Subject: Xmlize has problem with WithDeepCopy<Array<...
Posted by nixnixnix on Wed, 31 Oct 2012 01:01:23 GMT
View Forum Message <> Reply to Message

Hello,

I am trying to use Xmlize to save data to file in human readable format. It appears that I can do this with my own objects so long as I provide Xmlize functions like with Serialize. Also it appears the Array and Vector have Xmlize capability. My problem appears to be that I declare my Arrays and Vectors with WithDeepCopy and Xmlize does not like this. Is there a fix please?

c:\upp\uppsrc\core\Xmlize.h(71) : error C2039: 'Xmlize' : is not a member of 'Upp::WithDeepCopy<T>'
with
[
T=Upp::Vector<double>
]
c:\upp\uppsrc\core\Xmlize.h(82) : see reference to function template instantiation 'void Upp::Xmlize<T>(Upp::XmlIO &,T &)' being compiled
with
[
T=doubleArray
]
d:\AWSopenWind\trunk\OpenWind\TurbineType.cpp(253) : see reference to function template instantiation 'Upp::XmlIO Upp::XmlIO::operator ()<doubleArray>(const char *,T &)' being compiled
with
[
T=doubleArray
]

doubleArray is declared as

typedef WithDeepCopy<Vector <double> > doubleArray;

Thanks,

Nick

Subject: Re: Xmlize has problem with WithDeepCopy<Array<...
Posted by Mindtraveller on Wed, 31 Oct 2012 05:36:34 GMT

View Forum Message <> Reply to Message

I do use xmlization for U++ arrays without a problem.
The error you posted seems to me strange as WithDeepCopy is a pure helper-class and must not lead to any problems of this sort.
Could you post minimal excerpt from your code to show how your "problematic" classes are organized?

Subject: Re: Xmlize has problem with WithDeepCopy<Array<...
Posted by mirek on Thu, 01 Nov 2012 19:25:04 GMT
View Forum Message <> Reply to Message

Caused by template matching rules.

Fixed by refactoring Xmlize/Jsonize of containers to methods. Should work now (upptst/ize is a unit test for this issue).

Mirek

Subject: Re: Xmlize has problem with WithDeepCopy<Array<...
Posted by nixnixnix on Thu, 01 Nov 2012 21:45:51 GMT
View Forum Message <> Reply to Message

Thanks. I just downloaded version 5503 but still get the same error.

MindTraveller, there really wasn't anything I could add. Every instance of an Array or Vector that I have declared with WithDeepCopy gives the same error when I try to Xmlize with something like

```
void TurbineTable::Xmlize(XmlIO& xml)
{
 DUMP("In TurbineTable::Xmlize");

 String sType = "TurbineTable";

 xml
 ("Type",sType)
 ("TI_min",m_fTImin)
 ("TI_max",m_fTImax) // these work
 ;

 xml
 ("Velocity",m_fV) // << this is a double array which is typedef WithDeepCopy < Vector <double> >
doubleArray
 ("AirDensity",m_fAirDensity) // << this is a double array which is typedef WithDeepCopy < Vector
<double> > doubleArray
```

;

