

wxPython: Creating a Simple MP3 Player

🕒 April 20, 2010 📁 Cross-Platform, Python, wxPython 🔧 Python, wxPython 👤 Mike

Last month, I started a series of articles on creating simple applications with wxPython. The first couple of articles were on an Image Viewer program. While I won't abandon that project, I thought it was time for us to delve into something new. In this article we will start a journey into the wild and woolly world of playing MP3s. We will create a very simple interface that we can use to play, pause and stop a song with. We will also learn how to seek within a track and change the volume of the music. In future articles, we will add a display with music information (like title, artist, genre, etc), track lists, a random function, and more. Let's get started!

[Tweet](#)

[Share](#)

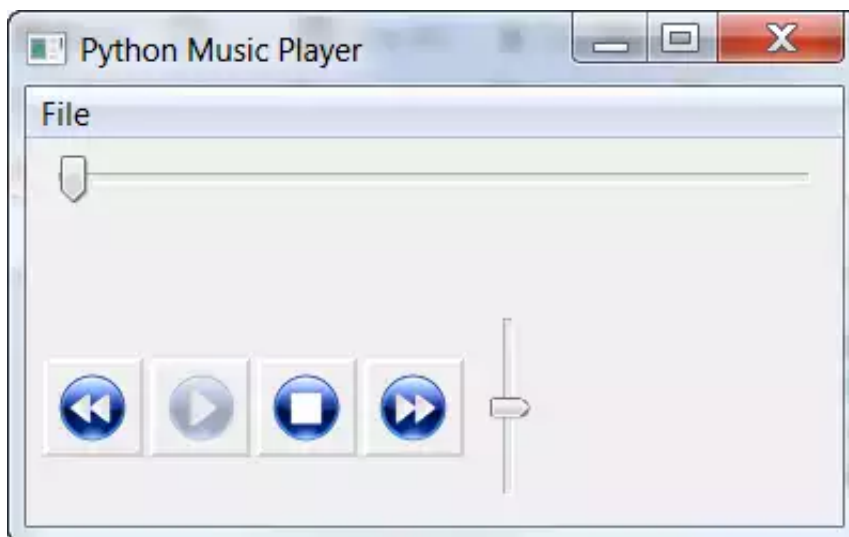
61

[reddit](#)

[G+1](#)

Getting Ready to Spin the Music

There are many different layouts that we could have gone with, but for this example, we'll use the traditional one where we have a horizontal song track slider widget along the top with player controls underneath and a volume control on the right. If you're familiar with the widgets of wxPython, then you might think that the ShapedButtons would be perfect for this application. I thought so too until I found out that they depended on the Python Imaging Library (PIL). While that's not a big deal, for a simple example I thought it complicated matters for this article and ended up using some generic buttons instead. When we enhance this program later, we may take the time to update the buttons to give them more pizzazz. For now, our application will look like this:



It's not the prettiest music player in the world, but we can fix that later. The point is learning how to

