Subject: A Few observations on U++ (lack of) documentation
Posted by arixion on Sun, 08 Oct 2006 13:34:10 GMT
View Forum Message <> Reply to Message

Hey hi, I'm a new user who just switched over from WxWidgets 'cos I foudn it a little too complex. I like UPP because I find that the code is generally cleaner and more readable (i.e. easier to debug). However, the major disadvantage UPP has to other tks like Wx, Swing, QT, and FOX, is that its features are poorly documented. Tutorials aside, what I feel is really needed for UPP is to document the extensive source code. I see that UPP has tremendous functionality, but unfortunately layout designer doesn't showcase the full set of functionality because half the classes are hidden behind USCControl, and htese are unfortuantely those controls which could make UPP shine or at least seem comparable to other tks, which it actually is. IDEs like NetBeans and WxDesigner feature the *complete* set of controls in their layout designers (and even event listeners sometimes).

I simply think that it is a shame that UPP which is really good is handicapped by the lack of even simple API documentation, which could go through a great deal in helping new users figure out the tk. It is especially sad that TPP, which would be good for navigating such documentation, isn't being sued to full capacity because there is simply no documentation to navigate!!

I sincerely think that the programmers of UPP should stop programming UPP for a while and focus on documenting all the classes. After all, the programmers do know everything about their own creations, which is much harder for teh end-user to discover on hsi or her own.

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by zsolt on Sun, 08 Oct 2006 16:36:58 GMT
View Forum Message <> Reply to Message

Quote:programmers of UPP should stop programming
I can not agree with this.
It would be better to find somebody, who like writing documentation and can ask good questions from programmers of upp.
In first step, it would be a good starting point to collect and organize the lot of information in the forum.

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by arixion on Mon, 09 Oct 2006 06:04:52 GMT
View Forum Message <> Reply to Message

zsolt wrote on Mon, 09 October 2006 00:36Quote:programmers of UPP should stop programming
I can not agree with this.
It would be better to find somebody, who like writing documentation and can ask good questions from programmers of upp.
In first step, it would be a good starting point to collect and organize the lot of information in the forum.

Hmm,

forgive me if I sounded a little rude or imposing; that wasn't my intention. And I wasn't suggesting that the programmers permanently stop programming UPP; what I meant is that it is quite important that they sit down and document their sources. This is for a number of reasons:-

1. Well-documented source-code helps in the writing of tutorials, and more than that, the easy writing of newbie tutorials. Newbies can navigate around the sources more easily. Seriously, in its current state, it would be nightmarish and intimidating for a newbie to try starting to learn UPP.

2. It aids end-users who want to be involved in further developing UPP. Clearly-documented source-code means that the third-party developers can easily see what needs to be added, what needs to be amended, and what extensions can be made. That will, I hazard to guess, attract more people to the development of UPP. Besides, it makes the software look more like its professional counterparts Wx and FOX, which have extensive source documentation. Poorly-documented code means that the third-party developer needs to spend tons of time navigating through and sorting out the source-code, time which could be productively spent on development of additional controls and features to enahnce the IDE and the software. For instance, (just as an example), I'm trying to program a CodeBlocks project import function, but firstly I can't comprehend the various methods involved in XML handling or File Browsing in the IDE. The XML reference example is far from descriptive. And while Assist++ aids tremendously in code navigation, it is extreme tedium to cycle through every method and locate the appropriate file.

3. I'm not saying that development of UPP needs to terminate. Rather, perhaps a breather is needed. I think that the programmers are the best ones to document the source-code, because they should know every component's functionality completely, since they designed them. It will be much harder for third-party documenters and end-users to do this.

4. The forum information is of good quality. But as is the problem with forums (and mailing lists as well), the topics are too focused on one aspect. They are good as a quick reference point, but they are not good if a end-user or third-party developer wishes to understand the overall functionality of the various controls and library components of UPP. And fuirther more, they don't give a basic overview, which is really what new users would need. Perhaps help forums could be aided by the inclusion of sticky FAQs explaining the basic functionality of each of the classes, especially for the complex GUI components like TreeView and DocEdit/LineEdit. Then, these would not be required is source-code were documented properly.

As a brief note: I don't mind trying to write documentation - as in tutorials or man pages - for UPP, but any attempt a trying to write documentation is hindered by incomplete understanding of the source-code, which is due to lack of API documentation.

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by zsolt on Mon, 09 Oct 2006 09:12:10 GMT

View Forum Message <> Reply to Message

Sorry for being rude in my previous post.
I mostly agree with you, that upp needs a lot of documentation. But I think, that an UPP user can start using the lot of examples and tutorials. The basics of upp are well documented, so the concepts can be understood.
I think, programming work of UPP developers is too valuable compared to doc writing.
This is my opinion of course. I'm not a developer of UPP, only a user.
If you need a deeper undertanding of the code, you can generate a source documentation using a tool like doxygen. This was my personal solution to the problem of the lack of deeper documentation. I generated docs with browsable source code and it is extremly useful in developing applications, as class and method names are very descriptive and the code of upp is very clean.
After you developed some applications using this harder way, you can start writing documentation. And you can always ask your questions in forum if you are stuck somewhere.

