
Subject: Re: Xmlize works only for storing
Posted by [mirek](#) on Fri, 03 Oct 2008 09:54:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mindtraveller wrote on Fri, 03 October 2008 03:13l see no critical problem with this. You may write your own class derived from `std::list<...>` with `Xmlize` function.
Besides I do not think it is good idea to mix STL and NTL libraries in code.

Well, I certainly would not recommend using STL but sometimes you perhaps need to deal with existing code...

You can define `Xmlize` as template function specialisation and that solves the problem of "external" types:

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

using namespace Upp;
using namespace std;

template<> void Upp::Xmlize(XmlIO xml, vector<int>& data) {
    if(xml.IsStoring())
        for(int i = 0; i < (int)data.size(); i++)
            Xmlize(xml.Add("item"), data[i]);
    else {
        data.clear();
        for(int i = 0; i < xml->GetCount(); i++)
            if(xml->Node(i).IsTag("item")) {
                data.push_back(0);
                Xmlize(xml.At(i), data.back());
            }
    }
}
```

```
CONSOLE_APP_MAIN
{
    vector<int> x;
    x.push_back(1);
    x.push_back(2);
    x.push_back(3);
    String s = StoreAsXML(x, "std-test");
    DUMP(s);
    vector<int> y;
    LoadFromXML(y, s);
    for(int i = 0; i < (int)y.size(); i++)
        DUMP(y[i]);
}
```

Mirek
