
Subject: Re: How to use LineEdit's default "Key" handler? no subclass

Posted by [Lance](#) on Fri, 04 Nov 2022 01:04:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Wakhshti,

I don't think these widgets support key event handler, so you will have to subclass them. However you can provide one in your subclass, then all objects of your subclass will be able to supply individual customised key event handler.

Here is a quick example. Let me know if you need further explanations.

Regards,

Lance

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
// subclass EditString, and define an Event(actually a Gate)
```

```
class MyEditString : public EditString
{
    bool Key(dword key, int rep)override
    {
        return (WhenKey && WhenKey(key, rep) ) ||
            this->EditString::Key(key, rep);
    }
public:
    Gate<dword, int> WhenKey;
};
```

```
// actually use it
```

```
class MyWin : public TopWindow
```

```
{
public:
    MyWin()
    {
        // the event handler will make the edit throw away key 'A';
        e.WhenKey = [=, this](dword key, int){
            return key == 'a' || key == 'A';
        };
    }
};
```

```
// the event handler will make the edit filter out digit key
```

```
f.WhenKey = [=, this](dword key, int){
    return key >='0' && key <='9';
};
```

```
Add(e.TopPosZ(5).LeftPosZ(5));  
Add(f.TopPosZ(35).LeftPosZ(5));  
}
```

```
MyEditString e, f;  
};
```

```
GUI_APP_MAIN  
{  
    MyWin().Run();  
}
```
