Subject: Re: Upp 2008.1beta2 v2 released
Posted by mdelfede on Thu, 10 Apr 2008 13:02:09 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 10 April 2008 02:59
Sorry. I am not gonna to fix this. I am no slave of some idiot who does not remember that && has higher priority than || and thus adds such idiotic warning.

What will be next? What about (x * y) + 10?

The whole purpose of operator priorities is to reduce the number of parenthesis. Do you really think this code is more readable

```
return ((c >= 'a') && (c <= 'z')) || ((c >= 'A') && (c <= 'Z')) || (c == '_');
```

Well, operator precedence warnings are (IMO) really not necessary, in particular for usual operator precedence rules.
I can see them useful only in few cases, as ternary operator ?:, but not on && and || which have well known precedence rules.
What I think we should make go away are other warnings :

1- unused variables. Those usually means that old/broken code is still present, or at least that code is still unpolished
2- missing return statement. That can hide nasty bugs too.
3- non virtual destructor on polymorphic classes. Well, in upp code style they may be not necessary, but can still bring bugs if classes are used in non-upp style code.
4- some automatic type conversions. In my experience, they can bring very difficult to find bugs.
5- un-initialized variables
6- some bad uses of references
7- other stuffs that now I don't remember

I don't know which of those are still caught by upp default warning style, but I've seen that compiling it with -wall bring really tons of warnings....

BTW, working on an open source well established library, I just caught a bug just enabling a bit more warnings on it... si I think that making compiler a bit stricter can bring benefits.

Max