
Subject: Re: Bug in Core06.cpp or in U++ Core value types tutorial

Posted by [Werner](#) on Wed, 05 Dec 2007 17:29:21 GMT

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

luzr wrote on Wed, 05 December 2007 14:59

Thank you. Bug reproduced and fixed (templates are implemented according to standard in GCC... while MSC interpreted certain code as I wanted.... .

To be sure, please replace these in Value.h:

```
template <class T>
inline dword ValueTypeNo(const T&)    { return StaticTypeNo<T>() + 0x80000000;; }

template<> inline dword ValueTypeNo(const int&)    { return INT_V; }
template<> inline dword ValueTypeNo(const int64&) { return INT64_V; }
template<> inline dword ValueTypeNo(const double&) { return DOUBLE_V; }
template<> inline dword ValueTypeNo(const bool&)   { return BOOL_V; }
template<> inline dword ValueTypeNo(const String&) { return STRING_V; }
template<> inline dword ValueTypeNo(const WString&) { return WSTRING_V; }
template<> inline dword ValueTypeNo(const Date&)   { return DATE_V; }
template<> inline dword ValueTypeNo(const Time&)   { return TIME_V; }

template <class T, dword type, class B = EmptyClass>
class AssignValueTypeNo : public B {
public:
    friend dword ValueTypeNo(const T&)          { return type; }

    void operator=(const AssignValueTypeNo&) {} // MSC 6.0 empty base class bug fix
};

template <class T>
bool IsType(const Value& x, T* = 0)          { return ValueTypeNo(*(T *)NULL) == x.GetType(); }

template <class T>
inline bool Value::Is() const
{
    return IsType<T>(*this);
}
```

..and test..

Mirek

Thanks. Now works as expected.

Just to do a quick test, I built TheIDE using gcc 4.2.1 with "MinGW Debug" and "MinGW Optimal".

No problems, too.

Werner
