Subject: How to get edit data in virtual ArrayCtrl?
Posted by steffen on Wed, 30 May 2007 21:21:06 GMT

View Forum Message <> Reply to Message

Hi,
I have a problem with ArrayCtrl with virtual data.

Background:
I'm trying to make a little utility for translation of texts for an embedded display solution I have. It's a proprietary display, with a lousy SDK, but I have been able to change the drawing routine to accept additional characters. I have then made some new fonts with extra characters for german, danish and polish languages. The standard display supports ASCCI chars from 0x21 to 0x7F, and I simply added the needed characters on top of that.

My translation utility should now take an array of unicode strings and export it C source files with C style strings where the extended characters are exchanged with /x00 values.

Problem:
Since it's a translation utility I need to be able to add and remove column as I please, and neither ArrayCtrl or GridCtrl seems to be able to do that. GridCtrl can remove columns, but it seems I can only add columns before I add row data.

So I made a GridData class holding all my text data, and gave it all the functions I needed. Now using the virtual mode of ArrayCtrl I'm able to display data from my data pool with the following function:

```
Array<EditString> edits;
Array<VirtualField> fields;
GridData map;

....

void CANtrakTexts::PopulateGrid()
{
 text_grid.Reset();
 edits.Clear();
 fields.Clear();
 for (int i=0 ; i < map.GetColumnCount() ; i++)
 {
  text_grid.AddRowNumColumn(map.GetColumn(i),
180).Edit(edits.Add()).SetConvert(fields.Add(new VirtualField(i, &map)));
 }
 text_grid.SetVirtualCount(map.GetRowCount());
}
```

The problem is that when I edit the contents of a row and tries to intercept it in the

WhenAcceptEdit callback, I have no idea where to find my submitted data:

```
void CANtrakTexts::OnAcceptEdit()
{
 String str;
 for (int i=0 ; i < text_grid.GetColumnCount() ; i++)
 {

// The ReadRow function returns nothing...
//  map.SetRow(i, text_grid.ReadRow(text_grid.GetClickRow()));

// The edits all returns 0...
  map.Set(i, text_grid.GetClickRow(), edits[i].GetData());

 }
}
```

Could someone please tell me if it's possible and how?

Thank you in advance,
Steffen

Subject: Re: How to get edit data in virtual ArrayCtrl?
Posted by mirek on Wed, 30 May 2007 21:33:56 GMT

View Forum Message <> Reply to Message

Not that this would be used too often, but...

What you get there is the line index. You are supposed to use Convert ("SetConvert") to connect this line index to the data. Convert is invoked by ArrayCtrl invoked before setting data to widget (Format) and back (Scan), so you should be able to use these methods to connect this with your GridData.

Mirek

Subject: Re: How to get edit data in virtual ArrayCtrl?
Posted by unodgs on Thu, 31 May 2007 09:35:17 GMT

View Forum Message <> Reply to Message

Quote:
> Since it's a translation utility I need to be able to add and remove column as I please, and neither ArrayCtrl or GridCtrl seems to be able to do that. GridCtrl can remove columns, but it seems I can only add columns before I add row data.

In 705-devX GridCtrl can add column after inserting data. Unfortunately it doesn't support virtual rows yet.

Subject: Re: How to get edit data in virtual ArrayCtrl?
Posted by steffen on Thu, 31 May 2007 09:42:01 GMT

View Forum Message <> Reply to Message

Thank you very much.

I had a hard time figuring out all those assertions I got when navigating while editing, but looking through the sources I finally understood how it should work.

I ended up with the following two Convert classes:

```
class VirtualField : public Convert
{
private:
 int col;
 GridData *map;
public:
 VirtualField(int aCol, GridData *pMap)
 {
  col = aCol;
  map = pMap;
 }

 virtual Value Format(const Value& q) const
 {
  return map->Get(col, q); // Convert index to value when showing.
 }
};

class VirtualEdit : public Convert
{
private:
 int col;
 GridData *map;
 ArrayCtrl *array;
public:
 VirtualEdit(int aCol, GridData *pMap, ArrayCtrl *pArray)
 {
  col = aCol;
  map = pMap;
  array = pArray;
 }
 virtual Value Format(const Value& q) const
 {
  return map->Get(col, array->GetCursor()); // Edit always at cursor, so return value to edit.
 }

 virtual Value Scan(const Value& text) const
 {
  map->Set(col, array->GetCursor(), text); // Store value after edit.
  return text;
 }
};
```

And then adding the conversion objects when populating:

```
class CANtrakTexts : public WithCANtrakTextsLayout<TopWindow> {
private:
 Array<EditString> edits;
 Array<Convert> converts;
 GridData map;

....

void CANtrakTexts::PopulateGrid()
{
 text_grid.Reset();
 edits.Clear();
 converts.Clear();
 for (int i=0 ; i < map.GetColumnCount() ; i++)
 {
  edits.Add().SetConvert(converts.Add(new VirtualEdit(i, &map, &text_grid)));
  text_grid.AddRowNumColumn(map.GetColumn(i),
180).Edit(edits[edits.GetCount()-1]).SetConvert(converts.Add(new VirtualField(i, &map)));
 }
 text_grid.SetVirtualCount(map.GetRowCount());
}
```

BTW: Is it ok to use the Array container to hold my edit and convert objects? Is it obsolete or can it be done smarter?

Coming from Delphi I'm thinking in callbacks and I was actually looking for something like WhenShowVirtualData(int col, int row) and WhenAcceptEdit(int col, int row, String val) or something like that. That would have required much less coding in my case...

Steffen