

Contents

VBctl3D Description

[Overview](#)

[Components and Installation](#)

[How to use VBctl3D](#)

Configuration Properties

[Overview](#)

[Property Descriptions](#)

The VBctl3D Configuration File

[Overview](#)

[The sections of VBCTL3D.INI](#)

Overview

For most users, a Windows application with a 3-D appearance looks more attractive than one developed with the standard Windows (2-D) controls. Though it is quite probable, that future versions of Windows will have built in 3-D support, today the developer has to do some additional work, if he wants to add 3-D effects to his programs. VBctl3D is designed to make this easier than ever before for Visual Basic programs.

Related Topics:

[Creating 3-D effects in Visual Basic](#)

[Automatic 3-D effects with VBctl3D](#)

Creating 3-D effects in Visual Basic

Without VBctl3D, a Visual Basic developer, who wants to create competitive software, has the following possibilities to create 3-D effects in his programs:

- Use standard controls and add 3-D effects with VB's drawing functions.
This is the less expensive way in terms of money (no add-on products), but the most expensive way in terms of programming work.
- Use 3rd party controls for the 3-D effects.
There are a lot of different and powerful controls on the market, which offer a 3-D look in addition to other enhancements over the standard VB controls. Besides the extra money for these controls, the developer has to face incompatibilities and unwanted or unnecessary features. It is also possible, that the control, which would best serve a special need, does have no 3-D properties at all.
- Use CTL3D.DLL for message boxes and dialogs.
This dynamic link library has been released by Microsoft to allow the creation of 3-D effects for dialogs and message boxes. It is quite easy to use and enhances VB programs a lot.

Automatic 3-D effects with VBctl3D

VBctl3D is a custom control, which allows to give a 3-D look to Visual Basic controls by adding an instance of this control to only one form of a project. VBctl3D works with the built in controls as well as with 3rd party controls, as long as these controls behave in a way comparable to the standard Windows objects. VBctl3D works without setting any properties, which makes it easy to use but nevertheless quite powerful. In fact, the first release of VBctl3D started without any properties at all, and the properties of the current version are only used for customization of the default control behaviour. For more information see [Property Descriptions](#).

VBctl3D also adds 3-D effects to dialogs and message boxes, so there is no need to use CTL3D.DLL together with VBctl3D.

Components and Installation

Contents of the distribution disk

Installation

Contents of the distribution disk

The VBCTL3D distribution disk contains the following files (demo versions may contain fewer or additional components, look for a READ.ME file with explanations). For electronically distributed versions, the VBCTL3D components may be compressed into a single file:

Filename	Description
VBCTL3D.VBX	The custom control file. You may distribute this file with your application.
VBCTL3D.LIC	This file, which should reside in the Windows System directory, contains license information. It is needed to compile your VBCTL3D application to an EXE file. If the license file cannot be found or is corrupted, the control is loaded in demonstration mode. The information in the license file can be displayed by selecting the (About) property.
VBCTL3D.HLP	On-line help for VBCTL3D.
VBCTL3D.INI	Sample VBCTL3D configuration file. This optional file is used for system wide configuration of VBCTL3D. For more information see <u>The sections of VBCTL3D.INI.</u>
LICENSE.WRI	License agreement in Windows Write format.

You are not allowed to distribute the license information file VBCTL3D.LIC. Distribution of the control file is free of charge.

Installation

If the installation is not accomplished by an installation program, simply copy the files from the distribution disk to a separate directory on your hard disk. The custom control file VBCTL3D.VBX and the license information file VBCTL3D.LIC should be copied to your Windows system directory. If you decide to use the configuration file, copy the sample file from the distribution disk to your Windows directory and customize its contents to your needs.

How to use VBctl3D

VBctl3D is used like any other Visual Basic custom control. For a quick start open a new project and follow these instructions:

- Use the Add File Command from the File menu to include the control library VBCTL3D.VBX into your project. You will notice that the conversion to 3-D is done immediately and that the interface components of the VB environment are also converted.
- You should now draw some of the built in controls on a form to see how VBctl3D works. You are also encouraged to add a menu to at least one of the forms in the project.
- To force the loading of the control for a compiled program, you must add an instance of the control to only one of the forms in your project.

Like the VB Timer control, VBctl3D is invisible at runtime, so the position of the control on the form is of no concern.

Related Topics:

[Controls, that work with VBctl3D](#)

[Support for Custom Controls](#)

[Support for Dialogs and Message Boxes](#)

[The Use of the BackColor Property](#)

Controls, that work with VBctl3D

VBctl3D works with any of the following built in controls of Visual Basic:

- Form
- Picture Box
- Label
- Frame
- Check Box
- Option Button
- Combo Box
- List Box
- Drive List Box
- Directory List Box
- File List Box
- Image
- Menu

VBctl3D does not do anything with command buttons, because these are displayed with a 3-D look by default.

Support for Custom Controls

VBctl3D can be extended to support custom controls, as long as these controls behave in a way comparable to the standard controls. There is built in support for the following custom controls:

Class Name	Control Description
MaskedTextBox	VB/Professional masked edit box
xEdit	Haas Service extended text box
xCheck	Haas Service extended check box
xOption	Haas Service extended option button

To support a specific custom control, you can

- use the VBctl3D configuration properties to add support for custom controls to an application. This is the preferred method. For more information see [Property Descriptions](#).
- edit the configuration file VBCTL3D.INI for system wide support of selected controls. This is supported to be compatible with older versions of the control. For more information see [The sections of VBCTL3D.INI](#).

