Visual Basic 5.0 Control Creation Edition Beta 1

Below is the list of know issues for this beta release of Visual Basic 5.0 Control Creation Edition [VB5 CCE]. Please review it carefully before using this product. Though every effort is made to list all open issues, there are some that have not been documented yet. Please report bugs not listed here using the support form provided at http://www.microsoft.com/vbasic/controls/resource.

INSTALLATION ISSUES

Extra erroneous registry key created when Visual Basic Control Creation Edition sample controls installed

In addition to the bonafide registry key for a sample control, the following erroneous extra registry key is created when you install the CCE sample controls:

HKEY_CLASSES_ROOT\CLSID\{0DE86A52-2BAA-11CF-A229-00AA003D7352}.

This key causes a blank item to show up in the list of ActiveX controls displayed in Access95 and VB4, and causes an "object server not registered" error message when you open the Custom Controls dialog in Visual Basic Control Creation Edition. It can be removed with the RegEdit utility.

Visual Basic setup doesn't register file types

When you setup Visual Basic, it doesn't automatically associate the Visual Basic-specific file extensions with Visual Basic. You can associate the file extensions yourself, though, through the File Types tab in the Options dialog box in File Explorer. (To get this, choose Options from the View menu in File Explorer.)

Visual Basic 5.0 beta 1 and beta 2 issues

If you are a member of the Visual Basic 5.0 beta program and are installing Visual Basic 5.0 Control Creation Edition on a computer with beta 1 or beta 2 of VB 5, building Internet CAB files with Application Setup Wizard may produce non-usable CAB files. The problem is that the ComCtl32.DEP and ComDlg32.DEP files are not updated correctly by the Control Creation beta 1 install. The solution is to delete these two .DEP files from the Windows system directory and reinstall the Control Creation Edition.

Uninstalling Office97 betas may cause VB5 CCE to be disabled

Office97 betas and VB5 CCE beta 1 share certain registry settings. If Offic97 beta is uninstalled, it may cause VB5 CCE to not load. Reinstall VB5 CCE to fix the problem. Future Office97 release will not exhibit this problem

INTERNET RELATED ISSUES

Displaying Visual Basic-created controls in browsers

Microsoft's Internet Explorer version 3.01 or later is required to host Visual Basic-created controls on the Internet.

Internet Explorer crashes when attempting to go to new web page after an error occurs in a Visual Basic-created ActiveX control.

If an error occurs in a Visual Basic-created ActiveX control which isn't handled by an internal error handling procedure (which generates a runtime error), you can't go to a new web page without crashing Internet Explorer.

This is a known issue in the Visual Basic Control Creation Edition. We are researching a solution to it and will post the results when known.

To avoid the problem, make sure all ActiveX controls contain error handling procedures.

Internet Explorer crashes when trying to open a modeless form in a Visual Basic-created ActiveX control

When a Visual Basic-created ActiveX control attempts to open a modeless form on a web page (such as `Form1.Show`), the error "369 - Operation not valid in an ActiveX DLL" occurs. If you refresh the page and attempt to open the modeless form again, Internet Explorer crahses with an invalid page fault. To avoid

this problem, use modal forms.

We are researching a solution to this problem and will post the results when known.

Downloading multiple controls on one web page

If two or more Visual Basic built controls are on one web page and each control exists in its own CAB file, only one control will get downloaded if MSVBVM50.DLL does not already exist on the user's machine.

There is a work around for this problem: use only one CAB file per web page. To do this, place all user controls into one project. Use the Application Setup Wizard to package all of the Visual Basic built controls you want to use on a web page into one CAB file. The Internet Explorer code download mechanism will know to download this CAB only once, no matter how many OBJECT tags refer to the CAB on that web page. Additionally, the MSVBVM50.DLL will only be downloaded once if needed.

Code sharing between VB CCE and VB4 when using ComCtl32.OCX controls

VB CCE overwrites comctl32.ocx that ships with VB4. VB 4 projects that use this control and persist collections (imagelists, panels, etc.) will load fine into VB CCE. If a VB 4 project using this control is saved in VB CCE and then loaded back into VB 4, the persistent collection data will be unreadable by VB 4. The persistence format is not currently backward compatible. The new comctl32.ocx can read the old format but cannot write the old format.

A fix for this issue is being evaluated for future releases of VB CCE.

CONTROL HOST ISSUES

Delphi host compatibility

Properties exposed from a Visual Basic built control are not displayed by the Delphi property browser. This is a result of differing methods of describing property information on a control's interface description.

To avoid this problem, create a custom property page for your control. Right-click on the control in Delphi to bring up your custom property page. This gives you full access to your control's properties.

A modification is being explored for future releases of the Control Creation Edition.