As a workaround I tried the following for 1 and 2 dimensional arrays.

```
 template<class T> static bool XmlizeArray1D(XmlIO& xml,String sCollection,String sItem,T& ar)
{
 DUMP("XmlizeArray1D");

 XmlIO& xmla = xml.Add(sCollection);

 int n;

 if(xml.IsStoring())
 {
  n = ar.GetCount();
  xmla("Count",n);
 }
 else // IsLoading
 {
  xmla("Count",n);
  if(n>=0)
   ar.SetCount(n);
  else
   return false;
 }

 for(int i=0;i<n;i++)
 {
  String s = Format("%s%d",sItem,i);
  xmla(s,ar[i]);
 }

 return true;
}

 template<class S,class T> static bool XmlizeArray2D(XmlIO& xml,String sCollection,String
Column,S& ss,T& ar)
{
 DUMP("XmlizeArray2D");

 XmlIO& xmla = xml.Add(sCollection);

 int n1,n2;
```

```
if(xml.IsStoring())
{
 n1 = ar.GetCount();
 xmla("Columns",n1);

 for(int i=0;i<n1;i++)
 {
  String sHeader = Format("%s%f",Column,ss[i]);
  XmlIO& xmlb = xmla.Add(sHeader);

  n2 = ar[i].GetCount();
  xmlb("Rows",n2);

  for(int j=0;j<n2;j++)
  {
   String label = Format("v%d-%d",i,j);

   xmlb(label,ar[i][j]);
  }
 }
}
else // IsLoading
{
 xmla("Columns",n1);
 ar.SetCount(n1);

 DUMP(n1);

 for(int i=0;i<n1;i++)
 {
  String sHeader = Format("%s%f",Column,ss[i]);
  XmlIO& xmlb = xmla.Add(sHeader);

  xmlb("Rows",n2);
  ar[i].SetCount(n2);

  DUMP(n2);

  for(int j=0;j<n2;j++)
  {
   String label = Format("v%d-%d",i,j);

   xmlb(label,ar[i][j]);
  }
 }
}

return true;
```

```
}
```

and it appears to write good clear XML but it wont read back in. It could be that I am misusing the Add() as I have never been clear really on how the XML should work.