Support for Dialogs and Message Boxes

Dialogs and message boxes are supported automatically. This extends to the VB environment itself.

The Use of the BackColor Property

In general, VBctl3D converts only those controls to 3-D, that have the default value set for their BackColor property (&H80000005, Window Background). Changing this property can be used to exclude some controls or complete forms from the conversion to 3-D.

If you are using VBctl3D with existing projects, you may have already changed the BackColor property of some controls and forms to let the application look more attractive. If you don't want to reset these properties but nevertheless have these objects been displayed in 3-D, you can use the **IgnoreColors** property or the corresponding entry in VBCTL3D.INI, which forces 3-D conversion without regard to the BackColor property.

There is one thing that should be noted with respect to forms and labels. To achieve an acceptable performance and to work also in the development environment, VBctl3D changes the BackColor property of forms and graphical controls (e.g. labels) to the Button Face color (&H8000000F). This makes it a little complicated to remove VBctl3D from a project. Though we are quite sure, that you never will need this, here is a simple trick to use in this situation:

- Use the 'Disabled' entry of the configuration file to temporarily shut down VBctl3D.
- Now open and close each form in your project. VBctl3D will convert back any of the forms and labels to the default BackColor property.
- Finally remove any VBctl3D instance and exclude the control from the project.

For more information about the configuration file see [The sections of VBCTL3D.INI.](#)

Overview

VBctl3D is designed to work without setting any properties and without coding any event handler. But there are situations where you might want to

- change the default behaviour of the control.
- add 3-D effects to custom controls other than those with VBctl3D built-in support.

In these situations you can use VBctl3D's configuration properties to enhance the control. In contrast to the configuration settings done with the configuration file VBCTL3D.INI (see [The sections of VBCTL3D.INI](#)), the scope of these settings is restricted to one application.

At runtime, the configuration changes implemented with properties come into effect only after the form with the (only) VBctl3D instance has been loaded. Therefore it is suggested to put that instance on the very first form that is displayed in an application (e.g. the about box).

Property Descriptions

This section describes the different configuration properties.

Related Topics:

[The Add Classes Dialog](#)

[The IgnoreColors Property](#)

[The No3DMenus Property](#)

The Add Classes Dialog

This property is used to support custom controls with VBCtl3D. If you want to add 3-D effects to a custom control which behaves like one of the standard controls, invoke this dialog and insert your control into one of the predefined control classes.

It is important that you know the exact name of the control's window class. You can use a tool like WinSight or Spy to determine this name. Please note that you don't have to include the leading Thunder signature common to all custom control classes.

The IgnoreColors Property

Set this property to True to disable checking of the BackColor property. For more information see [The Use of the BackColor Property](#)

This property is read-only at runtime.

The No3DMenus Property

Set this property to True, if you don't want 3-D menus at all, or if you already use another control for 3-D menus

This property is read-only at runtime.

.

Overview

The VBCTL3D.INI configuration file is used to

- customize VBctl3D
- add information about custom controls to support

The configuration file is optional and should be located in the Windows directory.

It is recommended to use configuration properties instead of the configuration file for VBctl3D customization.

Related Topics:

[Evaluation of the configuration file](#)

Evaluation of the configuration file

The contents of the configuration file is read in when the custom control file is loaded for the first time (i.e. from the first VBCtl3D application, including the VB environment). This implies that configuration changes are activated only after any application that uses VBCtl3D has been closed.

The sections of VBCTL3D.INI

VBCTL3D.INI is a standard Windows configuration file. It consists of different sections and can be created and customized with any text editor.

Related Topics:

[General section](#)

[Custom Control configuration sections](#)

General section

This section is used to customize VBctl3D to your personal preferences. It supports the following keywords:

Keyword	Settings	Description
Disabled	0 = FALSE (Default) 1 = TRUE	Set this to TRUE, if you want to temporarily disable VBctl3D, e.g. if the 3-D effects look bad with your current display configuration.
No3DMenus	0 = FALSE (Default) 1 = TRUE	If you don't want 3-D menus at all, or if you already use another control for 3-D menus, use this switch to disable VBctl3D's menu conversion.
IgnoreColors	0 = FALSE (Default) 1 = TRUE	Use this switch to disable checking of the BackColor property. For more information see The Use of the BackColor Property .

Custom Control configuration sections

If you want VBctl3D to support a custom control, which behaves similar to a standard control, simply insert a line of the form

```
ClassName=1
```

into one of the following sections (ClassName is the control's class name):

- TextBoxControls
- ListBoxControls
- ComboBoxControls
- CheckBoxControls
- OptionButtonControls
- FrameControls
- PictureBoxControls
- GraphicalControls

You can temporarily disable the support for a control by changing the corresponding line to:

```
ClassName=0
```

The sample file on the distribution disk contains empty sections for each of the possible control classes.

VBCtl3D is easier to use than any other control, but exactly that's what makes this control so powerful.

