
Subject: Writable references to ValueArray/ValueMap/Value elements

Posted by [mirek](#) on Sat, 13 Sep 2014 14:31:10 GMT

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

Lately I have found myself doing a lot of work on hierarchical Values (usually representing some JSON). Here, the strict non-mutable value-only features of Value make things a little bit complicated. Thus, after a long period of hesitation, I have introduced possibility to get a reference to "internal" Value contained in ValueArray/ValueMap/Value. It is now possible to write

```
Value v;
for(int i = 0; i < 3; i++) {
    Value& p = v.At(i)("person");
    p("name") = "Name" + AsString(i + 1);
    p("lastname") = "LastName" + AsString(i + 1);
}
LOG(AsJSON(v));
```

and result is

```
[{"person":{"name":"Name1","lastname":"LastName1"}}, {"person":{"name":"Name2","lastname":"LastName2"}}, {"person":{"name":"Name3","lastname":"LastName3"}}]
```

However, there is a strict rule limiting the validity of such reference (caused by general nature of Value):

They are only valid until the next operation on originating Value - including just reading it!

Examples of faulty code:

```
Value m;
Value& x = m("key");
x = m; // using m as source invalidates x
```

```
Value m;
Value& x = m("key");
Value& y = m("key2"); // Invalidates x
x = 123; // undefined
```

```
Value m;
```

```
Value& x = m.At(1);  
Value m2 = m; // Invalidates x  
x = "fail"; // undefined
```

Mirek
