

```

#Const clMaxRed = MAXRGB
#Const clMaxGreen = MAXRGB
#Const clMaxBlue = MAXRGB
#Const clMaxHue = MAXHUE
#Const clMaxSaturation = MAXHSB
#Const clMaxBrightness = MAXHSB

```

```

Private Const MAXRGB = 255
Private Const MAXHUE = 359
Private Const MAXHSB = 100

```

```

Public Enum EColorConstants
    clMaxRed = MAXRGB
    clMaxGreen = MAXRGB
    clMaxBlue = MAXRGB
    clMaxHue = MAXHUE
    clMaxSaturation = MAXHSB
    clMaxBrightness = MAXHSB
End Enum

```

```

Public Function RGBAsString(ByVal RGBValue As Long) As String
    Dim R As Long, G As Long, b As Long
    Call SplitRGB(RGBValue, R, G, b)
    RGBAsString = CStr(R) & ", " & CStr(G) & ", " & CStr(b)
End Function

```

```

Public Function HSBAsString(ByVal HSBValue As Long) As String
    Dim h As Long, s As Long, L As Long
    Call SplitHSB(HSBValue, h, s, L)
    HSBAsString = CStr(h) & ", " & CStr(s) & ", " & CStr(L)
End Function

```

```

Public Function RGBToLong(ByVal Red As Long, _
    ByVal Green As Long, _
    ByVal Blue As Long)

    Call xLimit(Red, 0, MAXRGB)
    Call xLimit(Green, 0, MAXRGB)
    Call xLimit(Blue, 0, MAXRGB)
    RGBToLong = RGB(Red, Green, Blue)
End Function

```

```

Public Function HSBToLong(ByVal Hue As Long, _
    ByVal Saturation As Long, _
    ByVal Brightness As Long) As Long

    Call xLimit(Hue, 0, MAXHUE)
    Call xLimit(Saturation, 0, MAXHSB)
    Call xLimit(Brightness, 0, MAXHSB)
    HSBToLong = (Hue * &H10000) Or (Saturation * 256) Or (Brightness)
End Function

```

```

Function Red(ByVal RGBValue As Long) As Long

    Red = RGBValue And 255&
End Function

```

```

Function Green(ByVal RGBValue As Long) As Long
    Green = (RGBValue And &HFF00&) \ 256
End Function

```

```

Function Blue(ByVal RGBValue As Long) As Long

    Blue = (RGBValue And &HFF0000) \ &H10000
End Function

```

```
Function Hue(ByVal HSBValue As Long) As Long
```

```
    Hue = xLimit((HSBValue \ &H10000), 0, MAXHUE)
End Function
```

```
Function Saturation(ByVal HSBValue As Long) As Long
```

```
    Saturation = xLimit(((HSBValue And &HFF00&) \ 256), 0, MAXHSB)
End Function
```

```
Function Brightness(ByVal HSBValue As Long) As Long
```

```
    Brightness = xLimit((HSBValue And &HFF&), 0, MAXHSB)
End Function
```

```
Sub SplitRGB(ByVal RGBValue As Long, _
```

```
ByRef Red As Long, _
ByRef Green As Long, _
ByRef Blue As Long)
```

```
    Red = (RGBValue And &HFF&)
    Green = (RGBValue And &HFF00&) \ 256
    Blue = (RGBValue And &HFF0000) \ &H10000
End Sub
```

```
Sub SplitHSB(ByVal HSBValue As Long, _
```

```
ByRef Hue As Long, _
ByRef Saturation As Long, _
ByRef Brightness As Long)
```

```
    Hue = xLimit((HSBValue \ &H10000), 0, MAXHUE)
    Saturation = xLimit(((HSBValue And &HFF00&) \ 256), 0, MAXHSB)
    Brightness = xLimit((HSBValue And &HFF&), 0, MAXHSB)
End Sub
```

```
Function HSBToRGB(ByVal HSBValue As Long) As Long
```

```
' Adapted from Java.awt.Color.java
```

```
Dim R As Long, G As Long, b As Long
Dim h As Long, s As Long, L As Long
Dim nH As Single, nS As Single, nL As Single
Dim nF As Single, nP As Single, nQ As Single, nT As Single
Dim lH As Long
```

```
Call SplitHSB(HSBValue, h, s, L)
```

```
If s > 0 Then
```

```
    nH = h / 60: nL = L / 100: nS = s / 100
```

```
    lH = Int(nH)
```

```
    nF = nH - lH
```

```
    nP = nL * (1 - nS)
```

```
    nQ = nL * (1 - nS * nF)
```

```
    nT = nL * (1 - nS * (1 - nF))
```

```
    Select Case lH
```

```
    Case 0
```

```
        R = nL * 255
```

```
        G = nT * 255
```

```
        b = nP * 255
```

```
    Case 1
```

```
        R = nQ * 255
```

```
        G = nL * 255
```

```
        b = nP * 255
```

```
    Case 2
```

```
        R = nP * 255
```

```
        G = nL * 255
```

