

## Table of Contents

Working with Menus in RwxWidgets ..... 1

# Working with Menus in RwxWidgets

We start by loading the RwxWidgets package.

```
library(RwxWidgets)
wxInit()
wxNO_BORDER = wxBORDER_NONE
```

And then we create a top-level window and set its menu bar. We explicitly create the menu bar and then associate it with the frame.

```
f = RFrame("Menu example", size = c(300, 300))

menuBar = wxMenuBar()
f$SetMenuBar(menuBar)
```

Now we can start creating the actual menus and their contents. We start with a simple menu and adding an item with the label "Entry 1".

```
menu = wxMenu()
i = menu$Append(wxID_ANY, "Entry 1")
```

We can register a callback on an individual item. Note that we call this method on a wxWindow object, e.g. f, rather than on the wxMenuItem itself. The identifier for the event is wxEVT\_COMMAND\_MENU\_SELECTED and not wxEVT\_MENU as one might glean from the documentation. The C++ header files expand wxEVT\_MENU into a call involving wxEVT\_COMMAND\_MENU\_SELECTED. So this is an important case where we want to know the event of interest and not the macro used in C++ code to identify the event in the event table construction code.

```
# Note that we are adding the callback to f (a wxWindow)
f$AddCallback(wxEVT_COMMAND_MENU_SELECTED, function(ev) cat("Menu item selected")),
```

Adding the callback on the wxMenuItem causes things to hang.

```
i$AddCallback(wxEVT_COMMAND_MENU_SELECTED, function(ev) cat("In direct wxMenuItem
```

We add another entry and append this menu to the menu bar. Note that it is the label here the is in the menu bar display of this menu. It is not the name given in the creation of the wxMenu itself. That, if specified, is used as an additional initial entry in the display of the menu when it is shown.

```
menu$Append(wxID_ANY, "Entry 2")
menuBar$Append(menu, "A")
```

Now, we add a sub-menu off of this one. We create a new menu (with a decorative title item "B") and then add three entries - Red, Green and Blue. Then we append this sub-menu to the "parent" wxMenu object with

---

which it is associated. Note that we are giving the label "My sub menu" to display on the parent/top-level menu.

```
subMenu = wxMenu("B")
subMenu$Append(wxID_ANY, "Red")
subMenu$Append(wxID_ANY, "Blue")
subMenu$Append(wxID_ANY, "Green")
menu$Append(wxID_ANY, "My sub menu", subMenu)
```

We can continue to add new items to the **menu** object. We add a separator to identify a new group of items. And then we add a checkbox widget and a radio button widget.

```
menu$AppendSeparator()
menu$AppendCheckItem(wxID_ANY, "A Check box")
menu$AppendRadioItem(wxID_ANY, "My Radio button")
```

We now move on to creating a separate menu which we add to the menu bar under the label "City".

```
menu = wxMenu("European Cities")
menuBar$Append(menu, "City")
```

We create the items in this menu by looping over the names of the cities in the **eurodist** object available in R. Note that in this case, we explicitly control the id values used when creating the wxMenuItems rather than using wxID\_ANY. We start these at 401 and have 402, 403, ... This allows us to refer to these items by id and by id range which is convenient when specifying events.

```
cityNames = attr(eurodist, "Labels")
sapply(seq(along = cityNames),
       function(i) menu$Append(400 + i, cityNames[i]))
```

And we specify a single callback for all of these items using the regular wxObject\_AddCallback function but specifying two values for objId. These values give the first and last window id to be included in this callback specification and includes all the values within this close range, i.e. 401, 402, ..., 426. (Again, we invoke the callback on the wxWindow and not a wxMenuItem.)

```
f$AddCallback(wxEVT_COMMAND_MENU_SELECTED,
             objId = c(401, 400 + length(cityNames)),
             function(ev) {
               i = ev$GetId()
               cat("Selected city", i, cityNames[i - 400], "\n")
               print(ev$GetString())
             })
```

Finally, we add a toolbar and put some controls/tools on it. We use JPEG files from the R distribution and we create a normal, check and radio item.

```
tbar = f$CreateToolBar()

imageNames = list.files(paste(Sys.getenv("R_HOME"), "doc", "html", sep = .Platform
                             "(left|right|up)\\.jpg$", full.names = TRUE))

type = c(wxITEM_NORMAL, wxITEM_CHECK, wxITEM_RADIO)
sapply(seq(along = imageNames),
       function(i)
```

---

```
tbar$AddTool(wxID_ANY,  
             gsub("\.jpg$", "", basename(imageNames[i])),  
             wxBitmap(imageNames[i], wxBITMAP_TYPE_JPEG),  
             kind = type[i])
```

We must realize the toolbar after adding the tools.

```
tbar$Realize()
```

And we show the top-level window

```
f$Show()
```

and run the event loop.

```
eloop = wxEventLoop()
```

```
eloop$Run()
```