How to create 2 separate windows in WPF with C#

I want to create 2 separate windows in xaml and I want to control them separately from the code part. Do you have any idea how to do that ? If you can provide some code examples, I will be appreciated.

Thank you from now...

c# wpf windows

edited Dec 23 '11 at 15:37
marc_s
458k 93 881 1044

asked Dec 23 '11 at 14:08
Samet
78 1 3 13

2 Have you tried a GOOGLE SEARCH..? type the same question in google and see the results along with many examples. – MethodMan Dec 23 '11 at 14:11

Why not simply make two windows? Must they be in the same XAML file? If so, why? – Martin Dec 23 '11 at 14:11

I tried the google search. But it gave me the result like this: One big main window,and one small window inside of this window. What I want to try is to control them seperately at different locations. Not inside of the one window... @Martin: They dont have to be in the same XAML file, but they have to be controlled from one code. – Samet Dec 23 '11 at 15:41

1 Answer

Add a second `Window` (the first one being MainWindow.xaml) in your project (right click your project-> Add -> Window). Let's call it BobbyWindow.

In the constructor of MainWindow.xaml.cs, call:

```
BobbyWindow bWin = new BobbyWindow();
bwin.Owner = this;
bWin.Show();
```

voila.

EDIT: additional info to reflect the comments

The main difference between this:

```
public MainWindow()
{
   InitializeComponent();
   Window1 bWin = new Window1();
   bWin.Owner = this;
   bWin.Show();
}
```

And that:

```
Window1 bWin = new Window1();

public MainWindow()
    {
       InitializeComponent();

       bWin.Owner = this;
       bWin.Show();
    }
```

Is that in the **first case**, bWin is **local to the MainWindow()** constructor, which means it only exists within the brackets of MainWindow().
In the **second case**, bWin is **local to the class**, which means it is only accessible from within the boundaries of the MainWindow **class**

edited Jan 3 '12 at 12:40 answered Dec 23 '11 at 14:15

Louis Kottmann
11k 3 36 71

Sorry for the delay, but I use " this" for another window. I tried to give another name to call, but it didnt work. I'm not proffessional in this topic. If you explain me what to write, I will be appreciated. Thanks... – Samet Jan 2 '12 at 12:45

"this" refers to the current object. Depending on where you write "this" it points to a different object. – Louis Kottmann Jan 2 '12 at 13:00

I have one main window and your window. I make lots of things with main window by using "this" . So, I have to use something else to reach your window. I mean I cant do this: BobbyWindow.Visibility =.... It doesnt accept this. I use this.Visibility for main window. How can I call your window to reach it? – Samet Jan 2 '12 at 16:37

Oh, i see, put the first line outside the method, then you can do dbWin.Visibility = ... anywhere in your main window. – Louis Kottmann Jan 2 '12 at 21:34

Maybe I put this small code somewhere wrong. Where I put these 3 code lines is here: public MainWindow() { InitializeComponent(); Window1 bWin = new Window1(); bWin.Owner = this; bWin.Show(); } So, If I write like this, I cant reach to bWin from anywhere. Let me try to put first line out of the method. – Samet Jan 3 '12 at 9:15