

### 11.5.3 Der Umgang mit Problemen und Bugs

Immer wieder kommt es bei der Programmierung mit Gambas zu Problemen. Dann ist es gut, wenn man sich an andere Programmierer und an die Entwickler von Gambas wenden kann, denn die Gambas-Community ist engagiert und kommunikativ. Die Chancen für Lösungsansätze und Lösungen stehen gut, wenn Sie sich mit einem aussagekräftigen Bug-Report melden. Beachten Sie, dass das Zusammenstellen eines Bug-Reports einigen Grundregeln folgt. In diesem Kapitel berichtet der Autor Claus D. von seinen Erfahrungen mit Fehler-Reports.

#### 11.5.3.1 Selbsthilfe

Basisinformationen zur Programmiersprache Gambas bietet das Gambas-Wiki unter <http://gambaswiki.org/wiki>. Unter <https://lists.gambas-basic.org/listinfo/user> findet man eine Liste von nationalen und internationalen Foren und Webseiten, die sich mit Gambas beschäftigen. Hervorzuheben ist das deutsch-sprachige Forum <https://www.gambas-club.de>.

Das Online-Gambas-Buch kann anhand seiner vielen Beispiele und Projekte aus verschiedensten Bereichen besonders dort hilfreich sein, wo die Gambas-Dokumentation nur unzureichende Informationen zu Klassen, Komponenten oder Bibliotheken liefert.

Darüber hinausgehende Probleme wie zum Beispiel

- unerwartetes Programmverhalten oder
- Programmabstürze

bedürfen allerdings i.d.R. des Kontakts zu den Entwicklern von Gambas. Die nachfolgend beschriebenen Vorgehensweisen versprechen erfahrungsgemäß die besten und schnellsten Ergebnisse, wenn es in Ihren Programmen oder zum Beispiel beim Einsatz einer Gambas-Komponente mal wieder klemmt.

#### 11.5.3.2 Erste Hilfe durch die Gambas Mailing List

Für solche Fälle, in denen Sie sich nicht hundertprozentig sicher ist, ob es sich um einen Bug handelt, steht die Gambas Mailing List unter <https://lists.gambas-basic.org/listinfo/user> zur Verfügung. Dieses Forum bietet erstklassige Unterstützung von Gambas-Entwicklern.

Vor dem Verschicken eines Bug-Reports und der dafür erforderlichen Registrierung sollten Sie hinreichend sicher sein, dass für das erkannte Problem noch keine Lösungen diskutiert oder angeboten wurde. Nutzen Sie deshalb die komfortable Suchfunktion unter <https://lists.gambas-basic.org/cgi-bin/search.cgi>. Hier ein Beispiel:

### Search the Gambas Mailinglists

Search for:

Extended

combobox : 3432, remove : 3278

Results 1-10 of 10 ( 0.030 seconds)

1 [Gambas-user] ComboBox Error when ComboBox.Remove("1") / possible bug in gb.gtk? [ 11.686% Popularity: 0.43250 ]

Abbildung 11.5.3.2.1: Beispiel: Abfrage in der Gambas Mailing List

Die Schilderung des Fehlers muss in englischer Sprache erfolgen, möglichst präzise formuliert sein und unbedingt alle Details liefern, die für eine Isolation oder Reproduktion des Fehlers relevant und hilfreich sein könnten. Dazu gehören folgende Angaben:

- Gambas-Version,
- Linux-Distribution (Suse, Debian, Mint, Ubuntu) mit Version und Desktop-Umgebung,
- Genutztes GUI Toolkit wie gb.qt4, gb.qt5 oder gb.gtk,
- Linux Kernel-Version und
- Architektur (32 oder 64 Bit).

Diese und weitere Informationen kann man über die Gambas-IDE über den Menüpunkt ?/ Systeminformationen .. abfragen und zur Weiterverarbeitung in die Zwischenablage kopieren.

Hier ein Auszug der Informationen – speziell zum Betriebssystem:

```
[System]
Gambas=3.12.2
OperatingSystem=Linux
Kernel=4.15.0-45-generic
Architecture=x86_64
Distribution=Linux Mint 18.3 Sylvia
Desktop=CINNAMON
Theme=Gtk
Language=de_DE.UTF-8
Memory=7893M
...
```

Einen (originalen) Bug-Report und die Reaktionen auf den Report können Sie hier nachlesen:

```
Hi Benoit,
in the Preferences> Projects form, two translations are missing:
Compress PNG ... and Automatic translation ...
I immediately sent the translations and made the merger request. Have I made a mistake of communication, do I have to wait for more translations?

Is it normal to always receive from GitLab: "gambas | Pipeline #46825672 has failed for italian-translations" ?

Regards
Gianluigi
```

```
Theoretically no, pratically yes: apparently the two ArchLinux pipelines do not work anymore. According to the error message, it seems that they rely on an image that is now missing. I can't say more, I didn't write the Gambas GitLab config file.
```

```
Regards,
--
Benoît Minisini
```

```
It's fixed. It's the name of the archlinux images on the docker website that changed.
--
Benoît Minisini
```

Schicken Sie den Entwicklern bereits in Ihrem ersten Bug-Report ein kleines Gambas-Projekt zu, bei dem der berichtete Fehler sicher ausgelöst wird. Das wäre die anschaulichste und effektivste Form, das erkannte Problem klar zu beschreiben.