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by mirek on Tue, 10 Oct 2006 07:23:44 GMT
View Forum Message <> Reply to Message

Actually, I think this was discussed here several times. One visible result of discussion is that we have now all these tutorials and that about 50% of U++ now has reference documentation (should be about 70% at the next major release).

So the work is in progress. The only topic I (as "leading developer") am reluctant to do is to start using comments in the code. I think that the good code does not need comments other that commenting the purpose of class and public methods (and MAYBE something more if implementation is unusual). I plan to introduce such Assist++ improvements in future releases that will make this information as easily accessible as if it was written as comments.

As for adding .usc description scripts for less frequently used classes, I agree, it is on ToDo list. (BTW, this is not the same as "reference/USCControl" - that one is mere example how to create .usc script for your own custom control). Maybe somebody could step in and contribute a bit

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by arixion on Tue, 10 Oct 2006 13:42:34 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 10 October 2006 15:23Actually, I think this was discussed here several times. One visible result of discussion is that we have now all these tutorials and that about 50% of U++ now has reference documentation (should be about 70% at the next major release).

So the work is in progress. The only topic I (as "leading developer") am reluctant to do is to start using comments in the code. I think that the good code does not need comments other that commenting the purpose of class and public methods (and MAYBE something more if implementation is unusual). I plan to introduce such Assist++ improvements in future releases that

will make this information as easily accessible as if it was written as comments.

As for adding .usc description scripts for less frequently used classes, I agree, it is on ToDo list. (BTW, this is not the same as "reference/USCControl" - that one is mere example how to create .usc script for your own custom control). Maybe somebody could step in and contribute a bit

luzr,

hmmm, I was actually referring to comments on class and publci methods, albeit those in teh source files. Perhaps, it is because I am using the release version (605) that I don't see the comments in the code? Ok, to be fair, most of the class methods are quite descriptive, but some do need a little elaboration:

1)those fns that require "flags" need to have extra elaboration on what these flags are.

2) fns which use abbreviations - just off the cuff - like RichTextCtrl's "GetSb": what is "sb"? Or say, TabCtrl's "ChParam"?

3) btw, I seem to see that the USC scripts seem merely partial copies of the CPP sources; what's the use of USC then?

4) I think a good way to test the descriptiveness of variable names is to look at the code from the point of view of a newcomer, and see if you understand the purpose of each class property and method completely at first go. For example, looking at most code files (even of the devt version), I can't tell at first glance what the various enums are meant for. And if there is no such code documentation, then at least u should avoid giving one-letter names for arguments e.g. "Button & b, int r". After all, arguments can be a very good clue to what the method is about, or at least direct users of the libraries will know the appropriate arguments to apss the functions. This is especially important, I feel, if you want to have a black-box interface to the library. I mean, let's have an analogy: say we have a vending machine with buttons that are labelled with no pictures and no text except "a", "b", "c", "d" and so on. A person who wishes to sue this machine will probably waste alot of money before he gets what he wants, unless s/he is extremely lucky. I hoep you can see my point.

But anyway, I do think UPP is a great effort, and the fact that I'm willing to criticize it this strongly shows really how much I wish it to improve. I seriously think a little more documentation would serve it properly. And btw, for gui_tutorials, u should probably include a brief tutorial text to go with each tutorial. Learning from source-code is seriously quite hard, especially for complex controls like SQL controls where a beginner to DB programming might not understand SQL terminology u take for granted.

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by unodgs on Tue, 10 Oct 2006 14:15:47 GMT
View Forum Message <> Reply to Message

arixion wrote on Tue, 10 October 2006 09:42

hmmm, I was actually referring to comments on class and publci methods, albeit those in teh source files. Perhaps, it is because I am using the release version (605) that I don't see the comments in the code? Ok, to be fair, most of the class methods are quite descriptive, but some do need a little elaboration:

1)those fns that require "flags" need to have extra elaboration on what these flags are.

2) fns which use abbreviations - just off the cuff - like RichTextCtrl's "GetSb": what is "sb"? Or say, TabCtrl's "ChParam"?

3) btw, I seem to see that the USC scripts seem merely partial copies of the CPP sources; what's the use of USC then?

