
Subject: U++ suggestions

Posted by [amrein](#) on Fri, 29 Aug 2008 16:43:02 GMT

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

I'm a newcomer so I'm not already completely "corrupted" by U++ . What I will say is like saying to "vi editor" people: your program is powerful but could be more newbies friendly.

Don't hate me too much:

- All files and directories in U++ should use lower-case names because of portability issue between Windows and Linux (Linux files names are case sensitive).
- Ctrl should be renamed Control.
- All class names should use complete names instead of developer's specific abbreviations.
- All .cpp and .h names should use complete names instead of developer's specific abbreviations. Exception: multi class files but the file name could still use complete purpose name. There is no more the 6+3 MSDOS file names restriction in new operating system and with automatic completion there is no reason to create new English word.
- "Assembly" "Nest" and "Package" should be just "Collection" "Directory" "Package". "Assembly" and "Nest" words are complicated and should die. I don't hate birds but I'm still confused by those names
- First TheIDE window (when you open TheIDE). It shows two rows "Assembly" and "Packages". Should be "Directories" and "Packages" and the main title should be "Select main package from "Default Collection". This doesn't break the "Assembly" original idea. People will clearly be able to understand what they are doing. People could still create Collection/Assemblies if they need it. (I hope this time I was more clear)
- All external dependencies from thirds party, those not created by U++ team but distributed with U++, should be in a separated directory "3rdparty" for easy directory understanding and easy license attribution.
- RichText is not RichTextFormat (RTF from MS). I was confused first when I saw RichText. Should be replaced by HTML implementation or perhaps real RTF or OASIS (OpenDocument file format from OpenOffice also known as ODF).
- Topic++ and other tools like that should have their own .exe. At least, for big enterprise or just contributors wanting to just create documentation, translation, ... or people wanting to use "make" without calling TheIDE each time.
- In TheIDE, the main window, just after the first one (i.e "Main package selection"). In the top left frame, main packages dependencies should be separated from main package with a clickable borderless button in grey saying "Dependencies list" (click on it, and select other dependencies). Separators also in bottom left frame...

(...)

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [mirek](#) on Fri, 29 Aug 2008 18:01:21 GMT

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

amrein wrote on Fri, 29 August 2008 12:43 I'm a newcomer so I'm not already completely "corrupted" by U++ . What I will say is like saying to "vi editor" people: your program is powerful but could be more newbies friendly.

Don't hate me too much:

It is OK, but we can hardly do more than explain current status

Quote:

- All files and directories in U++ should use lower-case names because of portability issue between Windows and Linux (Linux files names are case sensitive).

Well, we had some minor problems in the past with this, but in reality, it is not an issue.

Quote:

- "Assembly" "Nest" and "Package" should be just "Collection" "Directory" "Package". "Assembly" and "Nest" words are complicated and should die. I don't hate birds but I'm still confused by those names

Hehe, in fact, the only one I suggested to rename is "Package". IMO, the better name would be "Unit"

As for "Nest" vs "Directory", I am afraid that it would clash with normal directories, leading to various misunderstandings. E.g. "Package is directory. Packages as grouped in Directory".

But you are not the first one to complain about it, so we might still find a better name.

Quote:

- First TheIDE window (when you open TheIDE). It shows two rows "Assembly" and "Packages". Should be "Directories" and "Packages" and the main title should be "Select main package from "Default Collection".

Ehm, what is "Default Collection"?

Quote:

- All external dependencies from third party, those not created by U++ team but distributed with

U++, should be in a separated directory "3rdparty" for easy directory understanding and easy license attribution.

They are separated in "plugin". OK, over time, the problem appeared that some "inhouse" packages got there too...

Quote:

- RichText is not RichTextFormat (RTF from MS). I was confused first when I saw RichText. Should be replaced by HTML implementation or perhaps real RTF or OASIS (OpenDocument file format from OpenOffice also known as ODF).

RichText is just data container for rich texts. It can be now created (imported, loaded) from RTF or QTF. I would be happy to have HTML and ODF (and MS Office) support as well, but it is a lot of work - that is the real constraint.

