
Subject: callbacks and objects

Posted by [sapiency](#) on Wed, 05 Nov 2008 22:06:21 GMT

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

Hi,

I want to use one callback for multiple elements. Is it possible to use the element in the callback?

example:

```
class TEST
{
...
public:
typedef TEST CLASSNAME;

TEST();

void OnChange();

EditString s1;
EditString s2;
...
};

TEST::TEST()
{
Add(s1);
Add(s2);
s1.WhenAction = THISBACK(OnChange);
s2.WhenAction = THISBACK(OnChange);
...
}

TEST::OnChange()
{
OUTPUT(sx.GetData());
}
```

If I use THISBACK1 I got the message that the Upp::Ctrl& is private. If I use sn in the Callback, it works, but then I need one callback for each field ...

second:

I look for catching key events from special keys (return, Fn, ...). How can I catch them?

regards reinhard

Subject: Re: callbacks and objects

Posted by [mirek](#) on Thu, 06 Nov 2008 08:28:06 GMT

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

sapiency wrote on Wed, 05 November 2008 17:06Hi,

I want to use one callback for multiple elements. Is it possible to use the element in the callback?

example:

```
class TEST
{
...
public:
    typedef TEST CLASSNAME;

    TEST();

    void OnChange();

    EditString s1;
    EditString s2;
    ...
};

TEST::TEST()
{
    Add(s1);
    Add(s2);
    s1.WhenAction = THISBACK(OnChange);
    s2.WhenAction = THISBACK(OnChange);
    ...
}

TEST::OnChange()
{
    OUTPUT(sx.GetData());
}
```

If I use THISBACK1 I got the message that the Upp::Ctrl& is private. If I use sn in the Callback, it works, but then I need one callback for each field ...

I am sorry, but it is not quite clear what you are really doing

Use THISBACK1 if you want to pass some parameter with callback (specified inside THISBACK1, like

```
void TEST::OnChange(EditField *sx) { OUTPUT(sx->GetData()); }  
....  
s1 <<= THISBACK1(OnChange, &s1);  
s2 <<= THISBACK1(OnChange, &s2);
```

Quote:

I look for catching key events from special keys (return, Fn, ...). How can I catch them?

Override Ctrl::Key or Ctrl::HotKey. Check propagation rules in Ctrl reference.

Mirek
