

## 11.10.2 Installationspaket auf dem Zielrechner installieren

### 11.10.2.1 Informationen zum Zielrechner

Der Zielrechner wurde durch das Klonen einer virtuellen Mint-Master-Maschine unter VirtualBox auf dem Entwicklungsrechner des Autors realisiert. Dadurch verfügte der Zielrechner für alle Installations-Experimente stets über ein frisch installiertes System Mint 17.1 (Rebecca).

### 11.10.2.2 Notwendige Arbeiten auf dem Ziel-Rechner

Ein generiertes Installationspaket enthält nur Ihr Gambas-Projekt. Zusätzlich listet es alle Abhängigkeiten auf, die notwendigerweise vom Paket-Manager auf dem Ziel-Rechner (rekursiv) installiert werden müssen, bevor Ihr Installationspaket installiert werden kann.

Hier ein Ausschnitt aus der Datei `/DEBIAN/control` in einem Installationspaket mit der Liste der Abhängigkeiten:

```
Depends:
gambas3-runtime (>= 3.6),           ' Gambas-Abhängigkeiten
gambas3-runtime (<< 3.99.0),
gambas3-gb-args (>= 3.6),
gambas3-gb-args (<< 3.99.0),
gambas3-gb-image (>= 3.6), gambas3-gb-image (<< 3.99.0),
ambas3-gb-qt4 (>= 3.6), gambas3-gb-qt4 (<< 3.99.0),
gambas3-gb-form (>= 3.6), gambas3-gb-form (<< 3.99.0),
gambas3-gb-settings (>= 3.6),
gambas3-gb-settings (<< 3.99.0),
gambas3-gb-qt4-ext (>= 3.6),
gambas3-gb-qt4-ext (<< 3.99.0),
bluefish (>= 2.2.5)                ' Extra-Abhängigkeit
```

Ein Vorzug von Installationspaketen liegt darin, dass Sie die Abhängigkeiten nicht selbst herausuchen und auflösen müssen. Diese Abhängigkeiten werden *automatisch* installiert, wenn sie über den Paket-Manager des Systems verfügbar sind. Der Paketmanager löst auf was fehlt und genau das holt er aus den ihm bekannten Repositories. Um ein Gambas-Paket auflösen zu können und sich fehlende Pakete automatisch herunterladen zu können ist nur erforderlich, dass eine *passende* Gambas-Paketquelle angegeben wird und der Ziel-Rechner online ist.

Ein erster Installationsversuch eines Gambas-Installationspakets mit dem gdebi-Paket-Installationsprogramm endete mit dieser Fehlermeldung:

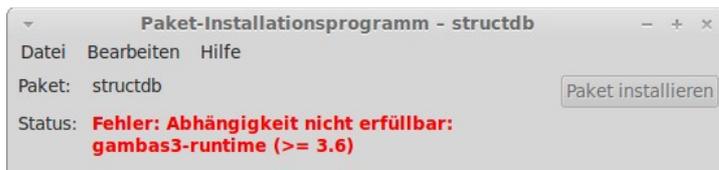


Abbildung 11.10.2.2.1: Fehlermeldung

Ein Blick in die Anwendungsverwaltung zeigte, dass

- Gambas-Pakete in MINT 17.1 in den Paket-Quellen vorhanden sind,
- die Gambas-Pakete nur in einer Version  $< 3.6$  vorliegen ( $\rightarrow$  Version: 3.1.1-2.2ubuntu4)

Die einfachste Art – unter Ubuntu und Mint – die passenden Gambas-Pakete in der stabilen Version auf dem Ziel-Rechner hinzuzufügen besteht darin, das vom 'Gambas Ubuntu Maintainers'-Team gepflegte PPA ( $\rightarrow$  Kapitel 2.3 Installation aus alternativen Quellen – PPA) zu verwenden:

```
mint@linux ~ $ sudo add-apt-repository ppa:gambas-team/gambas3
```

Anschließend müssen Sie die Paket-Quellen unbedingt aktualisieren:

```
mint@linux ~ $ sudo apt-get update
```

## Kapitel 11.10.2 - Installationspaket auf dem Zielrechner installieren

Hier sehen Sie das Ergebnis: Die Gambas-Paket-Quelle ist im Paket-Manager von MINT eingetragen:

