
Subject: CtrlBinder as an alternative for CtrlRetriever
Posted by [Sender Ghost](#) on Thu, 15 Dec 2011 08:28:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

There is useful class to retrieve value(s) from assigned Ctrl(s), named CtrlRetriever.

But there are situations, where you need to (re)assign value(s) to Ctrl(s), e.g. to cancel changes. Which leads to creation of CtrlBinder class, based on CtrlRetriever implementation:

Toggle Spoiler

```
class CtrlBinder {
public:
    struct Item {
        virtual void Assign() = 0;
        virtual void Retrieve() = 0;
        virtual ~Item() {}
    };

private:
    struct CtrlItem0 : Item {
        Ctrl *ctrl;
    };

    template <class T>
    struct CtrlItem : CtrlItem0 {
        T *value;

        virtual void Assign() { *ctrl <<= *value; }
        virtual void Retrieve() { *value = ~*ctrl; }
        virtual ~CtrlItem() {}
    };

protected:
    Array<Item> item;

public:
    void Put(Item *newitem) { item.Add(newitem); }

    template <class T>
    void Bind(Ctrl& ctrl, T& val, bool assign = false);

    template <class T>
    CtrlBinder& operator()(Ctrl& ctrl, T& val, bool assign = false) {
        Bind(ctrl, val, assign); return *this;
    }

    void Assign();
    void Retrieve();
};
```

```

Callback operator<<=(Callback cb);
};

template <class T>
void CtrlBinder::Bind(Ctrl& ctrl, T& val, bool assign)
{
    CtrlItem<T> *m = new CtrlItem<T>();
    m->ctrl = &ctrl;
    m->value = &val;
    if (assign)
        ctrl <<= val;
    Put(m);
}

Callback CtrlBinder::operator<<=(Callback cb)
{
    for(int i = 0; i < item.GetCount(); i++) {
        CtrlItem0 *m = dynamic_cast<CtrlItem0 *>(&item[i]);
        if(m)
            m->ctrl->WhenAction = cb;
    }
    return cb;
}

void CtrlBinder::Assign()
{
    for(int i = 0; i < item.GetCount(); i++)
        item[i].Assign();
}

void CtrlBinder::Retrieve()
{
    for(int i = 0; i < item.GetCount(); i++)
        item[i].Retrieve();
}

```

The main difference: CtrlBinder doesn't assign value to Ctrl, during the binding, by default. There is Assign method for this.

Therefore, CtrlBinder could be used in either direction - to assign or retrieve values.

In the attachment you could find TabCtrl serialize example, with using CtrlBinder to save/cancel changes.

Edit: Fixed include path to layout file for GCC compiler.

File Attachments

1) [TabCtrlExample.zip](#), downloaded 463 times
