
Subject: Re: NEW: generic Toupel grouper
Posted by [dolik.rce](#) on Sun, 15 Aug 2010 11:09:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Kohait

kohait00 wrote on Sun, 15 August 2010 09:16 dont worry, it's part of development, and didnt feel criticised . i am always looking forward to meeting better ideas..
I don't worry, it was a joke

kohait00 wrote on Sun, 15 August 2010 09:16
the `Get<1>()` option (idea from boost?) is a cool trick, but IMHO of little use because it's not a runtime check, but a compile time definition, thus `u.a`, `u.b`, `u.c` is much simpler and clearer in that sense, and less to type anyway. i mean, in terms of compile time specialisation `u.Get<1>` is same as `u.a`, you have to provide the index at compiletime, so you know which type.
Yes, it is from boost. As I said, it is mostly useless and the only way it might be helpful is making the code look better and hopefully better readable. But I don't insist on having it at all. At least it learned me some interesting new things about templates

kohait00 wrote on Sun, 15 August 2010 09:16
the `Value operator[](int i)` is a good idea though. to wrap / unwrap in value (boxing / unboxing is used in C# and others, though there in different context, as base class object).
Don't forget about the `GetCount()` too I just don't like my implementation of it very much, but I can't come up with anything better. And if possible I would also like to see `Begin()` and `End()` implemented, so I could do `DUMPC(touple)`...

kohait00 wrote on Sun, 15 August 2010 09:16
having Duo, Trio, etc is, as you pointed out, more or less useless, even if it's better to read so i added a 5th T and deaulted past second T. (i'd rater use `EmptyClass`, but there is no `Value(const EmptyClass &)` for it, so i used Nuller. might be usefull to have an `EmptyClass Value` as well?)
I would strongly prefer `EmptyClass` too. But there might be idealogical problem: Once you make it value compatible, it won't be empty any more Maybe we should do a special class for this purpose, let's say `DummyElement`, which would be `Value` and `Null` compatible.

Apart from what I said above, especially the missing `GetCount()`, your last code seems reasonable. Definitely not that difficult to read as mine (which is good)

Honza
