Subject: PROPOSAL: SerializeStore helper
Posted by kohait00 on Tue, 19 Oct 2010 10:05:34 GMT
View Forum Message <> Reply to Message

just a short add..

XmlizeStore exists, to be able to handle 'const T& x'
I couldnt find anything similar for Serialize..
here is a solution.

```
template<class T>
void SerializeStore(Stream& s, const T& x)
{
 ASSERT(s.IsStoring());
 s % const_cast<T&>(x);
}
```

this makes thing possible like

SerializeStore(s, mymap.GetKey(i));

without the hassle of a const_cast each time..and an ASSERT is there..

any better solution or didnt i think of every pitfall..

Subject: Re: PROPOSAL: SerializeStore helper
Posted by mirek on Tue, 19 Oct 2010 13:28:49 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 19 October 2010 06:05just a short add..

XmlizeStore exists, to be able to handle 'const T& x'
I couldnt find anything similar for Serialize..
here is a solution.

```
template<class T>
void SerializeStore(Stream& s, const T& x)
{
 ASSERT(s.IsStoring());
 s % const_cast<T&>(x);
}
```

this makes thing possible like

SerializeStore(s, mymap.GetKey(i));

without the hassle of a const_cast each time..and an ASSERT is there..

any better solution or didnt i think of every pitfall..

Well, the equivalent function is there, called "Store", but it now accepts only non-const parameter - so the change would be there...

The problem I possibly see with it (and in fact, maybe the XmlizeStore is wrong) is that you are giving option to client code to change const object... I mean, nothing prevents Serialize to mutate the object.

But perhaps I am wrong... Another opinion?

Subject: Re: PROPOSAL: SerializeStore helper
Posted by kohait00 on Tue, 19 Oct 2010 13:47:39 GMT

View Forum Message <> Reply to Message

i think at a cetain point one needs to handle over control to tested and working code. there, trust begins
otherwise, i wouldn't see the point of XmlizeStore as well.

also, to leave the user using const_cast at points he maybe doesn't really understand, 'just to make this damn thing compile', is probably more error prone.. it's to hide all that 'hacking' away from user.

just an opinion. you are to decide..