Subject: Documentation and Topic++
Posted by mirek on Tue, 13 Nov 2007 18:41:46 GMT

View Forum Message <> Reply to Message

Mindtraveller wrote on Tue, 13 November 2007 08:33OK. I started making new articles.

For now, two things impedes the most:
1) Unnatural division of package docs by "src" and "srcdoc". After some learning it seems like one stands for index while other stands for manual.
I would refuse such a division because index must be one for all packages (smth. like MSDN where I type first letters and got functions starting with them). So all we need in the package docs tree is articles (I would refuse treeview of structure too - as described in forum messages).

Well, first of all, the idea behind Topic++ is that it should serve as tool for documenting both the program structure (sources), but also the application being developed (help, application manual).

That is the original idea behind documentation groups.

Now "predefined" groups are

src - this should contain a straight *reference* to the code *interface*. Note that theide is able to bind documents to the code. We have big plans on improving on this. (think "reference" part of MSDN API documents).

srcdoc - this should be for other programming documents - guides, explanations etc...

srcimp - this is intended to contain the comments about code implementation.

app - application documents - help, manuals etc...

Quote:
2) Inexistence of first common page. I think there must be root page, and this page must be default (but when we click some package` article, it shows in help window).

For now this root article is finished and I started writing inner articles, which were linked from root.

"index" page is considered "first common". Obviously, there is a little problem as this really works well only for single group in single package...

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Wed, 14 Nov 2007 03:32:06 GMT

View Forum Message <> Reply to Message

I think this is of course great to have one help engine for development docs and user (not programmer) documentation on application developed. So I vote for the original idea.

Next, from the design point of view, treeview-control guiding system as well as "predefined groups" seems to me not very optimal.

I`ll try to make suggestions how to increase usability. Of course, everything below is my point of view and a subject for discussion:

1) Leving behind any "magical words" line "srcdocs","srcimp". We simply don`t need that kind of magic. We need simple things to concentrate attention to the help text itself - without user`s brain guiding through magical labyrinths.

2) Information on package must be concentrated in one articles-tree. Each article has index-tags. So user has 3 ways to look at docs: manual view mode, index and full-text-search modes. Last two modes I`d like to be as MSDN ones.

This approach seems to me little more general. General enough to handle with programmers` manual as easy as user help.

3) src* groups are presented as:
src - number of articles, generated by TheIDE (and linked from some other articles if necessary). I`d prefer some package article to have these links listed by logical or functional groups (alphabetic access is available through Index mode).

srcimp and srcdoc articles are represented as links from (or part of) articles about respective subjects.
The general approach is the same: different kinds of information about subject must be in easy reach from parent article about subject. It seems very obvious way of making things user-oriented.

4) Fimally, general view of help system with approach suggested will be something like this:

5) Thoughts about root article I`ll try to write later, as all ideas about it come in order.

File Attachments
1) Untitled-1.gif, downloaded 1609 times

Welcome to the **Ultimate++** framework.

Here you may learn all things you need to make safer and more effective programs with U++.

New to U++? Read this:
- Innovative approaches to organize program

Most commonly used topics are discussed in
- Most common and useful things
- Program control

Graphical user interface features of U++ are discussed in
- Windows and controls

Advanced topics are
- Character sets and internationalization
- Utility and extension classes
- Compilation and linking
- TheIDE

Kran
CtrlLib
CtrlCore
RichText
PdfDraw
Draw
plugin/bmp
Core
plugin\z
plugin\png

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 14 Nov 2007 10:46:01 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Tue, 13 November 2007 22:32I think this is of course great to have one help engine for development docs and user (not programmer) documentation on application developed. So I vote for the original idea.

Next, from the design point of view, treeview-control guiding system as well as "predefined groups" seems to me not very optimal.

Yep, I agree. The help page definitely needsd an improvement.

Quote:
I`ll try to make suggestions how to increase usability. Of course, everything below is my point of view and a subject for discussion:

1) Leving behind any "magical words" line "srcdocs","srcimp". We simply don`t need that kind of magic. We need simple things to concentrate attention to the help text itself - without user`s brain guiding through magical labyrinths.

Mostly agree, but I think medium advanced user will need to know those magic words too, so that he can use Topic++ to write docs...

Considering other points, the only thing I would like to add is the vision that the future ide will provide much better code/documentation integration.

One of ideas is that browsing the .h file, you will have small icons in left bar (when breakpoints usually are) for each code element (e.g. method declaration); moving the mouse over that icon will popup documentation for the item.

If documentation is not available for item, icon will look differently and clicking it will move you to topic++, generating the text template to document the method.

Similar capabilities should be available in new class browser too.

In .cpp, the idea is that instead of "src", these will lead to "srcimp".

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Wed, 14 Nov 2007 12:24:36 GMT
View Forum Message <> Reply to Message

Quote:Mostly agree, but I think medium advanced user will need to know those magic words too, so that he can use Topic++ to write docs...

One of ideas is that browsing the .h file, you will have small icons in left bar (when breakpoints usually are) for each code element (e.g. method declaration); moving the mouse over that icon will popup documentation for the item.

If documentation is not available for item, icon will look differently and clicking it will move you to topic++, generating the text template to document the method.

Similar capabilities should be available in new class browser too.

In .cpp, the idea is that instead of "src", these will lead to "srcimp". Could you please describe why exactly is it needed to use these words? I think it is way too hard for understanding "by default".

Yes, user may learn manual on that subject, but what for?

It`s nothing prevents us from adding these comments simply by clicking left icon and typing docs in the same window, without any additional identifiers. For me it is so natural, that I have to think of arguments to make this thought clear.

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 14 Nov 2007 12:41:18 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Wed, 14 November 2007 07:24Quote:Mostly agree, but I think medium advanced user will need to know those magic words too, so that he can use Topic++ to write docs...

One of ideas is that browsing the .h file, you will have small icons in left bar (when breakpoints usually are) for each code element (e.g. method declaration); moving the mouse over that icon will popup documentation for the item.

If documentation is not available for item, icon will look differently and clicking it will move you to topic++, generating the text template to document the method.

