Subject: switch date format
Posted by sapiency on Thu, 07 Jan 2010 10:12:52 GMT
View Forum Message <> Reply to Message

Hi,

I'm just playing around with DropDate ...

setting output with:

```
 SetDateFormat("%2:02d/%3:02d/%1:4d");
 SetDateScan("mdy");
```

works fine, but when I switch to "dmy" and open the calendar, it gets the date from the string and use "d" as "m" and "m" as "d" ...

is there allready a way to solve this?

regards

reinhard

Subject: Re: switch date format
Posted by mrjt on Thu, 07 Jan 2010 16:48:27 GMT
View Forum Message <> Reply to Message

The problem is caused by the EditField, namely this bit of code:

```
void EditField::GotFocus()
{
 if(autoformat && IsEditable() && !IsNull(text) && inactive_convert) {
  Value v = convert->Scan(text);
  if(!v.IsError()) {
   WString s = convert->Format(v);
   if(s != text) text = s;
  }
 }
 if(!keep_selection) {
  anchor = 0;
  cursor = text.GetLength();
 }
 Finish();
 SyncEdge();
}
```

So disabling AutoFormat of the ctrl fixes it.

Subject: Re: switch date format
Posted by sapiency on Thu, 07 Jan 2010 19:14:50 GMT
View Forum Message <> Reply to Message

Hi,

I tried AutoFormat(true) and AutoFormat(false), but in both cases I got the wrong result ...

what I did:
- open gui
- click on arrow (calender opens)
- click on current day
- switch output and scan format (output - string is still the same)
- click on arrow and view calender (wrong date)
- switch output and scan format to old style (output - string is still the same)
- click on arrow and view calender (correct date)

I use the following lines to switch:

```
if( ... )
{
 SetDateScan("mdy");
 SetLanguage( LNG_( 'E', 'N', 'G', 'B' ) );
 SetDateFormat("%2:02d/%3:02d/%1:4d");
}
else
{
 SetDateScan("dmy");
 SetLanguage( LNG_( 'D', 'E', 'D', 'E' ) );
 SetDateFormat("%3:02d/%2:02d/%1:4d");
}
```

I think this problem could be solved if all Controls derived from Date, receive a Signal that the Strings have been changed.

reinhard

Subject: Re: switch date format
Posted by mrjt on Fri, 08 Jan 2010 09:29:40 GMT
View Forum Message <> Reply to Message

Ah, I see. I was confused by your initial post.

There isn't really any mechanism for the sort of singnal you're proposing, but there is another solution:

```
void CtrlLibTest::SetFormat()
{
 static bool toggle = false;
 Backup();
 if(toggle)
 {
  SetDateScan("mdy");
  SetLanguage( LNG_( 'E', 'N', 'G', 'B' ) );
  SetDateFormat("%2:02d/%3:02d/%1:4d");
 }
 else
 {
  SetDateScan("dmy");
  SetLanguage( LNG_( 'D', 'E', 'D', 'E' ) );
  SetDateFormat("%3:02d/%2:02d/%1:4d");
 }
 Restore();
 toggle = !toggle;
}
```

- Backup serializes the values from the EditFields on the window
- the date format is changed
- Serialize again to restore the data

This makes the EditField re-format it's value and you get the correct result.

In practice you probably don't want to use Backup and Restore as this may interfere with other functions (like cancelling), but what they do is very simple and easy to reproduce. It's only using StringStreams.

If you want to apply this effect globally you could iterate through all the windows using GetTopWindows, or send your own signal to tell windows to do it themselves.

If you really want to send a signal straight to the DateTimeCtrl you'd have to make a derived class and send your own signal.