Subject: Xmlize values
Posted by mdelfede on Tue, 23 Feb 2010 23:12:28 GMT
View Forum Message <> Reply to Message

Here the sample code for few types :

```
template<> void Xmlize(XmlIO xml, Value& var)
{
 dword type;

 if(xml.IsLoading())
 {
  xml.Attr("type", type);
  switch(type)
  {
   case INT_V:
   {
    int i;
    xml.Attr("value", i);
    var = i;
    break;
   }
   case DOUBLE_V:
   {
    double d;
    xml.Attr("value", d);
    var = d;
    break;
   }
   case STRING_V:
   {
    String s;
    xml.Attr("value", s);
    var = s;
    break;
   }
   case BOOL_V:
   {
    bool b;
    xml.Attr("value", b);
    var = b;
    break;
   }
   case WSTRING_V:
   case DATE_V:
   case TIME_V:
   case VALUE_V:
```

```
  case VALUEARRAY_V:
  case INT64_V:
  case VOID_V:
  case ERROR_V:
  case VALUEMAP_V:
  case UNKNOWN_V:
  default:
   NEVER();
   break;
 }
}
else
{
 type = var.GetType();
 xml.Attr("type", type);
 switch(type)
 {
  case INT_V:
  {
   int i = var;
   xml.Attr("value", i);
   break;
  }
  case DOUBLE_V:
  {
   double d = var;
   xml.Attr("value", d);
   break;
  }
  case STRING_V:
  {
   String s = var;
   xml.Attr("value", s);
   break;
  }
  case BOOL_V:
  {
   bool b = var;
   xml.Attr("value", b);
   break;
  }
  case WSTRING_V:
  case DATE_V:
  case TIME_V:
  case VALUE_V:
  case VALUEARRAY_V:
  case INT64_V:
  case VOID_V:
```

```
   case ERROR_V:
   case VALUEMAP_V:
   case UNKNOWN_V:
   default:
    NEVER();
    break;
  }
 }
}
```

Ciao

Max

Subject: Re: Xmlize values
Posted by mdelfede on Tue, 23 Feb 2010 23:20:22 GMT
View Forum Message <> Reply to Message

For Koldo, a sample ArrayCtrl xmlized with some processing added on load/store.... just a small part of my app.

```
void GenericLoads::Xmlize(XmlIO xml)
{
 Vector<Vector<Value> >vc;
 Vector<Vector<Value> >vl;
 xml("DAN", DANName);
 if(xml.IsStoring())
 {
  for(int iCond = 0; iCond < conds.GetCount(); iCond++)
   vc.Add(conds.ReadRow(iCond));
  xml("Conds", vc);
  for(int iLoad = 0; iLoad < loads.GetCount(); iLoad++)
   vl.Add(ReadLoadLine(iLoad));
  xml("Loads", vl);
 }
 else
 {
  // setup measurement units reading them
  // from global settings
  distribUM = globalSettings().GetUnitaMisura().Distribuiti;
  concentrUM = globalSettings().GetUnitaMisura().Concentrati;
  lengthUM = globalSettings().GetUnitaMisura().Lunghezze;

  xml("Conds", vc);
```

```
  xml("Loads", vl);
  conds.SetCount(vc.GetCount());
  loads.SetCount(vl.GetCount());
  PropagateDAN(DANName);
  for(int iCond = 0; iCond < vc.GetCount(); iCond++)
   conds.Set(iCond, vc[iCond]);
  SyncConds();
  for(int iLoad = 0; iLoad < vl.GetCount(); iLoad++)
   WriteLoadLine(iLoad, vl[iLoad]);

  // synchronize labels
  SyncLoads();
 }
}
```

Subject: Re: Xmlize values
Posted by koldo on Wed, 24 Feb 2010 11:14:09 GMT
View Forum Message <> Reply to Message

Great Massimo

It is true that Value Xmlize implementation can be improved, but it is outside my knowledge .

Based in your code I have prepared Xmlize for GridCtrl so now it is possible to serialize a class with a GridCtrl inside with just this:

```
void MyClass::Xmlize(XmlIO xml) {
 xml
  ("grid", myGrid)
 ;
}
```

It is very simple

Unodgs: Could you include this en GridCtrl ?. From inside the class it would be much more efficient.