Powerbuilder host compatibility

Controls built in beta 1 of the Control Creation Edition of Visual Basic do not work in Powerbuilder 5.0 due to control interface implementation problems in Powerbuilder.

Powersoft has been notified of the issues and Microsoft is working with them on a solution.

Persisting Font/Picture properties causes error in ControlPad

ControlPad fails to call the Set Font property when modifying a font object using the property browser.

The work around is as follows:

First, ControlPad tries to persist the font/picture as text to the

HTML stream. This will fire the control's WriteProperties event. If the control tries to save a binary object, ControlPad will return an error, since ControlPad is expecting text. If your control does not have error handling (on error resume next), then your control will get a runtime error.

Second, ControlPad will realize that you got an error trying to save as text. It will then ask the control to save as binary. This will fire the control's WriteProperties event again. The attempt to save the picture/font object as binary will succeed. ControlPad will then write out a DATA tag instead of PARAM tag in to the HTML. The DATA tag will render fine in a browser, but it is not easily editable in notepad.

The scenario below demonstrates this problem:

1 Start VB5CCE

2 Add a Picture control and a TextBox to control and add the following code to the control

```
Private Sub Picture1_Click()

     On Error Resume Next

     Picture1.Picture =
LoadPicture(InputBox("Enter a picturename"))

End Sub

Public Property Get Picture() As Picture

     Set Picture = Picture1.Picture

End Property


Public Property Set Picture(ByVal vNewValue As
Picture)
```

```
     Set Picture1.Picture = vNewValue

End Property

Public Property Get Font() As Font

     Set Font = Text1.Font

End Property

Public Property Set Font(ByVal vNewValue As
Font)

     Set Text1.Font = vNewValue

     Text1.Refresh

End Property

Private Sub UserControl_ReadProperties(PropBag
As PropertyBag)

     On Error Resume Next

     Set Text1.Font =
PropBag.ReadProperty("Font")

     If (Err) Then MsgBox Err.Description

     Err.Clear

     Set Picture1.Picture =
PropBag.ReadProperty("Picture")

     If (Err) Then MsgBox Err.Description

End Sub
```

```
Private Sub UserControl_WriteProperties(PropBag
As PropertyBag)

     On Error Resume Next

     PropBag.WriteProperty "Font", Text1.Font

     If (Err) Then MsgBox Err.Description

     Err.Clear

     PropBag.WriteProperty "Picture",
Picture1.Picture

     If (Err) Then MsgBox Err.Description

End Sub
```

3 Set ProcID for FONT to FONT

4 Set EditAtDesignTime property to TRUE for the usercontrol

5 Build Project1.OCX

6 Start ActiveX Control Pad and Insert Project1.UserControl1 into the HTML page

7 Change the FONT property from the Property sheet.

8 From the design surface click on the control Picture Control (you may need a second click to fire the Click Event) - Enter a valid bmp/jpg filename

9 Close the designer

Data types not supported in VB4, Access95 property sheet implementation

Some data types are not supported by VB4 and Access95 property sheets. Properties of type byte, variant, and those that are declared as ByRef do not show up in VB4 and Access95 property sheets. Note, that ByRef is the default passing method if ByRef or ByVal is not explicitly stated.

Passing Byte parameters in control events crashes VB4 and VB5 CCE

Passing a Byte argument in a parameter to a control event causes VB4 and VB5 CCE to crash. Do not use Byte parameters in control events until a fix notification is posted. The problem is being evaluated for future releases of VB5 CCE.

ActiveX Control Pad Runtime Error #50006

If your control persists binary data, such as the Font object, you may see Runtime Error#50006 when closing the Control designer in ActiveX Control Pad. After this error occurs your control's persisted data will not be written correctly to the HTML stream. This error is most likely caused by an untrapped runtime error occuring in your control's WriteProperties event. To avoid this problem, make sure you have error handling code in your usercontrol_writeproperties event, such as On Error Resume Next.

Access95 fails when inserting VB5 CCE built control

Controls built using beta 1 of VB5 CCE may not run inside of Access95. This issue is known and a fix will be available in the next release of VB5 CCE.

GENRAL ISSUES

Graphics – Potential out-of-memory condition when loading a large .JPG image

If you attempt to load a large .JPG image into an application that's being run under Windows 95 in 16-color mode, you may get an out-of-memory error.

Icon property – Potential failure when setting a form's Icon property using ImageList

Your application may fail when using the ImageList control to set the Form.Icon property.