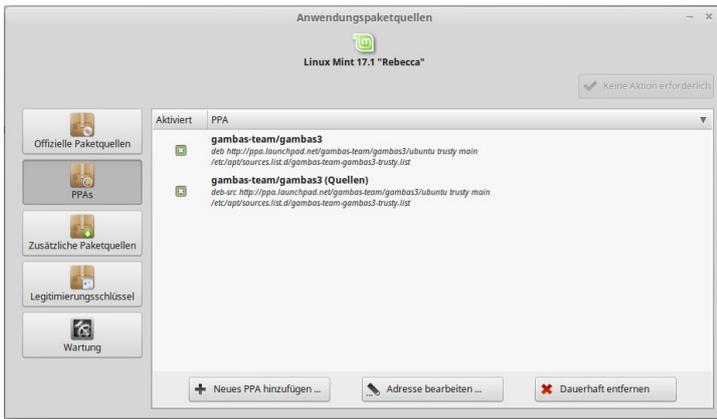


Abbildung 11.10.2.2.2: PPA im Paketmanager

### 11.10.2.3 Installationspaket auf dem Zielrechner installieren

Kopieren Sie den Ordner *IP\_StructDB* mit dem erzeugten Installationspaket vom Entwicklungsrechner in das Home-Verzeichnis des Benutzers der Datenbank-Anwendung *StructDB* auf dem Ziel-Rechner:



Abbildung 11.10.2.3.1: Inhalt des Installationspakets

In MINT 17 ist mit *gdebi* ein grafisches Installationsprogramm für Einzelpakete installiert, das desktop-unabhängig eingesetzt werden kann. Das Programm hat den großen Vorteil – im Gegensatz zum Programm *dpkg* – dass es benötigte Abhängigkeiten aus den Paket-Quellen nachlädt und installiert.

Starten Sie das *gdebi*-Paket-Installationsprogramm mit einem Doppelklick auf die *deb*-Datei:

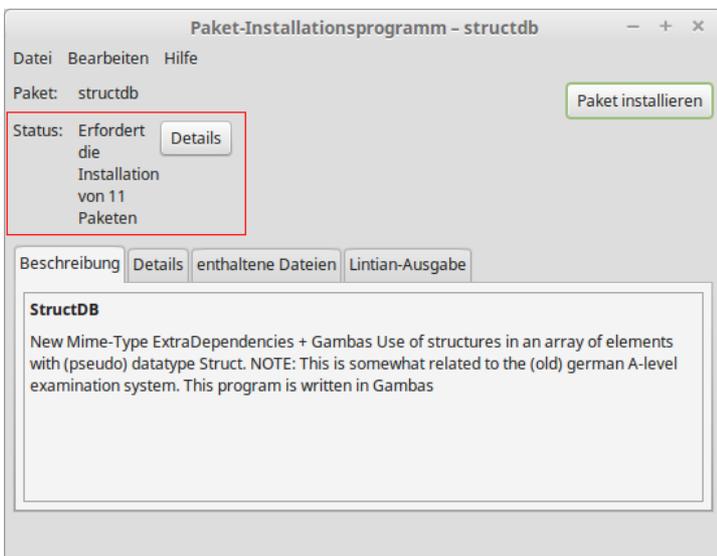


Abbildung 11.10.2.3.2: GUI Paket-Installationsprogramm

Sehen Sie sich unter Details (rot markiert) die 11 Pakete (Abhängigkeiten) an, die vor der Installation des Programms *StructDB* installiert werden müssen.

Starten Sie die Installation mit einem Klick auf *Paket installieren* mit den eingeforderten Root-Rechten.

