

4.5 IDE – Quelltext-Editor

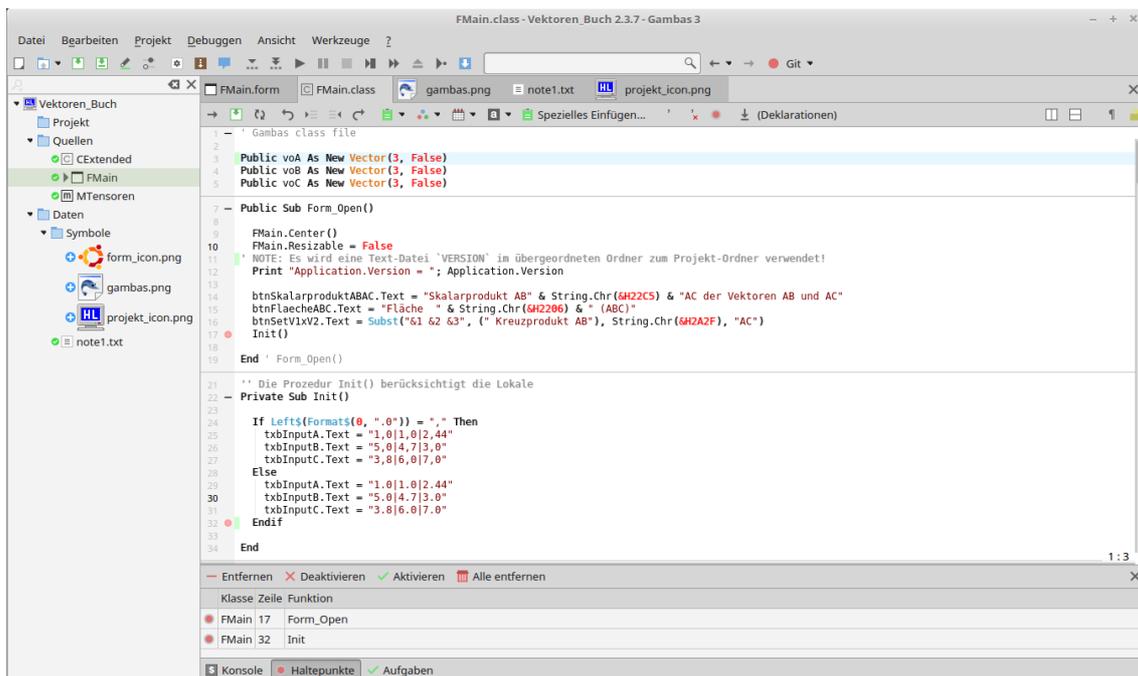


Abbildung 4.5.1: IDE – Quelltext-Editor

Der Quelltext-Editor der IDE bietet neben den normalen Editor-Funktionen wie Datei anlegen, Datei öffnen, Datei schließen, Datei speichern, Datei speichern unter ..., Text markieren, Text kopieren, Text einfügen, Text löschen, Text komplett markieren, Text suchen, Text suchen und ersetzen, Aktion rückgängig machen, Aktion wiederholen und umfangreichem Kontext-Menü u.a. folgende Features:

- Basis-Konfiguration des Quelltext-Editors
- Syntax-Hervorhebung (Syntax-Highlighting)
- Editor-Fenster maximieren/minimieren
- Vertikale und horizontale Teilung des Editor-Fensters
- Kommentare einfügen – 4 Kommentar-Typen
- Textzeile und Text-Block auskommentieren
- Einfügen von Text (CTRL+V, mittlere Maustaste, Menü > Bearbeiten > Einfügen)
- Spezielles Einfügen von Text (Dialog)
- Farben einfügen
- Aktuelles Datum in 17 unterschiedlichen Formaten einfügen
- Automatische Quelltext-Vervollständigung
- Quelltext-Kurzbefehle
- Quelltext *automatisch* formatieren
- Text ein- und ausrücken – Einzug erhöhen oder Einzug verringern
- Lesezeichen definieren, zu Lesezeichen navigieren, Lesezeichen oder alle Lesezeichen löschen
- Liste aller existierenden Deklarationen anzeigen (Methoden (public, private) und Ereignisse)
- Kurzbefehle bearbeiten und Tastenkürzel – Verwendung
- (Online-)Hilfe → Gamas-Dokumentation
- (Online-)Hilfe zu Definitionen anzeigen – Wort markieren und dann F2 drücken
- Verwendung von Präfixen für Steuerelemente und Variablen – optional