ColorConverter - 3

```
    b = nT * 255
Case 3
    R = nP * 255
    G = nQ * 255
    b = nL * 255
Case 4
    R = nT * 255
    G = nP * 255
    b = nL * 255
Case 5
    R = nL * 255
    G = nP * 255
    b = nQ * 255
End Select
Else
    R = (L * 255) / 100
    G = R: b = R
End If

HSBToRGB = RGBToLong(R, G, b)

End Function
```

Function RGBToHSB(ByVal RGBValue As Long) As Long

' Adapted from Java.awt.Color.java

```
Dim nTemp As Single
Dim lMin As Long, lMax As Long, lDelta As Long
Dim R As Long, G As Long, b As Long
Dim h As Long, s As Long, L As Long

Call SplitRGB(RGBValue, R, G, b)

lMax = IIf(R > G, IIf(R > b, R, b), IIf(G > b, G, b))
lMin = IIf(R < G, IIf(R < b, R, b), IIf(G < b, G, b))

lDelta = lMax - lMin

L = (lMax * 100) / 255

If lMax > 0 Then
    s = (lDelta / lMax) * 100
    If lDelta > 0 Then
        If lMax = R Then
            nTemp = (G - b) / lDelta
        ElseIf lMax = G Then
            nTemp = 2 + (b - R) / lDelta
        Else
            nTemp = 4 + (R - G) / lDelta
        End If
        h = nTemp * 60
        If h < 0 Then h = h + 360
    End If
End If

RGBToHSB = HSBToLong(h, s, L)

End Function
```

