

1.2.7 Parallelinstallation mehrerer Gambas-Versionen

In diesem Artikel wird beschrieben, wie Sie mindestens zwei verschiedene Versionen von Gambas parallel auf Ihrem Rechner einsetzen können. Es werden zwei Lösungswege beschrieben, von denen jeder seine Vor- und Nachteile hat.

Als Beispiel dient die nachträgliche Installation der aktuellen SVN-Revision (genannt „trunk“) neben einer bereits existierenden stabilen Version. Diese kann selbstkompiliert oder den Paketquellen Ihrer Distribution entnommen sein. Es wird nur vorausgesetzt, dass sie im Standardverzeichnis `/usr` installiert wurde und weiterhin als die primäre Installation gelten soll.

1.2.7.1 Vorbereitungen

Das Herunterladen, Konfigurieren, Kompilieren und Installieren der aktuellen SVN-Revision erfolgt nahezu analog zu Kapitel 1.2.3 Installation über SVN. Lediglich der Schritt der Konfiguration muss angepasst werden.

Die Quelltexte sollen im Verzeichnis `$HOME/src/gambas3_trunk` abgespeichert werden. Die private Dual-Installation soll in das Verzeichnis `$HOME/bin/gambas3_trunk`. Falls diese Verzeichnisse nicht existieren, werden sie mit

```
$ mkdir -p $HOME/src/gambas3_trunk
$ mkdir -p $HOME/bin/gambas3_trunk
```

angelegt.

Die folgenden Schritte, bis auf die Konfiguration, sollten Ihnen nach der Lektüre von 1.2.3 Installation über SVN vertraut sein:

```
$ cd ~/src
$ svn co svn://svn.code.sf.net/p/gambas/code/gambas/trunk gambas3_trunk
$ ./reconf-all
$ ./configure --prefix=$HOME/bin/gambas3_trunk
$ make -j5
$ make install
```

- Der Parameter `--prefix` des `./configure`-Skripts legt das Basisverzeichnis für die Installation fest. Der Default-Wert ist bekanntlich `"/usr"`. Indem wir ihn auf `$HOME/bin/gambas3_trunk` setzen, werden alle Ressourcen auch relativ zu diesem Verzeichnis installiert.
- Der Parameter des `-j`-Schalters sollte je nach Rechner angepasst werden. Vergleichen Sie mit → <http://stackoverflow.com/questions/3025587/whats-best-value-for-make-j>.
- Auf die `root`-Rechte bzw. den `sudo`-Teil bei `make install` kann verzichtet werden, da das Installationsverzeichnis dem aktuellen Benutzer gehört.

Alternative:

Sollten Sie bereits ein Verzeichnis mit einer konfigurierten und kompilierten Revision haben und scheuen den Aufwand einer Neukonfiguration, so ist es möglich, das Installationsverzeichnis nachträglich über eine Variable des `make`-Programms zu steuern:

```
$ make install DESTDIR=~/.bin/gambas3_trunk
```

Hier wird das Zielverzeichnis (engl. destination directory) direkt an den `make-install`-Prozess übergeben. Der Nachteil ist, dass sich der Installationspfad aus `$DESTDIR/$PREFIX` zusammensetzt. Das bedeutet bei einer Standardkonfiguration, dass Sie ihre Installation in `$HOME/bin/gambas3_trunk/usr` finden werden – die Wahl bleibt Ihnen überlassen.

Vereinfachend wird angenommen, dass Sie den `--prefix`-Parameter zur Konfiguration des

Installationsprozesses gesetzt haben.

Die eigentliche Schwierigkeit ist nicht die Installation verschiedener Gambas-Versionen in verschiedenen Plätzen, sondern das Verwenden einer dieser Versionen bzw. der Wechsel zwischen zwei Versionen. Hierfür existieren mindestens zwei Lösungen, welche nachfolgend erklärt werden.

1.2.7.2 Die offizielle Lösung

Die offiziell unterstützte Verfahrensweise zum Wechsel einer Gambas-Installation (ohne Neuinstallation) ist das Ersetzen von `/usr/bin/gbx3` – dem Gambas-Interpreter – durch einen symbolischen Link (Symlink).

Existiert `/usr/bin/gbx3` und ist ein Symlink, so wird diesem Link gefolgt und über das verlinkte Verzeichnis das neue Gambas-Installationsverzeichnis ermittelt (→ http://sourceforge.net/mailarchive/message.php?msg_id=30551350).

