

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();
    }
}

```