Den Quelltext-Editor öffnen Sie

- mit einem Doppelklick auf ein Steuerelement im Form-Editor,
- mit der Funktionstaste F12, wenn der Form-Editor geöffnet ist. Hinweis: Nach einem weiteren Klick auf F12 öffnet sich wieder der Form-Editor (Schalter-Funktion) oder
- mit einem Klick auf eine Klassen- oder Modul-Datei im Projektfenster.

4.5.1 Hinweise zur Basis-Konfiguration des Quelltext-Editors

Die Basis-Konfiguration des Quelltext-Editors wurde bereits im Kapitel 4.1.1 Konfiguration 1 beschrieben. Eingeschlossen ist die Festlegung der Schriften, der Syntax-Hervorhebung (Syntax-Highlighting) über die Themen, die Code-Formatierung und die Code-Kurzbefehle. In der Konfiguration können Sie unter **Werkzeuge > Einstellungen** die *Syntax-Hervorhebung* konfigurieren.

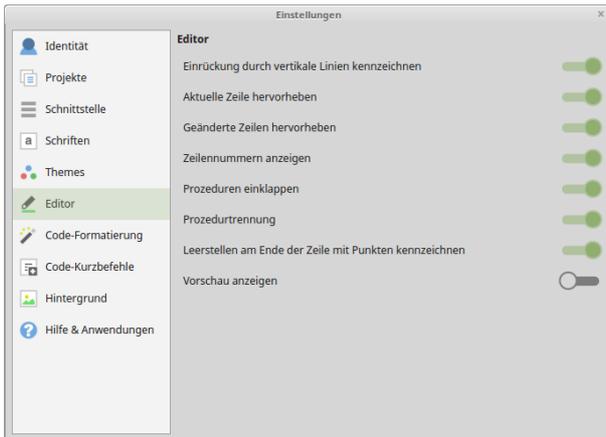


Abbildung 4.5.1.1: Einstellungen – Quelltext-Editor

4.5.2 Quelltext-Editor-Fenster ändern

Sie können das Editor-Fenster mit den Funktionstasten F10 und F11 maximieren oder wieder in Original-Größe darstellen. Eine Teilung des Editor-Fensters in zwei nebeneinander liegende Fenster erreichen Sie über die Tastenkombination CTRL+H. Mit CTRL+J teilen Sie das Editor-Fensters in zwei übereinander liegende Fenster. Der Inhalt der beiden Fenster wird automatisch synchronisiert.

Die beiden Tastenkombination CTRL+H und CTRL+J und die Funktionstasten F10 und F11 wirken dabei jeweils wie ein Umschalter.

4.5.3 Kommentare

In Gambas können Sie vier Arten von Kommentaren in den Quelltext einfügen:

- Einfache Kommentare
- Einfache Kommentare – mit Schlüsselwort – als Aufgaben
- Hilfe-Kommentare auf Klassenebene
- Hilfe-Kommentare auf Symbolebene

4.5.3.1 Einfache Kommentare

Einfache Kommentare im Quelltext beginnen mit einem Apostroph und gehen bis zum Ende der Zeile.

