

12.3.1 Workspace

Die Klasse `Workspace` (gb.form.mdi) implementiert einen Arbeitsbereich, in den Sie Top-Level-Fenster einfügen können. Um dem Arbeitsbereich neue Fenster hinzuzufügen, müssen diese zuerst instanziiert werden. Verwenden Sie die `Add()`-Methode, um ein Top-Level-Fenster in einen `Workspace` einzufügen. Sie können zwischen den Fenstern im Arbeitsbereich wechseln, indem Sie die Register-Karte wechseln. Die Begriffe `Arbeitsbereich`, `Workspace` und `Register` werden in diesem Kapitel synonym verwendet.

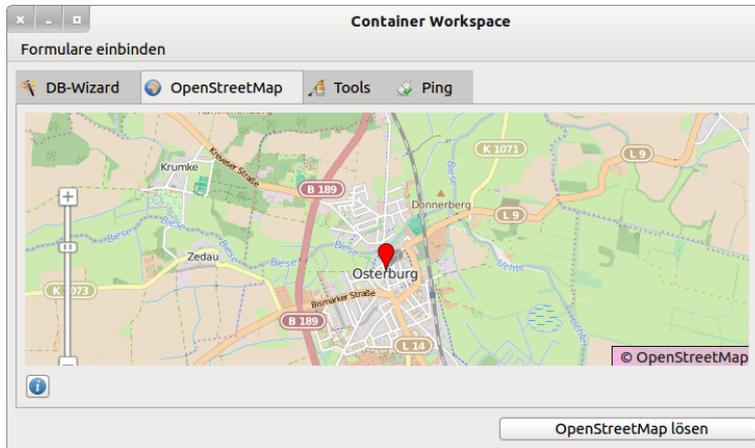


Abbildung 12.3.1.1: Vier Top-Level-Fenster in einem Arbeitsbereich

12.3.1.1 Eigenschaften

Die Klasse `Workspace` verfügt u.a. über diese ausgewählten Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
<code>ActiveWindow</code>	<code>Window</code>	Setzt das aktive Fenster oder gibt das aktive Fenster im <code>Workspace</code> zurück.
<code>Background</code>	<code>Integer</code>	Setzt die Hintergrundfarbe für den Fall, dass sich kein Fenster im <code>Workspace</code> befindet oder gibt den Farbwert zurück.
<code>Image</code>	<code>Image</code>	Gibt das Hintergrundbild für den <code>Workspace</code> zurück oder setzt das Bild (→ ab Version 3.7). Das Hintergrundbild wird nur dann angezeigt, wenn sich kein Fenster im <code>Workspace</code> befindet. Als Standard wird ein dunkler Hintergrund angezeigt.
<code>Count</code>	<code>Integer</code>	Gibt die Anzahl der Fenstern an, die vom <code>Workspace</code> verwaltet werden.
<code>Windows</code>	<code>Window[]</code>	Gibt eine Liste aller Fenster zurück, die im <code>Workspace</code> verankert sind.
<code>Children</code>	<code>.Container.Children</code>	Die virtuelle Klasse <code>.Container.Children</code> repräsentiert ein Top-Level-Fenster in einem Arbeitsbereich. Sie können einen Arbeitsbereich wie ein Array benutzen, um so ein Objekt der Klasse <code>.Container.Children</code> (also ein Top-Level-Fenster) über seinen Index zu erhalten.

Tabelle 12.3.1.1.1 : Eigenschaften der Klasse `Workspace`

12.3.1.2 Methoden

In der folgenden Tabelle werden einige Methoden der Klasse `Workspace` beschrieben:

Methode	Beschreibung
<code>Add(hWindow As Window) As Window</code>	Fügt ein Top-Level-Fenster in einen Arbeitsbereich ein und zeichnet diesen dann neu.
<code>Detach(hWindow As Window)</code>	Löst ein Fenster aus einem <code>Workspace</code> und macht es zu einem (freien) Top-Level-Fenster. Sie können das Fenster jederzeit wieder mit der <code>Attach()</code> -Methode im <code>Workspace</code> verankern.

Methode	Beschreibung
Attach(hWindow As Window)	Verankert ein mit der Detach()-Methode freigestelltes Top-Level-Fenster wieder in einem Workspace.
Exist(hWindow As Window) As Boolean	Die Funktion gibt <i>True</i> zurück, wenn das als Argument übergebene Fenster vom Workspace verwaltet wird.
Lock()	Sperrt den Arbeitsbereich. Die Register-Karten werden <u>nicht</u> aktualisiert, wenn der Arbeitsbereich gesperrt ist.
Unlock()	Entriegelt den Arbeitsbereich. Sobald der Arbeitsbereich freigegeben ist, werden die Register-Karten wieder aktualisiert.

