

Anyway, this is a new autotest:

```
template <class T>
void Test(T c)
{
    T h = c;

    {
        Value v = h;
        ASSERT(~v == AsString(c));
        ASSERT(v.Is<T>());
        ASSERT(v.To<T>() == h);
        T h1 = v;
        ASSERT(h == h1);
    }
    {
        Value v = RawToValue(h);
        ASSERT(v.Is<T>());
        ASSERT(v.To<T>() == h);
        T h1 = v.To<T>();
        ASSERT(h == h1);
    }
    { // this is not supported by Value definition (RichToValue should only be used in client type)
        Value v = RichToValue(h);
        ASSERT(~v == AsString(c));
        ASSERT(v.Is<T>());
        T h1 = v.To<T>();
        ASSERT(h == h1);
    }
}

CONSOLE_APP_MAIN
{
    StdLogSetup(LOG_COUT|LOG_FILE);

    {
        struct Foo { int x; } x;
        x.x = 123;
        Value v = RawToValue(x);
        ASSERT(v.Is<Foo>());
        ASSERT(v.To<Foo>().x == 123);
    }
}
```

```
Test<int>(1234);  
Test<String>("1234");  
Test<WString>("1234");  
  
LOG("=== Everything OK");  
}
```
