

### 11.10.1 Installationspaket erzeugen

- Überprüfen Sie noch einmal, welche Dateien Sie Ihrem Programm mitgeben wollen und ob diese Dateien in den richtigen Ordnern liegen.
- Kontrollieren Sie abschließend, ob alle Übersetzungen von der englischen in die deutsche Sprache *vollständig* vorgenommen worden sind.
- Stellen Sie sicher, dass Sie das zu exportierende Programm ausführlich getestet haben und das Projekt aufgeräumt wurde, was dann das Erzeugen einer ausführbaren Datei einschließt.

Den Assistenten zum Erzeugen eines Installationspakets (Packer-Assistent) in 10 Schritten finden Sie unter Projekt> Erstellen> Installationspaket *oder* Sie rufen ihn direkt in der IDE mit **CTRL+ALT+I** auf. Füllen Sie alle erforderlichen Angaben sorgfältig prüfend aus!

#### 11.10.1.1 1. Schritt – Paketinformation

The screenshot shows a dialog box titled "Installationspaket erstellen" with the following content:

- 1. Paketinformation**
- Paketname:** structdb-1.1.13
- Händlernamen vor den Paketnamen stellen
- Package version: 1
- Bearbeiterinformationen:**
  - Name: Hans Lehmann
  - E-mail: hans@linux
  - Anbietername: Online-Gambas-Buch
  - URL: www.gambas-buch.de
- Beschreibung:** Use of structures in an array of elements with (pseudo) datatype Struct. NOTE: This is somewhat related to the (old) german A-level examination system.
- Lizenz:** General Public License
- Buttons: << Vorheriger, Weiter >>, Abbrechen

Abbildung 11.10.1.1.1: Schritt 1 – Paketinformation

Die Angaben aus den Einstellungen im Menü-Punkt 'Tools' in der Gambas-IDE können Sie hier ändern oder ergänzen. Die Beschreibung 'Händlername' und 'Anbietername' werden hier synonym verwendet.

Wenn Sie keinen Anbieternamen eintragen, dann wird automatisch ein Standardtext eingefügt. Setzen Sie statt dessen dort die Zeichenkette "- - -" ein. Der Händlername wird jedoch standardmäßig nicht vor den Paketnamen gestellt.

#### 11.10.1.2 2. Schritt – Änderungsprotokoll

The screenshot shows a dialog box titled "Installationspaket erstellen" with the following content:

- 2. Änderungsprotokoll**
- Bitte tragen Sie die Änderungen an Ihrem Projekt ein.
- Thu Jan 01 2015 Hans Lehmann <hans@linux> 1.1.26
- Erstes Release - first release

Abbildung 11.10.1.2.1: Schritt 2 - Änderungsprotokoll

In diesem Schritt sollten Sie für den Programm-Nutzer den Projekt-Fortschritt und die Änderungen zu älteren Versionen dokumentieren.

Achten Sie darauf, diese Eintragungen beim Projekt *StructDB* in englischer und deutscher Sprache vorzunehmen, denn diese Eintragungen werden nicht zur Übersetzung im Übersetzungsassistenten angeboten!

### 11.10.1.3 3. Schritt – Ziel-Distribution

Wählen Sie die Ziel-Distributionen aus, die Sie für die Weitergabe auf dem Ziel-Rechner benötigen:

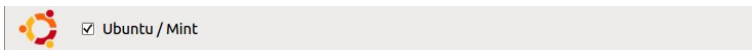


Abbildung 11.10.1.3.1: Schritt 3 – Wahl der Zieldistribution Mint

### 11.10.1.4 4. Schritt – Paket-Gruppe

Hier passt die Auswahl 'database' aus der vorgegebenen Liste für das Datenbank-Projekt gut.

