
Subject: Simple XML keymap parser

Posted by [Sc0rch](#) on Thu, 21 Jan 2010 18:07:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

Very simple, but, maybe, it can be useful for someone:

Header:

```
#ifndef XML_CONFIG_H
#define XML_CONFIG_H

#include <Core/Core.h>
using namespace Upp;

struct XMLConfig
{
    XMLConfig(const String& name = "", const String& lang = "EN-EN")
        : Name(name), Language(::LNGFromText(lang)) {}

    inline void SetString(const String& param, const String& value);
    inline void SetNumber(const String& param, const int& value);
    inline void SetLanguage(const String& lang);
    inline void Xmlize(XmlIO xml);
    inline void Dump();
    inline bool Load(const String& path = "Default");
    inline bool Save(const String& path = "Default");

    inline String GetString(const String& param);
    inline int GetNumber(const String& param, int defaultNum = 0);

    String Name;
    String Path;
    int Language;
    VectorMap<String, String> Map;
};

inline void XMLConfig::SetLanguage(const String& lang)
{
    Language = ::LNGFromText(lang);
}

inline bool XMLConfig::Load(const String& path)
{
    Path = path;
    return path == "Default" ? LoadFromXMLFile(*this) : LoadFromXMLFile(*this, path);
}

inline bool XMLConfig::Save(const String& path)
```

```

{
    Path = path;
    return path == "Default" ? StoreAsXMLFile(*this) : StoreAsXMLFile(*this, "Data", path);
}

inline void XMLConfig::SetString(const String& param, const String& value)
{
    if (Map.Find(param) >= 0) Map.Get(param) = value; else Map.Add(param, value);
}

inline void XMLConfig::SetNumber(const String& param, const int& value)
{
    SetString(param, AsString(value));
}

inline String XMLConfig::GetString(const String& param)
{
    return (Map.Find(param) >= 0) ? Map.Get(param) : "";
}

inline int XMLConfig::GetNumber(const String& param, int defaultNum)
{
    String value = GetString(param);
    return value == "" ? defaultNum : ScanInt(value);
}

inline void XMLConfig::Xmlize(XmlIO xml)
{
    xml("name", Name)("map", Map);
    XmlizeLang(xml, "language", Language);
}

inline void XMLConfig::Dump()
{
    DUMP(Name);
    DUMP(Path);
    for (int i = 0; i < Map.GetCount(); i++) LOG(Map.GetKey(i) << " ... " << Map[i]);
}

#endif // .. XML_CONFIG_HPP

```

Source:

```
#include "XMLConfig.h"
```

```
CONSOLE_APP_MAIN
```

```
{
if (!FileExists(ConfigFile("XMLConfig.xml")))
{
XMLConfig x("Test configuration!", "RU-RU");
x.SetString("K01", "V01");
x.SetString("K02", "V02");
x.Save();
}

XMLConfig y;
y.Load();
y.Dump();
}
```

Subject: Re: Simple XML keymap parser
Posted by [Sc0rch](#) on Sat, 23 Jan 2010 04:35:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

Another variant:

```
#ifndef XML_CONFIG_HPP
#define XML_CONFIG_HPP

#include <Core/Core.h>
using namespace Upp;

template<class K, class V>
class XMLConfig : public VectorMap<K, V>
{
public:
XMLConfig(const String& name = "", const String& lang = "EN-EN")
: Name(name), Language(::LNGFromText(lang)) {}

XMLConfig& Set(const K& param, const V& value);
XMLConfig& SetLanguage(const String& lang);
XMLConfig& Dump();

void Xmlize(XmlIO xml);
bool Load(const String& path = "");
bool Save(const String& path = "");

String Name;
String Path;
int Language;
};
```

```

template<class K, class V>
inline XMLConfig<K, V>& XMLConfig<K, V>::SetLanguage(const String& lang)
{
    Language = ::LNGFromText(lang);
    return *this;
}

template<class K, class V>
inline bool XMLConfig<K, V>::Load(const String& path)
{
    Path = path;
    return path == "" ? LoadFromXMLFile(*this) : LoadFromXMLFile(*this, path);
}

template<class K, class V>
inline bool XMLConfig<K, V>::Save(const String& path)
{
    Path = path;
    return path == "" ? StoreAsXMLFile(*this, "data") : StoreAsXMLFile(*this, "data", path);
}

template<class K, class V>
inline XMLConfig<K, V>& XMLConfig<K, V>::Set(const K& param, const V& value)
{
    if (VectorMap<K, V>::Find(param) >= 0) Get(param) = value; else Add(param, value);
    return *this;
}

template<class K, class V>
inline void XMLConfig<K, V>::Xmlize(XmlIO xml)
{
    xml("name", Name)("map", *((VectorMap<K, V>*)this));
    XmlizeLang(xml, "language", Language);
}

template<class K, class V>
inline XMLConfig<K, V>& XMLConfig<K, V>::Dump()
{
    DUMP(Name);
    DUMP(Path);
    for (int i = 0; i < VectorMap<K, V>::GetCount(); i++)
        LOG(AsString(VectorMap<K, V>::GetKey(i)) + " = " +
            AsString(VectorMap<K, V>::Get(VectorMap<K, V>::GetKey(i))));
    return *this;
}

#endif // .. XML_CONFIG_HPP

