Unfortunately, this seems not to work:


```
struct Test {
 Vector<int> a;
 int       b;

 Test(Vector<int>&& a, int b) : a(pick(a)), b(b) {}
};

CONSOLE_APP_MAIN
{
 Array<Test> h;
 Vector<int> v;
 v.Add(12);
 h.Create<Test>(v, 22);
 DDUMP(v.GetCount());
 h.Create<Test>(pick(v), 22);
 DDUMP(v.GetCount());
}
```


Here, I would expect const Args& for the first 'Create' and && for the second one.

For some reason, both end in && overload (I am really not sure why, IMO they should not, but they do with both MSC and GCC).

So unfortunately, to stay safe, I am going to rollback this feature. This has really trivial workaround (default constructor + Set method).

Mirek