

19.7.2 Text HighlighterStyle (gb.highlight)

Die Klasse TextHighlighterStyle präsentiert einen Style, mit denen Textabschnitte, die innerhalb der Hervorhebungsregeln definiert werden, eingefärbt und (optional) in engen Grenzen formatiert werden können. Ein Style der Klasse TextHighlighterStyle besteht

- aus der obligatorischen Angabe einer Farbe und
- einem optionalen Stil-Format (Bold: Standard = False, Dotted: Standard = False, Underline: Standard = False).

19.7.2.1 Eigenschaften

Die Klasse TextHighlighterStyle verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Background	Boolean	Gibt an, ob die Farbe des Styles zum Zeichnen eines Hintergrunds und nicht zum Zeichnen von Text bestimmt ist.
Bold	Boolean	Gibt zurück oder legt fest, ob der mit diesem Stil gezeichnete Text fett sein muss.
Color	Integer	Gibt die RGB-Farbe zurück oder legt die RGB-Farbe fest, die zum Färben eines Textes mit diesem Style verwendet wird.
Dotted	Boolean	Gibt zurück oder setzt, wenn der mit diesem Stil gezeichnete Text eine gepunktete Unterstreichung haben muss.
InvertedColor	Integer	Gibt die invertierte Farbe der Eigenschaft 'Color' zurück.
Key	String	Gibt den Namen zurück, dem der Style zugeordnet ist oder setzt ihn.
Name	String	Gibt den Anzeigenamen des Styles zurück.
Strikeout	Boolean	Gibt zurück oder setzt, wenn der mit diesem Stil gezeichnete Text durchgestrichen sein muss.
Underline	Boolean	Gibt zurück oder setzt, wenn der mit diesem Stil gezeichnete Text eine durchgehende Unterstreichung haben muss.

Tabelle 19.7.2.1.1 : Eigenschaften der Klasse TextHighlighterStyle

19.7.2.2 Methoden

Die Klasse TextHighlighterStyle besitzt nur diese zwei wesentlichen Methoden:

Methode	Rückgabotyp	Beschreibung
Copy ()	TextHighlighterStyle	Gibt eine Kopie des aktuellen Styles zurück.
ToString ()	String	Konvertiert den Inhalt des aktuellen Styles in eine Zeichenkette.

Tabelle 19.7.2.2.1 : Methoden der Klasse TextHighlighterStyle

19.7.2.3 Ereignisse

Die Klasse TextHighlighterStyle verfügt über dieses relevante Ereignis:

Ereignis	Beschreibung
Change ()	Dieses Ereignis wird ausgelöst, wenn sich der Style geändert hat.

Tabelle 19.7.2.3.1 : Ereignis der Klasse TextHighlighterStyle

19.7.2.4 Einen Text-Highlighter-Style erzeugen

Nachdem Sie für einen neuen Text-Highlighter den Inhalt seiner Definitionsdatei festgelegt haben, kennen Sie die Liste der Style-Namen, die Sie beispielsweise für den Text-Highlighter mit der Definitionsdatei lipa.highlight so auslesen können:

```
Print "Ordered list of the identifiers of the style names in the text marker 'lipa'"
Print String$(83, "-")
Print lTextHighlighter.States.Join()
```

Ausgabe in der Konsole der IDE:

```
Ordered list of the identifiers of the style names in the text marker 'lipa'
-----
AddressStyle,CommentStyle,EndStyle,InOutStyle,JumpStyle,LoadAStyle,OperationStyle,StoreStyle
```

ACHTUNG: Bevor Sie einen neuen TextHighlighter-Style erzeugen ist es notwendig, ein neues Highlighter-Thema zu erzeugen, in das Sie die neuen Styles (temporär) einfügen:

```
Private lTextHighlighterTheme As TextHighlighterTheme
'-- Create a new highlighter theme initialized with the default "gambas" theme!
lTextHighlighterTheme = New TextHighlighterTheme
```

Es gibt zwei Möglichkeiten, einen neuen Style zu erzeugen. Beide werden hier vorgestellt. Bei der ersten Möglichkeit wird die statische Funktion TextHighlighterStyle(...) verwendet. Bei der zweiten werden die Style-Eigenschaften für den Style separat festgelegt.

