Subject: Spin + Text
Posted by jerson on Mon, 06 May 2013 12:56:38 GMT
View Forum Message <> Reply to Message

I am looking to make a combination of a widget that houses a label with spin buttons. The intention is to use the spin buttons to change the text displayed in the widget from an array of texts

I have investigated various options including the template WithSpin<>. However, these things are beyond my understanding at this stage and I seek some help.

The way I did the task right now is very clumsy, but it works
I have an EditIntSpin on my layout that is shrunk to just show the spin buttons but not the value it holds. Another StaticLabel is lying adjacent to it that shows the String corresponding to the int value contained in the EditIntSpin. The way it works is when the spin is used, it callsback a function that loads the text to the label.

Example

index	Text
0	125
1	250
2	500
3	750

So, if the IntSpinner holds a value of 2, the Label will show 500

```
void A2001_13::ChangeFrequency()
{
 Test.lblFreq.SetText(xLegend[Test.spnFrequency]);
}

// in the topwindow code
 Test.spnFrequency <<= THISBACK(ChangeFrequency);
```

I know this can be done using an ArrayCtrl + WithSpin; but, I am not upto it at the moment. Help please

File Attachments

1) CropImage.jpg, downloaded 530 times

Subject: Re: Spin + Text
Posted by Sender Ghost on Mon, 06 May 2013 23:21:46 GMT
View Forum Message <> Reply to Message

Hello, Jerson.
jerson wrote on Mon, 06 May 2013 14:56I am looking to make a combination of a widget that houses a label with spin buttons. The intention is to use the spin buttons to change the text displayed in the widget from an array of texts
I created following example, based on SpinButtons and EditString classes, which may solve what you looking for:
Toggle Spoiler

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

using namespace Upp;

class EditStringSpin : public EditString {
protected:
 Vector<String> data;
 int index;
public:
 typedef EditStringSpin CLASSNAME;
 EditStringSpin();

 SpinButtons sb;

 void ShowData()      { index >= 0 ? SetText(data[index]) : SetText(String()); }
 void Inc();
 void Dec();
 void SetIndex(int i);
 int GetIndex() const { return index; }
 void GoBegin()       { SetIndex(0); }
 void GoEnd()         { SetIndex(data.GetCount() - 1); }

 virtual bool Key(dword key, int count);
 virtual void MouseWheel(Point p, int zdelta, dword keyflags);

 void SetValue(int i, const String& v);
 void SetValue(const String& v);
 String GetValue(int i) const;
 String GetValue() const;
 const Vector<String>& GetValues() const            { return data; }
 int GetCount() const                               { return data.GetCount(); }
 EditStringSpin& Add(const String& v)               { data.Add(v); return *this; }
 EditStringSpin& Add(const Vector<String>& v)       { data.Append(v); return *this; }
 void Clear()                                       { index = -1; data.Clear(); ShowData(); }
};

EditStringSpin::EditStringSpin() : index(-1)
{
 sb.inc.WhenRepeat = sb.inc.WhenAction = THISBACK(Inc);
 sb.dec.WhenRepeat = sb.dec.WhenAction = THISBACK(Dec);
```

```
 AddFrame(sb);
}

void EditStringSpin::Inc()
{
 const int count = data.GetCount() - 1;
 if (++index > count)
  index = count;

 ShowData();
}

void EditStringSpin::Dec()
{
 if (data.GetCount() && --index < 0)
  index = 0;

 ShowData();
}

void EditStringSpin::SetIndex(int i)
{
 if (index == i)
  return;

 const int count = data.GetCount() - 1;

 if (i < 0)
  index = -1;
 else
 if (i > count)
  index = count;
 else
  index = i;

 ShowData();
}

bool EditStringSpin::Key(dword key, int count)
{
 switch (key) {
 case K_UP:
  Inc(); return true;
 case K_DOWN:
  Dec(); return true;
 default:
  return EditString::Key(key, count);
 }
```

```
}

void EditStringSpin::MouseWheel(Point p, int zdelta, dword keyflags)
{
 zdelta > 0 ? Inc() : Dec();
}

void EditStringSpin::SetValue(int i, const String& v)
{
 ASSERT(i >= 0 && i < data.GetCount());
 data[i] = v;
 if (index == i)
  ShowData();
}

void EditStringSpin::SetValue(const String& v)
{
 if (index < 0)
  return;
 data[index] = v;
 ShowData();
}

String EditStringSpin::GetValue(int i) const
{
 ASSERT(i >= 0 && i < data.GetCount());
 return data[i];
}

String EditStringSpin::GetValue() const
{
 if (index < 0)
  return String::GetVoid();

 return data[index];
}

class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();
 // Ctrls
 EditStringSpin edit;
 Button btnFill, btnClear;
 // Events
 void OnClear();
 void OnFill();
};
```

```
App::App()
{
 Title("EditStringSpin example");
 const Size sz(240, 180);
 SetRect(sz); SetMinSize(sz);

 edit.SetReadOnly();
 OnFill();

 btnFill.SetLabel("Fill") <<= THISBACK(OnFill);
 btnClear.SetLabel("Clear") <<= THISBACK(OnClear);

 Add(edit.LeftPosZ(4, 100).TopPosZ(4, 20));
 Add(btnFill.LeftPosZ(4, 48).TopPosZ(28, 20));
 Add(btnClear.LeftPosZ(56, 48).TopPosZ(28, 20));
}

void App::OnClear()
{
 edit.Clear();
}

void App::OnFill()
{
 OnClear();

 for (int i = 0; i < 10; ++i)
  edit.Add(NFormat("Data #%d", i + 1));

 edit.GoBegin();
}

GUI_APP_MAIN
{
 Ctrl::GlobalBackPaint();

 App app;
 app.Run();
}
```

Where instead of EditString you might use StaticText or Label with NoIgnoreMouse(); and SetFrame(FieldFrame()); into constructor, but they will look a bit differently, than other Edit*Spin widgets.

Edit: Added GetValue methods. Changed some checks to ASSERT.

File Attachments
1) EditStringSpin.png, downloaded 452 times

Subject: Re: Spin + Text
Posted by jerson on Tue, 07 May 2013 00:48:44 GMT
View Forum Message <> Reply to Message

SenderGhost. That is really food for thought to me. Really appreciate your help.

Subject: Re: Spin + Text
Posted by jerson on Tue, 07 May 2013 03:42:56 GMT
View Forum Message <> Reply to Message

Any idea why the application(only the above code) would crash when compiled with MINGW optimal settings? MINGW debug works fine.

Subject: Re: Spin + Text
Posted by Sender Ghost on Tue, 07 May 2013 04:03:24 GMT
View Forum Message <> Reply to Message

jerson wrote on Tue, 07 May 2013 05:42Any idea why the application(only the above code) would crash when compiled with MINGW optimal settings? MINGW debug works fine.
The MinGW GCC v4.8.0 from nuwen.net compiled (and built) above example for Optimal and Debug build modes. They run without errors.

Check the U++ sources, SSE2 package configuration flag (disabled in my tests), the compiler.

Subject: Re: Spin + Text
Posted by jerson on Tue, 07 May 2013 05:01:35 GMT
View Forum Message <> Reply to Message

Thanks SenderGhost. SSE2 was the culprit.