### 11.10.1.5 5. Schritt – Menü-Eintrag

Auch in diesem Schritt wird Ihnen eine Liste angeboten, aus der Sie im oberen Teil sofort auswählen können. Für die Festlegung der Kategorie stehen Ihnen die Einträge aus der Combo-Box zur Verfügung:

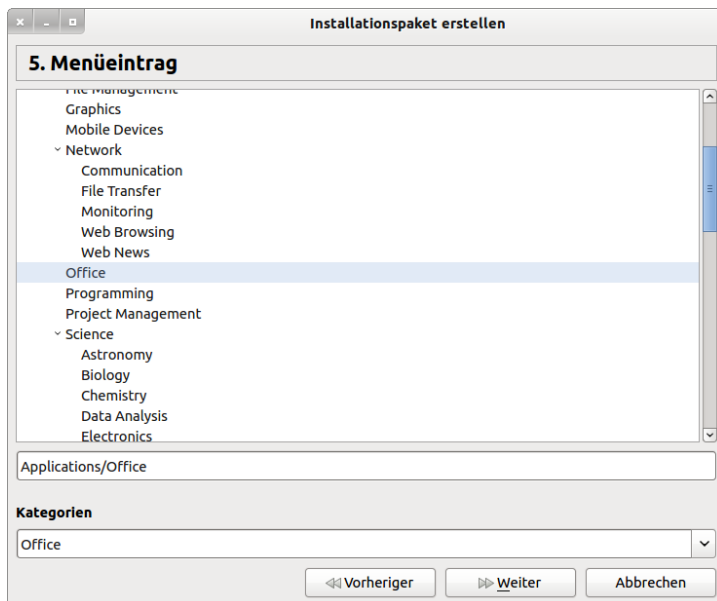


Abbildung 11.10.1.5.1: Schritt 5 – Menü-Eintrag

Offensichtlich werden Menü-Einträge auf einem Ziel-Rechner mit MINT 17.1 ignoriert, die nicht als Haupteinträge im MINT-Menü stehen. Das Programm StructDB erscheint dann im MINT-Menü unter dem Menü-Eintrag 'Sonstige'. Bei der Wahl von 'Applications/Office' finden Sie das Programm jedoch im richtigen Menü-Eintrag unter 'Büro'.

### 11.10.1.6 6. Schritt – Desktop-Konfigurationsdatei

#### 1. Variante

Das Programm *StructDB* verfügt über ein Speicher-Management, mit dem Sie die Datenbank-Struktur (Daten-Typ Struct → Kapitel 7.2 Strukturen - Struct) serialisiert in einer Datei abspeichern und wieder auslesen können. So bestimmen Sie unter Ubuntu den MIME-Typ einer Datei – wobei der Begriff *Content-Typ* geeigneter wäre:

```
hans@linux:~/11-Buchprojekt/$ xdg-mime query filetype ~/StructDB/example.sdb
application/octet-stream
```

Der Content-Typ für diese \*.sdb-Dateien ist also *application/octet-stream* und ist unabhängig von der selbst vergebenen Datei-Extension .sdb für die Datenbank-Dateien! Die Angabe des zutreffenden Mime-Typs *application/octet-stream* für die Datenbank-Datei im Packer-Assistenten erzeugt bei der Installation des Installationspakets auf dem Ziel-Rechner automatisch in der systemweit geltenden Datei */usr/share/applications/mimeinfo.cache* des Ziel-Rechners den Eintrag:

```
application/octet-stream=StructDB.desktop
```

Das sichert (programm-intern über die Args-Komponente), dass sich nach einem Doppel-Klick auf eine \*.sdb-Datei das Datenbank-Programm StructDB öffnet, die Datensätze aus der ausgewählten \*.sdb-Datei eingelesen und angezeigt werden. So lange nur ein einziger Mime-Eintrag für *application/octet-stream* existiert, ist das Datenbank-Programm *StructDB* das Standard-Programm; sonst müssen Sie mit der rechten Maustaste das Datenbank-Programm zum Öffnen auswählen. Unter dem Link: <http://wiki.ubuntuusers.de/MIME-Typ> finden Sie eine gute Beschreibung zum Thema MIME-Typ.

