

12.3.3 Action

Die beiden Klassen *Action (gb.qt4)* und *Action (gb.form.mdi)* ermöglichen es Ihnen, die Aktionen zu verwalten, die Sie in der Action-Eigenschaft von vielen Komponenten definieren können. Die Klasse *Action (gb.qt4)* verhält sich wie ein ReadOnly-Array. Mithilfe der Klasse *Action (gb.qt4)* kann man dann auf all diese so ausgezeichneten Steuerelemente synchron zugreifen. Die folgenden Komponenten verfügen über die Eigenschaft 'Action':

```
Button, ButtonBox, CheckBox, ColorButton, ColorChooser, ColumnView, ComboBox, Container, Control, DateBox,
DateChooser, Dial, DirChooser, DirView, Drag, DrawingArea, Editor, Embedder, Expander, FMain, FileChooser,
FileProperties, FileView, FontChooser, Form, Frame, GridView, HBox, HPanel, HSplit, IconPanel, IconView,
ImageView, LCDLabel, LCDNumber, Label, ListBox, ListContainer, ListView, MaskBox, Menu, MenuButton, Movie-
Box, Panel, PictureBox, ProgressBar, RadioButton, ScrollArea, ScrollBar, ScrollView, Separator, SidePanel,
Slider, SliderBox, SpinBox, TabPanel, TabStrip, TableView, TextArea, TextBox, TextEdit, TextLabel, Toggle-
Button, ToolButton, ToolPanel, TreeView, UserContainer, UserControl, VBox, VPanel, VSplit, ValueBox, Win-
dow, Wizard, _IconPanelContainer, _Split, _TabPanelContainer, _TreeView, _WizardContainer.
```

Beispiel:

Ein *ToolButtonS* (auf einer *Toolbar*) und ein Menü-Eintrag sind beide zum Speichern einer Datei vorge-
sehen. Beiden Komponenten würde man daher als Action-String in der IDE zum Beispiel "SaveFile"
zuweisen, um ihre Aktionen zu synchronisieren. Nur wenn die geöffnete Datei modifiziert wurde, dann
soll die Speicherfunktion aktiviert werden, die beim Öffnen der Datei für beide Komponenten abge-
schaltet wurde. Sowohl der entsprechende *ToolButtonS* als auch der assoziierte Menü-Eintrag sind
noch sichtbar – aber ausgegraut! Hier die beiden passenden Quelltextausschnitte:

```
Action["SaveFile"].Enabled = False      ' Speichern für ToolButton und Menü-Eintrag de-aktivieren
Action["SaveFile"].Enabled = True       ' Speichern für ToolButton und Menü-Eintrag aktivieren
```

Es hat sich als vorteilhaft erwiesen, die Action-Eigenschaft den entsprechenden Komponenten und
Menüeinträgen *direkt* in der IDE zuzuweisen, weil Sie nur dann sowohl den Dialog zur Festlegung von
ShortCuts durch den Benutzer aufrufen können als auch die Liste aller Aktionen auslesen können,
wenn Sie die Klasse *Action (gb.form.mdi)* verwenden. Wollen Sie darauf verzichten, dann definieren
Sie die Aktionen für ausgewählte Komponenten im *Quelltext* exemplarisch so:

```
mnu12SaveFile.Action = "SaveFile"
toolbSaveFile.Action = "SaveFile"
```

Im versteckten Projekt-Ordner *.action* befindet sich die Gambas-Action-Datei *formulardname.action*, die
zum Beispiel Angaben zum Namen der Aktion (Action-String), zum anzuzeigenden Text oder zum ge-
setzten ShortCut enthält:

```
# Gambas Action File 3.0
{ Actions
  { Action SaveFile
    Text = "Speichern"
    Shortcut = "CTRL+S"
    Picture = "icon:/16/save"
  }
  { Action HelpMe
    Text = "Programm-Hilfe"
    Shortcut = "F1"
    Picture = "icon:/16/help"
  }
  ...
}
```

12.3.3.1 Eigenschaften der Klasse Action (gb.qt4)

Die Eigenschaften der Klasse *Action (gb.qt4)* finden Sie in der folgenden Tabelle:

Action	Datentyp	Default	Beschreibung
Controls	Object[]	-	Gibt eine Liste aller assoziierten Komponenten aus, die mit der definierten Aktion verbunden sind.
Enabled	Boolean	True	Aktiviert oder deaktiviert alle Komponenten, die mit der Aktion verbunden sind.

