
Subject: Suggest Xmlize support for Value

Posted by [Mindtraveller](#) on Mon, 12 May 2008 22:34:44 GMT

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

In the process of mastering U++ XML abilities I've met a problem with storing Value variables in XML, also with storing Value vectors and maps. I suggest following code to solve this problem:

```
template<> void Upp::Xmlize(XmlIO xml, Value& v)
{
    if (xml.IsLoading())
    {
        String s;
        xml.Attr("value", s);
        StringStream ss(s);
        ss.SetLoading();
        ss % v;
    }
    else
    {
        StringStream ss;
        ss.SetStoring();
        ss % v;
        xml.Attr("value", (String) ss);
    }
}
```

P.S. Maybe it would also be useful to add raw data <-> string uuencoding if it is vital to comply XML standard at 100% (some XML readers doesn't read "non standard" characters like).

Subject: Re: Suggest Xmlize support for Value

Posted by [Mindtraveller](#) on Tue, 13 May 2008 01:19:01 GMT

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

The problem above appeared to be more serious: some serialized strings with trailing zero bytes are automatically truncated while added to the xml. So I suggest my solution by simple uuencoding bytes:

```
String UUEncode(const String &s)
{
    String out;
    out.Reserve(s.GetCount()*2);
    for (int i=0;i<s.GetCount();++i)
        out+=Format("%02X",0xFF & s[i]);

    return out;
}
```

```
String UUDecode(const String &s)
{
    String out;
    out.Reserve(s.GetCount()/2);
    for (int i=0;i<s.GetCount()/2;++i)
        out += (char) (0xFF & ScanInt(s.Mid(i*2,2).Begin(), NULL, 16));

    return out;
}
```

```
//-----
NAMESPACE_UPP
template<> void Xmlize(XmlIO xml, Value& v)
{
    if (xml.IsLoading())
    {
        String s;
        xml.Attr("value", s);
        StringStream ss(UUDecode(s));
        ss.SetLoading();
        ss % v;
    }
    else
    {
        StringStream ss;
        ss.SetStoring();
        ss % v;
        String s(UUEncode((String) ss));
        xml.Attr("value",s);
    }
}
END_UPP_NAMESPACE
```

Subject: Re: Suggest Xmlize support for Value
Posted by [Mindtraveller](#) on Wed, 14 May 2008 15:30:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Isn't it usable? Any thoughts?

Subject: Re: Suggest Xmlize support for Value
Posted by [unodgs](#) on Wed, 14 May 2008 15:52:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, it should be there. I lately needed it too.

Subject: Re: Suggest Xmlize support for Value
Posted by [mirek](#) on Wed, 28 May 2008 18:32:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Wed, 14 May 2008 11:30 Isn't it usable? Any thoughts?

The only problem I see is mostly philosophical, but quite serious one IMO.

Xmlize is intended to provide serialization of data in "common form". If you use binary serialization for Value, you bind the output XML file with U++ internal binary format. In that case, why use XML? Why not to use binary serialization directly?

Mirek

Subject: Re: Suggest Xmlize support for Value
Posted by [Mindtraveller](#) on Wed, 28 May 2008 19:51:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Because it handles with hierarchical config data very conveniently. You should not write your own binary containers. Moreover, data could be partially binary and partially textual. I heavily use this feature i.e. for saving library of objects with dynamic properties. This means I don't know types of each property at compile time, as user may add properties of any common type. This looks like rather common task and Xmlize works great for this (I even stopped writing my own XmlProperties class when implemented Value support for Xmlize).

Binding to binary format looks like a serious problem. But do we really do it? What we do is writing the same data output we have for all the types but in "textual" format. We could have the same output on all the platforms. The same bytes. My method simply emulates reading binary bytes. The same bytes you would read by using "direct" method. This is as far as I understand, maybe it's not that simple though.

Subject: Re: Suggest Xmlize support for Value
Posted by [mirek](#) on Fri, 30 May 2008 16:12:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, actually, binary Serialize is as good or even better handling hierarchies and even can easily handle versions (via simple int version = trick).

Anyway, maybe we should, instead doing this "implicit" and only for the Value, find some explicit way how to store binary serialized data into XML, I mean connect both systems....

Mirek

Subject: Re: Suggest Xmlize support for Value
Posted by [Mindtraveller](#) on Fri, 30 May 2008 21:42:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

The main problem is that you cannot automatically add serialized bytes into the XML file (due to the breaking XML file consistency) and this is very inconvenient. We may alternatively make XML serialization wrapper classes or member functions which will "textualize" binary output data from Serialize(). And this method should be as easy to use as Xmlize support for Value. Or users will write their own Xmlize support instead...

Subject: Re: Suggest Xmlize support for Value
Posted by [mirek](#) on Sat, 31 May 2008 16:04:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Fri, 30 May 2008 17:42
We may alternatively make XML serialization wrapper classes or member functions which will "textualize" binary output data from Serialize().

Of course - this is exactly what I propose

The net result will in fact be similar to your original idea, except that you will use different function...

Mirek
