

6.12.0 Temporäre Verzeichnisse und temporäre Dateien

Ein temporäres Verzeichnis dient zur zeitlich begrenzten Speicherung von Dateien. Temporäre Dateien nutzen Sie zur zeitlich begrenzten Speicherung von Daten. Temporäre Verzeichnisse und temporäre Dateien werden nach dem Gambas-Programm-Ende vom Betriebssystem automatisch gelöscht.

Wenn Sie zum Beispiel Ihre Scans in TIFF-Dateien archivieren, um sie später in das PNG- oder JPG-Format zu konvertieren, dann bietet es sich an, eine Auswahl an TIFF-Dateien in ein temporäres Verzeichnis zu kopieren und diese dann zu konvertieren. Beenden Sie abschließend Ihr Gambas-Konvertierungsprogramm, dann werden die ausgewählten TIFF-Dateien im temporären Verzeichnis und auch das temporäre Verzeichnis automatisch gelöscht.

Benötigt werden temporäre Dateien, wenn Sie etwa ein Bash-Skript in einem Gambas-Programm verwenden wollen und das Skript *script.sh* im Projekt-Ordner in einem Verzeichnis *data* liegt. Rufen Sie nach dem Start der ausführbaren Datei *.gambas das Skript zur Laufzeit auf, endet das mit einer Fehlermeldung, da Sie keine Berechtigung haben, das Skript auszuführen. Der Grund liegt darin, dass alle Dateien im Gambas-Archiv *.gambas nur gelesen werden können! Ein Ausweg mit dem Einsatz einer temporären Datei, die Sie lesen und ausführen können, wird im Folgenden kurz skizziert:

```
Public sScriptPath As String
Public sTempScriptPath As String
Public sStore As String

Public Sub Form_Open()

    sScriptPath = "./data/test_script.sh"           ' Pfad zum Skript (Original) festlegen
    sTempScriptPath = Temp(File.BaseName(sScriptPath)) ' Pfad zu einer temporären Datei festlegen
    Copy sScriptPath To sTempScriptPath           ' Original-Skript in temporäre Datei kopieren

    Chmod sTempScriptPath To "r-xr-xr-x"         ' Ausführungsrechte für die Kopie setzen
    ' Chmod sTempScriptPath To "..x..x..x"       ' Alternative
End

Public Sub btnRunScript_Click()
    Shell sTempScriptPath To sStore              ' Skript-Kopie in einer Shell-Instruktion
    ...                                          ' mit den erforderlichen Rechten ausführen ...
End
```

Die temporäre Datei */tmp/gambas.1000/<PID>/test_script.tmp* – mit der gearbeitet wurde – wird nach dem Programm-Ende automatisch gelöscht.

6.12.0.1 Funktionen Temp() und Temp\$()

```
File name = Temp$( [ Prefix ] )
File name = Temp( [ Prefix ] )
```

Die Funktionen Temp() und Temp\$() geben den Pfad für eine temporäre Datei oder ein temporäres Verzeichnis zurück. Der Pfad hat die folgende Form:

```
/tmp/gambas.<UserID>/<ProcessID>/<Prefix>.tmp
```

- UserID ist die Systemkennung des Benutzers.
- ProcessID ist die Prozess-ID (PID) des aktuellen Prozesses.
- Prefix repräsentiert einen (optionalen) Basis-Dateinamen (ohne Extension!).
- Die Extension einer temporären Datei oder eines temporären Verzeichnisses ist immer .tmp und wird automatisch vergeben.

Wenn kein Präfix angegeben ist, wird er durch eine Ganzzahl ersetzt, die bei jedem weiteren Aufruf dieser Funktion automatisch erhöht wird. Auf diese Weise ist der zurückgegebene Pfad-Name immer eindeutig. Diese Funktion gibt nur einen Pfad zurück. Sie können anschließend diesen Pfad verwenden, um zum Beispiel eine Datei, ein Verzeichnis, einen Unix-Socket oder einen Symlink temporär zu erzeugen.