Action	Datentyp	Default	Beschreibung
Picture	Picture	Null	Ermittelt oder legt fest, welches Icon in allen Komponenten angezeigt wird, die mit der Aktion verbunden sind.
Shortcut	String	Null	Ermittelt oder legt fest, welches Tastaturkürzel allen Komponenten zugewiesen wird, die mit der Aktion verbunden sind.
Text	String	Null	Ermittelt oder legt fest, welcher Text bei allen Komponenten angezeigt wird, die mit der Aktion verbunden sind.
ToolTip	String	Null	Ermittelt oder legt fest, welcher ToolTip bei allen Komponenten angezeigt wird, die mit der Aktion verbunden sind.
Visible	Boolean	True	Versteckt oder zeigt alle Komponenten, die mit der Aktion verbunden sind.
Value	Boolean	True	Ermittelt oder legt den Toogle-Wert fest für alle Komponenten, die mit der Aktion verbunden sind.

Tabelle 12.3.3.1.1: Action-Eigenschaften (gb.qt4)

Bitte beachten Sie, dass Sie nur die Eigenschaften setzen können, die für alle assoziierten Komponenten existieren.

```

mnu15Print.Action = "PrintImage"
toolbPrintImage.Action = "PrintImage"
...
Action["PrintImage"].ToolTip = "Bild-Datei drucken"

```

Die letzte Anweisung etwa erzeugt einen Fehler und die Fehlermeldung nennt auch den Grund: *Unbekanntes Symbol 'ToolTip' in Klasse 'Menu' in FMain:31; Eigenschaft existiert nicht in der Klasse Menu.*

Die Wirkung des Wertes der Eigenschaft *Action.Value* hängt von der betreffenden Komponente ab:

- Wenn die Komponente ein echtes Toggle-Steuerelement wie ein ToggleButton oder eine Tool-Button mit *gesetzter* Toggle-Eigenschaft ist, wird sein Toogle-Wert gesetzt oder nicht gesetzt.
- Handelt es sich aber zum Beispiel um ein SidePanel, eine Symbolleiste oder ein Fenster, dann wird dagegen die Visible-Eigenschaft gesetzt.

12.3.3.2 Der Event-Handler Action_Activate

Um zu wissen, wann eine Aktion ausgelöst wird, müssen Sie den Event-Handler "*Action_Activate*" im Formular definieren. Dieser Event-Handler übernimmt eine Zeichenkette als Argument, welches die ausgelöste Aktion bezeichnet.

```

Public Sub Action_Activate(sActionString As String) As Boolean

    Select Case sActionString
        Case "ImageOpen"
            ImageOpen
        Case "HelpMe"
            FHelp.Show
        Case "PrintFile"
            ' Prozedur für das Ausdrucken bearbeiteten Datei
        Case "FormClose"
            FHelp.Close
            FMain.Close
    End Select ' sActionString

End ' Action_Activate(..)

```

12.3.3.3 Eigenschaften und Methoden Klasse Action (gb.form.mdi)

Die Klasse Action (gb.form.mdi) verfügt nur über eine Eigenschaft und eine Methode:

- Die Klasse besitzt die Eigenschaft *List (Daten-Typ Array)*, deren Wert Sie nur auslesen können. In der Liste werden alle gefundenen Aktionen geordnet aufgelistet.
- Der Aufruf der Methode *Configure* startet einen Dialog, in dem der Benutzer eigene ShortCuts festlegen kann, die sofort projektweit gelten und vorhandene Einstellungen überschreiben.

12.3.3.4 Projekt

In vielen Programmen besteht die Möglichkeit, die interaktive Benutzerführung über eine Menüleiste durch eine Symbolleiste zu erweitern. Über die *Symbolleiste* wird ein Schnellzugriff auf die *wichtigsten Teilprogramme* des Anwendungsprogramms zur Verfügung gestellt.