My XML file looks like the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE AlstomECOClassA>
<AlstomECOClassA>
 <Name>Alstom ECO 80 1.67 Class 2A</Name>
 <HubHeight value="80"/>
 <RotorDiameter value="80"/>
 <VoltageV value="690"/>
 <CapacityKW value="1670"/>
 <IsPitchControlled value="1"/>
 <LowCutIn value="3"/>
 <HighCutOut value="25"/>
 <IEC_adjustment value="1"/>
 <NumBlades value="3"/>
 <LowTemperatureShutDown value="-30"/>
 <HighTemperatureShutDown value="45"/>
 <LowTemperatureUnits value="0"/>
 <HighTemperatureUnits value="0"/>
 <LowTemperatureRestart value="-20"/>
 <HighTemperatureRestart value="30"/>
 <IsVariableSpeed value="1"/>
 <TiltAngleDegrees value="5"/>
 <PeakOutputKW value="1670"/>
 <SpeedClass value="2"/>
 <TiClass value="0"/>
 <SpeedMax value="8.5"/>
 <TiMax value="16"/>
 <Comments>This is not a warrantied power curve. It is for demonstration purposes only. If you
need a warrantied power curve, please contact your turbine manufacturer.</Comments>
 <Power_Tables>
  <Count value="1"/>
  <Power_Table0>
   <Type>TurbineTable</Type>
   <TI_min value="0"/>
   <TI_max value="60"/>
   <Velocities>
    <Count value="26"/>
    <Velocity0 value="0"/>
    <Velocity1 value="1"/>
```

<Velocity2 value="2"/>
<Velocity3 value="3"/>
<Velocity4 value="4"/>
<Velocity5 value="5"/>
<Velocity6 value="6"/>
<Velocity7 value="7"/>
<Velocity8 value="8"/>
<Velocity9 value="9"/>
<Velocity10 value="10"/>
<Velocity11 value="11"/>
<Velocity12 value="12"/>
<Velocity13 value="13"/>
<Velocity14 value="14"/>
<Velocity15 value="15"/>
<Velocity16 value="16"/>
<Velocity17 value="17"/>
<Velocity18 value="18"/>
<Velocity19 value="19"/>
<Velocity20 value="20"/>
<Velocity21 value="21"/>
<Velocity22 value="22"/>
<Velocity23 value="23"/>
<Velocity24 value="24"/>
<Velocity25 value="25"/>
</Velocities>
<AirDensities>
<Count value="6"/>
<Rho0 value="1"/>
<Rho1 value="1.05"/>
<Rho2 value="1.1"/>
<Rho3 value="1.15"/>
<Rho4 value="1.2"/>
<Rho5 value="1.225"/>
</AirDensities>
<Values>
<Columns value="6"/>
<Rho1.000000>
<Rows value="26"/>
<v0-0 value="0"/>
<v0-1 value="0"/>
<v0-2 value="0"/>
<v0-3 value="8"/>
<v0-4 value="53"/>
<v0-5 value="129"/>
<v0-6 value="246"/>
<v0-7 value="409"/>
<v0-8 value="626"/>
<v0-9 value="904"/>

<v0-10 value="1231"/>
<v0-11 value="1522"/>
<v0-12 value="1637"/>
<v0-13 value="1662"/>
<v0-14 value="1670"/>
<v0-15 value="1670"/>
<v0-16 value="1665"/>
<v0-17 value="1614"/>
<v0-18 value="1517"/>
<v0-19 value="1409"/>
<v0-20 value="1300"/>
<v0-21 value="1191"/>
<v0-22 value="1083"/>
<v0-23 value="974"/>
<v0-24 value="867"/>
<v0-25 value="758"/>
</Rho1.000000>
<Rho1.050000>
<Rows value="26"/>
<v1-0 value="0"/>
<v1-1 value="0"/>
<v1-2 value="0"/>
<v1-3 value="8"/>
<v1-4 value="53"/>
<v1-5 value="129"/>
<v1-6 value="246"/>
<v1-7 value="409"/>
<v1-8 value="626"/>
<v1-9 value="904"/>
<v1-10 value="1231"/>
<v1-11 value="1522"/>
<v1-12 value="1637"/>
<v1-13 value="1662"/>
<v1-14 value="1670"/>
<v1-15 value="1670"/>
<v1-16 value="1665"/>
<v1-17 value="1614"/>
<v1-18 value="1517"/>
<v1-19 value="1409"/>
<v1-20 value="1300"/>
<v1-21 value="1191"/>
<v1-22 value="1083"/>
<v1-23 value="974"/>
<v1-24 value="867"/>
<v1-25 value="758"/>
</Rho1.050000>
<Rho1.100000>
<Rows value="26"/>

<v2-0 value="0"/>
<v2-1 value="0"/>
<v2-2 value="0"/>
<v2-3 value="7"/>
<v2-4 value="49"/>
<v2-5 value="123"/>
<v2-6 value="233"/>
<v2-7 value="390"/>
<v2-8 value="597"/>
<v2-9 value="864"/>
<v2-10 value="1182"/>
<v2-11 value="1480"/>
<v2-12 value="1626"/>
<v2-13 value="1658"/>
<v2-14 value="1670"/>
<v2-15 value="1670"/>
<v2-16 value="1665"/>
<v2-17 value="1614"/>
<v2-18 value="1517"/>
<v2-19 value="1409"/>
<v2-20 value="1300"/>
<v2-21 value="1191"/>
<v2-22 value="1083"/>
<v2-23 value="974"/>
<v2-24 value="867"/>
<v2-25 value="758"/>
</Rho1.100000>
<Rho1.150000>
<Rows value="26"/>
<v3-0 value="0"/>
<v3-1 value="0"/>
<v3-2 value="0"/>
<v3-3 value="8"/>
<v3-4 value="53"/>
<v3-5 value="129"/>
<v3-6 value="246"/>
<v3-7 value="409"/>
<v3-8 value="626"/>
<v3-9 value="904"/>
<v3-10 value="1231"/>
<v3-11 value="1522"/>
<v3-12 value="1637"/>
<v3-13 value="1662"/>
<v3-14 value="1670"/>
<v3-15 value="1670"/>
<v3-16 value="1665"/>
<v3-17 value="1614"/>
<v3-18 value="1517"/>

<v3-19 value="1409"/>
<v3-20 value="1300"/>
<v3-21 value="1191"/>
<v3-22 value="1083"/>
<v3-23 value="974"/>
<v3-24 value="867"/>
<v3-25 value="758"/>
</Rho1.150000>
<Rho1.200000>
<Rows value="26"/>
<v4-0 value="0"/>
<v4-1 value="0"/>
<v4-2 value="0"/>
<v4-3 value="9"/>
<v4-4 value="56"/>
<v4-5 value="136"/>
<v4-6 value="258"/>
<v4-7 value="428"/>
<v4-8 value="654"/>
<v4-9 value="944"/>
<v4-10 value="1277"/>
<v4-11 value="1558"/>
<v4-12 value="1643"/>
<v4-13 value="1663"/>
<v4-14 value="1670"/>
<v4-15 value="1670"/>
<v4-16 value="1658"/>
<v4-17 value="1595"/>
<v4-18 value="1488"/>
<v4-19 value="1380"/>
<v4-20 value="1269"/>
<v4-21 value="1159"/>
<v4-22 value="1050"/>
<v4-23 value="939"/>
<v4-24 value="829"/>
<v4-25 value="720"/>
</Rho1.200000>
<Rho1.225000>
<Rows value="26"/>
<v5-0 value="0"/>
<v5-1 value="0"/>
<v5-2 value="0"/>
<v5-3 value="9"/>
<v5-4 value="58"/>
<v5-5 value="139"/>
<v5-6 value="263"/>
<v5-7 value="437"/>
<v5-8 value="669"/>

