TinyXml In Practice

Keep It Small and Simple

Enable Using STL Mode

Get Element

Query Attribute Of Element

inviassassin@gmail.com

Enable Using STL Mode

TinyXml can be compiled to use or not use STL.

Using STL mode support std::string, std::istream, std::ostream, operator << and operator >>.

In Non-STL mode, no STL files are included. All the string classes are implemented by TinyXml itself.

Enable Using STL Mode

Set TINYXML_USE_STL in tinyxml/Makefile to YES

tinyxml/Makefile
TINYXML_USE_STL := YES

Define TIXML_USE_STL as the first line of tinyxml/tinyxml.h

tinyxml/tinyxml.h
#define TIXML_USE_STL

Traversing Through All Elements Of Document

ninja.xml

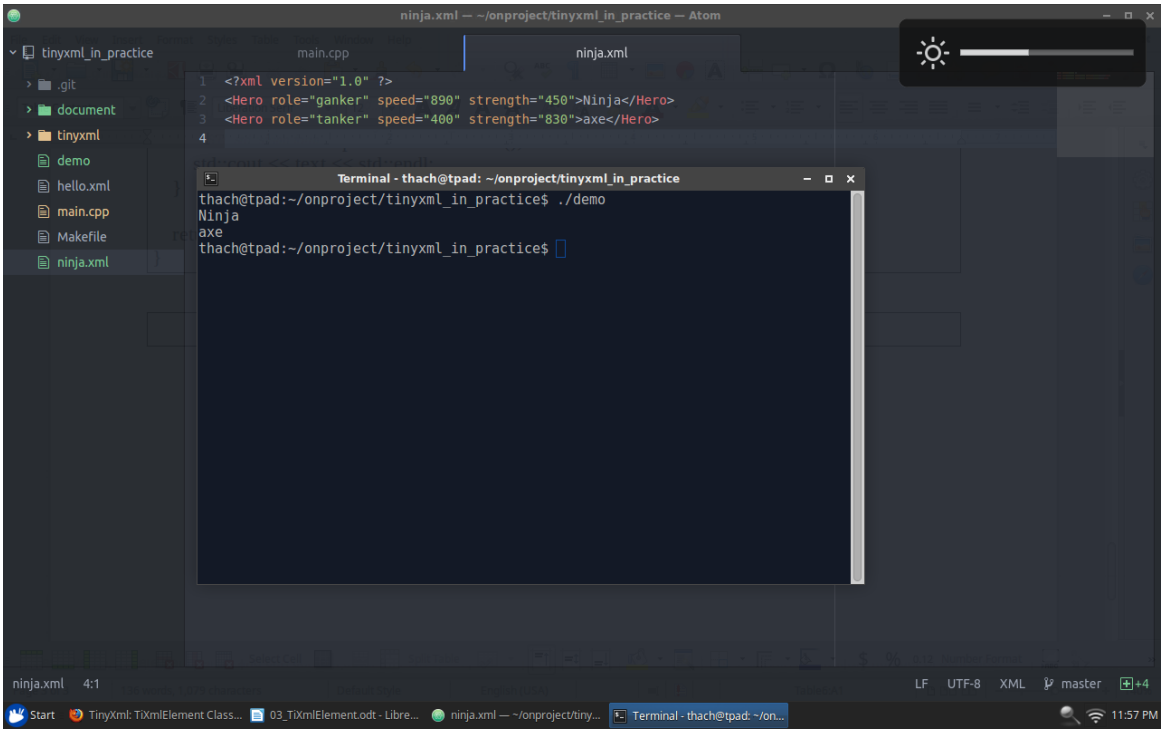
```
<?xml version="1.0" ?>
<Hero role="ganker" speed="890" strength="450">Ninja</Hero>
<Hero role="tanker" speed="400" strength="830">axe</Hero>
```

RootElement(): get the root element - the only top level element of the document. In that case, the root element is <Hero role="ganker" speed="890" strength="450">Ninja</Hero>.

NextSiblingElement(): get the next element. In this example, the next element of <Hero role="ganker" speed="890" strength="450">Ninja</Hero> is <Hero role="tanker" speed="400" strength="830">axe</Hero>.

main.cpp

```
#include "tinyxml.h"

int main()
{
    TiXmlDocument doc;
    doc.LoadFile("ninja.xml");

    for(TiXmlElement *pElement = doc.RootElement();
            pElement != 0;
            pElement = pElement->NextSiblingElement())
    {
        const char* text = pElement->GetText();
        std::cout << text << std::endl;
    }

    return 0;
}
```



