
Subject: how to define hot-keys not associated with any menu?

Posted by [White_Owl](#) on Wed, 28 Jan 2009 21:53:50 GMT

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

I have my own widget(s) inside TopWindow, this widget should react to hot-keys. Only widget itself knows what these keys are.

So I am trying to do:

```
class MainWindow: public TopWindow {
    BaseWidget widget*;
};
void MainWindow::ChooseWidget() {
    widget = WidgetFactory(widget_id);
    this.Add(&*widget);
    widget->SetupHotKeys(this);
}
Widgets are defined like this:
class BaseWidget: public Ctrl {
public:
    virtual void SetupHotKeys(TopWindow &tw) = 0;
};
class RealWidget: public BaseWidget {
    void foo1();
    void foo2();
    void SetupHotKeys(TopWindow &tw) {
        tw.SomeKindOfAssignKey(K_A, THISBACK(foo1));
        tw.SomeKindOfAssignKey(K_SHIFT_A, THISBACK(foo2));
    }
};
```

So the question is what is the name of the function which does actual key assignments? Am I going in the right direction? Any tutorial or examples with similar functionality?

Subject: Re: how to define hot-keys not associated with any menu?

Posted by [mirek](#) on Sun, 08 Feb 2009 07:18:18 GMT

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

IMO you can start here:

```
dword Switch::GetAccessKeys() const
{
    dword keys = 0;
    for(int i = 0; i < cs.GetCount(); i++)
        keys |= AccessKeyBit(cs[i].accesskey);
    return keys;
}

void Switch::AssignAccessKeys(dword used)
{
```

```
for(int i = 0; i < cs.GetCount(); i++) {
    Case& v = cs[i];
    if(!v.accesskey) {
        v.accesskey = ChooseAccessKey(v.label, used);
        if(v.accesskey) Refresh();
        used |= AccessKeyBit(v.accesskey);
    }
}
Ctrl::AssignAccessKeys(used);
}
```

Note that U++ normally assigns hotkeys automatically, but you can override that using venerable "&" (ChooseAccessKey takes care about all of that).

Mirek
