

24.9.0.1 Gambas und D-Bus

Die Komponente `gb.dbus` besteht aus zwei Teilen: Der Kern ist in C geschrieben und befindet sich in `gb.dbus/src`. Darauf aufbauend gibt es einen Teil, der in Gambas geschrieben ist und liegt im Pfad `gb.dbus/src/gb.dbus`. So ist zum Beispiel die Klasse `DBusVariant` in C geschrieben und liegt in der Datei `c_dbusvariant.c`, während die Klasse `DBusValue` komplett in Gambas programmiert ist.

Was Gambas und D-Bus betrifft, werden generell zwei Strategien verfolgt:

- Nutzung von vorhandenen Diensten auf dem D-Bus und
- Bereitstellung von Diensten auf dem D-Bus mit Gambas-Programmen.

Die Umsetzung dieser Strategien versetzt Sie in die Lage Folgendes für eine beliebige d-bus-fähige Anwendung zu tun:

- Aufrufen von Methoden,
- Methoden anbieten in exportierten D-Bus-Objekten,
- Lesen und Setzen von Werten von Eigenschaften,
- Abfangen von Signalen,
- Senden von Signalen.

24.9.0.1.1 Busse

D-Bus bietet mindestens zwei Busse zur Kommunikation zwischen Prozessen an. Einer ist der globale System-Bus und ein anderer der Session-Bus, zu dem jeder Desktop eines angemeldeten Benutzers innerhalb seiner Desktop-Sitzung automatisch verbunden wird. Der Standard in Gambas ist der Session-Bus.

Es laufen (mindestens) zwei D-Bus-Dämons auf einem Linux-System

```
$ env | grep DBUS_SESSION_BUS_ADDRESS
DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-CTaH4kQuEM,guid=59a12bb14e4d0385a446dd2257f34ab5
```

Sie können aber auch das Gambas-Programm `DBusView` einsetzen und erhalten diese Übersicht zu allen aktuell auf dem D-Bus registrierten Programmen:



Abbildung 24.9.0.1.1: Registrierte Programme auf dem D-Bus

Sie finden das Archiv zum Programm `DBusView` im Download-Bereich von Kapitel 24.9.2.

Mit den folgenden Befehlen können Sie eine Liste der Anwendungen auslesen, die am System-Bus oder am Session-Bus registriert sind. Die Liste enthält die D-Bus-Namen und die ihnen zugeordneten einzigartigen Verbindungs-IDs, denen stets ein Doppelpunkt vorangestellt wird. Eingesetzt wird das Programm `qdbus`:

```
$ qdbus --session | grep -v ":"
org.gnome.SessionManager
org.gtk.Private.UDisks2VolumeMonitor
...
org.PulseAudio1
org.pulseaudio.Server
org.freedesktop.DBus
```

```
$ qdbus --system | grep ":"
...
:1.44
:1.5
:1:21
```

Beachten Sie:

- Ein Gambas-Programm wird automatisch zum Session-D-Bus verbunden, wenn die Komponente *gb.dbus* geladen wird!
- Sobald Sie mindestens ein D-Bus-Objekt über die Register(...) -Methode zum D-Bus exportieren, wird Ihre Anwendung auf dem D-Bus unter dem Namen *org.gambas.<Anwendungsname>* registriert. Für *<Anwendungsname>* können Sie stets die Eigenschaft *Application.Name* einsetzen.

24.9.0.1.2 D-Bus-Nachricht

Die Kommunikation auf dem System-Bus und dem Session-Bus erfolgt durch Nachrichten – ähnlich dem Verschicken von Paketen in TCP/IP-Netzen. Im Gegensatz zum TCP/IP-Protokoll garantiert aber D-Bus die Übermittlung der kompletten Nachrichten in einem Zug. Es existieren vier Typen von D-Bus-Nachrichten:

- Method,
- Reply,
- Error,
- Signal.

Nachrichten werden als Methoden-Aufrufe oder Signale gesendet oder empfangen. Jeder Aufruf einer Methode wird immer mit einer Nachricht quittiert – nicht nur im Fehler-Fall oder bei der erfolgreichen Rückgabe eines Wertes.

In der Gambas-Klasse DBus (*gb.dbus*) werden die Nachrichten-Typen über die vier (Integer-)Konstanten *DBus.Method* (1), *DBus.Reply* (2), *DBus.Error* (3) und *DBus.Signal* (4) deklariert.

