

5.6 Gambas-Präprozessor

Betrachtet man diesen Quelltext genauer, dann entdeckt man im Quelltext Präprozessor-Direktiven:

```
#If Debug
  If Box Then ' Rechteck zeichnen, in dem der Text stehen soll
    Cairo.Rectangle(Box.X, Box.Y, Box.Width, Box.Height)
    Cairo.Dash = [5]
    Cairo.Stroke()
  Endif
#Endif
```

Gambas hat einen rudimentären Präprozessor. Das ist eine Einheit, die der lexikalischen Analyse – dem ersten Schritt bei der Kompilierung – vorgeschaltet ist. Was der Gambas-Präprozessor leistet, finden Sie in der Gambas-Dokumentation unter <http://gambaswiki.org/wiki/lang/.if>.

Eine Präprozessor-Direktive hat diese Syntax:

```
#If Preprocessor-Expression ... [ #Else If Preprocessor-Expression ... ] [ #Else ... ] #Endif
```

Eine #IF ... #Endif-Präprozessor-Direktive erlaubt die bedingte Kompilierung von Quelltext. Ein Präprozessor-Ausdruck ist ein rudimentärer boolescher Ausdruck, der folgendes einschließen kann:

- Eine Präprozessor-Boolean-Konstante
- Eine Präprozessor-Konstante - verglichen mit einer Zeichenkette
- Das OR- oder AND-Schlüsselwort
- Klammern, um Unterausdrücke zu gruppieren

Das sind gültige Präprozessor-Konstanten:

Konstante	Wert	Erlaubte Vergleichsoperatoren
System	Betriebssystem	=, <>
Architecture oder Arch	CPU-Architektur	=, <>
Version oder Gambas	Compiler-Version	=, <>, <, <=, >, >=
Debug	Debugging-Informationen eingeschaltet	keine
True	Wert 'wahr'	keine
False	Wert 'falsch'	keine

Tabelle 5.6.1 : Übersicht zu den Präprozessor-Konstanten

Im u.a. Beispiel-Quelltext wird nur eine der vier 'Print'-Anweisungen kompiliert:

```
Public Sub Main()
  #If System = "Linux"
    #If Architecture = "x86_64"
      Print "Linux 64 bits"
    #Else
      Print "Linux 32 bits"
    #Endif
  #Else If System = "FreeBSD"
    Print "FreeBSD ?"
  #Else
    Print "Other !?"
  #Endif
End ' Main()
```

Beachten Sie diesen Hinweis:

Durch die Verwendung der bedingten Kompilierung auf der Basis 'Betriebssystem' oder 'CPU-Architektur' wird bei der Kompilierung eine ausführbare Datei generiert, die *spezifisch* für das Betriebssystem oder die CPU-Architektur ist. Das ist in der Regel aber keine gute Idee. Versuchen Sie statt dessen, zur Laufzeit durch die Prüfung der Werte 'System.Family' und 'System.Architecture' so viel wie möglich vom Betriebssystem oder der CPU-Architektur in Ihrem Code zu erkennen.