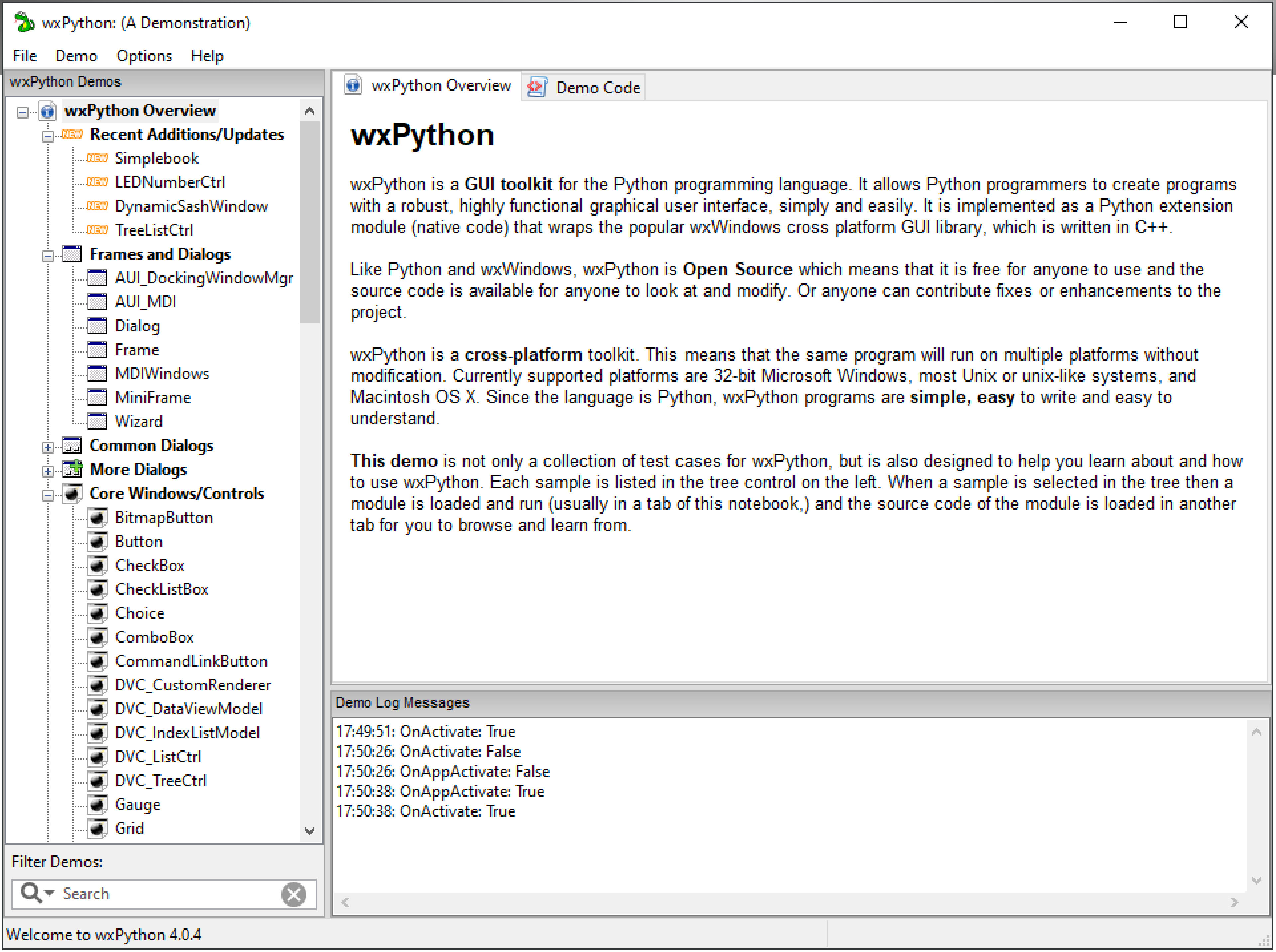
Gui mit wxDemo

Einstieg in die Entwicklung
grafischer Benutzeroberflächen
mit Hilfe von wxDemo

Gui mit wxDemo



wxPython

wxPython is a **GUI toolkit** for the Python programming language. It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily. It is implemented as a Python extension module (native code) that wraps the popular wxWindows cross platform GUI library, which is written in C++.

Like Python and wxWindows, wxPython is **Open Source** which means that it is free for anyone to use and the source code is available for anyone to look at and modify. Or anyone can contribute fixes or enhancements to the project.

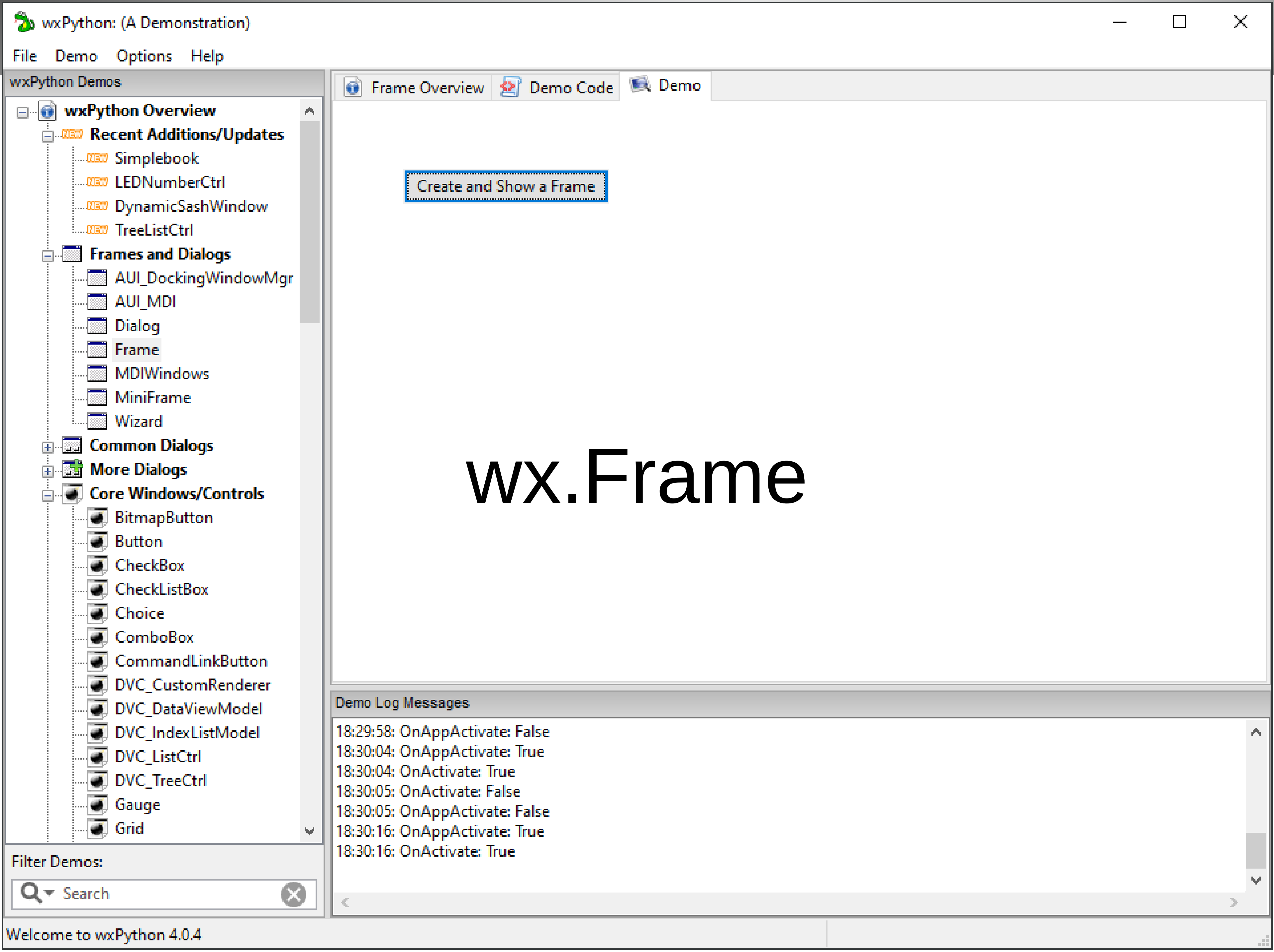
wxPython is a **cross-platform** toolkit. This means that the same program will run on multiple platforms without modification. Currently supported platforms are 32-bit Microsoft Windows, most Unix or unix-like systems, and Macintosh OS X. Since the language is Python, wxPython programs are **simple, easy** to write and easy to understand.

