
Subject: first try on a own (simple) widget...

Posted by [Wolfgang](#) on Wed, 30 Nov 2011 15:04:40 GMT

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

Hi there,

I'm trying myself on a own widget, a input field just for time (HH:MM)... found some piece of code in the forum but nothing fits exactly my needs.

Please note thats really my first try for such a thing..

```
class TimeInput : public EditString
{
public:
Value Format(const Value& q) const
{
//TimeInput::setTime(15,7);

String mq = q.ToString();
String rt = "";
int sep = mq.Find(':');
if (sep < 0)
{ /* no : */
int x = StrInt(mq);
if (x < 0 || x > 23) return "00:00";
if (mq.GetLength() == 1)
{
rt << "0" << mq << ":00";
return rt;
}
rt << mq << ":00";
return rt;
}
String x1 = mq.Mid(0,sep);
String x2 = mq.Mid(sep);
while ((sep = x1.Find(':')) != -1) x1.Remove(sep);
while ((sep = x2.Find(':')) != -1) x2.Remove(sep);

int hrs = StrInt(x1);
int min = StrInt(x2);
if (hrs < 0 || hrs > 23) x1 = "00";
else if (x1.GetLength() == 1) x1.Insert(0,'0');

if (min < 0 || min > 59)
{
rt << x1 << ":00";
return rt;
}
if (x2.GetLength() == 1) x2.Insert(0,'0');
```

```

    rt << x1 << ":" << x2;
    return rt;
}
int Filter(int chr) const
{ /* only accept digits and ':' */
    if(IsDigit(chr) || chr == ':') return chr;
    return 0;
}
String increaseMin()
{
    if (++min > 59)
    {
        min = 0;
        if (++hour > 23)
        {
            hour = 0;
            return "00:00";
        }
        if (hour < 10)
        {
            String rt = 0;
            rt << hour << ":00";
            return rt;
        }
        String rt;
        rt << hour << ":00";
        return rt;
    }
    String rt;
    if (hour < 10) rt << "0";
    rt << hour << ":";
    if (min < 10) rt << "0";
    rt << min;
    return rt;
}
private:
    int min;
    int hour;
    void setTime(int m, int h) { min = m; hour = h; }
};
typedef TimelInput EditTimeHM;

```

If I uncomment the `TimelInput::setTime(15,7);` in the `Format` procedure, it failes to compile.

d:\ultimate++\myapps\countdown\Countdown.h(13) : error C2662: 'TimelInput::setTime' : cannot convert 'this' pointer from 'const TimelInput' to 'TimelInput
&'

Conversion loses qualifiers

please be patient, help and tips always welcome!

This piece of code (with commented setTime part) compiles and works as I expect it to do..

best regards

Wolfgang

Subject: Re: first try on a own (simple) widget...

Posted by [cbpporter](#) on Wed, 30 Nov 2011 16:01:48 GMT

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

The format method is "const". This means that it is not supposed to modify the state of the object. In consequence you can't modify fields or call other methods that are not labeled as const.

Normally you would solve this by making the offending method const, but in your case we are talking about a setter method which by definition can't and shouldn't be const.

I suppose you wanted to call setTime for testing purposes? You should do this in the constructor for an easy temporary test. The Format method should not modify the state of the object. Format is a functional method, not a side effect one.

Subject: Re: first try on a own (simple) widget...

Posted by [Wolfgang](#) on Wed, 30 Nov 2011 18:17:29 GMT

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

yes, you're right. Its for testing but the "plan" behind the whole thing is to hold the value (represented as two int values (min and hour)) in the objekt itself to spare conversions. The method increaseMin should increase the int min.... you klick on a button which executes the the increaseMin method and the value of the field increases. But don't know how to do this..

And the modify out of the format method is based on the thought that it is effecient to save the min and hour value just at the moment I already converted it to int (to check it).

You see what I mean?

Any suggestions how I should do it? (or is my attempt itself bad?)

best regards

Wolfgang

Subject: Re: first try on a own (simple) widget...

Posted by [Wolfgang](#) on Thu, 01 Dec 2011 07:08:10 GMT

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

none?

how to add a method to the widget to increase the number of it by 1 for example?
is there a predefined constant or how does it work?