Der Einsatz einer Symbolleiste neben einem Menü geht von der Idee aus, dass die Auswahl eines bestimmten Menüpunktes den gleichen (Teil-)Programmaufruf bewirkt wie ein Maus-Klick auf den korrespondierenden ToolButton. Von entscheidender Bedeutung für die angestrebte Wirkung ist die *Verknüpfung* einzelner Menüs und deren Funktion mit den ausgewählten ToolButton. Im Projekt wird zur *Synchronisation* zwischen Menü und den assoziierten ToolButton auf einer Symbolleiste die Klasse *Action* eingesetzt. Für das Projekt mit der sehr überschaubaren Menüleiste und den wenigen ToolButton in der Symbolleiste ist das nicht zwingend erforderlich, fokussiert aber alle von Menü und ToolButton ausgelösten Programmaufrufe in einer Prozedur! Ein Vorteil den Sie zu schätzen wissen, wenn Sie eine Menüleiste mit sehr vielen Menüs sowie Untermenüs und eine Symbolleiste in einer umfangreichen Menü-Struktur einsetzen wollen.

Zur Umsetzung der Idee ist es von Nutzen, alle Menüs und (Tool-)Button in einer Tabelle zu erfassen, passende Bezeichner festzulegen sowie die assoziierten Komponenten festzulegen:

MCaption	UMCaption	Name	Picture	FT	Action-String	S	Komponente
Bild-Datei		mnuMenu1					
	Öffnen...	mnu11OpenImage			OpenImage	●	tbOpen
		mnu12Space					
	Drucken	mnu13PrintImage			PrintImage	●	tbPrint
	Beenden	mnu14CloseForm			CloseForm	●	tbCloseForm
Konfiguration		mnuMenu2					
	Action-Konfiguration	mnu21ConfigAction			ConfigAction	●	tbConfigAction
	Action-List	mnu22ListAction	view-detail	F11			
Hilfe		mnuMenu3					
	Programm-Hilfe	mnu31Help			HelpMe	●	tbHelpMe
	Hilfe im Internet	mnu32HelpNet	internet	F12			

Tabelle 12.3.3.4.1: Eintragungen im Menü-Editor (S → Synchronisation, FT → Funktionstaste)

Im vorgestellten Projekt wird das Menü – als Zusammenstellung vieler Menüs – nach den Angaben in der Tabelle 20.6.4.1 mit dem Menü-Editor angelegt und konfiguriert:

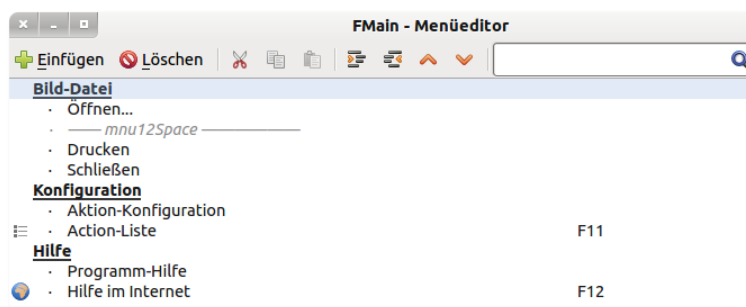


Abbildung 12.3.3.4.1: Menü-Editor (Ausschnitt)

- Bei 5 Menüs wird unter der Eigenschaft 'Action' der passende Action-String aus der Tabelle 20.6.4.1 eingetragen und die qualifizierten Werte für die Eigenschaften 'Visible' sowie 'Enabled' gesetzt.
- Für die 5 assoziierten Komponenten (ToolButton) werden in der IDE zur Entwicklungszeit auch die Action-Strings aus der Tabelle 20.6.4.1 eingetragen. Wichtig ist nur die Gleichheit der Bezeichner für die Action-Eigenschaft für die assoziierten Menüs und ToolButton!

- Eingesetzt wird die Komponente ToolBar als Container für die horizontale Symbolleiste, in der 6 ToolButton nebeneinander platziert werden. Zur *Entwicklungszeit* werden den 5 assoziierten Komponenten *gleiche* Icon (16px) aus dem Bestand von Gambas zugewiesen (damit man die ToolButton erkennt) und den Menüs im Editor keine! Erst zur Laufzeit werden sie mit geeigneten Icon ausgestattet.
- Bis auf die Höhe (32px) und eine zum Formular passende Weite der Toolbar werden keine weiteren Eigenschaften geändert. Die Anordnung der 5/6 ToolButton wird zur Laufzeit automatisch mit festem Abstand zueinander vorgenommen:



