
Subject: Strange issue with VectorMap<String, String>

Posted by [Sc0rch](#) on Mon, 12 Apr 2010 08:36:06 GMT

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

Very strange, I don't know why app crashes. Debugger stops execution at String::IsLarge() function. Compiled with MinGW.Debug. Sorry, if this is my fault.

XMLConfig.h

```
#ifndef XML_CONFIG_H
#define XML_CONFIG_H
```

```
#include <Core/Core.h>
```

```
NAMESPACE_UPP
```

```
class XMLConfig
```

```
{
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, bool value);
XMLConfig& Set(const String& param, int value);
XMLConfig& SetLanguage(const String& lang);
XMLConfig& DumpXML();
String Get(const String& param) const;
String Get(const String& param, const String& init);
bool GetBool(const String& param, bool init = false);
int GetNumber(const String& param, int init = 0, int min = INT_MIN, int max = INT_MAX);
```

```
void Remove(const String& param);
void Clear();
```

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

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

```
END_UPP_NAMESPACE
```

```
#endif // .. XML_CONFIG_H
```

XMLConfig.cpp

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

```
NAMESPACE_UPP
```

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

```
XMLConfig& XMLConfig::Set(const String& param, const String& value)
{
    Map.GetAdd(param) = value;
    return *this;
}
```

```
XMLConfig& XMLConfig::Set(const String& param, bool value)
{
    return Set(param, value ? "true" : "false");
}
```

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

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

```
String XMLConfig::Get(const String& param, const String& init)
{
    return Map.Find(param) >= 0 ? Map.Get(param) : Map.GetAdd(param) = init;
}
```

```
bool XMLConfig::GetBool(const String& param, bool init)
{
    String r = Get(param, init ? "true" : "false");
    return (r == "true" || r == "1") ? true : ((r == "false" || r == "0") ? false : init);
}
```

```
int XMLConfig::GetNumber(const String& param, int init, int min, int max)
{

```

```

int r = ScanInt( Get(param, AsString(init)) );
if (r < min) { r = min; Set(param, AsString(init)); }
if (r > max) { r = max; Set(param, AsString(init)); }
return r;
}

void XMLConfig::Remove(const String& param)
{
    int i = Map.Find(param);
    if (i < 0) return;
    Map.Remove(i);
}

void XMLConfig::Clear()
{
    Language= LNGFromText("EN-EN");
    Name.Clear();
    Path.Clear();
    Map.Clear();
}

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

    XmlIO prop(xml.GetAdd("properties"));
    String param, value;

    if (xml.IsStoring())
        for(int i = 0; i < Map.GetCount(); i++)
        {
            param = Map.GetKey(i);
            value = Map[i];
            prop.Add("property").Attr("name", param).Attr("value", value);
        }
    else
        for (int i = 0; i < prop->GetCount(); ++i)
        {
            const XmlNode* n = &prop->Node(i);
            Set(n->Attr("name"), n->Attr("value"));
        }
}

XMLConfig& XMLConfig::DumpXML()
{
    DUMP(Name);
    DUMP(Path);
}

```

```

for (int i = 0; i < Map.GetCount(); i++)
    LOG(Map.GetKey(i) + " = " + Map[i]);
return *this;
}

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

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

END_UPP_NAMESPACE

```

XMLConfigTest.cpp

```

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

```

CONSOLE_APP_MAIN

```

{
    XMLConfig x("Test!", "EN-EN");
    x.Set("K01", "V01");
    x.Save();
}

```

File Attachments

- 1) [XMLConfig.rar](#), downloaded 157 times
- 2) [XMLConfigTest.rar](#), downloaded 381 times

Subject: Re: Strange issue with VectorMap<String, String>

Posted by [Sc0rch](#) on Mon, 12 Apr 2010 11:58:29 GMT

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

Well, when I've removed this funcs:

```

XMLConfig& Set(const String& param, bool value);
XMLConfig& Set(const String& param, int value);

```

all is working now.

Why Set-func can't be overloaded?

Best regards and sorry for my English,
Anton

Subject: Re: Strange issue with VectorMap<String, String>

Posted by [mirek](#) on Fri, 16 Apr 2010 09:57:37 GMT

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

Sc0rch wrote on Mon, 12 April 2010 07:58Well, when I've removed this funcs:

```
XMLConfig& Set(const String& param, bool value);  
XMLConfig& Set(const String& param, int value);
```

all is working now.

Why Set-func can't be overloaded?

Best regards and sorry for my English,
Anton

They can. The problem is here:

```
XMLConfig& XMLConfig::Set(const String& param, bool value)  
{  
    return Set(param, value ? "true" : "false");  
}
```

Preferred conversion of const char * is to bool - it is straight, while -> String it requires constructor pass. So you have infinite recursion here.

Fix: Add

```
XMLConfig& XMLConfig::Set(const String& param, const char *value)  
{  
    Map.GetAdd(param) = value;  
    return *this;  
}
```

Subject: Re: Strange issue with VectorMap<String, String>

Posted by [Sc0rch](#) on Fri, 16 Apr 2010 10:15:20 GMT

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

luzr wrote on Fri, 16 April 2010 16:57

Preferred conversion of const char * is to bool - it is straight, while -> String it requires constructor pass. So you have infinite recursion here.

Fix: Add

```
XMLConfig& XMLConfig::Set(const String& param, const char *value)
```

```
{  
    Map.GetAdd(param) = value;  
    return *this;  
}
```

My mistake. All works now. Thank you for support!

P.S. The best toolkit.