Subject: Re: Xmlize values
Posted by mirek on Fri, 26 Feb 2010 10:57:36 GMT
View Forum Message <> Reply to Message

I was thinking about Xmlize Value problem and I think this should be quite a fine solution:

First, Xmlize itself should know all Values that are in the Core (I guess we are there already,

right?).

Then, add registering mechanism for unknown Values. Anyway, I think it would be better not to register any values automatically, because that would link all XML code into any application. So

Last but not least, for non-registered Values, use binary serialization and put a hex string there. Important - this should be signalled in the serialization so that even if type is registered later, old XML files can still be loaded.

What do you think?

Mirek

Subject: Re: Xmlize values
Posted by koldo on Fri, 26 Feb 2010 11:27:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 26 February 2010 11:57I was thinking about Xmlize Value problem and I think this should be quite a fine solution:

First, Xmlize itself should know all Values that are in the Core (I guess we are there already, right?).

Then, add registering mechanism for unknown Values. Anyway, I think it would be better not to register any values automatically, because that would link all XML code into any application. So

Last but not least, for non-registered Values, use binary serialization and put a hex string there. Important - this should be signalled in the serialization so that even if type is registered later, old XML files can still be loaded.

What do you think?

Mirek
Hello Mirek

This is out of my knowledge. I am not strong in templates but, it would be great if Value Xmlize would use automatically the Xmlize method for every class.

I mean, this is a detail of actual implementation:

```
template<> void Xmlize(XmlIO xml, Value& var) {
 dword t;

 if(xml.IsLoading()) {
  xml.Attr("type", t);
  switch(t) {
  case INT_V:
```

```
  int i;
  xml.Attr("value", i);
  var = i;
  break;
 case DOUBLE_V:
  double d;
  xml.Attr("value", d);
  var = d;
  break;
   ... the same for all Value types
```

However it would be great to have something like this (pseudocode):

```
template<> void Xmlize(XmlIO xml, Value& var) {
 dword t;

 if(xml.IsLoading()) {
  xml.Attr("type", t);
  xml.Attr("value", var->v as type t);  // force var to be as its type
 } else {
  ...
```

Subject: Re: Xmlize values
Posted by mdelfede on Fri, 26 Feb 2010 11:37:51 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 26 February 2010 11:57I was thinking about Xmlize Value problem and I think this should be quite a fine solution:

First, Xmlize itself should know all Values that are in the Core (I guess we are there already, right?).

Then, add registering mechanism for unknown Values. Anyway, I think it would be better not to register any values automatically, because that would link all XML code into any application. So

Last but not least, for non-registered Values, use binary serialization and put a hex string there. Important - this should be signalled in the serialization so that even if type is registered later, old XML files can still be loaded.

What do you think?

Mirek

That would be the best, besides (maybe) some problems that can arise in binary serialization of unknown types, if types are non-POD ones (don't know if Value can store such types).

About the non-auto registering of all value types.... fine, but if you forget to register it you'll have it

binary serialized, not very nice.
Maybe a conditional code part that does the registering only if XML code is included would be fine.

Ciao

Max

Subject: Re: Xmlize values
Posted by mirek on Fri, 26 Feb 2010 12:12:14 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Fri, 26 February 2010 06:37luzr wrote on Fri, 26 February 2010 11:57I was thinking about Xmlize Value problem and I think this should be quite a fine solution:

First, Xmlize itself should know all Values that are in the Core (I guess we are there already, right?).

Then, add registering mechanism for unknown Values. Anyway, I think it would be better not to register any values automatically, because that would link all XML code into any application. So

Last but not least, for non-registered Values, use binary serialization and put a hex string there. Important - this should be signalled in the serialization so that even if type is registered later, old XML files can still be loaded.

What do you think?

Mirek

That would be the best, besides (maybe) some problems that can arise in binary serialization of unknown types, if types are non-POD ones (don't know if Value can store such types).

Oh, I have meant only types that have binary serialization support in Value, of course...

