Subject: Can we have a ValueMap(Unique) ?
Posted by mingodad on Mon, 14 Apr 2014 13:52:06 GMT
View Forum Message <> Reply to Message

Today I spent a lot of time to figure out why some code was not working and I found that it was because I have added the same key more than once on a ValueMap, I was expecting getting back the last added value but no and till I found this problem a lot of time passed (I learned a bit more about U++ internals) but it was not fun.

Looking a the ValueMap implementation the ideal place to do it would be on the function "Add" but it is not virtual so I was thinking that adding a new type "VALUEMAPUNIQUE_V" and modify the "Add" function to:

```
void ValueMap::Add(const Value& key, const Value& value) {
 Data& d = Clone();
	if(data->GetType() == VALUEMAPUNIQUE_V)
	{
	  ///check if key already exists and throw an Exception
	}
 d.key.Add(key);
 d.value.Add(value);
}
```

What U++ users think about have a ValueMap variant like this ?

Cheers !

Subject: Re: Can we have a ValueMap(Unique) ?
Posted by mirek on Mon, 14 Apr 2014 18:29:51 GMT
View Forum Message <> Reply to Message

mingodad wrote on Mon, 14 April 2014 13:52Today I spent a lot of time to figure out why some code was not working and I found that it was because I have added the same key more than once on a ValueMap, I was expecting getting back the last added value but no and till I found this problem a lot of time passed (I learned a bit more about U++ internals) but it was not fun.

Looking a the ValueMap implementation the ideal place to do it would be on the function "Add" but it is not virtual so I was thinking that adding a new type "VALUEMAPUNIQUE_V" and modify the "Add" function to:

```
void ValueMap::Add(const Value& key, const Value& value) {
 Data& d = Clone();
	if(data->GetType() == VALUEMAPUNIQUE_V)
	{
	  ///check if key already exists and throw an Exception
	}
```

```
 d.key.Add(key);
 d.value.Add(value);
}
```

What U++ users think about have a ValueMap variant like this ?

Cheers !

Use Set instead of Add...

That said, there is another problem with ValueMap following VectorMap semantics: ordering of elements is meaningful.

Thus, if you have

```
ValueMap a, b;
a("x", 1)("y", 2);
b("y", 2)("x", 1);
```

then

```
a != b;
```

to solve that:

```
a.IsSame(b) // unordered compare
```

that said, it is not completely unlikely that we introduce some new type that has these issues altered..

Mirek
