
Subject: Value and Serialize

Posted by [mubeta](#) on Tue, 03 Apr 2007 15:33:55 GMT

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

Hi,

From long time ago, I am using the following VectorMap declaration for managing some data:

```
//  
VectorMap<int, Vector<Value> > MyData;  
//
```

And with the following two functions, I store a lot of this data in some database, using "Serialization":

```
//  
VectorMap<int, Vector<Value> > StringToVector(const String &s) {  
  
    StringStream ss(s);  
    VectorMap<int, Vector<Value> > m;  
    ss.SetLoading();  
    m.Serialize(ss);  
    //LoadFromString(m, s);  
    return m;  
}
```

```
String VectorToString(VectorMap<int, Vector<Value> > &v) {
```

```
    StringStream ss;  
    //ss.Create();  
    ss.SetStoring();  
    v.Serialize(ss);  
    return ss.GetResult();  
    //return StoreAsString(v);  
}  
//
```

I start to do this and compiling the application with UPP605. I give the application to some peoples, and now they have a lot of user file with data stored inside.

Now I am developing this application with UPP2007, but there is a problem: when I want to read the data stored in the file created in the past, the serialization don't recreate an correct VectorMap; the key values of the VectorMap it's correct, but not the values.

I have scanned manually the string generated with the old application, and I have compared with the string generated with new application. There is one difference: before every value of type "Value" it's preceded by the hex number 08, now in the new string of serialization there is the hex number 09.

With an hex editor I have changed some files, just modifying this number, and after this I am able

to read well also the old files.

So, after this test, I think: maybe, in the past, the type "Value" is recognized from the compiler with the number 0x08, now, in the last upp the "Value" type is recognized as number 0x09.

That mean that using a non "native" data type it's dangerous for future compatibility?

And how I can force the compiler to give the same identifier for Upp data type, for each release of Upp?

Thanks
Mauro Bottizzo.

Subject: Re: Value and Serialize
Posted by [mirek](#) on Tue, 03 Apr 2007 15:52:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

What type do you store into the Value?

Mirek

Subject: Re: Value and Serialize
Posted by [mubeta](#) on Tue, 03 Apr 2007 16:27:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Integer or double for this application.

Mauro.

Subject: Re: Value and Serialize
Posted by [mubeta](#) on Tue, 03 Apr 2007 16:33:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, for more detail: the data stored in the "Value" is get from an ArrayCtrl, so, in the serialization, the number is stored as a wstring.

Mauro.

Subject: Re: Value and Serialize
Posted by [mirek](#) on Wed, 04 Apr 2007 07:57:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

mubeta wrote on Tue, 03 April 2007 12:33Well, for more detail: the data stored in the "Value" is get from an ArrayCtrl, so, in the serialization, the number is stored as a wstring.

Mauro.

Why is it WString? Are you using StringEdit in ArrayCtrl?

Mirek

Subject: Re: Value and Serialize
Posted by [mubeta](#) on Thu, 05 Apr 2007 14:12:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes, in fact I use the Vector<Value> for generic array of many data. Can be integer, double, string, etc.

Note that I found an "fast2" solution for my problem:
in Core/Value.cpp file, Value::Serialize(), after

```
if(s.IsLoading()) {  
    s / type;
```

I have added a new line:

```
if(type == 7) type = 8;
```

I fix my problem with this, but I don't know very well why in the past the serialization of the "Value" data, is stored as type "VALUE_V" and now, after compiling with irc2, the same type is stored as type "WSTRING_V".

I give a fast look around the Pack() functions in Core/Stream files, but without understand as well how the string of serialization is created, so I don't understand if the different number in the serialization result come from a bug in the "Value" or "Stream" code. Or, otherwise, in some my problems.

Mauro Bottizzo.

Subject: Re: Value and Serialize
Posted by [mirek](#) on Fri, 06 Apr 2007 13:32:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, I have just checked - we have not done the mistake of changing those values from 605 up to now.

upp605

```
const int VOID_V = 0;

const int INT_V = 1;
const int DOUBLE_V = 2;
const int STRING_V = 3;
const int DATE_V = 4;
const int TIME_V = 5;

const int ERROR_V = 6;

const int VALUE_V = 7;

const int WSTRING_V = 8;

const int INT64_V = 10;

const int UNKNOWN_V = (int)0xffffffff;
```

current:

```
const int VOID_V = 0;

const int INT_V = 1;
const int DOUBLE_V = 2;
const int STRING_V = 3;
const int DATE_V = 4;
const int TIME_V = 5;

const int ERROR_V = 6;

const int VALUE_V = 7;

const int WSTRING_V = 8;

const int INT64_V = 10;
const int BOOL_V = 11;

const int UNKNOWN_V = (int)0xffffffff;
```

There must be something else involved...

Mirek
