Subject: Re: Interesting struggle with "Moveable<T>" usage in GCC
Posted by mrjt on Tue, 22 Jul 2008 10:40:51 GMT
View Forum Message <> Reply to Message

The explanation for this is quite straightforward I think.

Structs with just public data members are essentially C aggregates, which can be initialised using the {,} style list.

As soon as you add anything that stops it being a 'pure' aggregate - such as a constructor, inheritance (even from an empty class as is the case with Moveable<T>) or virtual functions - the compiler starts treating the struct as a C++ class. As classes cannot be initialised like aggregates you get the error.

For example:

```
// This works with {,} initialisation
struct ST {
  Point Get() { return Point(x, y); }
  int x, y;
};

// These don't
struct BC {
};
struct ST : BC {
  int x, y;
};

struct ST {
  ST() : x(0), y(0) { }
  int x, y;
};

struct ST {
  virtual Point Get() { return Point(x, y); }
  int x, y;
};
```As far as I can tell this is correct compiler behaviour, although it would be nice if it was a bit more forgiving.

I'm not sure about your problem with NTL_MOVEABLE though, I just tried it with MingW and MSC8 and had no problem with missing DeepCopyConstructs, but presumably you're using a more complex structure than my test.