make an application with wxPython that works in a cross-platform manner. When I run this application on my Windows XP machine, it appears to use ffdshow. When I run it on Windows 7, I think it's using Windows Media Player on the backend. As I understand it, wx.MediaCtrl will wrap GStreamer on Linux. Anyway, let's take a look at the source:

```
#-----  
-----  
# player_skeleton2.py  
#-----  
-----  
  
import os  
import wx  
import wx.media  
import wx.lib.buttons as buttons  
  
dirName = os.path.dirname(os.path.abspath(__file__))  
bitmapDir = os.path.join(dirName, 'bitmaps')  
  
#####  
#####  
class MediaPanel(wx.Panel):  
    """  
  
    #-----  
    -----  
  
    def __init__(self, parent):  
        """Constructor"""  
        wx.Panel.__init__(self, parent=parent)  
  
        self.frame = parent  
        self.currentVolume = 50  
        self.createMenu()  
        self.layoutControls()  
  
        sp = wx.StandardPaths.Get()  
        self.currentFolder = sp.GetDocumentsDir()  
  
        self.timer = wx.Timer(self)  
        self.Bind(wx.EVT_TIMER, self.onTimer)  
        self.timer.Start(100)  
  
    #-----
```

```

-----
def layoutControls(self):
    """
    Create and layout the widgets
    """

    try:
        self.mediaPlayer = wx.media.MediaCtrl(self,
style=wx.SIMPLE_BORDER)
    except NotImplementedError:
        self.Destroy()
        raise

    # create playback slider
    self.playbackSlider = wx.Slider(self,
size=wx.DefaultSize)
    self.Bind(wx.EVT_SLIDER, self.onSeek,
self.playbackSlider)

    self.volumeCtrl = wx.Slider(self,
style=wx.SL_VERTICAL|wx.SL_INVERSE)
    self.volumeCtrl.SetRange(0, 100)
    self.volumeCtrl.SetValue(self.currentVolume)
    self.volumeCtrl.Bind(wx.EVT_SLIDER, self.onSetVolume)

    # Create sizers
    mainSizer = wx.BoxSizer(wx.VERTICAL)
    hSizer = wx.BoxSizer(wx.HORIZONTAL)
    audioSizer = self.buildAudioBar()

    # layout widgets
    mainSizer.Add(self.playbackSlider, 1, wx.ALL|wx.EXPAND,
5)
    hSizer.Add(audioSizer, 0, wx.ALL|wx.CENTER, 5)
    hSizer.Add(self.volumeCtrl, 0, wx.ALL, 5)
    mainSizer.Add(hSizer)

    self.SetSizer(mainSizer)
    self.Layout()

#-----
def buildAudioBar(self):

```

```

"""
Builds the audio bar controls
"""

audioBarSizer = wx.BoxSizer(wx.HORIZONTAL)

self.buildBtn({'bitmap': 'player_prev.png',
'handler': self.onPrev,
              'name': 'prev'},
              audioBarSizer)

# create play/pause toggle button
img = wx.Bitmap(os.path.join(bitmapDir,
"player_play.png"))
self.playPauseBtn = buttons.GenBitmapToggleButton(self,
bitmap=img, name="play")
self.playPauseBtn.Enable(False)

img = wx.Bitmap(os.path.join(bitmapDir,
"player_pause.png"))
self.playPauseBtn.SetBitmapSelected(img)
self.playPauseBtn.SetInitialSize()

self.playPauseBtn.Bind(wx.EVT_BUTTON, self.onPlay)
audioBarSizer.Add(self.playPauseBtn, 0, wx.LEFT, 3)

btnData = [{'bitmap': 'player_stop.png',
            'handler': self.onStop, 'name': 'stop'},
            {'bitmap': 'player_next.png',
            'handler': self.onNext, 'name': 'next'}]
for btn in btnData:
    self.buildBtn(btn, audioBarSizer)

return audioBarSizer

```

#-----

```

def buildBtn(self, btnDict, sizer):
    """
    bmp = btnDict['bitmap']
    handler = btnDict['handler']

    img = wx.Bitmap(os.path.join(bitmapDir, bmp))
    btn = buttons.GenBitmapButton(self, bitmap=img,

```

```

name=btnDict['name'])
    btn.SetInitialSize()
    btn.Bind(wx.EVT_BUTTON, handler)
    sizer.Add(btn, 0, wx.LEFT, 3)

#-----
def createMenu(self):
    """
    Creates a menu
    """
    menubar = wx.MenuBar()

    fileMenu = wx.Menu()
    open_file_menu_item = fileMenu.Append(wx.NewId(),
"&Open", "Open a File")
    menubar.Append(fileMenu, '&File')
    self.frame.SetMenuBar(menubar)
    self.frame.Bind(wx.EVT_MENU, self.onBrowse,
open_file_menu_item)

#-----
def loadMusic(self, musicFile):
    """
    Load the music into the MediaCtrl or display an error
dialog
    if the user tries to load an unsupported file type
    """
    if not self.mediaPlayer.Load(musicFile):
        wx.MessageBox("Unable to load %s: Unsupported
format?" % musicFile,
                        "ERROR",
                        wx.ICON_ERROR | wx.OK)
    else:
        self.mediaPlayer.SetInitialSize()
        self.GetSizer().Layout()
        self.playbackSlider.SetRange(0,
self.mediaPlayer.Length())
        self.playPauseBtn.Enable(True)

#-----

