Subject: EditTime - how to change format?- [SOLVED]-EditField & Convert
Posted by Garry on Sat, 07 Jan 2006 23:37:04 GMT
View Forum Message <> Reply to Message

Well the title pretty much describes my query.

When I enter a time into an EditTime widget and leave the field, the time entered is automatically formatted to include the date.

For example, I enter 8:00 and this automatically becomes "01/08/2006 08:00:00" which is more than enough information for my needs. Is there any way to change how this data is displayed?

Thanks,
Garry

Subject: Re: EditTime - Is it possible to change displayed format?
Posted by mirek on Sun, 08 Jan 2006 09:36:04 GMT
View Forum Message <> Reply to Message

Not using EditTime, but you can quickly cook the widget you need using the generic EditField, creating appropriate Convert and combining them together. 15 lines effort...

There are three methods in Convert class:

```
 virtual Value  Format(const Value& q) const;
```

-> Use Format to convert your Time into String.

```
 virtual Value  Scan(const Value& text) const;
```

-> Use Scan to convert String into your Time

```
 virtual int    Filter(int chr) const;
```

-> Use Filter to limit/convert characters that can be entered to EditField. Each character is Filter-ed - if returned value is zero, it is discarded, otherwise returned value is inserted to the EditField. Means you can perform some conversions too...

Subject: Re: EditTime - Is it possible to change displayed format?
Posted by Garry on Mon, 09 Jan 2006 18:07:21 GMT
View Forum Message <> Reply to Message

Thanks for you're reply, I can see where to go now.

I can see that the Format function is the one that needs to be adjusted for my scenario, however

when I follow the sources I see that the default Format function in Convert is the following:

```
Value  Convert::Format(const Value& q) const {
 if(IsVoid(q) || q.IsNull()) return String();
 switch(q.GetType()) {
 case INT_V:
  return IntStr((int)q);
 case DOUBLE_V:
  return DblStr((double)q);
 case DATE_V:
  return ::Format(Date(q));
 case TIME_V:
  return ::Format(Time(q));
 case STRING_V:
 case WSTRING_V:
  return q;
 }
```

I'm quite new to C++ but trying to catch up quick, but I'm unsure where ::Format(Time(q)) belongs to. I can see that Time() is a declared operator function of Value(and therefore q), but I can't seem to find the definition.

Sorry for the newbie-ness!!

Subject: Re: EditTime - Is it possible to change displayed format?
Posted by mirek on Mon, 09 Jan 2006 19:13:28 GMT
View Forum Message <> Reply to Message

::Format(Time(q))

- q is converted to Time and global Format(const Time&) is called.

However, this "standard" Format is a little but misleading in this case, as what you will really need is Format just for one type (Time). So it might more likely look like this:

```
Value  MyTimeConvert::Format(const Value& q) const {
   Time tm = q;
   return ::Format("%d:%d:%d", tm.hour, tm.minute, tm.second);
}
```

Note that this Convert would crash (or fail the assert in debug mode) if Value non-convertible to Time is passed, but that is OK in this case (you are going to use it just for Time anyway).

Subject: Re: EditTime - Is it possible to change displayed format?
Posted by Garry on Mon, 09 Jan 2006 22:26:20 GMT
View Forum Message <> Reply to Message

Thanks a million - it took me a while to figure out how to do it properly, but I've got it now. Certainly a great learning experience too. Now I realise how to make much better use out of Browser++ too!!

Anyway, here's my code now:

```
class MyTimeConvert: public ConvertTime {

 public:
 Value Format(const Value& q) const {
  Time tm = q;
  return ::Format("%02d:%02d", tm.hour, tm.minute);      //ensures xx:xx format
 }
 int Filter(int chr) const {
 if(IsDigit(chr) || chr == ':')                    //only accepts digits and ":"
  return chr;
 return 0;
 }
};
```

```
typedef EditMinMax<Time, MyTimeConvert> EditTimeHM;
```

I only need Format to display hours and minutes with two digits for each unit.

This abridged version of Filter only allows digits and the colon symbol.

The standard version of Scan seems to work well at interpreting the time in this manner, but I might alter it in my own program so that invalid times are converted to proper times (eg 25:01 becomes 01:01).

By the way, now that I understand the actual processes behind this, I think there's a bug in ConvertTime. The global Format produces a String of Time with forward slashes "/" dividing days, months and years but ConvertTime::Scan and ConvertTime::Filter expect full stops instead ".". This causes EditTime to refuse to close once any time has been entered.

All the best, and thanks again!
Garry

*edit fixed a typo

Subject: Re: EditTime - Is it possible to change displayed format?
Posted by lectus on Thu, 05 Jul 2012 20:39:54 GMT
View Forum Message <> Reply to Message

I'm having trouble with Time validation too.

To me it's much more useful to have EditTime in HH:MM format.

EDIT: Found the solution below.

Subject: Re: EditTime - Is it possible to change displayed format?
Posted by lectus on Wed, 11 Jul 2012 16:57:21 GMT
View Forum Message <> Reply to Message

Garry wrote on Mon, 09 January 2006 17:26
Anyway, here's my code now:

Thanks to your code I managed to understand how to do validation for my own code.
I wanted the user to be able to only input time in the format hh:mm.

Here's the solution:
1) I put one EditString called "e" and one Button called "b" in the layout.

2) Add this code:

```
// Custom character filter
class MyTimeConvert: public Convert {

 public:
 int Filter(int chr) const {
  if(IsDigit(chr) || chr == ':')                    //only accepts digits and ":"
   return chr;
  return 0;
 }
};

// Validation callback. Uses regular expressions:
void MainWin::ValidateTime()
{
 RegExp r0("\\d\\d:\\d\\d");
 if (!r0.Match(~e))
  Exclamation("Invalid time!");
}

e.MaxChars(5); // limit it to 5 chars
e.SetConvert(Single<MyTimeConvert>()); // sets custom filtering
b.WhenPush = THISBACK(ValidateTime); // when clicked check for format hh:mm.
```