Private Function xLimit(Value As Long, _
Lower As Long, Higher As Long) As Long

```
    If Value < Lower Then Value = Lower
    If Value > Higher Then Value = Higher
    xLimit = Value
End Function
```

```
Dim k As Integer
Dim theDoc As New Doc
Dim FileName
Dim NumberH, NumberV, BlockWidth, BlockHeight, MarginB, MarginTop, MarginL As Integer
Private cLObject As ColorConverter

Private Sub Start_Click()
    PosterCreator.Counter = 0

    SavePDF.ShowSave
    'MsgBox SavePDF.FileName
    If Trim(SavePDF.FileName) = "" Then
        Exit Sub
    End If
    FileName = SavePDF.FileName
    k = 0
    Set theDoc = CreateObject("ABCpdf4.Doc")
    With theDoc
        With Form1

            theDoc.Units = "mm"
            theDoc.MediaBox = "0 0 " & Trim(Form1.ppWidth) & " " & Trim(Form1.ppHeight)
            theDoc.Rect = theDoc.MediaBox
            'theDoc.AddFont "Chaparral Display"
            'theDoc.Font = "Chaparral Display"
            'theDoc.FontSize = 120
            'theDoc.Pos.Y = 1500
            'theDoc.AddText ("Beta 1.0")
            NumberH = Int((.ppWidth - .mLeft * 2) / .blWidth)
            PosterCreator.MaxX = NumberH
            Form1.Example2.ScaleWidth = NumberH
            MarginL = (.ppWidth - (NumberH * .blWidth)) / 2
            NumberV = Int((.ppHeight - .mTop - .mBottom) / .blHeight)
            PosterCreator.MaxY = NumberV
            PosterCreator.Total = NumberV * NumberH
            Form1.Example2.ScaleHeight = NumberV
            'Form1.Example2.Height = Form1.Example.Width * (NumberV / NumberH)
            Form1.Frame6.Height = Form1.Frame6.Width * (NumberV / NumberH)
            MarginB = .ppHeight - (.mTop + NumberV * .blHeight)
            BlockHeight = .blHeight
            BlockWidth = .blWidth
            Form1.Bar1.Max = NumberH
            Call PosterCreator.AddColor(0, 100, 100, 0, 0)
            Call PosterCreator.AddCollection(0, 0)

        End With
    End With

    Timer2.Enabled = True
End Sub

Private Sub Command2_Click()
    Timer2.Enabled = False
    theDoc.Save FileName
    Bar1.Value = 0
    Progress = "0 %"
    k = 0
    i = 0
End Sub

Private Sub Form_Load()
    Set cLObject = New ColorConverter
    SavePDF.Filter = "All files (*.*)|*.*|Video Files (*.pdf)|*.pdf"
    SavePDF.FilterIndex = 2
End Sub

Private Sub Test()
End Sub

Sub DrawColor(x, y, h, s, b)
    Dim HSB
    If s = -1 Then s = 100
    If b = -1 Then b = 100
    HSB = cLObject.HSBToLong(h, s, b)
    With theDoc
```

```

.Color.Red = clObject.Red(clObject.HSBToRGB(HSB))
.Color.Blue = clObject.Blue(clObject.HSBToRGB(HSB))
.Color.Green = clObject.Green(clObject.HSBToRGB(HSB))
Form1.Color1.BackColor = clObject.RGBToLong(.Color.Red, .Color.Green, .Color.Blue)
Form1.Example2.Line (x, y)-Step(1, 1), Form1.Color1.BackColor, BF
End With
End Sub

```

```

Private Sub Timer1_Timer()
    Dim Heu, Saturation, Brightness, HSB, Lowerbound

    With theDoc
        If k < NumberH - 1 Then
            For i = 0 To NumberV
                .Rect.Bottom = i * BlockHeight + MarginB
                .Rect.Left = k * BlockWidth + MarginL
                .Rect.Width = BlockWidth
                .Rect.Height = BlockHeight

                Heu = PosterCreator.Collection(PosterCreator.Raster(k, i).Index).Hue

                Saturation = PosterCreator.Collection(PosterCreator.Raster(k, i).Index).Saturation

                Brightness = PosterCreator.Collection(PosterCreator.Raster(k, i).Index).Brightness
                HSB = clObject.HSBToLong(Heu, Saturation, Brightness)

                .Color.Red = clObject.Red(clObject.HSBToRGB(HSB))
                .Color.Blue = clObject.Blue(clObject.HSBToRGB(HSB))
                .Color.Green = clObject.Green(clObject.HSBToRGB(HSB))
                'Form1.Color1.BackColor = clObject.RGBToLong(.Color.Red, .Color.Green, .Color.Blue)

                Form1.Color1.BackColor = clObject.RGBToLong(.Color.Red, .Color.Green, .Color.Blue)
                .FillRect
                Form1.Example.Line (k, i)-Step(1, 1), Form1.Color1.BackColor, BF
            Next
            k = k + 1
            Bar1.Value = Bar1.Value + 1
            Progress = Str(Int((Bar1.Value / Bar1.Max) * 100)) & " %"
        Else
            theDoc.Save FileName
            k = 0
            Bar1.Value = 0
            Progress = "Finished"
            Timer1.Enabled = False
        End If
    End With
End Sub