This demo is not only a collection of test cases for wxPython, but is also designed to help you learn about and how to use wxPython. Each sample is listed in the tree control on the left. When a sample is selected in the tree then a module is loaded and run (usually in a tab of this notebook,) and the source code of the module is loaded in another tab for you to browse and learn from.

```
17:49:51: OnActivate: True
17:50:26: OnActivate: False
17:50:26: OnAppActivate: False
17:50:38: OnAppActivate: True
17:50:38: OnActivate: True
```

Welcome to wxPython 4.0.4

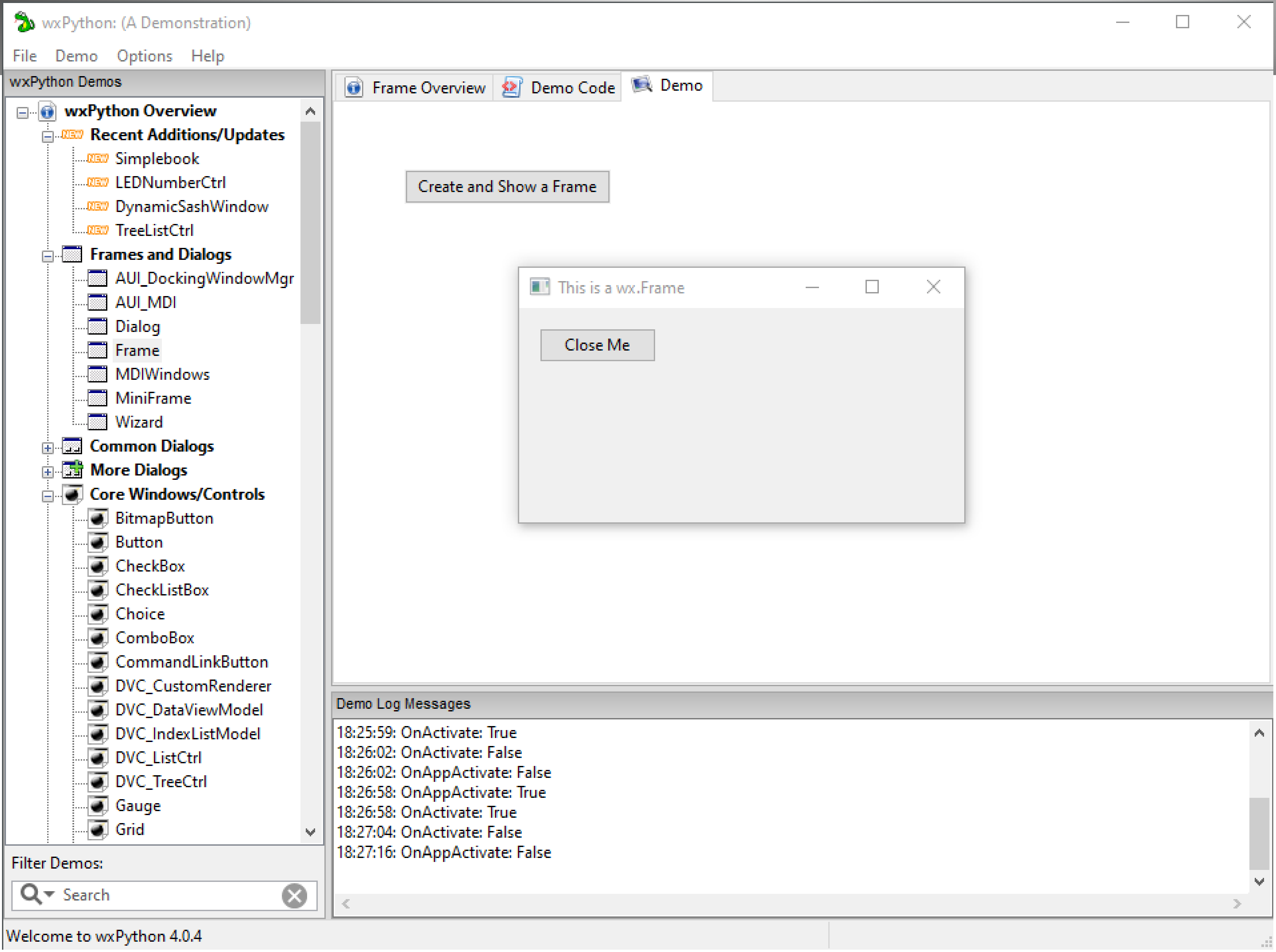
Gui mit wxDemo



```
18:29:58: OnAppActivate: False
18:30:04: OnAppActivate: True
18:30:04: OnActivate: True
18:30:05: OnActivate: False
18:30:05: OnAppActivate: False
18:30:16: OnAppActivate: True
18:30:16: OnActivate: True
```

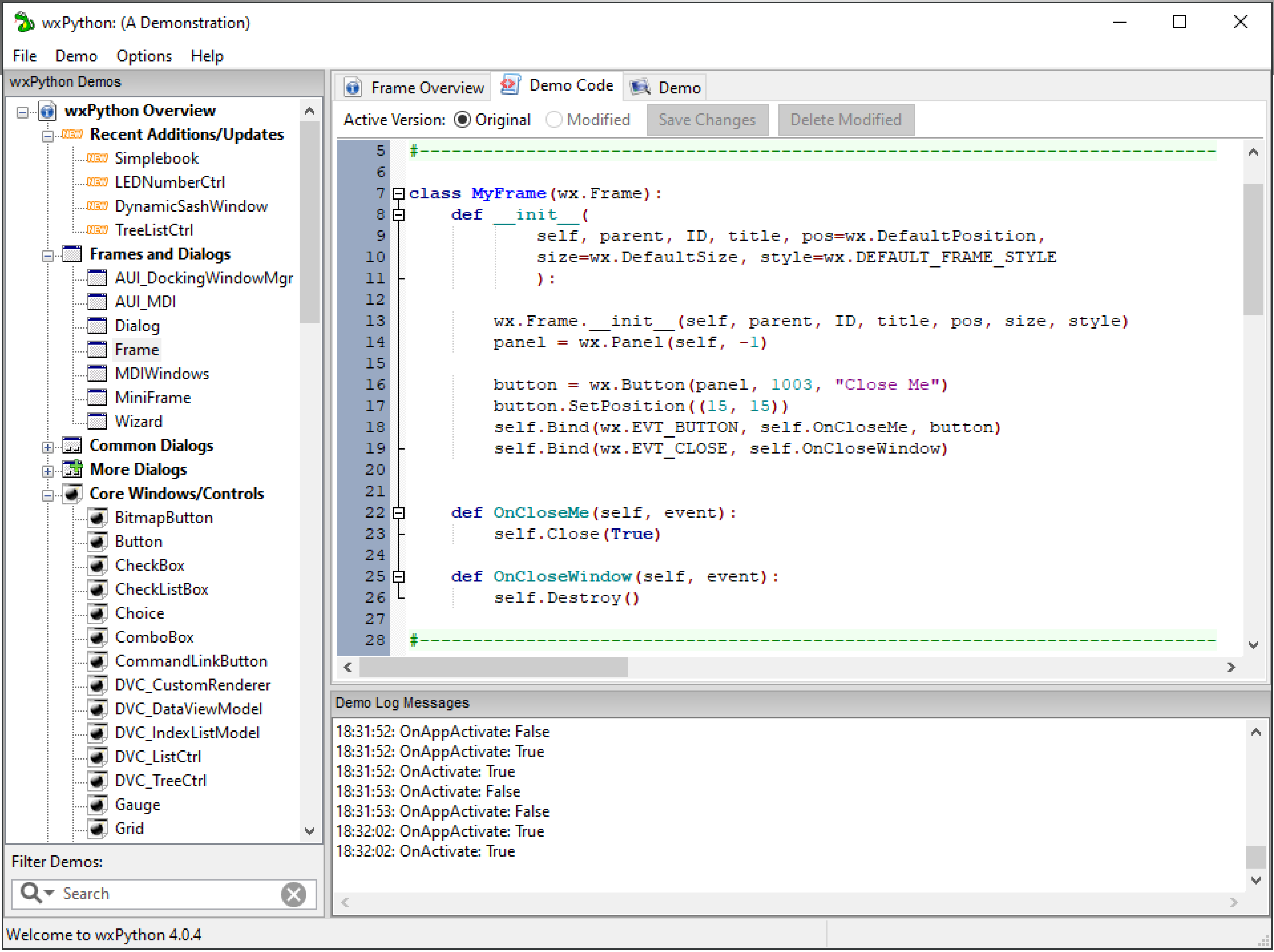
wx.Frame

Gui mit wxDemo



```
18:25:59: OnActivate: True
18:26:02: OnActivate: False
18:26:02: OnAppActivate: False
18:26:58: OnAppActivate: True
18:26:58: OnActivate: True
18:27:04: OnActivate: False
18:27:16: OnAppActivate: False
```

