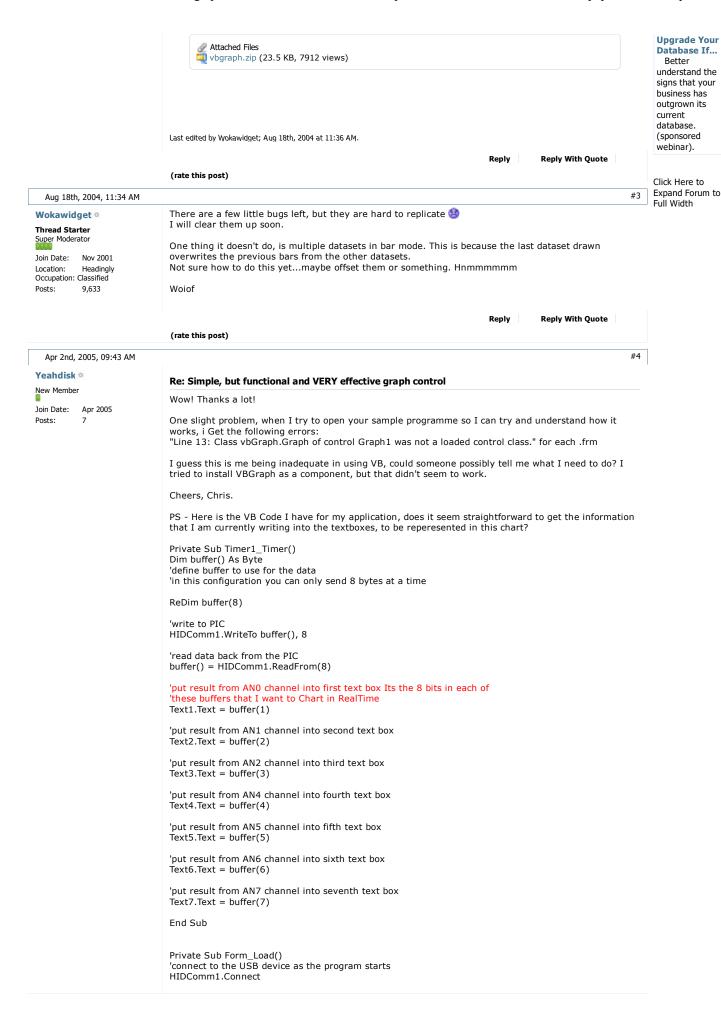
PHP User Warning: fetch_template() calls should be replaced by the vB_Template class. Template name: bbcode_highlight in/includes/functions.php on line 4197 PHP User Warning: fetch_template() calls should be replaced by the vB_Template class. Template name: bbcode_highlight in/includes/functions.php on line 4197 A Developer.com Site Welcome, dz32 Notifications My Profile Settings Log Out Custom Search Forums Advertiser Disclosure What's New? Forum New Posts Private Messages FAQ Calendar Community Forum Actions Quick Links Advanced Search VBForums VBForums CodeBank CodeBank - Visual Basic 6 and earlier Simple, but functional and VERY effective graph control Page 1 of 2 1 2 1 Last >> Results 1 to 40 of 49 + Reply to Thread **Featured** Thread: Simple, but functional and VERY effective graph control → *new* **Replace Your** Thread Tools Search Thread Rate This Thread Display Oracle **Database and** Aug 14th, 2004, 03:46 PM #1 **Deliver the** Personalized, Wokawidget 9 Simple, but functional and VERY effective graph control Responsive Thread Starter **Experiences** I have written a small compact graph usercontrol control. Super Moderator **Customers** Crave Join Date: Nov 2001 Features: Get practical Location: • Squigillians of points advice and learn Occupation: Classified Draw Line Graph best practices for Posts: 9,633 • Draw Bar Graph moving your · All Custom Colors applications from • Draw Points RDBMS to the Draw Axis Couchbase · Fixed Items Engagement Save ALL grid and point properties to a file Database. • Load a graph from a saved file (sponsored) → Unleash I think that's it. Your DevOps With the graph it's possible to do the following: Strategy by · Draw a moving graph, like you see in task manager **Synchronizing** • Draw a histogram, bar and line graphs App and • Draw mathematical equations, like Sin Wave or x^2 **Database** • Draw a moving music graphic equaliser thingy like in media player Changes · Create a progress bar Learn to shorten database It's something I knocked up in the last 24hrs. I can't find any bugs, but I am sure one of the little dev cycles, bastards is still lurking around 💮 integrate code quality reviews Any comments or suggestions would be great. into Continuous Integration Cheers. workflow, and deliver code 40% Woka faster. (sponsored) → Build Attached Files Planet-Scale 闻 vbgraph.zip (11.3 KB, 17203 views) Apps with Azure Cosmos **DB** in Minutes See a demo showing how you can build a globally distributed, Last edited by Wokawidget; Aug 16th, 2004 at 04:09 AM. planet-scale apps Reply **Reply With Quote** in minutes with Azure Cosmos (rate this post) DB. (sponsored webinar) Aug 18th, 2004, 11:18 AM #2 Comprehensive Wokawidget o Version 2.0 **Guide to Cloud** Thread Starter Computing Super Moderator OK. Here's Version 2.0 of the control. A complete It can now handle multiple datasets 🌐 overview of Join Date: Nov 2001 Cloud Computing Location: Headingly This allows you to add even more functionality...check out the music equalizer demo 🙂 focused on what Occupation: Classified you need to Posts: 9.633 This demo show's you most of the styles and functionality you can do with this control, except the know, from Loading and Saving of the graph. selecting a I will write a form and add it into the demo to do this. platform to choosing a cloud Wooooof vendor. → It Might be Time to



```
End Sub
Private Sub Form_Terminate()
'disconnect from \overset{-}{\operatorname{the}} USB device as prgoram ends
'THIS IS IMPORTANT
HIDComm1.Uninit
End Sub
Private Sub HIDComm1_ConnectSuccess(ByVal Status As Long)
Caption = "Huzzah! - Its Connected"
End Sub
Private Sub HIDComm1_Disconnected(ByVal Status As Long)
Caption = "Not Connected"
End Sub
Private Sub Timer2_Timer()
'try and reconnect the PIC
If HIDComm1.Connected = False Then
HIDComm1.Connect
End If
End Sub
```