Similar capabilities should be available in new class browser too.

In .cpp, the idea is that instead of "src", these will lead to "srcimp". Could you please describe why exactly is it needed to use these words? I think it is way too hard for understanding "by default". Yes, user may learn manual on that subject, but what for?

It`s nothing prevents us from adding these comments simply by clicking left icon and typing docs in the same window, without any additional identifiers. For me it is so natural, that I have to think of arguments to make this thought clear.

Well, if nothing else, groups are important if you are importing Topic++ into your code - you do so by package-group quantities.

So perhaps for documenting the code, you could do without it, but for using Topic++ as application document resources, understanding of groups is unavoidable.

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Fri, 16 Nov 2007 00:36:14 GMT
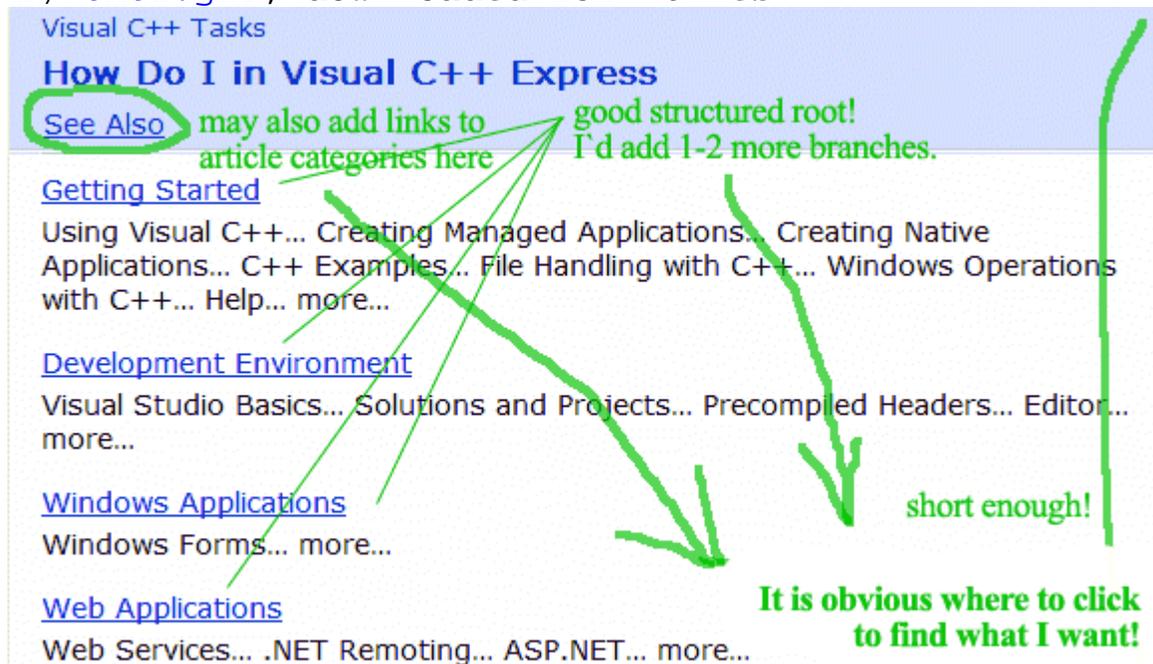View Forum Message <> Reply to Message

luzr, I temporarily lost all the words and arguments to tell that simple things are better than complex ones.   Just need time to collect arguments for the thing I feel subconciously.)
This strong subconcious feeling tells (actually, sreams when I think of usability) me this (see

below). And I`m just looking for formal and constructive words to express all that feeling...

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Fri, 16 Nov 2007 00:37:35 GMT
View Forum Message <> Reply to Message

File Attachments
1) tht1.gif, downloaded 1527 times

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Fri, 16 Nov 2007 00:40:44 GMT
View Forum Message <> Reply to Message

Just think: the most important thing as finding an answer is packed into 1/5 of window filled with ugly small uniform text.

Why? Or, is it OK for programmers to find anything in a hard way?

File Attachments
1) tht2.gif, downloaded 1465 times

File   Edit   View   Tools   Window   Help

Back | How Do I   ▾   Search

Contents   ▾   ⤓   ×

Filtered by:

(unfiltered)

- ⊞ Mobile and Embedded Development
- ⊞ .NET Development
- ⊞ Office Solutions Development
- ⊞ Web Development
- ⊟ Win32 and COM Development
  - ⊞ Administration and Management
  - ⊞ Component Development
  - ⊞ Data Access and Storage
  - ⊞ Development Guides
  - ⊟ Graphics and Multimedia
    - ⊞ Audio and Video
    - ⊞ DirectX
    - ⊟ GDI
      - ⊟ Windows GDI
        - Security Considerations: Microsoft Windows
        - ⊞ Bitmaps
        - ⊞ Brushes   M$ "Text Monster"
        - ⊞ Clipping
        - ⊞ Colors   is really a
        - ⊞ Coordinate Spaces and Transformations
        - ⊞ Device Contexts
        - ⊞ Filled   **Totally**
        - ⊞ Fonts and Text   **Unguidable**
        - ⊞ Lines and Curves   **and**
        - ⊞ Metafiles   **Unreadable**
        - ⊞ Multiple Display Monitors **Crap!**
        - ⊞ Painting and Drawing
        - ⊞ Paths
        - ⊞ Pens
        - ⊞ Printing and Print Spooler
        - ⊞ Rectangles
        - ⊞ Regions
      - ⊞ Technical Articles
    - ⊞ GDI+
    - ⊞ Image Color Management (ICM)
    - ⊞ Open GL
    - ⊞ Still Image
    - ⊞ Windows Image Acquisition
    - ⊞ Windows Multimedia
  - ⊞ Messaging and Collaboration
  - ⊞ Networking
  - ⊟ Platform SDK
    - ⊞ Introduction
    - ⊞ What's New in the SDK
    - Platform SDK Documentation Contents
    - ⊟ Using the Platform SDK
      - Installing the Platform SDK with Visual Studio
      - Building the SDK Samples
      - ⊞ Building Applications Using Win32.mak

◄ ▐ ▌ ►

🔖 Contents   📖 Index   ⭐ Help Favorites

Ready

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Fri, 16 Nov 2007 00:42:00 GMT

View Forum Message <> Reply to Message

File Attachments

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Fri, 16 Nov 2007 00:54:20 GMT

File Attachments
1) tht4.gif, downloaded 1494 times

Google™ Web Search Help Center

Google Help > Web Search Help

Google Home

About Google

Web Search Help

Welcome to Web Search Help. Check out our Top 5 Questions, search using keywords, or browse an
Just getting started? You may be interested in our glossary or guided introduction to Google.

