

12.5 Embedder (gb.qt4)

Die Klasse *Embedder (gb.qt4)* bietet ein Steuer-Element, über das X11-Fenster einer anderen X11-Anwendung in ein Gambas-Fenster eingebettet werden können. Die Klasse Embedder gilt mit *gb.qt5* als veraltet und sollte nicht weiter eingesetzt werden! Hinweise zu den Hintergründen können Sie nachlesen, wenn Sie diesen Links folgen:

<http://gambas.8142.n7.nabble.com/Embedder-deprecated-in-gb-qt5-td52992.html>
<http://www.mail-archive.com/gambas-user@lists.sourceforge.net/msg20712.html>

Die beeindruckenden Möglichkeiten der Klasse *Embedder* sollen in einem Projekt im → Abschnitt 12.5.4 gezeigt werden, bei dem Sie aus einem Gambas-Programm heraus ein anderes Programm starten und dessen Fenster in das Gambas-Programm einbetten. Wenn Sie zum Beispiel das Fenster des Editors LibreOffice-Writer einbetten, so können Sie einen Text im eingebetteten Fenster von LibreOffice-Writer schreiben oder bearbeiten, den Text speichern und in das PDF-Format umwandeln.

Das folgende Bild zeigt ein Gambas-Programm mit dem eingebetteten Fenster des Programms 'Gnu-Plot', dem als Argument eine Skript-Datei mit GnuPlot-Befehlen mitgegeben wurde:

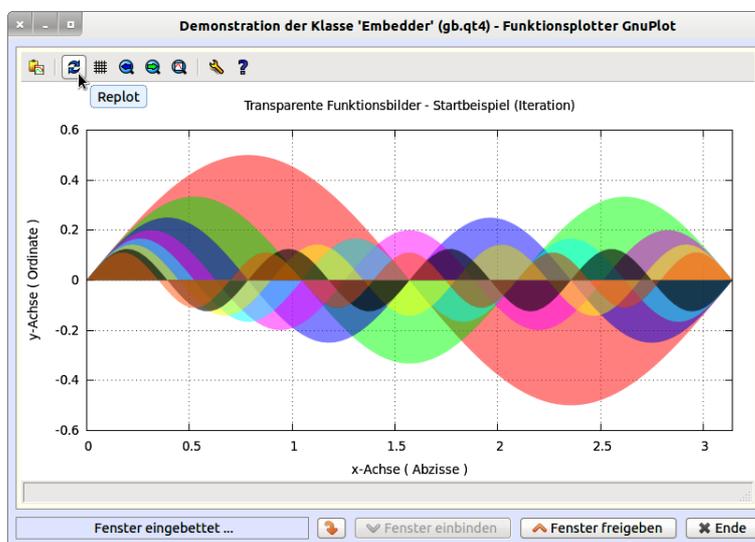


Abbildung 12.5.1: Programm 'GnuPlot' – eingebettet in ein Gambas-Programm

12.5.1 Eigenschaften

Von der Klasse Embedder werden nur wesentliche Eigenschaften beschrieben:

Eigenschaft	Datentyp	Beschreibung
Client	Integer	Gibt die X11-Fensterkennung der einzubettenden (externen) Anwendung zurück.
Handle	Integer	Liefert das interne X11-Fenster-Handle des Steuer-Elementes.

Tabelle 12.5.1.1 : Eigenschaften der Klasse Embedder

12.5.2 Methoden

Die Klasse *Embedder* verfügt u.a. über diese Methoden:

Methode	Beschreibung
Embed (Client As Integer)	Bettet das Fenster einer anderen X11-Anwendung ein. Das Argument 'Client' ist die Kennung des einzubettenden X11-Fensters. Es sollte vorbereitend geklärt sein, dass der Client dem XEmbed-Protokoll (weitgehend) folgt.
Discard ()	Löst das eingebettete Fenster aus dem Embedder.

