
Subject: AcceptRow broken when SetCtrl is used
Posted by [mrjt](#) on Wed, 09 Jul 2008 13:41:28 GMT

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

ArrayCtrl::AcceptRow doesn't check any child ctrls set with SetCtrl() when accepting rows, which in turn can lead to dialogs being accepted with invalid data (a big problem!). A quick test case:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;
```

```
GUI_APP_MAIN
```

```
{
  TopWindow wnd;
  ArrayCtrl ctrl;
  Button ok;
  Button cancel;
  Array<EditInt> edit;

  Rect r = Rect(0, 0, 200, 300);
  wnd.Sizeable().SetMinSize(r.GetSize());
  wnd.SetRect(r + GetScreenSize() / 2 - r.GetSize()/2);
  wnd.Add(ctrl.HSizePos().VSizePos(0, 28));
  wnd.Add(ok.SetLabel("OK").RightPos(88, 80).BottomPos(4, 22));
  wnd.Add(cancel.SetLabel("Cancel").RightPos(4, 80).BottomPos(4, 22));
  wnd.Acceptor(ok, IDOK);
  wnd.Rejector(cancel, IDCANCEL);

  edit.SetCount(20);
  ctrl.AddColumn("EditInts (0-1)");
  ctrl.SetCount(20);
  for (int i = 0; i < edit.GetCount(); i++) {
    edit[i].Min(0).Max(1) <<= 1;
    ctrl.SetCtrl(i, 0, edit[i]);
  }

  wnd.Execute();
}
```

Just type 2 into one of the rows and then change row, this should trigger an Exclamation but doesn't. You will be also able to 'OK' the form with the invalid data still in the ctrl.

I've attempted to fix this myself but just adding a check in AcceptRow causes a stack overflow due to focus changes when an Exclamation is shown by the child's Accept() function, so I'll leave it to someone with better knowledge of ArrayCtrl.

I'd settle for just checking all the ctrls in the ArrayCtrl's Accept function if fixing AcceptRow is impossible/very difficult.

Cheers.

Subject: Re: AcceptRow broken when SetCtrl is used

Posted by [mirek](#) on Fri, 11 Jul 2008 10:36:13 GMT

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

mrjt wrote on Wed, 09 July 2008 09:41 ArrayCtrl::AcceptRow doesn't check any child ctrls set with SetCtrl() when accepting rows, which in turn can lead to dialogs being accepted with invalid data (a big problem!). A quick test case:

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

```
using namespace Upp;
```

```
GUI_APP_MAIN
```

```
{
  TopWindow wnd;
  ArrayCtrl ctrl;
  Button ok;
  Button cancel;
  Array<EditInt> edit;

  Rect r = Rect(0, 0, 200, 300);
  wnd.Sizeable().SetMinSize(r.GetSize());
  wnd.SetRect(r + GetScreenSize() / 2 - r.GetSize()/2);
  wnd.Add(ctrl.HSizePos().VSizePos(0, 28));
  wnd.Add(ok.SetLabel("OK").RightPos(88, 80).BottomPos(4, 22));
  wnd.Add(cancel.SetLabel("Cancel").RightPos(4, 80).BottomPos(4, 22));
  wnd.Acceptor(ok, IDOK);
  wnd.Rejector(cancel, IDCANCEL);
```

```
  edit.SetCount(20);
  ctrl.AddColumn("EditInts (0-1)");
  ctrl.SetCount(20);
  for (int i = 0; i < edit.GetCount(); i++) {
    edit[i].Min(0).Max(1) <= 1;
    ctrl.SetCtrl(i, 0, edit[i]);
  }
```

```
  wnd.Execute();
}
```

Just type 2 into one of the rows and then change row, this should trigger an Exclamation but doesn't. You will be also able to 'OK' the form with the invalid data still in the ctrl.

I've attempted to fix this myself but just adding a check in AcceptRow causes a stack overflow due to focus changes when an Exclamation is shown by the child's Accept() function, so I'll leave it to someone with better knowledge of ArrayCtrl.

I'd settle for just checking all the ctrls in the ArrayCtrl's Accept function if fixing AcceptRow is impossible/very difficult.

Cheers.

This seems to fix the problem:

```
bool ArrayCtrl::AcceptRow() {
    ASSERT(IsCursor());
    int i;
    for(i = 0; i < column.GetCount(); i++) {
        Column& m = column[i];
        if(m.edit && !m.edit->Accept())
            return false;
        if(IsCtrl(cursor, i)) {
            Ctrl *c = GetCtrl(cursor, i).ctrl;
            acceptingrow++;
            bool b = c->Accept();
            acceptingrow--;
            if(!b)
                return false;
        }
    }
    for(i = 0; i < control.GetCount(); i++)
        if(!control[i].ctrl->Accept())
            return false;
    acceptingrow++;
    bool ar = WhenAcceptRow() && UpdateRow();
    acceptingrow--;
    if(!ar) {
        SetCursorEditFocus();
        return false;
    }
    bool b = editmode;
    EndEdit();
    SetCtrls();
    ClearModify();
    if(b)
        WhenAcceptEdit();
    return true;
}
```

Not that I would like such fragile code

Mirek

Subject: Re: AcceptRow broken when SetCtrl is used
Posted by [mrjt](#) on Fri, 30 Jan 2009 12:14:49 GMT

luzr wrote on Fri, 11 July 2008 11:36mrjt wrote on Wed, 09 July 2008 09:41ArrayCtrl::AcceptRow doesn't check any child ctrls set with SetCtrl() when accepting rows, which in turn can lead to dialogs being accepted with invalid data (a big problem!). A quick test case:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;
```

```
GUI_APP_MAIN
```

```
{
  TopWindow wnd;
  ArrayCtrl ctrl;
  Button ok;
  Button cancel;
  Array<EditInt> edit;

  Rect r = Rect(0, 0, 200, 300);
  wnd.Sizeable().SetMinSize(r.GetSize());
  wnd.SetRect(r + GetScreenSize() / 2 - r.GetSize()/2);
  wnd.Add(ctrl.HSizePos().VSizePos(0, 28));
  wnd.Add(ok.SetLabel("OK").RightPos(88, 80).BottomPos(4, 22));
  wnd.Add(cancel.SetLabel("Cancel").RightPos(4, 80).BottomPos(4, 22));
  wnd.Acceptor(ok, IDOK);
  wnd.Rejector(cancel, IDCANCEL);

  edit.SetCount(20);
  ctrl.AddColumn("EditInts (0-1)");
  ctrl.SetCount(20);
  for (int i = 0; i < edit.GetCount(); i++) {
    edit[i].Min(0).Max(1) <<= 1;
    ctrl.SetCtrl(i, 0, edit[i]);
  }

  wnd.Execute();
}
```

Just type 2 into one of the rows and then change row, this should trigger an Exclamation but doesn't. You will be also able to 'OK' the form with the invalid data still in the ctrl.

I've attempted to fix this myself but just adding a check in AcceptRow causes a stack overflow due to focus changes when an Exclamation is shown by the child's Accept() function, so I'll leave it to someone with better knowledge of ArrayCtrl.

I'd settle for just checking all the ctrls in the ArrayCtrl's Accept function if fixing AcceptRow is impossible/very difficult.

Cheers.

This seems to fix the problem:

```

bool ArrayCtrl::AcceptRow() {
    ASSERT(IsCursor());
    int i;
    for(i = 0; i < column.GetCount(); i++) {
        Column& m = column[i];
        if(m.edit && !m.edit->Accept())
            return false;
        if(IsCtrl(cursor, i)) {
            Ctrl *c = GetCtrl(cursor, i).ctrl;
            acceptingrow++;
            bool b = c->Accept();
            acceptingrow--;
            if(!b)
                return false;
        }
    }
    for(i = 0; i < control.GetCount(); i++)
        if(!control[i].ctrl->Accept())
            return false;
    acceptingrow++;
    bool ar = WhenAcceptRow() && UpdateRow();
    acceptingrow--;
    if(!ar) {
        SetCursorEditFocus();
        return false;
    }
    bool b = editmode;
    EndEdit();
    SetCtrls();
    ClearModify();
    if(b)
        WhenAcceptEdit();
    return true;
}

```

Not that I would like such fragile code

Mirek

This is either broken again or was never fixed, the code in the repo doesn't match the code above.

Subject: Re: AcceptRow broken when SetCtrl is used

Posted by [mirek](#) on Sun, 08 Feb 2009 09:40:33 GMT

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

Not sure what went wrong. It is back again.

If you have some time to check if it is fixed now, would be great...

Mirek