### 11.5.3.3 Der Gambas Bug Tracker

Eine Meldung von Fehlern sollte ausschließlich über den Gambas Bug-Tracker erfolgen, den Sie unter <http://gambaswiki.org/bugtracker> erreichen. Sie können jedoch nur dann einen Bug-Report verschicken, wenn Sie sich *vorher* mit einer gültigen EMail-Adresse auf der o.a. Website registriert haben.

Für ein Posting sollten Sie so vorgehen:

- Wählen Sie aus, ob es sich um einen Bug oder einen Antrag für eine neue Funktion handelt.
- Legen Sie nach Ihrer Einschätzung die Priorität Ihres Reports fest (High, Low, Medium).
- Wählen Sie die betroffene Komponente beziehungsweise den Themenbereich aus der angebotenen Liste aus.
- Wählen Sie die Gambas-Version aus.
- Fügen Sie in das Feld "Description" alle wichtigen Informationen ein:
  - Dreistellige Gambas Version
  - Linux-Distribution (Suse, Debian, Mint, Ubuntu) mit Version und Desktop-Umgebung
  - Genutztes GUI Toolkit wie gb.qt4, gb.qt5 oder gb.gtk
  - Linux Kernel-Version
  - Architektur (32 oder 64 Bit)

- Kopieren Sie unbedingt alle Systeminformationen in das Feld "System description".
- Laden Sie ein Programmbeispiel hoch, in dem das Problem isoliert wurde.
- Liefern Sie möglichst zeitnah Informationen, wenn diese vom Entwickler angefordert werden.
- Ändern Sie bei Bedarf den Thread-Status, der in einer Auswahlliste angeboten wird.

Zur besseren Veranschaulichung von GUI-Problemen sollten Sie auch die Zusendung von Bildschirm-Videos in Betracht ziehen.

Alle Vorgänge in einem Bug-Thread sind mit einer Ticket-Nummer versehen und werden an Ihre hinterlegte EMail-Adresse geschickt, so dass Sie immer auf dem Laufenden gehalten werden.

#### 11.5.3.4 Dokumentation von Programm-Abstürzen

Um Programmabstürze, die beispielsweise durch eine Speicherschutzverletzung (segmentation/signal #11 fault) ausgelöst werden, für eine Meldung im Bug-Tracker dokumentieren zu können, werden von den Gambas-Entwicklern zwei Programme empfohlen:

- GNU Debugger
- Programm Valgrind

#### 11.5.3.5 Verwendung des GNU Debuggers

Zur Vorbereitung des Tests öffnet man eine Konsole und macht das Gambas-Projekt-Verzeichnis zum aktuellen Verzeichnis:

```
$ cd /home/claus/Gambas/project_folder
```

Geben Sie dann folgendes Kommando ein, um den GNU Debugger zu starten:

```
$ gdb gbx3
```

Die Konsolenausgabe des Debuggers für das Gambas-Projekt 'test' sah so aus:

```
claus@claus-VirtualBox:~/Gambas/test$ gdb gbx3
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
...
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from gbx3...(no debugging symbols found)...fertig.
(gdb)
```

Um das Programm im aktuellen Projekt-Verzeichnis mit dem Gambas-Interpreter zu starten, gibt man hinter dem Debugger-Prompt (*gdb*) das run-Kommando ein und bringt das Programm an die Stelle, an dem es abstürzt.

Die Ausgabe des Debuggers sah dann beispielsweise so aus:

```
(gdb) run
Starting program: /usr/bin/gbx3
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7fffea517700 (LWP 4722)]
qt5ct: using qt5ct plugin
[New Thread 0x7fffe0231700 (LWP 4723)]
Thread 1 "gbx3" received signal SIGSEGV, Segmentation fault.
0x00005555555573f47 in ?? ()
(gdb)
```

Es handelt sich in diesem Fall um eine erkannte Speicherschutzverletzung (Segmentation Fault).