Quote:
About the non-auto registering of all value types.... fine, but if you forget to register it you'll have it binary serialized, not very nice.

True, but XML being backward compatible gives you oportunity to fix that later...

Quote:
Maybe a conditional code part that does the registering only if XML code is included would be fine.

Maybe. Perhaps needs more thinking...
Ciao

Max
[/quote]

Subject: Re: Xmlize values
Posted by mirek on Sun, 28 Feb 2010 14:08:37 GMT
View Forum Message <> Reply to Message

OK, all is implemented now, including a new reference/XmlizeCustomValue example.

As for the problem of registering non-Core Values, I have found only one Value compatible type where this is a sort of problem: Font. I think I can live with that for now...

Mirek

Subject: Re: Xmlize values
Posted by koldo on Sun, 28 Feb 2010 14:44:05 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 28 February 2010 15:08OK, all is implemented now, including a new reference/XmlizeCustomValue example.

As for the problem of registering non-Core Values, I have found only one Value compatible type where this is a sort of problem: Font. I think I can live with that for now...

Mirek
Hello Mirek

Thank you for including additional basic types.

Value .xml implementation has been changed so I have lost some data but I will recover it .

Are you going to add new Xmlize functions for CtrlLib classes ?

Subject: Re: Xmlize values
Posted by mirek on Sun, 28 Feb 2010 16:03:59 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 28 February 2010 09:44luzr wrote on Sun, 28 February 2010 15:08OK, all is

implemented now, including a new reference/XmlizeCustomValue example.

As for the problem of registering non-Core Values, I have found only one Value compatible type where this is a sort of problem: Font. I think I can live with that for now...

Mirek
Hello Mirek

Thank you for including additional basic types.

Value .xml implementation has been changed so I have lost some data but I will recover it .

Actually, strange - I was coming from your sources. What has changed?

Quote:
Are you going to add new Xmlize functions for CtrlLib classes ?

There is one generic Ctrl::Xmlize.

Mirek

Subject: Re: Xmlize values
Posted by koldo on Sun, 28 Feb 2010 16:32:44 GMT
View Forum Message <> Reply to Message

Hello Mirek

Quote:Actually, strange - I was coming from your sources. What has changed?
Before there was a separate "type" and a "value" for all types.

Now it is not exactly the same, but no problem :

<item type="String">This is a text</item>
<item type="Time" value="20020101T00:00:00"/>

Quote:There is one generic Ctrl::Xmlize

Where is it ?

Subject: Re: Xmlize values
Posted by mdelfede on Sun, 28 Feb 2010 19:41:14 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 28 February 2010 15:08OK, all is implemented now, including a new reference/XmlizeCustomValue example.

As for the problem of registering non-Core Values, I have found only one Value compatible type where this is a sort of problem: Font. I think I can live with that for now...

Mirek

Perfect
I removed my quick-and-dirty value xmlizer and used yours in my app. Different file format, indeed, so I'm happy it happened before deploying my app

Ciao

Max

Subject: Re: Xmlize values
Posted by mirek on Sun, 28 Feb 2010 22:09:41 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 28 February 2010 11:32Hello Mirek

Quote:Actually, strange - I was coming from your sources. What has changed?
Before there was a separate "type" and a "value" for all types.

Now it is not exactly the same, but no problem :

<item type="String">This is a text</item>
<item type="Time" value="20020101T00:00:00"/>

Quote:There is one generic Ctrl::Xmlize

Where is it ?

In Ctrl::Xmlize?

Mirek

Subject: Re: Xmlize values
Posted by mdelfede on Mon, 01 Mar 2010 08:53:22 GMT
View Forum Message <> Reply to Message

Btw, I was wondering why for string type the 'value' tag is missing..... or why it is present for other value types.

Max

Subject: Re: Xmlize values
Posted by mirek on Mon, 01 Mar 2010 09:20:18 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 01 March 2010 03:53Btw, I was wondering why for string type the 'value' tag is missing..... or why it is present for other value types.

Max

Frankly, there is no hard reason, except maybe estetics.