Abbildung 12.3.3.4.2: Menü und Symbolleiste

Der Quelltext wird komplett angegeben und in wesentlichen Teilen kommentiert:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]
[5]     FMain.Center
[6]     FMain.Resizable = False
[7]
[8]     FMain.Text = "Synchronisation Menü und Symbolleiste"
[9]     PictureBox1.Border = Border.Sunken
[10]    PictureBox1.Picture = Picture["blume.jpg"]
[11]    lblPictureFileName.Text = "Blumen im Allgäu - Augentrost"
[12]
[13]    Action["OpenImage"].Enabled = True
[14]    Action["OpenImage"].Picture = Picture["icon:/16/open"]
[15]    Action["OpenImage"].Shortcut = "Ctrl+O"
[16]    Action["OpenImage"].Visible = True
[17]
[18]    Action["PrintImage"].Enabled = True
[19]    Action["PrintImage"].Picture = Picture["icon:/16/print"]
[20]    Action["PrintImage"].Shortcut = "Ctrl+P"
[21]    Action["PrintImage"].Visible = False ' (!)
[22]
[23]    Action["CloseForm"].Enabled = True
[24]    Action["CloseForm"].Picture = Picture["icon:/16/quit"]
[25]    Action["CloseForm"].Shortcut = "Ctrl+Q"
[26]    Action["CloseForm"].Visible = True
[27]    Action["CloseForm"].Text = " Beenden"
[28]
[29]    Action["ConfigAction"].Enabled = True
[30]    Action["ConfigAction"].Picture = Picture["icon:/16/access"]
[31]    Action["ConfigAction"].Shortcut = "F6"
[32]    Action["ConfigAction"].Visible = True
[33]
[34]    Action["HelpMe"].Enabled = True
[35]    Action["HelpMe"].Picture = Picture["icon:/16/help"]
[36]    Action["HelpMe"].Shortcut = "F1"
[37]    Action["HelpMe"].Visible = True
[38]
[39]    tbSyncList.Tooltip = "Liste der" & gb.NewLine & "synchronisierten Komponenten"
[40]
[41] End ' Form_Open()
[42]
[43] Public Sub Action_Activate(sActionString As String) As Boolean
[44]
[45]     Select Case sActionString
```

```

[46] Case "OpenImage"
[47]     OpenImage
[48] Case "PrintImage"
[49]     PrintImage
[50]     Action["PrintImage"].Enabled = False
[51] Case "CloseForm"
[52]     Form_Close()
[53] Case "ConfigAction"
[54]     Action.Configure
[55] Case "HelpMe"
[56]     FHelp.Show
[57] End Select ' sActionString
[58]
[59] End ' Action_Activate(..)
[60]
[61] Public Sub OpenImage()
[62]
[63]     Dialog.Title = "Importieren Sie eine Bild-Datei!"
[64]     Dialog.Filter = ["*.png", "Bild-Dateien"]
[65]
[66]     If Dialog.OpenFile(False) = True Then ' Multiselect=False (Standard)
[67]         Message.Info("Das Öffnen der Bild-Datei wurde abgebrochen!")
[68]         Return ' Cancel-Button gedrückt
[69]     Else
[70]         Try PictureBox1.Picture = Picture.Load(Dialog.Path)
[71]         If Error Then
[72]             Message.Error("Fehler beim Öffnen der Bild-Datei")
[73]             Return
[74]         Else
[75]             lblPictureFileName.Alignment = Align.Left
[76]             lblPictureFileName.Text = "Dateiname: " & File.Name(Dialog.Path)
[77]             Action["PrintImage"].Visible = True
[78]             Action["PrintImage"].Enabled = True
[79]         Endif ' ERROR ?
[80]     Endif ' Dialog.OpenFile ?
[81]
[82] End ' OpenImage()
[83]
[84] Public Sub PrintImage()
[85]     Message.Info("Druck-Routine hier einfügen!\nDas Bild wird gedruckt.")
[86] End ' PrintImage()
[87]
[88] ' Hinweis: Der Menü-Name ist gleichzeitig auch der Bezeichner für den Event-Handler
[89] Public Sub mnu32HelpNet_Click()
[90]     Desktop.Open("http://www.gambas-buch.de/dw/doku.php?id=k20:k20.6:start")
[91] End ' mnu32HelpNet_Click()
[92]
[93] Public Sub mnu22ListAction_Click()
[94]     Dim iCount As Integer
[95]     Dim sMessage, sTrennLinie, sSpace, sSpace2 As String
[96]
[97]     sTrennLinie = "-----"
[98]     sSpace = " » "
[99]     sSpace2 = " * "
[100]     sMessage = sTrennLinie & sTrennLinie & gb.NewLine
[101]     sMessage &= "Liste aller Action-Strings" & gb.NewLine
[102]     sMessage &= sTrennLinie & sTrennLinie & gb.NewLine & gb.NewLine
[103]
[104]     For iCount = 1 To Action.List.Count
[105]         If Action.List[iCount - 1] Like "[A-Z]*" Then
[106]             sMessage &= iCount & sSpace & Action.List[iCount - 1] & gb.NewLine
[107]         Else
[108]             sMessage &= iCount & sSpace2 & Action.List[iCount - 1] & gb.NewLine
[109]         Endif ' Action.List[iCount - 1] Like "[A-Z]*" ?
[110]     Next ' iCount
[111]
[112]     Message.Title = gb.tab
[113]     Message.Info(sMessage)
[114]
[115] End ' mnu22ListAction_Click()
[116]
[117] Public Sub tbSyncList_Click()
[118]     Dim iCount, k As Integer = 1
[119]     Dim sActionString, sMessage, sTrennLinie, sSpace As String
[120]
[121]     sTrennLinie = "-----"
[122]     sSpace = " » "
[123]     sMessage = sTrennLinie & sTrennLinie & gb.NewLine
[124]     sMessage &= "Liste der synchronisierten Komponenten" & gb.NewLine
[125]     sMessage &= sTrennLinie & sTrennLinie & gb.NewLine & gb.NewLine
[126]
[127]     For Each sActionString In Action.List
[128]         If Action[sActionString].Controls.Count >= 2 Then
[129]             For k = 0 To Action[sActionString].Controls.Max

```