(rate this post) #5

Reply

Reply With Quote

Apr 3rd, 2005, 02:31 AM

CVMichael •

PowerPoster

Join Date: Feb 2002 Location: Canada, Toronto Posts: 5.792

Re: Simple, but functional and VERY effective graph control

Hi Wokawidget,

I did not look at your usercontrol, but reading about it, I got an idea. In this thread http://www.vbforums.com/showthread.php?t=328242, I did some reserch on fixing the distorted sound, and i got the solution with the trend thing, I tought you can use the same thing to draw graphs...

Anyways, I wrote some code to see what I mean, put a picturebox on the form, and paste the following code to see it in action.

The only functions I wrote are: DrawTrend, ExpandPoints, ExpandPointsB, and how to use them in $Form_Load$.

Maybe you can incorporate this in your graphing usercontrol...

VB Code:

```
1. Option Explicit
3. Private Type POINT
4.
       X As Double
        Y As Double
6. End Type
8. Private Sub Form_Load()
9.
       Dim SinglePoints(4) As Double, InTrendPoints() As POINT, OutTrendPoints(
10.
11.
        SinglePoints(0) = 10
12.
        SinglePoints(1) = 35
        SinglePoints(2) = 50
13.
        SinglePoints(3) = 18
14.
15.
        SinglePoints(4) = 13
16.
        ^{\mbox{\tiny I}} increase the "ExpandTo" param to have even more precision
17.
        InTrendPoints = ExpandPoints(SinglePoints, 60)
18.
19.
        'InTrendPoints = ExpandPointsB(SinglePoints, 60) ' a simpler expand...
        OutTrendPoints = Trend(InTrendPoints, 3)
20.
21.
22.
        Picture1.AutoRedraw = True
23.
        DrawTrend Picture1, InTrendPoints, RGB(180, 180, 180)
24.
        DrawTrend Picturel, OutTrendPoints, vbBlue
25. End Sub
27. Private Sub DrawTrend(Pic As PictureBox, Points() As POINT, ByVal Color As I
       Dim MaxY As Double, MaxX As Double, K As Long, EmptySpaceX As Double, Em
30.
        For K = 0 To UBound(Points)
           If Points(K).X > MaxX Then MaxX = Points(K).X
            If Points(K).Y > MaxY Then MaxY = Points(K).Y
32.
33.
        Next K
34.
35.
        EmptySpaceX = MaxX * 0.1
36.
        EmptySpaceY = MaxY * 0.1
37.
38.
        Pic.Scale (-EmptySpaceX, MaxY + EmptySpaceY) - (MaxX + EmptySpaceX, -Empty
```

```
Picture1.DrawWidth = 1
         Pic.Line (-EmptySpaceX, 0)-(MaxX + EmptySpaceX, 0), RGB(128, 128, 128)
         Pic.Line (0, MaxY + EmptySpaceY) - (0, -EmptySpaceY), RGB(128, 128, 128)
 43.
         Picture1.DrawWidth = 2
 44.
45.
        Pic.PSet (Points(0).X, Points(0).Y), Color
 46.
 47.
        If WithPoints Then
          For K = 1 To UBound (Points)
 48.
 49.
                 Pic.PSet (Points(K).X, Points(K).Y), Color
5.0
            Next K
51.
        Else