```

```

def onBrowse(self, event):
    """
    Opens file dialog to browse for music
    """
    wildcard = "MP3 (*.mp3)|*.mp3|" \
               "WAV (*.wav)|*.wav"
    dlg = wx.FileDialog(
        self, message="Choose a file",
        defaultDir=self.currentFolder,
        defaultFile="",
        wildcard=wildcard,
        style=wx.OPEN | wx.CHANGE_DIR
    )
    if dlg.ShowModal() == wx.ID_OK:
        path = dlg.GetPath()
        self.currentFolder = os.path.dirname(path)
        self.loadMusic(path)
    dlg.Destroy()

```

#-----

```

def onNext(self, event):
    """
    Not implemented!
    """
    pass

```

#-----

```

def onPause(self):
    """
    Pauses the music
    """
    self.mediaPlayer.Pause()

```

#-----

```

def onPlay(self, event):
    """
    Plays the music
    """
    if not event.GetIsDown():
        self.onPause()

```

```

        return

    if not self.mediaPlayer.Play():
        wx.MessageBox("Unable to Play media : Unsupported
format?",
                        "ERROR",
                        wx.ICON_ERROR | wx.OK)

    else:
        self.mediaPlayer.SetInitialSize()
        self.GetSizer().Layout()
        self.playbackSlider.SetRange(0,
self.mediaPlayer.Length())

        event.Skip()

#-----
def onPrev(self, event):
    """
    Not implemented!
    """
    pass

#-----
def onSeek(self, event):
    """
    Seeks the media file according to the amount the slider
has
    been adjusted.
    """
    offset = self.playbackSlider.GetValue()
    self.mediaPlayer.Seek(offset)

#-----
def onSetVolume(self, event):
    """
    Sets the volume of the music player
    """
    self.currentVolume = self.volumeCtrl.GetValue()
    print "setting volume to: %s" % int(self.currentVolume)
    self.mediaPlayer.SetVolume(self.currentVolume)

```

```

#-----
def onStop(self, event):
    """
    Stops the music and resets the play button
    """
    self.mediaPlayer.Stop()
    self.playPauseBtn.SetToggle(False)

#-----

def onTimer(self, event):
    """
    Keeps the player slider updated
    """
    offset = self.mediaPlayer.Tell()
    self.playbackSlider.SetValue(offset)

#####
#####
class MediaFrame(wx.Frame):

#-----

    def __init__(self):
        wx.Frame.__init__(self, None, wx.ID_ANY, "Python Music
Player")
        panel = MediaPanel(self)

#-----

# Run the program
if __name__ == "__main__":
    app = wx.App(False)
    frame = MediaFrame()
    frame.Show()
    app.MainLoop()

```

You might notice that this is actually version #2 of this application. In the downloadable source at the bottom of this article, I include the original which uses [Andrea Gavana's](#) ShapedButton widgets for the player's controls. I recommend that you get the [SVN version](#) of this widget from official wxPython repositories as there is a known bug in the version that is included with the default installation of wx. It

also depends on the bitmaps that are included with the wxPython Demo application, so you'll need to copy those to the appropriate location too.

Anyway, let's go over a few things in the code above. First off, we set up a couple of "global" variables that hold the application's directory path so we can find our bitmaps folder. Next, we create the application. Since a lot of people store their music in their Documents folder, we use `wx.StandardPaths` to find that location in a cross-platform way and set the `currentFolder` attribute to that location. We will use this attribute to store the last opened folder when we browse for music to listen to. We also set up a timer that is used to update the player's track slider when it's playing. This is copied verbatim from the wxPython's `MediaCtrl` demo.