Da für eine Datenbank-Datei des Programms StructDB der MIME-Typ *application/octet-stream* ist, wird dieser Typ hier eingetragen:



Abbildung 11.10.1.6.1: Schritt 6 – Festlegung MIME-Typ – Variante 1

## 2. Variante

In dieser Variante wird durch Einträge sowohl im Schritt 6 als auch im Schritt 8 ein neuer MIME-Typ auf dem Zielrechner erzeugt und registriert:



Abbildung 11.10.1.6.2: Schritt 6 – Festlegung MIME-Typ – Variante 2

Der Eintrag *x-structdb* muss bereits hier eingetragen werden und steht in der XML-MIME-Definitionsdatei → Schritt 8. Dabei steht das *x-* für einen eigenen MIME-Typ und *structdb* für das Programm.

Die .desktop-Dateien assoziieren bestimmte Datei-Content-Typen (MIME-Typen) mit Programmen und legen so fest, mit welchen Programmen Dateien eines bestimmten Typs bearbeitet werden. Das ist ein Ausschnitt aus der Datei *structdb.desktop* nach Variante 2:

```
[Desktop Entry]
Version=1.0
Name=StructDB                                ' Programm-Name
...
Exec=/usr/bin/StructDB                       ' Programm-Pfad
Icon=/usr/share/pixmaps/structdb.png        ' Pfad zum Programm-Icon
Terminal=false
Type=Application
StartupNotify=true
Categories=Office;                          ' Menü-Eintrag
MimeType=application/x-structdb            ' Programm-spezifischer MIME-Typ
```

### 11.10.1.7 7. Schritt – Extra-Abhängigkeiten

Für das Projekt *StructDB* existiert eine (fiktive) Extra-Abhängigkeit. Es wird angenommen, dass das Programm *StructDB* den HTML-Editor *BlueFish* aufruft, um eine bestimmte HTML-Datei zu bearbeiten. Das Programm würde nicht funktionieren, wenn auf dem Zielrechner dieser Editor nicht installiert ist. Sie müssen daher sicherstellen, dass der HTML-Editor *BlueFish* während der Installation des Installationspakets installiert wird, wenn er noch nicht installiert wurde.

Diese Information müssen Sie bei den Extra-Abhängigkeiten dem Installationspaket einfügen. Genau dafür ist dieser 7. Schritt gedacht. Wenn es erforderlich ist, so können Sie auch die benötigten Versionsnummern (min/max) eintragen.

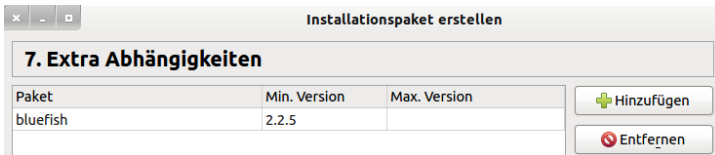


Abbildung 11.10.1.7.1: Schritt 7 – Festlegung von Extra-Abhängigkeiten

Diese Szenarien sind möglich:

- Der Editor *BlueFish* ist auf dem Ziel-Rechner in der angegebenen (Minimal-)Version installiert. Dann ist diese Extra-Abhängigkeit aufgelöst.
- Auf dem Ziel-Rechner ist der Editor *BlueFish* nicht installiert. Dann wird der Editor automatisch installiert, wenn eine passende Paket-Quelle auf dem Ziel-Rechner existiert. Das war bei allen Erprobungen unter MINT 17.1 stets der Fall.
- Wenn der Debian-Installer die Extra-Abhängigkeit nicht auflösen kann, dann werden Sie darüber informiert und eine Lösung angeboten, die bisher stets zum Ziel führte.

### 11.10.1.8 8. Schritt – Zusätzliche Dateien

In diesem Schritt haben Sie die Möglichkeit, dem Programm zum Beispiel Dateien mitzugeben, die jedoch von allen System-Benutzern nur gelesen werden können. Sie können die Dateien nur einzeln mit 'Hinzufügen' auswählen (ein Multi-Select steht nicht zur Verfügung), die sich im *logischen* "Projekt"-Verzeichnis befinden. Als Ziel-Ordner geben Sie einen Ordner an, in dem normale System-Benutzer nur lesen können und die Dateien so vor schreibendem Zugriff geschützt sind.



