

Forum:[Hack Forums](#)

Thread: [Custom Drag'n Drop Control](#)

VirinomTR - January 29, 2015, 12:55 am

Hi guys,

I made a useful custom control for drag'n drop operations and i thought many people can need it and i'm sharing it with.

General Features

Extension Filter

Can show a messagebox if dropped file's extension not allowed.

Enable/Disable Multiple File

Can show a messagebox when user trying to drag'n drop multiple file while it's not allowed.

Raise custom event after file dropped

Here's source code

Code Class DragDrop

```
Inherits ContainerControl
Private Shape As GraphicsPath
Private Multiple As Boolean
Private files As String()
Private filter As String()
Private msgMultiple As String
Private msgExtension As String
Public Event FileDropped As EventHandler
```

```
Sub New()
    SetStyle(ControlStyles.SupportsTransparentBackColor, True)
    SetStyle(ControlStyles.UserPaint, True)
    BackColor = Color.Transparent
    Me.Size = New Size(187, 117)
    Me.AllowDrop = True
    Padding = New Padding(1, 1, 1, 1)
    DoubleBuffered = True
End Sub
Property AllowMultiple() As Boolean
    Get
        Return Multiple
    End Get
```

```
Set(ByVal Value As Boolean)
    Multiple = Value
```

```

End Set
End Property
Property ExtensionFilter() As String()
    Get
        Return filter
    End Get
    Set(ByVal Value As String())
        filter = Value
    End Set
End Property
Property MultipleError() As String
    Get
        Return msgMultiple
    End Get
    Set(ByVal Value As String)
        msgMultiple = Value
    End Set
End Property
Property ExtensionError() As String
    Get
        Return msgExtension
    End Get
    Set(ByVal Value As String)
        msgExtension = Value
    End Set
End Property
ReadOnly Property DroppedFiles As String()
    Get
        Return files
    End Get
End Property
Private Sub Me_DragDrop(sender As Object, e As DragEventArgs) Handles Me.DragDrop
    Dim extMatch As Boolean = False
    Dim tmpArray = e.Data.GetData(DataFormats.FileDrop)
    If tmpArray.Length > 1 And Multiple = False Then
        If msgMultiple.Length > 0 Then
            MsgBox(msgMultiple, MsgBoxStyle.Critical, "Error")
        End If
    Else
        If filter.Length > 0 Then
            Dim i As Integer
            For i = 0 To tmpArray.Length - 1
                For Each ext As String In filter
                    If tmpArray(i).EndsWith(ext) Then
                        extMatch = True
                    End If
                Next
            Next
            If extMatch = True Then
                files = e.Data.GetData(DataFormats.FileDrop)
                RaiseEvent FileDropped(Me, e)
            Else

```

```

    If msgExtension.Length > 0 Then
        MsgBox(msgExtension, MsgBoxStyle.Critical, "Error")
    End If
End If
Else
    files = e.Data.GetData(DataFormats.FileDrop)
    RaiseEvent FileDropped(Me, e)
End If
End If
End Sub

```

```

Private Sub Me_DragEnter(sender As Object, e As DragEventArgs) Handles Me.DragEnter
    If e.Data.GetDataPresent(DataFormats.FileDrop) Then
        e.Effect = DragDropEffects.Copy
    End If
End Sub

```

```

Protected Overrides Sub OnResize(ByVal e As System.EventArgs)
    MyBase.OnResize(e)
    Shape = New GraphicsPath
    With Shape
        .AddArc(0, 0, 10, 10, 180, 90)
        .AddArc(Width - 11, 0, 10, 10, -90, 90)
        .AddArc(Width - 11, Height - 11, 10, 10, 0, 90)
        .AddArc(0, Height - 11, 10, 10, 90, 90)
        .CloseAllFigures()
    End With
End Sub

```

```

Protected Overrides Sub OnPaint(ByVal e As System.Windows.Forms.PaintEventArgs)
    MyBase.OnPaint(e)