Note that QTF was created because HTML lacks true typography and other formats are too hard to be coded manually. You certainly would not want to create text in message boxes in ODF....

Quote:

- Topic++ and other tools like that should have their own .exe. At least, for big enterprise or just contributors wanting to just create documentation, translation, ... or people wanting to use "make" without calling TheIDE each time.

The main beauty of Topic++ will be the tight integration with the code. I cannot see how to do that without having it in TheIDE.

One thing I really wanted to have (and it was told several times) is non-GUI "umk".

Quote:

- In TheIDE, the main window, just after the first one (i.e "Main package selection"). In the top left frame, main packages dependencies should be separated from main package with a clickable borderless button in grey saying "Dependencies list" (click on it, and select other dependencies).

That is fine suggestion. OTOH, I do not see this as critical feature, in fact I think that for complex overview and dependencies management, users should learn to invoke Package organizer...

Quote:

Separators also in bottom left frame...

?

Mirek

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [amrein](#) on Fri, 29 Aug 2008 21:52:14 GMT

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

Quote:

Quote:

- All files and directories in U++ should use lower-case names because of portability issue between Windows and Linux (Linux files names are case sensitive).

Well, we had some minor problems in the past with this, but in reality, it is not an issue.

For the U++ team perhaps. For a large project with hundred header file, the portability is lost and you need to edit your .cpp + rename a few .h (need to resync .cpp #include statements with your .h names).

Without a strict layout, if I get a package from someone using Windows only and want to compile it to build a rpm or a deb package, I will need to add fix just for this. And for each release, I will have to update those patches as soon as something has changed in one of those #include statements or if there are new .h.

Quote:

Quote:

- "Assembly" "Nest" and "Package" should be just "Collection" "Directory" "Package". "Assembly" and "Nest" words are complicated and should die. I don't hate birds but I'm still confused by those names

Hehe, in fact, the only one I suggested to rename is "Package". IMO, the better name would be "Unit"

Where is my gun to shot myself? LOL.

You mean, you would like it to be more complicated? Please don't. I beg you. Really. Don't over complicate Ultimate++.

Your toolkit, your baby, is the best toolkit I ever saw. For me, It could replace GTK, QT, wxWindows... It's simple. It just needs to be simpler, not the contrary. It gave me back the desire to code.

I'm not a company. I'm alone. I'm in vacation. I'm just trying to do what I like to do. I'm not an everyday programmer. Programming is just an hobby. My company do not sell software. They are not around applications and libraries. They use mainly PHP/Javascript/MySQL/Linux for servers and Linux+Windows for clients. That's all. No need for U++.

What I mean is please, don't do this error. I know we are not lemmings (I love this game) but really, you shouldn't go this way, but the other one, the simplification one.

Quote:

As for "Nest" vs "Directory", I am afraid that it would clash with normal directories, leading to various misunderstandings. E.g. "Package is directory. Packages as grouped in Directory".

But you are not the first one to complain about it, so we might still find a better name.

You have the same problem with the old naming. That won't change.

When someone says "A package", it's like opening an HTML file. Anyone understands "open a specific file in a directory. This file contains all information about how to link all together and how to build and show the final target".

Quote:

Quote:

- First TheIDE window (when you open TheIDE). It shows two rows "Assembly" and "Packages". Should be "Directories" and "Packages" and the main title should be "Select main package from "Default Collection".

Ehm, what is "Default Collection"?

The one TheIDE should use the first time you open it.

The idea of Collection is to break "Assembly", "Nest" and "Packages" into "Collections", "Directories" and "Packages".

When you open a "Directory", you get all Packages there. When you open a "Collection", you get all "Directories" with "Packages" you would like to work together.

Example: you want to create "TheIDE 2.0". You don't want to mess with the old TheIDE directory. You create a new Collection and you add all directories except the old TheIDE one. That way, your new TheIDE won't conflict with the old one and you will still be able to get all other packages functionalities.

Quote:

Quote:

- All external dependencies from third party, those not created by U++ team but distributed with U++, should be in a separated directory "3rdparty" for easy directory understanding and easy license attribution.

They are separated in "plugin". OK, over time, the problem appeared that some "inhouse" packages got there too...