Gui mit wxDemo



Active Version: [✓ Original] Modified

```
#----------------------------------------------------------------------------

class MyFrame(wx.Frame):
    def __init__(
            self, parent, ID, title, pos=wx.DefaultPosition,
            size=wx.DefaultSize, style=wx.DEFAULT_FRAME_STYLE
            ):

        wx.Frame.__init__(self, parent, ID, title, pos, size, style)
        panel = wx.Panel(self, -1)

        button = wx.Button(panel, 1003, "Close Me")
        button.SetPosition((15, 15))
        self.Bind(wx.EVT_BUTTON, self.OnCloseMe, button)
        self.Bind(wx.EVT_CLOSE, self.OnCloseWindow)


    def OnCloseMe(self, event):
        self.Close(True)

    def OnCloseWindow(self, event):
        self.Destroy()

#----------------------------------------------------------------------------
```

```
18:31:52: OnAppActivate: False
18:31:52: OnAppActivate: True
18:31:52: OnActivate: True
18:31:53: OnActivate: False
18:31:53: OnAppActivate: False
18:32:02: OnAppActivate: True
18:32:02: OnActivate: True
```

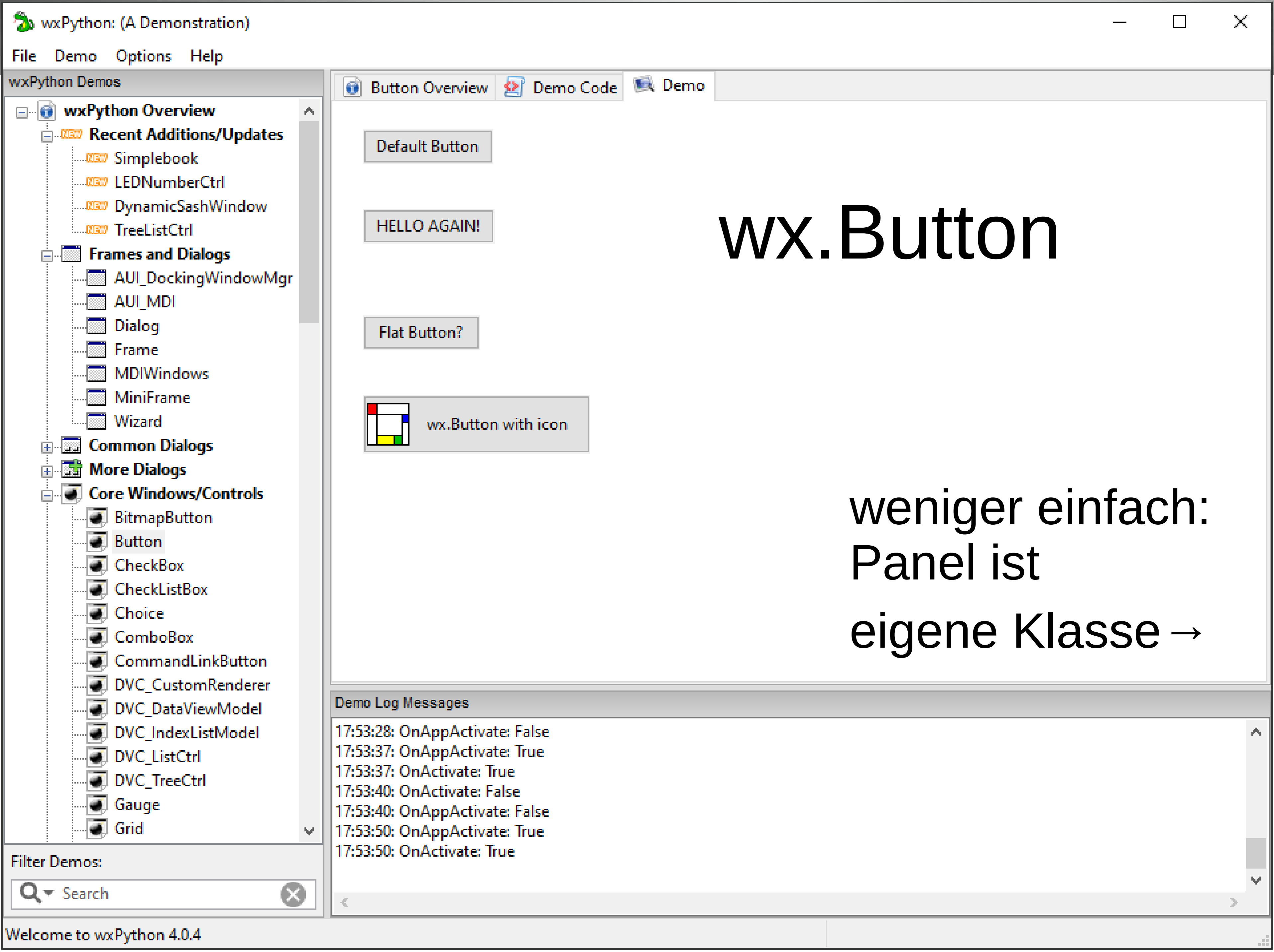
Welcome to wxPython 4.0.4

Gui mit wxDemo

```
import wx

### ---------------------------------------------------
class MyFrame(wx.Frame):
   def __init__(
         self, parent, ID, title, pos=wx.DefaultPosition,
         size=wx.DefaultSize, style=wx.DEFAULT_FRAME_STYLE
         ):

      ... (siehe Bild)

   def OnCloseMe(self, event):
      self.Close(True)

   def OnCloseWindow(self, event):
      self.Destroy()

### ---------------------------------------------------
if __name__ == '__main__':
   app=wx.App()
   fenster=MyFrame(None, -1, "Fenster")
   app.SetTopWindow(fenster)
   fenster.Show(True)
   app.MainLoop()
```

Ergänzungen

Gui mit wxDemo

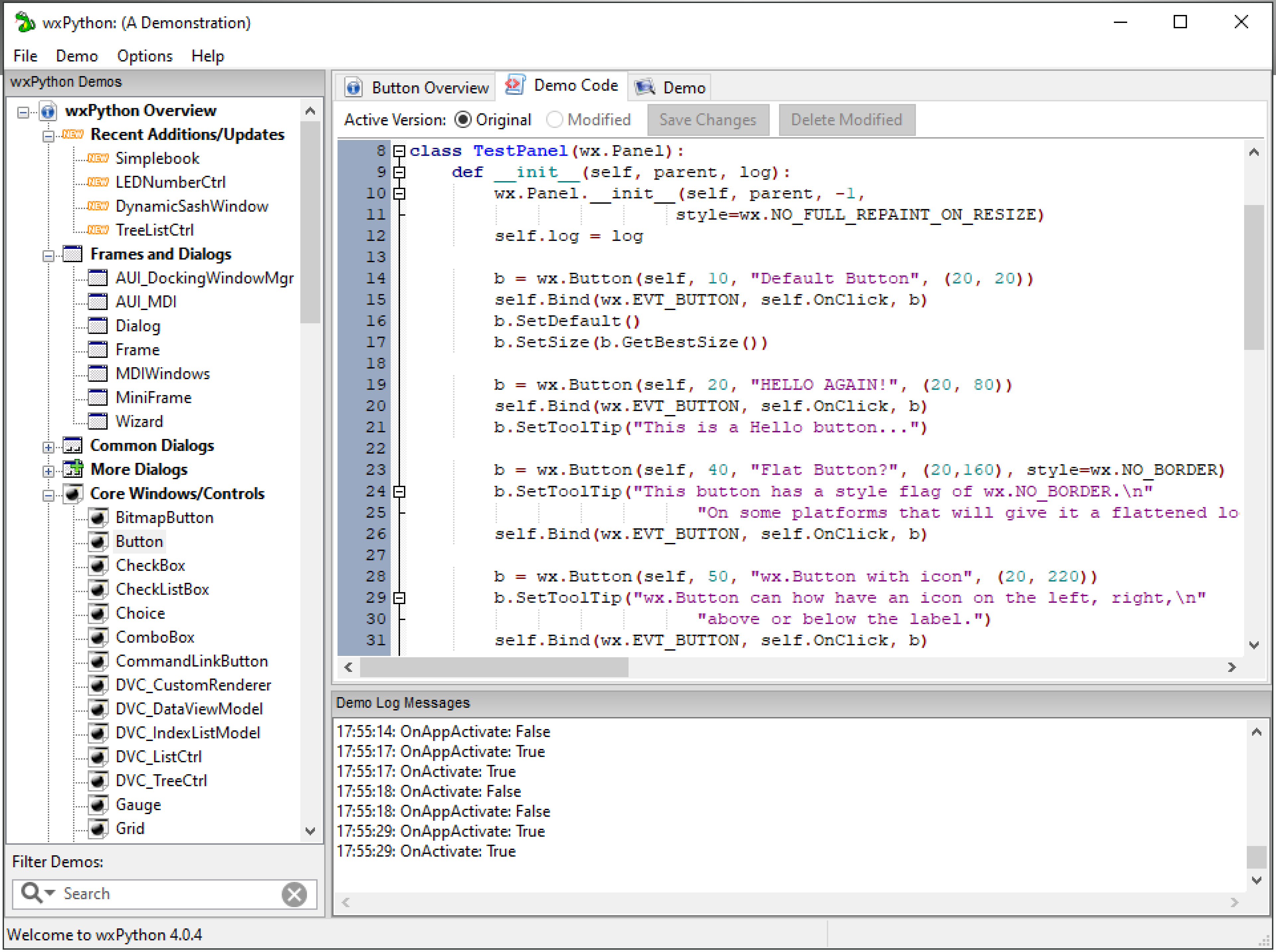


```
17:53:28: OnAppActivate: False
17:53:37: OnAppActivate: True
17:53:37: OnActivate: True
17:53:40: OnActivate: False
17:53:40: OnAppActivate: False
17:53:50: OnAppActivate: True
17:53:50: OnActivate: True
```