Tabelle 12.5.2.1 : Ausgewählte Methoden der Klasse Embedder

12.5.3 Ereignisse

Von der Klasse *Embedder* werden Sie vor allem diese Ereignisse benutzen:

Ereignis	Beschreibung
Embed ()	Das Ereignis wird ausgelöst, wenn das Client-Fenster (erfolgreich) eingebettet wurde.
Error ()	Das Ereignis wird ausgelöst, wenn das Einbetten des Client-Fensters fehlgeschlagen ist.
Close ()	Das Ereignis wird ausgelöst, wenn die <i>eingebettete</i> Anwendung beendet wird.

Tabelle 12.5.3.1 : Methoden der Klasse Embedder

12.5.4 Projekt

Diese Hinweise sollten Sie beachten:

- Als notwendige Voraussetzung für ein erfolgreiches Einbetten gilt: Die einzubettende Anwendung *muss* dem XEmbed-Protokoll (weitgehend) folgen.
- Wenn die einzubettende Anwendung dem XEmbed-Protokoll nicht folgt, dann werden Fokus-Management, Fenster-Aktivierung und andere Funktionen nicht richtig funktionieren.
- Erkunden Sie sorgfältig die Arbeit mit dem Programm, dessen Fenster Sie in ein Gambas-Programm einbetten wollen.

Sie können das Fenster eines Programms in ein Gambas-Programm einbetten, das vor dem Start des Gambas-Programms bereits gestartet wurde oder zur Laufzeit des Gambas-Programms daraus gestartet wird. Der letzte Fall wird im Projekt verfolgt. Um ein Fenster eines anderen Programms in ein Gambas-Programm über die `Embed(x11id)`-Methode einzubetten, benötigen Sie dessen Fenster-X11ID. Unter Einbeziehung von Klassen der Komponente *gb.desktop* wurde im Projekt die folgende Strategie umgesetzt:

- Zuerst wird die Liste mit allen Top-Level-Fenstern aktualisiert → Zeile 14.
- Dann werden nur die X11-IDs der Top-Level-Fenster in einem Integer-Array gespeichert → Zeile 17, deren Fenster-Titel einem vorgegebenen Muster → Zeilen 9 und 16 entsprechen.
- Abschließend wird das letzte Element im Integer-Array → Zeile 21 als aktuelles Argument für die `Embed(..)`-Methode → Zeile 22 eingesetzt.

Hier sehen Sie einen Auszug aus dem Quelltext:

```
[1] ...
[2] Public Sub btnEmbed_Click()
[3]     Dim sPattern As String
[4]     Dim iX11Id As Integer
[5]     Dim DTWindow As DesktopWindow
[6]     Dim aX11List As New Integer[]
[7] ...
[8] ' Version mit vorgegebener URL
[9]     sPattern = "*Firefox*"
[10]    hProcess = Shell "firefox http://www.gambas-buch.de/dw/doku.php" For Read As "hProcess"
[11] ...
[12]    Wait fWaitPeriod
[13]
[14]    Desktop.Windows.Refresh ' Aktualisiert die Liste aller Top-Level-Fenster
[15]    For Each DTWindow In Desktop.Windows
[16]        If DTWindow.VisibleName LIKE sPattern Then
[17]            aX11List.Add(DTWindow.Id)
[18]        Endif
[19]    Next
[20]
[21]    Try iX11Id = aX11List[aX11List.Max]
[22]    Try embX11.Embed(iX11Id)
[23]    If Error Then Message.Warning(Error.Text)
[24]
[25] End ' btnEmbed_Click()
[26] ...
```

In der Zeile 9 wird das Suchmuster definiert, während in der Zeile 10 das Programm gestartet wird, dessen Fenster in das Gambas-Programm eingebettet werden soll. Selbstverständlich können Sie diesem Programm auch geeignete Parameter mitgeben, um beispielsweise im Webbrowser *Firefox* sofort

eine bestimmte Webseite anzuzeigen zulassen. Das funktioniert aber nur dann fehlerfrei, wenn kein weiteres Firefox-Fenster existiert oder in den Einstellungen des Firefox folgendes steht:

- Tabs**
- Statt neuem Fenster neuen Tab öffnen
 - Tabs erst laden, wenn sie angewählt werden
 - Tabs im Vordergrund öffnen



