Subject: DeXml() incomplete
Posted by Didier on Sun, 13 Jun 2010 22:35:59 GMT
View Forum Message <> Reply to Message

Hi,

I've been trying to put PNG images in an XML file, but it always failed.

After some investigation it seems that DeXml function is incomplete:

It takes a 'char*' as input data and then searches for the '0' value to find end of string !

With PNG images this does not work !

If you change 'char*' to String and use Begin() & End() methods instead of searching for '0' value, then it works.

Here is a sample code that illustrates what I say:

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

#define IMAGECLASS CtrlImg
#define IMAGEFILE <CtrlLib/Ctrl.iml>
#include <Draw/iml_header.h>

#define IMAGECLASS CtrlImg
#define IMAGEFILE <CtrlLib/Ctrl.iml>
#include <Draw/iml_source.h>

String MyDeXml(const String& str, byte charset, bool escapelf)
{
 if(charset == CHARSET_DEFAULT)
  charset = GetDefaultCharset();
 StringBuffer result;
 const char* s= str.Begin();
 const char* end= str.End();
 for(; s != end; s++)
  /**/ if(*s == '<')  result.Cat("&lt;");
  else if(*s == '>')  result.Cat("&gt;");
  else if(*s == '&')  result.Cat("&amp;");
  else if(*s == '\'') result.Cat("&apos;");
  else if(*s == '\"') result.Cat("&quot;");
  else if((byte)*s < ' ' && (escapelf || *s != '\n' || *s != '\t'))
   result.Cat(NFormat("&#x%02x;", (byte)*s));
  else if(!(*s & 0x80) || charset == CHARSET_UTF8) result.Cat(*s);
```

```
  else result.Cat(ToUtf8(ToUnicode(*s, charset)));
 return result;
}


CONSOLE_APP_MAIN
{
   PNGEncoder pngEncoder;
   String str = pngEncoder.SaveString( CtrlImg::error() );

   XmlNode xmlDoc;
   XmlNode& myDoc = xmlDoc.Add("MY_DOC_TAG");

   XmlNode& myTextTag = myDoc.Add("MyTextTag");
   myTextTag.AddText( str );

   LOG("----------- STR() =");
   LOG(String("length=") << str.GetLength());
   LOG(str);
   LOG("-----------");
   LOG("-----------AsXml() =");
   LOG(String("length=") << AsXML(xmlDoc).GetLength());
   LOG(AsXML(xmlDoc));
   LOG("-----------");
   LOG("-----------DeXml() =");
   LOG(String("length=") << DeXml(str).GetLength());
   LOG(DeXml(str));
   LOG("-----------");
   LOG("-----------DeXml(CHARSET_DEFAULT, true) =");
   LOG(String("length=") << DeXml(str, CHARSET_DEFAULT, true).GetLength());
   LOG(DeXml(str, CHARSET_DEFAULT, true));
   LOG("-----------");
   LOG("-----------MyDeXml() =");
   LOG(String("length=") << MyDeXml(str, CHARSET_DEFAULT, true).GetLength());
   LOG(MyDeXml(str, CHARSET_DEFAULT, true));
}
```

the resulting log file is the following:

```
----------- STR() =
length=419
```

```
----------
----------AsXml() =
length=186
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE MY_DOC_TAG>
<MY_DOC_TAG>

</MY_DOC_TAG>

----------
----------DeXml() =
length=28

----------
----------DeXml(CHARSET_DEFAULT, true) =
length=28

----------
----------MyDeXml() =
length=837
```

&#x19;

@k

Note: there is a 100% growth factor on the data size when converted to xml: this could be optimized using base 64 encoding for example (30% growth factor)

I suggest adding overloads with 'String' parameter for some functions/methods in order to avoid this drawback.