Subject: Anomaly in tree control
Posted by gprentice on Tue, 16 Nov 2010 10:38:25 GMT
View Forum Message <> Reply to Message

I just noticed what seems to be a problem in the tree control. I was trying the tree control reference example and with the tree showing a directory I was testing to see whether the tree remembered the collapsed/ uncollapsed state of lower level nodes when the top level node was uncollapsed -> collapsed -> uncollapsed. I found the lower level plus/minus icon then showed as uncollapsed (minus) but the node wasn't actually expanded.
i.e.
start with tree all collapsed
use right arrow key to expand first level
use down arrow key to go to first child
use right arrow key to expand first child
use up arrow key to go back to root node
use left arrow key to collapse root node
use right arrow key to expand root node

now you should see the problem where the first child node has a minus icon but is uncollapsed. Sometimes the problem doesn't appear the first time.

Graeme

Subject: Re: Anomaly in tree control
Posted by mrjt on Tue, 16 Nov 2010 10:43:31 GMT
View Forum Message <> Reply to Message

Is this not a specific problem with that example? If I remember correctly it creates the entries for sub-trees on demand (when you click the plus) and removes then when it's closed. Therefor it will not remember the openness of nested trees.

I'm pretty sure it works correctly for more normal situations.

Subject: Re: Anomaly in tree control
Posted by mrjt on Tue, 16 Nov 2010 10:57:33 GMT
View Forum Message <> Reply to Message

I does appear to be a bug. The TreeCtrl uses a pool of nodes to reduce allocation times but doesn't reset Node::isopen on a new Insert. In this case that means that the node thinks it's open but iot doesn't have a sub-tree because the WhenOpen callback hasn't been triggered.

The additon of:
m.isopen = false;
to TreeCtrl::Insert fixes the problem.

This example will still not be able to remember the openness of nest trees though. If you want dynamic sub-tree creation while remembering the opneness of sub-trees you'll have to remember fodler states yourself.

Subject: Re: Anomaly in tree control
Posted by gprentice on Tue, 16 Nov 2010 11:00:37 GMT
View Forum Message <> Reply to Message

Not sure.
close directory in the example is this

```
  void CloseDir(int id) {
    tree1.RemoveChildren(id);
  }
```

I changed it to this but it didn't fix the problem

```
  void CloseDir(int id) {
    tree1.Close(id);
    tree1.RemoveChildren(id);
  }
```

so I guess the minus sign is showing there's no children.
I'm wondering if there should be no plus or minus icon on a node that has no children - but then for this example you wouldn't be able to re-populate the node so perhaps remove-children isn't quite the normal thing to do in CloseDir.

[Edit] - oops, didn't see your second answer before I posted this. Thanks.

Subject: Re: Anomaly in tree control
Posted by mirek on Wed, 01 Dec 2010 18:40:23 GMT
View Forum Message <> Reply to Message

mrjt wrote on Tue, 16 November 2010 05:57m.isopen = false;
to TreeCtrl::Insert fixes the problem.

Thanks, patch applied.