Zuerst werden zusätzliche Pakete heruntergeladen, danach die Abhängigkeiten installiert und abschließend wird das Programm *StructDB* installiert:

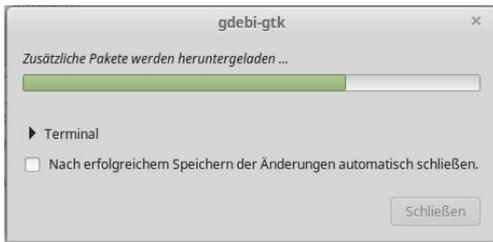


Abbildung 11.10.2.3.3: 1. Schritt: Zusätzliche Pakete herunterladen



Abbildung 11.10.2.3.4: 2. Schritt: Abhängigkeiten installieren

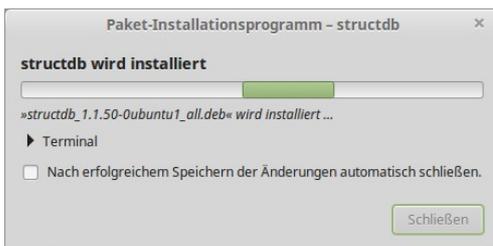


Abbildung 11.10.2.3.5: 3. Schritt: Das Programm *StructDB* wird installiert

#### 11.10.2.4 Erfolgskontrolle

Öffnen Sie das Menü, so sehen Sie unter dem Menü-Eintrag "Büro" (office) das Programm *StructDB*, das Sie nach einem Doppelklick starten können:

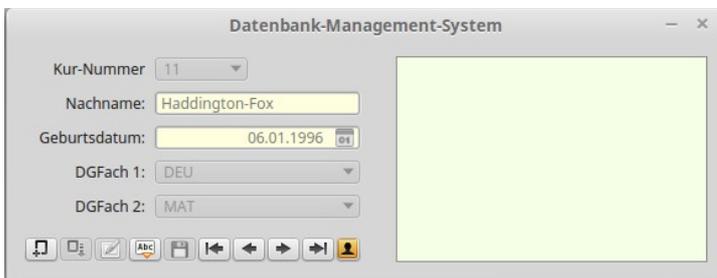


Abbildung 11.10.2.4.1: GUI Programm *StructDB*

- Eine Programm-Hilfe rufen Sie mit der Funktionstaste F1 auf. Mit der Escape-Taste schließen Sie das Hilfenfenster.
- Mit dem letzten Button nach der Navigation können Sie Informationen zu den Programmentwicklern aufrufen.
- Sie können sofort damit beginnen neue Datensätze einzugeben und in einem Struct-Array abzuspeichern sowie dieses Struct-Array mit allen Datensätzen (serialisiert) in eine Datei mit der Extension *.sdb* in einen Ordner Ihrer Wahl zu exportieren. Alternativ können Sie die mit installierte Datenbank-Datei *example.sdb* im Ordner *~/StructDB* importieren:



Abbildung 11.10.2.4.2: Daten-Import

Sie können das Programm auch in der Konsole aufrufen; im zweiten Fall mit der (Muster-)Datenbank-Datei *example.sdb* im Ordner *~/StructDB* als Argument:

```
mint@linux ~ $ StructDB
mint@linux ~ $ StructDB -- /home/mint/StructDB/example.sdb
```

Wechseln Sie in den beim ersten Programmstart angelegten Ordner *StructDB*, so sehen Sie die dort hin kopierte Datei *example.sdb* mit eigenem Datei-Icon:



Ein Doppelklick auf das Datei-Symbol startet das Programm *StructDB* mit der Datenbank-Datei *example.sdb* als Argument.

### 11.10.2.5 Ergänzungen und Alternativen

#### Lokale Paket-Quellen

Im Absatz 2.6.2.2 findet man die Forderung, dass man online sein muss, um fehlende Pakete nachzuladen. Ist das für den Ziel-Rechner nicht zu realisieren, dann nutzen Sie doch eine lokale Paketquelle. Das sind Paketquellen auf Datenträgern (CD/DVD, USB-Stick, externer Festplatte) oder im lokalen Netz (LAN). Unter dem Link [http://wiki.ubuntuusers.de/Lokale\\_Paketquellen](http://wiki.ubuntuusers.de/Lokale_Paketquellen) finden Sie Anregungen, wie Sie eine lokale Paketquelle anlegen. Geben Sie dann im Paket-Manager Ihre lokale Paketquelle an. Normalerweise sind DVD/CD bereits als Default-Paketquelle im Paket-Manager registriert, so dass hier keine gesonderte Aktion nötig ist.