```

Example:

```
#include "XMLConfig.hpp"

CONSOLE_APP_MAIN
{
if (!FileExists(ConfigFile("XMLConfig.xml")))
{
XMLConfig<String, String> x("Test configuration!", "RU-RU");
x.Set("K01", "V01");
x.Set("K02", "V02");
x.Save();
}
}
```

Subject: Re: Simple XML keymap parser
Posted by [kasome](#) on Sat, 23 Jan 2010 17:03:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

It is useful to me.
Thanks for your sharing, Sc0rch

BTW, one member function of first version of the Simple XML keymap parser,

```
inline bool XMLConfig::Load(const String& path){
    if (!FileExists(path))
        return false;

    Path = path;
    return path == "Default" ? LoadFromXMLFile(*this) : LoadFromXMLFile(*this, path);
}
```

should be

```
inline bool XMLConfig::Load(const String& path){
    if ( path != "Default" && !FileExists(path) )
        return false;

    Path = path;
    return path == "Default" ? LoadFromXMLFile(*this) : LoadFromXMLFile(*this, path);
}
```

right?

Subject: Re: Simple XML keymap parser
Posted by [ScOrch](#) on Sun, 24 Jan 2010 10:25:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
kasome wrote on Sat, 23 January 2010 18:03
inline bool XMLConfig::Load(const String& path){
    if ( path != "Default" && !FileExists(path) )
        return false;

    Path = path;
    return path == "Default" ? LoadFromXMLFile(*this) : LoadFromXMLFile(*this, path);
}
```

right?

You can remove "FileExists" at all. "LoadFromXMLFile" returns false if file not exists. My mistake, thank you!

Subject: Re: Simple XML keymap parser
Posted by [ScOrch](#) on Mon, 25 Jan 2010 10:02:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Final code (for me =)):

```
#ifndef XML_CONFIG_HPP
#define XML_CONFIG_HPP

#include <Core/Core.h>
using namespace Upp;

class XMLConfig : public VectorMap<String, String>
{
public:
    XMLConfig(const String& name = "", const String& lang = "EN-EN")
        : Name(name), Language(::LNGFromText(lang)) {}

    XMLConfig& Set(const String& param, const String& value);
    XMLConfig& Set(const String& param, int value);
    XMLConfig& SetLanguage(const String& lang);
    XMLConfig& Dump();

    int GetNumber(const String& param, int min = -32768, int max = 32767);

    void Xmlize(XmlIO xml);
    bool Load(const String& path = "");
    bool Save(const String& path = "");
};
```

```

String Name;
String Path;
int Language;
};

inline XMLConfig& XMLConfig::SetLanguage(const String& lang)
{
    Language = ::LNGFromText(lang);
    return *this;
}

inline bool XMLConfig::Load(const String& path)
{
    Path = path;
    return path == "" ? LoadFromXMLFile(*this) : LoadFromXMLFile(*this, path);
}

inline bool XMLConfig::Save(const String& path)
{
    Path = path;
    return path == "" ? StoreAsXMLFile(*this, "data") : StoreAsXMLFile(*this, "data", path);
}

inline XMLConfig& XMLConfig::Set(const String& param, const String& value)
{
    if (VectorMap<String, String>::Find(param) >= 0)
        VectorMap<String, String>::Get(param) = value;
    else
        VectorMap<String, String>::Add(param, value);
    return *this;
}

inline XMLConfig& XMLConfig::Set(const String& param, int value)
{
    return Set(param, AsString(value));
}

inline int XMLConfig::GetNumber(const String& param, int min, int max)
{
    int r = ScanInt(Get(param));
    if (r < min) r = min;
    if (r > max) r = max;
    return r;
}

inline void XMLConfig::Xmlize(XmlIO xml)
{

```

```
xml("name", Name)("map", *((VectorMap<String, String>*)this));  
XmlizeLang(xml, "language", Language);  
}
```

```
inline XMLConfig& XMLConfig::Dump()  
{  
    DUMP(Name);  
    DUMP(Path);  
    for (int i = 0; i < VectorMap<String, String>::GetCount(); i++)  
        LOG(GetKey(i) + " = " + Get(GetKey(i)));  
    return *this;  
}
```

```
#endif // .. XML_CONFIG_HPP
```

Best regards!
Anton

--

Subject: Re: Simple XML keymap parser
Posted by [kasome](#) on Mon, 25 Jan 2010 16:38:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

It's good.
Thanks a lot, Sc0rch