In the `layoutControls` method, we create the necessary widgets and add them to the appropriate sizers as needed. We also do the widget event bindings here. The rest of the code handles other layout duties or the events that are generated. They are pretty self-explanatory as well. Note that the `onPrev` and `onNext` methods don't do anything. This is intended since we don't currently have a way to load more than one song at a time. We will add that functionality in a future version of the program.

So, the basic steps to play a song with this application are as follows:

1. Go to File and Open
2. Navigate to an MP3 file and open it
3. Press the Play button underneath the playback slider to start listening to your music

That's all there is to it. Now you have a fully functional music player. Yes, it is limited, but it should give you an idea of how powerful wxPython is and how easy it would be to extend this example to fit your needs.

Downloads

- [Music_Player.zip](#)
- [Music_Player.tar](#)

Comments for this thread are now closed.



7 Comments **Mouse Vs. the Python**

Login ▾

Recommend 3 Share

Sort by Best ▾



GrooveWalrus
6 years ago



Playing with [wx.media](#) recently, the above works fine in Windows 7. If say you wanted to load and play at the same time (ex. getting the first selection on a previously loaded playlist) eg.

```
def OnPlayClick(self, event):
...
self.mediaPlayer.Load("c:\some.mp3")
self.mediaPlayer.Play()
```

It won't play and it'll fill you with murderous rage.

What you need is 2 things, specify a backend and Bind a media loaded event:

```
self.mdaPlyr=wx.media.MediaCtrl(...szBackend=wx.media.MEDIABACKEND_WMP10)
self.Bind(wx.media.EVT_MEDIA_LOADED, self.PlayWxMedia)
```

2 ^ v



Ran4

6 years ago



It gives me an error, which I'm quite puzzled by.

Python25\lib\site-packages\wx-2.8-msw-unicode\wx_core.py", line 3440, in ImageFromBitmap
val = _core_.new_ImageFromBitmap(*args, **kwargs)
wx._core.PyAssertionError: C++ assertion "bmp.Ok()" failed at ..\..\src\msw\dib.cpp(148) in
wxDIB::Create(): wxDIB::Create(): invalid bitmap

Odd, I must say... What might be the fix?

1 ^ v



Mike Driscoll → Ran4

6 years ago



Are you using the archive download or did you just copy and paste the code from the post? That message means that it either can't find the bitmaps for the buttons or the bitmap is corrupt.

- Mike

1 ^ v



Mark Muzenhardt

6 years ago



There is an error in the Program. This function is to change so that the volume slider works properly:

```
def onSetVolume(self, event):
"""
Sets the volume of the music player
"""
self.currentVolume = self.volumeCtrl.GetValue()
print "setting volume to: %s" % int(self.currentVolume)
```

```
self.mediaPlayer.SetVolume(float(self.currentVolume) / 100)
```

0 ^ v



Mike Driscoll → Mark Muzenhardt

6 years ago



Oops...someone else showed me a similar solution to this quite some time ago, but I forgot to update the code in the post. Thanks for the heads-up!

- Mike

0 ^ v



haseeb ahmed

6 years ago



thanks you, very nice
going to watch image viewer now

0 ^ v



ProgMan (Mike)

6 years ago



Very nice work Mike, thanks for your effort.

ProgMan (Mike from the wxPy IRC channel)

0 ^ v

ALSO ON MOUSE VS. THE PYTHON

Reportlab – How to Add Charts / Graphs

1 comment • 2 months ago



optic — after we build PDF by reportlab how can print the document directly by python? are you have any standard method ?

Python 101: All about imports

2 comments • 2 months ago



Mike Driscoll — I actually meant to mention this method, but it slipped my mind. Thanks!

Python 201 – What's a deque?

4 comments • 4 days ago



Mike Driscoll — Yes, there should, but I didn't mean to copy that particular example in there twice. I removed the second occurrence.

wxPython's Context Managers

2 comments • 6 months ago



Mike Driscoll — Not sure what you mean by "ethical". That's where I saw the thread, so that's what I linked to.

