6.6 Klasse Stat

Die Klasse Stat (gb) stellt Ihnen alle Informationen zur Verfügung, die durch das System zu einer *bestimmten* Datei zurückgegeben werden. Es muss betont werden, dass ein Stat-Objekt ungeeignet ist, um ausgewählte *Änderungen* an den Datei-Eigenschaften vorzunehmen! Die Klasse kann nicht erzeugt werden.

- Die Stat()-Funktion liefert ein Stat-Objekt zurück, mit dem Sie Informationen zu Datei- und Verzeichniseigenschaften auslesen können.
- Die Informationen werden zu genau einer ausgewählten Datei ausgelesen, deren Pfad Sie in der *Path-Eigenschaft* (Tabelle Eigenschaften) angeben müssen.
- Achtung: Bei den Dateien werden sechs Typen unterschieden.
- Beachten Sie, dass die Informationen zu einer Datei beim Einsatz dieser Klasse statisch sind, weil sie zu einem *bestimmten Zeitpunkt* ausgelesen worden sind. Wenn Sie bestimmte Änderungen einer ausgewählten Datei zur Laufzeit (über einen Task) dynamisch erfassen und auswerten wollen, dann finden Sie in der Komponente *gb.inotify* die dafür geeignete Klasse *Watch*.

6.6.1 Eigenschaften

Die Klasse Stat verfügt über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Path	String	Gibt den Pfad der Datei zurück, die durch das Stat-Objekt referenziert wird oder setzt den Pfad.
Auth	String	Gibt die Datei-Berechtigungen als Zeichenfolge zurück; mit der gleichen Syntax wie die Instruktion CHMOD.
Group	String	Gibt die Gruppe zurück, zu der die Datei gehört.
Hidden	Boolean	Gibt True zurück, wenn die Datei versteckt ist.
LastAccess	Date	Gibt die das Datum und die Zeit für den letzten Zugriff auf die Datei aus.
LastChange	Date	Gibt die Zeit aus, zu der die Attribute der Datei zum letzten Mal geändert wurden.
LastModified	Date	Gibt die Zeit aus, zu der der Inhalt der Datei zum letzten Mal geändert wurde.
Time	Date	Synonym für die LastModified-Eigenschaft.
Link	String	Wenn die Datei ein symbolischer Link ist, dann wird der Pfad der Datei zurückgegeben.
Mode	Integer	Gibt den Modus (Zugriffsmaske) der Datei als Zahl kodiert zurück. Sie sollten die spezifischeren und einfacher zuhandhabenden Eigenschaften wie zum Beispiel Perm oder SetUID benutzen.
SetGID	Boolean	Gibt True zurück, wenn das SetGID-Bit in der Rechte-Maske gesetzt ist.
SetUID	Boolean	Gibt True zurück, wenn das SetUID-Bit in der Rechte-Maske gesetzt ist.
Size	Long	Gibt die Datei-Größe in Bytes zurück.
Sticky	Boolean	Gibt True zurück, wenn das Sticky-Bit in der Rechte-Maske gesetzt ist.
User	String	Gibt den Namen des Benutzers/Eigentümers zurück.
Type	Integer	Gibt den Typ einer Datei zurück (Typ-Symbol oder Zahl).
Perm	.Stat.Perm	Gibt eine virtuelle Klasse zurück, welche die Dateiberechtigungen (Rechte-Maske) beschreibt.

Tabelle 6.6.1.1 : Eigenschaften der Klasse Stat

6.6.2 Datei-Typen

Der Typ einer Datei kann durch eine der folgenden Konstanten beschrieben werden:

Typ - Symbol	Typ - Nummer	Beschreibung
gb.File	1	Normale Datei (Regular file)

Typ - Symbol	Typ - Nummer	Beschreibung
gb.Directory	2	Verzeichnis (Directory)
gb.Device	3	Spezial-Datei für ein Gerät (Special file for a device)
gb.Pipe	4	Benannte Pipe (FIFO-Datei; Named pipe)
gb.Socket	5	Spezial-Datei für einen Socket (Special file for a socket)
gb.Link	6	Symbolischer Link (Symbolic link)

Tabelle 6.6.2.1 : Datei-Typ-Konstanten der Klasse Stat

6.6.3 Klasse .Stat.Perm

Die *virtuelle* Klasse *.Stat.Perm* (gb) beschreibt die Dateiberechtigungen für die ausgewählte Datei und kann wie ein Nur-Lese-Array verwendet werden.

Eigenschaft	Datentyp	Beschreibung
.Stat.Perm.User	String	Gibt die Benutzer/Eigentümer-Berechtigung der Datei als Zeichenkette zurück.
.Stat.Perm.Group	String	Gibt die Gruppen-Berechtigung der Datei als Zeichenkette zurück.
.Stat.Perm.Other	String	Gibt die Standard-Berechtigungen als Zeichenkette zurück.

Tabelle 6.6.3.1 : Eigenschaften der Klasse .Stat.Perm

In allen drei Fällen enthält der (Rechte-)String die folgenden Zeichen, falls das jeweilige Recht vergeben wurde:

r → für das Lese-Recht,
w → für das Schreibrecht,
x → für das Ausführungsrecht.