Abbildung 11.10.1.8.1: Schritt 8 – Zusätzliche Dateien

Wenn Sie sich entschieden hatten, einen eigenen MIME-Typ auf dem Ziel-Rechner erzeugen und registrieren zu lassen → Kapitel 2.6.0.11 (Variante 2) und im 6. Schritt den MIME-Typ auch eingetragen hatten, dann müssen Sie hier im 8. Schritt die notwendigen Informationen angeben, wie der neue MIME-Typ für Datenbank-Dateien, die lokal mit dem Programm StructDB angelegt werden, definiert wird. Dateien mit der Endung \*.sdb sollen dem neuen MIME-Typ zugeordnet und mit einem Datei-Symbol sowie der Zuordnung zur Standard-Anwendung StructDB versehen werden, so dass sich auch nach einem Doppelklick auf das Datei-Symbol das Programm öffnet, die Datensätze aus der Datei eingelesen und angezeigt werden. Zunächst wird dazu eine XML-Datei mit der Beschreibung des MIME-Typs (mime-type), der Datei-Extension (glob pattern) und des Datei-Icons (icon name) benötigt.

Der Inhalt der (erprobten) Datei *structdb.xml* folgt den Vorgaben auf [www.freedesktop.org](http://www.freedesktop.org):

```
<?xml version="1.0" encoding="UTF-8"?>
<mime-info xmlns="http://www.freedesktop.org/standards/shared-mime-info">
  <mime-type type="application/x-structdb">
    <comment xml:lang="de">StructDB Projekt</comment>
    <comment xml:lang="en">StructDB Project</comment>
    <comment xml:lang="es">Proyecto StructDB</comment>
    <comment xml:lang="fr">StructDB projet</comment>
    <comment xml:lang="nl">StructDB Project</comment>
    <glob pattern="*.sdb" />
    <icon name="structdb_dfi" />
  </mime-type>
</mime-info>
```

Ein herzlicher Dank geht an Willy Raets (Niederlande – <http://howtogambas.org>), der aus einem seiner Projekte eine Muster-XML-Datei zur Ansicht zur Verfügung stellte.

Die zu kopierende XML-Datei und die Icon-Datei müssen Sie im 8. Schritt des Packer-Assistenten auswählen (...) und eintragen:

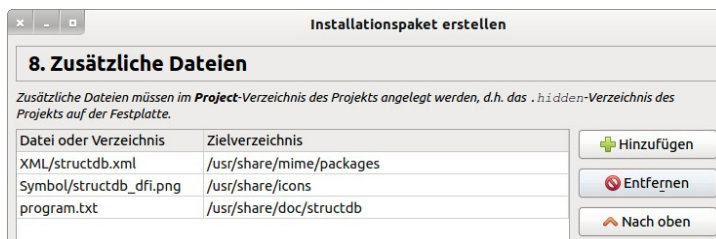


Abbildung 11.10.1.8.2: Schritt 8 – Packer-Assistent

Das sind die Standard-Verzeichnisse für die XML-Datei und für die Icon-Datei auf dem Ziel-Rechner:

```
/usr/share/mime/packages/  
/usr/share/icons/
```

Sowohl die XML-Datei als auch die Symbol-Datei müssen im physischen `.hidden`-Ordner im Projekt-Ordner liegen, was dem logischen Ordner 'Projekt' in der IDE entspricht. Sie können beide Dateien auch in Unter-Ordner wie `XML` und `Symbol` kopieren → Abbildung 2.6.1.8.2.

#### 11.10.1.9 9. Schritt – Zielverzeichnis

Wählen Sie in diesem Schritt einen Ordner auf dem Entwicklungsrechner aus, in dem der Packer-Assistent alle Dateien des Installationspakets abspeichert und sichert oder legen Sie diesen Ordner über das aufrufbare Kontext-Menü an. Die Option (rot markiert) ist nur dann zu aktivieren, wenn Sie im 3. Schritt mehrere Ziel-Distributionen angegeben haben.