Wichtig zu wissen:

- Ein angemeldeter System-Benutzer hat Schreib- und Leserechte auf das temporäre Verzeichnis oder auf die temporäre Datei.

- Alle Dateien im temporären Verzeichnis /tmp/gambas.<UserID>/<ProcessID><*.tmp> werden nach dem Ende des Gambas-Programms automatisch entfernt.
- Auch das Verzeichnis /tmp/gambas.<UserID>/<ProcessID><*.tmp> wird gelöscht.

Beispiel:

```
Dim sTempDir As String = Temp$()

Mkdir sTempDir

' Entpackt ein Archiv in das angelegte temporäre Verzeichnis.
Exec ["tar" "-zxf", "<path_to_archiv>/archive.tar.gz", "-C", sTempDir] Wait

' Erzeugt die Instruktion EXEC [...] einen Fehler, werden Sie so informiert:
If Process.LastValue Then Print Error.Code
...

```

Mit diesem Quelltext-Abschnitt

```
' Gambas class file

Public Sub btnStart_Click()

    Dim sRuntimeDir, sTempDir As String

    txaResults.Clear()
    txaResults.Insert(gb.Lf)
    ' 1. Call of the function Temp$()
    sTempDir = Temp$()
    txaResults.Insert("File-Path 1 = " & sTempDir & gb.Lf)
    File.Save(sTempDir, "1")

    ' 2. Call of the function Temp() - alternative syntax
    sTempDir = Temp()
    txaResults.Insert("File-Path 2 = " & sTempDir & gb.Lf) ' 2. Aufruf der Funktion mit alternativer Syntax
    File.Save(sTempDir, "2")

    ' 3. Call of the function Temp(...) with default of a freely defined file name 'test'
    sTempDir = Temp$("test")
    txaResults.Insert("File-Path 3 = " & sTempDir & gb.Lf)
    File.Save(sTempDir, "3")

    ' 4. Creation of the file with the path /tmp/gambas.1000/<PID>/link.tmp and the content "gambas-book.net"
    sTempDir = Temp$("content")
    txaResults.Insert("File-Path 4 = " & sTempDir & gb.Lf)
    File.Save(sTempDir, "gambas-book.net") ' Datei
    txaResults.Insert("File-Content = " & File.Load(sTempDir) & gb.Lf)

    ' 5. Creation of a file in a special temporary directory
    sRuntimeDir = Desktop.RuntimeDir & "time.txt"
    txaResults.Insert("File-Path 5 = " & sRuntimeDir & gb.Lf)
    File.Save(sRuntimeDir, Date.ToUnixTime(Now())) ' File
    txaResults.Insert("Unix Timestamp = " & File.Load(sRuntimeDir) & gb.Lf)
    txaResults.Insert("Date from Timestamp = " & Date.FromUnixTime(File.Load(sRuntimeDir)) & gb.Lf)

    ' 6. Creation of an empty text file in a temporary directory
    sTempDir = File.Dir(Temp()) & "gambas-book"
    If Exist(sTempDir) Then
        DeleteDirRecursively(sTempDir)
        Rmdir sTempDir
    Endif
    ' If Exist(sTempDir) Then Shell.Rmdir(sTempDir) ' Effective alternative!
    Shell.Mkdir(sTempDir)
    File.Save(sTempDir & "values.log", "")
    txaResults.Insert("File-Path 6 = " & sTempDir & "values.log" & gb.Lf)

End

Public Sub DeleteDirRecursively(sDir As String)

    Dim sFile, sDirectory As String

    For Each sFile In RDir(sDir, "")
        If Not IsDir(sFile) Then
            Kill sDir & sFile
        Endif
    Next
    For Each sDirectory In RDir(sDir, "", gb.Directory)
        Rmdir sDirectory
    Next
End