52
            For K = 1 To UBound (Points)
                Pic.Line - (Points(K).X, Points(K).Y), Color
53.
54.
            Next K
        End If
55.
56. End Sub
57.
58. Private Function ExpandPointsB(InPoints() As Double, ExpandTo As Long) As PO
        Dim OutPoints() As POINT, K As Long, Per As Double, X As Long
59.
60.
         If ExpandTo <= 2 Then Exit Function</pre>
 61.
        ReDim OutPoints (ExpandTo - 1)
 63.
        For K = 0 To UBound (OutPoints)
             Per = K / CDbl(UBound(OutPoints))
             X = Round(UBound(InPoints) * Per)
             OutPoints(K).X = K + 1
69.
             OutPoints(K).Y = InPoints(X)
 70.
         Next K
71.
72.
         ExpandPointsB = OutPoints
73. End Function
75. Private Function ExpandPoints(InPoints() As Double, ExpandTo As Long) As POI
76
        Dim OutPoints() As POINT, K As Long, Per As Double, PerX As Double, LngX
77.
 78.
         If ExpandTo <= 2 Then Exit Function</pre>
 79.
         ReDim OutPoints (ExpandTo - 1)
 80.
        For K = 0 To UBound (OutPoints) - 1
81.
             Per = K / CDbl (UBound (OutPoints))
 82.
 83.
             PerX = UBound (InPoints) * Per
 84.
             LngX = Fix(PerX)
 86.
             OutPoints(K).X = K + 1
             OutPoints(K).Y = InPoints(LngX) + (InPoints(LngX + 1) - InPoints(LngX
        Next K
         OutPoints(K).X = K + 1
         OutPoints(K).Y = InPoints(UBound(InPoints))
         ExpandPoints = OutPoints
 94.
95. End Function
 97. Private Function Trend(Data() As POINT, ByVal Degree As Long) As POINT()
98.
         'degree 1 = straight line y=a+bx
99.
        'degree n = polynomials!!
100.
101.
        Dim a() As Double
102.
        Dim Ai() As Double
103.
         Dim b() As Double
104.
        Dim P() As Double
105.
         Dim SigmaA() As Double
106.
        Dim SigmaP() As Double
         Dim PointCount As Long
107.
108.
        Dim MaxTerm As Long
109.
        Dim m As Long, n As Long
Dim i As Long, J As Long
110.
111.
        Dim Ret() As POINT
112.
113.
         Degree = Degree + 1
114.
         MaxTerm = (2 * (Degree - 1))
116.
         PointCount = UBound(Data) + 1
118.
         ReDim SigmaA(MaxTerm - 1)
         ReDim SigmaP(MaxTerm - 1)
120.
         ' Get the coefficients lists for matrices A, and P
121.
122.
         For m = 0 To (MaxTerm - 1)
123.
             For n = 0 To (PointCount - 1)
                 \label{eq:sigmaA(m) = SigmaA(m) + (Data(n).X ^ (m + 1))} SigmaP(m) = SigmaP(m) + ((Data(n).X ^ m) * Data(n).Y)
124.
125.
126.
            Next
127
        Next
128.
129.
          ' Create Matrix A, and fill in the coefficients
130.
         ReDim a(Degree - 1, Degree - 1)
```