wx.Button

weniger einfach:
Panel ist
eigene Klasse→

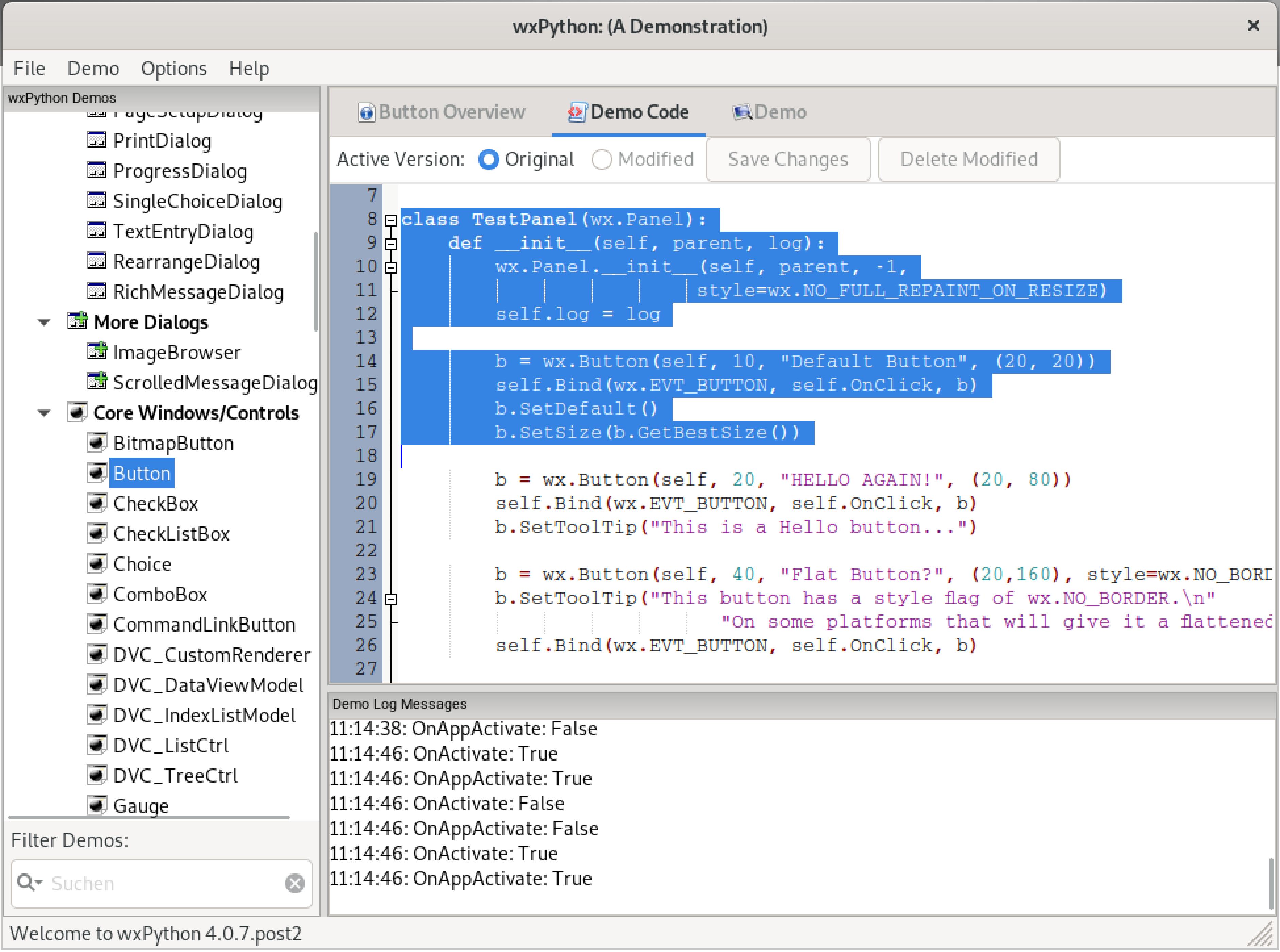
Gui mit wxDemo



```python
class TestPanel(wx.Panel):
    def __init__(self, parent, log):
        wx.Panel.__init__(self, parent, -1,
                          style=wx.NO_FULL_REPAINT_ON_RESIZE)
        self.log = log

        b = wx.Button(self, 10, "Default Button", (20, 20))
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)
        b.SetDefault()
        b.SetSize(b.GetBestSize())

        b = wx.Button(self, 20, "HELLO AGAIN!", (20, 80))
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)
        b.SetToolTip("This is a Hello button...")

        b = wx.Button(self, 40, "Flat Button?", (20,160), style=wx.NO_BORDER)
        b.SetToolTip("This button has a style flag of wx.NO_BORDER.\n"
                          "On some platforms that will give it a flattened lo
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)

        b = wx.Button(self, 50, "wx.Button with icon", (20, 220))
        b.SetToolTip("wx.Button can how have an icon on the left, right,\n"
                          "above or below the label.")
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)
```

```
17:55:14: OnAppActivate: False
17:55:17: OnAppActivate: True
17:55:17: OnActivate: True
17:55:18: OnActivate: False
17:55:18: OnAppActivate: False
17:55:29: OnAppActivate: True
17:55:29: OnActivate: True
```

Gui mit wxDemo



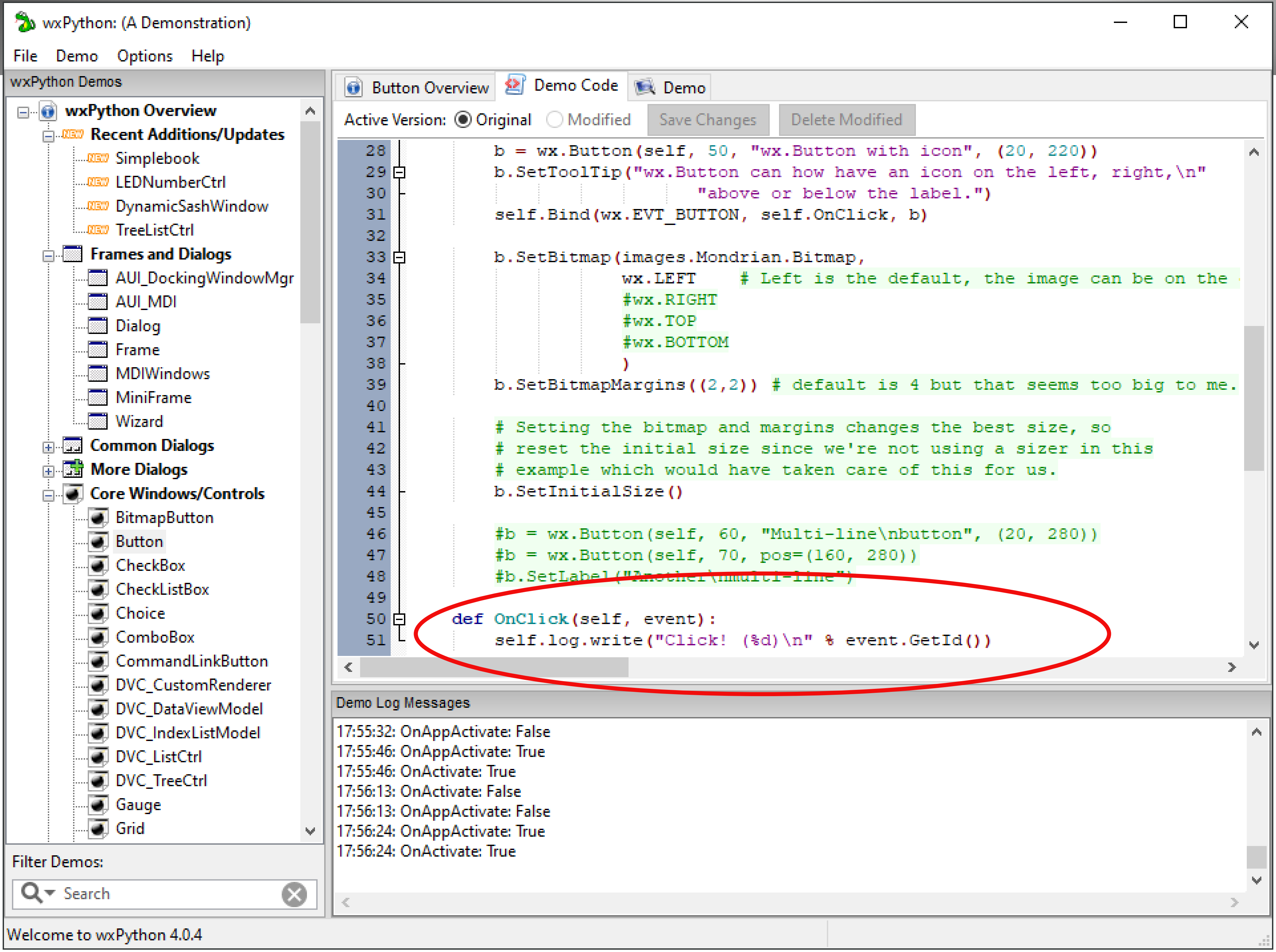
Active Version: [✓ Original] Modified

```

class TestPanel(wx.Panel):
    def __init__(self, parent, log):
        wx.Panel.__init__(self, parent, -1,
                          style=wx.NO_FULL_REPAINT_ON_RESIZE)
        self.log = log

        b = wx.Button(self, 10, "Default Button", (20, 20))
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)
        b.SetDefault()
        b.SetSize(b.GetBestSize())

        b = wx.Button(self, 20, "HELLO AGAIN!", (20, 80))
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)
        b.SetToolTip("This is a Hello button...")

        b = wx.Button(self, 40, "Flat Button?", (20,160), style=wx.NO_BORD
        b.SetToolTip("This button has a style flag of wx.NO_BORDER.\n"
                     "On some platforms that will give it a flattened
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)