Tabelle 12.3.1.2.1 : Methoden der Klasse Workspace

12.3.1.3 Ereignisse

Von den Events der Klasse Workspace werden Ihnen nur diese vier vorgestellt:

Ereignis	Beschreibung
Activate()	Das Ereignis wird ausgelöst, wenn das aktive Fenster geändert wurde.
Attach(Window As Window)	Das Ereignis wird ausgelöst, wenn ein Fenster wieder zum Arbeitsbereich hinzugefügt wurde.
Detach(Window As Window)	Das Ereignis wird ausgelöst, wenn ein Fenster aus dem Arbeitsbereich herausgelöst wurde.
Close(Window As Window)	Das Ereignis wird ausgelöst, wenn eine Register-Karte geschlossen wird.

Tabelle 12.3.1.3.1 : Ausgewählte Ereignisse der Klasse Workspace

Der Arbeitsbereich verfügt über ein Kontext-Menü – aufrufbar in der Register-Karten-Reiter-Zeile – mit dem Sie folgende Features realisieren können:

- Anzeige eines bestimmten Fensters → Fensterliste.
- Sortierung der Register-Karten-Reiter – jedoch nur in einer Sortierreihenfolge.
- Löschen aller Register-Karten-Reiter.
- Löschen des linken oder rechten Register-Karten-Reiters – in Abhängigkeit vom aktiven Fenster *oder* löschen aller Register-Karten-Reiter – außer dem aktiven Reiter.
- Ein Fenster, das von einem Workspace verwaltet wird – also dort verankert ist – aus dem Arbeitsbereich herauslösen *oder* ein Top-Level-Fenster, das von einem Workspace verwaltet wird, wieder in den Arbeitsbereich einfügen.

Hinweise:

- Wenn Sie mit der Maus über einem Register-Karten-Reiter im Arbeitsbereich stehen, so können Sie mit dem Mousrad (vorsichtig) durch die verankerten Programme navigieren.
- Die Register-Karten-Reiter können Sie auch sortieren, indem Sie mit gedrückter linker Maustaste den markierten Reiter an die gewünschte neue Position schieben.
- Wenn Sie ein Top-Level-Fenster im Register-Karten-Reiter schließen, dann wird das Fenster nicht mehr vom Workspace verwaltet.

12.3.1.4 Projekt

Das Projekt stellt u.a. einen Workspace zur Verfügung, in den 4 Top-Level-Fenster verankert werden, hinter denen 4 externe Programme stecken. Es hat sich als vorteilhaft erwiesen, alle einzufügenden Programme in eigenen Verzeichnissen im Projektordner zu speichern. Zusätzlich verfügt das Hauptprogramm über ein Menü, mit dem Sie ein Fenster wieder zur Arbeitsfläche hinzufügen können, wenn Sie es aus der Verwaltung des Workspaces entfernt hatten. Ein spezielles Fenster im Arbeitsbereich (OpenStreetMap) können Sie aus dem Arbeitsbereich lösen und wieder hinzufügen. Die 4 Top-Level-Fenster im vorgestellten Projekt gehören inhaltlich nicht zusammen. Im produktiven Umfeld würde man Programme wählen, mit denen eine Aufgabe zu lösen ist. Gut würde zum Beispiel ein Programm zur Berechnung von Wertetabellen zu einem Funktionsplotter passen.

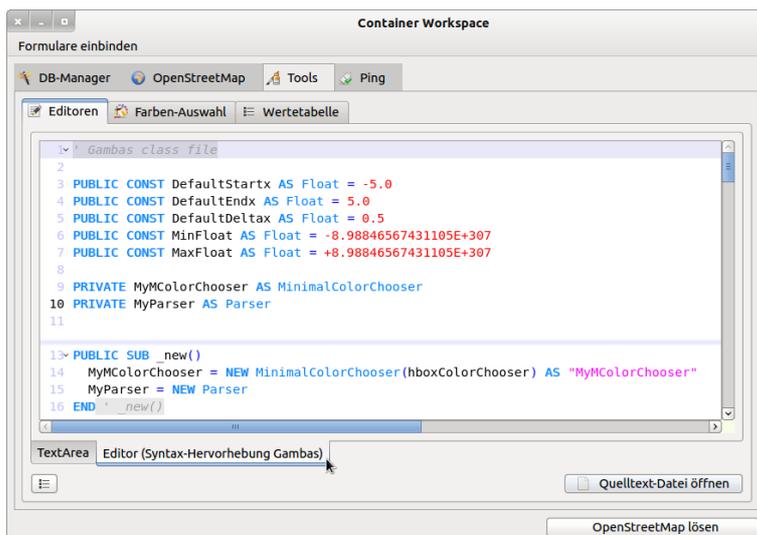


Abbildung 12.3.1.4.1: Hauptprogramm mit dem Arbeitsbereich (4 Fenster)