4) I think a good way to test the descriptiveness of variable names is to look at the code from the point of view of a newcomer, and see if you understand the purpose of each class property and method completely at first go. For example, looking at most code files (even of the devt version), I can't tell at first glance what the various enums are meant for. And if there is no such code documentation, then at least u should avoid giving one-letter names for arguments e.g. "Button &b, int r". After all, arguments can be a very good clue to what the method is about, or at least direct users of the libraries will know the appropriate arguments to apss the functions. This is especially important, I feel, if you want to have a black-box interface to the library. I mean, let's have an analogy: say we have a vending machine with buttons that are labelled with no pictures and no text except "a", "b", "c", "d" and so on. A person who wishes to sue this machine will probably waste alot of money before he gets what he wants, unless s/he is extremely lucky. I hoep you can see my point.

But anyway, I do think UPP is a great effort, and the fact that I'm willing to criticize it this strongly shows really how much I wish it to improve. I seriously think a little more documentation would serve it properly. And btw, for gui_tutorials, u should probably include a brief tutorial text to go with each tutorial. Learning from source-code is seriously quite hard, especially for complex controls like SQL controls where a beginner to DB programming might not understand SQL terminology u take for granted.

Frankly, I like upp coding style a lot. I'm sure many more advanced programmers here agree with me. And I wouldn't change there anything (Instead of code comments I would like to see decription of ideas and concepts behind it or description of tricks used here and there).
From my point of view newbie doesn't need to look into sources, especialy now, when we have most stuff documented quite well. I agree that it probably could be better organised, even written. But what upp really need are tutorials. More and more. The only problem is that Mirek is the only person doing this now...
But even now you have references and examples as well as gui_tutorials. More info about sql you can find in upp introduction. I think it's enough to start writing some nice apps

---

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by mirek on Tue, 10 Oct 2006 14:25:58 GMT

Quote:
because I am using the release version (605)

Maybe this is the part of problem, a lot was finished since last major release. There are now quite good GUI, NTL (well, that one existed before, but is now fixed) and Image tutorials. We will add more as time comes (perhaps more chapters to GUI, something about Core, short Draw tutorial and SQL tutorial).

I cannot quite recommend current dev release (it is working, but because of flaw in install process, you cannot recompile IDE with it), but when next dev goes out, try it.

Tutorial are also accessible via website:

http://upp.sourceforge.net/www$uppweb$documentation$en-us.ht ml

Quote:
3) btw, I seem to see that the USC scripts seem merely partial copies of the CPP sources;

Good to hear that That was the design decision (and the main reason to develop Esc scripting language).

Quote:
what's the use of USC then?

It is script interpreted by TheIDE, so that you can use them at design time. Sure, alternatively this could be also managed by compilating .cpp in some form of .dll with defined design mode interface (Windows .ocx widgets are using this approach), but that is much more heavyweight IMO.

Quote:
But anyway, I do think UPP is a great effort, and the fact that I'm willing to criticize it this strongly shows really how much I wish it to improve. I seriously think a little more documentation would serve it properly. And btw, for gui_tutorials, u should probably include a brief tutorial text to go with each tutorial. Learning from source-code is seriously quite hard, especially for complex controls like SQL controls where a beginner to DB programming might not understand SQL terminology u take for granted.

As I said, working on it. Anyway, in fact, neverbody so far said U++ is tool for beginners. In reality, many of users seem to be experienced C++ programmers that are disgusted with existing tools. That of course does not mean we should not work on improving the learning curve. We do.

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by arixion on Wed, 11 Oct 2006 10:50:45 GMT
View Forum Message <> Reply to Message

unodgs wrote on Tue, 10 October 2006 22:15arixion wrote on Tue, 10 October 2006 09:42
hmmm, I was actually referring to comments on class and publci methods, albeit those in teh source files. Perhaps, it is because I am using the release version (605) that I don't see the comments in the code? Ok, to be fair, most of the class methods are quite descriptive, but some do need a little elaboration:

1)those fns that require "flags" need to have extra elaboration on what these flags are.

2) fns which use abbreviations - just off the cuff - like RichTextCtrl's "GetSb": what is "sb"? Or say, TabCtrl's "ChParam"?

3) btw, I seem to see that the USC scripts seem merely partial copies of the CPP sources; what's the use of USC then?

4) I think a good way to test the descriptiveness of variable names is to look at the code from the point of view of a newcomer, and see if you understand the purpose of each class property and method completely at first go. For example, looking at most code files (even of the devt version), I can't tell at first glance what the various enums are meant for. And if there is no such code documentation, then at least u should avoid giving one-letter names for arguments e.g. "Button & b, int r". After all, arguments can be a very good clue to what the method is about, or at least direct users of the libraries will know the appropriate arguments to apss the functions. This is especially important, I feel, if you want to have a black-box interface to the library. I mean, let's have an analogy: say we have a vending machine with buttons that are labelled with no pictures and no text except "a", "b", "c", "d" and so on. A person who wishes to sue this machine will probably waste alot of money before he gets what he wants, unless s/he is extremely lucky. I hoep you can see my point.