Using Web Search

- Search tips
- Features and tools
- Preferences, settings, and languages
- Connecting to Google

Search Guides
  Basics of Search
  Advanced Search
  Search Results Page
  Setting Preferences

Personalizing iGoogle

- iGoogle info
- iGoogle instructions and tips
- iGoogle problem solving

Search Features
Spell checker, calculator
& more

Information in Our Search Results

- How Google finds and ranks pages
- Concerns about content in our index
- Unexpected results

Services & Tools
Desktop Search,
Toolbar, News & more

More About Google

- Policies and principles
- Fun facts
- Corporate info

There`s no need to reinvent the whee

We all see good structured manuals a
everyday. See throughout the years.

In sites with adequate and usable des

Subject: Re: Documentation and Topic++
Posted by mirek on Fri, 16 Nov 2007 11:05:06 GMT

Well, frankly, if I am about to find info about Win32 API, I always use google

IMO, the most useful thing to evolve in help is therefore such full-text search navigation.

I know that the tree sucks, anyway, it is there as the "low-level" navigational tool. I guess it should
be tabbed with fulltext search or something like that.

Plus, perhaps index documents should really behave as introduction/guide/sumarry pages.

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Tue, 04 Dec 2007 15:21:41 GMT

Excuse my little vacation.
Mirek, I think that at least a part of documentation is to be done as some guide to writing specific applications. M$ tried to slve this problem by adding "How do I" topics, but their relization made them totally unusable.

So I propose some kind of well-structured manual, as you could see in google page above.

For example:
GUI
-One-window applications
--Applications with main dialog window

OR:
CORE
-Working with collections of objects
--NTL containers
---How and where to use Vector class

How you think? Is it good solution?
Of course it is tabbed with fulltext search.

P.S. Message #13000

Subject: Re: Documentation and Topic++
Posted by mirek on Tue, 04 Dec 2007 16:10:44 GMT

Mindtraveller wrote on Tue, 04 December 2007 10:21Excuse my little vacation.
Mirek, I think that at least a part of documentation is to be done as some guide to writing specific applications. M$ tried to slve this problem by adding "How do I" topics, but their relization made them totally unusable.

So I propose some kind of well-structured manual, as you could see in google page above.

For example:
GUI
-One-window applications
--Applications with main dialog window

OR:
CORE
-Working with collections of objects
--NTL containers
---How and where to use Vector class

How you think? Is it good solution?
Of course it is tabbed with fulltext search.

P.S. Message #13000

Well, that should be the role of these "tutorials" in getting started.

Anyway, my plan was and is: Better some documentation than none. Priority is to get reference documentation and "fast paced" tutorials. When there is a good mass of documentation available, consider better structuring.

Sort of bottom-up - experiment - refactor approach we like to use when programming U++/with U++...

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Tue, 04 Dec 2007 18:25:20 GMT
View Forum Message <> Reply to Message

What do you mean by "fast paced tutorials"?

Subject: Re: Documentation and Topic++
Posted by mirek on Tue, 04 Dec 2007 22:32:48 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Tue, 04 December 2007 13:25What do you mean by "fast paced tutorials"?

Well, it is more about my general view of how I like to learnin things.

Usually, if am about to learn some programming issue or API, I like to see the code and minimum explanation. That is why I have introduced those reference examples and that is why tutorials contain a lot of code and relatively low amount of explanation.

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Wed, 05 Dec 2007 17:19:16 GMT

Ok.
I suggest having 2 types of docs: autogenerated from sources and manual. Manual articles may hypelink to source-autogenerated articles. All these docs are accessible by 3 tabs:

1. Index
indexed source-autogenerated articles;

2. Manual
manual docs, something like site with hyperlinks;

3. Search
full-text search within source and manual docs;
user may check areas for search: source, manual, or both.

Only tabs, no magic words. Plain and simple.

Is it OK, any corrections?

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 05 Dec 2007 18:18:41 GMT

Mindtraveller wrote on Wed, 05 December 2007 12:19Ok.
I suggest having 2 types of docs: autogenerated from sources and manual. Manual articles may hypelink to source-autogenerated articles. All these docs are accessible by 3 tabs:

1. Index
indexed source-autogenerated articles;

2. Manual
manual docs, something like site with hyperlinks;

3. Search
full-text search within source and manual docs;
user may check areas for search: source, manual, or both.

Only tabs, no magic words. Plain and simple.

Is it OK, any corrections?

Sure, but I am not sure whether something like such plan is very important right now.

I mean, yes, we plan to produce more docs. We plan to implement better ways how to access it.

IMO, both 1. and 3. are already "initiated".

Mirek

Subject: Re: Documentation and Topic++
Posted by Novo on Thu, 06 Dec 2007 19:05:07 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Wed, 05 December 2007 12:19

Is it OK, any corrections?

Documentation in source code would be very handy.

One line per method (at least).

Subject: Re: Documentation and Topic++
Posted by zsolt on Thu, 06 Dec 2007 20:05:05 GMT
View Forum Message <> Reply to Message

Novo wrote on Thu, 06 December 2007 20:05Mindtraveller wrote on Wed, 05 December 2007 12:19

Is it OK, any corrections?

Documentation in source code would be very handy.

One line per method (at least).

Current (separated) type of reference docs is very useful, I think. Some time ago I prefered doxygen docs of sources, but now, I don't care about it. The only problem is this tabbed help. Previous (windowed) version was better for me (using Alt+TAB is more easy, than clicking mouse). So I'm using online HTML docs currently.