Ein Kommentar wird vom Interpreter ignoriert.

```
Public Sub TBGroup_KeyPress() ' Gilt für die *TextBox-Gruppe* ↳ TBGroup!
  If Left$(Format$(0, ".0")) = "," Then
    CheckInput("+-", |0123456789")
  Else
    CheckInput("+-.", |0123456789")
  Endif
End
' Wird intern NICHT verwendet, da das Gruppen-Event eine höhere Priorität hat.
```

```
'
' Public Sub txbInputA_KeyPress()
'     CheckInput("+-,|0123456789")
' End
' ..
' Public Sub txbInputC_KeyPress()
'     CheckInput("+-,|0123456789")
' End
```

Im Quelltext können Sie eine Text-Zeile oder einen Text-Block markieren und mit der Tastenkombination CTRL+K als Kommentar formatieren oder über das Editor-Menü oder ihn mit CTRL+U wieder entfernen.

Als Alternative für längere Kommentare bietet es sich an, den Kommentar-Text aus einer anderen Quelle zu kopieren und anschließend über das `Spezielles Einfügen` im sich öffnenden Dialog in den Quelltext als Kommentar einzufügen.

4.5.3.2 Kommentare als Aufgaben

Jeder Kommentar im Quellcode, der mit 'TODO:', 'FIXME:' oder 'NOTE:' beginnt, wird automatisch in einem neuen Tab mit dem Bezeichner 'Aufgaben' im unteren Debugging-Panel aufgeführt. Die Schlüsselwörter TODO, FIXME und NOTE – nach einem '-'Zeichen – müssen groß geschrieben werden und mit einem Doppelpunkt abgeschlossen werden. Dem Doppelpunkt vor dem Aufgabentext sollte ein Leerzeichen folgen (optional).

```
' NOTE: Es wird eine Text-Datei `VERSION` im übergeordneten Ordner zum Projekt-Ordner verwendet!
Print "Application.Version = "; Application.Version
```

```
' FIXME: Den Format-String `"+0.###"` prüfen und u.U. ändern.
lblValue.Text = Format$(Deg(ACos(fCosinusAlpha)), "+0.###") & " °"
```

```
' TODO: Die Funktion IsVector(sInput As String) einsetzen, um valide Daten zu verwenden.
Private Function IsVector(sInput As String) As Boolean
```

Diese drei speziellen Kommentare im Quelltext erzeugen diese Ausgabe unter `Aufgaben`:

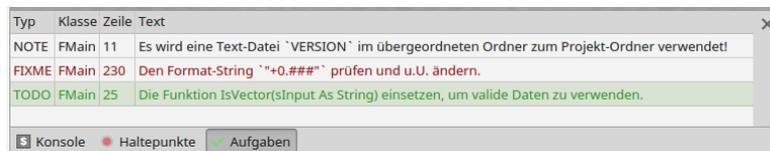


Abbildung 4.5.3.2.1: Anzeige festgelegter Aufgaben

4.5.3.3 Hilfe-Kommentare auf Symbolebene

Ein Hilfe-Kommentar auf Symbolebene beginnt mit zwei '-'Zeichen, dem ein Leerzeichen vor dem Kommentar folgt und steht vor der Zeile mit der Prozedur-Deklaration. In diesem Fall kann der Kommentar auch mehrzeilig sein.

```
'' Set the path to a Markdown project folder in the dialog.
'' The function value is an empty string -> dialog was aborted.
'' The function value is a path specification for a directory.
Private Function SetMDProjectDir() As String

    Dim sDialogDir As String

    sDialogDir = FNewDir.RunDialog()
    If sDialogDir = "" Then
        bCancelDialog = True
        Return ""
    Else
        sCurrentMDDir = sDialogDir
        Return sDialogDir
    Endif
End
```

Wenn Sie dann im Quelltext `SetMD...` schreiben, springt bereits die Auto-Vervollständigung an und zeigt Ihnen in einem kleinen Fenster den *Hilfe*-Kommentar der Funktion an:



Abbildung 4.5.3.3.1: Hilfe-Kommentar auf Symbolebene

Steht der Kommentar nach einer Deklaration, dann ist er stets einzeilig:

```
Property Name As String ' Die Name Eigenschaft zeigt den Namen des Kunden, der dieses Konto besitzt.
```

4.5.3.4 (Hilfe-)Kommentare auf Klassenebene

Hilfe-Kommentare auf Klassenebene müssen von der ersten Zeile im Klassenkopf *Gambas class file* durch eine oder mehrere Leerzeilen getrennt und durch *mindestens* eine Leerzeile von allen folgenden Kommentaren oder folgendem Quelltext getrennt sein.

