

《wxPython的事件与事件处理小记》

一般GUI框架都是基于**事件**的流程控制——程序大部分事件都是在接收和响应用户生成的事件，比如用户点击一个按钮、拖动窗口等。

一、wxPython对事件的描述分为三个部分：

1. 事件类型： `EventType`，事件的类型的唯一标识。
2. 事件类： `Event`，每个发生的事件都有与之关联的信息，而这些信息由事件类对象表示，并由事件携带。
3. 事件源：即发生事件的源对象，可以是一个窗口对象，也可以是各种控件对象。通常，不同的对象可能发生相同的事件(比如，一个窗口中有多个按钮，这些按钮都会发生点击事件)，wxPython通过检查事件源或其id来区分。

二、wxPython中事件处理的主要方法，是调用 `Bind()` 来绑定一个事件的处理函数。这个处理函数可以是：

1. 当前窗口对象或其他窗口对象的方法
2. 普通函数，如静态方法、全局函数
3. 可调用对象

注意，事件处理函数，接收唯一参数 `event`（事件类对象，该对象携带一定信息）

解除事件绑定使用 `Unbind()`

```
1 import wx
2
3 class MainWin(wx.Frame):
4     def __init__(self, *args, **kwargs):
5         super().__init__(*args, **kwargs)
6         self.b = wx.Button(self, label='b1')
7
8         self.Bind(wx.EVT_BUTTON, self.handler, self.b)
9
10    def handler(self, event):
11        print('button clicked!')
```

三、wxPython的事件处理流程

当从窗口系统中接收到一个事件，wxPython会调用 `wx.EvtHandler.ProcessEvent` 作为第一个事件处理。通常情况下， `ProcessEvent` 按照如下顺序找到处理函数。如果找到则事件处理流程结束，除非在事件处理函数中调用 `Skip()` 事件被视为未处理并继续查找。

事件处理函数查找流程如下：

1. 在所有流程之前， `wx.AppConsole.FilterEvent` 会被首先调用。当该函数默认返回 `-1` 事件继续流转。如果返回 `-1` 以外任意值，则事件处理流程立刻终止。——这被称为wxPython的事件过滤。
2. 如果通过调用 `wx.EvtHandler.SetEvtHandlerEnabled` 禁用事件处理，则跳过接下来的三步骤，并事件处理程序在(5)中恢复。

3. 如果当前对象是一个 `wx.Window`，并且关联了一个 `wx.validator`，那么该校验器有机会处理该事件。
4. 查询动态绑定事件处理程序的列表，即那些被 `Bind()` 调用的事件处理程序。
5. 检查包含使用此类及其基类中的事件表宏定义的所有处理程序的事件表。
6. 该事件被传递给事件处理程序链中的下一个事件处理程序。
7. 如果对象是一个 `wx.Window` 并且事件被设置为传播（默认情况下只有从 `wx.CommandEvent` 派生的事件类型被设置为传播），那么处理从步骤（1）重新开始（不包括步骤（7））为父窗口。如果此对象不是窗口但存在下一个处理程序，则事件将传递给它的父窗口（如果它是窗口）。这确保在（可能有几个）非窗口事件处理程序被推送到窗口顶部的常见情况下，事件最终到达窗口父级。
8. 最后，如果事件仍未处理，`wx.App` 对象本身（派生自 `wx.EvtHandler`）将获得最后的机会来处理它。

四、自定义事件

使用 `wx.lib.newevent` 内部定义函数创建。

```
1 import wx
2 from wx.lib.newevent import NewEvent, NewCommandEvent
3
4 # 本窗口处理，不可传播
5 SomeNewEvent, EVT_SOME_NEW_EVENT = NewEvent()
6 # 可向上传播
7 SomeNewCommandEvent, EVT_SOME_NEW_COMMAND_EVENT = NewCommandEvent()
```

绑定事件

```
1 self.Bind(EVT_SOME_NEW_EVENT, self.handler)
2 EVT_SOME_NEW_EVENT(self, self.handler)
```

事件附加数据

```
1 # Create the event
2 evt = SomeNewEvent(attr1="hello", attr2=654)
3 # Post the event
4 wx.PostEvent(target, evt)
```

在处理具有此类任意数据的事件时，您可以通过属性获取数据，其名称与创建事件实例时传入的名称相同。

```
1 def handler(self, evt):
2     # Given the above constructed event, the following is true
3     evt.attr1 == "hello"
4     evt.attr2 == 654
```