Subject: Sorting a ValueArray ?
Posted by mingodad on Mon, 14 Apr 2014 09:49:07 GMT
View Forum Message <> Reply to Message

Hello !

How to sort a ValueArray ? Trying to use Sort() gives this error:

```
static bool CompareGetDistances(ValueMap &a, ValueMap &b) {return a[5] < b[5];}
...
ValueArray &rows = SomeValueArrayWithValueMaps();
...
Sort(rows, CompareGetDistances);
...
-----
/home/mingo/upp/uppsrc/Core/Algo.h:819:31: error: 'class Upp::ValueArray' has no member named 'Begin'
  Sort(c.Begin(), c.End(), less);
-----
```

Should we have the missing Begin() and End() implemented on ValueArray ?

Thanks in advance for any help !

Subject: Re: Sorting a ValueArray ?
Posted by mirek on Mon, 14 Apr 2014 11:46:57 GMT
View Forum Message <> Reply to Message

mingodad wrote on Mon, 14 April 2014 09:49Hello !

How to sort a ValueArray ? Trying to use Sort() gives this error:

Right now, you would have to create Vector<Value> from elements, Sort, create new ValueArray.

It is pretty untypical requirement.

Quote:
Should we have the missing Begin() and End() implemented on ValueArray ?

Would not really help as they would have to be const. It is impossible to make non-const references to ValueArray elements available. The reason is that if allowed, it is pretty simple to create referential cycles, resulting in memory leaks.

Think: ValueArray va; va.Add(); va[0] = va;

Mirek

Subject: Re: Sorting a ValueArray ?
Posted by mingodad on Mon, 14 Apr 2014 15:16:30 GMT

View Forum Message <> Reply to Message

After reading your answer and thinking a bit I've got this code to work:

```
static bool CompareGetDistances(Value &a, Value &b)
{
 return a[5] < b[5];
}
...
    ValueArray &rows = SomeValuaArrayOfValueMap();
    Vector<Value> &vv = const_cast< Vector<Value>& >(rows.Get());
    Sort(vv, CompareGetDistances);
...
```

Thanks for your help !

Subject: Re: Sorting a ValueArray ?
Posted by mirek on Mon, 14 Apr 2014 18:21:50 GMT

View Forum Message <> Reply to Message

mingodad wrote on Mon, 14 April 2014 15:16After reading your answer and thinking a bit I've got this code to work:

```
static bool CompareGetDistances(Value &a, Value &b)
{
 return a[5] < b[5];
}
...
    ValueArray &rows = SomeValuaArrayOfValueMap();
    Vector<Value> &vv = const_cast< Vector<Value>& >(rows.Get());
    Sort(vv, CompareGetDistances);
...
```

Thanks for your help !

That is not quite a good idea. If there exists a copy of ValueArray, it will get sorted too...

To stay on the safe side, I would probably rather used something like

```
 Vector<Value> v = clone(va.Get());
 Sort(v);
 va = ValueArray(pick(v));
```

(but thinking about the whole issue, it looks like "Pick" method would not be a bad idea too here...)

Mirek