

19.7.3 Klasse TextHighlighterTheme (gb.highlight)

Die Klasse TextHighlighterTheme implementiert ein Text-Highlighter-Thema, das in einer Datei *.theme gespeichert werden kann.

So erzeugen Sie ein neues Text-Highlighter-Thema:

```
Private hTextHighlighterTheme As TextHighlighterTheme
hTextHighlighterTheme = New TextHighlighterTheme
```

in das Sie u.a. mit der Add-Methode die neuen Styles Ihres benutzer-definierten Text-Highlighters einfügen können.

Bitte beachten Sie: Wird ein neues TextHighlighter-Thema erzeugt, so wird es mit dem Standard-Thema 'gambas' initialisiert!

Das können Sie unmittelbar nachprüfen, denn die Anweisung

```
Print hTextHighlighterTheme.ToString()
```

erzeugt in der Konsole der IDE die Ausgabe einer längere Zeichenkette aus einzelnen Styles, die durch ein Semikolon voneinander getrennt sind. Ein solcher Style besteht aus der Angabe eines Style-Namens, dem nach einem Gleichheitszeichen ein Style folgt – bestehend aus der obligatorischen Farbangabe und dem optimalen Stil-Format – so wie es zum Beispiel bei **Datatype** hervorgehoben wurde:

```
Background=#FFFFFF;Normal=#000000;Keyword=#000000,Bold;Datatype=#E67E22,Bold;Function=#3398C3,Bold;Operator=#000000,Bold;Symbol=#000000;Number=#E62222;String=#7F0000;Comment=#888786;Documentation=#808080,Bold;Preprocessor=#1CC01C,Bold;Breakpoint=#FF5F5F;...;Doctype=#1CC01C,Bold;RegExp=#E67E22,Bold
```

Diese Klasse funktioniert wie ein Lese-/Schreib-Array.

So können Sie einen definierten Style in das aktuelle Highlighter-Thema einfügen – hier mit dem Style-Namen 'LoadAStyle':

```
hTextHighlighterTheme["LoadAStyle"] = hTextHighlighterStyle
```

Alternativ fügen Sie den Style direkt mit der Add()-Methode ein:

```
hTextHighlighterTheme.Add("LoadAStyle", &H7F00FF, True)
```

Wollen Sie sich die Style-Eigenschaften für den Style-Namen 'JumpStyle' ansehen, dann können Sie so auf den Style und dann auf ausgewählte Eigenschaften zugreifen:

```
hTextHighlighterStyle = hTextHighlighterTheme["JumpStyle"]
Print "Color = "; hTextHighlighterStyle.Color
```

19.7.3.1 Statische Eigenschaften

Die Klasse TextHighlighterTheme verfügt über diese zwei statischen Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
AllStates	String[]	Gibt die Liste aller Style-Namen der registrierten Hervorhebungsregeln als String-Array zurück.
States	String[]	Gibt die Liste aller Style-Namen als String-Array zurück, die in <u>jedem</u> registrierten Highlighter implementiert sind.

Tabelle 19.7.3.1.1 : Statische Eigenschaften der Klasse TextHighlighterTheme

Mit den folgenden Anweisungen können Sie die Werte der Eigenschaften AllStates und States auslesen und in der Konsole der IDE anzeigen:

```
Print "AllStates"
Print String$(76, "-")
Print TextHighlighterTheme.AllStates.Join()
```

```
AllStates
-----
Normal,Background,Selection,Current,Breakpoint,Highlight,Added,Removed,Error,Comment,Documentation,Keyword,
Function,Operator,Symbol,Number,String,Datatype,Preprocessor,Escape,Constant,Shebang,Expansion,Command,At-
Rule,Class,PseudoClass,Id,Tag,TagAttribute,Color,Unit,Important,Diff,Index,File,Position,Doctype,Entity,Mar-
kup,Attribute,Value,RegExp,Label,InOutStyle,LoadAStyle,StoreStyle,JumpStyle,OperationStyle,EndStyle,Ad-
dressStyle,CommentStyle
```

```
Print "Ordered list: States"
Print String$(76, "-")
Print TextHighlighterTheme.States.Join()
```

```
Ordered list: States
-----
Added,Background,Breakpoint,Comment,Constant,Current,Datatype,Documentation,Error,Escape,Function,High-
light,Keyword,Normal,Number,Operator,Preprocessor,Removed,Selection,String,Symbol
```

19.7.3.2 Statische Methoden

Die Klasse TextHighlighterTheme besitzt diese beiden statischen Methoden:

Methode	Rückgabotyp	Beschreibung
GetStateIndex (Key As String)	Integer	Liefert den Index einer bestimmten Hervorhebungsbedingung. Key ist der Schlüssel der Bedingung. Wenn die Hervorhebungsbedingung entweder 'Normal' oder leer ist, so wird 0 zurückgegeben. Wenn keine Hervorhebungsbedingung mit dem angegebenen Schlüssel registriert wurde, wird -1 zurückgegeben.
Load (Path As String)	TextHighlighterTheme	Erzeugt ein Thema aus einer Datei mit `Path` als Datei-Pfad.

