
Subject: Re: DropList events

Posted by [Sender Ghost](#) on Wed, 08 Aug 2012 03:29:31 GMT

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

Hello, Nick.

nixnix wrote on Tue, 07 August 2012 21:45: Is there any way to disable this mousewheel functionality for some DropLists or even better would be if I can still generate the two events and have it work like it does just now but with the mousewheel?

Yes, there is. It is possible to create custom DropList with your requirements.

You need to override MouseWheel and Key (because, there are also up, down, left and right keys to select values) virtual DropList methods.

A possible implementation is as follows:

Toggle Spoiler

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

```
using namespace Upp;
```

```
// Choose the variant
```

```
#if 1
```

```
class CustomDropList : public DropList {
```

```
public:
```

```
virtual bool Key(dword key, int count)
```

```
{  
    if (!IsPopUp() && key != K_ALT_DOWN)  
        return false;
```

```
    return DropList::Key(key, count);
```

```
}
```

```
virtual void MouseWheel(Point p, int zdelta, dword keyflags)
```

```
{  
    if (!IsPopUp())  
        return;
```

```
    DropList::MouseWheel(p, zdelta, keyflags);
```

```
}
```

```
};
```

```
#else
```

```
class CustomDropList : public DropList {
```

```
public:
```

```
virtual bool Key(dword key, int count)
```

```
{  
    if (!IsPopUp() && GetCount())  
        if (key == K_LEFT || key == K_RIGHT || key == K_DOWN || key == K_UP)  
            WhenDrop();  
    else  
        if (key >= 32 && key < 65536)
```

```

    return false;

    return DropList::Key(key, count);
}

virtual void MouseWheel(Point p, int zdelta, dword keyflags)
{
    if (!IsPopUp() && GetCount())
        WhenDrop();

    DropList::MouseWheel(p, zdelta, keyflags);
}
};
#endif

class App : public TopWindow {
public:
    typedef App CLASSNAME;
    App();

    ArrayCtrl event;
    CustomDropList list;

    void OnDrop();
    void OnAction();
};

App::App()
{
    Title("Custom DropList example");
    Sizeable().Zoomable();
    const Size sz(320, 240);
    SetRect(sz); SetMinSize(sz);

    event.NoWantFocus();
    event.AddRowNumColumn("#").HeaderTab().Fixed(50);
    event.AddColumn("Event");

    list.WhenDrop = THISBACK(OnDrop);
    list.WhenAction = THISBACK(OnAction);

    for (int i = 1; i <= 100; ++i)
        list.Add(FormatIntRoman(i, true));

    Add(event.VSizePosZ(4, 27).HSizePosZ(4, 4));
    Add(list.BottomPosZ(4, 19).HSizePosZ(4, 4));
}

```

```
const int rowCount = 1000;

void App::OnDrop()
{
    GuiLock __;
    if (event.GetCount() > rowCount)
        event.Clear();

    event.Add("WhenDrop");
    event.GoEnd();
}

void App::OnAction()
{
    GuiLock __;
    if (event.GetCount() > rowCount)
        event.Clear();

    event.Add("WhenAction");
    event.GoEnd();
}

GUI_APP_MAIN
{
    Ctrl::GlobalBackPaint();

    App app;
    app.Run();
}
```
