

### 19.3.0 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 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* angeben müssen → Tabelle 19.3.0.1.1.
- Achtung: Bei den Dateien werden sechs Typen unterschieden → Kapitel 19.3.0.2.
- 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*.

#### 19.3.0.1 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.
Auth	String	Gibt die Datei-Berechtigungen als Zeichenfolge zurück; mit der gleichen Syntax wie CHMOD → Kapitel 19.3.1.
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 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 zu handhabenden 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 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, dem die Datei gehört.

Eigenschaft	Datentyp	Beschreibung
Type	Integer	Gibt den Typ einer Datei zurück.
Perm	.Stat.Perm	Gibt eine virtuelle Klasse zurück, welche die Dateiberechtigungen (Rechte-Maske) beschreibt.

Tabelle 19.3.0.1.1 : Eigenschaften der Klasse Stat

#### 19.3.0.2 Datei-Typen

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

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

Typ	Typ	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 19.3.0.2.1 : Datei-Typ-Konstanten der Klasse Stat

### 19.3.0.3 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 19.3.0.3.1 : Eigenschaften der Klasse .Stat.Perm

In allen drei Fällen enthält der String die folgenden Zeichen:

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

falls das jeweilige Recht vergeben wurde. 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.

### 19.3.0.4 Projekt

Zuerst wird im vorgestellten Projekt eine temporäre Datei generiert und eine Zeichenkette als Inhalt eingefügt. Dann werden die Datei-Rechte explizit geändert. Anschließend wird ein Stat-Objekt erzeugt und die Datei-Informationen der temporären Datei ausgelesen und in der Konsole der IDE angezeigt.

Der Quelltext wird vollständig angegeben:

```
' Gambas class file

Public sFilePath As String = Temp(".datei")

Public Sub Form_Open()
  Dim sGroup As String

  FMain.Center
  FMain.Resizable = False

  Exec ["id", "-gn"] To sGroup ' Gruppe des aktuellen Benutzers ermitteln
  sGroup = Trim$(sGroup)

  Shell Subst$("touch &l; echo 'TEST TEXT TEXT' > &l", sFilePath) Wait
  Chown sFilePath To User.Name
  Chmod sFilePath To "rwxrw-r--"
  Chgrp sFilePath To sGroup

End ' Form_Open()

Public Sub btnGetInformation_Click()
  Dim FileInfo As Stat
  Dim vValue As Variant
  Dim aTypes As String[]
  Dim cCollection As Collection

  aTypes = ["Normale Datei", "Verzeichnis", "Gerät", "Pipe (FIFO)", "Socket", "Symbolischer Link"]
```

```

FileInfo = Stat(sFilePath)

cCollection = New Collection
' cCollection[Key As String] = Value As Variant → Zuerst der Schlüssel und dann der Wert
cCollection["Pfad = "] = FileInfo.Path
cCollection["Datei-Typ = "] = aTypes[FileInfo.Type - 1]
cCollection["Rechte = "] = FileInfo.Auth
cCollection["Datei-Modus = "] = FileInfo.Mode
cCollection["Benutzer = "] = FileInfo.User
cCollection["Gruppe = "] = FileInfo.Group
cCollection["Versteckt? = "] = FileInfo.Hidden
cCollection["Letzter Zugriff auf die Datei = "] = FileInfo.LastAccess
cCollection["Letzte Änderung Datei-Inhalt = "] = FileInfo.LastModified
cCollection["Letzte Änderung Datei-Attribute = "] = FileInfo.LastChange
cCollection["Datei-Größe = "] = FileInfo.Size & " Byte"
cCollection["Sticky-Bit gesetzt? = "] = FileInfo.Sticky
If FileInfo.Link = Null Then
    cCollection["Symbolischer Link ? = "] = "Nein"
Else
    cCollection["Symbolischer Link ? = "] = FileInfo.Link
Endif
cCollection["Benutzer-Rechte = "] = FileInfo.Perm.User
cCollection["Gruppen-Rechte = "] = FileInfo.Perm.Group
cCollection["Andere-Rechte = "] = FileInfo.Perm.Other

For Each vValue In cCollection
    Print cCollection.Key, vValue
Next

Print "Pfad = ", sFilePath; "    Recht R+W = ", Access(sFilePath, gb.Read Or gb.Write)
Print "Pfad = ", sFilePath; "    Recht R = ", Access(sFilePath, gb.Read)
Print "Pfad = ", sFilePath; "    Recht W = ", Access(sFilePath, gb.Write)
Print "Pfad = ", sFilePath; "    Recht X = ", Access(sFilePath, gb.Exec)
' gb.read ist Standard, wenn das optionale Mode-Argument fehlt
Print "Pfad = ", sFilePath; "    Recht (R) = ", Access(sFilePath)

End ' GetInformation

```

Ausgabe in der Konsole der IDE:

```

Pfad =          /tmp/gambas.1000/6251/.datei.tmp
Datei-Typ =     Normale Datei
Rechte =       rwxrw-r--
Datei-Modus =   500
Benutzer =     hans
Gruppe =       hans
Versteckt? =   True
Letzter Zugriff auf die Datei = 01.10.2015 14:32:10
Letzte Änderung Datei-Inhalt = 01.10.2015 14:32:10
Letzte Änderung Datei-Attribute = 01.10.2015 14:32:10
Datei-Größe =   15 Byte
Sticky-Bit gesetzt? = False
Symbolischer Link ? = Nein
Benutzer-Rechte = rwx
Gruppen-Rechte = rw
Andere-Rechte = r
Pfad =          /tmp/gambas.1000/6251/.datei.tmp    Recht R+W = True
Pfad =          /tmp/gambas.1000/6251/.datei.tmp    Recht R = True
Pfad =          /tmp/gambas.1000/6251/.datei.tmp    Recht W = True
Pfad =          /tmp/gambas.1000/6251/.datei.tmp    Recht X = True
Pfad =          /tmp/gambas.1000/6251/.datei.tmp    Recht (R) = True

```

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

Zum Ändern von Datei-Attributen stehen Ihnen die drei Instruktionen `CHGRP`, `CHMOD` und `CHOWN` zur Verfügung → Kapitel 19.3.1.