Contents

VBCtl3D Description

The VBCtl3D configuration file

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 support for 3-D controls, nowadays the developer has to do some additional work 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 two 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. VBCtlD has no special properties, which makes it <u>easy</u> to use but nevertheless quite powerful.

VBCtl3D also adds 3-D effects to dialogs and message boxes, so there is no need to use CTLD.DLL together with VBCtl3D.

Components and 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	Development version of the control.
VBCTL3DR.VBX	Runtime version of the control. If you want to distribute an application, that uses VBCtl3D, rename this file to VBCTL3D.VBX and add it to your distribution disks.
VBCTL3D.HLP	On-line help for VBCtl3D.
VBCTL3D.INI	Sample VBCtl3D configuration file. This **optional** file is used to configure VBCtl3D to your personal preferences (for more information see The sections of VBCTL3D.INI).

You are not allowed to distribute the development version of the control. Distribution of   the runtime version 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 should also be copied to your Windows system directory to allow access for compiled programs. 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. It is planned to add 3-D captions to frames and menus in a future version. This will then be extended to command buttons.

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:

- MaskEdBox   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, the configuration file VBCTL3D.INI must contain an entry for this control. For more infomation 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** 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 grahical 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 any of the forns 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

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.

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.