#### Start der englischen Programm-Version

Mit diesem Aufruf gelingt es Ihnen das Programm *temporär* in der englischen Version zu starten:

```
mint@linux ~ $ LC_ALL=en_GB.utf8 StructDB
```

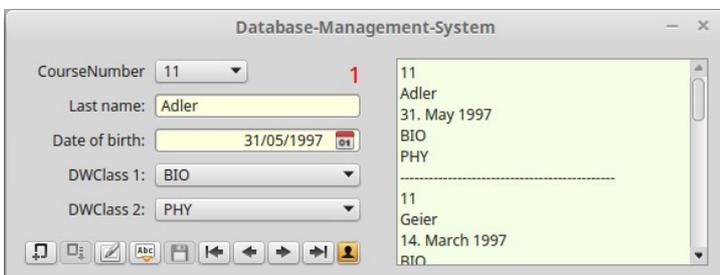


Abbildung 11.10.2.5.1: Programm in der englischen Original-Version

#### Alternative Installation Installationspaket

Um das Installationspaket im Ordner *~/StructDB* erfolgreich über die Konsole zu installieren, genügt dieser Vierzeiler:

```

mint@linux ~ $ sudo add-apt-repository ppa:gambas-team/gambas3
mint@linux ~ $ sudo apt-get update
mint@linux ~ $ sudo dpkg --install ~/IP_StructDB/structdb_1.1.51-0ubuntu1_all.deb
mint@linux ~ $ sudo apt-get install --fix-broken

```

Da das Programm *dpkg* die Abhängigkeiten der zu installierenden Pakete *nicht automatisch* auflöst, müssen Sie das 4. Kommando mit der Option *--fix-broken* oder *-f* aufrufen – ohne Angabe von Paketnamen – damit fehlende Abhängigkeiten nachinstalliert werden.

### Programm-Start-Variante

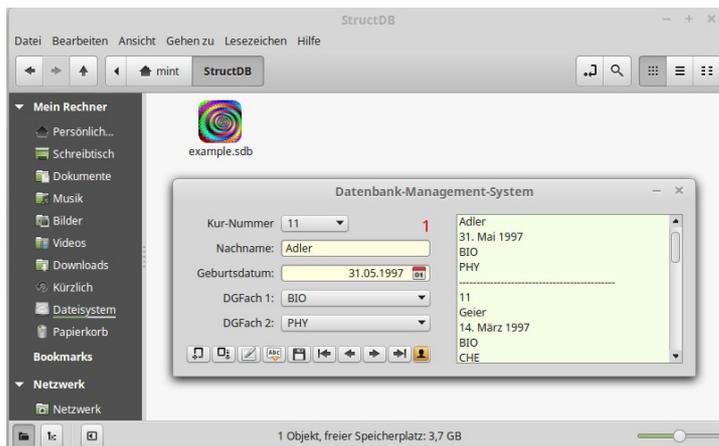


Abbildung 11.10.2.5.2: Programmstart mit Doppelklick auf das Datei-Icon

### Gambas auf dem Entwicklungsrechner

Der Paketmanager, der das *.deb*-Paket auf dem Ziel-Rechner installiert prüft, ob alle Abhängigkeiten erfüllt sind. Der Manager kennt nur die Pakete, die er selbst installiert hat! Auch wenn Sie Gambas → Kapitel 2.1 bis 2.5 zum Beispiel auf dem Ziel-Rechner *selbst kompiliert und mit Erfolg installiert* haben – aber damit an der Paketverwaltung vorbei – gelingt es Ihnen nicht, ein Gambas-Installationspaket erfolgreich zu installieren! Beim Erzeugen des Installationspakets auf dem Entwicklungsrechner werden nur die Paketnamen als Abhängigkeiten eingetragen, zum Beispiel "gambas3-gb-settings", falls das Projekt die Komponente *gb.settings* benutzt. Da diese Namen standardisiert sind, können die Eintragungen unabhängig von den lokal installierten Paketen auf dem Entwicklungsrechner vorgenommen werden. Daher gibt es keine Forderungen, *wie* Sie Gambas auf dem Entwicklungsrechner(!) installiert haben – wohl aber die Empfehlung, vor der Installation stets eine De-Installation einer bestehenden Gambas-Installation vorzunehmen.