```

```

Private Sub Timer2_Timer()
    Dim Position As Cord
    If PosterCreator.Counter < PosterCreator.Total Then
        Position = PosterCreator.FindNext
        Call PosterCreator.AddCollection(Position.x, Position.y)
    Else
        MsgBox "finished" & UBound(PosterCreator.Collection) & " " & PosterCreator.Total
        Dim extra
        extra = PosterCreator.CalcExtra
        If extra = 0 Then
            MsgBox "Finished"
            Timer2.Enabled = False
            k = 0
            Timer1.Enabled = True
        End If
        PosterCreator.Total = PosterCreator.Total + extra
    End If
End Sub

```

PosterCreator - 1

Public MaxX, MaxY, Total, CurX, CurY, Last, PrevX, PrevY, Wrong As Long

Dim PrevFound As Boolean

Public Type Colorspot

Hue As Integer

Saturation As Integer

Brightness As Integer

x As Long

y As Long

Index As Long

End Type

Public Type Cord

x As Integer

y As Integer

End Type

Public Type Neighbours

Top As Boolean

TopRight As Boolean

Rights As Boolean

Number As Integer

End Type

Public Type Point

Taken As Boolean

Index As Long

End Type

Public LastColors(3) As Cord

Public Raster(600, 600) As Point

Public Collection() As Colorspot

Public Counter As Long

Dim PrevX1, PrevY1 As Integer

Function AddCollection(x, y)

Form1.Label3 = x & " " & y

If x = PrevX1 And y = PrevY1 Then

Wrong = Wrong + 1

If Wrong > 100 Then

Last = 0

Wrong = 0

Call Fill

End If

Else

Wrong = 0

End If

PrevX1 = x

PrevY1 = y

Form1.Label2 = "Addcollection"

Last = 0

Dim Value

Dim Opti As String

Dim X1, X2, Y1, Y2 As Long

Dim Area As Neighbours

Area = FindNeighbours(x, y)

Dim Hue, Saturation, Brightness, Index

For i = 0 To UBound(Collection) - 1

If Collection(i).x = x And Collection(i).y = y Then

Hue = Collection(i).Hue

Saturation = Collection(i).Saturation

Brightness = Collection(i).Brightness

Index = Collection(i).Index

End If

Next

Select Case Area.Number

Case 0

Opti = "1,2,3,4,5,6,7,8"

X1 = x

X2 = x + 1

Y1 = y + 1

Y2 = y

Case 1

```
Opti = "1,2,3,4,5,6"
If Area.Top Then
    X1 = x + 1
    X2 = x + 1
    Y1 = y
    Y2 = y + 1
ElseIf Area.TopRight Then
    X1 = x
    X2 = x + 1
    Y1 = y
    Y2 = y + 1
Else
    X1 = x
    X2 = x + 1
    Y1 = y + 1
    Y2 = y + 1
End If
```

Case 2

```
Opti = "1,5"
If Not Area.Top Then
    X1 = x
    X2 = x
    Y1 = y + 1
    Y2 = y + 1
ElseIf Not Area.TopRight Then
    X1 = x + 1
    X2 = x + 1
    Y1 = y + 1
    Y2 = y + 1
Else
    X1 = x + 1
    X2 = x + 1
    Y1 = y
    Y2 = y
End If
```

Case 3

```
Exit Function
```

End Select

Dim Values

```
Values = Split(Opti, ",")
```

```
Select Case Values(Int((UBound(Values) + 1) * Rnd))
```

Case 1

```
'Same hue
Collection(Index).Brightness = 100
Select Case Int(2 * Rnd + 1)
    Case 1
        Call AddColor(Hue, -1, 70, X1, Y1)
    Case Else
        Call AddColor(Hue, -1, 70, X2, Y2)
End Select
```

Case 2

```
'Triad
Call AddColor(CalcHue(Hue, 120), -1, -1, X1, Y1)
Call AddColor(CalcHue(Hue, -120), -1, -1, X2, Y2)
```

Case 3

```
'split components
Call AddColor(CalcHue(Hue, -150), -1, -1, X1, Y1)
Call AddColor(CalcHue(Hue, 150), -1, -1, X2, Y2)
```

