转 TinyXML：一个优秀的C++ XML解析器

2013年10月08日 23:42:50　阅读数：2705

读取和设置xml配置文件是最常用的操作，试用了几个C++的XML解析器，个人感觉TinyXML是使用起来最舒服的，因为它的API接口和Java的十分类似，面向对象性很好。

TinyXML是一个开源的解析XML的解析库，能够用于C++，能够在Windows或Linux中编译。这个解析库的模型通过解析XML文件，然后在内存中生成DOM模型，从而让我们很方便的遍历这棵XML树。

DOM模型即文档对象模型，是将整个文档分成多个元素（如书、章、节、段等），并利用树型结构表示这些元素之间的顺序关系以及嵌套包含关系。

如下是一个XML片段：

```cpp
<Persons>
      <Person ID="1">
          <name>周星星</name>
          <age>20</age>
      </Person>
      <Person ID="2">
          <name>白晶晶</name>
          <age>18</age>
      </Person>
  </Persons>
```

在TinyXML中，根据XML的各种元素来定义了一些类：

TiXmlBase：整个TinyXML模型的基类。

TiXmlAttribute：对应于XML中的元素的属性。

TiXmlNode：对应于DOM结构中的节点。

TiXmlComment：对应于XML中的注释

TiXmlDeclaration：对应于XML中的申明部分，即<？versiong="1.0" ?>。

TiXmlDocument：对应于XML的整个文档。

TiXmlElement：对应于XML的元素。

TiXmlText：对应于XML的文字部分

TiXmlUnknown：对应于XML的未知部分。

TiXmlHandler：定义了针对XML的一些操作。

TinyXML是个解析库，主要由DOM模型类（TiXmlBase、TiXmlNode、TiXmlAttribute、TiXmlComment、TiXmlDeclaration、TiXmlElement、TiXmlText、TiXmlUnknown）和操作类（TiXmlHandler）构成。它由两个头文件（.h文件）和四个CPP文件（.cpp文件）构成，用的时候，只要将（tinyxml.h、tinystr.h、tinystr.cpp、tinyxml.cpp、tinyxmlerror.cpp、tinyxmlparser.cpp）导入工程就可以用它的东西了。如果需要，可以将它做成自己的DLL来调用。举个例子就可以说明一切。。。

对应的XML文件：

```cpp
<Persons>
    <Person ID="1">
        <name>phinecos</name>
        <age>22</age>
    </Person>
</Persons>
```

读写XML文件的程序代码：

```cpp
#include <iostream>
#include "tinyxml.h"
#include "tinystr.h"
#include <string>
#include <windows.h>
#include <atlstr.h>
using namespace std;
```

```
8.
9.  CString GetAppPath()
10. {//获取应用程序根目录
11.     TCHAR modulePath[MAX_PATH];
12.     GetModuleFileName(NULL, modulePath, MAX_PATH);
13.     CString strModulePath(modulePath);
14.     strModulePath = strModulePath.Left(strModulePath.ReverseFind(_T('\\')));
15.     return strModulePath;
16. }
17.
18. bool CreateXmlFile(string& szFileName)
19. {//创建xml文件,szFilePath为文件保存的路径,若创建成功返回true,否则false
20.     try
21.     {
22.         //创建一个XML的文档对象。
23.         TiXmlDocument *myDocument = new TiXmlDocument();
24.         //创建一个根元素并连接。
25.         TiXmlElement *RootElement = new TiXmlElement("Persons");
26.         myDocument->LinkEndChild(RootElement);
27.         //创建一个Person元素并连接。
28.         TiXmlElement *PersonElement = new TiXmlElement("Person");
29.         RootElement->LinkEndChild(PersonElement);
30.         //设置Person元素的属性。
31.         PersonElement->SetAttribute("ID", "1");
32.         //创建name元素、age元素并连接。
33.         TiXmlElement *NameElement = new TiXmlElement("name");
34.         TiXmlElement *AgeElement = new TiXmlElement("age");
35.         PersonElement->LinkEndChild(NameElement);
36.         PersonElement->LinkEndChild(AgeElement);
37.         //设置name元素和age元素的内容并连接。
38.         TiXmlText *NameContent = new TiXmlText("周星星");
39.         TiXmlText *AgeContent = new TiXmlText("22");
40.         NameElement->LinkEndChild(NameContent);
41.         AgeElement->LinkEndChild(AgeContent);
42.         CString appPath = GetAppPath();
43.         string seperator = "\\";
44.         string fullPath = appPath.GetBuffer(0) +seperator+szFileName;
45.         myDocument->SaveFile(fullPath.c_str());//保存到文件
46.     }
47.     catch (string& e)
48.     {
49.         return false;
50.     }
51.     return true;
52. }
53.
54. bool ReadXmlFile(string& szFileName)
55. {//读取Xml文件，并遍历
56.     try
57.     {
58.         CString appPath = GetAppPath();
59.         string seperator = "\\";
60.         string fullPath = appPath.GetBuffer(0) +seperator+szFileName;
61.         //创建一个XML的文档对象。
62.         TiXmlDocument *myDocument = new TiXmlDocument(fullPath.c_str());
63.         myDocument->LoadFile();
64.         //获得根元素，即Persons。
65.         TiXmlElement *RootElement = myDocument->RootElement();
66.         //输出根元素名称，即输出Persons。
67.         cout << RootElement->Value() << endl;
68.         //获得第一个Person节点。
69.         TiXmlElement *FirstPerson = RootElement->FirstChildElement();
70.         //获得第一个Person的name节点和age节点和ID属性。
71.         TiXmlElement *NameElement = FirstPerson->FirstChildElement();
72.         TiXmlElement *AgeElement = NameElement->NextSiblingElement();
73.         TiXmlAttribute *IDAttribute = FirstPerson->FirstAttribute();
74.         //输出第一个Person的name内容，即周星星；age内容，即；ID属性，即。
75.         cout << NameElement->FirstChild()->Value() << endl;
76.         cout << AgeElement->FirstChild()->Value() << endl;
77.         cout << IDAttribute->Value()<< endl;
78.     }
79.     catch (string& e)
80.     {
81.         return false;
82.     }
83.     return true;
84. }
85. int main()
86. {
87.     string fileName = "info.xml";
88.     CreateXmlFile(fileName);
89.     ReadXmlFile(fileName);
90. }
```

原文地址： http://www.cnblogs.com/phinecos/archive/2008/03/11/1100912.html

TinyXML下载地址： http://download.csdn.net/detail/leixiaohua1020/6372061

文章标签： tinyxml xml 解析 c++ 库

个人分类： 纯编程