Subject: Unterminated processing info in XmlParser
Posted by nixnixnix on Mon, 20 Jan 2014 23:55:39 GMT
View Forum Message <> Reply to Message

Hi,

I have an interested puzzle. When my programme is called from command line with an XML script it works fine but when it is called using a Python programme using MDAO then it throws the following error

Unterminated processing info

which I searched for and found in XmlParser::Next()

It causes everything to crash but I have no idea why. It was working fine with my programme from 8 months or so ago and I don't think I changed anything that wold affect it but not sure. If anyone has any ideas it would be much appreciated as it is very hard to figure out why it would work fine from command line but not when called from MDAO/Python. If there is a workaround that would be great.

Thanks,

Nick

Subject: Re: Unterminated processing info in XmlParser
Posted by mirek on Tue, 21 Jan 2014 12:31:49 GMT
View Forum Message <> Reply to Message

It is definitely possible that it is a bug in U++; would it be possible to post here the .xml file to be parsed?

Are you using just XmlParser, or XmlNode?

Subject: Re: Unterminated processing info in XmlParser
Posted by nixnixnix on Fri, 23 May 2014 23:51:18 GMT
View Forum Message <> Reply to Message

It ended up being a user error but I have a similar and more vexing example now. This code and script worked fine yesterday and then the same executable and the same xml file stopped working which to me is pure nonsense. I have checked the file for corruption but would appreciate your take on it. Here is the code I use:

```
bool OwScripterDlg::LoadS(String path)
{
 FileIn iFile(path);

 if(!iFile.IsOpen())
 {
  return false;
 }

 int len = (int)iFile.GetSize();
 Buffer<char> pBuf;
 pBuf.Alloc(len);

 iFile.GetAll(pBuf,len);

 iFile.Close();

 String xml = ~pBuf;

 XmlNode xn = ParseXML(xml);

 OwScript script;

 int n = xn["OpenWindScript"]["AllOperations"].GetCount(); // gets subtags of OpenWind

 script.m_sPath = xn["OpenWindScript"]["ReportPath"].Attr("value");
 script.m_bAfter = xn["OpenWindScript"]["AppendOperations"].Attr("value")!="Sideways";
 script.m_bArray = xn["OpenWindScript"]["ArrayEfficiencyField"].Attr("value")=="true";
 script.m_bFreeWS = xn["OpenWindScript"]["FreeWindspeedField"].Attr("value")=="true";
 script.m_bGross = xn["OpenWindScript"]["GrossEnergyField"].Attr("value")=="true";
 script.m_bMeanWS = xn["OpenWindScript"]["MeanWindspeedField"].Attr("value")=="true";
 script.m_bNet = xn["OpenWindScript"]["NetEnergyField"].Attr("value")=="true";
 script.m_bSite = xn["OpenWindScript"]["SiteNameField"].Attr("value")=="true";
 script.m_bTI = xn["OpenWindScript"]["TurbulenceTotalField"].Attr("value")=="true";
 script.m_bTT = xn["OpenWindScript"]["TurbineTypeField"].Attr("value")=="true";
 script.m_bLabel = xn["OpenWindScript"]["TurbineLabelField"].Attr("value")=="true";
 script.m_bIndex = xn["OpenWindScript"]["TurbineIndexField"].Attr("value")=="true";
 script.m_bTX = xn["OpenWindScript"]["TurbineXField"].Attr("value")=="true";
 script.m_bTY = xn["OpenWindScript"]["TurbineYField"].Attr("value")=="true";
 script.m_bTI15 = xn["OpenWindScript"]["TI15"].Attr("value")=="true";
 script.m_bAmbientTI = xn["OpenWindScript"]["AmbientTI"].Attr("value")=="true";
 script.m_bArrayEnergy = xn["OpenWindScript"]["ArrayEnergyField"].Attr("value")=="true";

 script.m_ops.SetCount(n);

 for(int i=0;i<n;i++)
 {
```

```
  String type = xn["OpenWindScript"]["AllOperations"][i]["Type"].Attr("value");

  if(type=="Energy Capture")
  {
   script.m_ops[i].op = OP_EC;