```
    <v5-9 value="964"/>
    <v5-10 value="1300"/>
    <v5-11 value="1580"/>
    <v5-12 value="1647"/>
    <v5-13 value="1666"/>
    <v5-14 value="1670"/>
    <v5-15 value="1670"/>
    <v5-16 value="1655"/>
    <v5-17 value="1583"/>
    <v5-18 value="1476"/>
    <v5-19 value="1365"/>
    <v5-20 value="1254"/>
    <v5-21 value="1144"/>
    <v5-22 value="1033"/>
    <v5-23 value="922"/>
    <v5-24 value="812"/>
    <v5-25 value="701"/>
   </Rho1.225000>
  </Values>
 </Power_Table0>
</Power_Tables>
<Thrust_Table>
 <Type>TurbineTable</Type>
 <TI_min value="0"/>
 <TI_max value="60"/>
 <Velocities>
  <Count value="26"/>
  <Velocity0 value="0"/>
  <Velocity1 value="1"/>
  <Velocity2 value="2"/>
  <Velocity3 value="3"/>
  <Velocity4 value="4"/>
  <Velocity5 value="5"/>
  <Velocity6 value="6"/>
  <Velocity7 value="7"/>
  <Velocity8 value="8"/>
  <Velocity9 value="9"/>
  <Velocity10 value="10"/>
  <Velocity11 value="11"/>
  <Velocity12 value="12"/>
  <Velocity13 value="13"/>
  <Velocity14 value="14"/>
  <Velocity15 value="15"/>
  <Velocity16 value="16"/>
  <Velocity17 value="17"/>
  <Velocity18 value="18"/>
  <Velocity19 value="19"/>
  <Velocity20 value="20"/>
```

