Subject: Re: NEW: generic Toupel grouper
Posted by kohait00 on Tue, 31 Aug 2010 12:04:26 GMT
View Forum Message <> Reply to Message

void* is simple but makes things possible

```
template<class A, class B, class C=void*, class D=void*>
class Tupel {
public:
 Tupel(const A& a, const B& b, const C& c = NULL, const D& d = NULL)
  : a(a), b(b), c(c), d(d) {}
 Tupel() {}
 A a; B b; C c; D d;
};
```

EDIT: union is problematic, if T=One<int> i.e.

```
template<class A, class B, class C=void*, class D=void*>
class Tupel {
public:
 Tupel(const A& a, const B& b, const C& c = NULL, const D& d = NULL)
  : a(a), b(b), c(c), d(d) {}
 Tupel() {}
 union { A a, key, v1; };
 union { B b, value, v2; };
 union { C c, v3; };
 union { D d, v4; };
};
```

Quote:

```
error: member 'Upp::One<int> Tupel<int, Upp::One<int>, Upp::String, int>::<anonymous
union>::b' with constructor not allowe
 d in union
```

TDMGCC