```

```

    Dim p As New Pen(Color.FromArgb(180, 180, 180)) With {.DashStyle = DashStyle.Dash, .Width
= 5}

```

```

    Dim B As New Bitmap(Width, Height)
    Dim G = Graphics.FromImage(B)
    G.SmoothingMode = SmoothingMode.HighQuality
    G.Clear(Color.Transparent)
    G.FillPath(Brushes.White, Shape)
    G.DrawPath(p, Shape)

```

```

    Dim F As New Font("Trebuchet MS", 20, FontStyle.Bold)
    Dim sf As New StringFormat With {.LineAlignment = StringAlignment.Center, .Alignment =
StringAlignment.Center}
    G.DrawString(Me.Text, F, New SolidBrush(Color.FromArgb(221, 221, 221)), ClientRectangle,
sf)

```

```

    G.Dispose()
    e.Graphics.DrawImage(B.Clone(), 0, 0)
    B.Dispose()
End Sub

```

End Class

How to use properties.

Code DragDrop1.AllowMultiple=False 'This will disable multiple file dropping.

DragDrop1.MultipleError="Only 1 file can be drop." 'Leave blank for no messagebox.

DragDrop1.ExtensionFilter = {".exe", ".jpg", ".png"} 'Use string array with this format.Leave blank for no filter.

DragDrop1.ExtensionError = "This file type not allowed." 'Leave blank for no messagebox.

Dim files as string() = DragDrop1.DroppedFiles() 'This property will always return dropped files as string array whether allow multiple selected true or false.

'You can use DragDrop1.DroppedFiles(0) if you set AllowMultiple to false.

How to use custom event.

Code Private Sub DragDrop1_FileDropped(sender As Object, e As EventArgs) Handles DragDrop1.FileDropped

'This event only raised if no multiple selection error or extension error occurred.

End Sub

Credits

AeonHack : I looked his themebase and learned how to desing controls.

Josh? - January 29, 2015, 7:56 am

Not tested it but seems nice. May use it on a later project. ([Click to View](#)) Posting to bookmark for said project

SirenNinja - January 29, 2015, 8:31 am

This looks very good. Thanks for the share. :D

realrnb - January 29, 2015, 8:44 am

Looks really nice, good Job OP! I will propably need this sometime :)

Only3 - January 29, 2015, 9:08 am

i made one much simple <https://hackforums.net/showthread.php?tid=4240622>

SIM-1456245 - January 29, 2015, 11:59 am

(Jan 29, 2015 10:08 AM)Only3 Wrote: [\[CITE\]](#) i made one much simple

<https://hackforums.net/showthread.php?tid=4240622>

It's nice, but stop it with the whole "simplicity" thing that you are having.

SubGenius - January 29, 2015, 12:26 pm

(Jan 29, 2015 10:08 AM)Only3 Wrote: [\[CITE\]](#) i made one much simple

<https://hackforums.net/showthread.php?tid=4240622>

This is what you coded:

Spoiler ([Click to View](#))

[img]http://www.moffatcan.org/shop/wp-content/uploads/2014/09/greek-potato.standard-460x345.gif[/img]

and this what OP made:

Spoiler ([Click to View](#))

[img]http://news.sciencemag.org/sites/default/files/styles/thumb_article_/public/media/sn-gravity.JPG?itok=ld_2QTiS[/img]

Are they the same? Yeah they are.

Although french fries look tastier.

[Add only 3 emoticons here:]
??????????????

SIM-1456245 - January 29, 2015, 12:31 pm

(Jan 29, 2015 01:26 PM)SubGenius Wrote: [\[CITE\]](#) This is what you coded:

Spoiler [\(Click to View\)](#)

[img]http://www.moffatcan.org/shop/wp-content/uploads/2014/09/greek-potato.standard-460x345.gif[/img]

and this what OP made:

Spoiler [\(Click to View\)](#)

[img]http://news.sciencemag.org/sites/default/files/styles/thumb_article_l/public/media/sn-gravity.JPG?itok=ld_2QTiS[/img]

Are they the same? Yeah they are.
Although french fries look tastier.