Now I understand why I couldn't appreciate the main purpose of this directory.

Quote:

Quote:

- RichText is not RichTextFormat (RTF from MS). I was confused first when I saw RichText. Should be replaced by HTML implementation or perhaps real RTF or OASIS (OpenDocument file format from OpenOffice also known as ODF).

RichText is just data container for rich texts. It can be now created (imported, loaded) from RTF or QTF. I would be happy to have HTML and ODF (and MS Office) support as well, but it is a lot of work - that is the real constraint.

Note that QTF was created because HTML lacks true typography and other formats are too hard to be coded manually. You certainly would not want to create text in message boxes in ODF....

Qt use HTML tag in their help. A few years ago, I translated all Qtopia 1.7.x + OPIE help and applications.

Quote:

Quote:

- Topic++ and other tools like that should have their own .exe. At least, for big enterprise or just contributors wanting to just create documentation, translation, ... or people wanting to use "make" without calling TheIDE each time.

The main beauty of Topic++ will be the tight integration with the code. I cannot see how to do that without having it in TheIDE.

One thing I really wanted to have (and it was told several times) is non-GUI "umk".

Let me explain what I think is one of the most missing application for programmers. An application answering those questions:

- How can I have the documentation directly into my code
- How can I see where are undocumented classes or functions
- How can I read good documentation about the API without messing with the entire source code

Doxygen gives a part of the answer. Topic++ gives another part.

My idea is to have Topic++ for internal code documentation (but IN the .h/.cpp) and use doxygen to create the result.

Quote:

Quote:

- In TheIDE, the main window, just after the first one (i.e "Main package selection"). In the top left frame, main packages dependencies should be separated from main package with a clickable borderless button in grey saying "Dependencies list" (click on it, and select other dependencies).

That is fine suggestion. OTOH, I do not see this as critical feature, in fact I think that for complex overview and dependencies management, users should learn to invoke Package organizer...

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [mirek](#) on Fri, 29 Aug 2008 23:52:03 GMT

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

amrein wrote on Fri, 29 August 2008 17:52Quote:

Quote:

- All files and directories in U++ should use lower-case names because of portability issue between Windows and Linux (Linux files names are case sensitive).

Well, we had some minor problems in the past with this, but in reality, it is not an issue.

For the U++ team perhaps. For a large project with hundred header file, the portability is lost and you need to edit your .cpp + rename a few .h (need to resync .cpp #include statements with your .h names).

Then keep rules for the code in your large project. In U++, everything matches. No need to worry. Move on.

Besides, in large multiplatform project, you will have much harder problems to solve than this (just compiler differences tend to be nasty).

Quote:

Quote:

Hehe, in fact, the only one I suggested to rename is "Package". IMO, the better name would be "Unit"

Where is my gun to shot myself? LOL.

You mean, you would like it to be more complicated? Please don't. I beg you. Really. Don't over complicate Ultimate++.

What makes it more complicate? A renaming?

IMO, the closest thing to "package" I know is old Turbo Pascal unit. Except U++ unit is represented by directory, not a single file.

Quote:

What I mean is please, don't do this error. I know we are not lemmings (I love this game) but really, you shouldn't go this way, but the other one, the simplification one.

What error?

Quote:

The one TheIDE should use the first time you open it.

The idea of Collection is to break "Assembly", "Nest" and "Packages" into "Collections", "Directories" and "Packages".

When you open a "Directory", you get all Packages there. When you open a "Collection", you get all "Directories" with "Packages" you would like to works together.

Why do I have a strange feeling that you still do not really understand the concept?

Quote:

Example: you want to create "TheIDE 2.0". You don't want to mess with the old TheIDE directory. You create a new Collection and you add all directories except the old TheIDE one. That way, your new TheIDE won't conflict with the old one and you will still be able to get all other packages functionalities.

Yep. That is one purpose. In fact, you can even leave "old" theide in uppsrc and just prepend your new package with new theide. First package match wins.

Quote:

Qt use HTML tag in their help. A few years ago, I translated all Qtopia 1.7.x + OPIE help and applications.

HTML does not work for reports and office documents -> end of story.

