

24.9.6.2 Projekte – D-Bus-Signal aufbereiten, senden und abfangen

Im Gegensatz zu den ersten beiden Projekten, in denen von einem Gambas-Client ausgewählte Signale anderer d-bus-fähiger Anwendungen auf dem D-Bus nur abgefangen und teilweise auch der Inhalt im Variant-Array Arguments ausgewertet wurde, werden Ihnen hier zwei Projekte vorgestellt, mit denen Sie einerseits ein selbst definiertes Signal senden (Server) oder andererseits (Client) abfangen und auswerten können:



Abbildung 24.9.6.2.1: Signal-Sender

24.9.6.2.1 Projekt – Signal-Sender

Die Quelltexte für ein Projekt, mit dem auch D-Bus-Signale gesendet werden sollen, umfassen stets die Aufbereitung des zu sendenden Signals in einer speziellen Klassen-Datei – deren Namen Sie frei festlegen können – und den eigentlichen Signal-Sender in der Start-Klasse wie zum Beispiel in der Klasse FMain.class.

Signal-Definition

Die Aufbereitung des zu sendenden Signals erfolgt in der Klassen-Datei SGBVersion.class. Der Präfix S steht für Signaldefinition und GBVersion für den Signalnamen, der auch *Member* genannt wird. Den Signalnamen können Sie selbst festlegen:

```
[1] ' Gambas class file
[2]
[3] Inherits DBusObject
[4] Create Static
[5]
[6] ' (Pseudo-) signature of the list of arguments: "s" => String
[7]
[8] Event org_gambas_SignalTX_SGBVersion_GBVersion(Version As String)
```

Die Aufbereitung des zu sendenden Signals ist schnell erledigt. Die Klasse SGBVersion.class erbt in der Zeile 3 von der Klasse DBusObject. Daher ist jede Instanz von SGBVersion.class vom Datentyp DBusObject. In der Zeile 8 wird das Signal definiert. Ein D-Bus-Signal wird in Gambas grundsätzlich als Ereignis definiert. Zur Syntax der zu sendender Signale finden Sie eine Beschreibung im Kapitel 24.9.6.0.4 *Signale senden*. Da die Signatur des Signals "s" ist – es wird nur ein String gesendet – besteht die Liste der Argumente nur aus einem Element: Version mit dem Datentyp String. Der native Gambas-Datentyp 'String' wird beim Senden automatisch in den passenden D-Bus-Datentyp konvertiert.

Signal-Sender

Bevor Sie ein Signal senden können, müssen Sie das DBus-Objekt, in dem das Signal definiert wurde (SGBVersion.class), mit der Methode DBus.Raise(...) oder DBus[AppName].Raise(...) zum Session-D-Bus exportieren, damit d-bus-fähige Programme die definierten Signale beobachten können.

Quelltext Sender:

```
[1] ' Gambas class file
[2]
[3] Private $sDBusName As String
[4] Private $hDBusObject As SGBVersion
[5] Private $sDBusObjectPath As String
[6]
[7] Public Sub Form_Open()
[8]
[9]     FMain.Resizable = False
[10]     DBus.Unique = True
[11] ' For tests only: DBus.Debug = True
[12]
```

```
[13] $hDBusObject = New SGBVersion
[14] $sDBusObjectPath = "/SGBVersion"
[15]
[16] DBus.Session.Register($hDBusObject, $sDBusObjectPath)
[17]
[18] End
[19]
[20] Public Sub btnSendSignal_Click()
[21]
[22]     lblCaption.Foreground = Color.Red
[23]     SendSignal()
[24]     Wait 0.1
[25]     lblCaption.Foreground = Color.Black
[26]
[27] End
[28]
[29] Private Sub SendSignal()
[30]
[31]     Dim aArguments As New Variant[]
[32]
[33]     aArguments.Add(System.FullVersion)
[34]
[35]     DBus.Raise($hDBusObject, "org.gambas.SignalTX.SGBVersion.GBVersion", aArguments)
[36] ' $sDBusName = DBus.Name
[37] ' DBus[$sDBusName].Raise($hDBusObject, "org.gambas.SignalTX.SGBVersion.GBVersion", aArguments)
[38]
[39] End
[40]
[41] Public Sub Form_Close()
[42]
[43]     If DBus.IsRegistered($hDBusObject) Then DBus.Session.Unregister($hDBusObject)
[44]     FMain.Close()
[45]
[46] End
```

Kommentar:

- In der Zeile 16 wird das D-Bus-Objekt mit dem D-Bus-Objektpfad '/SGBVersion' zum D-Bus exportiert.
- In den Zeilen 22 und 25 wird nur kurz die Farbe in der Anzeige gewechselt, um das Senden des Signals zu signalisieren.
- Der Inhalt des Signals ist ein String, der mit *System.FullVersion* die aktuelle Gambas-Version enthält (Zeile 33).
- Das Senden des Signals wird in der Zeile 35 oder alternativ in der Zeile 37 angeschoben.
- Da ein D-Bus-Objekt mit dem D-Bus-Objektpfad '/SGBVersion' zum D-Bus exportiert wurde, wird es in der Zeile 43 wieder abgemeldet, bevor das Programm beendet wird.