   script.m_ops[i].ec.uMin = atof(xn["OpenWindScript"]["AllOperations"][i]["Umin"].Attr("value"));
   script.m_ops[i].ec.uMax = atof(xn["OpenWindScript"]["AllOperations"][i]["Umax"].Attr("value"));
   script.m_ops[i].ec.uStep = atof(xn["OpenWindScript"]["AllOperations"][i]["Ustep"].Attr("value"));
   script.m_ops[i].ec.SetWake(xn["OpenWindScript"]["AllOperations"][i]["WakeModel"].Attr("value"))
;
   script.m_ops[i].ec.m_nDirs =
int(atof(xn["OpenWindScript"]["AllOperations"][i]["TotalDirections"].Attr("value")));
   script.m_ops[i].ec.SetFirstStep(int(atof(xn["OpenWindScript"]["AllOperations"][i]["FirstDirection"].
Attr("value"))));
   script.m_ops[i].ec.SetLastStep(int(atof(xn["OpenWindScript"]["AllOperations"][i]["LastDirection"].
Attr("value"))));
   script.m_ops[i].ec.m_fDirOffset =
atof(xn["OpenWindScript"]["AllOperations"][i]["DirectionOffset"].Attr("value"));
   script.m_ops[i].ec.SetOnly1sector((script.m_ops[i].ec.GetLastStep()!=script.m_ops[i].ec.m_nDirs
-1 || script.m_ops[i].ec.GetFirstStep()!=0));
   script.m_ops[i].ec.SetPXX(max(50.0,atof(xn["OpenWindScript"]["AllOperations"][i]["Pxx"].Attr("va
lue"))));
   script.m_ops[i].ec.m_bTI = true;
  }
  else if(type=="Change Workbook")
  {
   script.m_ops[i].op = OP_BLB;
   script.m_ops[i].path = xn["OpenWindScript"]["AllOperations"][i]["Path"].Attr("value");
  }
  else if(type=="Global Parameters")
  {
   script.m_ops[i].op = OP_GLOBALS;
   script.m_ops[i].globals.SetAdjustToTrueNorth(bool(atof(xn["OpenWindScript"]["AllOperations"][i][
"AdjustToNorth"].Attr("value"))));
   script.m_ops[i].globals.SetAirDensityLapseRate(atof(xn["OpenWindScript"]["AllOperations"][i]["Ai
rDensityLapseRate"].Attr("value")));
   if(xn["OpenWindScript"]["AllOperations"][i].FindTag("SetAirDensityLapseRate"))
   {
    if(int(atof(xn["OpenWindScript"]["AllOperations"][i]["SetAirDensityLapseRate"].Attr("value"))))
     script.m_ops[i].globals.SetAirDensityLapseRate();
    else
     script.m_ops[i].globals.SetTemperatureLapseRate();
   }
   script.m_ops[i].globals.SetTemperatureLapseRate(atof(xn["OpenWindScript"]["AllOperations"][i][
"TemperatureLapseRate"].Attr("value")));
   script.m_ops[i].globals.SetDefaultTI(atof(xn["OpenWindScript"]["AllOperations"][i]["DefaultTurbul
enceIntensity"].Attr("value")));
  }
```

```
  else if(type=="Time-Series Energy Capture")
  {
   int year,day,month,hour,minute;

   script.m_ops[i].op = OP_ECTS;
   script.m_ops[i].ec.SetTimeSeries();

   script.m_ops[i].ec.m_nDirs =
atoi(xn["OpenWindScript"]["AllOperations"][i]["TotalDirections"].Attr("value"));
   script.m_ops[i].ec.SetWake(xn["OpenWindScript"]["AllOperations"][i]["WakeModel"].Attr("value"))
;
   script.m_ops[i].ec.uStep = atof(xn["OpenWindScript"]["AllOperations"][i]["Ustep"].Attr("value"));

   year = atoi(xn["OpenWindScript"]["AllOperations"][i]["StartYear"].Attr("value"));
   month = atoi(xn["OpenWindScript"]["AllOperations"][i]["StartMonth"].Attr("value"));
   day = atoi(xn["OpenWindScript"]["AllOperations"][i]["StartDay"].Attr("value"));
   hour = atoi(xn["OpenWindScript"]["AllOperations"][i]["StartHour"].Attr("value"));
   minute = atoi(xn["OpenWindScript"]["AllOperations"][i]["StartMinute"].Attr("value"));

   script.m_ops[i].ec.SetStart(Time(year,month,day,hour,minute,0));

   year = atoi(xn["OpenWindScript"]["AllOperations"][i]["EndYear"].Attr("value"));
   month = atoi(xn["OpenWindScript"]["AllOperations"][i]["EndMonth"].Attr("value"));
   day = atoi(xn["OpenWindScript"]["AllOperations"][i]["EndDay"].Attr("value"));
   hour = atoi(xn["OpenWindScript"]["AllOperations"][i]["EndHour"].Attr("value"));
