

24.9.6.3 Projekte – D-Bus-Signale aufbereiten, senden und abfangen

Im Gegensatz zu den beiden Projekten im Kapitel 24.9.6.2, bei denen die Signatur des Signals mit "s" angegeben wurde, wird in Ihnen in diesem Kapitel ein Signal mit einer komplexen Signatur vorgestellt.

24.9.6.3.1 Projekt – Signal-Sender

Signal-Definition

Die Aufbereitung des zu sendenden Signals erfolgt in der Klassen-Datei SExample.class. Den Namen der Klassen-Datei und den Signalnamen – hier SComplex – können Sie selbst festlegen:

```
[1] ' Gambas class file
[2]
[3] Inherits DDBusObject
[4] Create Static
[5]
[6] ' (Pseudo-) Signature: "sia{si}" => String Integer Collection(Key-Type = String, Value-Type = Integer)
[7]
[8] Event org_gambas_SignalTXC_SExample_SComplex(GBVersion As String, Month As Integer, c As Collection)
```

Wie Sie erkennen, hat das Signal 'SComplex' genau drei Argumente mit drei unterschiedlichen Daten-Typen und folgt damit der Signatur "sia{si}".

Signal-Sender



Abbildung 24.9.6.3.1: Signal-Sender

Der folgende Quelltext weist mit den Zeilen 8, 25, 71 und dem Bereich mit den Zeilen 30 bis 44 eine Besonderheit auf. Das gesendete Signal wird sofort vom Server abgefangen und der Inhalt in der Konsole für Kontrollzwecke ausgegeben. Diese Besonderheit half bei der Erprobung des Projektes.

Quelltext Sender:

```
[1] ' Gambas class file
[2]
[3] Private $sDBusName As String
[4] Private $hDBusObject As SExample
[5] Private $sDBusObjectPath As String
[6] Private $sDBusInterface As String
[7]
[8] Private $hDBusSignal As DDBusSignal
[9]
[10] Public Sub Form_Open()
[11]
[12]   FMain.Resizable = False
[13]
[14]   DBus.Unique = True
[15]
[16] ' For tests only: DBus.Debug = True
[17]
[18]   $sDBusName = DBus.Name
[19]   $hDBusObject = New SExample
[20]   $sDBusObjectPath = "/SExample"
[21]   $sDBusInterface = "org.gambas.SignalTXC.SExample"
[22]
[23]   DBus.Session.Register($hDBusObject, $sDBusObjectPath)
[24] ' Only for control
[25]   $hDBusSignal = New DDBusSignal(DBus.Session, $sDBusInterface, True) As "ComplexSignal"
[26]
[27] End
[28]
[29] ' Only for control
[30] Public Sub ComplexSignal_Signal(Signal As String, Arguments As Variant[])
[31]
[32]   Dim c As Collection
[33]   Dim vValue As Variant
[34]
[35]   If Signal = "SComplex" Then
```

```

[36]     Print Signal; " : "; Arguments[0]
[37]     Print "Month : " & Arguments[1]
[38]     c = Arguments[2]
[39]     For Each vValue In c
[40]         Print c.Key; " : "; vValue
[41]     Next
[42] Endif
[43]
[44] End
[45]
[46] Public Sub btnSend_Click()
[47]
[48]     Dim SGambasVersion, sSignalName As String
[49]     Dim iMonth As Integer
[50]     Dim cCollection As Collection
[51]     Dim aArguments As Variant[]
[52]
[53]     sSignalName = "org.gambas.SignalTXC.SExample.SComplex"
[54]     sGambasVersion = System.FullVersion
[55]     iMonth = Month(Now())
[56]     cCollection = [{"First": 1, "Second": 2, "Third": 3}]
[57]
[58]     aArguments = [sGambasVersion, iMonth, cCollection]
[59]
[60]     lblCaption.Foreground = Color.Red
[61]     DBus[$sDBusName].Raise($hDBusObject, sSignalName, aArguments)
[62] ' DBus.Raise($hDBusObject, sSignalName, [SGambasVersion, iMonth, cCollection]) ' Alternative
[63]     Wait 0.1
[64]     lblCaption.Foreground = Color.Black
[65]
[66] End
[67]
[68] Public Sub Form_Close()
[69]
[70]     If DBus.IsRegistered($hDBusObject) Then DBus.Session.Unregister($hDBusObject)
[71]     If $hDBusSignal Then $hDBusSignal.Enabled = False
[72]     FMain.Close()
[73]
[74] End

```

Kommentar:

- In der Zeile 23 wird das D-Bus-Objekt mit dem Objektpfad '/SEexample' zum Session D-Bus exportiert.
- In den Zeilen 54 bis 56 werden die Werte der drei Signal-Argumente festgelegt und in der Zeile 58 dem Array der Argumente 'aArguments' als Elemente übergeben.
- Das Senden des Signals wird in der Zeile 61 oder alternativ in der Zeile 62 angeschoben.
- Da ein D-Bus-Objekt mit dem D-Bus-Objektpfad '/SEexample' zum D-Bus exportiert wurde, wird es in der Zeile 70 wieder abgemeldet.
- In der Zeile 71 wird der zusätzlich etablierte Signal-Observer ausgeschaltet, bevor das Programm beendet wird.

24.9.6.3.2 Projekt – Client

Der Client SignalRXC beobachtet permanent das Signal mit dem Namen 'SComplex' des D-Bus-Objektes mit dem Pfad '/SEexample' der Anwendung mit dem D-Bus-Namen org.gambas.SignalTXC auf dem Session-D-Bus.

