Xml\_homework\_DTD

1. Xml文档

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE Library SYSTEM "library.dtd">

<Library>

<Book BookId="1">

<Title>XML</Title>

<Author>chimy</Author>

<Price>100</Price>

<Publisher>chimy</Publisher>

<Cost>20</Cost>

</Book>

<Member>

<Name>

<FirstName>a</FirstName>

<MiddleName>b</MiddleName>

<LastName>c</LastName>

</Name>

<Address>

<HouseNumber>709</HouseNumber>

<Street>chengnan</Street>

<City>jingzhou</City>

</Address>

</Member>

</Library>

1. DTD文档

<!ELEMENT Library (Book,Member)>

<!ATTLIST Book BookId CDATA #REQUIRED>

<!ELEMENT Book (Title,Author,Price,Publisher,Cost)>

<!ELEMENT Member (Name,Address)>

<!ELEMENT Name (FirstName,MiddleName,LastName)>

<!ELEMENT Address (HouseNumber,Street,City)>

DTD也可以在xml文件中进行声明

<!DOCTYPE tagname[内容]>在xml中设置dtd

<!DOCTYPE file SYSTEM "fileurl">//引用本地文件

<!DOCTYPE file PUBLIC "name" "fileurl">//引用外部公共文件

3.DTD文档元素定义

语法格式：<!ELEMENT 元素名称 元素类型>

其中元素类型可以是元素内容，或类型。如为元素内容，则需要使用（），如

<!ELEMENT name (name1,name2)>

<!ELEMENT name(#PCDATA)>

如为元素类型，则直接书写，例：EMPTY定义空元素，ANY表示元素内容为任意类型

元素内容可以使用逗号和|分隔，逗号表示内容的出现顺序必须与声明时一致，|表示多个只能出现一个

元素内容中使用+、?、\*等符号表示元素出现次数

+：一次或多次

?：0次或一次

\*：0次或多次

可以用括号批量设置。

例：<!ELEMENT MYFILE((TITLE\*,AUTHOR?,EMAIL)\*|COMMENT)>

属性定义attribute

在 DTD 中，属性通过 ATTLIST 声明来进行声明。

声明属性

属性声明使用下列语法：

<!ATTLIST 元素名称 属性名称 属性类型 默认值>

DTD 实例:

<!ATTLIST payment type CDATA "check">

XML 实例:

<payment type="check" />

以下是属性类型的选项：

类型 描述

CDATA 值为字符数据 (character data)

(en1|en2|..) 此值是枚举列表中的一个值

ID 值为唯一的 id

IDREF 值为另外一个元素的 id

IDREFS 值为其他 id 的列表

NMTOKEN 值为合法的 XML 名称

NMTOKENS 值为合法的 XML 名称的列表

ENTITY 值是一个实体

ENTITIES 值是一个实体列表

NOTATION 此值是符号的名称

xml: 值是一个预定义的 XML 值

默认值参数可使用下列值：

值 解释

值 属性的默认值

#REQUIRED 属性值是必需的

#IMPLIED 属性不是必需的

#FIXED value 属性值是固定的

规定一个默认的属性值

DTD:

<!ELEMENT square EMPTY>

<!ATTLIST square width CDATA "0">

合法的 XML:

<square width="100" />

在上面的例子中，"square" 被定义为带有 CDATA 类型的 "width" 属性的空元素。如果宽度没有被设定，其默认值为0 。

#IMPLIED

语法

<!ATTLIST 元素名称 属性名称 属性类型 #IMPLIED>

例子

DTD:

<!ATTLIST contact fax CDATA #IMPLIED>

合法的 XML:

<contact fax="555-667788" />

合法的 XML:

<contact />

假如您不希望强制作者包含属性，并且您没有默认值选项的话，请使用关键词 #IMPLIED。

#REQUIRED

语法

<!ATTLIST 元素名称 属性名称 属性类型 #REQUIRED>

例子

DTD:

<!ATTLIST person number CDATA #REQUIRED>

合法的 XML:

<person number="5677" />

非法的 XML:

<person />

假如您没有默认值选项，但是仍然希望强制作者提交属性的话，请使用关键词 #REQUIRED。

#FIXED

语法

<!ATTLIST 元素名称 属性名称 属性类型 #FIXED "value">

例子

DTD:

<!ATTLIST sender company CDATA #FIXED "Microsoft">

合法的 XML:

<sender company="Microsoft" />

非法的 XML:

<sender company="W3School" />

如果您希望属性拥有固定的值，并不允许作者改变这个值，请使用 #FIXED 关键词。如果作者使用了不同的值，XML 解析器会返回错误。

列举属性值

语法：

<!ATTLIST 元素名称 属性名称 (en1|en2|..) 默认值>

DTD 例子:

<!ATTLIST payment type (check|cash) "cash">

XML 例子:

<payment type="check" />

或者

<payment type="cash" />

4.通过tinyxml解析

#include <iostream>

#include "tinystr.h"

#include "tinyxml.h"

#include <string>

using namespace std;

void readSchoolXml() {

const char \* xmlFile = "library.xml";

TiXmlDocument doc;

if (doc.LoadFile(xmlFile)) {//打开library.xml文件

doc.Print();

}

else {

cout << "can not parse library.xml" << endl;//打开失败弹出提醒

return;

}

TiXmlElement\* libraryElement = doc.RootElement(); //library元素 也就是根元素

TiXmlElement\* bookElement = libraryElement->FirstChildElement(); // book元素 根元素的第一个儿子元素

TiXmlAttribute\* bookid = bookElement->FirstAttribute(); //获得book的id属性

cout << bookid->Name() << " : " << bookid->Value() << std::endl; //显示book的属性

TiXmlElement\* TietleElement = bookElement->FirstChildElement();//获得TietleElement title是book的第一个儿子

for (; TietleElement != NULL; TietleElement = TietleElement->NextSiblingElement()) { //当title还有兄弟节点，循环一直进行

string contactType = TietleElement->Value();

string contactValue = TietleElement->GetText();

cout << contactType << " : " << contactValue << std::endl;//显示标签和值

}

TiXmlElement\* memberElement = bookElement->NextSiblingElement(); // member元素 是根节点的第二个儿子，也就是book节点的兄弟

TiXmlElement\* nameElement = memberElement->FirstChildElement();//name是member的第一个儿子

TiXmlElement\* FirstnameElement = nameElement->FirstChildElement();//Firstname又是name的第一个儿子

for (; FirstnameElement != NULL; FirstnameElement = FirstnameElement->NextSiblingElement()) { //当Firstname还有兄弟节点，循环一直进行

string contactType = FirstnameElement->Value();

string contactValue = FirstnameElement->GetText();

cout << contactType << " : " << contactValue << std::endl;//显示标签和值

}

TiXmlElement\* addressElement = nameElement->NextSiblingElement();//address元素 是member的第二个儿子，也就是name节点的兄弟

TiXmlElement\* HouseNumberElement = addressElement->FirstChildElement();//Housenumber是address的第一个儿子

for(;HouseNumberElement!=NULL;HouseNumberElementHouseNumberElement->NextSiblingElement()) {//当Housenumber还有兄弟节点，循环一直进行

string contactType = HouseNumberElement->Value();

string contactValue = HouseNumberElement->GetText();

cout << contactType << " : " << contactValue << std::endl;//显示标签和值

}

}

int main() {

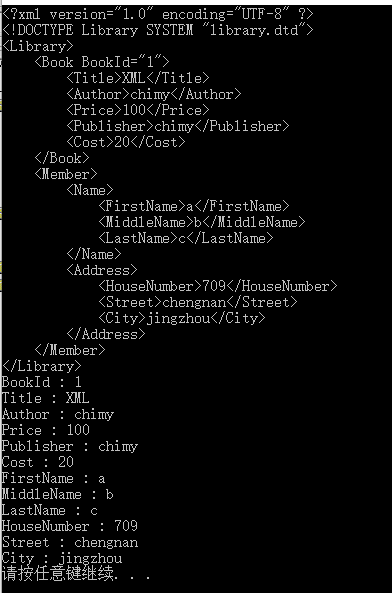
readSchoolXml();

system("pause");

return 1;

}

5.解析结果截图



6.心得体会

在用xml解析时出现错误

C++错误unresolved external symbol \_WinMain@16 10

Linking... LIBCD.lib(wincrt0.obj) : error LNK2001: unresolved external symbol \_WinMain@16 Debug/test.exe : fatal error LNK1120: 1 unresolved externals ...

这个文件中使用了WinMian而不是main作为入口函数。vc这时的默认设置是针对控制台程序的。

解决方法：

1.进入project->setting->c/c++, 在category(第一行)中选择preprocessor,在processor definitions中删除\_WINDOWS, 添加\_CONSOLE

2.进入project->setting->Link, 在Project options中将/subsystem:windows.改为/subsystem:console

3.保存设置，Rebuild All.

error LNK1117: syntax error in option "subsystem:console/incremental:yes"

在project ->setting->link中的subsystem:console和incremental:yes之间加一个空格就可以了