Maybe it would be useful, to create a separate help viewer window or application with an "Edit" button, to allow editing topics, like in a wiki.

Subject: Re: Documentation and Topic++

Posted by mirek on Thu, 06 Dec 2007 20:10:19 GMT

Novo wrote on Thu, 06 December 2007 14:05Mindtraveller wrote on Wed, 05 December 2007 12:19

Is it OK, any corrections?

Documentation in source code would be very handy.

One line per method (at least).

Planned de facto equivalent of what you want.

Mirek

Subject: Re: Documentation and Topic++
Posted by Novo on Thu, 06 Dec 2007 20:49:33 GMT

luzr wrote on Thu, 06 December 2007 15:10Novo wrote on Thu, 06 December 2007 14:05Mindtraveller wrote on Wed, 05 December 2007 12:19

Is it OK, any corrections?

Documentation in source code would be very handy.

One line per method (at least).

Planned de facto equivalent of what you want.

Mirek

Do you mean "indexed source-autogenerated articles accessed via Topic++"?

No real documentation in source code, documentation is dynamically generated by Topic++?

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Wed, 12 Dec 2007 22:20:56 GMT

Writing "manual" help pages in current TheIDE version, what package should I add them to? I`d like these articles in the future to be "standalone" help articles accessible through "Manual" tab of new version of help system. Of course, after approval and changing of U++ authors.

For now, I`ve made different .tpp file, but didn`t find the way to open it not in view mode (not edit) and don`t know if it will be possible to move these articles to "manual" help mode.

Subject: Re: Documentation and Topic++
Posted by mirek on Thu, 13 Dec 2007 17:27:38 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Wed, 12 December 2007 17:20
For now, I`ve made different .tpp file, but didn`t find the way to open it not in view mode (not edit) and don`t know if it will be possible to move these articles to "manual" help mode.

I do understand "open it not in view mode (not edit)" part..

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Fri, 14 Dec 2007 04:57:40 GMT
View Forum Message <> Reply to Message

I misspelled. That should read as: "in view mode, not in edit".

Subject: Re: Documentation and Topic++
Posted by mirek on Fri, 14 Dec 2007 16:17:06 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Thu, 13 December 2007 23:57I misspelled. That should read as: "in view mode, not in edit".

In that case, it should not matter. As soon as you place it to src or srcdoc group, you should be able to see it in view mode. (somewhere in the tree...)

Mirek

Subject: Re: Documentation and Topic++
Posted by mrjt on Mon, 14 Apr 2008 09:27:58 GMT
View Forum Message <> Reply to Message

Is there currently any way to have TheIDE create a structured src documentation file for a

---

class/package. i.e. a file with all the methods and parameters listed in Upp style with only the comments missing?

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 14 Apr 2008 12:26:13 GMT
View Forum Message <> Reply to Message

mrjt wrote on Mon, 14 April 2008 05:27Is there currently any way to have TheIDE create a structured src documentation file for a class/package. i.e. a file with all the methods and parameters listed in Upp style with only the comments missing?

Yes , if I understand well the question...:

http://www.ultimatepp.org/appideTopic$en-us.html

Mirek

Subject: Re: Documentation and Topic++
Posted by mrjt on Mon, 14 Apr 2008 12:40:09 GMT
View Forum Message <> Reply to Message

Perfect! Thanks.

Subject: Re: Documentation and Topic++
Posted by cbpporter on Tue, 10 Jun 2008 15:13:05 GMT
View Forum Message <> Reply to Message

Since we are getting pretty close to 2008 release, do you think it's time to reopen the discussion about a new Topic++ and documentation? Do you think this is going to be a focus area before the release of the first 8xx dev?

Subject: Re: Documentation and Topic++
Posted by mirek on Tue, 10 Jun 2008 16:35:17 GMT
View Forum Message <> Reply to Message

Well, yes, for me, assist++ and topic++ are now in focus

Mirek

## Subject: Re: Documentation and Topic++

Posted by [cbpporter](#) on Thu, 12 Jun 2008 11:02:51 GMT

Well, I propose something like this:

What can not be seen from this mock-up, is that all parts except the class description, the see also section and the logical grouping of methods should be auto generated. Also future auto generatations would not destroy available documentation.