24.9.6.2.2 Projekt – Client

Der Client SignalRX beobachtet das Signal mit dem Namen 'GBVersion' und dem D-Bus-Objekt-Pfad '/SGBVersion' der Anwendung mit dem D-Bus-Namen org.gambas.SignalTX auf dem Session-D-Bus. Trifft das Signal ein, so wird der Inhalt – die aktuelle Gambas-Version auf dem System – ausgelesen und zusammen mit dem vorangestelltem Datum und der aktuellen Zeit in einer Text-Area angezeigt:

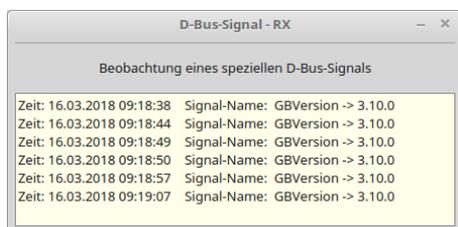


Abbildung 24.9.6.2.2: Signal-Empfänger

Der Quelltext für den Client wird vollständig angegeben und anschließend kommentiert:

```
[1] ' Gambas class file
[2]
[3] Private $hDBusSignal As DBusSignal
[4] Private $sDBusInterface As String
[5] Private $sDBusSignalName As String
[6]
[7] Public Sub Form_Open()
```

```

[8]
[9] Dim sMessage As String
[10]
[11] If Not DBus.Session.Applications.Exist("org.gambas.SignalTX") Then
[12]     sMessage = ("There is no suitable data server on the session bus!")
[13]     sMessage &= "<center><font color='red'>"
[14]     sMessage &= ("The program is terminated.")
[15]     sMessage &= "</font></center>"
[16]     Message.Warning(sMessage)
[17]     Quit
[18] Endif
[19]
[20] FMain.Resizable = False
[21] lblCaption.Text = ("WATCHING SPECIAL SIGNAL")
[22] ' For tests only: DBus.Debug = True
[23]
[24] $sDBusInterface = "org.gambas.SignalTX.SGBVersion"
[25] $sDBusSignalName = "GBVersion" ' The signal name is called 'Member'
[26]
[27] $hDBusSignal = New DBusSignal(DBus.Session, $sDBusInterface, True) As "ObservedSignal"
[28]
[29] End
[30]
[31] Public Sub ObservedSignal_Signal(Signal As String, Arguments As Variant[])
[32]
[33]     If Signal = $sDBusSignalName Then
[34]         TextAreal.Insert("Zeit: " & Format(Now(), "dd.mm.yyyy hh:nn:ss") & "      Signal-Name: " & Signal
[35]         & " -> " & Arguments[0] & gb.NewLine)
[36]         TextAreal.Pos = TextAreal.Length
[37]     Endif
[38] End
[39]
[40] Public Sub Form_Close()
[41]
[42]     If $hDBusSignal Then $hDBusSignal.Enabled = False
[43]     FMain.Close()
[44]
[45] End
    
```

Kommentar:

- In der Zeile 11 wird geprüft, ob das (Server-)Programm existiert. Diese Art der Überprüfung funktioniert aber nur dann, wenn im Quelltext des Servers die Eigenschaft `DBus.Unique` auf den Wert `True` gesetzt wurde. Im Kapitel 24.9.2.1 können Sie nachlesen, welche Alternative es für den Standard `DBus.Unique = False` gibt.

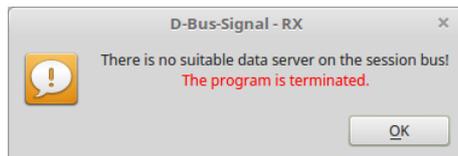


Abbildung 24.9.6.2.3: Fehlstart ...

- Ein neuer Signal-Observer mit dem Event-Namen 'ObservedSignal' wird in der Zeile 27 erzeugt.
- Wie reagiert wird, wenn das beobachtete Signal eintrifft und damit das Ereignis 'Signal' auslöst, wird in der Ereignis-Behandlungsroutine in den Zeilen 31 bis 38 festgelegt.
- Der Signal-Observer wird in der Zeile 42 ausgeschaltet, bevor das Client-Programm beendet wird.