Im Quelltext der einzelnen externen Programme sollten Sie die Eigenschaften *Caption* und *Icon* mit geeigneten Werten setzen. Beachten Sie: Das Programm-Icon wird auch als Icon für den entsprechenden Register-Karten-Reiter im Workspace übernommen. Leider wird das Icon – im Gegensatz zum Programm-Icon – nicht auf die erforderliche Größe skaliert. Wählen Sie deshalb bereits ein Programm-Icon mit einer Größe von 16x16 Pixeln → Zeile 4 im folgenden Quelltext-Ausschnitt:

```

[1] Public Sub Form_Open()
[2]   FGeoMap.Center
[3]   FGeoMap.Caption = " OpenStreetMap"
[4]   FGeoMap.Icon = Picture["icon:/16/internet"]
[5]   ShowMap()
[6] End ' Form_Open()
  
```

Der Quelltext ist überschaubar und birgt keine Überraschungen:

```

[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]
[5]   FMain.Center
[6]   FMain.Resizable = True
[7]   FMain.Caption = "Container Workspace"
[8]
[9]   Workspace1.Image = Image.Load("Symbols/intro.jpg")
[10]  Workspace1.Add(FDBWizard)
[11]  Workspace1.Add(FGeoMap)
[12]  Workspace1.Add(FTools)
[13]  Workspace1.Add(FPing)
[14]  Workspace1.ActiveWindow = FTools
[15]
[16]  btnInOut.Text = " OpenStreetMap lösen"
[17]
[18] End ' Form_Open
[19]
[20] Public Sub Menu1_Click()
[21]   If Not Workspace1.Exist(FDBWizard) Then Try Workspace1.Add(FDBWizard)
[22] End ' Menu1
[23]
[24] Public Sub Menu2_Click()
[25]   If Not Workspace1.Exist(FGeoMap) Then Workspace1.Add(FGeoMap)
[26]   btnInOut.Visible = True
[27]   Print Last.Text
[28] End ' Menu2
[29]
[30] Public Sub Menu3_Click()
[31]   If Not Workspace1.Exist(FTools) Then Workspace1.Add(FTools)
[32] End ' Menu3
[33]
[34] Public Sub Menu4_Click()
[35]   If Not Workspace1.Exist(FPing) Then Workspace1.Add(FPing)
[36]   Print Last.Text
[37] End ' Menu4
  
```

```
[38]
[39] Public Sub btnInOut_Click()
[40]
[41]     If Workspacel.Exist(FGeoMap)
[42]         If Not Workspacel.Tag Then
[43]             Workspacel.Detach(FGeoMap)
[44]             Workspacel.Tag = "A"
[45]             btnInOut.Text = " OpenStreetMap verankern"
[46]         Else
[47]             Workspacel.Tag = ""
[48]             Workspacel.Attach(FGeoMap)
[49]             btnInOut.Text = " OpenStreetMap lösen"
[50]             Workspacel.ActiveWindow = FGeoMap
[51]         Endif
[52]     Endif
[53]
[54] End ' btnInOut_Click()
[55]
[56] Public Sub Workspacel_Activate()
[57]     Dim sProgramName As String
[58]     ' Zur Kontrolle:
[59]     If Workspacel.Count > 1 Then Print Workspacel.ActiveWindow.Name
[60] End ' Workspacel_Activate()
[61]
[62] Public Sub Workspacel_Close(hWindow As Window)
[63]     ' Zur Kontrolle:
[64]     Print "OK 1"
[65]     If hWindow = FGeoMap.Window Then btnInOut.Visible = False
[66] End ' Workspacel_Close(hWindow As Window)
```

Hinweise:

- Als Alternative für das Hintergrundbild – bei fehlendem Inhalt für den Arbeitsbereich – in der Zeile 9 könnte beispielsweise das genutzt werden: `Workspace1.Background = Color.LightGray`.
- In den Zeilen 20 bis 37 werden die Aktionen in den 4 Menü-Einträgen festgelegt.
- Das Herauslösen für ein spezielles Fenster aus dem Arbeitsbereich und das erneute Einfügen in den Arbeitsbereich wird in den Zeilen 39 bis 54 definiert.
- Die zwei letzten Prozeduren halfen bei der Entwicklung des Programms und können entfallen.
- Bei der Erprobung des Projektes nutzen Sie bitte im Programm 'DB-Manager' (→ Menü) als Datenbank-Management-System *SQLite3* und die mitgelieferte Datenbank 'kontakte.sqlite'.