```
 <Velocity21 value="21"/>
 <Velocity22 value="22"/>
 <Velocity23 value="23"/>
 <Velocity24 value="24"/>
 <Velocity25 value="25"/>
</Velocities>
<AirDensities>
 <Count value="6"/>
 <Rho0 value="1"/>
 <Rho1 value="1.05"/>
 <Rho2 value="1.1"/>
 <Rho3 value="1.15"/>
 <Rho4 value="1.2"/>
 <Rho5 value="1.225"/>
</AirDensities>
<Values>
 <Columns value="6"/>
 <Rho1.000000>
  <Rows value="26"/>
  <v0-0 value="0"/>
  <v0-1 value="0"/>
  <v0-2 value="0"/>
  <v0-3 value="1.02"/>
  <v0-4 value="0.85"/>
  <v0-5 value="0.74"/>
  <v0-6 value="0.75"/>
  <v0-7 value="0.76"/>
  <v0-8 value="0.76"/>
  <v0-9 value="0.75"/>
  <v0-10 value="0.69"/>
  <v0-11 value="0.64"/>
  <v0-12 value="0.46"/>
  <v0-13 value="0.35"/>
  <v0-14 value="0.27"/>
  <v0-15 value="0.22"/>
  <v0-16 value="0.18"/>
  <v0-17 value="0.15"/>
  <v0-18 value="0.12"/>
  <v0-19 value="0.09"/>
  <v0-20 value="0.07"/>
  <v0-21 value="0.06"/>
  <v0-22 value="0.05"/>
  <v0-23 value="0.04"/>
  <v0-24 value="0.03"/>
  <v0-25 value="0.02"/>
 </Rho1.000000>
 <Rho1.050000>
  <Rows value="26"/>
```

<v1-0 value="0"/>
<v1-1 value="0"/>
<v1-2 value="0"/>
<v1-3 value="1.02"/>
<v1-4 value="0.85"/>
<v1-5 value="0.74"/>
<v1-6 value="0.75"/>
<v1-7 value="0.76"/>
<v1-8 value="0.76"/>
<v1-9 value="0.75"/>
<v1-10 value="0.69"/>
<v1-11 value="0.64"/>
<v1-12 value="0.46"/>
<v1-13 value="0.35"/>
<v1-14 value="0.27"/>
<v1-15 value="0.22"/>
<v1-16 value="0.18"/>
<v1-17 value="0.15"/>
<v1-18 value="0.12"/>
<v1-19 value="0.09"/>
<v1-20 value="0.07"/>
<v1-21 value="0.06"/>
<v1-22 value="0.05"/>
<v1-23 value="0.04"/>
<v1-24 value="0.03"/>
<v1-25 value="0.02"/>
</Rho1.050000>
<Rho1.100000>
<Rows value="26"/>
<v2-0 value="0"/>
<v2-1 value="0"/>
<v2-2 value="0"/>
<v2-3 value="1.02"/>
<v2-4 value="0.85"/>
<v2-5 value="0.74"/>
<v2-6 value="0.75"/>
<v2-7 value="0.76"/>
<v2-8 value="0.76"/>
<v2-9 value="0.75"/>
<v2-10 value="0.69"/>
<v2-11 value="0.64"/>
<v2-12 value="0.46"/>
<v2-13 value="0.35"/>
<v2-14 value="0.27"/>
<v2-15 value="0.22"/>
<v2-16 value="0.18"/>
<v2-17 value="0.15"/>
<v2-18 value="0.12"/>

<v2-19 value="0.09"/>
<v2-20 value="0.07"/>
<v2-21 value="0.06"/>
<v2-22 value="0.05"/>
<v2-23 value="0.04"/>
<v2-24 value="0.03"/>
<v2-25 value="0.02"/>
</Rho1.100000>
<Rho1.150000>
<Rows value="26"/>
<v3-0 value="0"/>
<v3-1 value="0"/>
<v3-2 value="0"/>
<v3-3 value="1.02"/>
<v3-4 value="0.85"/>
<v3-5 value="0.74"/>
<v3-6 value="0.75"/>
<v3-7 value="0.76"/>
<v3-8 value="0.76"/>
<v3-9 value="0.75"/>
<v3-10 value="0.69"/>
<v3-11 value="0.64"/>
<v3-12 value="0.46"/>
<v3-13 value="0.35"/>
<v3-14 value="0.27"/>
<v3-15 value="0.22"/>
<v3-16 value="0.18"/>
<v3-17 value="0.15"/>
<v3-18 value="0.12"/>
<v3-19 value="0.09"/>
<v3-20 value="0.07"/>
<v3-21 value="0.06"/>
<v3-22 value="0.05"/>
<v3-23 value="0.04"/>
<v3-24 value="0.03"/>
<v3-25 value="0.02"/>
</Rho1.150000>
<Rho1.200000>
<Rows value="26"/>
<v4-0 value="0"/>
<v4-1 value="0"/>
<v4-2 value="0"/>
<v4-3 value="1.02"/>
<v4-4 value="0.85"/>
<v4-5 value="0.75"/>
<v4-6 value="0.76"/>
<v4-7 value="0.76"/>
<v4-8 value="0.77"/>

<v4-9 value="0.75"/>
<v4-10 value="0.69"/>
<v4-11 value="0.6"/>
<v4-12 value="0.44"/>
<v4-13 value="0.34"/>
<v4-14 value="0.26"/>
<v4-15 value="0.21"/>
<v4-16 value="0.17"/>
<v4-17 value="0.14"/>
<v4-18 value="0.11"/>
<v4-19 value="0.09"/>
<v4-20 value="0.07"/>
<v4-21 value="0.06"/>
<v4-22 value="0.04"/>
<v4-23 value="0.04"/>
<v4-24 value="0.03"/>
<v4-25 value="0.02"/>
</Rho1.200000>
<Rho1.225000>
<Rows value="26"/>
<v5-0 value="0"/>
<v5-1 value="0"/>
<v5-2 value="0"/>
<v5-3 value="1.02"/>
<v5-4 value="0.85"/>
<v5-5 value="0.76"/>
<v5-6 value="0.76"/>
<v5-7 value="0.77"/>
<v5-8 value="0.77"/>
<v5-9 value="0.75"/>
<v5-10 value="0.69"/>
<v5-11 value="0.59"/>
<v5-12 value="0.44"/>
<v5-13 value="0.33"/>
<v5-14 value="0.26"/>
<v5-15 value="0.21"/>
<v5-16 value="0.17"/>
<v5-17 value="0.14"/>
<v5-18 value="0.11"/>
<v5-19 value="0.08"/>
<v5-20 value="0.07"/>
<v5-21 value="0.05"/>
<v5-22 value="0.04"/>
<v5-23 value="0.03"/>
<v5-24 value="0.03"/>
<v5-25 value="0.02"/>
</Rho1.225000>
</Values>

