

Hopefully, this should fix the second problem:

```
void TreeCtrl::SyncTree()
{
    if(!dirty)
        return;
    if(noroot)
        Open(0);
    Ptr<Ctrl> restorefocus = GetFocusChildDeep();
    hasctrls = false;
    int cursorid = GetCursor();
    for(int i = 0; i < item.GetCount(); i++)
        item[i].linei = -1;
    line.Clear();
    Size treesize = Size(0, 0);
    if(noroot) {
        if(GetChildCount(0))
            treesize.cy = -item[0].GetSize().cy;
        ReLine(0, -1, treesize);
    }
    else
        ReLine(0, 0, treesize);
    treesize.cy = max(0, treesize.cy);
    treesize.cx += levelcx;
    sb.SetTotal(treesize);
    cursor = -1;
    dirty = false;
    if(cursorid >= 0)
        SetCursor(cursorid, false, false, false);
    SyncCtrls(true, restorefocus);
    SyncInfo();
}

void TreeCtrl::SetCursorLine(int i, bool sc, bool sel, bool cb)
{
    if(nocursor)
        return;
    if(sel && multiselect) {
        ClearSelection();
        SelectOne(line[i].itemi, true);
    }
    if(i != cursor) {
```

```

i = minmax(i, 0, line.GetCount() - 1);
if(i < 0) return;
Item& m = item[line[i].itemi];
if(sc)
    sb.ScrollIntoY(line[i].y, m.GetSize().cy);
RefreshLine(cursor);
cursor = i;
RefreshLine(cursor);
if(m.ctrl && m.ctrl->SetWantFocus())
    return;
if(cb) {
    WhenCursor();
    WhenSel();
}
}
}

void TreeCtrl::SetCursor(int id, bool sc, bool sel, bool cb)
{
    while(id > 0) {
        ASSERT(id >= 0 && id < item.GetCount());
        MakeVisible(id);
        SyncTree();
        const Item& m = item[id];
        if(m.linei >= 0) {
            SetCursorLine(m.linei, sc, sel, cb);
            return;
        }
        id = m.parent;
    }
    SetCursorLine(0, sc, sel, cb);
}

```
