

---

Subject: Re: PolyDeepCopyNew: MSC / GCC differ in behaviour

Posted by [kohait00](#) on Mon, 18 Oct 2010 07:35:46 GMT

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

---

hi mirek

it's connected because i used PolyDeepCopyNew to specify the DeepCopyNew function to use the t.Copy() function, which is obligatorily defined in Copyable<> (to make sure it is implemented) and accessible from this interfaces).

the weird point is, in MSC it works.. gcc does not recognize the friend DeepCopyNew as the specialisation from template DeepCopyNew, or the containers (here Array) don't get to understand that a global DeepCopyNew is available (from PolyDeepCopyNew, as friend), without the need to use the template DeepCopyNew..

with which mingw version did it work for you?

EDIT:

maybe it's the same problem that led you or the guys to implement the Moveable<> in the way it is done now.

```
template <class T>
```

```
inline void AssertMoveable(T *t = 0) { if(t) AssertMoveable0(t); }
```

```
#if defined(COMPILER_MSC) || defined(COMPILER_GCC) && (__GNUC__ < 4 ||  
__GNUC_MINOR__ < 1)
```

```
#define NTL_MOVEABLE(T) inline void AssertMoveable0(T *) {}
```

```
#else
```

```
#define NTL_MOVEABLE(T) template<> inline void AssertMoveable<T>(T *) {}
```

```
#endif
```

maybe because there are problems with recognition of the template / friend function same names

...