Der Programmablauf vor dem Programmabsturz ist für die Fehleranalyse des Entwicklers von großem Nutzen und kann über ein sogenanntes Backtrace ermittelt werden. Die Backtrace-Funktion wird durch die Eingabe des bt-Kommandos nach dem Debugger-Prompt gestartet, wobei die Ausgabe dann so aussieht:

```
(gdb) bt
#0 0x00007ffff5e827a8 in ?? () from /usr/lib/gambas3/gb.qt5.so
#1 0x00007ffff432f5b5 in QObject::activate(QObject*, int, int, void**) ()
    from /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#2 0x00007ffff5373b82 in QAbstractButton::clicked(bool) ()
    from /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
...
#5 0x00007ffff537536d in QAbstractButton::mousePressEvent(QMouseEvent*) ()
    from /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#6 0x00007ffff52c1038 in QWidget::event(QEvent*) ()
    from /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#7 0x00007ffff528282c in QApplicationPrivate::notify_helper(QObject*, QEvent*)
    () from /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#8 0x00007ffff528a64f in QApplication::notify(QObject*, QEvent*) ()
```

Achtung: Dieses Beispiel entspricht keinem echten Bug und bildet nur einen Auszug der Ausgaben des GNU Debuggers ab!

Um den GNU Debugger zu beenden geben Sie folgendes Kommando ein:

```
(gdb) quit
```

Für eine Fehleranalyse benötigen die Gambas-Entwickler alle Ausgaben des GNU Debuggers nach den Kommandos run und bt, die Sie in den Bug-Report hineinkopieren sollten.

### 11.5.3.6 Verwendung von Valgrind

Manche Programmabstürze finden im Verhältnis zum eigentlichen Bug mit einer Verzögerung statt, weshalb ein Stack Backtrace nicht immer ausreicht. Um auch hier Analysen durchführen zu können bietet sich die freie Software *Valgrind* an, die die Ausführung eines Programms in einer emulierten CPU-Umgebung erlaubt und so umfangreiche Beobachtungen und Analysen ermöglicht. Der Programmablauf des zu testenden Gambas-Programms wird dadurch erheblich verlangsamt, was angesichts des angestrebten Zieles akzeptiert werden muss.

Die Installation von Valgrind kann in einer Konsole durchgeführt werden:

```
$ sudo apt install valgrind
```

Zur Vorbereitung des Tests öffnen Sie eine Konsole und machen das Gambas-Projekt-Verzeichnis zum aktuellen Verzeichnis:

```
$ cd /home/claus/Gambas/project_folder
```

Der Test wird durch folgendes Kommando in einer Konsole gestartet:

```
$ valgrind --tool=memcheck --num-callers=50 gbx3 > valgrind.out 2>&1
```

Zur Protokollierung der Testergebnisse wird dabei die Datei *valgrind.out* im Projektverzeichnis angelegt, die Sie den Gambas-Entwicklern in einem Report vollständig zur Verfügung stellen sollten. Da Valgrind versuchen wird das Programm nach einem Fehler weiter auszuführen, kann es erforderlich sein, das Programm durch die Eingabe von Strg+C zu stoppen.

Hier ein Auszug des Inhalts der Datei valgrind.out:

```
==32717== Process terminating with default action of signal 11 (SIGSEGV)
==32717== Access not within mapped region at address 0xFFFFFFFFFFFFFFFF9
==32717== at 0x127F47: ??? (in /usr/bin/gbx3)
==32717== by 0x1372B9: ??? (in /usr/bin/gbx3)
```

```
==32717== by 0x1550F6: ??? (in /usr/bin/gbx3)
==32717== by 0x15AFBC: ??? (in /usr/bin/gbx3)
==32717== by 0x15DC58: ??? (in /usr/bin/gbx3)
==32717== by 0x13B228: ??? (in /usr/bin/gbx3)
==32717== by 0x13D2EF: ??? (in /usr/bin/gbx3)
==32717== by 0x6F87B58: ??? (in /usr/lib/gambas3/gb.qt5.so.0.0.0)
==32717== by 0x6F8A129: ??? (in /usr/lib/gambas3/gb.qt5.so.0.0.0)
==32717== by 0x15BD42: ??? (in /usr/bin/gbx3)
==32717== by 0x15DC8E: ??? (in /usr/bin/gbx3)
==32717== by 0x1261B9: ??? (in /usr/bin/gbx3)
==32717== by 0x4E5DB96: (below main) (libc-start.c:310)
==32717== If you believe this happened as a result of a stack
==32717== overflow in your program's main thread (unlikely but
==32717== possible), you can try to increase the size of the
==32717== main thread stack using the --main-stacksize= flag.
==32717== The main thread stack size used in this run was 8388608.
```

Die rote markierte Zeile ist die Stelle, an der der Crash passierte. In diesem Fall war es ein Signal 11-Fehler (Schutzverletzung).