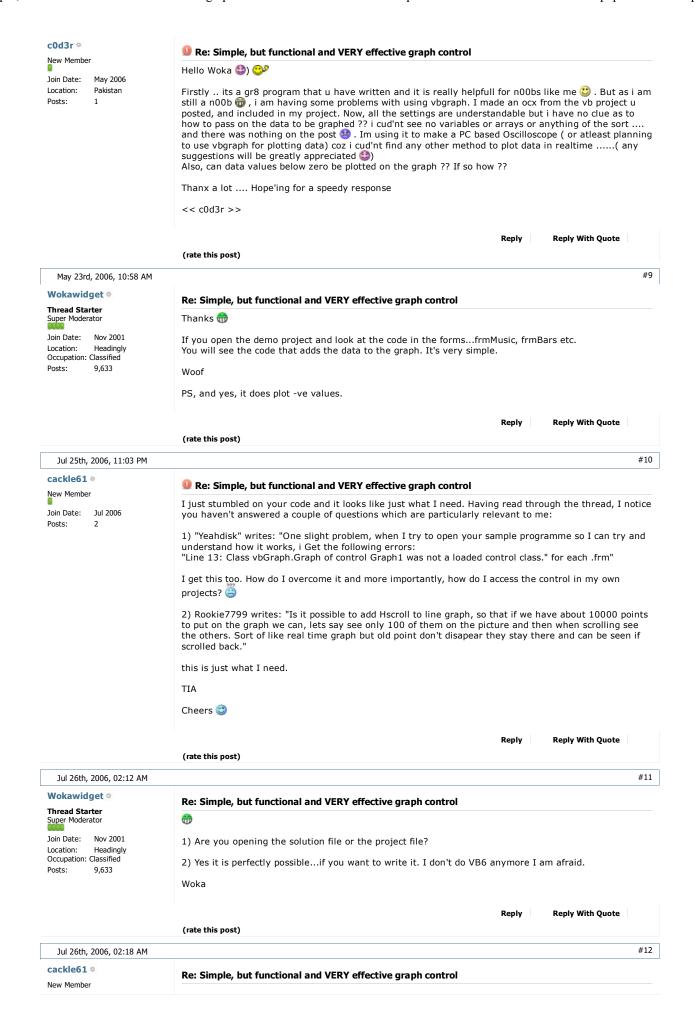
```
For i = 0 To (Degree - 1)
132.
            For J = 0 To (Degree - 1)
133.
                If i = 0 And J = 0 Then
134.
                     a(i, J) = PointCount
135.
136.
                   a(i, J) = SigmaA((i + J) - 1)
137.
                 End If
138.
            Next
139.
        Next
140.
141.
         ^{\mbox{\tiny L}} Create Matrix P, and fill in the coefficients
         ReDim P(Degree - 1, 0)
142
        For i = 0 To (Degree - 1)
143.
144
          P(i, 0) = SigmaP(i)
145.
146.
         ' We have A, and P of AB=P, so we can solve B because B=AiP
147.
        Ai = MxInverse(a)
148.
        b = MxMultiplyCV(Ai, P)
149.
150.
         ' Now we solve the equations and generate the list of points
151.
         PointCount = PointCount - 1
152.
         ReDim Ret (PointCount)
153.
         ' Work out non exponential first term
155.
        For i = 0 To PointCount
         Ret(i).X = Data(i).X
            Ret(i).Y = b(0, 0)
160.
161.
        ' Work out other exponential terms including exp 1
162.
        For i = 0 To PointCount
163.
            For J = 1 To Degree - 1
164.
                Ret(i).Y = Ret(i).Y + (b(J, 0) * Ret(i).X ^ J)
165.
            Next
166.
         Next
167.
168
         Trend = Ret
169. End Function
170.
171. Private Function MxMultiplyCV(Matrix1() As Double, ColumnVector() As Double)
172.
        Dim i As Long
173.
        Dim J As Long
174.
        Dim Rows As Long
175.
        Dim Cols As Long
176.
        Dim Ret() As Double
177.
178.
         Rows = UBound (Matrix1, 1)
        Cols = UBound (Matrix1, 2)
181.
        ReDim Ret(UBound(ColumnVector, 1), 0) 'returns a column vector
        For i = 0 To Rows
            For J = 0 To Cols
184.
                Ret(i, 0) = Ret(i, 0) + (Matrix1(i, J) * ColumnVector(J, 0))
185.
186.
            Next
187.
188.
189.
         MxMultiplyCV = Ret
190. End Function
191.
192. Private Function MxInverse(Matrix() As Double) As Double()
193.
        Dim i As Long
194.
        Dim J As Long
195.
        Dim Rows As Long
196.
        Dim Cols As Long
197.
        Dim Tmp() As Double
        Dim Ret() As Double
198.
199.
        Dim Degree As Long
200.
201.
        Tmp = Matrix
202.
203.
         Rows = UBound (Tmp, 1)
         Cols = UBound (Tmp, 2)
204.
205.
         Degree = Cols + 1
206.
207.
         'Augment Identity matrix onto matrix M to get [M|I]
208.
         ReDim Preserve Tmp(Rows, (Degree * 2) - 1)
        For i = Degree To (Degree * 2) - 1
209.
210.
            Tmp(i Mod Degree, i) = 1
212.
         ^{\prime} Now find the inverse using Gauss-Jordan Elimination which should get u
213.
214.
215.
216.
         ' Copy the inverse (A-1) part to array to return
217.
        ReDim Ret(Rows, Cols)
218.
         For i = 0 To Rows
219.
            For J = Degree To (Degree * 2) - 1
220.
                Ret(i, J - Degree) = Tmp(i, J)
221.
            Next
222.
         Next
```