Well, maybe we should actually do without "value" attribute (putting all values as text between two tags). That would be more code in Xmlize.cpp.. (because for basic types, we like to have them as attributes too).

But for plain text, it just seemed too weird to put it with attr:)

Mirek

Subject: Re: Xmlize values
Posted by Mindtraveller on Mon, 01 Mar 2010 09:53:00 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 01 March 2010 12:20But for plain text, it just seemed too weird to put it with attr:)May be not, if xmlizing text will require different call. Value may represent different types with the same interface, so should value's xmlization too. This will be right solution IMO.

Subject: Re: Xmlize values
Posted by mdelfede on Mon, 01 Mar 2010 10:08:41 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 01 March 2010 10:20
Frankly, there is no hard reason, except maybe estetics.

Well, maybe we should actually do without "value" attribute (putting all values as text between two tags). That would be more code in Xmlize.cpp.. (because for basic types, we like to have them as attributes too).

But for plain text, it just seemed too weird to put it with attr:)

Mirek

Ah, no need to change (please, don't do !), I was just curious.

Ciao

Max

Subject: Re: Xmlize values
Posted by Mindtraveller on Mon, 08 Mar 2010 23:12:22 GMT
View Forum Message <> Reply to Message

Recently I had runtime exception with this code when trying to load from XML file:

```
class XXXXX: public XXXXParent
{
public:
 virtual void Xmlize(XmlIO &xml)
 {
  XXXXParent::Xmlize(xml);
  xml.Attr("id", id);
  xml ("v", v); // <-- exception here!!
 }
private:
 String id;
 Value v;
};
```

file contained these tags:

Quote:

```
<element type="2" i="5404" x="24" y="17" l="20" dir="0" dir2="-1" input0="4587"
input1="0" link0="5405" link1="0" id="asd">
   <v type="String">234234234234234</v>
  </element>
```

Subject: Re: Xmlize values
Posted by mirek on Tue, 09 Mar 2010 08:35:42 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Mon, 08 March 2010 18:12Recently I had runtime exception with this code when trying to load from XML file:

```
class XXXXX: public XXXXParent
{
```

```
public:
 virtual void Xmlize(XmlIO &xml)
 {
  XXXXParent::Xmlize(xml);
  xml.Attr("id", id);
  xml ("v", v); // <-- exception here!!
 }
private:
 String id;
 Value v;
};
```

file contained these tags:
Quote: <element type="2" i="5404" x="24" y="17" l="20" dir="0" dir2="-1" input0="4587" input1="0" link0="5405" link1="0" id="asd">
  <v type="String">234234234234234</v>
 </element>

What about to tell us about XXXParent::Xmlize? Or kind of runtime exception?

Full testcase would be highly appreciated!

Mirek

Subject: Re: Xmlize values
Posted by Mindtraveller on Tue, 09 Mar 2010 12:40:17 GMT
View Forum Message <> Reply to Message

Finally the problem is with de-xmlizing Value set to some integer value with stored something like String.

```
#include <Core/Core.h>
using namespace Upp;

struct A
{
 void Xmlize(XmlIO &xml) { xml ("v",v); } //<-- exception on 2nd call

 Value v;
};

CONSOLE_APP_MAIN
{
 A a;
 a.v = "test";
 StoreAsXMLFile(a, "XmlizeTest", "xmlizeTtest");
```

```
 a.v = 0;
 LoadFromXMLFile(a, "xmlizeTtest");
}
```

Subject: Re: Xmlize values
Posted by mirek on Wed, 10 Mar 2010 07:55:32 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Tue, 09 March 2010 07:40Finally the problem is with de-xmlizing Value set to some integer value with stored something like String.

```
#include <Core/Core.h>
using namespace Upp;

struct A
{
 void Xmlize(XmlIO &xml) { xml ("v",v); } //<-- exception on 2nd call

 Value v;
};

CONSOLE_APP_MAIN
{
 A a;
 a.v = "test";
 StoreAsXMLFile(a, "XmlizeTest", "xmlizeTtest");
 a.v = 0;
 LoadFromXMLFile(a, "xmlizeTtest");
}
```

Thank you, fixed.

Mirek