```

```
11:14:38: OnAppActivate: False
11:14:46: OnActivate: True
11:14:46: OnAppActivate: True
11:14:46: OnActivate: False
11:14:46: OnAppActivate: False
11:14:46: OnActivate: True
11:14:46: OnAppActivate: True
```

Welcome to wxPython 4.0.7.post2

Gui mit wxDemo



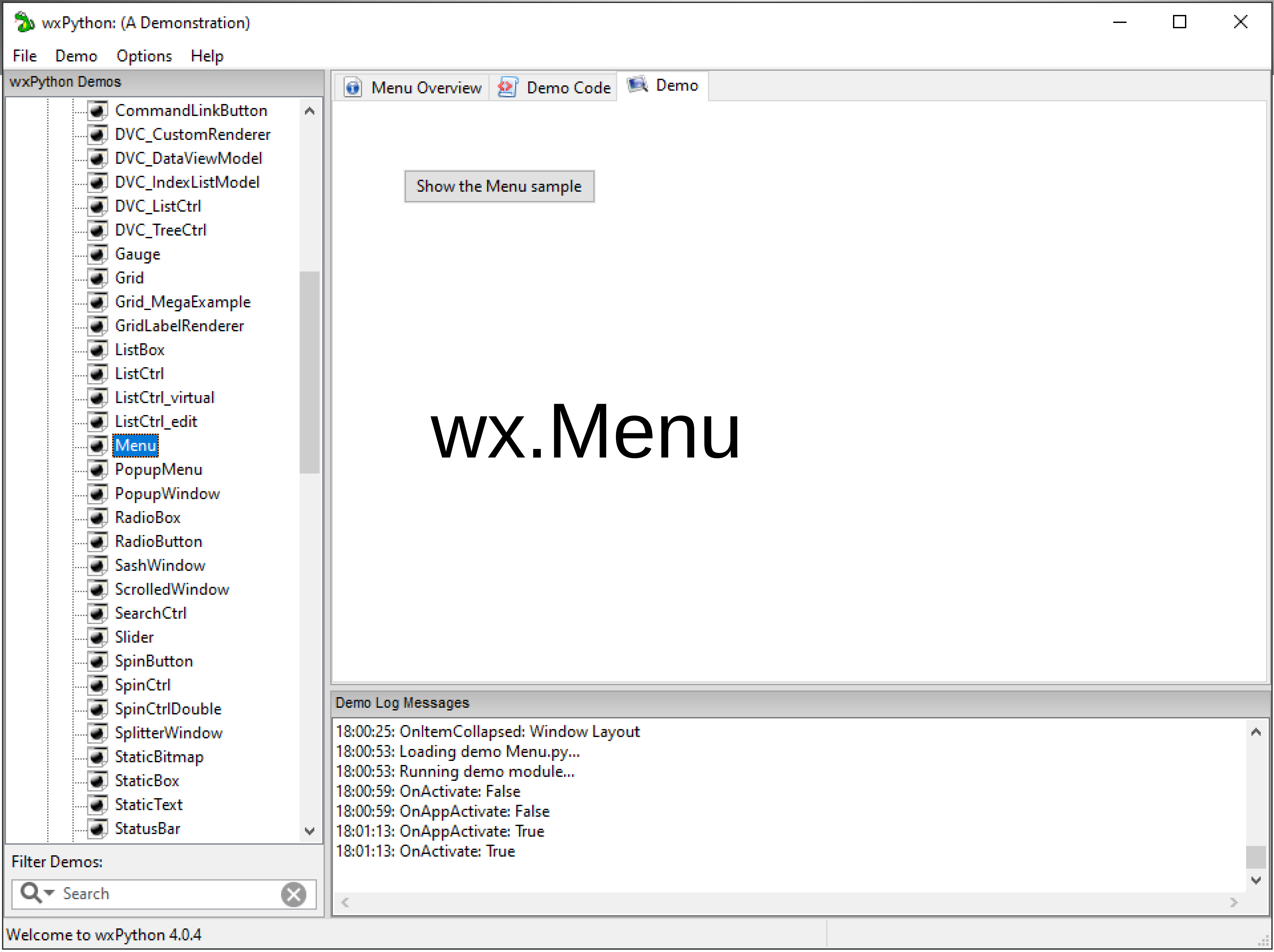
Active Version: [✓ Original] Modified

```
        b = wx.Button(self, 50, "wx.Button with icon", (20, 220))
        b.SetToolTip("wx.Button can how have an icon on the left, right,\n"
                     "above or below the label.")
        self.Bind(wx.EVT_BUTTON, self.OnClick, b)

        b.SetBitmap(images.Mondrian.Bitmap,
                    wx.LEFT    # Left is the default, the image can be on the
                    #wx.RIGHT
                    #wx.TOP
                    #wx.BOTTOM
                    )
        b.SetBitmapMargins((2,2)) # default is 4 but that seems too big to me.

        # Setting the bitmap and margins changes the best size, so
        # reset the initial size since we're not using a sizer in this
        # example which would have taken care of this for us.
        b.SetInitialSize()

        #b = wx.Button(self, 60, "Multi-line\nbutton", (20, 280))
        #b = wx.Button(self, 70, pos=(160, 280))
        #b.SetLabel("Another\nmulti-line")

    def OnClick(self, event):
        self.log.write("Click! (%d)\n" % event.GetId())
```

```
17:55:32: OnAppActivate: False
17:55:46: OnAppActivate: True
17:55:46: OnActivate: True
17:56:13: OnActivate: False
17:56:13: OnAppActivate: False
17:56:24: OnAppActivate: True
17:56:24: OnActivate: True
```

