Subject: Re: how to check specific end tag in a xml file?
Posted by Sender Ghost on Thu, 22 Mar 2012 09:45:59 GMT
View Forum Message <> Reply to Message

Hello, Ayana.

ayana wrote on Wed, 21 March 2012 11:57How to check for a specific end tag in the large XML file while parsing it?

ayana wrote on Wed, 21 March 2012 11:57The user enters a name in an edit field of GUI, if the name entered matches with the gathered text of 'name tag' the corresponding person details should get add to XMLTree.

There is XML reference example about methods to parse XML files with XmlParser and XmlNode. As well as AddressBookXML and AddressBookXML2 examples.

I will show you following source code about how to search "name" tag for specified names and changing found "person" data, using XmlNode:
Toggle Spoiler

```
#include <Core/Core.h>

using namespace Upp;

#define TAG_AddressBook "AddressBook"
#define TAG_person "person"
#define TAG_name "name"
#define TAG_surname "surname"
#define TAG_address "address"
#define TAG_email "email"

String PrintData(const XmlNode& node) {
 const XmlNode& ls = node[0];
 StringBuffer text;

 for (int i = 0, n = ls.GetCount(); i < n; ++i) {
  const XmlNode& ll = ls[i];
  if (ll.IsTag(TAG_person)) {
   text << "- Person:\n";
   const XmlNode& lname = ll[TAG_name];
   if (!lname.IsVoid())
    text << TAG_name << ": " << lname[0].GetText() << '\n';

   const XmlNode& lsurname = ll[TAG_surname];
   if (!lsurname.IsVoid())
    text << TAG_surname << ": " << lsurname[0].GetText() << '\n';

   const XmlNode& laddress = ll[TAG_address];
   if (!laddress.IsVoid())
```

```
    text << TAG_address << ": " << laddress[0].GetText() << '\n';

   const XmlNode& lemail = ll[TAG_email];
   if (!lemail.IsVoid())
    text << TAG_email << ": " << lemail[0].GetText() << '\n';

   text << '\n';
  }
 }

 return text;
}

void ChangeXmlText(XmlNode& node, const char *tag, const String& txt) {
 int q = node.FindTag(tag);
 if (q < 0) // Adding new tag in case of not found
  node.Add(tag).AddText(txt);
 else  // Changing found tag
  node.At(q).At(0).CreateText(txt);
}

CONSOLE_APP_MAIN
{
 const Vector<String>& cmdLine = CommandLine();

 if (cmdLine.GetCount() == 0) {
  Cout() << "Specify XML file to load\n";
  return;
 }

 String fileName(NormalizePath(cmdLine[0]));

 // Loading contents of file to String
 const String data = LoadFile(fileName);
 if (data.IsVoid()) {
  Cout() << "Error, while loading '" << NormalizePath(fileName) << "' file\n";
  SetExitCode(1);
  return;
 }

 XmlNode node;
 try { // Trying to parse XML file
  node = ParseXML(data);
 }
 catch (XmlError e) {
  Cout() << "XmlError " << e << '\n';
  SetExitCode(1);
  return;
```

```
}

if (node.GetCount() == 0 || node[0].GetTag() != TAG_AddressBook) {
 Cout() << "There is no following tag for XML file: " << TAG_AddressBook << '\n';
 return;
}

Cout() << "Existing data:\n" << PrintData(node);
// The names of persons for changing data
const String names[] = { "qw", "rer", "new name" };
const String text("*"); // some changed data
// Selecting contents of "AddressBook" tag
XmlNode& ls = node.At(0);

for (int j = 0, m = __countof(names); j < m; ++j) { // Iterate through names array
 bool found = false;
 for (int i = 0, n = ls.GetCount(); i < n; ++i) { // Iterate through "AddressBook" tags
  XmlNode& ll = ls.At(i);
  if (ll.IsTag(TAG_person)) { // Checking for "person" tag
   int q = ll.FindTag(TAG_name);
   if (q < 0)
    continue;
   // Selecting contents of "name" tag
   XmlNode& lname = ll.At(q).At(0);
   if (lname.GetText() == names[j]) { // Comparing names
    found = true;
    // Changing contents of "person" tag
    lname.CreateText(String().Cat() << names[j] << " *");

    ChangeXmlText(ll, TAG_surname, text);
    ChangeXmlText(ll, TAG_address, text);
    ChangeXmlText(ll, TAG_email, text);

    break; // in case of no duplicates
   }
  }
 }

 if (!found) { // Adding new "person" tag with their contents, if name not found
  XmlNode& ll = ls.Add(TAG_person);
  ll.Add(TAG_name).AddText(names[j]);
  ll.Add(TAG_surname).AddText(text);
  ll.Add(TAG_address).AddText(text);
  ll.Add(TAG_email).AddText(text);
 }
}

Cout() << "Changed data:\n" << PrintData(node);
```

```
 fileName = AppendFileName(GetFileDirectory(fileName),
  String().Cat() << GetFileTitle(fileName) << " (changed)" << GetFileExt(fileName));
 // Saving changed XmlNode(s) to different file path
 if (SaveFile(fileName, AsXML(node)))
  Cout() << "Changed file '" << fileName << "' written successfully\n";
 else
  Cout() << "Error, while writing '" << fileName << "' file\n";
}
```

With following XML file:
Toggle Spoiler

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE AddressBook>
<AddressBook>
 <person>
  <name>qw</name>
  <surname>we</surname>
  <address>we</address>
  <email>er</email>
 </person>
 <person>
  <name>rer</name>
  <surname>ty</surname>
  <address>ui</address>
  <email>io</email>
 </person>
 <person>
  <name>oo</name>
  <surname>jlj</surname>
  <address>bh</address>
  <email>cft</email>
 </person>
</AddressBook>
```

You will get following output:
Toggle Spoiler

```
Existing data:
- Person:
name: qw
surname: we
address: we
email: er

- Person:
name: rer
```

surname: ty
address: ui
email: io

- Person:
name: oo
surname: jlj
address: bh
email: cft

Changed data:
- Person:
name: qw *
surname: *
address: *
email: *

- Person:
name: rer *
surname: *
address: *
email: *

- Person:
name: oo
surname: jlj
address: bh
email: cft

- Person:
name: new name
surname: *
address: *
email: *

Changed file 'C:\AddressBook (changed).xml' written successfully

and changed XML file:
Toggle Spoiler

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE AddressBook>
<AddressBook>
 <person>
  <name>qw *</name>
  <surname>*</surname>
  <address>*</address>
  <email>*</email>
```

```
 </person>
 <person>
  <name>rer *</name>
  <surname>*</surname>
  <address>*</address>
  <email>*</email>
 </person>
 <person>
  <name>oo</name>
  <surname>jlj</surname>
  <address>bh</address>
  <email>cft</email>
 </person>
 <person>
  <name>new name</name>
  <surname>*</surname>
  <address>*</address>
  <email>*</email>
 </person>
</AddressBook>
```