24.9.0.1.3 D-Bus-Adresse

Da mehrere Anwendungen – respektive deren Prozesse – mit dem System-Bus oder dem Session-Bus verbunden sein können ist es notwendig, jede Nachricht eindeutig zu adressieren. Die Adresse besteht aus der Angabe

- des Dienstes (Name der Anwendung),
- des Objekt-Pfades und
- (optional) der Schnittstellen (Interfaces).

24.9.0.1.4 D-Bus-Name

Ein Dienst repräsentiert eine aktive Verbindung zwischen einem Programm und dem System-D-Bus oder dem Session-D-Bus. Diese Verbindung wird durch den lesbaren D-Bus-Namen beschrieben. Für diese Beschreibung wird ein "ReverseDomainName" eingesetzt. D-Bus-Namen besitzen die Syntax: "ReverseDomainName".Anwendungsname. Beachten Sie, dass D-Bus-Namen case-sensitiv sind.

Für D-Bus-Namen in Gambas gilt die Syntax: *org.gambas.xxx* mit *xxx* als Gambas-Projekt-Name, der unter dem Wert der Eigenschaft *Application.Name* bereitgestellt und für *xxx* eingesetzt wird. Diese D-Bus-Namen zum Beispiel sind korrekt:

```
org.freedesktop.NetworkManager
org.gambas.support
```

```
org.Cinnamon.LookingGlass
org.gtk.Private.RemoteVolumeMonitor
```

Wird mehr als eine Instanz einer Anwendung am D-Bus registriert, dann wird die Prozess-ID an den Namen des Dienstes angehängt, so dass dann ein eindeutiger D-Bus-Name für die Verbindung existiert.

Gleichzeitig wird für jede Verbindung einer Anwendung zum D-Bus auch eine eindeutige ID generiert. Diese ID beginnt mit einem Doppelpunkt (:), dem eine Ziffernfolge, ein Punkt und eine weitere Ziffernfolge folgen. Eine gültige ID wäre zum Beispiel :1.63 .

24.9.0.1.5 D-Bus-Objekt

Objekte bieten den Zugriff auf Schnittstellen. Sie werden über Bezeichner wie zum Beispiel /org/freedesktop/NetworkManager oder /org/gambas/P2DBUS identifiziert, deren Syntax an Datei-Pfade erinnert. Die folgenden Regeln definieren einen validen Objekt-Pfad:

- Ein Pfad kann jede Länge besitzen.
- Ein Pfad beginnt mit dem ASCII-Zeichen '/' (ASCII dezimal 47).
- Ein einzelnes '/'-Zeichen steht für das Root-Verzeichnis.
- Mehrere '/'-Zeichen (nacheinander) sind nicht erlaubt.
- Jedes Pfad-Element nach einem '/'-Zeichen enthält nur Zeichen aus "[A-Z][a-z][0-9]_" .
- Kein Element darf ein leerer String sein.

```
qdbus --session \
org.gambas.export_dbus \
/org/gambas/export_dbus \
org.gambas.export_dbus.Compute(7,13)
```

Beachten Sie: Eine d-bus-fähige Anwendung kann einen fehlerhaft deklarierten Objekt-Pfad ignorieren!

Eine gültige Adresse zum Beispiel setzt sich aus diesen drei Angaben zusammen:

- | | |
|-------------------------------------|-------------------------------------|
| ■ Dienst (D-Bus-Name der Anwendung) | org.freedesktop.NetworkManager |
| ■ Objekt-Pfad (Path) | /org/freedesktop/NetworkManager |
| ■ Schnittstelle (Interface) | org.freedesktop.DBus.Introspectable |

und wird als gültige Adresse so verwendet:

```
org.freedesktop.NetworkManager /org/freedesktop/NetworkManager org.freedesktop.DBus.Introspectable
```

24.9.0.1.6 D-Bus-Schnittstelle (D-Bus-Interface)

Eine (aktive) Schnittstelle verfügt über Methoden, Eigenschaften und Ereignisse sowie Signale und ist am D-Bus registriert. Die Beschreibung einer Schnittstelle kann in einem XML-Dokument stehen, um für Entwickler und Benutzer als Referenz zu dienen (Introspection).