Tabelle 19.7.3.2.1 : Statische Methoden der Klasse TextHighlighterTheme

19.7.3.3 Eigenschaften

Die Klasse TextHighlighterTheme verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Dark	Boolean	Ermittelt, ob ein dunkler Hintergrund verwendet wird.
Settings	String	Gibt den Inhalt des Themas als Einstellungsstring zurück oder setzt ihn.
Styles	TextHighlighterStyle []	Gibt die mit einem Thema verbundenen Styles zurück.

Tabelle 19.7.3.3.1 : Eigenschaften der Klasse TextHighlighterTheme

19.7.3.4 Methoden

Die Klasse TextHighlighterTheme besitzt diese wesentlichen Methoden:

Methode	Rückgabotyp	Beschreibung
Add (Key As String, Color As Integer [, Bold As Boolean, Underline As Boolean, Dotted As Boolean])	TextHighlighterStyle	Shortcut-Methode zur Erzeugung eines Styles. Key: Farbname, Color: RGB-Farbe, Bold: Option - Wenn der Text fett gezeichnet werden soll, Underline: Option - Wenn der Text mit einer durchgehenden Unterstreichung gezeichnet werden muss und Dotted: Option - Wenn der Text mit einer gepunkteten Unterstreichung gezeichnet werden muss. Der neu erzeugte Style wird zurückgegeben.
Copy ()	TextHighlighterTheme	Kopiert ein Thema und gibt das kopierte Thema vom Typ TextHighlighterTheme zurück.
FromString (Theme As String)	-	Initialisiert ein Thema aus einer String-Beschrei-

Methoden	Rückgabotyp	Beschreibung
		bung. 'Theme' ist die Beschreibung des Themas und sollte der Syntax folgen, die eine Beschreibung liefert, die mit der Methode ToString erzeugt wurde.
ToString ()	String	Konvertiert den Inhalt des ausgewählten Themas in eine Zeichenkette.
Invert ()	-	Invertiert die Farben aller Styles eines Themas.
Save (Path As String)	-	Speichert den Inhalt des ausgewählten Themas in einer Datei *.theme, wobei 'Path' der Dateipfad zur Thema-Datei ist.

Tabelle 19.7.3.4.1 : Methoden der Klasse TextHighlighterTheme

19.7.3.5 Ereignisse

Die Klasse TextHighlighterTheme verfügt über dieses Ereignis:

Ereignis	Beschreibung
Change ()	Dieses Ereignis wird ausgelöst, wenn ein Style des Themas geändert wurde.

Tabelle 19.7.3.5.1 : Ereignisse der Klasse TextHighlighterTheme

19.7.3.6 Ein benutzer-definiertes Text-Highlighter-Thema speichern oder laden

Wenn Sie ein neues Text-Highlighter-Thema – kurz Thema – erzeugt und die für Ihr Projekt notwendigen, erprobten Styles in das neue Thema eingefügt haben, dann sollten Sie Ihr Thema in einer Datei mit einem geeigneten Pfad abspeichern. Für den Text-Highlighter für die Programmiersprache LIPA reicht diese eine Zeile aus:

```
lTextHighlighterTheme.Save(User.Home &/ "lipa.theme")
```

Das abgespeicherte Thema für Ihren benutzer-definierten Text-Highlighter enthält somit alle Gambas-Styles und die neuen Styles. Die Datei lipa.theme können Sie in alle weiteren Projekt-Verzeichnisse kopieren, die den Text-Highlighter in einem LIPA-Quelltexteditor benötigen.

Mit dieser Anweisung laden Sie ein im Projektverzeichnis gespeichertes Thema:

```
Private lTextHighlighterTheme As TextHighlighterTheme

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

lTextHighlighterTheme = lTextHighlighterTheme.Load(Application.Path &/ "theme/lipa.theme")
```