Das folgende Beispiel zeigt einen Quelltext-Abschnitt aus der Datei *DegTableView.class* mit allen 4 Typen von Kommentaren:

```
' Gambas class file

''' Deganius DegTableView, delivers a Gambas TableView
''' based on data of an array or db-result or a class DegOrm.
...
''' Can display a DegComboBox inside and bind it's index to a column.

'TODO: BeforeUpdate - Cancel does not work right

Export
Inherits TableView

' Component Definition
Public Const _Group As String = "Deg"
Public Const _Family As String = "Form"

Public Const _DrawWith As String = "TableView"
Public Const _Properties As String = "*", ComboButtons, Readonly, AllowDelete, AllowInsert, AllowCopyPaste"

' Constants for ShowAs
Public Const AsString As Integer = 0
Public Const AsCheckbox As Integer = 1
Public Const AsRichText As Integer = 2

'' Number of row that has been changed. If nothing changed, it is -1.
Property RowChanged As Integer
...

```

Unter <https://gitlab.com/christhal/gb.deg.form/blob/master/.src/Comp/DegTableView.class> finden Sie den kompletten Quelltext der Komponente, die Christof Thalhofer entwickelt hat.

Hilfe-Kommentare auf Symbol- und Klassenebene können Sie durch die Verwendung von Text in Markdown-Syntax erweitern.

4.5.4 Lesezeichen

Um ein Lesezeichen zu setzen, müssen Sie zuerst eine Quelltext-Zeile markieren und dann das Lesezeichen mit CTRL+B (Bookmark) definieren. Ein gesetztes Lesezeichen erkennen Sie im Quelltext-Editor an einem blauen Symbol am Zeilenanfang .

- Zwischen den einzelnen Lesezeichen navigieren Sie mit den Tastenkombinationen ALT+Bild ▲ oder mit ALT+Bild ▼.
- Ein einzelnes Lesezeichen löschen Sie mit CTRL+B, nachdem die entsprechende Quelltext-Zeile markiert wurde.
- Alle Lesezeichen löschen Sie nur über Menü> Bearbeiten> Lesezeichen> `Alle Lesezeichen löschen` oder wesentlich schneller über den passenden Eintrag im Kontext-Menü des Quelltext-Editors.

4.5.5 Liste aller existierenden Deklarationen anzeigen

Um im Quelltext schnell zu navigieren, können Sie sich die Liste *aller existierenden Deklarationen* in den Kategorien Methoden (public, private) und Ereignisse anzeigen lassen.

Der Liste im Menü des Quelltext-Editors ist dieses Symbol  vorangestellt. Wählen Sie einen Eintrag mit einem Klick aus (MenüButton), so springt der Cursor zu dieser ausgewählten Deklaration. Sie können dann den Quelltext lesen oder ändern. Das ist eine Funktionalität, die Sie sicher nutzen werden.

4.5.6 Kurzbefehle und Tastenkürzel

Die Gambas-IDE bietet einen modifizierbaren Satz von Quelltext-Kurzbefehlen, die den Schreibaufwand im Quelltext-Editor erleichtern können. Die mit der IDE ausgelieferten Standard-Quelltext-Kurzbefehle finden Sie unter Menü> Werkzeuge> Einstellungen> Code-Kurzbefehle. Um die vordefinierten Quelltext-Kurzbefehle zu verwenden, muss das Kontrollkästchen (Switch-Button) `Code-Kurzbefehle aktivieren` aktiviert sein. Über den Button `Neu...` können Sie im Dialog eigene Kurzbefehle, die so genannten Trigger-Strings, auch mit *mehrzeiligen* Quelltext-Abschnitten in den Erweiterungsmakros anlegen. Hinweise zu den Trigger-Strings und zur Syntax der Erweiterungsmakros finden Sie unter <http://gambaswiki.org/wiki/ide/idesnippets>.