Get Element By Value

ninja.xml

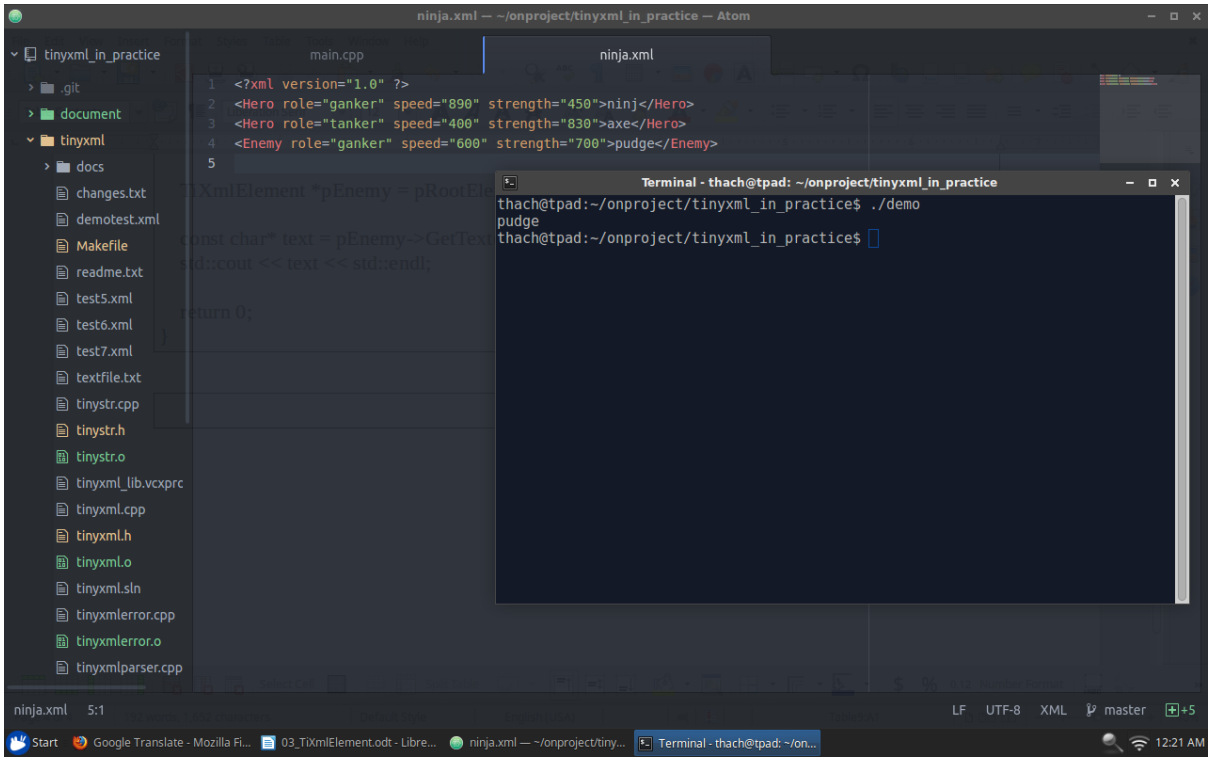
```
<?xml version="1.0" ?>
<Hero role="ganker" speed="890" strength="450">ninj</Hero>
<Hero role="tanker" speed="400" strength="830">axe</Hero>
<Enemy role="ganker" speed="600" strength="700">pudge</Enemy>
```

Find Element Had Value: "**Enemy**".

NextSiblingElement("Enemy"): find the next element which has value is "Enemy".

main.cpp

```
#include "tinyxml.h"

int main()
{
    TiXmlDocument doc;
    doc.LoadFile("ninja.xml");

    TiXmlElement *pRootElement = doc.RootElement();

    TiXmlElement *pEnemy = pRootElement->NextSiblingElement("Enemy");

    const char* text = pEnemy->GetText();
    std::cout << text << std::endl;

    return 0;
}
```



Query Attribute Of Element

ninja.xml

```
<?xml version="1.0" ?>
<Hero role="ganker" speed="890" strength="450">ninja</Hero>
<Hero role="tanker" speed="400" strength="830">axe</Hero>
<Enemy role="ganker" speed="600" strength="700">pudge</Enemy>
```

Query***Attribute("role", &role): find the attribute has name "role" and assign the value of the attribute to role variable.

main.cpp

```
#include "tinyxml.h"

int main()
{
    TiXmlDocument doc;
    doc.LoadFile("ninja.xml");

    TiXmlElement *pElement = doc.RootElement();

    const char* text = pElement->GetText();

    std::string role;
    pElement->QueryStringAttribute("role", &role);
```

```
    int speed = 0;
    pElement->QueryIntAttribute("speed", &speed);

    int strength = 0;
    pElement->QueryIntAttribute("strength", &strength);

    std::cout << text << " : " << role
            << speed << " : " << strength << std::endl;

    return 0;
}
```