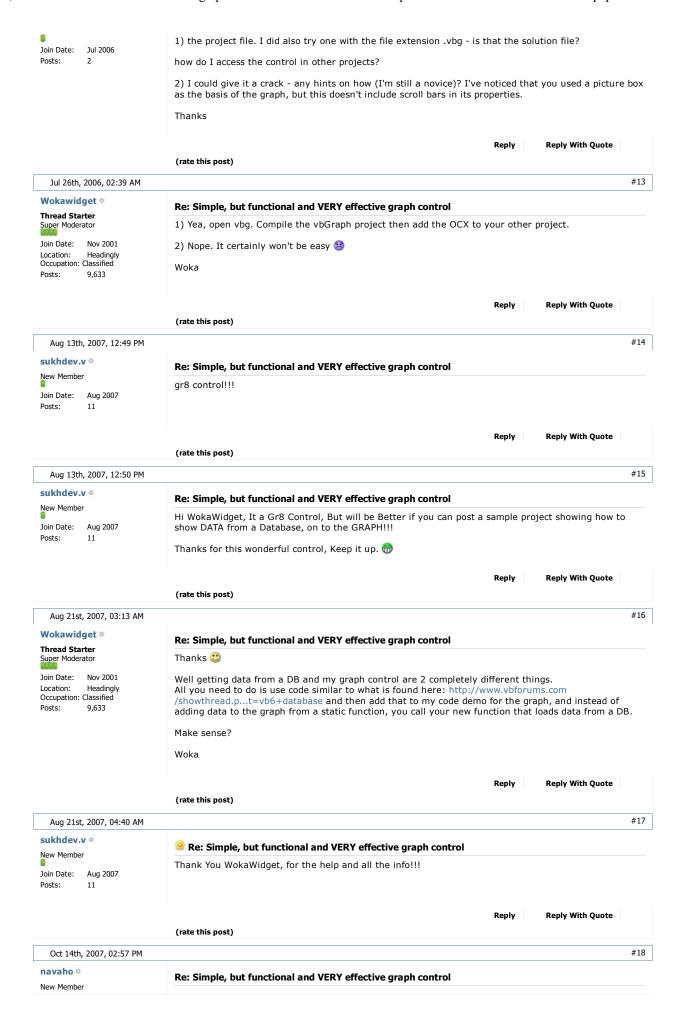
Join Date:

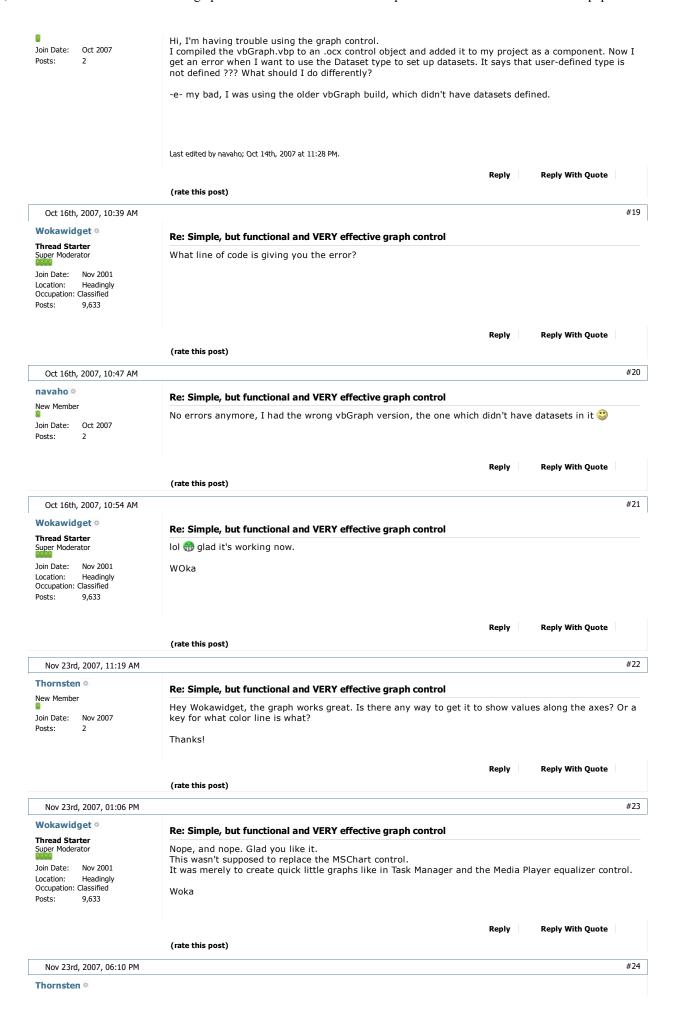
Location:

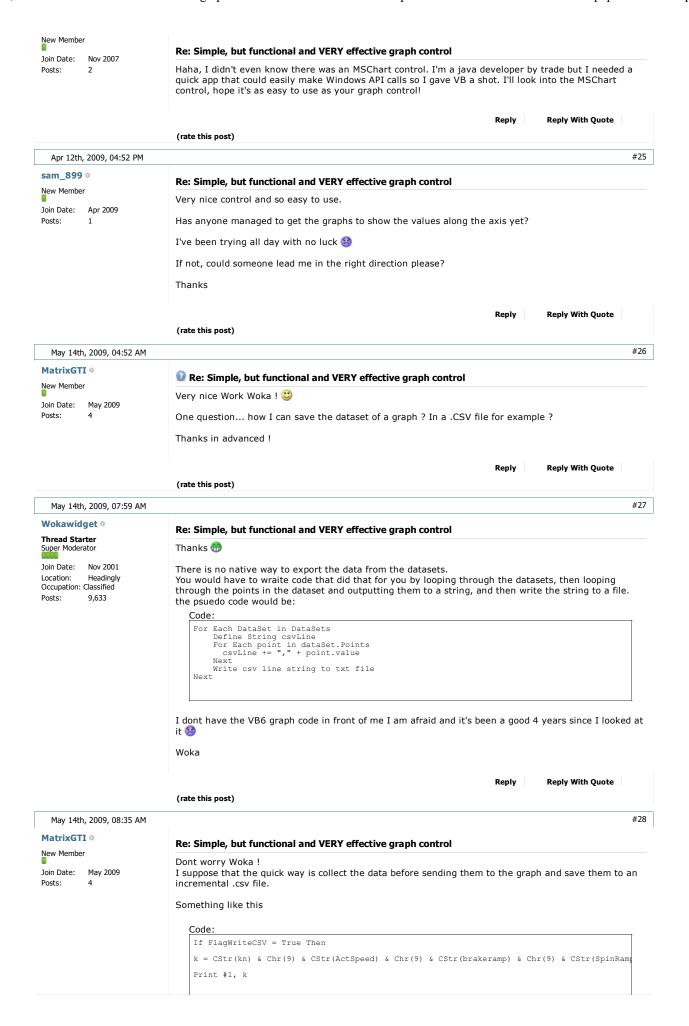
Posts:

```
224.
                                           MxInverse = Ret
                                 225. End Function
                                 226.
                                 227. Private Sub MxGaussJordan(Matrix() As Double)
                                           Dim Rows As Long
                                 228.
                                 229.
                                           Dim Cols As Long
                                 230.
                                           Dim P As Long
                                 231.
                                           Dim i As Long
                                 232.
                                           Dim J As Long
                                 233.
                                           Dim m As Double
                                 234
                                           Dim d As Double
                                 235.
                                           Dim Pivot As Double
                                 236
                                 237.
                                           Rows = UBound (Matrix, 1)
                                           Cols = UBound (Matrix, 2)
                                 238.
                                 239.
                                            ' Reduce so we get the leading diagonal
                                 240.
                                           For P = 0 To Rows
                                 241.
                                 242.
                                                Pivot = Matrix(P, P)
                                 243.
                                                For i = 0 To Rows
                                 244.
                                                     If Not P = i Then
                                 245.
                                                          m = Matrix(i, P) / Pivot
                                 246.
                                                                                                     Reply
                                                                                                                 Reply With Quote
                              (rate this post)
  Apr 3rd, 2005, 08:28 AM
                                                                                                                                     #6
Wokawidget 9
                              Re: Simple, but functional and VERY effective graph control
Thread Starter
Super Moderator
                              Very nice code indeed!
          Nov 2001
                              Should be quite easy to add in.
          Headingly
                              All that should be need, at 1st glance, if a function to take the data from one dataset, and pass it
Occupation: Classified
                              through those functions above, and return another dataset with many many point (so it looks like a
          9,633
                              curve).
                              Woka
                                                                                                     Reply
                                                                                                                 Reply With Quote
                              (rate this post)
                                                                                                                                     #7
  Aug 14th, 2005, 06:03 AM
rookie7799 o
                              Re: Version 2.0
Lively Member
                                    R Originally Posted by Wokawidget
Join Date: Feb 2005
          115
                                    OK. Here's Version 2.0 of the control.
                                    It can now handle multiple datasets 📆
                                    This allows you to add even more functionality...check out the music equalizer demo 🙂
                                    This demo show's you most of the styles and functionality you can do with this control, except the
                                    Loading and Saving of the graph.
                                    I will write a form and add it into the demo to do this.
                                    Wooooof
                              This is one great program! Thanks a lot for sharing it.
                              I have couple of questions tho 😃
                              1. Is it possible to add Hscroll to line graph, so that if we have about 10000 points to put on the graph
                              we can, lets say see only 100 of them on the picture and then when scrolling see the others. Sort of
                              like real time graph but old point don't disapear they stay there and can be seen if scrolled back.
                              2. Zoom in, zoom out 3 Maybe?
                              3. A small tutorial on how to do this of course not the whole thing just to have a idea...
                              4. Make the graph interactive. show the points, label them when mouse point on one it will show it's
                              coordinates
                              5. Just kidding m that's a lot to ask already.
                              GREAT JOB MAN 🎥
                                                                                                                 Reply With Quote
                                                                                                     Reply
                              (rate this post)
                                                                                                                                     #8
  May 23rd, 2006, 05:19 AM
```