Achtung: Insbesondere wird für fehlende Berechtigungen kein Zeichen ausgegeben. Dies unterscheidet sich von der Ausgabe von Kommandos wie 'ls -l', bei denen fehlende Berechtigungen mit einem Minus-Zeichen an ihrer festen Position notiert werden.

6.6.4 Projekt

Zuerst wird im vorgestellten Projekt eine temporäre Datei erzeugt und Inhalt eingefügt. Dann werden die Datei-Rechte explizit geändert. Anschließend wird ein Stat-Objekt erzeugt und die Dateiinformationen der temporären Datei werden ausgelesen und angezeigt.

```
' Gambas class file

Public sFilePath As String

Public Sub Form_Open()

  Dim sGroup, sRandomFileName As String

  FMain.Center
  FMain.Resizable = False

  sRandomFileName = SetRandomFileName()
  sFilePath = Temp(sRandomFileName)
  File.Save(sFilePath, "Text line 1.\nText line 2.")

  Chown sFilePath To User.Name
  Exec ["id", "-gn"] To sGroup ' Find out the group of the current user
  sGroup = Trim$(sGroup)
  Chgrp sFilePath To sGroup

  Chmod sFilePath To "rwxrw-r--" ' Test 1 perfect
' Chmod sFilePath To "rwSrwSr-T" ' Test 2 not error-free
' Chmod sFilePath To "rwxrw-rw-" ' Test 3 perfect

' Error correction:
' Shell "chmod u+s " & sFilePath Wait
' Shell "chmod g+s " & sFilePath Wait
' Shell "chmod o+t " & sFilePath Wait
```

```
End

Public Sub btnGetInformation_Click()

  Dim FileInfo As Stat
  Dim aTypes As String[] = ["Regular file", "Directory", "Special file for a device", "Named pipe", /
                            "Special file for a socket", "Symbolic link"]
  Dim hFile As File
  Dim sLine, sLink As String

  FileInfo = Stat(sFilePath) ' txaOutput(TypeOf(Stat(sFilePath)) --> Stat-Object

  txaOutput.Clear()
  txaOutput.Insert(gb.Lf)
  txaOutput.Insert("Path = " & FileInfo.Path & gb.Lf)
  txaOutput.Insert("File-Type (Integer) = " & FileInfo.Type & gb.Lf)
  txaOutput.Insert("File-Type = " & aTypes[FileInfo.Type - 1] & gb.Lf) ' aTypes array index starts with 0
  txaOutput.Insert("Permissions (Symbolic notation) = " & FileInfo.Auth & gb.Lf)
  txaOutput.Insert("File-Modus (Numeric notation)   = " & Oct$(FileInfo.Mode) & gb.Lf)
  txaOutput.Insert("SetUID set? = " & Str(FileInfo.SetUID) & gb.Lf)
  txaOutput.Insert("SetGID set? = " & Str(FileInfo.SetGID) & gb.Lf)
  txaOutput.Insert("Sticky-Bit set? = " & Str(FileInfo.Sticky) & gb.Lf)
  txaOutput.Insert("User permissions  = " & FileInfo.Perm.User & gb.Lf)
  txaOutput.Insert("Group permissions = " & FileInfo.Perm.Group & gb.Lf)
  txaOutput.Insert("Other permissions = " & FileInfo.Perm.Other & gb.Lf)
  txaOutput.Insert(gb.Lf)
  txaOutput.Insert("File-Size = " & FileInfo.Size & " Byte" & gb.Lf)
  sLink = IIf(FileInfo.Link, FileInfo.Link, "No")
  txaOutput.Insert("Symbolic link? = " & sLink & gb.Lf)
  txaOutput.Insert("Contents of the text file:" & gb.Lf)
  hFile = Open sFilePath For Input
  While Not Eof(hFile)
    Line Input #hFile, sLine
    txaOutput.Insert(sLine & gb.Lf)
  Wend
  txaOutput.Insert(gb.Lf)
  txaOutput.Insert("User = " & FileInfo.User & gb.Lf)
  txaOutput.Insert("Group = " & FileInfo.Group & gb.Lf)
  txaOutput.Insert("File hidden? = " & Str(FileInfo.Hidden) & gb.Lf)
  txaOutput.Insert("Permissions (Numeric notation)  = " & ModeStringToOctalString(FileInfo.Auth) & gb.Lf)
  txaOutput.Insert("Last access to file = " & FileInfo.LastAccess & "(UTC)" & gb.Lf)
  txaOutput.Insert("Last change of file content = " & FileInfo.LastModified &  "(UTC)" &  gb.Lf)
  txaOutput.Insert("Last change of file attributes  = " & FileInfo.LastChange &  "(UTC)" & gb.Lf)
  txaOutput.Insert("Permission R or W  = " & Str(Access(sFilePath, gb.Read Or gb.Write)) & gb.Lf)
  txaOutput.Insert("Permission R       = " & Str(Access(sFilePath, gb.Read)) & gb.Lf)
  txaOutput.Insert("Permission W       = " & Str(Access(sFilePath, gb.Write)) & gb.Lf)
  txaOutput.Insert("Permission X       = " & Str(Access(sFilePath, gb.Exec)) & gb.Lf)
' gb.read is default if the optional mode argument is missing
  txaOutput.Insert("Permission R       = " & Str(Access(sFilePath)) & gb.Lf)

End

Private Function SetRandomFileName() As String

  Dim sFileName As String

  sFileName = Hex$(Rand(0, 2 ^ 32 - 1)) ' Random file name
  Return sFileName

End

Public Sub ModeStringToOctalString(ModeString As String) As String

  Dim i, k, iValue, iExtendedFileAttribute As Integer
  Dim sOctal, s As String

  For k = 1 To 7 Step 3
    s = Mid(ModeString, k, 3)
    iValue = 0
    For i = 1 To 3
      Select Mid(s, i, 1)
        Case "r"
          iValue += 4
        Case "w"
          iValue += 2
        Case "x", "s", "t"
          iValue += 1
      End Select

      If k = 1 Then
         Select Mid(s, i, 1)
           Case "s", "S"
             iExtendedFileAttribute += 4
```

```
        End Select
      Endif
      If k = 4 Then
         Select Mid(s, i, 1)
           Case "s", "S"
           iExtendedFileAttribute += 2
         End Select
      Endif
      If k = 7 Then
         Select Mid(s, i, 1)
           Case "t", "T"
           iExtendedFileAttribute += 1
         End Select
      Endif
    Next
    sOctal &= CStr(iValue) ' String!
  Next

  If iExtendedFileAttribute = 0 Then
     Return sOctal
  Else
     Return CStr(iExtendedFileAttribute) & sOctal ' String!
  Endif

End
```