Die Prozedur zur Ausführung des Interpreters Ihrer SVN-Revision ist folgende:

```
[1] $ sudo mv /usr/bin/gbx3 /usr/bin/gbx3_backup
[2] $ sudo ln -s ~/bin/gambas3_trunk/bin/gbx3 /usr/bin/gbx3
[3] $ gambas3
[4] $ sudo mv /usr/bin/gbx3_backup /usr/bin/gbx3
```

- Dabei wird mit Zeile [1] eine Sicherheitskopie Ihres Interpreters angelegt und die Originaldatei gelöscht.
- Der Befehl in der Zeile [2] legt einen Symlink `/usr/bin/gbx3` an, der auf den neuen Interpreter unter `$HOME/bin/gambas3_trunk` zeigt.
- Zeile [3] ist nur exemplarisch für die Benutzung des neuen Interpreters. Es wird hier die IDE gestartet.
- Mit Zeile [4] wird der Symlink wieder vom alten Interpreter überschrieben und Ihre aktive Gambas-Installation ist nun wieder die stabile Version.

Es werden die folgenden zwei Skripte vorgestellt, um zwischen den Gambas-Installationen zu wechseln. Sie werden hier in leicht gekürzter Fassung wiedergegeben. Im Download-Bereich finden Sie die vollständigen Skripte.

```
#!/bin/bash
# Empfohlener Name des Skriptes: set_gambas3.sh

if [ -z $1 ]
then echo "Usage: $0 INSTALL_PATH" >&2
exit 1
fi

if [ \! -L /usr/bin/gbx3 ]
then mv /usr/bin/gbx3 /usr/bin/gbx3_backup
fi
ln -s $1/bin/gbx3 /usr/bin/gbx3
```

Das Skript sollte mit dem Basisverzeichnis Ihrer alternativen Gambas-Installation aufgerufen werden, wie es auch als `--prefix` angegeben wurde. Es führt dann die Schritte [1] und [2] aus der obenstehenden Prozedur aus, ohne dabei eine Sicherheitskopie des alten Interpreters durch einen Symlink zu überschreiben. Dies wäre der Fall, wenn das Skript zweimal hintereinander ausgeführt würde. Der Interpreter würde gelöscht und eine Neuinstallation notwendig. Dies soll vermieden werden.

Nach der Ausführung dieses Skriptes ist die alternative Gambas-Version eingerichtet und kann benutzt werden. Möchten Sie zu seiner stabilen Version zurückkehren, führen Sie das nachfolgende Skript aus:

```
#!/bin/bash
# Empfohlener Name des Skriptes: reset_gambas3.sh
mv /usr/bin/gbx3_backup /usr/bin/gbx3
```

Beide Skripte benötigen root-Privilegien, um ordnungsgemäß arbeiten zu können und müssen damit als

```
$ sudo set_gambas3.sh $HOME/bin/gambas3_trunk
$ sudo reset_gambas3.sh
```

aufgerufen werden.

Beachten Sie die folgenden Nachteile der offiziellen Lösung:

- Sie benötigen root-Rechte zum Wechsel der Gambas-Version.
- Da sich die Gambas-Installation in Ihrem Home-Verzeichnis befindet und es sich bei der Erstellung des Symlinks `/usr/bin/gbx3` um eine globale Änderung handelt, werden andere Benutzer ihres Rechners, die keinen Zugriff auf Ihre lokale Gambas-Installation haben, Gambas nicht mehr ausführen können. Diesen wird dann der „Zugriff verweigert“.
- Es können nicht zwei verschiedene Versionen von Gambas gleichzeitig laufen, da der Symlink `/usr/bin/gbx3` nur einen Wert annehmen kann. Eine Änderung des Symlinks während der Laufzeit von Gambas wird ausdrücklich nicht empfohlen.
- Die Gambas-IDE und möglicherweise andere in Gambas geschriebene Programme nehmen an, dass Gambas und seine Ressourcen unter `/usr` installiert sind. Sie lesen dadurch möglicherweise Dateien unter `/usr`, die mit der aktuell verwendeten Gambas-Version nicht kompatibel sind.

1.2.7.3 Ein zweiter Ansatz

Diese Lösung wurde vom Autor dieses Artikels entwickelt um Nachteile der offiziellen Lösung aufzuheben.

Der Interpreter kann auf einfache Weise so modifiziert werden, dass er den Pfad der Gambas-Installation aus einer Umgebungsvariablen abliest – genauso wie er ihn auch aus dem Symlink ablesen würde. Der Unterschied besteht darin, dass Umgebungsvariablen wesentlich leichter zu setzen sind als Symlinks unter `/usr` – es werden keine root-Rechte benötigt. Weiterhin sind Umgebungsvariablen lokal zum Programm, in dem sie gesetzt werden, zum Beispiel einer Shell und werden von dort weiter an Kindprozesse vererbt.