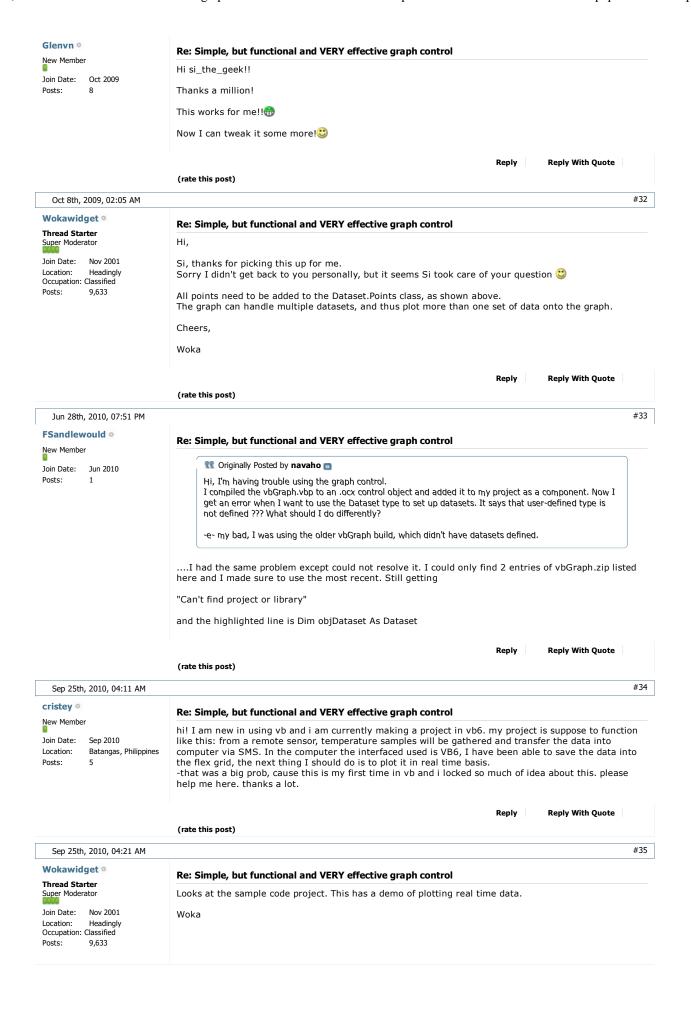


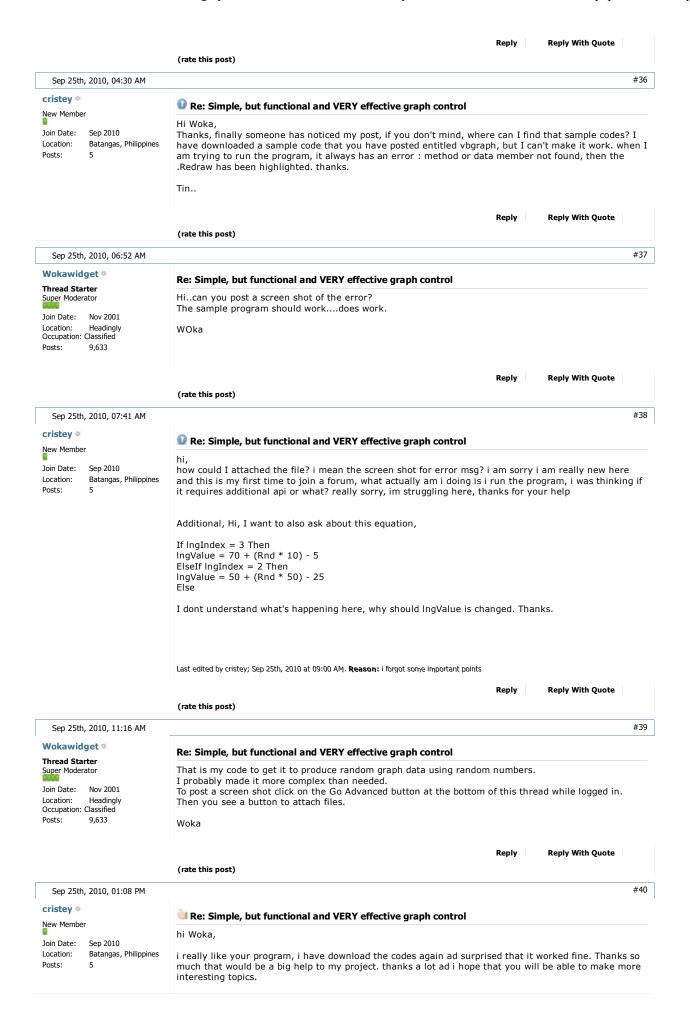


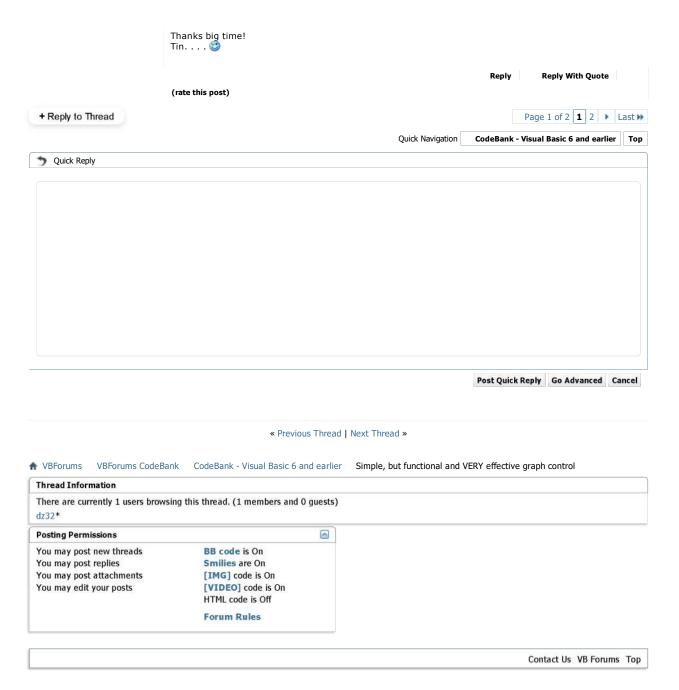




```
kn = kn + 1
                                 Label17.Caption = "Logging:" & " " & k
                                 startlogging.BackColor = vbGreen
                                 Else
                                 stoplogging.BackColor = vbRed
                                 End If
                                                                                                    Reply
                                                                                                               Reply With Quote
                             (rate this post)
 Oct 1st, 2009, 07:07 AM
                                                                                                                                  #29
Glenvn o
                             Re: Simple, but functional and VERY effective graph control
New Member
Join Date:
          Oct 2009
                             Thank you for your code! Its just what I am looking for!
                             I am trying to use your line graph but I am struggling...
                             The data source is a listbox - list2 (There are 47 numeric entries)
                             How do I make the graph use these entries?( without scrolling function)
                             I have this so far:
                                Code:
                                  1. Private Sub SetupDatasets()
                                  3. Dim objDataset As Dataset
                                  4. Dim list As Integer
                                  6.
                                           Set objDataset = Graph1.Datasets.Add
                                  8.
                                           With objDataset
                                               .Visible = False
                                 10.
                                               .ShowPoints = True
                                 11.
                                               .ShowLines = True
                                 12.
                                               .ShowBars = False
                                 13.
                                               .ShowCaps = False
                                 14.
                                               .LineColor = RGB(0, 0, 255)
                                 16.
                                                .PointColor = RGB(150, 150, 255)