[Add only 3 emoticons here:]
??????????????

Best reply ever. Made my day. This is like 100/100 true.

Hawk HF - January 29, 2015, 12:35 pm

Cool release. I haven't tested or tried this, but I trust that it works. It's good to see someone learning by creating something constructive instead of jumping in the .NET section and asking how to FUD a crypter or code a RAT. For that, I'm going to take some time to help you out.

Here are suggestions that will improve your code:

1. Properties

Properties are used to perform actions when retrieving or setting data or for making a value read-only or write-only. Most of your properties don't do either of these, so it would save time, effort, and lines of code if you simply used a public variable. If you still insist on using properties, you can also do a shorthand (automatic) property so that you don't need to absorb the lines for your get and set blocks and variable declarations.

2. Overriding the OnPaint Subroutine

There is no need to call the MyBase.OnPaint() method here since you're painting everything yourself. Although it will be minimal, removing that line will reduce the amount of time it takes to invalidate your control. Also, you don't need to use a bitmap object. A graphics object is already there for you in the PaintEventArgs argument. Remove the bitmap and instead set your graphics variable g to e.Graphics.

3. Organization

In my opinion, the most beneficial feature of object oriented programming is that it's modular. This means that your methods can be reused by yourself as well as other programmers. It would be good for you to get in the habit of organizing your code and making it look good. Keep line breaks between your methods and try to group lines of code that relate to each other.

Good luck with your future projects.
I hope to see more from you.

?

VirinomTR - January 29, 2015, 12:46 pm

(Jan 29, 2015 01:35 PM)Hawk HF Wrote: [\[CITE\]](#) Cool release. I haven't tested or tried this, but I trust that it works. It's good to see someone learning by creating something constructive instead of jumping in the .NET section and asking how to FUD crypter or code a RAT. For that, I'm going to take some time to help you out.

Here are suggestions that will improve your code:

1. Properties

Properties are used to perform actions when retrieving or setting data or for making a value read-only or write-only. Most of your properties don't do either of these, so it would save time, effort, and lines of code if you simply used a public variable. If you still insist on using properties, you can also do a shorthand (automatic) property so that you don't need to absorb the lines for your get and set blocks and variable declarations.

2. Overriding the OnPaint Subroutine

There is no need to call the MyBase.OnPaint() method here since you're painting everything yourself. Although it will be minimal, removing that line will reduce the amount of time it takes to invalidate your control. Also, you don't need to use a bitmap object. A graphics object is already there for you in the PaintEventArgs argument. Remove the bitmap and instead set your graphics variable g to e.Graphics.

3. Organization

In my opinion, the most beneficial feature of object oriented programming is that it's modular. This means that your methods can be reused by yourself as well as other programmers. It would be good for you to get in the habit of organizing your code and making it look good. Keep line breaks between your methods and try to group lines of code that relate to each other.

Good luck with your future projects.
I hope to see more from you.

?

Thanks so much for nice comments. It's motivate me for working on same project. I will make another version according to your suggestions. Thanks again.

titanbullet - January 30, 2015, 12:13 pm

I love it! I will use it in my next program for sure!

CodeAssassin - January 31, 2015, 8:39 am

Just a question why have you made a custom drag and drop control rather than just adding it to previous controls? IE: textbox etc or even taking any control in the drag and drop control to add functionality to other controls. I just code per use. but nice share for the rest of the community.

VirinomTR - January 31, 2015, 12:18 pm

(Jan 31, 2015 09:39 AM)sweetysuckscock Wrote: [\[CITE\]](#) Just a question why have you made a custom drag and drop control rather than just adding it to previous controls? IE: textbox etc or even taking any control in the drag and drop control to add functionality to other controls. I just code per use. but nice share for the rest of the community.

Yeah i can also do it with this way but i like making something new. And this seems better and easy to redesign for different themes.