
Subject: `EditField::operator String()` removed
Posted by [mirek](#) on Mon, 13 Jun 2016 07:11:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is a very unfortunate change, because it breaks existing code, OTOH is very easy to fix...

The problem is:

```
EditField f;
```

```
f << [&] { ... };
```

This supposed to assign lambda to `EditField`. However, with implicit cast to `String`, an alternative overload is

```
template <class T>  
operator<<(String&&, const T&)
```

because of `EditField::operator String()`.

Well, direct type assignment and retrieval from/to widgets was deprecated for some time, so in the end I decided that it is more important for us to be able to write e.g.

```
String() << foo << bar;
```

and always use operator `~` when retrieving value of widget.

So, if after updating to U++ trunk you see error like this:

```
c:\u\laws\openwind\OpenWind.h(1722): error C2664: 'Upp::FileSel &Upp::FileSel::ActiveDir(const Upp::String &)': cannot convert argument 1 from 'Upp::EditString' to 'const Upp::String &'  
c:\u\laws\openwind\OpenWind.h(1722): note: Reason: cannot convert from 'Upp::EditString' to 'const Upp::String'  
c:\u\laws\openwind\OpenWind.h(1722): note: No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
```

please just add `'~'` before the widget....

Mirek