[quote]

- How can I have the documentation directly into my code
 - How can I see where are undocumented classes or functions
 - How can I read good documentation about the API without messing with the entire source code
- [/quite]

Patience. We will get there soon. Right now I still have to solve some other issues.

But, speaking about it, the idea is that you will see documentation icons in the left bar. Hovering mouse over it will display existing documentation, clicking will open T++ with preformatted entry (if missing).

Quote:

My idea is to have Topic++ for internal code documentation (but IN the .h/.cpp) and use doxygen to create the result.

My idea is to try something different first.

Mirek

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [amrein](#) on Sat, 30 Aug 2008 02:10:52 GMT

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

Quote:Quote:

What I mean is please, don't do this error. I know we are not lemmings (I love this game) but really, you shouldn't go this way, but the other one, the simplification one.

What error?

Package -> Unit

Quote:Quote:

The one TheIDE should use the first time you open it.

The idea of Collection is to break "Assembly", "Nest" and "Packages" into "Collections", "Directories" and "Packages".

When you open a "Directory", you get all Packages there. When you open a "Collection", you get all "Directories" with "Packages" you would like to works together.

Why do I have a strange feeling that you still do not really understand the concept?

You are true. I really need more time to understand it. I tried again an again and each time someone show me I'm missing something.

Well, 34 years old, sys+net admin, and U++ make me feel I'm becoming old.

My problem: I compare "Assemblies" with Linux file system layout. "Nest" with LD_LIBRARY_PATH + PATH. "Packages" with SRPMS and their build dependencies.

Quote:Quote:

Qt use HTML tag in their help. A few years ago, I translated all Qtopia 1.7.x + OPIE help and applications.

HTML does not work for reports and office documents -> end of story.

I don't understand. You can have a complete book in HTML. Doxygen output can be HTML. You can select whatever font you want, colours, styles, tables, frames, even scripts... All features from QTF are in HTML. Only problem: include images.

Quote:Quote:

- How can I have the documentation directly into my code
- How can I see where are undocumented classes or functions
- How can I read good documentation about the API without messing with the entire source code

Patience. We will get there soon. Right now I still have to solve some other issues.

But, speaking about it, the idea is that you will see documentation icons in the left bar. Hovering mouse over it will display existing documentation, clicking will open T++ with preformatted entry (if missing).

Cool!! If this help could be in fact in the source (with HTML tags, because doxygen can handle them) and if TheIDE could show or hide it from the source, then it would completely turn upside down the world of IDEs.

Quote:Quote:

My idea is to have Topic++ for internal code documentation (but IN the .h/.cpp) and use doxygen to create the result.

My idea is to try something different first.

Mirek

Thanks for your answers. I don't bother you more. Ask me if you want help for the automatic build system.

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [mirek](#) on Sat, 30 Aug 2008 08:05:48 GMT

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

amrein wrote on Fri, 29 August 2008 22:10

You are true. I really need more time to understand it. I tried again an again and each time someone show me I'm missing something.

Well, 34 years old, sys+net admin, and U++ make me feel I'm becoming old.

My problem: I compare "Assemblies" with Linux file system layout. "Nest" with LD_LIBRARY_PATH + PATH. "Packages" with SRPMS and their build dependencies.

Well, I think this nicely demonstrates what is wrong with "Package" name

Package is basically a library in source form with dependencies etc.

Nest is just a directory that contains packages.

Assembly is ordered set of nests. Means, if something is close to LD_LIBRARY_PATH + PATH, it is Assembly. Also, you can have multiple assemblies. Means assembly is assembly of various groups of packages.

Quote:

I don't understand. You can have a complete book in HTML. Doxygen output can be HTML. You can select whatever font you want, colours, styles, tables, frames, even scripts... All features from QTF are in HTML. Only problem: include images.

Yep, that is one problem. There are more.

Quote:

Cool!! If this help could be in fact in the source (with HTML tags, because doxygen can handle them) and if TheIDE could show or hide it from the source, then it would completely turn upside down the world of IDEs.

For now, the idea is to have then in T++. That is BTW one of reasons why T++ is inside package - it will always travel with the code.