   minute = atoi(xn["OpenWindScript"]["AllOperations"][i]["EndMinute"].Attr("value"));

   script.m_ops[i].ec.SetStop(Time(year,month,day,hour,minute,0));

   script.m_ops[i].ec.SetInterval(atoi(xn["OpenWindScript"]["AllOperations"][i]["IntervalSeconds"].At
tr("value")));
   script.m_ops[i].ec.SetScaleToYear(atoi(xn["OpenWindScript"]["AllOperations"][i]["ScaleToYear"].
Attr("value"))!=0);
   script.m_ops[i].ec.SetTimeSeriesTI(atoi(xn["OpenWindScript"]["AllOperations"][i]["TurbulenceInt
ensity"].Attr("value"))!=0);
   script.m_ops[i].ec.SetTempShutdown(atoi(xn["OpenWindScript"]["AllOperations"][i]["Temperatur
eShutdown"].Attr("value"))!=0);
   script.m_ops[i].ec.SetAirDensity(atoi(xn["OpenWindScript"]["AllOperations"][i]["AirDensity"].Attr("
value"))!=0);

   script.m_ops[i].ec.m_fDirOffset =
atof(xn["OpenWindScript"]["AllOperations"][i]["DirectionOffset"].Attr("value"));

   script.m_ops[i].ec.m_bTI = true;
  }
  else if(type=="Replace Met Data")
  {
   script.m_ops[i].op = OP_METDATA;
```

```
   script.m_ops[i].path = xn["OpenWindScript"]["AllOperations"][i]["Path"].Attr("value");
  }
  else if(type=="OCOE")
  {
   script.m_ops[i].op = OP_OCOE;
  }
  else if(type=="OCOE Test")
  {
   script.m_ops[i].op = OP_OCOE_EC;
  }
  else if(type=="Optimise")
  {
   script.m_ops[i].op = OP_OPTIMISE;
  }
  else if(type=="Site Properties")
  {
   script.m_ops[i].op = OP_SITEPROP;

   script.m_ops[i].sName = xn["OpenWindScript"]["AllOperations"][i]["SiteName"].Attr("value");
   script.m_ops[i].bEnable =
atoi(xn["OpenWindScript"]["AllOperations"][i]["Enable"].Attr("value"))!=0;
   script.m_ops[i].bFixed = atoi(xn["OpenWindScript"]["AllOperations"][i]["Fixed"].Attr("value"))!=0;
   script.m_ops[i].bGrow = atoi(xn["OpenWindScript"]["AllOperations"][i]["Grow"].Attr("value"))!=0;
   script.m_ops[i].bOptimise =
atoi(xn["OpenWindScript"]["AllOperations"][i]["IncludeInOptimiser"].Attr("value"))!=0;
   script.m_ops[i].bSwitch =
atoi(xn["OpenWindScript"]["AllOperations"][i]["SetTurbineType"].Attr("value"))!=0;
   script.m_ops[i].sType = xn["OpenWindScript"]["AllOperations"][i]["TurbineType"].Attr("value");
  }
  else if(type=="Exit")
  {
   script.m_ops[i].op = OP_EXIT;
  }
  else if(type=="Replace Turbine Type")
  {
   script.m_ops[i].op = OP_TURBINE;

   script.m_ops[i].sType = xn["OpenWindScript"]["AllOperations"][i]["TurbineName"].Attr("value");
   script.m_ops[i].path = xn["OpenWindScript"]["AllOperations"][i]["TurbinePath"].Attr("value");
  }
 }

 Set(script);

 return true;
}
```

and I have uploaded the file. I get an exception of "Unterminated tag" but when I run it in debug mode it loads up fine so that is puzzling too.

Any help would be greatly appreciated.

Nick

File Attachments
1) testDAWMev_backCast.xml, downloaded 386 times

Subject: Re: Unterminated processing info in XmlParser
Posted by mirek on Sat, 24 May 2014 15:40:20 GMT
View Forum Message <> Reply to Message

The obvious bug I can see is that ParseXml expects zero terminated string, but you are loading the exact size from the file, without adding zero at the end.

Try

```
String xml = LoadFile(path);
if(xml.IsVoid()) return false;
```

instead of

```
 FileIn iFile(path);

 if(!iFile.IsOpen())
 {
  return false;
 }

 int len = (int)iFile.GetSize();
 Buffer<char> pBuf;
 pBuf.Alloc(len);

 iFile.GetAll(pBuf,len);

 iFile.Close();

 String xml = ~pBuf;
```

Subject: Re: Unterminated processing info in XmlParser
Posted by nixnixnix on Mon, 26 May 2014 16:12:22 GMT
View Forum Message <> Reply to Message

Doh! Thanks. I suspect I have made that same mistake elsewhere as well.