Welcome to wxPython 4.0.4

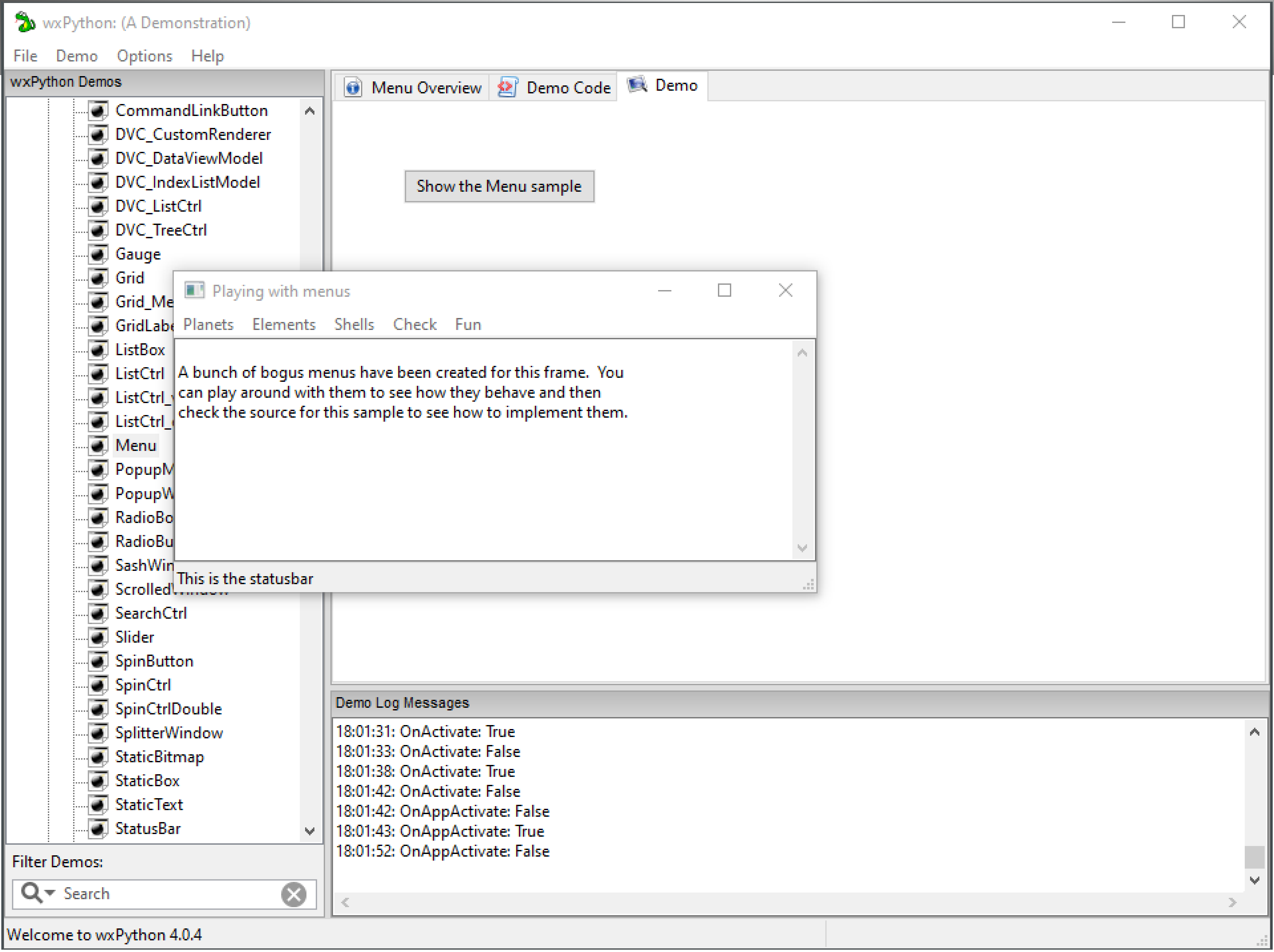
Gui mit wxDemo



```
18:00:25: OnItemCollapsed: Window Layout
18:00:53: Loading demo Menu.py...
18:00:53: Running demo module...
18:00:59: OnActivate: False
18:00:59: OnAppActivate: False
18:01:13: OnAppActivate: True
18:01:13: OnActivate: True
```

wx.Menu

Gui mit wxDemo



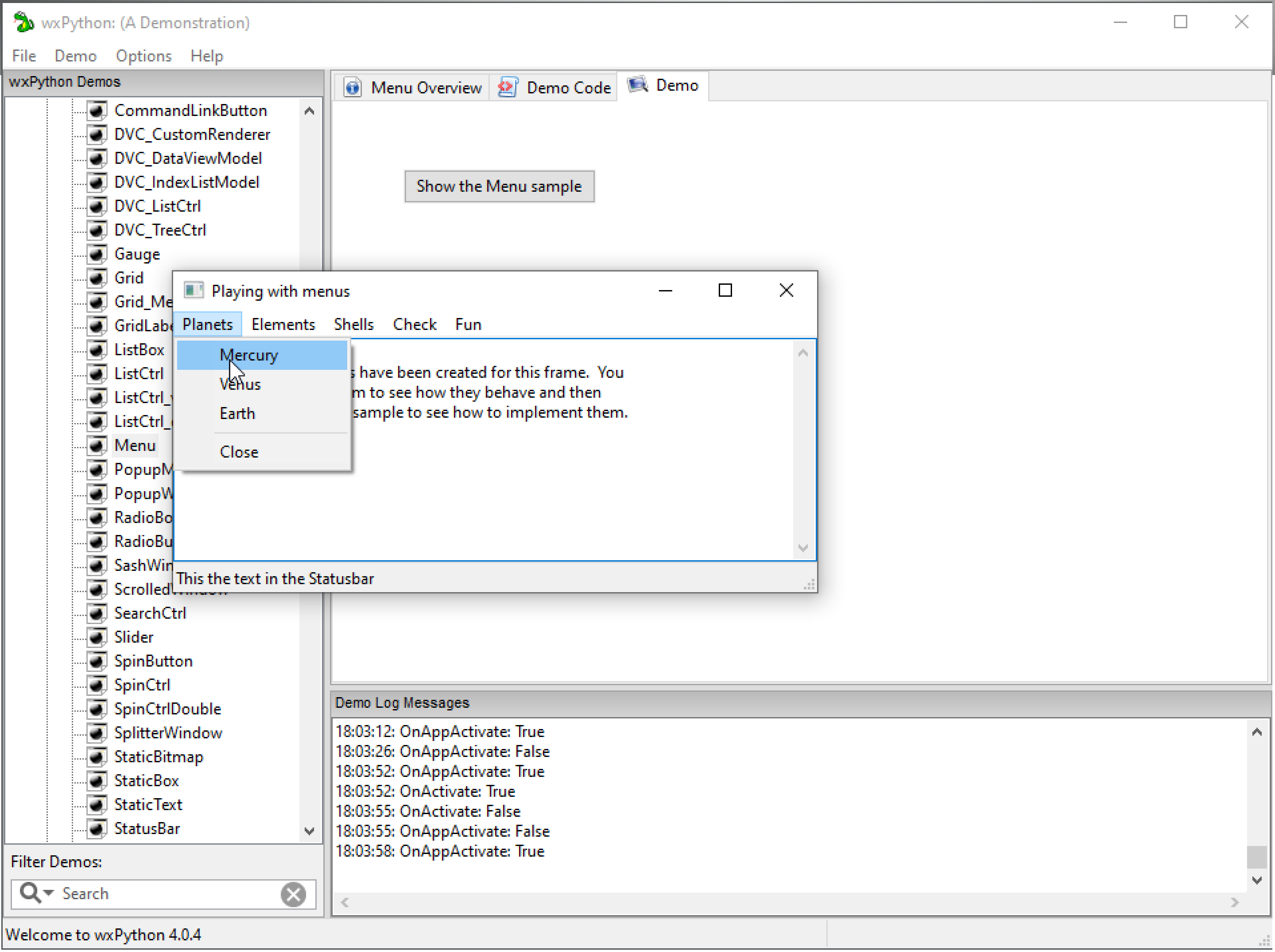
A bunch of bogus menus have been created for this frame. You can play around with them to see how they behave and then check the source for this sample to see how to implement them.