In fact, bases of this already exist, see

[http://www.ultimatepp.org/app\\$ide\\$Topic\\$en-us.html](http://www.ultimatepp.org/appideTopic$en-us.html)

the two problems are that

- a) without the left bar in the editor, the process is tedious
- b) our C++ parser needs to be more reliable

Quote:

Thanks for your answers. I don't bother you more. Ask me if you want help for the automatic build system.

It is OK, keep complaining. I do not see this as completely unproductive, if nothing else, it keeps me remembering why we have chosen some ways...

Mirek

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [cbpporter](#) on Sat, 30 Aug 2008 12:05:48 GMT

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

Quote:My problem: I compare "Assemblies" with Linux file system layout. "Nest" with LD_LIBRARY_PATH + PATH. "Packages" with SRPMS and their build dependencies. To understand this concept I think you need to think a little out-of-the-box and forget the stupid concept of a module that has been created by using text inclusion in C/C++. You need to consider things from the standpoint of a modern and true module system.

But it could be that different persons have different understanding of what packages, nest and assemblies mean. Here is my take: packages define a group of classes and functions which are tightly related logically and which try to offer the means to solve a problem. Like Core offering a basic library of IO and data structures, CtrlCore offering a mechanism to create widgets and CtrlLib offering a set of concrete widgets. Depending on definition, packages could also be considered modules, since U++ packages tend to be more general (i.e. Core being a basic module, which could contain packages like: IO, Containers, i18n, MemoryAllocator, MT,... (not the case though)). Packages are also independent, as in they don't require to physically contain other packages, and dependencies are simply referred to. Packages are also movable in the directory structure.

Nest are group of packages and can be considered modules or libraries. Depending on your need, they can be considered projects also, but more often I think they are group of projects (from a Visual Studio perspective).

Assembles are sets of nests which are used to build an application. For example, you could have the U++ nest, and external nest for ODF parsing and a nest that defines a GUI. You combine them in an assembly and are limited to these three nests together with their packages. By combining them, you get the source code equivalent of you application.

It would be really fun if by these statements I manage to shoot myself in the foot and have a completely wrong opinion about this layout . If that's the case, then Mirek, please don't give up and try to explain things so we can both get it.

But I agree that it is confusing. Especially the package chooser, where you get to choose your assembly, but have no idea what nests it includes. Also, there are assemblies formed out of one nest, and because nests have no name or visible entity, you can get confused. Also, by grouping all packages from all nests in you assembly alphabetically, the tight logical order which may be present in a nest is broken. Why do I have an IO package next to a GUI package? It would be nice if if nest where nameable and we could get a tree in the package chooser where nests are isolated on unique top level branches and packages are leaves.

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...

Posted by [mirek](#) on Sat, 30 Aug 2008 13:12:47 GMT

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

cbpporter wrote on Sat, 30 August 2008 08:05

Especially the package chooser, where you get to choose your assembly, but have no idea what nests it includes.

Well, for starters, maybe a column showing a nest where the package is found would be a good idea?

Mirek

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...
Posted by [cocob](#) on Mon, 01 Sep 2008 09:31:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

From my side i think that a++ module/package system is perfect.
The only feature missing for me is to allow package with sources in subdirectories.

cocob

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...
Posted by [cbpporter](#) on Mon, 01 Sep 2008 09:36:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

cocob wrote on Mon, 01 September 2008 12:31 From my side i think that a++ module/package system is perfect.
The only feature missing for me is to allow package with sources in subdirectories.

cocob

That is possible right now. If you package has subfolders, you can use files from there also.

You can also place packages in a subfolder, like with "plugin/nnn" packages.

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...
Posted by [cocob](#) on Mon, 01 Sep 2008 09:47:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes it works but is there a way to hide the path ?

Subject: Re: User lists of "bad" naming of classes, functions etc in U++...
Posted by [mirek](#) on Mon, 01 Sep 2008 12:00:49 GMT

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

cocob wrote on Mon, 01 September 2008 05:47 Yes it works but is there a way to hide the path ?

Is that a reasonable requirement?

For me, if it is in subdir relative to package dir, I want to know...

Mirek
