11.11 Sicherung von Projekten

Der Autor arbeitet seit vielen Jahren unter http://www.gambas-buch.de an der Online-Version des Gambas-Buches. Daher ist die Anzahl der bearbeiteten Gambas-Projekte in den über 30 Buch-Kapiteln sehr groß. Diese Projekte werden deshalb von Anfang an regelmäßig auf mehrere mobile Datenträger gesichert.

11.11.1 Variante 1

In unregelmäßigen Abständen werden die Gambas-Projekte auf DVD gebrannt und gut verwahrt.

11.11.2 Variante 2

Für die regelmäßige Sicherung der Gambas-Projekte wird das Programm 'Datensicherungswerkzeug' (→ Mint 17.3) eingesetzt, das einfach zu bedienen ist und mit den erprobten Einstellungen ein Archiv auf dem Ziel-Datenträger speichert:

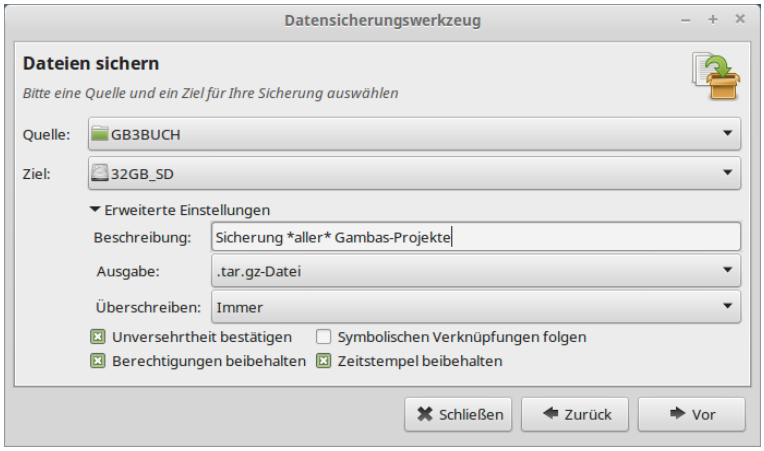

Abbildung 11.11.2.1: Datensicherungswerkzeug in Mint 17.3

Hinweise:

- Sie können in einem weiteren Schritt ausgewählte Ordner und Dateien von der Sicherung ausschließen.
- Bevor Sie die Sicherung starten, sollten Sie die Einstellungen überprüfen!
- Der Archiv-Name folgt der Syntax: jahr-monat-tag-id-backup.tar.gz, wenn Sie bei der Ausgabe ein tar.gz-Archiv wählen:

Abbildung 11.11.2.2: Archiv-Datei

11.11.3 Variante 3

Favorit bei der Sicherung der Gambas-Projekte ist immer noch ein kleines Shell-Skript für das Programm 'rsync'. Das Skript synchronisiert den Ordner mit allen Gambas-Projekten gegen den (gleichnamigen) Ordner auf einem mobilen Datenträger. Alle Änderungen im Projekt-Ordner werden in einem Backup-Ordner gespeichert, so dass zum Beispiel auch gelöschte Projekte dort mit gesichert werden. Es ist von Vorteil, dass man in der Ordner-Struktur der Sicherung jederzeit nach interessierenden Projekten suchen kann. Der Bash-Quelltext für das Skript (projekte2usbstick.sh) ist erfreulich kurz:

```
[1]  #!/bin/sh
[2]  #
[3]  # FARB-WERTE:
[4]  RO="\033[31m"                  # Schriftfarbe rot
[5]  FR="\033[1m\033[31m"           #
[6]  FG="\033[1m\033[42m\033[37m"   # Schriftfarbe fett + weiß + grüner Hintergrund
```

```
[7]  NO="\033[0m"                  # Schriftfarbe normal
[8]  #
[9]  echo
[10] STICK="16_GB_P" # Geräte-Label USB-Stick (16GB)
[11] #
[12] df -k | grep /media/$USER/$STICK 1>/dev/null 2>/dev/null
[13] if [ $? -gt 0 ];
[14] then
[15]   echo "${RO}Der USB-Stick $STICK muss eingehängt sein!";
[16]   echo "Das Sicherungsprogramm wird deshalb beendet!${NO}";
[17]   echo
[18]   echo "Beenden mit ENTER!";
[19]   read a
[20]   exit 1;
[21] else
[22] echo
[23] echo "${FG}Sicherung Gambas-Projekte auf USB-Stick $STICK mit ENTER starten ...${NO}"
[24] echo "_____________________________________________________________________________";
[25] echo
[26] read a
[27] # Muster:
[28] # sudo rsync -ab --progress --delete --backup-dir=BACKUP-DIR SOURCE-DIR/ TARGET-DIR
[29] sudo rsync -ab --progress --delete --backup-dir=/media/$USER/$STICK/GB3Backup $HOME/GB3Projekte/
       /media/$USER/$STICK/GB3Projekte
[30] #
[31] echo
[32] echo "_____________________________________________________________________________";
[33] echo
[34] echo "${FR}Ende der Sicherung auf $STICK.${NO}"
[35] echo
[36] echo "${FR}Der USB-Stick $STICK wird anschließend ausgehängt!${NO}"
[37] echo "${FR}Abschluss mit ENTER!${NO}"
[38] read b
[39] fi
[40] echo
[41] echo "${FG}Unbedingt warten, bis der USB-Stick $STICK ausgehangen worden ist!${NO}";
[42] echo
[43] umount /media/$USER/$STICK
```

Hinweise:

- Prüfen Sie, ob das Programm 'rsync' installiert ist. Holen Sie die Installation u.U. nach.
- Bevor Sie ernsthaft loslegen, lesen Sie sich bitte Details zum Konsolen-Programm *rsync* mit 'man rsync' und 'rsync -h' aufmerksam durch.
- Viele Beispiele zeigt die Seite https://wiki.ubuntuusers.de/rsync/.
- Nutzen Sie den verwendeten USB-Stick nur für Sicherungen.
- Das Skript muss ausführbar sein.
- Für erste Tests sollten Sie mit einem Test-Verzeichnis mit 2 zu sichernden Verzeichnissen und einigen Dateien beginnen.
- Die ersten Tests sollten Sie mehrfach mit der Option --dry-run (Kurzform -n) durchführen bis alles passt – so erleben Sie keine (bösen) Überraschungen. Bedenken Sie, dass Sie mit Root-Rechten arbeiten und in großem Umfang Verzeichnisse und Dateien gelöscht, verschoben und kopiert werden.

```
$ sudo rsync -ab --dry-run  --progress --delete --backup-dir=/media/$USER/$STICK/GB3Backup $HOME/GB3Projek-
te/ /media/$USER/$STICK/GB3Projekte
```

Es folgt die (stark gekürzte) Ausgabe für einen Trocken-Synchronisationsdurchlauf :

```
Sicherung Gambas-Projekte auf USB-Stick 16_GB_P mit ENTER starten ...
_____________________________________________________________________

[sudo] password for hans:

sending incremental file list
created directory /media/hans/16_GB_P/GB3Projekte
Created backup_dir /media/hans/16_GB_P/GB3Backup/
./
CSV2XML_WRITER/
CSV2XML_WRITER/.directory
CSV2XML_WRITER/.gitignore
...
CSV2XML_WRITER/Symbols/form_icon.png
CSV2XML_WRITER/Symbols/projekt_icon.png
DATA2XML_WRITER/
DATA2XML_WRITER/.directory
```

```
...
DBDATA2XML_WRITER/Symbols/report32.png

sent 3,478 bytes  received 476 bytes  7,908.00 bytes/sec
total size is 424,536  speedup is 107.37 (DRY RUN)
_____________________________________________________________________

Ende der Sicherung auf 16_GB_P.
Der USB-Stick 16_GB_P wird anschließend ausgehängt!
Abschluss mit ENTER!

Unbedingt warten, bis der USB-Stick 16_GB_P ausgehangen worden ist!
```

- Wenn der USB-Stick für die Sicherung nicht eingehängt ist, so wird das erkannt und angezeigt sowie das Skript beendet:

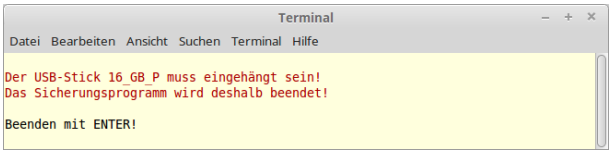

Abbildung 11.11.3.1: Fehleranzeige

- Wenn alle Tests ohne Fehler durchgeführt worden sind, dann können Sie Ihre Gambas-Projekte sichern.
- Nach einem Synchronisationsdurchlauf sind die ausgewählten Ordner auf der Festplatte (Quelle) auch 1:1 auf dem USB-Stick (Ziel). Veränderungen befinden sich im Backup-Ordner auf dem USB-Stick.

Diese Zeile* zur synchronen Sicherung eines Verzeichnisses – mit Backup – leistet die Hauptarbeit:

```
$ sudo rsync -ab --progress --delete --backup-dir=/media/$USER/$STICK/GB3Backup $HOME/GB3Projekte/ /media/$USER/$STICK/GB3Projekte
```

Option - Parameter	Beschreibung
-a	Liste erprobter Optionen (→ Manual)
-b	Backups werden erstellt.
--dry-run (Kurzform: -n)	Es wird eine Synchronisation simuliert. Es erfolgt ein Trocken-Testlauf, der nichts wirklich synchronisiert!
--progress	Der Fortschritt während des Synchronisierens wird angezeigt.
--delete	Dateien, die in der Quelle nicht mehr existieren, werden auch im Ziel gelöscht.
--backup-dir= /media/$USER/$STICK/GB3Backup	Was nicht synchronisiert werden kann, weil verschoben oder im Original gelöscht, wird im Backup-Verzeichnis aufgehoben.
$HOME/GB3Projekte/	Quellverzeichnis im Home-Verzeichnis auf Festplatte. Achten Sie auf das Slash-Zeichen am Ende!
/media/$USER/$STICK/GB3Projekte	Zielverzeichnis auf dem USB-Stick im Wurzelverzeichnis.

Tabelle 11.11.3.1 : Übersicht zu Optionen und Parametern in der Kommando-Zeile*

Sie können jederzeit weitere Zeilen in den Skript-Quelltext einfügen, wenn weitere Verzeichnisse mit ihren Unterverzeichnissen und den darin enthaltenen Dateien synchronisiert gesichert werden sollen.

- Im Download-Bereich finden Sie ein Bash-Skript mit dem o.a. Quelltext. Sie müssen für Ihre Sicherung den Quelltext in den Zeilen 10 und 29 ändern, wenn Sie auf einen USB-Stick sichern.
- Beachten Sie: Die Nutzung des Skripts *projekte2usbstick.sh* erfolgt auf eigene Gefahr!

11.11.4 Variante 4

Diese Variante der Datensicherung wurde in Form eines Gambas-Frontends für das Konsolen-Programm 'rsync' realisiert. Diese Anwendung erlaubt das Anlegen und Editieren einer Jobliste für das Synchronisieren von Verzeichnissen oder einzelner Dateien.

Die optionale Anwendung von Root-Rechten bei Shell-Hintergrundprozesse wird durch die Option '-S' des 'sudo'-Kommandos erreicht, die eine Umleitung der Passwortabfrage in die Error-Routine des Prozesses bewirkt, wo das Root-Passwort an den Prozess übergeben werden kann:

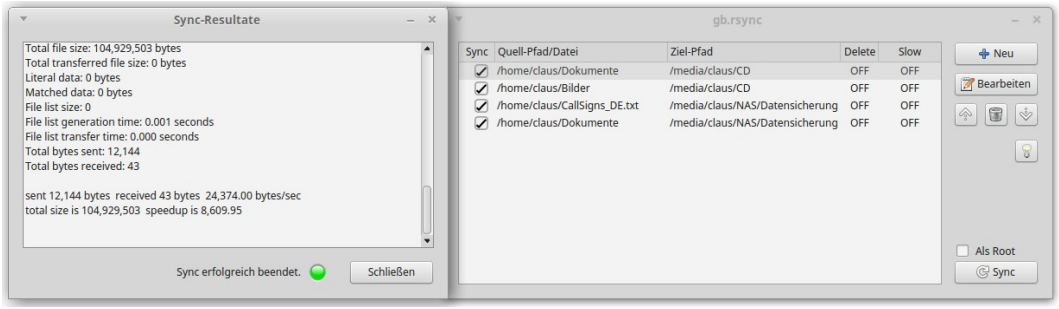

Abbildung 11.11.4.1: Gambas-Frontend 'gb.rsync'

Beispiel für den Start eines Shell-Prozesses mit Root-Rechten:

```
SCommand = "LC_ALL=en_GB.utf8 sudo -S rsync -ab --progress --delete /home/user/Documents /media/user/STICK"

' Start rsync process with shell command
hShellProcess = Shell sCommand For Read Write As "rsyncShellProcess"
```

Die Übergabe und Überprüfung des Root-Passwortes erfolgt in der Error-Ereignis-Routine des Shell-Prozesses:

```
Public Sub rsyncShellProcess_Error(sErr As String)

  Dim PW As String

  If sErr Like "*password*" Then
     PW = TextBoxPW.Text
     Print #hShellProcess, PW & gb.LF
     PW = ""
  Else If sErr Like "*try again*" Then
     If hShellProcess
        hShellProcess.Close()
        hShellProcess.Kill()
     Endif
     Message.Error(("Wrong sudo password!") & GB.LF & ("Please try again."), "OK")
   Else
     ErrorList &= sErr & gb.Lf
  Endif
End
```

Da es bei parallel ablaufenden 'rsync'-Hintergrundprozessen zu unberechenbaren Abläufen kommen kann, sorgt das Programm für sequenzielle Shell-Aufrufe. Nur so ist gewährleistet, dass der vollständige Abschluss *aller* Synchronisierungsaufgaben zuverlässig erkannt wird und auftretende Fehlermeldungen der Reihe nach erfasst werden.

Als Besonderheit ist die Anwendung des ColumnView-Steuerelements zu erwähnen, dem (Pseudo-) Checkboxen in Form von Grafiken hinzu gefügt wurden. Eine Beschreibung der Eigenschaften, Methoden und Ereignisse des Steuerelements ColumnView finden Sie im Kapitel → 17.5 ColumnView.

Hinweise für die Nutzung des Programms:

- Das Programm 'gb.rsync' prüft, ob das erforderliche Konsolen-Programm 'rsync' auf Ihrem Rechner installiert ist. Falls nicht, erfolgt ein entsprechender Hinweis und eine Anweisung, wie man es installiert.
- Eingehängte Medien wie zum Beispiel USB-Sticks finden Sie normalerweise im Verzeichnis '/medien/user'.
- Falls das Programm Fehler der Art "Operation not permitted" während einer Synchronisation meldet, so führen Sie das gleiche Backup noch einmal unter Root-Rechten → CheckBox 'Als Root' durch.
- Die Funktion der Option '--delete' ist bereits in der oben beschriebenen Variante 3 erläutert, allerdings wird im Programm 'gb.rsync' im Ziel-Verzeichnis ein Backup-Verzeichnis mit dem Namen '/Backup' angelegt.

- Die Option '--slow' bewirkt ein Komprimierung der übertragenen Daten beim Transfer über langsame Verbindungen.

- Beachten Sie: Die Nutzung des Programms gb.rsync erfolgt auf eigene Gefahr!

11.11.5 Variante 5

Wenn Ihnen von einem Anbieter Cloud-Speicher in hinreichender Größe zur Verfügung gestellt wird, dann sollten Sie erwägen, ihre Gambas-Projekte (auch) dort verschlüsselt zu speichern. Das hätte den Vorteil, dass Ihre Daten – im Gegensatz zu den anderen Varianten – an einem *unabhängigen* Ort gespeichert sind. Bisher wurde diese Variante vom Autor noch nicht umgesetzt – wird aber erwogen.