Case 4

```
'Contrasting hues
Collection(Index).Brightness = 75
Call AddColor(CalcHue(Hue, -120), -1, 50, X1, Y1)
Call AddColor(CalcHue(Hue, 120), -1, 50, X2, Y2)
```

Case 5

```
'Complements
Select Case Int(2 * Rnd + 1)
    Case 1
        Call AddColor(CalcHue(Hue, 180), -1, -1, X1, Y1)
    Case Else
        Call AddColor(CalcHue(Hue, 180), -1, -1, X2, Y2)
```

```

        End Select
    Case 6
        'Similiar Hues
        Value = Int(2 * Rnd + 1) * 15 + 15
        Call AddColor(CalcHue(Hue, -Value), -1, -1, X1, Y1)
        Call AddColor(CalcHue(Hue, Value), -1, -1, X2, Y2)
    Case 7
        'Square
        Call AddColor(CalcHue(Hue, 90), -1, -1, x, y + 1)
        Call AddColor(CalcHue(Hue, 180), -1, -1, x + 1, y)
        Call AddColor(CalcHue(Hue, 270), -1, -1, x + 1, y + 1)
    Case 8
        'rectangle
        Call AddColor(CalcHue(Hue, 30), -1, -1, x, y + 1)
        Call AddColor(CalcHue(Hue, 150), -1, -1, x + 1, y)
        Call AddColor(CalcHue(Hue, 240), -1, -1, x + 1, y + 1)
End Select
Dim Mistake As Boolean
Mistake = FindHoles
While Mistake
    Mistake = FindHoles
Wend

End Function

Function FindNeighbours(x, y) As Neighbours
    If PrevX = x And PrevY = y Then
        Call NewLine
    Else
        Form1.Label2 = "FindNeighbours" & x & " " & y
        For i = 0 To UBound(Collection) - 1
            If (Collection(i).x = x + 1) And (Collection(i).y = y + 1) Then
                FindNeighbours.TopRight = True
                FindNeighbours.Number = FindNeighbours.Number + 1
            End If
            If (Collection(i).x = x + 1) And (Collection(i).y = y) Then
                FindNeighbours.Rights = True
                FindNeighbours.Number = FindNeighbours.Number + 1
            End If
            If (Collection(i).x = x) And (Collection(i).y = y + 1) Then
                FindNeighbours.Top = True
                FindNeighbours.Number = FindNeighbours.Number + 1
            End If
        Next
        'If x = MaxX Then
        '    FindNeighbours.Rights = True
        'End If
        'If y = MaxY Then
        '    FindNeighbours.Top = True
        '    FindNeighbours.TopRight = True
        'End If
    End If
    PrevX = x
    PrevY = y
End Function

Function AddColor(Hue, Saturation, Brightness, x, y)
    Form1.Label2 = "AddColor"
    If Raster(x, y).Taken Then
        i = i
    Else

        Last = Last + 1
        LastColors(Last - 1).x = x
        LastColors(Last - 1).y = y

        Counter = Counter + 1

        ReDim Preserve Collection(Counter)
        Collection(Counter - 1).Brightness = Brightness
        Collection(Counter - 1).Saturation = Saturation
        Collection(Counter - 1).Hue = Hue
        Collection(Counter - 1).x = x
        Collection(Counter - 1).y = y
        Collection(Counter - 1).Index = Counter
        Raster(x, y).Taken = True
        Raster(x, y).Index = Counter - 1
    End If
End Function