File Attachments

1) untitled2.PNG, downloaded 1074 times
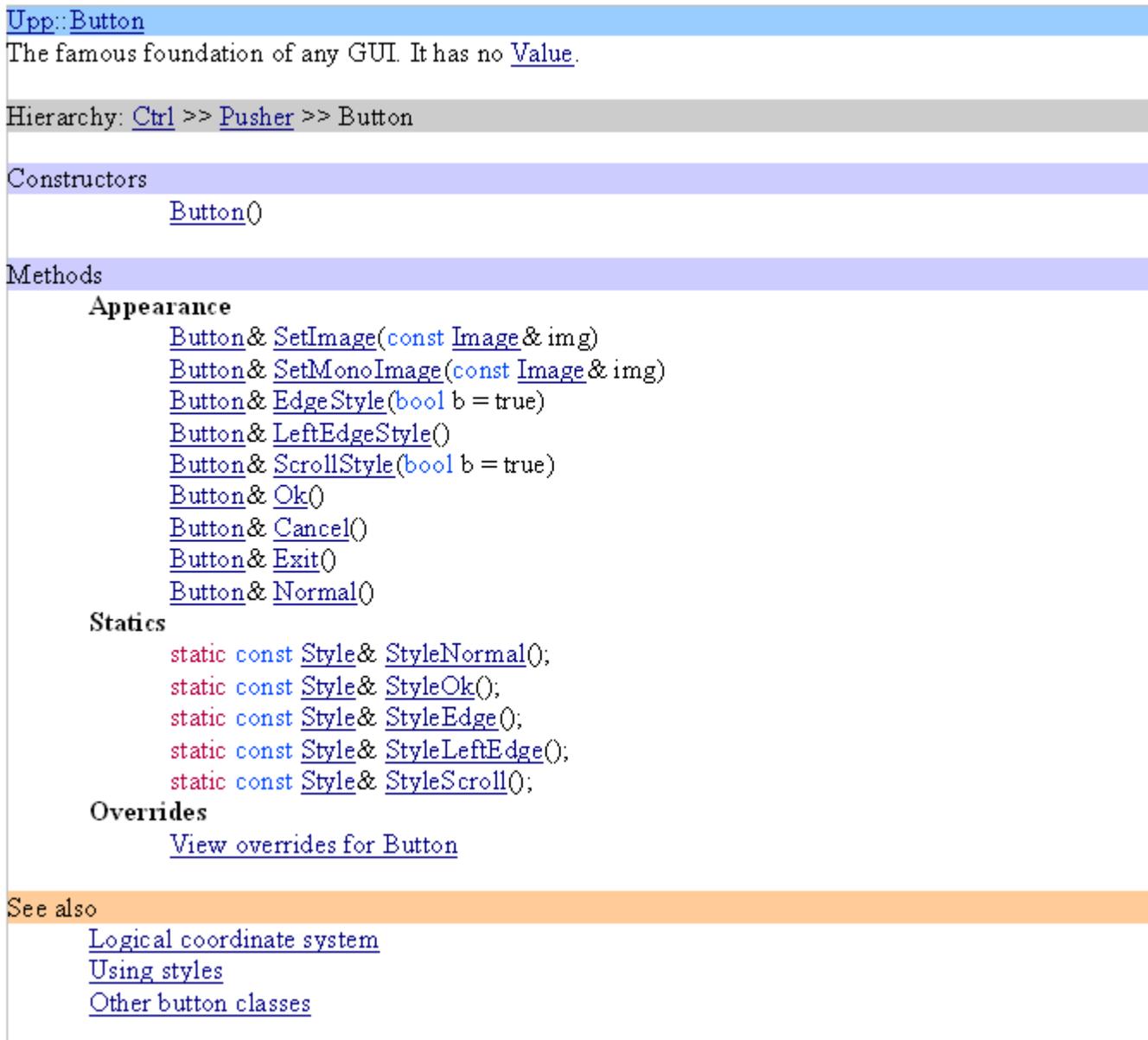
Upp::Button

The famous foundation of any GUI. It has no Value.

Hierarchy: Ctrl >> Pusher >> Button

Constructors

Button()

Methods

Appearance

Button& SetImage(const Image& img)
Button& SetMonoImage(const Image& img)
Button& EdgeStyle(bool b = true)
Button& LeftEdgeStyle()
Button& ScrollStyle(bool b = true)
Button& Ok()
Button& Cancel()
Button& Exit()
Button& Normal()

Statics

static const Style& StyleNormal();
static const Style& StyleOk();
static const Style& StyleEdge();
static const Style& StyleLeftEdge();
static const Style& StyleScroll();

Overrides

View overrides for Button

See also

Logical coordinate system
Using styles
Other button classes

Subject: Re: Documentation and Topic++
Posted by mirek on Thu, 12 Jun 2008 15:36:50 GMT

Well, this sounds good - for class browser.

Anyway, the goal I am currently pursuing is to show topic++ related to regular code editing. The idea is that in the left bar area, where breakpoints are, all lines with items with existing or even *potential* Topic++ entry will have an icon. Moving the mouse over that will display tooltip (rather big one with help, clicking it will move you to the entry in Topic++ - if does not exist, it will create one.

Another issue is related to the need of community provided docs, so that we will finally start moving here... I guess it it will be very useful to be able to split source tree and topic tree (as option), so that all U++ docs can be kept and maintained separately (I think tha topic++ repository can have more relaxed access).

Mirek

Subject: Re: Documentation and Topic++
Posted by zsolt on Thu, 12 Jun 2008 18:25:38 GMT

luzr wrote on Thu, 12 June 2008 17:36Well, this sounds good - for class browser.

Anyway, the goal I am currently pursuing is to show topic++ related to regular code editing. The idea is that in the left bar area, where breakpoints are, all lines with items with existing or even *potential* Topic++ entry will have an icon. Moving the mouse over that will display tooltip (rather big one with help, clicking it will move you to the entry in Topic++ - if does not exist, it will create one.

Another issue is related to the need of community provided docs, so that we will finally start moving here... I guess it it will be very useful to be able to split source tree and topic tree (as option), so that all U++ docs can be kept and maintained separately (I think tha topic++ repository can have more relaxed access).

Mirek
I think, this is a very very good idea.

Subject: Re: Documentation and Topic++
Posted by cbpporter on Thu, 12 Jun 2008 20:46:42 GMT

luzr wrote on Thu, 12 June 2008 18:36Well, this sounds good - for class browser.

Well this was more of an idea for the main documentation presentation, a replacement for the

current structure of the Topic++ articles.

This is something of a solution to what I perceive as the tree main disadvantages of the current system:
1. It is far from complete (as in coverage of the API) and quite selective. Some parts are explained in great detail, while others are barely mentioned. It is also not uniform, neither as formating or style.
2. It is not that cross referenced. For example, when I'm browsing the SetStyle function for a class, I would like to have the definition of that style and a srcdoc style article related to what is Chameleon and how to use it, at my fingertips, both only a click way. When I'm browsing the documentation for String, I want to have a list will all the stand alone functions that work on string, preferably ordered by category. For example, I had often had a self written Join function, before I found out that there was one already implemented.
3. The current documentation for classes are hard to read because the text follows every method. This is daunting for someone new to the framework, who is trying to find something and has at least to skim through all that text. And is annoying for someone familiar with the framework, because these persons really don't need to have text displayed for the methods they already know.

But my idea and your approach are anyway not auto-exclusive. Both can use the same information, only the presentation is different. And the analogy with a browser is logical, since I was thinking of extending a browser window with an editor to enter a description for each item and a category for it. This meta information would them be used for the auto generated pages and could also allow atomic commits to that repository you talked about. When someone is using the normal browser, an icon could show that it has no documentation attached, and somebody might be willing to fill it out, submit it and not have to worry about formating or ruining something in a larger document.

