

---

Subject: user 'Value' [FEATURE REQUEST]  
Posted by [qwerty](#) on Thu, 25 May 2006 09:04:34 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

one user Value on Arrays etc. It come would handy.

like:

```
Array<int> ints;
```

```
Value ints.userValue;
```

something like Tag. it could be on other classes, but my knowledge of upp are not so far....

---

---

Subject: Re: user 'Value' [FEATURE REQUEST]  
Posted by [mr\\_ped](#) on Thu, 25 May 2006 13:17:44 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Usually this is solved by defining your own class/struct, something like

```
struct ArrayWithVal {  
    Array<int> ints;  
    Value    userValue;  
};
```

In C++ probably construction like

```
class ArrayWithVal : public Array<int> {  
    Value    userValue;  
};  
(err.. you should probably write that as template)
```

would maybe give you advantage in some situations (ability to store both Array<> and ArrayWithVal into the Array<> \* pointer, or to give some function ArrayWithVal instead of Array<>, etc..)

Depends on what you need and how much sure you are about what you are doing, when you are designing new class.

---

---

Subject: Re: user 'Value' [FEATURE REQUEST]  
Posted by [qwerty](#) on Thu, 25 May 2006 14:13:26 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

err, yes, I tried to put a void onto this , a leave that for upp fathers or at least some upp guru, cause they know, if it is possible and easier to implement into upp or to make user his own like

you wrote...

when it's about upp, one never know

---

---

Subject: Re: user 'Value' [FEATURE REQUEST]  
Posted by [mr\\_ped](#) on Thu, 25 May 2006 19:26:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I think the solution with inheritance is the most easy and usefull one.

```
template <class T>
class ArrayWithVal : public Array<T>, Moveable< ArrayWithVal<T> > {
public:
    Value userValue;
};
```

(Maybe "MoveableAndDeepCopyOption< ... >" can be used, but Value is just "Moveable", and so I'm not sure if the DeepCopy would be still possible)

I don't think such short piece of code is worth of addition to UPP core. Actually this is the most simple example of what is OOP capable of, and why it has advantages over procedural programming for certain tasks (like this one).

Anyway, I think the "Value" is a good choice in UPP as long as you don't care about it's size in memory (i.e. you will not produce hundrets instances of that class together).

Other common solution is to use void \*, or DWORD type. Both of them allow you to store 32bit ints into user value. The void \* is also more understandable if you put there pointer pointing to your custom structure holding more values. Both those two are usual for plain C code, and as I said, Value is very good choice in UPP, unless you are memory consumption concerned.

---

---

Subject: Re: user 'Value' [FEATURE REQUEST]  
Posted by [mr\\_ped](#) on Thu, 25 May 2006 19:28:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Of course the userValue can be private/protected, then add some public methods to work with the value, if you want to hide that data from direct access.  
I think the default constructor should be enough for everyone (in this special case, not in general).

---

---

Subject: Re: user 'Value' [FEATURE REQUEST]

---

Posted by [qwerty](#) on Thu, 25 May 2006 21:24:20 GMT

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

---

more than enough so far, Upp deity bless you !

---