```
</Thrust_Table>
<RPM_Table>
 <Type>TurbineTable</Type>
 <TI_min value="0"/>
 <TI_max value="60"/>
 <Velocities>
  <Count value="26"/>
  <Velocity0 value="0"/>
  <Velocity1 value="1"/>
  <Velocity2 value="2"/>
  <Velocity3 value="3"/>
  <Velocity4 value="4"/>
  <Velocity5 value="5"/>
  <Velocity6 value="6"/>
  <Velocity7 value="7"/>
  <Velocity8 value="8"/>
  <Velocity9 value="9"/>
  <Velocity10 value="10"/>
  <Velocity11 value="11"/>
  <Velocity12 value="12"/>
  <Velocity13 value="13"/>
  <Velocity14 value="14"/>
  <Velocity15 value="15"/>
  <Velocity16 value="16"/>
  <Velocity17 value="17"/>
  <Velocity18 value="18"/>
  <Velocity19 value="19"/>
  <Velocity20 value="20"/>
  <Velocity21 value="21"/>
  <Velocity22 value="22"/>
  <Velocity23 value="23"/>
  <Velocity24 value="24"/>
  <Velocity25 value="25"/>
 </Velocities>
 <AirDensities>
  <Count value="1"/>
  <Rho0 value="1.225"/>
 </AirDensities>
 <Values>
  <Columns value="1"/>
  <Rho1.225000>
   <Rows value="26"/>
   <v0-0 value="0"/>
   <v0-1 value="0"/>
   <v0-2 value="0"/>
   <v0-3 value="11"/>
   <v0-4 value="12"/>
   <v0-5 value="13"/>
```

```
   <v0-6 value="14"/>
   <v0-7 value="15"/>
   <v0-8 value="16"/>
   <v0-9 value="17"/>
   <v0-10 value="18"/>
   <v0-11 value="19"/>
   <v0-12 value="20"/>
   <v0-13 value="20.16"/>
   <v0-14 value="20.32"/>
   <v0-15 value="20.48"/>
   <v0-16 value="20.64"/>
   <v0-17 value="20.8"/>
   <v0-18 value="20.96"/>
   <v0-19 value="21.12"/>
   <v0-20 value="21.28"/>
   <v0-21 value="21.44"/>
   <v0-22 value="21.6"/>
   <v0-23 value="21.76"/>
   <v0-24 value="21.92"/>
   <v0-25 value="22"/>
  </Rho1.225000>
 </Values>
</RPM_Table>
<TotalCost value="2000000"/>
<FoundationCost value="100000"/>
<PeriodicCosts>
 <Count value="2"/>
 <PeriodicCost0>
  <Type>PeriodicCost</Type>
  <Component>Drive Train</Component>
  <Cost value="300000"/>
  <PeriodYears value="7"/>
  <CostVariable value="0"/>
  <PeriodVariable value="0"/>
  <IsVariablePeriod value="0"/>
  <IsVariableCost value="0"/>
  <CostExponent value="1"/>
  <PeriodExponent value="1"/>
  <CostFactor value="1"/>
  <PeriodFactor value="1"/>
 </PeriodicCost0>
 <PeriodicCost1>
  <Type>PeriodicCost</Type>
  <Component>Blades</Component>
  <Cost value="200000"/>
  <PeriodYears value="15"/>
  <CostVariable value="0"/>
  <PeriodVariable value="0"/>
```