But anyway, I do think UPP is a great effort, and the fact that I'm willing to criticize it this strongly shows really how much I wish it to improve. I seriously think a little more documentation would serve it properly. And btw, for gui_tutorials, u should probably include a brief tutorial text to go with each tutorial. Learning from source-code is seriously quite hard, especially for complex controls like SQL controls where a beginner to DB programming might not understand SQL terminology u take for granted.

Frankly, I like upp coding style a lot. I'm sure many more advanced programmers here agree with me. And I wouldn't change there anything (Instead of code comments I would like to see decription of ideas and concepts behind it or description of tricks used here and there).
From my point of view newbie doesn't need to look into sources, especialy now, when we have most stuff documented quite well. I agree that it probably could be better organised, even written. But what upp really need are tutorials. More and more. The only problem is that Mirek is the only person doing this now...
But even now you have references and examples as well as gui_tutorials. More info about sql you can find in upp introduction. I think it's enough to start writing some nice apps

unod,

I see your point. In fact, actually what you were saying is exactly what I ahve been trying to say, save for lack of concise language. What I am asking for essentially is to have more tutorials which are documented properly. Reference examples can be learnt from, but only if they are given at least a brief comment on each line of code that relates to a particular demonstrated function, or a summary of what the code means afterward. Otherwise, it's just as good as someone dumping the source-code of a complex program and asking you to learn the programming language from studying that.

ANd perhaps I need to clarify: I am a newbie to UPP, not to GUI programming (I have tried Swing a few years back and FOX more recently) or C++ Programming in general. I come to UPP with 2 intentions - as I suppose is typical of users of open-source programs - : To develop programs using UPP and to extend UPP with additional components. And at this point in time, I'm focusing on the second part, more specifically I am trying to port (FX)Scintilla to UPP and to create a version of Irrlicht embedded in UPP. I am attempting to identify links between Scintilla and CodeEdit, so that I don't need to repeat coding of fucntionality i.e. I can simply extend UPPScintilla from CodeEdit, with the only addition being code-folding capability. For the first, the CodeEdit source is fully blank on any fucntionality documentation that can help a beginning developer. And for the second, I don't really have any idea how to compile the Irrlicht Source-code (available with *.cbp) on TheIDE.

Btw, is Mirek the ONLY developer of UPP??

Subject: Re: A Few observations on U++ (lack of) documentation
Posted by unodgs on Wed, 11 Oct 2006 11:42:36 GMT
View Forum Message <> Reply to Message

arixion wrote on Wed, 11 October 2006 06:50

I see your point. In fact, actually what you were saying is exactly what I ahve been trying to say, save for lack of concise language. What I am asking for essentially is to have more tutorials which are documented properly. Reference examples can be learnt from, but only if they are given at least a brief comment on each line of code that relates to a particular demonstrated function, or a summary of what the code means afterward. Otherwise, it's just as good as someone dumping the source-code of a complex program and asking you to learn the programming language from studying that.

Agreed. All example should have descriptions. From the other hand at least in my opinion examples are not complex, they seem rather simple but longer than references.
Quote:
ANd perhaps I need to clarify: I am a newbie to UPP, not to GUI programming (I have tried Swing a few years back and FOX more recently) or C++ Programming in general

I'm sorry if I suggested in my previous post lower qualifications

Quote:
. I come to UPP with 2 intentions - as I suppose is typical of users of open-source programs - : To develop programs using UPP and to extend UPP with additional components. And at this point in time, I'm focusing on the second part,

I agree that it is easier to extend upp if the code of it is good documented. But personally I don't like to comment the code (but I try to comment only "hard-parts") and to read over-commented code as well. Maybe if theide was able to hide some parts of text like visual studio in c# (#refion #endregion) everyone would be happy then. Newcomers turn them on, more familiar with upp code users turn them off.

Quote:
more specifically I am trying to port (FX)Scintilla to UPP and to create a version of Irrlicht embedded in UPP. I am attempting to identify links between Scintilla and CodeEdit, so that I don't need to repeat coding of fucntionality i.e. I can simply extend UPPScintilla from CodeEdit, with the only addition being code-folding capability. For the first, the CodeEdit source is fully blank on any fucntionality documentation that can help a beginning developer. And for the second, I don't really have any idea how to compile the Irrlicht Source-code (available with *.cbp) on TheIDE.

Interesting, I wanted to port scintilla myself. Unfortunately because of lack of time I only downloaded the sources...
Anyway, that would be great if you could do that!

Quote:
Btw, is Mirek the ONLY developer of UPP??

No. There is Tomas Rylek too, but he seems to be busy man now...
I'm trying to help implementing new widgets.
Some users like zsolt are sending patches from time to time.