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

```

[1] ' Gambas class file
[2]
[3] Private $hDBusSignal As DBusSignal
[4] Private $sDBusInterface As String
[5]
[6] Public Sub Form_Open()
[7]
[8]     Dim sMessage As String
[9]
[10]    If Not DBus.Session.Applications.Exist("org.gambas.SignalTXC") Then
[11]        sMessage = ("There is no suitable data server on the session bus!")
[12]        sMessage &= "<center><font color='red'>"
[13]        sMessage &= ("The program is terminated.")
[14]        sMessage &= "</font></center>"
[15]        Message.Warning(sMessage)
[16]        Quit
[17]    Endif

```

```

[18]
[19] FMain.Resizable = False
[20] lblCaption.Text = ("WATCHING SPECIAL SIGNAL")
[21] ' For tests only: DBus.Debug = True
[22]
[23] $sDBusInterface = "org.gambas.SignalTXC.SExample"
[24]
[25] $hDBusSignal = New DBusSignal(DBus.Session, $sDBusInterface, True) As "ObservedSignal"
[26]
[27] End
[28]
[29] Public Sub ObservedSignal_Signal(Signal As String, Arguments As Variant[])
[30]
[31] Dim c As Collection
[32] Dim vValue As Variant
[33]
[34] If Signal = "SComplex" Then
[35]     txtarResult.Insert(gb.NewLine)
[36]     txtarResult.Insert("Gambas-Version = " & Arguments[0] & gb.NewLine)
[37]     txtarResult.Insert("Month : " & Arguments[1] & gb.NewLine)
[38]     c = Arguments[2]
[39]     For Each vValue In c
[40]         txtarResult.Insert(c.Key & " : " & vValue & gb.NewLine)
[41]     Next
[42]     txtarResult.Pos = txtarResult.Length
[43] Endif
[44]
[45] End
[46]
[47] Public Sub btnIntrospection_Click()
[48]
[49] Dim hDBusApplication As DBusApplication
[50] Dim hDBusType As DBusConnection
[51] Dim hDBusProxy As DBusProxy
[52]
[53] Dim sDBusName As String
[54] Dim sDBusObjectPath As String
[55] Dim sIntrospection As String
[56]
[57] If DBus.Session.Applications.Exist("org.gambas.SignalTXC") Then
[58]     hDBusType = DBus.Session
[59]     sDBusName = "org.gambas.SignalTXC"
[60]     hDBusApplication = New DBusApplication(hDBusType, sDBusName)
[61]
[62]     sDBusObjectPath = "/SExample"
[63]
[64]     Try hDBusProxy = New DBusProxy(hDBusApplication, sDBusObjectPath)
[65]     If Error Then
[66]         Message.Error("ERROR!" & gb.NewLine & Error.Where & gb.NewLine & Error.Text)
[67]         Return
[68]     Endif
[69]     sIntrospection = hDBusProxy.Introspect()
[70]     txtarResult.Insert(gb.NewLine & sIntrospection)
[71]     txtarResult.Pos = txtarResult.Length
[72] Endif
[73]
[74] End
[75]
[76] Public Sub Form_Close()
[77]
[78] If $hDBusSignal Then $hDBusSignal.Enabled = False
[79] FMain.Close()
[80]
[81] 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 `True` gesetzt wurde. Im Kapitel 24.9.2.1 können Sie nachlesen, welche Alternative es für den Standard-Fall `DBus.Unique = False` gibt.
- Ein neuer Signal-Observer mit dem Event-Namen 'ObservedSignal' wird in der Zeile 25 erzeugt.
- In der Ereignis-Behandlungsroutine in den Zeilen 29 bis 45 werden die Signal-Argumente ausgelesen und angezeigt.
- Die Zeilen 47 bis 74 sollen zeigen, dass es für Signale – im Gegensatz zu Eigenschaften und Methoden – in d-bus-fähigen Gambas-Programmen nicht möglich ist, ein beschreibendes XML-Dokument auszulesen (Introspection). Offensichtlich ist das für Signale nicht umgesetzt worden. Daher ist es wichtig, dass Sie die Signaturen der Signale im Quelltext gut dokumentieren. Sonst wird es aufwändiger, die abgefangenen Signale korrekt auszuwerten!

- Der Signal-Observer wird in Zeile 42 ausgeschaltet, bevor das Client-Programm beendet wird.

Wird das beobachtete Signal mit der Signatur "sia{si}" abgefangen, so wird dessen Inhalt ausgelesen. In einer Text-Area werden eine Zeichenkette, eine Zahl und die drei Elemente einer Collection mit geeigneten Labeln angezeigt:

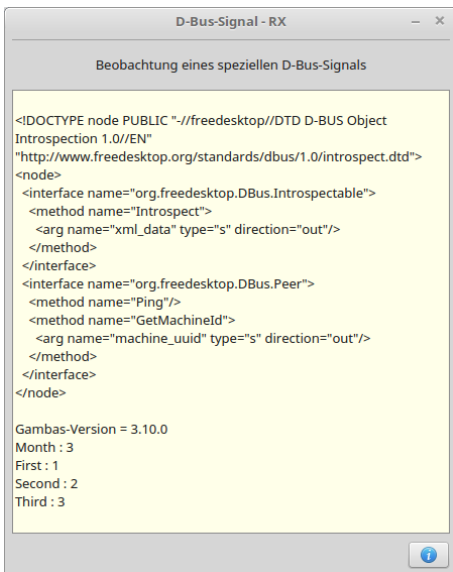


Abbildung 24.9.6.3.2: Signal-Empfänger