```

ergeben sich diese Ausgaben:

```

Temporary directories and temporary files
File-Path 1 = /tmp/gambas.1000/6837/1.tmp
File-Path 2 = /tmp/gambas.1000/6837/2.tmp
File-Path 3 = /tmp/gambas.1000/6837/test.tmp
File-Path 4 = /tmp/gambas.1000/6837/content.tmp
File-Content = gambas-book.net
File-Path 5 = /run/user/1000/time.txt
Unix Timestamp = 1539592297
Date from Timestamp = 10/15/2018 08:31:37
File-Path 6 = /tmp/gambas.1000/6837/gambas-book/values.log
Start

```

Abbildung 6.12.0.1.1: Ergebnisse 1

Bei einem weiteren Start ergeben sich weitere temporäre Dateien sowie geänderte Inhalte:

```

Temporary directories and temporary files
File-Path 1 = /tmp/gambas.1000/6837/4.tmp
File-Path 2 = /tmp/gambas.1000/6837/5.tmp
File-Path 3 = /tmp/gambas.1000/6837/test.tmp
File-Path 4 = /tmp/gambas.1000/6837/content.tmp
File-Content = gambas-book.net
File-Path 5 = /run/user/1000/time.txt
Unix Timestamp = 1539592319
Date from Timestamp = 10/15/2018 08:31:59
File-Path 6 = /tmp/gambas.1000/6837/gambas-book/values.log
Start

```

Abbildung 6.12.0.1.2: Ergebnisse 2

Das Besondere im Quelltext ist die Löschung des temporären Verzeichnisses mit den alternativen Ansätzen über `Shell.Rmdir(sTempDir)` oder über die Prozedur `DeleteDirRecursively(sDir As String)`.

6.12.0.2 Exkurs: Eigenschaft `Desktop.RuntimeDir` (gb.desktop)

Das Verzeichnis `/run` ist ein virtuelles, temporäres Dateisystem (<https://de.wikipedia.org/wiki/Tmpfs>). Es existiert im Arbeitsspeicher und wird automatisch geleert, wenn der Computer neu gestartet wird. Die Umgebungsvariable `$XDG_RUNTIME_DIR` definiert einen guten Platz für Ihre Laufzeit-Dateien. Es ist das Verzeichnis `/run/user/<user-id>`, wenn die Umgebungsvariable `$XDG_RUNTIME_DIR` auf den Standardwert gesetzt ist. Es scheint, als ob `/run/user/<user-id>` dem `/tmp`-Verzeichnis Konkurrenz macht. Offensichtlich scheint `/run` moderner zu sein, wie es die Beschreibung auf <https://unix.stackexchange.com/questions/13972/what-is-this-new-run-filesystem/13983#13983> andeutet. Es ist sicher ein guter Plan, wenn jeder Benutzer unter `/run/user/<user-id>` von vorn herein sein eigenes (temporäres) Verzeichnis hat.

```
$ echo $XDG_RUNTIME_DIR
/run/user/1000
```

Obwohl Gambas gegenwärtig mit `Temp$()` nur `/tmp` unterstützt und sein "gambas.<user-id>" selbst erzeugt haben Sie auch Zugriff auf das (Basis-)Verzeichnis `/run/user/1000`. Sie erhalten als Benutzer Zugriff über den Wert der Eigenschaft `Desktop.RuntimeDir` (gb.desktop). Diese Eigenschaft (Datentyp String) gibt das Basis-Verzeichnis zurück, in dem benutzerspezifische, aber nicht zwingend zur Laufzeit benötigte Daten und andere Stream-Objekte wie zum Beispiel Unix-Sockets oder benannte Pipes (Named Pipes) gespeichert werden sollen. Der Benutzer muss mit Unix-Zugriffsmodus 0700 der einzige sein, der vollen Zugriff auf das Verzeichnis `/run/user` hat - was schnell geprüft werden kann:

```

hans@mint-183 ~ $ cd /run/user
hans@mint-183 /run/user $ ls -l
insgesamt 0
drwx----- 7 hans hans 180 Aug 24 10:37 1000

```