For example, the following scenario will produce the error:

10 **Start Visual Basic and add an ImageList control to Form1.**

11 **Bring up the General property page and click the 16x16 option**

12 **Click the Images tab, click "Insert picture" and load a bitmap.**

13 **Add the following code to Form_Load:**

```
14      Me.Icon = ImageList1.ListImages(1).ExtractIcon
```

15 **Run the application. A general failure will result.**

MDI Child windows do not maintain their WindowState settings

Under Windows 95 and Windows NT 4.0, MDI child windows do not maintain their WindowState settings when the MDI form is set as the startup form and the MDIChild property of each form is set to True.

MultiLine text box – Application failure on form load

If you set a text box control to multiline, and at design-time set Enabled to False, your application will fail when you go into run mode.

OLE container control

Under certain scenarios, you application may fail if you repeatedly load and unload an OLE container control.

Context sensitive help disabled for beta 1

Context sensitive help is disabled for beta 1 of the Control Creation Edition. Online help can still be accessed through the "Search Reference Index…" menu command in the Help menu of Visual Basic.

Online help is not automatically installed with the Visual Basic 5.0 Control Creation Edition. Online help and useful reference information can be downloaded from the Visual Basic website at www.microsoft.com.

Palette – Using OLE drag and drop loses palette for image

You may temporarily lose image palettes under the following scenario:

16 **Create a new Standard Exe project.**

17 **Create an image and a picture box and set OleDrag and OleDrop both to automatic.**

18 **Set the form's Palette property to bitmap1.bmp.**

19 **Set the image's Picture property to gif1.gif.**

20 **Run the application and drag a picture from the image to the picture box.**

The palette characteristics of the image may become incorrect.

Saving projects – Potential code loss when saving shared items in projects

When you share project items between projects that are open in the same instance of the development environment and you make modifications to the same file in each open project (you must open the project item, modify it, and close it in both projects ignoring all warnings), you may end up losing data if you save the entire project group. Only the first copy of the changes will be saved. It is suggested that you only modify shared project items in one project.

Error "398 - Client site not available" after cutting, copying, or pasting Visual Basic-created ActiveX control

When you attempt to perform a clipboard-related operation (such as cutting, copying, or pasting) with a Visual Basic-created ActiveX control, you get the error "398: Client site not available".

This occurs due to an internal problem with Visual Basic-created ActiveX controls.

To work around the problem, insert error-handling code in the ActiveX control.

Error "Application-defined or object-defined error" when ActiveX control event is deleted in Break mode

When you delete an event procedure of an ActiveX control in Break mode, you get the error, "Application-defined or object-defined error" in the host application.

This is a known issue in the Visual Basic Control Creation Edition. We are researching a solution to it and will post the

results when known.

Visual Basic crashes when large statement contains runtime error within a Then clause or Case of Select statement

If a large statement (executable code-wise) contains a runtime error within its Then clause or Case of its Select statement, it will cause Visual Basic to crash. Avoid long statements in this scenario.

For example, the following:

```
Sub main()

   On Error Resume Next

   If True Then

    MsgBox "test" + tst(Array(1, 2, 3), 2, 2,
3, 4, 5, _

       CInt("abc"))

    ' long stmt with an error

   Else

    x = 2

   End If

End Sub
```

This is a known issue in the Visual Basic Control Creation Edition. We are researching a solution to it and will post the results when known.

Edit and Continue operations in a Visual Basic built control may cause VB to crash

Performing an edit to code in a user control while in break mode may cause VB to crash. This error occurs when the code edit forces a reset of the project (for example, changing the project name space changes). A fix for this problem is currently being evaluated.

Control version compatibility and debugging

When debugging VB CCE built controls, use the 'Project Compatibility' setting on the Component tab of the Project Properties dialog. This ensures that the GUID of the control remains constant between builds of your control, greatly saving time during debugging. The samples that ship with VB CCE have the 'No Compatibility' option set. It is recommended that you change this setting for the sample controls to be 'Project Compatibility' before compiling the control.

Binary compatibility may cause improper control name to display in Components dialog

VB CCE built controls using the 'Binary Compatibility' setting on the Component tab of the Project Properties dialog may display the wrong friendly name in Components dialog. The list of controls on the 'Controls' tab of the 'Components' dialog will sometimes show the full path to the control instead of the control name. There are no programmatic effects from this problem. This issue will be addressed in the next release of VB5 CCE.

Application Setup Wizard generates a misleading comment in sample HTML

When creating CAB files for the Internet download of controls,

the Application Setup Wizard creates sample HTML with an EMBED tag for the control. A comment in the HTML gives information regarding the location of a license key packaging tool and refers to a tool that is not shipped with VB5 CCE. The comment should read:

"If any of the controls on this page require licensing, you must create a license package file. Run LPK_TOOL.EXE to create the required LPK file. LPK_TOOL.EXE can be found on the ActiveX SDK, http://www.microsoft.com/intdev/sdk/sdk.htm"