                                           End With
                                           For i = 0 To List2.ListCount
                                           list = List2.list(i)
                                 22.
                                               Next i
                                 23.
                                         list = objDataset
                             I know this is old but it is still good code and it gets the job done!
                             Any help would be appreciated!
                                                                                                    Reply
                                                                                                               Reply With Quote
                              (rate this post)
                                                                                                                                  #30
 Oct 1st, 2009, 02:15 PM
si_the_geek o
                             Re: Simple, but functional and VERY effective graph control
Super Moderator
                             Welcome to VBForums 💚
Join Date:
          Jul 2002
Location:
          Bristol, UK
                             Based on the example project given (but no testing), I think this is what you want:
Posts:
          41,279
                                     For i = 0 To List2.ListCount objDataset.Points.Add CInt(List2.list(i))
Next i
                                                                                                    Reply
                                                                                                               Reply With Quote
                             (rate this post)
                                                                                                                                  #31
  Oct 2nd, 2009, 01:04 AM
```







Acceptable Use Policy



A Developer.com Property

Terms of Service | Licensing & Reprints | About Us | Privacy Notice | Contact Us | Advertise | | California - Do Not Sell My Info Copyright 2020 QuinStreet Inc. All Rights Reserved.

Advertiser Disclosure: Some of the products that appear on this site are from companies from which QuinStreet receives compensation. This compensation may impact how and where products appear on this site including, for example, the order in which they appear. QuinStreet does not include all companies or all types of products available in the marketplace.

All times are GMT -5. The time now is 08:53 AM.