
Subject: Re: this one worked for me

Posted by [mrjt](#) on Thu, 20 Aug 2009 14:26:00 GMT

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

Well, the problem is that you are trying to use types that cannot have a Null defined for them (this is usually the case for smaller variants of a type, float and short for instance). These cannot be directly converted to and from Values.

I actually should have removed all of these types from my version as it won't work compile (as you discovered). Hacking in arbitrary Null values isn't a very good solution IMO.

What I hadn't thought of before was that the task would be much more easily accomplished with something like:

```
struct ConvertUnsignedChar : public ConvertInt
{
    ConvertUnsignedChar() { MinMax(0, 255); }
};
```

```
template <class CONVERT, const int CHARS>
struct EditCustom : public EditMinMax<int, CONVERT >
{
    EditCustom() { MaxChars(CHARS); }
};
```

```
typedef EditCustom<ConvertUnsignedChar, 3> EditUnsignedChar;(This would be even simpler if
you didn't need the max chars)
```

Since you're doing all the RTTI typeid stuff internally anyway, why not just do it in a function somewhere to create the correct type of EditField/Convert and avoid all the horrific bodging?
