Subject: Usage of "private:" in uppsrc - A strong vote for "protected:"
Posted by Zardos on Fri, 27 Jul 2007 18:10:40 GMT
View Forum Message <> Reply to Message

While using U++ more and more and still loving it. There is one thing I always fear:

As good as ultimate is - there are always some things you need or you may do different.

In this case I work around the problem and use my own classes/functions etc. But sometimes I have to change and add new fuctions to existing classes / controls.

What can I do?

1.) Change the source and hope the patch is accepted. The best way - if the patch is accepted. If not - for what ever (valid) reason - the problems start...

2.) If not accepted I can try to open a personal branch of U++ and try to keep in sync with internal patches with the public U++ source. => Not nice and much work

3.) Try do use the C++ tools to extend classes and functions. A very good way... But exactly here is the problem:

Many important member variables and functions are declared "private"! As an example: "image" in "ButtonOption". If I need access to this image I'm lost...

So I would like to suggest: If you use private in a class - Think about it again from the perspective of a U++ user who would not like to change the U++ source and may want to extend a class.

I'm not voting to make everything "public". And I understand the idea of information hiding.
But I think it is far more work to keep a modified U++ version then to change a derived class if the internals of the parent classes have changed.
So, if in doubt to make anything private or protected -> go for protected - and only for very, very rare things use private.

I really hope nobody feels offended. I just wanted to show you the problem from a U++ user perspective.

Kind regards,

Ralf

Subject: Re: Usage of "private:" in uppsrc - A strong vote for "protected:"
Posted by mirek on Fri, 27 Jul 2007 20:09:32 GMT
View Forum Message <> Reply to Message

I understand your pain, but the real problem is that parts defined as "private" have freedom to change. These are implementation things....

OK, just today with premulatiplied alpha design fiasco (or lack of), leading to fundamental interface change, this argument might sound moot, but protected really makes things "fixed".

Therefore I think you should try 1) in most cases. Please, sometimes if core team does not react, feel free to repeat the request. We are just busy doing something else...

If 1) does not work, it in most cases means you need something else from the class that we have intended it for. If it is not something "core" (like Ctrl interface), I would just copy the class and maintained the special version under different name.

Mirek
