

I think this might do what you need:

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

using namespace Upp;

class EditableTree : public TreeCtrl {
    Array<EditString> edits;

public:
    typedef EditableTree CLASSNAME;

    EditableTree() {
        NoRoot().HighlightCtrl(false);
    }

    int AddNode(int parent_id) {
        EditString& edit = edits.Add();
        edit.AutoSize().SetFrame(NullFrame());
        static int i;
        edit <<= "NODE" + AsString(GetChildCount(parent_id) + 1);
        return Add(parent_id, Null, edit);
    }

    void OnAddNode() {
        if(!IsCursor())
            return;
        SetCursor(AddNode(GetCursor()));
    }
};

GUI_APP_MAIN
{
    TopWindow wnd;

    Button btn;
    wnd.Add(btn.SetLabel("Add").LeftPos(10).TopPos(10));
    Button btn2;
    wnd.Add(btn2.SetLabel("Show").LeftPos(100).TopPos(10));

    EditableTree tree;
    wnd.Add(tree.VSizePos(40).HSizePos());
}
```

```
tree.AddNode(0);  
tree.AddNode(0);  
btn <<= callback(&tree, &EditableTree::OnAddNode);  
    wnd.Run();  
}
```

I apologize for delay - I had to fix some subtle focus issues in TreeCtrl for this particular mode of operation... (so you need latest sources for it to work correctly).

Mirek
