

25.1.5 CairoPattern

Die Klasse repräsentiert ein Cairo-Oberflächen-Pattern (oder Oberflächen-Muster oder im engeren Sinn von Cairo einen Oberflächen-Pinsel), wie es von *Cairo.SolidPattern* oder *Cairo.LinearGradient* zurückgegeben wird.

Beispiel Cairo.LinearGradient:

```
Public Sub CairoScriptLinearGradient()
    Dim aColors0 As Integer[] = [Color.Blue, Color.Yellow, Color.Red, Color.Gray, Color.Green, Color.Orange]
    Dim aColors As Float[][]
    Dim fPosition As Float

    aColors = BuildColorStops(aColors0, Null)

    GenerateNewImage()
    SetImageBorder()
    Cairo.Begin(hImage)
    Cairo.Translate(xTranslate, yTranslate)
    Cairo.Scale(xScale, yScale)
    DrawCoordinateSystem() ' +y ▲

    Cairo.MoveTo(50, 40)
    Cairo.Source = Cairo.LinearGradient(50, 40, 450, 200, aColors)
    Cairo.Rectangle(50, 40, 450, 200)
    Cairo.Fill()
    Cairo.End
End ' CairoScriptLinearGradient()
```

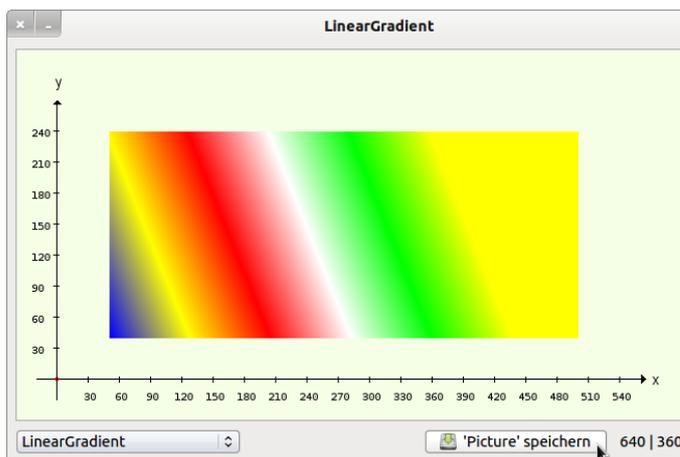


Abbildung 25.1.5.1: Farbverlauf von sechs Farben als Oberflächen-Muster

Die Klasse *CairoPattern* hat nur zwei Eigenschaften:

- CairoPattern.Filter
- CairoPattern.Matrix

25.1.5.1 Filter

Setzt oder gibt den Filter zurück, der bei der Vergrößerung unter diesem Pattern verwendet wird. Ein "Filter" ist hierbei als Algorithmus zu verstehen, der eine Vergrößerung berechnet. Bei der Auswahl des Algorithmus ist zwischen diesen zwei gegensätzlichen Interessen abzuwägen: benötigte Rechenzeit und Qualität. Die Filter-Eigenschaft kann einen der folgenden Werte annehmen:

- Cairo.FilterFast
- Cairo.FilterGood
- Cairo.FilterBest
- Cairo.FilterNearest
- Cairo.FilterBilinear
- Cairo.FilterGaussian (gegenwärtig *nicht* implementiert → 1. Mai 2015)

Wenn Sie den Filter kontrollieren wollen, ohne explizit ein CairoPattern-Objekt zu erzeugen, so verwenden Sie *Cairo.Source*, um Zugriff auf das aktuelle Muster zu erhalten:

```
Cairo.Source.Filter = Cairo.FilterNearest
```

25.1.5.2 Matrix

Setzt oder gibt die Pattern-Transformationsmatrix zurück. Diese Matrix transformiert User-Space in Pattern-Space. Wenn ein Pattern-Objekt erzeugt wird, so ist diese Matrix die Identität, also ist der Pattern-Space gleich dem User-Space.

Beachten Sie, dass die Richtung der Transformation von User-Space nach Pattern-Space ist. Das bedeutet – wenn Sie sich die Transformation eines Musters vom Pattern- über den User- in den Device-Space vorstellen – dass der erste Schritt von Pattern- nach User-Space durch die Inverse der Cairo-Pattern-Matrix beschrieben wird. Wenn Sie zum Beispiel ein Muster zweimal so groß machen wollen, wie es normalerweise ist, benutzen Sie:

```
Dim MyPattern As CairoPattern = ...  
MyPattern.Matrix = CairoMatrix().Scale(0.5, 0.5)
```

Würden Sie die Werte 2.0 statt 0.5 benutzen, dann hätte das Muster wegen der inversen CairoPattern-Matrix nur die *halbe* statt der *doppelten* Größe!