Subject: Re: first try on a own (simple) widget...
Posted by [Wolfgang](#) on Thu, 01 Dec 2011 09:12:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Now I have some working methods...
my next step is to execute this methods by buttons on the widget itself (like the EditIntSpin has this "Spin" Buttons)...

can someone tell me how?

I've attached the whole project..

File Attachments

1) [Countdown.zip](#), downloaded 207 times

Subject: Re: first try on a own (simple) widget...
Posted by [cbpporter](#) on Mon, 05 Dec 2011 09:19:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

I don't understand exactly what you want to do, but you are doing things very complicated.

Everywhere you are setting data, using strings, converting to from strings, etc. This is not really the U++ way. EditIntSpin what you use in some cases has "Int" in its name. So using Value semantics, it defaults to int in the right context. So instead of using:

```
editHour.SetData(0);  
editMin.SetData(0);
```

you could use:

```
editHour <=<= 0;  
editMin <=<= 0;
```

This is not important here though. Where it begins to count is in:

```
void Countdown::btnHourClick()  
{  
    int i = StrToInt(editHour.GetText().ToString());  
    if (i > 0) intime.increaseHour(i);  
}
```

```
else intime.decreaseHour((i*(-1)));

return;
}
```

You should write this as:

```
void Countdown::btnHourClick()
{
    int i = editHour;
    if (i > 0) intime.increaseHour(i);
    else intime.decreaseHour((i*(-1)));

    return;
}
```

You take and EditIntSpin which stores an int in a value, call GetText, call ToString on it and finally you convert it back to an int.

The way you should implement this class IMO is make all the methods that set data to update the hour and minute fields like you wanted and then in the rest of implementation no longer keep obtaining the string representation and parse it.

Subject: Re: first try on a own (simple) widget...

Posted by [cbpporter](#) on Mon, 05 Dec 2011 09:25:33 GMT

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

PS: You probably don't want to derive from EditString. You want to derive from an EditValue with base type probably set to the default Time class from U++. The Time class also support several operations, including conversion to string. The Time class also support seconds, but you can ignore seconds in your control if you don't need them. Using the Time class would make your increaseMin and like methods obsolete/considerably shorter.

Subject: Re: first try on a own (simple) widget...

Posted by [Wolfgang](#) on Mon, 05 Dec 2011 12:08:07 GMT

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

Sounds good but I have no idea how to derive vom EditValue, tried a lot of variants but nothing will work.

Maybe a skeleton of such a class would help... just found the headers for EditField etc. but I can't follow it.

btw. the "EditIntSpin" isn't in my class, just used SpinButtons...

Subject: Re: first try on a own (simple) widget...
Posted by [cbpporter](#) on Mon, 05 Dec 2011 12:27:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

You need to do something like this:

```
class TimeInput : public EditValue<Time, MyConvertTime>
```

where MyConvertTime is a class similar to CovertTime, where you implement your own Scan, Filter and Format Values, similar to how you already did. Look at the implementation of the standard ConvertTime.

And you do use EditIntSpin, not in you custom control class, but you do use it. The examples where I used a better way to interact with these classes were using this class (editHour, editMin).

Subject: Re: first try on a own (simple) widget...
Posted by [Wolfgang](#) on Mon, 05 Dec 2011 14:59:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Okay, thank you very much - I'll try..

Subject: Re: first try on a own (simple) widget...
Posted by [cbpporter](#) on Mon, 05 Dec 2011 15:36:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wolfgang wrote on Mon, 05 December 2011 16:59Okay, thank you very much - I'll try..
Basically the class EditTime from U++ is very similar to what you want, but handles full date and time.

Subject: Re: first try on a own (simple) widget...
Posted by [Wolfgang](#) on Mon, 05 Dec 2011 18:58:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes but on the one hand I want to create such a widget by myself just to see and learn how somewhat is done and on the other hand I just want hour and minute values in it, nothing else.

I've got problems finding the right .cpp and .h files that contains the EditTime... a part of it seems to be in EditCtrl.h but its hard to separate it correctly.

Subject: Re: first try on a own (simple) widget...
Posted by [Wolfgang](#) on Mon, 05 Dec 2011 19:41:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hmmm I think it's too hard for me at this time... with the EditString it worked but to derive from the EditValue there's a lot more work... maybe in a while... *g*