This is the statusbar

```
18:01:31: OnActivate: True
18:01:33: OnActivate: False
18:01:38: OnActivate: True
18:01:42: OnActivate: False
18:01:42: OnAppActivate: False
18:01:43: OnAppActivate: True
18:01:52: OnAppActivate: False
```

Welcome to wxPython 4.0.4

Gui mit wxDemo



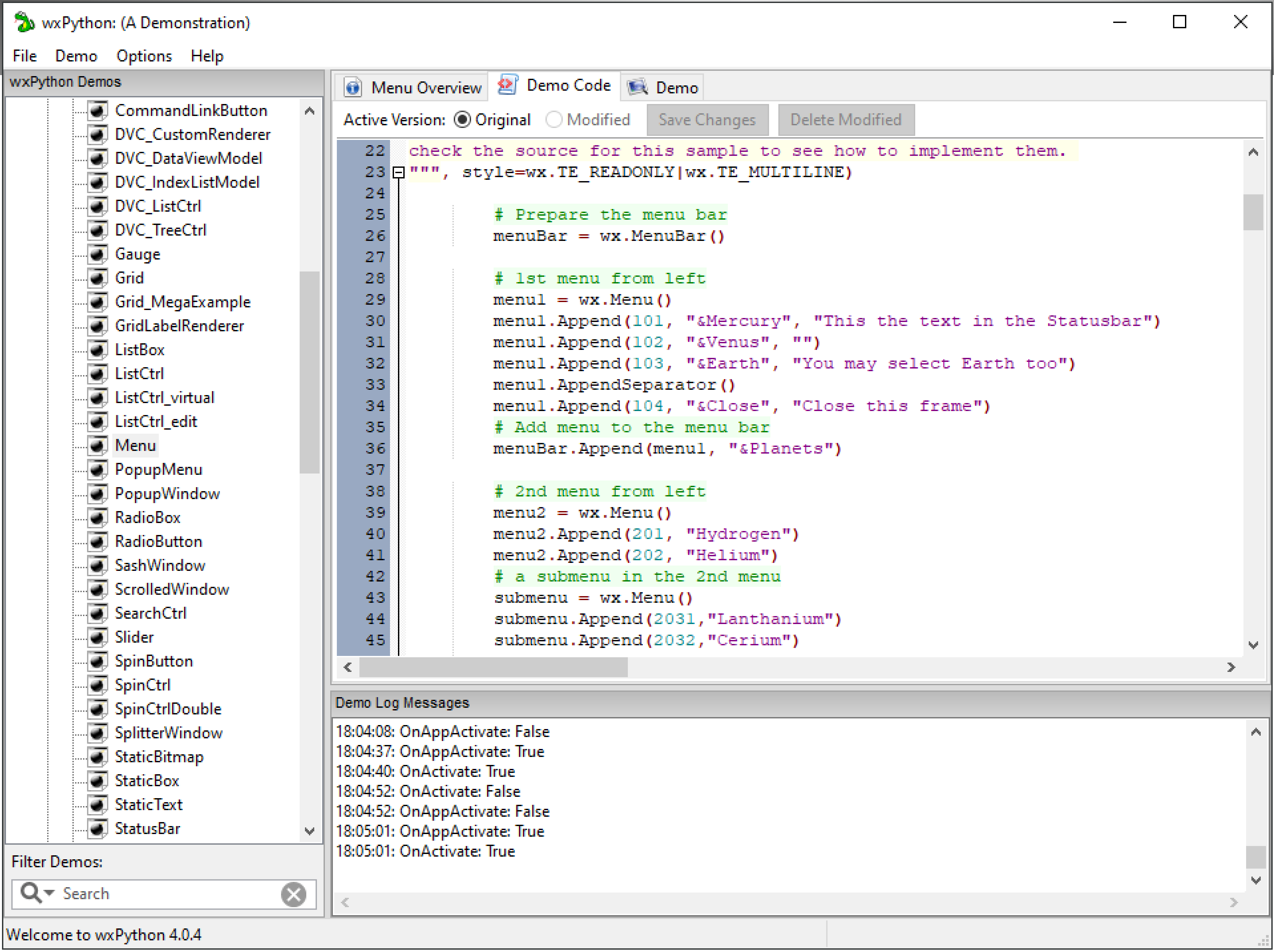
s have been created for this frame. You
m to see how they behave and then
sample to see how to implement them.

This the text in the Statusbar

```
18:03:12: OnAppActivate: True
18:03:26: OnAppActivate: False
18:03:52: OnAppActivate: True
18:03:52: OnActivate: True
18:03:55: OnActivate: False
18:03:55: OnAppActivate: False
18:03:58: OnAppActivate: True
```

Welcome to wxPython 4.0.4

Gui mit wxDemo



```
check the source for this sample to see how to implement them.
""", style=wx.TE_READONLY|wx.TE_MULTILINE)

        # Prepare the menu bar
        menuBar = wx.MenuBar()

        # 1st menu from left
        menu1 = wx.Menu()
        menu1.Append(101, "&Mercury", "This the text in the Statusbar")
        menu1.Append(102, "&Venus", "")
        menu1.Append(103, "&Earth", "You may select Earth too")
        menu1.AppendSeparator()
        menu1.Append(104, "&Close", "Close this frame")
        # Add menu to the menu bar
        menuBar.Append(menu1, "&Planets")

        # 2nd menu from left
        menu2 = wx.Menu()
        menu2.Append(201, "Hydrogen")
        menu2.Append(202, "Helium")
        # a submenu in the 2nd menu
        submenu = wx.Menu()
        submenu.Append(2031,"Lanthanium")
        submenu.Append(2032,"Cerium")
```

```
18:04:08: OnAppActivate: False
18:04:37: OnAppActivate: True
18:04:40: OnActivate: True
18:04:52: OnActivate: False
18:04:52: OnAppActivate: False
18:05:01: OnAppActivate: True
18:05:01: OnActivate: True
```

Gui mit wxDemo

Eine einfache Lösung
(keine eigene Panel-Klasse)

Gui mit wxDemo

… (siehe erste Code-Folie)

```
panel = wx.Panel(self, -1)

### Menu-Abschnitt
menuBar = wx.MenuBar()          # Prepare the menu bar

menu1 = wx.Menu()               # 1st menu from left
menu1.Append(101, "&Mercury", "This the text in the Statusbar")
menu1.Append(102, "&Venus", "")
menu1.Append(103, "&Earth", "You may select Earth too")
menu1.AppendSeparator()
menu1.Append(104, "&Close", "Close this frame")

menuBar.Append(menu1, "&Planets")      # Add menu to the menu bar

self.SetMenuBar(menuBar)

# Menu events
## self.Bind(wx.EVT_MENU_HIGHLIGHT_ALL, self.OnMenuHighlight) ## raus!

self.Bind(wx.EVT_MENU, self.Menu101, id=101)
self.Bind(wx.EVT_MENU, self.Menu102, id=102)
self.Bind(wx.EVT_MENU, self.Menu103, id=103)
self.Bind(wx.EVT_MENU, self.OnCloseWindow, id=104) ## Korrektur

### Button-Abschnitt
```

… (siehe nächste Code-Folie)