19.7.2.4.1 Erste Möglichkeit

Die Klasse TextHighlighterStyle kann als statische Funktion so verwendet werden:

```
Static Function TextHighlighterStyle ( Color As Integer [ , Bold As Boolean, Underline As Boolean, Dotted
As Boolean ] ) As TextHighlighterStyle
```

Diese Funktion erzeugt einen neuen TextHighlighter-Style aus einigen seiner Eigenschaften, wobei die letzten drei Argumente optional sind:

- Color: RGB-Farbe, die beim Hervorheben von Textabschnitten mit diesem Style verwendet wird.
- Bold: Der mit diesem Stil hervorgehobene Text soll fett sein.
- Underline: Der mit diesem Stil hervorgehobene Text soll unterstrichen sein.
- Dotted: Der mit diesem Stil hervorgehobene Text soll eine gepunktete Unterstreichung haben.

Erzeugung des Styles mit dem Style-Namen 'LoadAStyle':

```
Private hTextHighlighterStyle As TextHighlighterStyle
hTextHighlighterStyle = New TextHighlighterStyle
hTextHighlighterStyle = TextHighlighterStyle(&H7F00FF, True, False, False)
```

Einfügen des Styles in das aktuelle Highlighter-Thema:

```
lTextHighlighterTheme.Add("LoadAStyle", hTextHighlighterStyle)
```

Beachten Sie, dass Sie mit der statischen Funktion das Stil-Format 'Strikeout' nicht festlegen können. Das gelingt Ihnen explizit nur mit der Zuweisung: hTextHighlighterStyle.Strikeout = True | False.

Wesentlich schneller geht das direkte Einfügen des neuen Styles in das aktuelle Highlighter-Thema:

```
lTextHighlighterTheme.Add("LoadAStyle", &H7F00FF, True, False, False)
```

19.7.2.4.2 Zweite Möglichkeit

Erzeugung des Styles mit dem Style-Namen 'JumpStyle':

Zuerst erzeugen Sie einen neuen Style:

```
Private hTextHighlighterStyle As TextHighlighterStyle
hTextHighlighterStyle = New TextHighlighterStyle
```

Dann weisen Sie dem neuen Style die Style-Eigenschaften separat zu:

```
hTextHighlighterStyle.Color = &H0000FF
hTextHighlighterStyle.Bold = True
```

```
hTextHighlighterStyle.Dotted = False
'-- Please note that you can only use the styles 'Strikeout' and 'Underline' alternatively!
'-- hTextHighlighterStyle.Strikeout = True
hTextHighlighterStyle.Underline = False
```

Abschließend fügen Sie den neuen Style zum aktuellen Highlighter-Thema hinzu. Der verwendete Key ist der Style-Name:

```
lTextHighlighterTheme["JumpStyle"] = hTextHighlighterStyle
```

So können Sie den Style mit dem Style-Namen 'JumpStyle' im aktuellen Highlighter-Thema auslesen und dessen Eigenschaften in der IDE-Konsole ausgeben:

```
hTextHighlighterStyle = lTextHighlighterTheme["JumpStyle"]
Print "Key      = "; hTextHighlighterStyle.Key
Print "Name     = "; hTextHighlighterStyle.Name
Print "Color    = "; hTextHighlighterStyle.Color
Print "Bold     = "; hTextHighlighterStyle.Bold
Print "Dotted   = "; hTextHighlighterStyle.Dotted
Print "Strikeout = "; hTextHighlighterStyle.Strikeout
Print "Underline = "; hTextHighlighterStyle.Underline
```

Ausgabe:

```
Key      = JumpStyle
Name     = JumpStyle
Color    = 255
Bold     = True
Dotted   = False
Strikeout = False
Underline = False
```

Wenn Sie die Eigenschaft `hTextHighlighter.Name` nicht ausdrücklich gesetzt haben, dann wird als Name automatisch der Key ausgegeben.