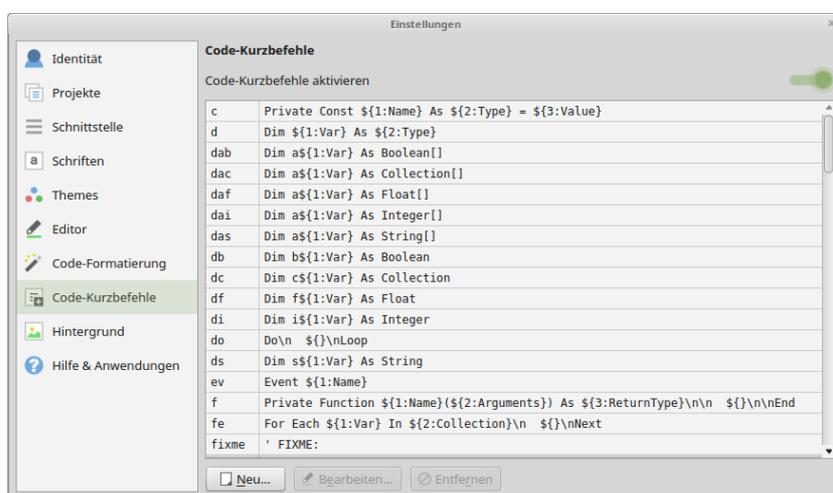


Abbildung 4.5.6.1: (Vordefinierte) Quelltext-Kurzbefehle

Wenn Sie einen Quelltext-Kurzbefehl am Anfang einer Zeile eingeben und anschließend die Tabulator-Taste drücken, so wird der Quelltext-Kurzbefehl automatisch zu einem definierten Quelltext-Abschnitt erweitert. Zum Beispiel gibt es den Kurzbefehl "_c" für die `_compare`-Methode. Sie geben "_c" ein und drücken dann die Tabulator-Taste. Das aktiviert den im Erweiterungsmakro hinterlegten Quelltext-Abschnitt.

Beispiel: Aus dem Quelltext-Kurzbefehl

```
if
```

wird nach dem Drücken der Tabulator-Taste

```
If Test Then
Endif
```

Auf jeden Fall sollten Sie einmal die Quelltext-Kurzbefehle gpl, f und pf ausprobieren und sich dann jeweils die Syntax des Erweiterungsmakros ansehen.

Beachten Sie, dass es unter Menü> Werkzeuge> auch den Eintrag `Kurzbefehle...` gibt, die sich aber auf die Menüs in der IDE beziehen.

So schalten Sie zum Beispiel mit dem Kurzbefehl CTRL+H zu einer geteilten Ansicht mit zwei Quelltext-Editoren um. Eine differenzierte Anzeige zu den aktuell definierten Kurzbefehlen erhalten Sie unter Menü> ?> Tastenkürzel... .

4.5.7 Hinweise zu den Tooltips

Da ein Tooltip-Text auch HTML-Tags unterstützt, können Sie u.a. die *ToolTip-Breite* frei wählbar festlegen, indem Sie eine HTML-Tabelle einfügen und dann die Tabellengröße mit der Width-Eigenschaft festlegen. So erfolgt exemplarisch die Festlegung des Tooltip-Textes im Eigenschaftfenster des Steuerelementes `Button` mit dem Namen btnFlaecheABCV2 unter der Eigenschaft `ToolTip`:

```
<table width=400>
<tr>
<td>
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore
et dolore magna aliquyam erat, sed diam voluptua.
</td>
</tr>
</table>
```

oder im Quelltext für den ausgewählten Button:

```
btnFlaecheABCV2.ToolTip = "<table width=400><tr><td>Lorem ipsum dolor sit amet, consetetur sadipscing
elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua.</td></tr></table>"
```

Das Ergebnis sieht in beiden Fällen identisch aus:



Abbildung 4.5.7.1: Tooltip-Text-Fenster mit einstellbarer Breite

4.5.8 Hilfen

Generell können Sie sich in der IDE mit der Funktionstaste F1 die komplette Gambas-Dokumentation anzeigen lassen – einen Netz-Zugang vorausgesetzt.

