

24.9.5.0 DBusObserver

Die Klasse *DBusObserver* (gb.dbus) ermöglicht es Ihnen, jede über den D-Bus gesendete Nachricht abzufangen und auszuwerten.

Die Klasse *DBusObserver* hat zwei Eigenschaften, zwei Methoden und ein Ereignis.

24.9.5.0.1 Eigenschaften

Die Klasse *DBusObserver* verfügt über diese beiden Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Enabled	Boolean	Der Standard-Wert ist True. Das bedeutet, dass ein D-Bus-Observierer stets eingeschaltet ist, wenn Sie ein Programm starten, das diese Klasse nutzt. Sie können den Wert lesen als auch festlegen.
Message	DBusObserver.Message	Message vom Typ <i>.DBusObserver.Message</i> enthält die Nachricht, die vom Observer empfangen wurde.

Tabelle 24.9.5.0.1 : Eigenschaften der Klasse DBusObserver

24.9.5.0.2 Eigenschaften der Klasse .DBusObserver.Message

Die virtuelle Klasse *.DBusObserver.Message*, die nicht erzeugt werden kann, repräsentiert eine Nachricht, die vom DBusObserver-Objekt empfangen wurde. Sie hat die folgenden Eigenschaften, die nur ausgelesen werden können und bis auf 'Arguments' als Meta-Daten zu verstehen sind:

Eigenschaft	Datentyp	Beschreibung
Arguments	Variant[]	Ein Variant-Array, das alle Argumente einer Nachricht enthält. Es wird NULL zurückgegeben, wenn die empfangene Nachricht <i>keine</i> Argumente besitzt.
Destination	String	Nachrichten-Ziel.
Interface	String	Nachrichten-Interface.
Member	String	Nachrichten-Member (Methode oder Signal).
Object	String	Pfad des Nachrichten-Objekts.
Sender	String	Nachrichten-Sender.
Serial	Integer	Nachrichten-Serien-Nummer.
Type	Integer	Nachrichten-Typ, der einen der folgenden Werte – repräsentiert durch eine der folgenden Konstanten – annehmen kann: DBus.Method (1), DBus.Reply (2), DBus.Error (3) oder DBus.Signal (4).

Tabelle 24.9.5.0.2 : Eigenschaften der virtuellen Klasse .DBusObserver.Message

24.9.5.0.3 Methoden

Die Klasse *DBusObserver* hat nur diese beiden Methoden:

Methode	Beschreibung
Error ([Error As String, Type As String])	Es wird als Antwort auf eine Nachricht mit einer Fehlermeldung reagiert. Für die beiden Parameter gilt: Error ist die Fehlermeldung. Type ist der Fehlertyp. Standardmäßig ist es "org.freedesktop.DBus.Error.Failed". Sehen Sie sich in der D-Bus-Dokumentation die Liste der erlaubten Fehlertypen an.
Reply ([Signature As String, Arguments As Array])	Antwort auf den Aufruf der Methode <i>Nachricht</i> . Für die beiden Parameter gilt: Signature ist die Signatur der Antwort. Es sollte die Kennzeichnung der Ausgabeargumente des Methodenaufrufs sein. Arguments ist ein Array mit den zurückgegebenen Antwort-Werten.

Tabelle 24.9.5.0.3 : Methoden der Klasse DBusObserver

24.9.5.0.4 Ereignis

Die Klasse DbusObserver verfügt nur das Ereignis *Message()*. Dieses Ereignis wird ausgelöst, wenn der Dbus-Observer eine Nachricht von einem bestimmten Objekt empfängt. Die empfangene Nachricht wird in der Message-Eigenschaft vom Typ *.DBusObserver.Message* gespeichert.

Diese Klasse können Sie erzeugen. Sie sollten diese Klasse dann einsetzen, wenn Sie zusätzliche Informationen – wie zum Beispiel die oben erwähnten Meta-Daten – zu den beobachteten Nachrichten benötigen. Verwenden Sie sonst die Klasse *DBusSignal*.

24.9.5.0.5 DbusObserver erzeugen

Im folgenden Beispiel erfahren Sie, wie Sie mit *New DbusObserver(parameterliste)* formal einen neuen Dbus-Observer erzeugen können:

```
Dim hDBusObserver As DbusObserver
hDBusObserver = New DbusObserver ( Connection As DbusConnection, Type As Integer [ , Object As String, Member As String, Interface As String, Destination As String ] ) As "ObserverEventName"
```

Der o.a. Quelltext erzeugt einen neuen D-Bus-Observer, um Nachrichten auf dem angegebenen Dbus abzufangen. Für die Parameter gilt:

- Connection ist das Dbus-Connection-Object das den D-Bus repräsentiert.
- Type ist einer der 4 möglichen Nachrichten-Typen.
- Object ist das Objekt, das beobachtet werden soll. Ist der optionale Parameter *nicht* angegeben, dann werden alle Objekte beobachtet.
- Member ist die zu beobachtende Methode oder der Name des zu beobachtenden Signals. Ist der optionale Parameter *nicht* angegeben, werden alle Methoden-Aufrufe und alle Signale beobachtet.
- Interface ist das zu beobachtende Interface. Ist der optionale Parameter *nicht* angegeben, so werden alle Interfaces beobachtet.
- Destination ist das Ziel der Nachricht. Wenn nichts anderes angegeben ist, dann werden nur die Nachrichten abgefangen, die an Ihre Anwendung gesendet werden. Wenn für den optionalen Wert von Destination das Muster "*" deklariert ist, dann werden alle Nachrichten von allen registrierten Anwendungen abgefangen.

So erzeugen Sie einen neuen Observer, um Signale des Objektes abzufangen, dessen Objekt-Pfad mit */org/gtk/Private/RemoteVolumeMonitor* angegeben ist:

```
Dim hDBusObserver As DbusObserver
hDBusObserver = New DbusObserver(DBus.Session, Dbus.Signal, "/org/gtk/Private/RemoteVolumeMonitor", "VolumeAdded", "org.gtk.Private.RemoteVolumeMonitor", "*") As "hDBusObserver"
```