Read-Me Graph Paper Templates

These templates are graph paper set to certain grids.

The graph papers were prepared using a macro published long ago by Word MVP Bill Coan. His article can be found in web archives here: [Quick, Get Me Some Graph Paper!](http://web.archive.org/web/20070421084938/http:/pubs.logicalexpressions.com/Pub0009/LPMArticle.asp?ID=596) <http://web.archive.org/web/20070421084938/http://pubs.logicalexpressions.com/Pub0009/LPMArticle.asp?ID=596>

This package can be found on my [Downloads Page](http://addbalance.com/word/download.htm#graph). This is Version 19.1. Copyright © 2019 Charles Kenyon, Sun Prairie, WI, USA

Permission granted to copy and redistribute without charge so long as entire package (zip folder is distributed)

# Here is my modified version of the macro:

Sub GraphPaperCreate()

' TechTrax - Bill Coan

' http://web.archive.org/web/20070421084938/http://pubs.logicalexpressions.com/Pub0009/LPMArticle.asp?ID=596

' modified by Charles Kenyon to:

' reduce margins to .25

' lighter line weight (.3 points rather than 2 points)

' grid is set behind text in case there is text

'

' Sets margins at .25" Charles Kenyon

'

With ActiveDocument.PageSetup

.TopMargin = InchesToPoints(0.25)

.BottomMargin = InchesToPoints(0.25)

.LeftMargin = InchesToPoints(0.25)

.RightMargin = InchesToPoints(0.25)

End With

'Declare variables

Dim sglBeginX As Single

Dim sglEndX As Single

Dim sglBeginY As Single

Dim sglEndY As Single

Dim Interval As Single

Dim LineCount As Integer

Dim VerticalGridLines As Integer

Dim HorizontalGridLines As Integer

Dim GridWidth As Single

Dim GridHeight As Single

Dim UserChoice As String

Dim oLine As Shape

'set the line weight, in points

Const LINE\_WEIGHT As Long = 0.5 'changed from Integer = 2 by Charles Kenyon - this is the minimum

'set whether to delete the old grid,

'(assuming there is one)

Const DELETE\_OLD\_GRID As Boolean = True

'Delete the old grid

If DELETE\_OLD\_GRID Then

On Error Resume Next

ActiveDocument.range.ShapeRange.Delete

On Error GoTo 0

End If

'Prompt the user to specify the size of squares desired

PromptUser:

UserChoice = InputBox \_

("Enter size of squares in points:" & vbLf & vbLf & \_

"(Value must be between 1 and 72)" & vbLf & \_

"9 points = 1/8 inch" & vbLf & \_

"12 points = 1/6 inch" & vbLf & \_

"18 points = 1/4 inch" & vbLf & \_

"24 points = 1/3 inch" & vbLf & \_

"36 points = 1/2 inch" & vbLf & \_

"48 points = 2/3 inch" & vbLf & \_

"72 points = one inch", "Create graph paper")

'If the size requested is outside the allowed range,

'then go back and prompt the user again,

'but if user has clicked cancel, then quit

If IsNumeric(UserChoice) Then

Interval = Val(UserChoice)

If Interval > 72 Or Interval < 2 Then

MsgBox "Value must be between 1 and 72.", \_

vbOKOnly + vbInformation, "Value out of range"

GoTo PromptUser

End If

ElseIf UserChoice = "" Then

GoTo EndGracefully

Else

GoTo PromptUser

End If

'Calculate beginning and ending positions

'based on page size and margins

With ActiveDocument.Sections.First.PageSetup

sglBeginX = .LeftMargin

sglEndX = .PageWidth - .RightMargin

sglBeginY = .TopMargin

sglEndY = .PageHeight - .BottomMargin

End With

'Calculate how many squares of the requested size

'can fit in the available area

HorizontalGridLines = Int((sglEndY - sglBeginY) / Interval)

VerticalGridLines = Int((sglEndX - sglBeginX) / Interval)

'Calculate the size of the finished grid, based on

'size of the number of lines and the size of the squares

GridWidth = VerticalGridLines \* Interval

GridHeight = HorizontalGridLines \* Interval

'draw the horizontal lines

For LineCount = 0 To HorizontalGridLines

Set oLine = ActiveDocument.Shapes.AddLine( \_

BeginX:=sglBeginX, \_

BeginY:=sglBeginY + LineCount \* Interval, \_

EndX:=sglBeginX + GridWidth, \_

EndY:=sglBeginY + LineCount \* Interval)

oLine.line.Weight = LINE\_WEIGHT

oLine.ZOrder msoSendBehindText 'Charles Kenyon

Next LineCount

'draw the vertical lines

For LineCount = 0 To VerticalGridLines

Set oLine = ActiveDocument.Shapes.AddLine( \_

BeginX:=sglBeginX + LineCount \* Interval, \_

BeginY:=sglBeginY, \_

EndX:=sglBeginX + LineCount \* Interval, \_

EndY:=sglBeginY + GridHeight)

oLine.line.Weight = LINE\_WEIGHT

oLine.ZOrder msoSendBehindText 'Charles Kenyon

Next LineCount

'release the oLine object from memory

Set oLine = Nothing

'end gracefully

EndGracefully:

End Sub

[Instructions for Installing Macros from Forums or Websites by Graham Mayor, MVP](http://www.gmayor.com/installing_macro.htm)