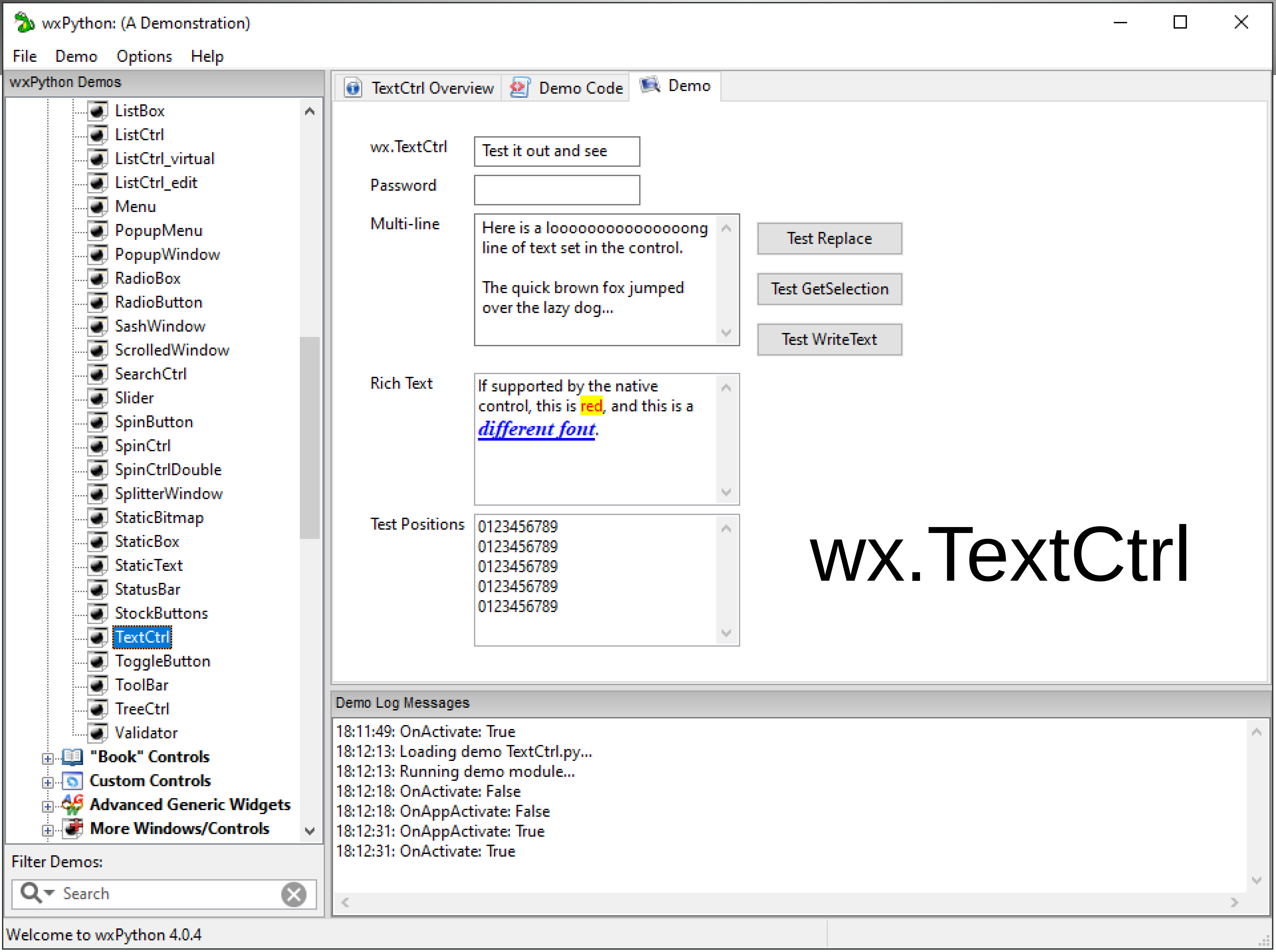
Gui mit wxDemo

… (siehe vorige Code-Folie)

```
    ### Button-Abschnitt
    button = wx.Button(panel, 1003, "Close Me")
    button.SetPosition((15, 15))
    self.Bind(wx.EVT_BUTTON, self.OnCloseMe, button)
    self.Bind(wx.EVT_CLOSE, self.OnCloseWindow)

### Ereignisbehandlung Menu-Abschnitt
def Menu101(self, event):
    ## self.log.write('Welcome to Mercury\n') # anpassen:
    print('Welcome to Mercury\n')

def Menu102(self, event):
    print('Welcome to Venus\n')          ## s.o.

def Menu103(self, event):
    print('Welcome to the Earth\n')          ## s.o.

### Ereignisbehandlung Button-Abschnitt
def OnCloseMe(self, event):
    self.Close(True)

def OnCloseWindow(self, event):
    self.Destroy()
```

… (siehe erste Code-Folie)

Gui mit wxDemo



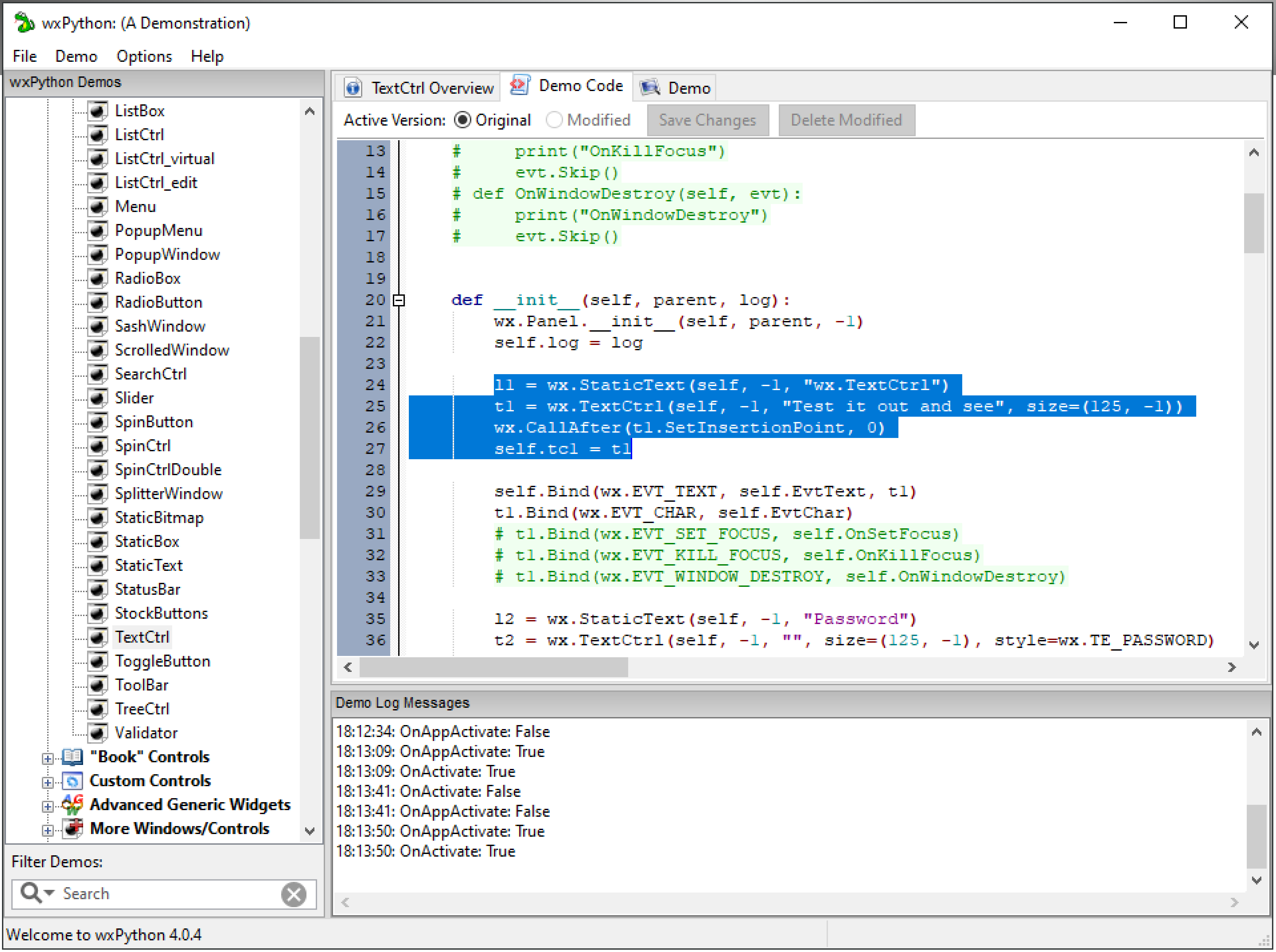
wx.TextCtrl	Test it out and see
Password	
Multi-line	Here is a looooooooooooooong line of text set in the control. The quick brown fox jumped over the lazy dog...
Rich Text	If supported by the native control, this is red, and this is a *different font*.
Test Positions	0123456789 0123456789 0123456789 0123456789 0123456789

```
18:11:49: OnActivate: True
18:12:13: Loading demo TextCtrl.py...
18:12:13: Running demo module...
18:12:18: OnActivate: False
18:12:18: OnAppActivate: False
18:12:31: OnAppActivate: True
18:12:31: OnActivate: True
```

Welcome to wxPython 4.0.4

wx.TextCtrl

Gui mit wxDemo



Active Version: [✓ Original] Modified

```
        #     print("OnKillFocus")
        #     evt.Skip()
        # def OnWindowDestroy(self, evt):
        #     print("OnWindowDestroy")
        #     evt.Skip()


        def __init__(self, parent, log):
            wx.Panel.__init__(self, parent, -1)
            self.log = log

            l1 = wx.StaticText(self, -1, "wx.TextCtrl")
            t1 = wx.TextCtrl(self, -1, "Test it out and see", size=(125, -1))
            wx.CallAfter(t1.SetInsertionPoint, 0)
            self.tc1 = t1

            self.Bind(wx.EVT_TEXT, self.EvtText, t1)
            t1.Bind(wx.EVT_CHAR, self.EvtChar)
            # t1.Bind(wx.EVT_SET_FOCUS, self.OnSetFocus)
            # t1.Bind(wx.EVT_KILL_FOCUS, self.OnKillFocus)
            # t1.Bind(wx.EVT_WINDOW_DESTROY, self.OnWindowDestroy)

            l2 = wx.StaticText(self, -1, "Password")
            t2 = wx.TextCtrl(self, -1, "", size=(125, -1), style=wx.TE_PASSWORD)
```

```
18:12:34: OnAppActivate: False
18:13:09: OnAppActivate: True
18:13:09: OnActivate: True
18:13:41: OnActivate: False
18:13:41: OnAppActivate: False
18:13:50: OnAppActivate: True
18:13:50: OnActivate: True
```

Gui mit wxDemo

… (zusätzlich im Konstruktor)

```
    ### Label (StaticText) und TextCtrl
    self.label1 = wx.StaticText(self, -1, "Ein/Ausgabe")
    self.label1.SetPosition((155, 15))
    self.textCtrl1 = wx.TextCtrl(self, -1, "noch ohne Inhalt", size=(205, -1))
    self.textCtrl1.SetPosition((155, 45))
    self.Bind(wx.EVT_TEXT, self.EvtText, self.textCtrl1)      # sinnvoll?
```

… (zusätzlich in Ereignisbehandlung)

```
### Ereignisbehandlung
### TextCtrl
def EvtText(self, event):                    ## (so noch nicht sinnvoll)
    print('EvtText')
```

… (siehe vorige Code-Folie)