```



```

        Call Form1.DrawColor(x, y, Hue, Saturation, Brightness)
    End If

End Function

Function CalcHue(Hue, Value)
    Hue = Hue + Value
    If Hue > 360 Then
        Hue = Hue - 360
    ElseIf Hue < 0 Then
        Hue = Hue + 360
    End If
    CalcHue = Hue
End Function

Function FindNext() As Cord
    Dim Temp As Integer
    Temp = Int(Rnd * Last)
    FindNext.x = LastColors(Temp).x
    FindNext.y = LastColors(Temp).y
    If FindNext.x = MaxX Then FindNext.x = 0
    If FindNext.y = MaxY Then FindNext.y = 0
End Function

Function FindHoles() As Boolean

    Dim Found, Restored As Boolean

    For i = 0 To MaxY
        Found = False
        For k = 0 To MaxX
            If Raster(MaxX - k, i).Taken Then
                Found = True
            End If
            If (Not Raster(MaxX - k, i).Taken) And Found Then
                FindHoles = True
                Call DrawHole(MaxX - k, i)
                PrevFound = True
                Exit For
            End If
        Next
        If FindHoles Then Exit For
    Next

    Form1.Label2 = "FindHoles" & " " & FindHoles & k & " " & i
End Function

Function DrawHole(x, y)
    Form1.Label2 = "DrawHole"
    Last = 1
    Dim Hue
    Hue = Collection(Raster(x + 1, y).Index).Hue
    Select Case Int(2 * Rnd + 1)
        Case 0
            Value = Int(2 * Rnd + 1) * 15 + 15
        Case 1
            Value = Int(2 * Rnd + 1) * 15 + 15
        Case 2
            Value = -180
        Case Else
            Value = 180
    End Select

    Call AddColor(CalcHue(Hue, Value), -1, -1, x, y)
End Function

Function NewLine()
    Form1.Label2 = "newline"
    Dim y
    For i = 0 To MaxY
        If Raster(MaxX, i).Taken = False Then
            y = i
            Exit For
        End If
    Next
    For i = 0 To MaxX
        If Raster(MaxX - i, y).Taken = True Then
            Last = 1
            Call AddColor(Int(Rnd * 359), 100, 100, MaxX - i + 1, 0)
        End If
    Next
End Function

```

```

        Exit For
    End If
Next
End Function

Function CalcExtra() As Long
    Label2 = "CalcExtra"
    For i = 0 To MaxX
        For k = 0 To MaxY
            If Raster(i, k).Taken = False Then
                CalcExtra = CalcExtra + 1
            End If
        Next
    Next
End Function

Function Fill() As Long
    Label2 = "Fill"
    For i = 0 To MaxX
        For k = 0 To MaxY
            If Raster(i, k).Taken = False Then
                Call AddColor(Int(Rnd * 359), -1, -1, i, k)
                Last = 0
            End If
        Next
    Next
    For i = 0 To MaxX
        For k = 0 To MaxY
            If Collection(Raster(i, k).Index).Saturation = -1 Then
                If i > MaxX / 5 Then
                    Lowerbound = 100 - i * (100 / (4 * MaxX / 5))
                    Collection(Raster(i, k).Index).Saturation = Int((100 - Lowerbound + 1) * Rnd +
Lowerbound)

                Else
                    Collection(Raster(i, k).Index).Saturation = 100
                End If
            End If

            If Collection(Raster(i, k).Index).Brightness = -1 Then
                If i < MaxX / 5 Then
                    Select Case Int(9 * Rnd + 1)
                        Case Is < 2
                            Collection(Raster(i, k).Index).Brightness = Int((100 + 1) * Rnd)
                        Case Else
                            Collection(Raster(i, k).Index).Brightness = 100
                    End Select
                ElseIf i < MaxX / 5 * 3 Then
                    Select Case Int(9 * Rnd + 1)
                        Case Is < 7

                            Collection(Raster(i, k).Index).Brightness = Int((100 + 1) * Rnd)
                        Case Else
                            Collection(Raster(i, k).Index).Brightness = 100
                    End Select
                Else
                    Collection(Raster(i, k).Index).Brightness = Int((100 + 1) * Rnd)
                End If
            End If
            Call Form1.DrawColor(i, k, Collection(Raster(i, k).Index).Hue, Collection(Raster(i, k).
Index).Saturation, Collection(Raster(i, k).Index).Brightness)
        Next
    Next
End Function

```