```
[130]         If k = 0 Then sMessage &= "Action = " & sActionString & gb.NewLine
[131]             sMessage &= sSpace & Action[sActionString].Controls[k].Name & gb.NewLine
[132]         Next ' k
[133]     Endif ' Action[sActionString].Controls.Count >= 2 ?
[134] Next ' sActionString
[135]
[136] Message.Title = gb.Tab
[137] Message.Info(sMessage)
[138]
[139] End ' tbSyncList_Click()
[140]
[141] Public Sub Form_Close()
[142]     FHelp.Close
[143]     FMain.Close
[144] End ' Form_Close()
```

Kommentare:

- In den Zeilen 13 bis 37 werden ausgewählte Action-Eigenschaften für die 5 assoziierten Komponenten gesetzt.
- Der Zeilen 43 bis 57 tragen die Festlegungen im Event-Handler "Action_Activate".
- Die De-Aktivierung der Druck-Funktion finden Sie in der Zeile 50, wenn der Druck des Bildes erfolgreich war. In den Zeilen 18 und 21 wurde diese Funktion zum Programmstart de-aktiviert.
- Nach dem Laden eines Bildes könnte es gedruckt werden. Daher wird die Druckfunktion in den Zeilen 77 und 78 aktiviert.
- In den Zeilen 104 bis 110 werden alle Action-Strings ausgelesen und zur Anzeige aufbereitet. Da alle selbst definierten Action-Strings mit einem großen Buchstaben beginnen – im Gegensatz zu den von Gambas vor-definierten – erhalten diese ein »-Zeichen.

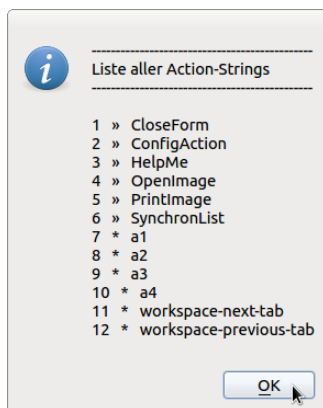


Abbildung 12.3.3.4.3: Liste alle Action-Strings

- Die Aufbereitung der Zusammenstellung aller assoziierten Komponenten in den Zeilen 127 bis 134 gestaltet sich etwas umfangreicher, da in die Zusammenstellung nur die Komponenten aufgenommen werden, die mindestens 2 assoziierte Komponenten enthalten. Den Action-String 'SynchronList' gibt es nur einmal – für die Komponente tbSyncList (→ Abbildung 20.6.4.3 und Projekt MenuAction) und daher fehlt die Komponente tbSyncList in der Abbildung 20.6.4.4.

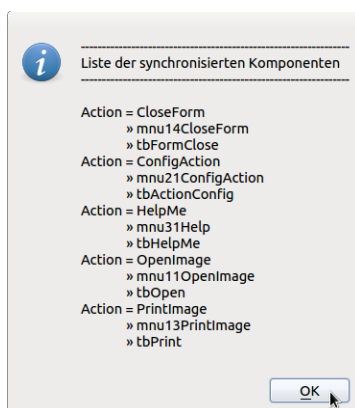


Abbildung 12.3.3.4.4: Liste aller assoziierten Komponenten

Hier sehen Sie einen Quelltext-Ausschnitt für eine (Teil-)Synchronisation von Menüpunkt "Öffnen..." und ToolButton *tbOpen* den Sie verwenden müssten, wenn Sie die Klasse Action nicht verwenden:

```
Public Sub mnullOpenImage_Click()
    ImageOpen()
    mnul3PrintImage.Enabled = True
    tbPrint.Enabled = True
End ' mnullOpenImage_Click()

Public Sub tbOpen_Click()
    ImageOpen()
    mnul3PrintImage.Enabled = True
    tbPrint.Enabled = True
End ' tbOpen_Click()
```

Dagegen sieht der entsprechende Quelltext mit gleicher Wirkung in den Zeilen 48 und 50 in der Prozedur *Action_Activate(sActionString As String)* mehr als bescheiden aus.

Unabhängig von der gewählten Vorgehensweise bei der Synchronisation von Menü und ToolButton in der ToolBar müssen Sie alle Prozeduren programmieren, um dem Programm die gewünschte Funktionalität zu verleihen. Zum Programmstart finden Sie nur 5 ToolButton auf der Symbolleiste, weil die Druckfunktion de-aktiviert wurde. Den entsprechenden ToolButton finden Sie in der Symbolleiste nach dem erfolgreichen Öffnen einer Bilddatei:

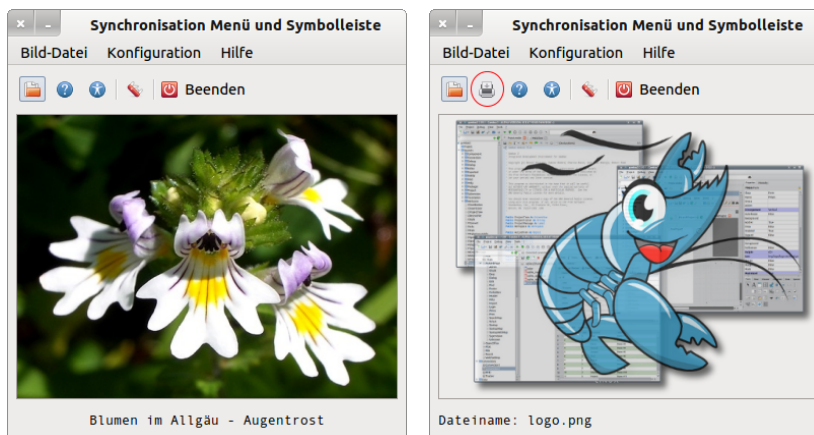


Abbildung 12.3.3.4.5: Symbolleisten zur Laufzeit

Die Konfiguration der ShortCuts können Sie mit F6, mit einem Klick auf den entsprechenden Menü-Eintrag oder ToolButton anschieben:

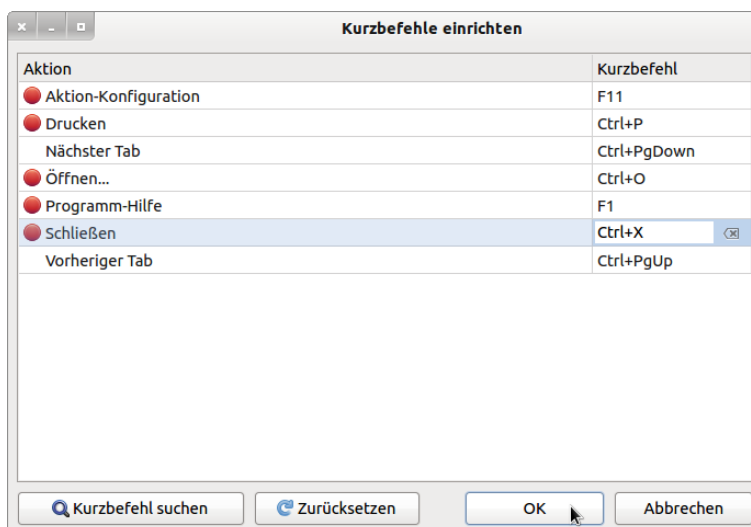


Abbildung 12.3.3.4.6: Konfiguration Kurzbefehle