

---

Subject: How can I detect Button press / release  
Posted by [jerson](#) on Sat, 23 Oct 2010 16:22:38 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Greetings everyone

Can someone point me to a callback for button which does the following

WhenPush - I know this one  
WhenRelease - does this exist?

I am trying to implement a feature which needs a particular pin on an I/O device to be high as long as the button is pushed/repeat action. The pin has to go low when the button is released.

How can I achieve this?

Thank you for your attention

---

---

Subject: Re: How can I detect Button press / release  
Posted by [dolik.rce](#) on Sat, 23 Oct 2010 18:20:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

jerson wrote on Sat, 23 October 2010 18:22Can someone point me to a callback for button which does the following

WhenPush - I know this one  
WhenRelease - does this exist?  
Hi Jerson,

WhenAction is called when the mouse button is released above the button widget. If you also need to detect when user moves the mouse pointer out of the widgets area (which you probably want, otherwise your I/O pin would stay in high state), then you can either inherit from Button and overwrite MouseLeave() to take care of that. Other solution (without making your own class) might be some ugly use of WhenRepeat with timecallbacks setting turning the state of pin after some time unless another WhenRepeat is called, but that would be ugly and error prone.

Best regards,  
Honza

---

---

Subject: Re: How can I detect Button press / release  
Posted by [koldo](#) on Sat, 23 Oct 2010 20:31:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello Jerson

I would do a new class from Button deriving virtual LeftDown() and LeftUp() methods.

---

---

Subject: Re: How can I detect Button press / release  
Posted by [jerson](#) on Sun, 24 Oct 2010 02:18:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Koldo

I'm a bit fuzzy on this one. Can you help me a bit more as to how to do this? This is how I did it now

```
class JfButton : public Button {  
public:  
    virtual void LeftDown(Point, dword);  
    virtual void LeftUp(Point, dword);  
};
```

```
and in my App, this  
void JfButton::LeftDown(Point p, dword dw)  
{  
    SetLabel("Pushed");  
}
```

```
void JfButton::LeftUp(Point p, dword dw)  
{  
    SetLabel("Released");  
}
```

I understand Honza's point about releasing the button outside the widget rectangle and can see it hang. Surely I need to look at how to release when the Mouse leaves the widget rectangle.

Edit:

I no longer see the button being redrawn to pushed state. Is there something more that needs to be done to get this?

Regards

---

---

Subject: Re: How can I detect Button press / release  
Posted by [dolik.rce](#) on Sun, 24 Oct 2010 08:24:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Jerson,

This should take care of everything:

```
class JfButton : public Button {
public:
    virtual void LeftDown(Point, dword);
    virtual void LeftUp(Point, dword);
    virtual void MouseLeave();
    virtual void MouseEnter(Point, dword);
};
void JfButton::LeftDown(Point p, dword dw){
    SetLabel("Pushed");
}
void JfButton::LeftUp(Point p, dword dw){
    SetLabel("Released");
}
void JfButton::MouseLeave(){
    SetLabel("Released");
    Button::MouseLeave();
}
void JfButton::MouseEnter(Point p,dword dw){
    if(dw&K_MOUSELEFT){SetLabel("Pushed");}
    Button::MouseEnter(p,dw);
}
```

Honza

---

---

Subject: Re: How can I detect Button press / release  
Posted by [jerson](#) on Sun, 24 Oct 2010 12:34:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Honza

thanks for your example code. I really appreciate your help.

I was getting some strange behaviour on Push and Release like having to move the mouse to see the button push painting or moving the mouse after releasing the button to see button release painting.

This is what made it work

```
void JfButton::LeftDown(Point p, dword dw){
    SetLabel("Pushed");
    KeyPush(); // draw the button push state
}
void JfButton::LeftUp(Point p, dword dw){
    SetLabel("Released");
    FinishPush(); // draw button release state
}
```

Regards

---