Anyway, my ideas are heavily influenced by three really good help systems (IMHO): Delphi help files, PHP documentation and JavaDoc.

Subject: Re: Documentation and Topic++
Posted by mrjt on Thu, 12 Jun 2008 21:35:22 GMT
View Forum Message <> Reply to Message

I'll contribute some documenting once you guys have figured out how it's going to work

My only suggestion is that instead of just saying 'submit some documentation', what would make contributing easier is a list of all (or many) of the things (classes, features, genrenal areas whatever) that need docs. That way people could look at the list and say to themselves 'I know about that, I'll write some docs', rather than not knowing where to start. IMHO obviously, but it's one of the things that has stopped me so far.

Subject: Re: Documentation and Topic++

Posted by mirek on Fri, 13 Jun 2008 07:36:28 GMT

mrjt wrote on Thu, 12 June 2008 17:35I'll contribute some documenting once you guys have figured out how it's going to work

My only suggestion is that instead of just saying 'submit some documentation', what would make contributing easier is a list of all (or many) of the things (classes, features, genrenal areas whatever) that need docs.

Actually, the main point of my above idea is that you will see what methods are not documented while editing the code.. (with different icon in the left bar).

Mirek

Subject: Re: Documentation and Topic++
Posted by mirek on Fri, 13 Jun 2008 07:45:32 GMT

cbpporter wrote on Thu, 12 June 2008 16:46luzr wrote on Thu, 12 June 2008 18:36Well, this sounds good - for class browser.

Well this was more of an idea for the main documentation presentation, a replacement for the current structure of the Topic++ articles.

This is something of a solution to what I perceive as the tree main disadvantages of the current system:
1. It is far from complete (as in coverage of the API) and quite selective. Some parts are explained in great detail, while others are barely mentioned. It is also not uniform, neither as formating or style.
2. It is not that cross referenced. For example, when I'm browsing the SetStyle function for a class, I would like to have the definition of that style and a srcdoc style article related to what is Chameleon and how to use it, at my fingertips, both only a click way. When I'm browsing the documentation for String, I want to have a list will all the stand alone functions that work on string, preferably ordered by category. For example, I had often had a self written Join function, before I found out that there was one already implemented.
3. The current documentation for classes are hard to read because the text follows every method. This is daunting for someone new to the framework, who is trying to find something and has at least to skim through all that text. And is annoying for someone familiar with the framework, because these persons really don't need to have text displayed for the methods they already know.

But my idea and your approach are anyway not auto-exclusive. Both can use the same information, only the presentation is different. And the analogy with a browser is logical, since I was thinking of extending a browser window with an editor to enter a description for each item and a category for it. This meta information would them be used for the auto generated pages and

could also allow atomic commits to that repository you talked about. When someone is using the normal browser, an icon could show that it has no documentation attached, and somebody might be willing to fill it out, submit it and not have to worry about formating or ruining something in a larger document.

Anyway, my ideas are heavily influenced by three really good help systems (IMHO): Delphi help files, PHP documentation and JavaDoc.

Well, I have to say that I have had "internal fight" between the two concepts for years - I mean, more database like concept of documentation as you suggest, where each method has its own entry, and document like approach like we have now.

In fact, a couple of years before, the first Topic++ iteration worked like database.

In the end, I prefer current model because of several tiny advantages (if they are advantages):

- you have little bit more flexible ways to organize docs. E.g. quite often it is nice to document a group of methods with single description.

- you have all formating capabilities of RichEdit, so you can put in pictures etc (well, this might be possible with some DB schemes as well I guess)

- and I think you are not really loosing the possiblity to present documentation in class browser JUST LIKE YOU SUGGEST - the topics++ are marked with code labels, so in fact existing topics can act as database too.

Mirek

P.S.: 2. - that is only a matter of putting these links into the text

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Sun, 15 Jun 2008 12:59:10 GMT
View Forum Message <> Reply to Message

I would like to see more manual-oriented docs. It seems much more effective than simple listing of classes (which is very useful in a number of cases). I like it because knowing all the methods doesn`t guarantee knowing how to use them effectively.
This is especially actual for U++ which introduces innovative and uncommon approach in many aspects of C++ programming.
These two doc types - reference and manual, should be tightly connected and crosss-referenced.
So, agreeing with words above about making big list of all planned classes tipics, I`d add that we should plan the general structure of manual-type docs. That is the thing I mensioned some time ago and now I put it into discussion again.

Subject: Re: Documentation and Topic++
Posted by mdelfede on Sun, 15 Jun 2008 17:03:53 GMT

I agree... IMHO the actual document structure isn't enough.
I'd prefere too a different approach, borland's user guides were very effective and comfortable to read... I think that could be a good example. Good old Microsoft guides in chm format were not bad at all too.

Maybe a good stuff would be to have a good underlying doc structure (maybe xml ?) ,separate viewer/browser AND separated doc editor.
We could even have more viewers, to accomodate user's preferences.

But, in my opinion, documents should be cathegorized, indexed and cross-referenced. Looking to a function/class should provide some "see-also" stuff, some examples, link to one or more main topics and so on.
And we should have cathegorized indexes.

Another good point would be to separate (as it was proposed sometimes) documentation from code, and give a more relaxed access to main repository's docs, allowing users to contribute more effectively.

We could even start a new sf project (with its svn repository) to hold just documentation for upp....

Max

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Sun, 15 Jun 2008 19:26:16 GMT

I think it is time to create topic where we discuss manual docs structure.
Classes and functions reference should be another one parallel topic.

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 16 Jun 2008 11:00:17 GMT

Well, before going further into details, I guess we should consider:

- how much effort we are really able to put into it

- how much of it can be automated. E.g. IMO, that nice mockup posted a while ago can be fully generated using current Topic++ scheme

Subject: Re: Documentation and Topic++
Posted by mdelfede on Mon, 16 Jun 2008 11:58:41 GMT

View Forum Message <> Reply to Message

luzr wrote on Mon, 16 June 2008 13:00Well, before going further into details, I guess we should consider:

- how much effort we are really able to put into it

Well, that depends on how comfortable are instruments to do it.
IMHO in current help engine, most of time is spent formatting documentation rather than documenting code.... Something to just write stuffs and that does automatically (and uniform) formatting would be of great help.