Ein Benutzer kann ein Programm mit seiner stabilen Version ausführen und gleichzeitig von einer anderen Shell eines mit der aktuellen Revision! Ebenfalls kann zur gleichen Zeit ein anderer Benutzer in seiner Shell-Sitzung eine ganz andere Revision von Gambas benutzen, ohne dass es zu Konflikten kommt. Der Grund liegt darin, dass das Setzen der Umgebungsvariablen eine lokale Änderung ist, die andere Bereiche des Systems nicht beeinflusst.

Um die Änderungen wirksam werden zu lassen, muss der Interpreter erneut kompiliert werden. Im Download-Bereich finden Sie eine Patch-Datei, die Sie entweder über Ihren Webbrowser oder das nachfolgende Kommando in Ihrem SVN-Quelltextordner abspeichern:

```
$ wget http://www.gambas-buch.de/dw/lib/exe/fetch.php?media=k1:k1.2:k1.2.7:gbx-support-alternate-gambas-locations.patch -O ~/src/gambas3_trunk/gbx-support-alternate-gambas-locations.patch
```

Um den Patch einzuspielen und den Interpreter dann erneut zu kompilieren und zu installieren, sind folgende Schritte notwendig:

```
$ cd ~/src/gambas3_trunk
```

```
$ patch -p0 <gbx-support-alternate-gambas-locations.patch
$ cd main
$ make -j5
$ make install
```

Der neue Interpreter wertet die Umgebungsvariable `GAMBAS_ROOT` aus:

- Enthält sie einen Pfad zu einem Verzeichnis, so wird dieses als Basisverzeichnis der Gambas-Installation im Sinne des "--prefix"-Parameters interpretiert. Ist `GAMBAS_ROOT` invalid, führt Gambas die üblichen Schritte aus, um sein Installationsverzeichnis zu bestimmen – die offizielle Lösung ist also auch mit diesem Patch möglich.
- Ist `GAMBAS_ROOT` gesetzt, enthält aber keinen Pfad, wird dies als Irrtum seitens des Benutzers interpretiert. Es erscheint dann eine Warnung im Terminal.
- Um die Gambas-IDE mit der SVN-Installation auszuführen, muss `GAMBAS_ROOT` einerseits und auch die `PATH`-Variable andererseits gesetzt sein, denn nur der neue (gepatchte) Interpreter, der in `$HOME/bin/gambas3_trunk` installiert wurde, beachtet `GAMBAS_ROOT`. Das Kommando `gbx3` im Terminal soll nicht den Interpreter in `/usr`, sondern den im SVN-Installationsverzeichnis ausführen. Aus diesem Grund muss das `bin`-Verzeichnis der SVN-Installation in `PATH` eine hohe Priorität bekommen, also an erster Stelle stehen.

```
$ GAMBAS_ROOT=~/.bin/gambas3_trunk PATH=~/.bin/gambas3_trunk/bin:$PATH gambas3
```

Diese Zeile setzt zuerst `GAMBAS_ROOT` auf das SVN-Verzeichnis, danach erweitert sie `PATH` am Anfang um den Pfad zum gepatchten Interpreter und startet anschließend mit diesen Einstellungen die IDE, welche nun mit dem neuen Interpreter läuft, wie auch alle ihre Kindprozesse.

Da die Zeile recht lang ist, soll die Umstellung zwischen den Interpretern ebenfalls über Skripte realisiert werden. Fügen Sie den folgenden Alias ihrer `.bashrc` (in Ihrem Home-Verzeichnis) hinzu:

```
alias gbx3='GAMBAS_ROOT=$(cat $HOME/.config/gambas_root) PATH=$GAMBAS_ROOT/bin:$PATH gbx3'
```

In jeder folgenden Shell-Sitzung wird "`gbx3`" durch die längere Sequenz hinter dem Gleichheitszeichen ersetzt. Der Alias lädt mittels "cat" eine (zu erstellende) Konfigurationsdatei `gambas_root` aus einem versteckten Konfigurationsordner in Ihrem Home-Verzeichnis. Der in dieser Datei gespeicherte Pfad wird benutzt, um `GAMBAS_ROOT` und danach `PATH` zu ersetzen. Je nach Inhalt dieser Konfigurationsdatei `$HOME/.config/gambas_root` können Sie verschiedene Interpreter benutzen.

