Component Drag Overview

CmpDrag.ocx is a 32bit ocx file. The control will allow the programmer, to allow the end user to move components around a form at run time.

The Component Drag control does not move controls in the standard Drag and Drop method. The end user can simply click on the control specified in the **ControlName** property of the Component Drag control and reposition a control.

The Component Drag control can be used to move any control on a form that has the **Top** and **Left** properties.

For example, to move a Command Button on a form you would set the **ControlName** property of the Component Drag control to the name of the Command Button. CmpDrag1.ControlName = Command1.Name

Now when the Component Drag control is Enabled, and the end user presses and holds down the mouse button over the Command Button they can reposition the Command Button on the form.

If the Component Drag control is disabled or the **ControlName** is set to null the Command Button will act in its normal fashion.

Properties
Order Information
License Agreement
Example
Other Controls

Properties

ControlIndex
ControlName
Enabled

Order Information

To order the Component Drag Control send check or money order
for $15.00 U.S. dollars to

Steve C. Hetrick
P.O. Box 854
Crestview, Fl. 32536

OR

You can purchase and download the registered version of Component Drag at

https://www.regnow.com/softsell/nph-softsell.cgi?item=1125-4
qcsoft@cyou.com

ControlName

The ControlName property is a string value used to determine which control to move.

The property can be set in code or at design time.

See ControlIndex
See Enabled
See Example

Enabled

The Enabled property is a Boolean value that determines if the Component Drag control will Move the control specified in the ControlName property or ignore the control.

If set to True, the Component Drag Control will move the specified control in the ControlName property if the a Mouse Button is held down over the control specified and moved to a new location on the form.

See ControlIndex
See ControlName
See Example

License Agreement

This is a Shareware version of the Component Drag Control. You are licensed
to use the shareware version of Component Drag Control for evaluation
purposes only. You may not lease, sale, rent, or in any way use the shareware
version of the Component Drag control for profit.

You may not modify, translate,reverse engineer, decompile, or disassemble the
Component Drag Control.

THE PROGRAM AND ACCOMPANYING MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE AUTHOR OF COMPONENT DRAG CONTROL
DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE PROGRAM WILL MEET YOUR
REQUIREMENTS OR THAT THE OPERATION WILL BE UNINTERRUPTED OR ERROR FREE. THE
ENTIRE RISK AS TO THE USE, QUALITY, AND PERFORMANCE OF THE PROGRAM IS WITH YOU
THE USER.

IN NO EVENT WILL THE AUTHOR OF COMPONENT DRAG CONTROL BE LIABLE FOR ANY
CONSEQUENTIAL, INCIDENTAL, SPECIAL OR INDIRECT DAMAGES, INCLUDING LOSS OF DATA,
LOST PROFITS OR LOST SAVINGS ARISING FROM THE USE OF THE PROGRAM OR ANY OF ITS
ACCOMPANYING MATERIALS, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY.

BY USING THE COMPONENT DRAG CONTROL YOU ARE AGREEING TO THE TERMS OF THIS
LICENSE AGREEMENT.

Component Drag Example

```
'To begin, place 2 Command Button controls, 1 Label Control
' and 1 Component Drag control on Form1.
'Add the following code to Form1's General Declarations Section.

Private Sub Form_Load()
With Form1
   .Caption = "Component Drag"
   .Left = (Screen.Width - Me.Width) / 2
   .Top = (Screen.Height - Me.Height) /2
   .Height = 4545
   .Width = 5460
End With
With Command1
   .Caption = "Enable Move"
   .Top = 120
   .Left = 1800
   .Width = 1575
   .Height = 255
End With
With Command2
   .Caption = "Click For Message"
   .Top = 1680
   .Left = 1800
   .Width = 1575
   .Height = 375
End With
With Label1
   .Caption = "Component Drag"
   .Top = 1320
   .Left = 2040
   .Width = 1215
   .Height = 255
End With
End Sub

Private Sub Command1_Click()

   CmpDrag1.Enabled = True

End Sub

Private Sub Label1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

   CmpDrag1.ControlName = Label1.Name

End Sub

Private Sub Command2_Click()

    Label1.ForeColor = RGB(255 * Rnd, 255 * Rnd, 255 * Rnd)
```

```
        aval = Label1.Top 'Int(Label1.Top)
        bval = Command2.Top 'Int(Command2.Top)

    If aval < bval Then
        msg = "The label is positioned higher then the command button."
        msg = msg & "Try moving the label below the command button."
        MsgBox msg
    End If
    If aval > bval Then
        msg = "The Label is positioned lower then the command button."
        msg = msg & "Try moving the the command button below the Label"
        MsgBox msg
    End If

End Sub

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

    CmpDrag1.ControlName = Command2.Name

End Sub
```

Other Controls

Fader
Form Crony
EZCall
DScan
EZLink
QCBack Ground
QCWav
Wrapper

Fader

Fader is a 32bit ocx file that will allow you to fade a PictureBox picture 6 different ways.

Random - Fades the picture in a random fashion.
Outward - Fades the picture from the center to the outer edges.
LeftToRight - Fades the picture from the left to the right.
RightToLeft - Fades the picture from the right to the left.
TopDown - Fades the picture from the top to the bottom.
BottomUp - Fades the picture from the bottom to the top.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

EZCall

EZCall is a 32bit ocx file designed for Visual Basic. The control will
simplify some of the Windows API calls for Visual Basic programers. Instead
of making calls to the API try setting EZCall properties to true and get the
information from EZCall.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

Form Crony

Form Crony has several features to simplify some of the more common manipulation of a form. With Form Crony you can flash the forms title bar , center the form , make the form the topmost form on the screen , and you can also use Form Crony to size the controls on the form to fit the form as the forms size changes.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

DScan

DScan is a 32bit ocx file designed to scan all directories and sub directories for a file or file type. The DScan control will scan the specified drive and return the path to any file that is found on the drive that matches the file name or file type specified.

Each time a file matching the SearchName property is found the control will trigger a found event and the controls path property will be set to the path to the found file.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

EZLink

EZLink is a 32bit ocx file. EZLink can be added to an about box or any form to provide a link to your web page or any web page from your program. EZLink is similar to a Label control with the added feature of be able to launch the default browser and load the specified URL.

For more Information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

QCWav

QCWav is a 32bit ocx file designed to allow you
to play wav files.

Enter a path to a wav file in the QCWav's Path property
and set the Enabled property to True to play a wav file.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

Wrapper

Wrapper is a 32bit ocx file designed to allow you to print text to a PictureBox or Form and have the text wrap to fit the size of the PictureBox or Form.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

QCBack Ground

The QCBack Ground Control is a 32bit ocx file that will allow you to add a background effect to your forms like the back ground for the Windows desk top.

For more information go to
http://www.cyou.com/~shetrick/qchome.htm

qcsoft@cyou.com

ControlIndex

The ControlIndex property is an integer value used to determine which control to move if the control is part of a control array.

If the control to be moved is not part of an array the ControlIndex value will be ignored.

The property can be set in code or at design time.

EXAMPLE:
For a control array.
Under the MouseMove event of one of the controls in an array enter this code.

CmpDrag1.ControlIndex = Index
CmpDrag1.ControlName = "The name of the control"

See ControlName
See Enabled
See Example