
Subject: Missing callback trigger in OptionTree? [SOLVED/FIXED]

Posted by [James Thomas](#) on Thu, 01 Jun 2006 09:29:13 GMT

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

I've got an OptionTree in my application and I would like to dynamically repond to state changes (ie: an option being checked).

This cannot be accomplished by assigning a callback to each Option control because OptionTree used this callback to implement it's hierarchical checking (when you check a parent option it also checks it's children).

Now, OptionTree has a Callback called WhenOption that is never triggered. I suspect that this was intended to be used for the purpose of allowing a response to a state change but was forgotten.

I believe it should be put into OptionCtrl::SetOption as follows:

```
void OptionTree::SetOption(int id)
{
    Option *opt = option[id];
    ASSERT(opt);
    SetChildren(id, opt->Get());
    for(;;) {
        id = GetParent(id);
        if(id < 0)
            break;
        bool t = false;
        bool f = false;
        bool n = false;
        for(int i = 0; i < GetChildCount(id); i++) {
            int chid = GetChild(id, i);
            Option *opt = option[chid];
            if(opt) {
                int x = opt->Get();
                if(x == 1)
                    t = true;
                else
                    if(x == 0)
                        f = true;
                    else
                        n = true;
            }
        }
    }
    opt = option[id];
    if(opt) {
        if(t && f || n) {
            opt->Set(Null);
        }
    }
    else
```

```

    opt->Set(t);
}
}
WhenOption(); <<----- This is the addition
}

```

With the extra line the control now works as I would expect. I can't see any other reason for WhenOption, so is this how it should work?

Subject: Re: Missing callback trigger in OptionTree?

Posted by [mirek](#) on Thu, 01 Jun 2006 19:09:39 GMT

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

James Thomas wrote on Thu, 01 June 2006 05:29: I've got an OptionTree in my application and I would like to dynamically repond to state changes (ie: an option being checked).

This cannot be accomplished by assigning a callback to each Option control because OptionTree used this callback to implement it's hierarchical checking (when you check a parent option it also checks it's children).

Now, OptionTree has a Callback called WhenOption that is never triggered. I suspect that this was intended to be used for the purpose of allowing a response to a state change but was forgotten.

I believe it should be put into OptionCtrl::SetOption as follows:

```

void OptionTree::SetOption(int id)
{
    Option *opt = option[id];
    ASSERT(opt);
    SetChildren(id, opt->Get());
    for(;;) {
        id = GetParent(id);
        if(id < 0)
            break;
        bool t = false;
        bool f = false;
        bool n = false;
        for(int i = 0; i < GetChildCount(id); i++) {
            int chid = GetChild(id, i);
            Option *opt = option[chid];
            if(opt) {
                int x = opt->Get();
                if(x == 1)
                    t = true;
                else
                    if(x == 0)
                        f = true;
            }
        }
    }
}

```

```
    else
        n = true;
    }
}
opt = option[id];
if(opt) {
    if(t && f || n) {
        opt->Set(Null);
    }
    else
        opt->Set(t);
}
}
WhenOption(); <<----- This is the addition
}
```

With the extra line the control now works as I would expect. I can't see any other reason for WhenOption, so is this how it should work?

You are right! Thanks!

Mirek