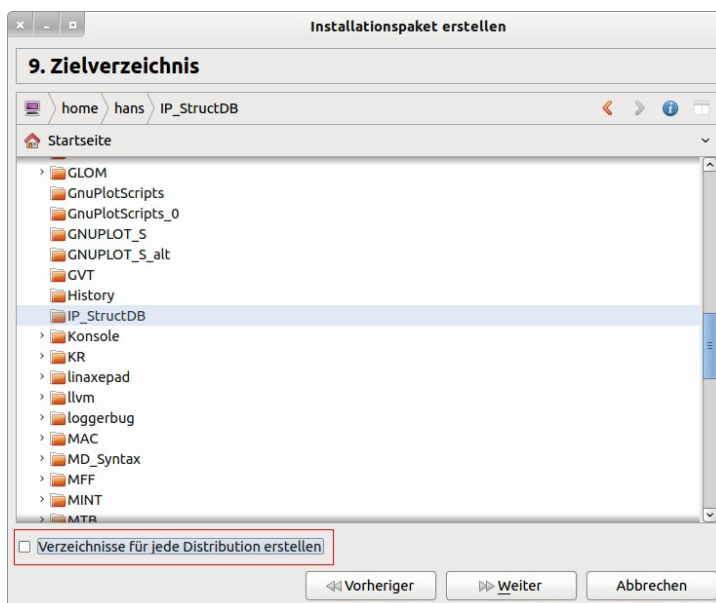


Abbildung 11.10.1.9.1: Schritt 9 – Zielverzeichnis festlegen

#### 11.10.1.10 10. Schritt – Paket erstellen

Jetzt wird es ernst. Mit OK schieben Sie das Erzeugen des Installationspakets im Schritt 10 an. Dieser Vorgang dauert ein wenig. Wenn das Erzeugen aller Dateien im Installationspaket *erfolgreich* abgeschlossen wurde, dann werden Sie darüber unterrichtet:

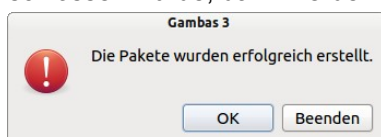


Abbildung 11.10.1.10.1: Schritt 10 – Das Installationspaket wurde erfolgreich geschnürt

Mit OK kehren Sie zum Schritt 10 zurück. Nur nicht nervös werden und voller Begeisterung auf den OK-Button des Packer-Assistenten klicken, denn dann würde ein weiterer Durchlauf zum Erzeugen eines Installationspakets starten. Klicken Sie auf 'Abbrechen' oder schließen Sie das Fenster des Packer-Assistenten!

Im Anschluss finden Sie in Ihrem – im Dialog benannten (Export-)Ordner *IP\_StructDB* im Home-Verzeichnis – diese Dateien:

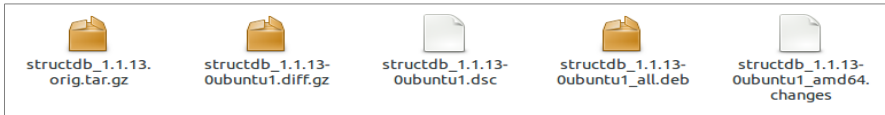


Abbildung 11.10.1.10.2: Inhalt Installationspaket

Unter Ubuntu oder MINT können Sie sich den strukturierten Inhalt der deb-Datei ansehen, indem Sie nach einem Klick mit der rechten Maustaste auf die deb-Datei den Eintrag "Mit Archivverwaltung öffnen" oder unter MINT den Eintrag "Öffnen mit Archiv-Manager" im sich öffnenden Kontext-Menü auswählen. Sehen Sie sich in aller Ruhe die Struktur des Installationspakets an. Lesen Sie sich den Inhalt der zahlreichen Dateien durch!

Beachten Sie:

- Es darf kein Ordner im Projektordner den *gleichen* Namen tragen wie der Projektordner selbst, weil dann das Generieren eines Installationspakets misslingt.
- Wenn Sie einer *eigenen Komponente(!)* ein Icon mitgeben wollen, das in der Entwicklungsumgebung angezeigt werden soll, dann müssen Sie das Icon im Projektordner im png-Format im versteckten Ordner *.hidden/control* speichern.