Abbildung 12.5.4.1: Firefox-Fenster im Gambas-Projekt-Programm

Bei der Erprobung des Projekts können Ihnen diese Hinweise helfen:

- Die Programme, deren Fenster Sie in ein Gambas-Programm einbetten wollen, müssen notwendigerweise auf dem System installiert sein. Sie sollten das Programm gut kennen.
- Das eingebettete Fenster ist vor dem Programm-Ende freizugeben.
- Die Wartezeit in der Zeile 12 ist notwendig, damit die Liste der Top-Level-Fenster auch das zur Laufzeit gestartete Programm enthält. Die Wartezeiten schwanken je nach Programm zwischen 0,2 Sekunden und 4 Sekunden und sind für jedes Programm experimentell zu ermitteln.
- So aktiviert man einige Sondertasten oder fängt die Maus ein: Auf den Reset-Knopf drücken! – dann funktioniert auch das ... (fast immer).
- Im Projektordner wurden eine Vielzahl von Dateien abgespeichert, damit Sie Bilder oder den Inhalt von PDF-Dateien anzeigen können. Einige Dateien enthalten erprobte Plot-Skripte.
- Achten Sie im Quelltext darauf, den Fenstertitel so zu ändern, damit beim Programmstart des Projekt-Programms erkennbar ist, welches Programm-Fenster Sie einbetten wollen.
- Sie können auch das (Haupt-)Fenster eines Gambas-Programms → Kapitel 12.3.4 in ein Gambas-Programm einbetten:



Abbildung 12.5.4.2: Gambas-Programm-Fenster – eingebettet im Gambas-Projekt-Programm

Hier sehen Sie die Liste der erprobten Programme, deren Fenster erfolgreich in das Gambas-Programm eingebettet werden konnten.

```
sPattern = "*VLC*"
hProcess = Shell "vlc" For Read As "hProcess"

sPattern = "*Bluefish*"
hProcess = Shell "bluefish" For Read As "hProcess"

sPattern = "*Dokument*"
hProcess = Shell "abiword" For Read As "hProcess"

' Allgemeine Version
sPattern = "*Firefox*"
hProcess = Shell "firefox http://www.gambas-buch.de/dw/doku.php" For Read As "hProcess"

' Version mit vorgegebener URL
sPattern = "*Firefox*"
hProcess = Shell "firefox http://www.gambas-buch.de/dw/doku.php" For Read As "hProcess"

sPattern = "hans@linux: ~"
hProcess = Shell "gnome-terminal" For Read As "hProcess"

sPattern = "*Gambas 3*"
hProcess = Shell "gambas3" For Read As "hProcess"

sPattern = "*Bild-Betrachter*"
hProcess = Shell "gbr3 " & Application.Path & / "GPE/gpe.gambas" For Read As "hProcess"

' Allgemeine Version
sPattern = "*Dokument*"
hProcess = Shell "evince --fullscreen"

' Version mit vorgegebener PDF-Datei
sPattern = "*k25.1.13.pdf*"
hProcess = Shell "evince --fullscreen " & Application.Path & / "Text/k25.1.13.pdf" For Read As "hProcess"

' Allgemeine Version
sPattern = "*LibreOffice Writer*"
hProcess = Shell "soffice --writer --nologo"

' Version mit vorgegebener ODT-Datei
sPattern = "*LibreOffice Writer*"
hProcess = Shell "soffice --writer --nologo " & Application.Path & / "Text/k25.1.13.odt" For Read As "hProcess"

sPattern = "*Gnuplot*"
hProcess = Shell "gnuplot -persist " & Application.Path & / "GnuPlotScripts/start.plot" For Read As "hProcess"
```

Bei den Programm-Tests vom Autor Ingo Beckert funktionierte selbst das fehlerfrei – ein Windows-Programm aus einem Gambas-Programm heraus über die Laufzeitumgebung *Wine* starten:

```
sPattern = "*Ole*"
hProcess = Shell "wine oleview" For Read As "hProcess"
```