Eine Hilfe zu Definitionen wird angezeigt, wenn Sie ein (Schlüssel-)Wort im Quelltext markieren und dann die Funktionstaste F2 drücken. Es reicht bereits aus, wenn der Cursor im (Schlüssel-)Wort oder an dessen Ende steht. Es öffnet sich ein Hilfe-Browser mit dem Hilfetext zum ausgewählten (Schlüssel-)Wort als Ausschnitt der Gambas-Dokumentation. Es wird der englische Text angezeigt, wenn keine deutsche Übersetzung existiert:

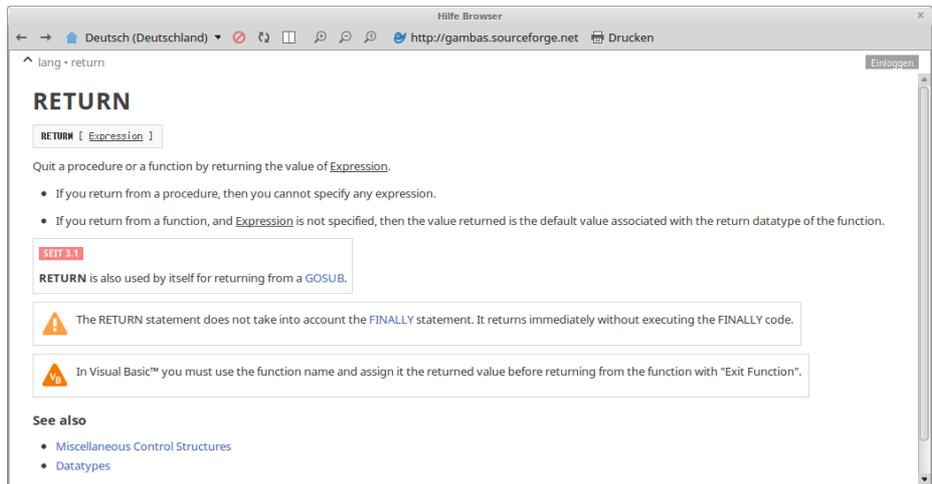


Abbildung 4.5.8.1: Hilfe zu Definitionen

4.5.9 Farbwerte einfügen

Sie können einen Farbwert im Dialog über den Menü-Eintrag  auswählen und ihn einer Farbeigenschaft in drei unterschiedlichen Formaten zum Beispiel so zuweisen:

```
lblValue.Foreground = &HFF0000
```

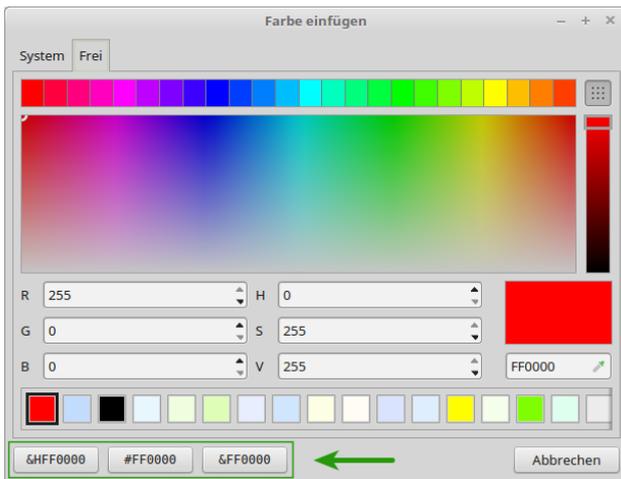


Abbildung 4.5.9.1: Farbwert zuweisen – 1. Format wird genutzt

4.5.10 Aktuelles Datum einfügen

In dieser Liste von aktuellen Datumswerten unter dem Symbol  finden Sie garantiert ein Format, dass Sie benötigen und mit einem Klick in den Quelltext an der Position des Cursors einfügen:

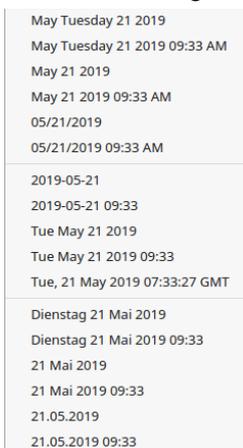


Abbildung 4.5.10.1: Datumsformate

4.5.11 Sonderzeichen einfügen



Abbildung 4.5.11.1: Sonderzeichen

Nur einen weiteren Klick entfernt auf  im Menü des Quelltext-Editors steht Ihnen wie in der Abbildung 4.5.1.1 diese Auswahl von Sonderzeichen zum Einfügen zur Verfügung!

4.5.12 Einfügen von Text und 'Spezielles Einfügen von Text'

Das Einfügen von Text über einen Klick auf  ist mit den drei vorgegebenen Möglichkeiten schnell erledigt:



Abbildung 4.5.12.1: Einfügen von Text

Der letzte Eintrag im Menü besitzt auch ein eigenes Menü und stellt Ihnen mehrere Möglichkeiten zur Verfügung, in welcher Form der Text eingefügt wird. Der Mime-Typ des Inhaltes der Zwischenablage wird automatisch erkannt:

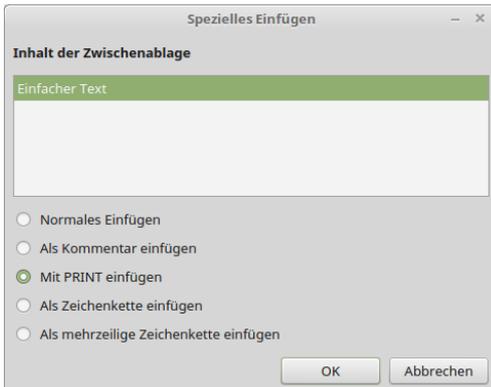


Abbildung 4.5.12.2: Spezielles Einfügen

Wenn Sie etwa eine Datei kopierten, dann wird Ihnen als erkannter Inhalt u.a. auch `URI-Liste` angeboten. Wählen Sie diese Option und anschließend `Mit PRINT einfügen`, so wird der *absolute Pfad zur Datei* so eingefügt:

```
Print "file:///home/hans/BildTon/apple-red.png\r"
```

4.5.13 Quelltext formatieren

Text-Abschnitte im Quelltext können Sie automatisch formatieren lassen, wenn Sie die Option `Beim Laden & Speichern formatieren` in den Einstellungen in der Rubrik *Code-Formatierung* aktivieren. Danach reicht ein CTRL+UMSCHALT+F, um den kompletten Quelltext nach den Vorgaben der IDE – die Sie nicht ändern können – zu formatieren.

4.5.14 Quelltext sperren

Sie können den kompletten Text über Menü> Bearbeiten> Gesperrt auf `schreibgeschützt` setzen oder die Sperre wieder aufheben.

4.5.15 Verwendung von Präfixen

Das Gambas-Buch folgt den Vorgaben unter <http://gambaswiki.org/wiki/doc/namingconvention> für Präfixe für Steuerelemente und Variablen in einer angepassten Form. Quelltexte für die zahlreichen Projekte zu den einzelnen Kapiteln haben somit eine einheitliche Diktion und werden damit vor Allem besser lesbar.

