handles keyword was mistakenly generated also for methods that didn’t exactly match the event signature
- FIXED: when adding more than 32676 elements to a ComboBox or Listbox, an overflow occurred. In VB6 no error is raised, and the NewIndex property is assigned a negative value.
- FIXED: the SSPanel didn’t expose the Windowless property; this property has been added to fix compilation errors, even though it is obsolete and unsupported
- FIXED: the MaskColor of the ImageList control associated with a Toolbar control wasn’t honored when adding a button to the toolbar.
- FIXED: Type...End Type blocks containing elements named CStr, CInt, etc. caused a compilation error after being migrated to .NET.
- FIXED: the VB6MSHFlexGrid control didn’t expose the BackColorHeader property.
- FIXED: the CDate6 and IsDate6 methods also recognize dates in yyyy/mm/dd format, as in VB6.
- FIXED: The App6 object was shared among the client EXE and all its DLL (in VB6 each executable has its own App object); in current version the App6 object in a DLL returns project properties of the DLL, not the main EXE.
- FIXED: The Screen6 object was shared among the client EXE and all its DLL (in VB6 each executable has its own Screen6 object); in current version the ActiveForm and ActiveControl properties correctly return a form or control that belongs to the same executable that is querying the object.
- FIXED: The Forms6 collection was shared among the client EXE and all its DLL (in VB6 each executable has its own Forms6 collection); in current version it correctly returns an empty collection if queried from inside a DLL project that doesn’t display any form.
- FIXED: a statement such as If x=0 Then Gosub One Else Gosub Two didn’t cause the 02A4 warning to be emitted.
- FIXED: collection classes migrated to VB.NET didn’t correctly support enumeration (For Each) from COM/VB6 clients; this problem has been fixed by forcing the “NewEnum” enumerator method to be renamed as GetEnumerator.
- FIXED: if the current form is unloaded when a MonthView control is clicked, an ObjectDisposed exception could occur.
- FIXED: the SmallChange property of HScrollBar and VScrollBar controls (and their windowless counterparts) wasn’t correctly retained in code-behind portions.
- FIXED: the Nodes.Add method threw an “Invalid Key” exception when the SelectedImage optional argument is omitted.
- FIXED: the MaskedTextBox control didn’t display the mask character until the control got the focus.
- FIXED: the design-time appearance of the MaskedTextBox control didn’t reflect the current value of the Mask property, as it does in VB6. Notice that starting with this version, the Text property of the MaskedTextBox control isn’t listed in the Properties window (as in VB6).
- FIXED: if a method uses ReDim that references a global Variant variable, a local variable named after the ReDim argument was mistakenly defined, and could not be removed by means of a DeclareImplicitVariables pragma.
- FIXED: the default Printer was mistakenly changed by an enumeration over the Printers6 collection.
- FIXED: the Round6 method ignores the numDigit argument if the 1st argument is an Object or a VB6Variant, and mistakenly throws an exception when the argument is Null (or a VB6Variant that contains Null).
- FIXED: cloning a UDT didn’t correctly clones the fixed-length strings it contains.
- FIXED: FilePut and FileGet methods didn’t work correctly with UDTs on Japanese version of Windows.
- FIXED: in some cases, the InferType pragma generated data types with a System prefix (e.g. System.Object)
- FIXED: the SetOption RemarkMethodBody didn’t exclude code automatically generated at the top of a method.
- FIXED: the Unload6 overload that takes a DataEnvironment as an argument wasn’t correctly implemented.

- FIXED: conversion methods (CBool6, CDb16, CDec6, and the newly added CInt6, CLng6, CSng6, CByte6) correctly trim the argument to the first ASCII 0 char and discard leading/trailing spaces
- FIXED: opening an existent file for writing with the FileSystemObject class creates a new file under VB6, but doesn't in VB.NET, regardless of whether the Create parameter is False or True.
- FIXED: the FileSystemObject class opens text file for reading in exclusive mode (instead of shared mode)
- FIXED: the RightToLeft property wasn't serialized correctly in the code-behind portion of a form.
- FIXED: the RightToLeft property was missing in the VB6PictureBox control.
- FIXED: setting the TabVisible property of a SSTab control before the control has been created resulted in an exception.
- FIXED: the Controls collection didn't include controls on invisible tabs of a SSTab control
- FIXED: a Line or Shape control hosted inside a SSTab control was moved to the form's surface when the form is modified inside Visual Studio designer.
- FIXED: the Parent and Container properties of controls hosted on an invisible tab of a SSTab control returned Nothing.
- FIXED: setting the Selected(index) property to False of a ListBox, WLList, and FileListBox control didn't work correctly.
- FIXED: the PrintForm method didn't work well with controls on a TabStrip (the fix requires setting VB6Config.UseVBPowerPack=True)
- FIXED: the DisableMessage pragma had no effect on warning 03E2
- FIXED: the code generated for IVB6FileGetPut interface methods correctly reads/writes SafeArrays, which improves compatibility with VB6 binary files
- FIXED: reading/writing UDTs containing dynamic arrays and multi-dimensional arrays didn't work correctly.
- FIXED: REM statements whose comment begins with "=" weren't translated correctly
- FIXED: File I/O with arrays of fixed-length strings didn't work correctly
- FIXED: the BorderStyle property of the VB6Form wasn't visible in the Properties window
- FIXED: in some cases, the DefaultMemberSupport pragma failed to insert a call to GetDefaultMember6 method around arguments passed to scalar method parameters.
- FIXED: the UpDown control threw an exception if the end user types a value that is outside the (Min,Max) range and then clicks on either up or down arrow.
- FIXED: in certain circumstances, events raised from code-behind for controls belonging to a control array raised an exception
- FIXED: the VB6FixedString_NNN class wasn't generated in certain circumstances
- FIXED: the DoubleBuffered property of VB6Form, VB6PictureBox, and VB6UserControl classes has been enhanced to greatly improve execution speed and reduce flickering.
- FIXED: controls hosted on a container (e.g a PictureBox) mistakenly triggered events when the code-behind portion of the form is executed.
- FIXED: forms with background picture and no other visible controls on it didn't fire the Paint event.
- FIXED: invisible Label and Image controls (e.g. Labels with empty caption, Images with no picture) fired two Click events in some circumstances
- FIXED: The OpenTextFile method of the FileSystemObjec class didn't truncate an existing file when ioMode=ForWriting and create=True
- FIXED: if the default item of a collection class was a Function instead of a property, a #0338 warning ("Unable to translate next statement") emitted.

- FIXED: in certain circumstances, wrapper classes generated by AxWrapperGen fail to load on 64-bit systems.
- FIXED: a temporary memory leakage occurred when using VBA enumerated constants, eg vbCr or vbCritical.