Ausgabe:

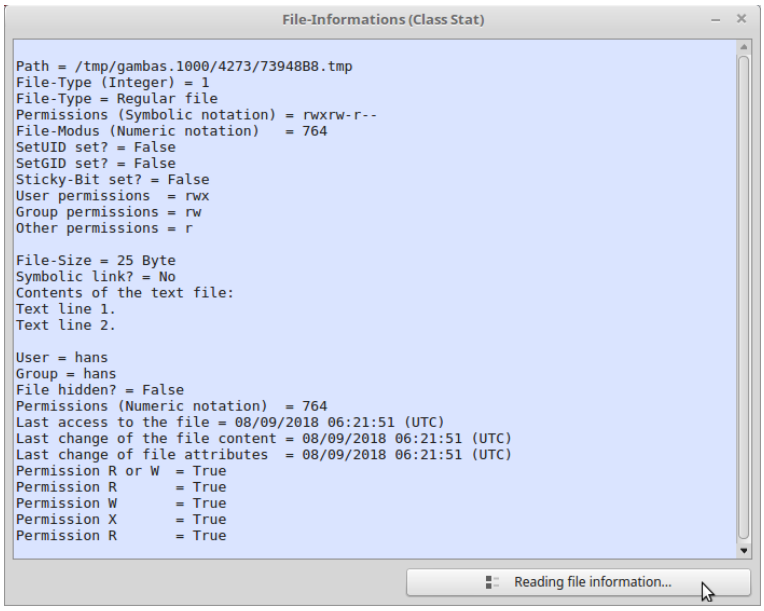

Abbildung 6.6.4.1: Ausgabe Datei-Informationen

Eine Alternative zum Einsatz der virtuellen Klasse *.Stat.Perm* bietet die Access-Funktion an, zu der Sie Informationen unter dem Link http://gambaswiki.org/wiki/lang/access nachlesen können und die auch punktuell im o.a. Projekt verwendet wurde.

Achtung: Alle Dateien im logischen Ordner Data (und auch in allen Unterverzeichnissen darin) im Projektverzeichnis werden als read-only behandelt, egal wie die tatsächlichen Modus-Einstellungen lauten. Das gilt auch bei Dateien mit dem Modus 777 (voller Lese- und Schreibzugriff für Besitzer, Gruppe und Andere). Das heißt zum Beispiel, dass Access("texts/mytext.txt") niemals gb.Write zurück gibt. Der Grund liegt darin, dass beim Ausführen der ausführbaren Datei *.gambas alle Projektdateien aus dem Ordner Data in die ausführbare Datei eingefügt werden und daher zur Laufzeit nicht geändert werden können!

Beachten Sie, dass die Instruktion CHMOD bis zur (stabilen) Gambas-Version 3.11.3 (teilweise) fehlerhaft ist. Eine Korrekturmöglichkeit wird im vorgestellten Projekt angegeben. Der Fehler wurde mit Commit d44b4bd7f in der Entwicklerversion beseitigt.

Zum Ändern von Datei-Attributen stehen Ihnen die drei Instruktionen CHGRP, CHMOD und CHOWN zur Verfügung, die im nächsten Kapitel beschrieben werden.