

Rapid XML

개요

Rapid XML은, 매우 빠르고 효율적인 XML파서 입니다.

같은 DOM형식의 TinyXML과는 다르게 별도의 Memory pool을 운영하기 때문에 속도가 빠르며, 이미 몇몇 HTC 모바일 핸드폰에서는 사용하고 있다고 합니다.

설치 방법

RapidXML의 사이트(<http://rapidxml.sourceforge.net/>)에 들어가서, 필요한 헤더파일만 다운받으신 후, 프로젝트에 추가하시면 됩니다. 총 4개의 헤더파일이 있습니다.

간단한 예제 - 1

XML파일을 읽어오는 간단한 예제입니다.

파일의 규격에 따라, 아래 내용이 틀려지겠지만 기본적인 내용은 아래와 같습니다.

```
void LoadXML(const char *strFile)
{
    std::basic_ifstream<char> xmlFile(strFile);
    xmlFile.seekg(0, std::ios::end);

    unsigned int size = xmlFile.tellg();
    xmlFile.seekg(0);

    std::vector<char> xmlData(size+1);
    xmlData[size] = 0;
    xmlFile.read(&xmlData.front(), (std::streamsize)size);

    rapidxml::xml_document<char> xmlDoc;
    xmlDoc.parse<0>(&xmlData.front());

    char *name;
    char *value;

    rapidxml::xml_node<char> *item;
    rapidxml::xml_node<char> *subItem;

    rapidxml::xml_node<char> *root = xmlDoc.first_node();

    for( item = root->first_node(); item; item = item->next_sibling() )
    {
        name = item->name();

        for( subItem = item->first_node(); subItem; subItem = subItem->next_sibling() )
        {
            name = subItem->name(); //key
            value = subItem->value(); //value
        }
    }
}
```

```
xmlFile.close();  
}
```

간단한 예제 - 2

이 예제는, xml파일을 작성하는 예제입니다.
파일의 규격에 따라 아래 내용이 틀려지겠지만, 기본적인 내용은 아래와 같습니다.

```
void WriteXML(const char *strFile)  
{  
    rapidxml::xml_document<> doc;  
  
    //xml declaration  
    rapidxml::xml_node<> *decl = doc.allocate_node(node_declaration);  
    decl->append_attribute(doc.allocate_attribute("version", "1.0"));  
    decl->append_attribute(doc.allocate_attribute("encoding", "utf-8"));  
    doc.append_node(decl);  
  
    //root node  
    xml_node<>* root = doc.allocate_node(node_element, "rootnode");  
    root->append_attribute(doc.allocate_attribute("version", "1.0"));  
    root->append_attribute(doc.allocate_attribute("type", "example"));  
    doc.append_node(root);  
  
    //write contents  
  
    //child node  
    rapidxml::xml_node<> *child = doc.allocate_node(node_element, "childnode");  
    root->append_node(child);  
  
    //--  
  
    //change data type  
    std::string xml_as_string;  
    printf(std::back_inserter(xml_as_string), doc);  
  
    //write xml file  
    std::ofstream fileOut;  
    fileOut.open(strFile);  
    fileOut<<xml_as_string;  
    fileOut.clear();  
}
```