4.5.15.1 Präfix für Steuerelemente:

```
DateChooser: dach, dac
DirChooser:  dich, dic
FileChooser:  fich, fic
ColorChooser: coch, coc
FontChooser: foch, foc
-----
ButtonBox:  bubox, bub
CheckBox:   ckbox,  ckb
ComboBox:  cobox,  cmb
MovieBox:  mobox,  mbox
```

```

PictureBox:  pibox, pib
SliderBox:  slbox
SpinBox:    sbox
TextBox:    txbox
ValueBox:   vbox, vbox
MaskBox:    mabox
DateBox:    dabox
DirBox:     dibox
FontBox:    fobox
ListBox:    libox

```

```

-----
Button:      btn
ColorButton: cobtn
MenuButton:  mebbtn, mbtn
RadioButton: opt, rabtn, rbtn, opb
SwitchButton: swbtn, sbtn
ToggleButton: tglbtn, tbtn
ToolButton:  tobtn

```

```

-----
ProgressBar:  pbar
ScrollBar:    sbar
Separator:    sep
Slider:       sld
Spinner:      spi
ColorPalette: cop
TextArea:    txa
TextEdit:    txe
Dial:        dial
Editor:      edi
FileProperties: fip
IconPanel:   icp
ListEditor:  lie
Form:        frm

```

```

-----
Label:       lbl
TextLabel:   txl
URLLabel:    ulbl, url1
LCDLabel:    lcdlbl, lcd1

```

```

-----
ColumnView:  cov
DirView:     div
DocumentView: dov
FileView:    fiv
GridView:    grv
IconView:    icv
ListView:    liv
ImageView:   imv
TableView:   tav
TreeView:    trv
WebView:     wev
MessageView: mev
MapView:     mav

```

4.5.15.2 Präfix für Variablen (Datentyp)

```

Boolean:     b
Integer:     i
Byte:        ib
Short:       is
Long:        il
Single:      si
Float:       f
Date:        d, date
String:      s
Variant:     v
Array:       a
Collection:  c
Struct:      str, struc
Pointer:     p
Objekt:      o, h

```

Beispiel-Quelltext:

```

Public Sub TCP_IP_Socket_Read()

    Dim sResult, sDate, sTime As String
    Dim s1900 As String = "01/01/1900"
    Dim s1935 As String = "01/01/1935"
    Dim s1970 As String = "01/01/1970"
    Dim i, iCurrentSeconds As Integer
    Dim ilNTPSeconds, ilSeconds19001970 As Long
    Dim dDate1970, dCurrentDateUTC As Date

```

```
' The timestamp in the NTP is 64 bits long.
' 32 bits encode the (past) seconds since January 1st 1900 00:00:00 (!)

If TCPIP_Socket.Status = Net.Connected Then
  Read #TCPIP_Socket, sResult, Lof(TCPIP_Socket)
  ' Conversion of the transmitted 32-bit data word into an integer number (typ long)
  For i = 1 To Len(sResult)
    ilNTPSeconds = ilNTPSeconds * 256 + Asc(Mid$(sResult, i, 1))
  Next
Endif
' Alternative: Number of seconds = b3*2563 + b2*2562 + b1*256 + b0

lTimeResult = ilNTPSeconds
ilSeconds19001970 = DateDiff(s1900, s1935, gb.Second) + DateDiff(s1935, s1970, gb.Second)
iCurrentSeconds = ilNTPSeconds - ilSeconds19001970
dDate1970 = Date(1970, 01, 01, 0, 0, 0)
dCurrentDateUTC = DateAdd(dDate1970, gb.Second, iCurrentSeconds)
sDate = Format$(dCurrentDateUTC, "dddd") & ", " & Format$(dCurrentDateUTC, "d. mmmm yyyy")
sTime = Format$(Time(dCurrentDateUTC), "hh:nn:ss") & " Uhr"
' txbDateTime.Text = "Date " & String.Chr(187) & " " & sDate & " - " & sTime
txbDateTime.Alignment = Align.Center
txbDateTime.Text = sDate & " - " & sTime

End ' TCPIP_Socket_Read
```