

19.1.1 Settings – Konfigurationsdateien

Gambas bietet die Möglichkeit, die Konfiguration eines Programms in sogenannten Initialisierungs- oder Konfigurationsdateien *.conf zu speichern und den Inhalt dieser Datei zum Programmstart einzulesen und auszuwerten.

Der Standard für Konfigurationsdateien wird von *freedesktop.org* so definiert: Eine Konfigurationsdatei `<Anwendung-Name>.<Extension>` wird im Verzeichnis `~/config/` gespeichert. Die Extension der Konfigurationsdatei ist frei wählbar. Empfohlen wird 'conf'. Gambas3 hält sich mit der Klasse Settings sowie mit den eigenen Konfigurationsdateien an diesen Standard.

Die Konfigurationsdatei ist in der Standard*einstellung* der eingesetzten Settings-Komponente eine einfache Textdatei *.conf und kann zum Beispiel zum Einfügen von Kommentaren mit einem Texteditor nachträglich bearbeitet werden. Das ist möglich, weil der Pfad einer Konfigurationsdatei ein Lesen und Schreiben vom angemeldeten Benutzer zulässt:

```
User.Home &/ ".config/gambas3" &/ Application.Name & ".conf"
```

Folgende Hinweise für den Inhalt einer Konfigurationsdatei sind zu beachten:

- Eine leere Zeile wird ignoriert.
- Ein Kommentar beginnt mit dem Zeichen # oder einem Semikolon und wird ignoriert.
- Der Bezeichner für eine Sektion (Slot) – der nur einmal in der Konfigurationsdatei vorkommen darf – steht in eckigen Klammern.
- Unter dem Sektions-Bezeichner [Bezeichner] stehen in weiteren Zeilen Schlüssel-Wert-Paare. Das Trennzeichen zwischen dem Schlüssel und dem Wert ist ein Gleichheitszeichen. Der Wert nach dem Gleichheitszeichen ist vom Typ *String* und wird deshalb von "..." eingeschlossen.
- Sektionen werden mit einer Leerzeile beendet; sie trennt Sektionen voneinander.

```
[Expander]
ExpanderHiddenStatus=False

[V24Konfiguration]
Port-Name="ttyUSB0"
Geschwindigkeit="9600"
Parität="NONE"
...
Datenflusskontrolle="NONE"
<LEERZEILE>
```

- Auf die Konfigurationsdaten in der Konfigurationsdatei wird zur Programmlaufzeit über die Kombination Sektionsname/Schlüssel oder beim *Programmstart* bei nicht vorhandenem Schlüssel-Wert-Paar auf den Vorgabewert zugegriffen, wie der folgende Ausschnitt aus dem Quelltext für das Projekt *Temperaturmessung* im Ereignis *Form_Open()* zeigt:

```
Syntax: Wert = Setting-Objektname["Sektionsbezeichner/Schlüsselname", optional "Vorgabewert"]

expRX_TX.Hidden = v24Settings["Expander/ExpanderHiddenStatus", "TRUE"]      ---> False
cmbSpeed.Text    = v24Settings["V24Konfiguration/Geschwindigkeit", "4800"]   ---> 9600
cmbParity.Text   = v24Settings["V24Konfiguration/Parität", "NONE"]           ---> NONE
```

In den folgenden Abschnitten werden Projekte vorgestellt, bei denen

- der Standardpfad für eine Konfigurationsdatei genutzt wird oder
- die Konfigurationsdatei in einem selbst angelegten Verzeichnis gespeichert wird oder
- mit Profilen gearbeitet wird, wie sie zum Beispiel für ein FTP-Programm mit unterschiedlichen FTP-Konten notwendig sind.

Für alle vorgestellten Projekte sind die folgenden Schwerpunkte gesetzt:

- Anlegen der Konfigurationsdatei,
- Bereitstellung von Vorgabe-Werte beim ersten Programmstart, da in diesem Fall noch keine Konfigurationsdatei existiert,
- Anlegen von Sektionen und setzen der dazu gehörigen Schlüssel-Wert-Paare,

- Auslesen von Schlüssel-Wert-Paaren und zuweisen der Werte (Komponenten-Eigenschaften und Variable),
- Ändern von Schlüssel-Wert-Paaren innerhalb einer Sektion und
- Löschen von Sektionen mit allen Schlüssel-Wert-Paaren; nur im Projekt mit Profilen.

Zur Verwaltung der Konfiguration der Anwendung müssen Sie die Komponente *gb.settings* in Ihr Projekt einbinden.