```
   <IsVariablePeriod value="0"/>
   <IsVariableCost value="0"/>
   <CostExponent value="1"/>
   <PeriodExponent value="1"/>
   <CostFactor value="1"/>
   <PeriodFactor value="1"/>
  </PeriodicCost1>
 </PeriodicCosts>
 <TotalNoise value="100"/>
 <TonalNoise value="100"/>
 <Noise63hz value="0"/>
 <Noise125hz value="0"/>
 <Noise250hz value="0"/>
 <Noise500hz value="0"/>
 <Noise1000hz value="0"/>
 <Noise2000hz value="0"/>
 <Noise40000hz value="0"/>
 <Noise80000hz value="0"/>
</AlstomECOClassA>
```

Which appears exactly as I would want it but for some reason it will not Xmlize back in

It would be good if XmlIO gave more errors when it can't find something. When I debug, it appears to find the Columns variable "....s=6??..." but instead sets my n1 variable in the 2D array to

Any suggestions welcome.

Nick

Update:

It is the Add() function that is causing me problems. When I use

```
 template<class T> static bool XmlizeArray1D(XmlIO& xml,String sCollection,String sItem,T& ar)
 {
  DUMP("XmlizeArray1D");

//  XmlIO& xmla = xml.Add(sCollection);

  XmlIO& xmla = xml;

  int n;

  if(xml.IsStoring())
```

```
  {
   n = ar.GetCount();
   DUMP(n);
   xmla("Count",n);
  }
  else // IsLoading
  {
   xmla("Count",n);

   DUMP(n);

   if(n>=0)
    ar.SetCount(n);
   else
    return false;
  }

  for(int i=0;i<n;i++)
  {
   String s = Format("%s%d",sItem,i);
   xmla(s,ar[i]);
  }

  return true;
 }

 template<class S,class T> static bool XmlizeArray2D(XmlIO& xml,String sCollection,String
Column,S& ss,T& ar)
 {
  DUMP("XmlizeArray2D");

//  XmlIO& xmla = xml.Add(sCollection);
  XmlIO& xmla = xml;

  int n1,n2;

  if(xml.IsStoring())
  {
   n1 = ar.GetCount();
   xmla("Columns",n1);

   for(int i=0;i<n1;i++)
   {
    String sHeader = Format("%s%f",Column,ss[i]);
///    XmlIO& xmlb = xmla.Add(sHeader);
    XmlIO& xmlb = xml;

    n2 = ar[i].GetCount();
```

```
    xmlb("Rows",n2);

    for(int j=0;j<n2;j++)
    {
     String label = Format("v%d-%d",i,j);

     xmlb(label,ar[i][j]);
    }
   }
  }
  else // IsLoading
  {
   xmla("Columns",n1);
   ar.SetCount(n1);

   DUMP(n1);

   for(int i=0;i<n1;i++)
   {
    String sHeader = Format("%s%f",Column,ss[i]);
//    XmlIO& xmlb = xmla.Add(sHeader);
    XmlIO& xmlb = xml;

    xmlb("Rows",n2);
    ar[i].SetCount(n2);

    DUMP(n2);

    for(int j=0;j<n2;j++)
    {
     String label = Format("v%d-%d",i,j);

     xmlb(label,ar[i][j]);
    }
   }
  }

  return true;
 }
```

instead then it reads back in fine but this means that each object can only contain one 1D array or vector and 1 2D array or vector as the Count variables will overwrite each other. I need to use the Add(sCollection) in the original functions in order to make them properly useful. Is there any alternative that would achieve the same thing please?

Subject: Re: Xmlize has problem with WithDeepCopy<Array<...
Posted by nixnixnix on Fri, 02 Nov 2012 00:47:20 GMT
View Forum Message <> Reply to Message

OK got it. Not sure why Xmlize is not working for WithDeepCopy <Vector <double> > but in any case the following two functions give slightly more functionality