Quote:
- how much of it can be automated. E.g. IMO, that nice mockup posted a while ago can be fully generated using current Topic++ scheme

That depends also on tools. I don't know (I didn't go too in depth on it) if topic++ scheme can store as many information as needed and if it's easy to parse as xml. If so, we could easily separate data structure from viewer.

So, IMHO the question can be also separated in 2 :
1) How much effort can we put in coding new tools ?
2) How much effort can we put in writing docs ?

About point 1, I don't know... For the point 2, I'd be happy to contribute while using upp. But I'd need something to speed up stuffs.

Max

___

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 16 Jun 2008 13:34:38 GMT

View Forum Message <> Reply to Message

mdelfede wrote on Mon, 16 June 2008 07:58luzr wrote on Mon, 16 June 2008 13:00Well, before going further into details, I guess we should consider:

- how much effort we are really able to put into it

Well, that depends on how comfortable are instruments to do it.
IMHO in current help engine, most of time is spent formatting documentation rather than documenting code.... Something to just write stuffs and that does automatically (and uniform) formatting would be of great help.

Quote:
- how much of it can be automated. E.g. IMO, that nice mockup posted a while ago can be fully generated using current Topic++ scheme

That depends also on tools. I don't know (I didn't go too in depth on it) if topic++ scheme can store as many information as needed and if it's easy to parse as xml. If so, we could easily separate data structure from viewer.

Topic++ stores reference points for all documented code entities - code items (methods, variables etc). It is easy to get document and place in the document for specific code item.

This is complemented (or should be with ability to parse C++ and actually get the complete list of all these code items. IMO, with these two systems should give you ability to produce some pretty nice results...

Mirek

Subject: Re: Documentation and Topic++
Posted by cbpporter on Mon, 16 Jun 2008 15:21:35 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 16 June 2008 14:00Well, before going further into details, I guess we should consider:

- how much effort we are really able to put into it

- how much of it can be automated. E.g. IMO, that nice mockup posted a while ago can be fully generated using current Topic++ scheme

I believe that the effort needed to enable Topic++ as a tool to do what we want is not a great one, so I say we should be willing to put all the effort that is needed into it. As you said, Topic++ can be used to generate pages similar to previous mock-ups and others also, so we should be able to do the coding changes in no time and get a tool that has all the issues regarding formating automated, or at least with minimal user input. It is more a question of style.

The more difficult part is actually writing comprehensive documentation for the parts where an API reference is not enough. Here we must ask how much effort are we going to put into it and who will do it. I for one like more to do coding and even just read code (I'm chewing through String and Value classes right now), but am not too fond of writing comprehensive and even witty manual style documentation pages. Still, I do have some ideas for some parts which I could try to put on paper .

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 16 Jun 2008 17:38:59 GMT

Quote:
I believe that the effort needed to enable Topic++ as a tool to do what we want is not a great one, so I say we should be willing to put all the effort that is needed into it.

Well, the effort for T++ is not that high.. however for it to work well, we absolutely need better C++ parser. And THAT is the problem.

Actually, I have restarted the work on parser this weekend. Unfortunately, with so many conflicting demands for the parser, it ain't easy...

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Mon, 16 Jun 2008 18:48:46 GMT

luzr wrote on Mon, 16 June 2008 15:00Well, before going further into details, I guess we should consider:

- how much effort we are really able to put into it

- how much of it can be automated. E.g. IMO, that nice mockup posted a while ago can be fully generated using current Topic++ scheme
I may (and will - if it will be possible) contribute to the manual topics. As well as into the overall manuals structure (which is VERY important IMO). I can`t do it full-time, but I`ll try to lend some evening time for these topics. Because I believe in great importance of good docs in U++ future which is important for me - for now it is my main dev platform.

About an automation. Of course better parsing would be good, but this is more actual for reference docs, not for manuals.
What we really need for simplifying manuals creation process:
* Easy and visual hyperlinkage between pages and page parts (easy to use, support for i18n in 1-2 clicks, etc)
* Predefined template and styles support. Creating new standard-looking page should take seconds and 2-3 clicks
* Better embedded images support - no bluring, easy import (gif, jpg, png, Clipboard w/o bugs)
* Template-predefined hyperlink, headers and text styles.

For creating docs we shouldn`t really need Word functionality. We really don`t need to have a large choice of text, color, etc styles. We should have 1-2 styles for all common elements (2-3 heading levels, illustration, comment, code, hyperlink). We should have ability to set them in 1 click. Adding article to general manuals tree should be intuitive and take 1-2 clicks.

It is much more important than having office-wide rich text editing system. Our task is far away from what secretaries need.

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 16 Jun 2008 18:54:24 GMT
View Forum Message <> Reply to Message

Mindtraveller wrote on Mon, 16 June 2008 14:48

It is much more important than having office-wide rich text editing system. Our task is far away from what secretaries need.

Well, uhm, ok, but if we already have one? Should I start developing another editor only because rich-text is "too much"?

(And in fact, it WAS originaly developed for secretaries In its second life, or maybe in the first, RichEdit is widely used in municipal agenda

Mirek

Subject: Re: Documentation and Topic++
Posted by mdelfede on Mon, 16 Jun 2008 21:12:27 GMT
View Forum Message <> Reply to Message

I still think that the problem is not the richedit, but its usage... It should be used to browse docs (maybe with some addition....) but not to create them.
Creating docs, formatting and linking manually is too time expensive.

Max

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 16 Jun 2008 22:55:41 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 16 June 2008 17:12I still think that the problem is not the richedit, but its usage... It should be used to browse docs (maybe with some addition....) but not to create them.
Creating docs, formatting and linking manually is too time expensive.

Max

So what you WANT to create docs?

Text processor? And putting all formatting into xml tags? Images into separate file?

Sorry, I am still more happy with RichText here.

