

---

Subject: Editing TreeCtrl Node Value

Posted by [bushman](#) on Tue, 22 Nov 2011 18:24:14 GMT

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

---

Is there a way of changing a TreeCtrl Node Value, say, the same way as you do in an ArrayCtrl Column, when you click a second time on a column previously assigned for editing through ArrayCtrl::Edit(Ctrl& ctrl)?

I couldn't find a corresponding method for TreeCtrl, such as TreeCtrl::Node::Edit(Ctrl& ctrl).

Any suggestions on how I should go about it, like a code sample ?

Thanks!

---

Subject: Re: Editing TreeCtrl Node Value

Posted by [bushman](#) on Fri, 25 Nov 2011 00:50:37 GMT

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

---

I improvised with the following code:

.h:

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

private:
    struct TreeEdit : public EditString {
        Callback WhenLostFocus;
        TreeEdit() {}
        virtual void LostFocus() { WhenLostFocus(); }
    };

    bool        isedit;
    int         nodeid;
    TreeCtrl::Node editnode, node;
    TreeEdit    edit;
    TreeCtrl    tree;

    void EditBranch();
    void CloseEdit();
};
```

.cpp:

```
App::App() : isedit(false), nodeid(-1) {
    Add(tree.SizePos());
    /*
    ...
    Populate TreeCtrl here
    ...
    */
    tree.WhenLeftClick = THISBACK(EditBranch);
    editnode.SetCtrl(edit);
    edit.WhenEnter = THISBACK(CloseEdit);
    edit.WhenLostFocus = THISBACK(CloseEdit);
}
```

```
void App::EditBranch()
{
    if(isedit) {
        CloseEdit();
        return;
    }
    if(nodeid != tree.GetCursor()) {
        nodeid = tree.GetCursor();
        return;
    }
    isedit = true;
    node = tree.GetNode(nodeid);
    editnode.ctrl->SetData(node.value);
    editnode.image = node.image;
    groups.SetNode(nodeid, editnode);
    editnode.ctrl->Show();
}
```

```
void App::CloseEdit()
{
    node.value = editnode.ctrl->GetData();
    tree.SetNode(nodeid, node);
    tree.SelectOne(nodeid);
    editnode.ctrl->Hide();
    isedit = false;
}
```

It works in a way, but I guess there must be an easier way of accomplishing this.

thanks for any help.

---

---

Subject: Re: Editing TreeCtrl Node Value

Posted by [mirek](#) on Fri, 25 Nov 2011 12:49:21 GMT

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

---

kropniczki wrote on Thu, 24 November 2011 19:50I improvised with the following code:

.h:

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

private:
    struct TreeEdit : public EditString {
        Callback WhenLostFocus;
        TreeEdit() {}
        virtual void LostFocus() { WhenLostFocus(); }
    };

    bool        isedit;
    int         nodeid;
    TreeCtrl::Node editnode, node;
    TreeEdit edit;
    TreeCtrl    tree;

    void EditBranch();
    void CloseEdit();
};
```

.cpp:

```
App::App() : isedit(false), nodeid(-1) {
    Add(tree.SizePos());
    /*
    ...
    Populate TreeCtrl here
    ...
    */
    tree.WhenLeftClick = THISBACK(EditBranch);
    editnode.SetCtrl(edit);
    edit.WhenEnter = THISBACK(CloseEdit);
    edit.WhenLostFocus = THISBACK(CloseEdit);
}

void App::EditBranch()
{
    if(isedit) {
```

```

    CloseEdit();
    return;
}
if(nodeid != tree.GetCursor()) {
    nodeid = tree.GetCursor();
    return;
}
isedit = true;
node = tree.GetNode(nodeid);
editnode.ctrl->SetData(node.value);
editnode.image = node.image;
groups.SetNode(nodeid, editnode);
editnode.ctrl->Show();
}

void App::CloseEdit()
{
    node.value = editnode.ctrl->GetData();
    tree.SetNode(nodeid, node);
    tree.SelectOne(nodeid);
    editnode.ctrl->Hide();
    isedit = false;
}

```

It works in a way, but I guess there must be an easier way of accomplishing this.

thanks for any help.

Unfortunately, this mode of edit is not directly supported (yet?).

**Subject: Re: Editing TreeCtrl Node Value**

Posted by [bushman](#) on Fri, 25 Nov 2011 14:31:31 GMT

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

Quote:Unfortunately, this mode of edit is not directly supported (yet?).

All right, thank you, now I know officially it is not supported (yet), so I'll try to cope with what we got for now at least, for I read in your question mark maybe it might be included in a future version of TreeCtrl, right?

Thanks.