Subject: Jsonize/Xmlize with lambda (and common template example)
Posted by mirek on Thu, 15 Nov 2018 08:36:10 GMT
View Forum Message <> Reply to Message

It is now possible to use lambda definition of structure of data or array element with jsonizing/xmlizing. The new reference/IzeLambda show this as well as method of using single method for both JSON and XML:

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

using namespace Upp;

struct Item {
 int  value;
};

struct Data {
 Array<Item> array;
 Point       p;

 template <class IO>
 void Ize(IO& io) { // define single template function for both JSON and XML
  io
   .Var("p", p, [=] (IO& io, Point& m) { // use lambda to define how to 'ize' structure
    io("X", m.x)("Y", m.y);
   })
   .Array("values", array, [=] (IO& io, Item& m) { // use lambda to define how to 'ize' elements
    io("value", m.value);
   }, "element") // this is ignored in Json, provides tag of single element
  ;
 }

 void Xmlize(XmlIO& io)   { Ize(io); }
 void Jsonize(JsonIO& io) { Ize(io); }
};

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT|LOG_FILE);

 Data data;
 data.array.Add().value = 12345;
 data.p.x = 1;
 data.p.y = 2;

 LOG(StoreAsXML(data));
 LOG(StoreAsJson(data));
```

```
}
```

Note that, while not shown here, the important feature of using lambda (instead of Xmlize method) is the ability to define different structure, possibly even based on serialized data - this was in fact the original motivation for this...