Beachten Sie, dass der Name des Alias `gbx3` ist, also dem Namen des Gambas-Interpreters entspricht. Daher arbeitet der Alias transparent. Er gilt aber offensichtlich nur für den Interpreter `gbx3`. Sie können damit zum Beispiel nicht auf den Zweit-Compiler `gbc3` oder die IDE `gambas3` zugreifen. Analoge Aliase für die weiteren Programme des Gambas-Pakets zu erstellen dürfte Ihnen nicht schwer fallen.

Die Gambas-Installation wird über den Inhalt der folgenden Konfigurationsdatei festgelegt:

```
[01] $ echo >~/.config/gambas_root
[02] $ gbx3 /tmp/avltree-test/
[03] ...
[04]
[05] End of tests. Program terminating.
[06]
[07] $ echo "$HOME/bin/gambas3_trunk" >~/.config/gambas_root
[08] $ gbx3 /tmp/avltree-test/
[09] MMain.AvlTreeTest.12: #2: Cannot load class 'AvlTree': Unable to load class file
[10] 1: MMain.AvlTreeTest.12
[11] 2: MMain.Main.4
```

- In Zeile [01] wird die Konfigurationsdatei geleert und u.U. erstellt. Sie sehen in der ersten Ausführung des Interpreters (Zeile [02]) eine personalisierte Standardinstallation. Bedenken Sie:

Die Konfigurationsdatei ist leer. Das bedeutet, dass GAMBAS_ROOT leer und somit invalid ist. Der Interpreter benutzt in diesem Fall die Standardinstallation. Er lädt eine neue Version der Komponente gb.data, um ein Test-Projekt zur Klasse AviTree auszuführen.

- Die Ausgaben in Zeilen [03] - [06] entstammen diesem Programm. Gegenwärtig existiert die AviTree-Klasse noch nicht im offiziellen Gambas-Repositorium. Da die Parallelinstallation aus dem SVN-Repositorium heruntergeladen und kompiliert wurde, verfügt die zugehörige Komponente gb.data noch nicht über die neue AviTree-Klasse.
- In Zeile [07] wird der Pfad zur Zweitinstallation in die Konfigurationsdatei geschrieben. Eine erneute Ausführung der gleichen Kommandozeile startet nun aber den Zweit-Interpreter. Dieser verwendet ordnungsgemäß seine Komponente unterhalb von GAMBAS_ROOT und muss aufgrund der fehlenden Klasse abbrechen.

Da der Wechsel zwischen mehreren Gambas-Versionen im hier beschriebenen Umfeld allein durch eine Änderung des Dateiinhalts von \$HOME/.config/gambas_root möglich ist, wird Ihnen die Aufgabe überlassen, die zu set_gambas3.sh und reset_gambas3.sh analogen Skripte zu schreiben (Tipp: Sehen Sie sich Zeilen [01] und [07] noch einmal an). Oder Sie entwickeln und testen ein Gambas-Programm, um Ihre lokalen Gambas-Installationen komfortabel graphisch zu verwalten... .

Auch die Nachteile dieser alternativen Lösung sollen erwähnt werden:

- Sie müssen einen inoffiziellen Patch anwenden.
- Wegen des Aliases in der Datei .bashrc können Sie die vorgestellte Vereinfachung über die Konfigurationsdatei nur in interaktiven Shell-Sitzungen nutzen.
- Einige Programme setzen eine einzige Gambas-Installation unter /usr voraus. Es kann zu Kompatibilitätsproblemen kommen, wenn diese Programme auf Ressourcen unter /usr zugreifen, die nicht auf die verwendete, lokale Gambas-Version passen.

1.2.7.4 Probleme bei Updates

Da es sich bei der angebotenen 2. Lösung um einen inoffiziellen Patch handelt, der nicht im offiziellen Gambas-Repositorium enthalten ist, haben Sie eine geänderte Version der Interpreter-Quelltexte vorliegen. Werden offizielle Änderungen in der gleichen Datei vorgenommen, kann es zu Konflikten kommen, die Ihnen Subversion bei einem Update jedoch mitteilt.

Sollten Sie den Konflikt nicht selbst lösen können, ist es immer möglich, den Patch überschreiben zu lassen. Das Kommando

```
$ svn revert $HOME/src/gambas3_trunk/main/share/gb_file_temp.h
```

setzt die Änderungen des Patches zurück, spielt die neue SVN-Version der Datei ein und markiert den Konflikt dann als gelöst. Beachten Sie, dass Ihr Interpreter nun nicht mehr über die Möglichkeit verfügt, alternative Gambas-Installationen anhand von GAMBAS_ROOT zu verwenden.