```
template<class T> static bool XmlizeArray1D(XmlIO& xml,String sCollection,String sItem,T& ar)
{
 DUMP("XmlizeArray1D");

 int n;

 if(xml.IsStoring())
 {
  XmlIO xmla = xml.Add(sCollection);

  n = ar.GetCount();
  DUMP(n);
  xmla("Count",n);

  for(int i=0;i<n;i++)
  {
   String s = Format("%s%d",sItem,i);
   xmla(s,ar[i]);
  }

 }
 else // IsLoading
 {
  int ii = -1;
  for(int i = 0; i < xml->GetCount(); i++)
  {
   if(xml->Node(i).IsTag(sCollection))
    ii = i;
  }

  if(ii<0)
  {
   DUMP("FAILED IN XmlizeArray1D");
   return false;
  }
  XmlIO xmla = xml.At(ii);

  xmla("Count",n);

  DUMP(n);
```

```
  if(n>=0)
   ar.SetCount(n);
  else
   return false;

  for(int i=0;i<n;i++)
  {
   String s = Format("%s%d",sItem,i);
   xmla(s,ar[i]);
  }

 }


 return true;
}

 template<class S,class T> static bool XmlizeArray2D(XmlIO& xml,String sCollection,String
Column,S& ss,T& ar)
{
 DUMP("XmlizeArray2D");

 int n1,n2;

 if(xml.IsStoring())
 {
  XmlIO xmla = xml.Add(sCollection);

  n1 = ar.GetCount();
  xmla("Columns",n1);

  for(int i=0;i<n1;i++)
  {
   String sHeader = Format("%s%f",Column,ss[i]);
   XmlIO xmlb = xmla.Add(sHeader);

   n2 = ar[i].GetCount();
   xmlb("Rows",n2);

   for(int j=0;j<n2;j++)
   {
    String label = Format("v%d-%d",i,j);

    xmlb(label,ar[i][j]);
   }
  }
 }
 else // IsLoading
```

```
{
 int ii = -1;
 for(int i = 0; i < xml->GetCount(); i++)
 {
  if(xml->Node(i).IsTag(sCollection))
   ii = i;
 }

 if(ii<0)
 {
  DUMP("FAILED IN XmlizeArray2D1");
  return false;
 }

 XmlIO xmla = xml.At(ii);

 xmla("Columns",n1);
 ar.SetCount(n1);

 DUMP(n1);

 for(int i=0;i<n1;i++)
 {
  String sHeader = Format("%s%f",Column,ss[i]);

  int jj = -1;
  for(int j = 0; j < xmla->GetCount(); j++)
  {
   if(xmla->Node(j).IsTag(sHeader))
    jj = j;
  }

  if(jj<0)
  {
   DUMP("FAILED IN XmlizeArray2D2");
   return false;
  }

  XmlIO xmlb = xmla.At(jj);

  xmlb("Rows",n2);
  ar[i].SetCount(n2);

  DUMP(n2);

  for(int j=0;j<n2;j++)
  {
   String label = Format("v%d-%d",i,j);
```

```
    xmlb(label,ar[i][j]);
   }
  }
 }

 return true;
}
```

Subject: Re: Xmlize has problem with WithDeepCopy<Array<...
Posted by mirek on Fri, 02 Nov 2012 06:55:47 GMT
View Forum Message <> Reply to Message

5503 is not enough, 5506 is the commit solving this problem.

No need to bury into this deeper, the reason why it was not working with WithDeepCopy is apparent (WithDeepCopy<Vector<T>> is not the same type as Vector<T> and template specialization for Xmlize was defined for Vector<T> only).

It is now fixed by moving Xmlize as Vector method, thus completely avoiding template specializations.

Mirek