Mirek

Subject: Re: Documentation and Topic++
Posted by mdelfede on Mon, 16 Jun 2008 23:02:30 GMT

luzr wrote on Tue, 17 June 2008 00:55mdelfede wrote on Mon, 16 June 2008 17:12I still think that the problem is not the richedit, but its usage... It should be used to browse docs (maybe with some addition....) but not to create them.
Creating docs, formatting and linking manually is too time expensive.

Max

So what you WANT to create docs?

Text processor? And putting all formatting into xml tags? Images into separate file?
....

No, just a simple dialog to input docs in a quick way, with no formatting and some predefined fields, which in turn formats data and puts them in whathever you like... which can be also tpp.
I'd never use a text proc to write documents.
But well... that's just my opinion.

Max

Subject: Re: Documentation and Topic++
Posted by mr_ped on Tue, 17 Jun 2008 09:59:16 GMT

Reading about first impressions from UPP in other thread I got idea for yet another type of doc.
A "Quick reference of UPP vs C++", i.e. docs for somebody who's very familiar with common C++ (I'm not even trying to define what is common C++, I think this one will be difficult to solve), so he can quickly check for common pitfalls when using UPP.
Like description of GUI_APP_MAIN, correct way of including packages to project, NTL basics, examples of common main's + inits for applications (but not the easy Hello world stuff, but rather complex things like MT app with GUI + some network + some timer events going on, all together ..
BTW a working example application with complex behavior would be nice too).

And all this focused on things which are UPP specific, with omitting/ignoring the pure C++ stuff.

This may help some C++ coders to switch to UPP more easily.

I'm checking the current Manual pages, and this stuff is pretty much all the available topics there, so having a very bare quick reference linked well with other tutorials would be probably best form for this doc. It should be as short as possible, yet it should make any C++ coder, who's wondering how to make his "hello world" to work in "UPP way", happy.

I'm not sure how much this idea is reasonable, any comments/feels if you can imagine a helpful topic with such content, and what would you expect from it?

Subject: Re: Documentation and Topic++
Posted by mirek on Tue, 17 Jun 2008 11:40:47 GMT

View Forum Message <> Reply to Message

mr_ped wrote on Tue, 17 June 2008 05:59Reading about first impressions from UPP in other thread I got idea for yet another type of doc.
A "Quick reference of UPP vs C++", i.e. docs for somebody who's very familiar with common C++ (I'm not even trying to define what is common C++, I think this one will be difficult to solve), so he can quickly check for common pitfalls when using UPP.
Like description of GUI_APP_MAIN, correct way of including packages to project, NTL basics, examples of common main's + inits for applications (but not the easy Hello world stuff, but rather complex things like MT app with GUI + some network + some timer events going on, all together .. BTW a working example application with complex behavior would be nice too).

And all this focused on things which are UPP specific, with omitting/ignoring the pure C++ stuff.

This may help some C++ coders to switch to UPP more easily.

I'm checking the current Manual pages, and this stuff is pretty much all the available topics there, so having a very bare quick reference linked well with other tutorials would be probably best form for this doc. It should be as short as possible, yet it should make any C++ coder, who's wondering how to make his "hello world" to work in "UPP way", happy.

I'm not sure how much this idea is reasonable, any comments/feels if you can imagine a helpful topic with such content, and what would you expect from it?

Yes. Go for it

BTW, get inspired (cz-cz only):

 http://www.stdout.cz/projekty/2008/5/29/clanky/uvod-do-upp-c ast-1/

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Wed, 18 Jun 2008 07:16:52 GMT

luzr wrote on Mon, 16 June 2008 22:54Mindtraveller wrote on Mon, 16 June 2008 14:48

It is much more important than having office-wide rich text editing system. Our task is far away from what secretaries need.

Well, uhm, ok, but if we already have one? Should I start developing another editor only because rich-text is "too much"?

(And in fact, it WAS originaly developed for secretaries In its second life, or maybe in the first, RichEdit is widely used in municipal agenda

Mirek
Mirek, don`t get me wrong: U++ rich-text editing model is cool. All I want to say now is that for editing docs it would be more efficient to have toolbar like this:

[Common/Text] [Title] [Code] [Comment] ...

One of the buttons in switched on, others are switched off.
You should not change a lot - you just need to replace help editor`s standard formatting control with a multi-button control which is switching current text style with one of predefined types. And that`s all concerning text editing.

Subject: Re: Documentation and Topic++
Posted by mdelfede on Wed, 18 Jun 2008 07:55:48 GMT

Or, even better, a page for each function, or class or whathever.
A dialog with fixed fields for input, like this :

Title : xxxxxxxxxxxxxxxxxxxxxx

Description :
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx

Usage :
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx

Parameters :
xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx

Result :
xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxx

Caveats :
xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx

Example :
xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx

See also :
xxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxx

that's all. Click OK and you have it autoformatted in richedit or whathever you like. A button to insert links would be good also. Quick to enter documentation, not loosing time to format it.
Of course, a button to get it back into dialog for editing.
BTW, I'd prefere a doc page for one item, too... putting 30-40 items with short description in one page is not enough and time consuming to look for info, IMHO.

Max

Subject: Re: Documentation and Topic++
Posted by cbpporter on Wed, 18 Jun 2008 08:30:54 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 16 June 2008 20:38
Well, the effort for T++ is not that high.. however for it to work well, we absolutely need better C++ parser. And THAT is the problem.

Actually, I have restarted the work on parser this weekend. Unfortunately, with so many conflicting demands for the parser, it ain't easy...

Mirek
Well maybe we should take the demands one by one. Since you are already working on it, as far as parsers go, I think it would be better for the first upgrade to improve Assist++ and reduce the number of places where it does not work. If we fix this we can add further support with things needed for Topic++. I know that this will probably delay the work on documentation (but not by a significant amount of time), but I think that right now a better Assist++ would benefit those who already use U++ better than Topic++, and maybe even help attract new people. At times, it behaves a lot better than the one from VS6, which is still the de facto standard in some areas, and having a better parser than that is certainly going to be an asset. And then other times it outright

fails.

As for the RichEdit, I don't think we need to touch it (other than maybe disable a couple of buttons). We just need an interface that allows us to manage the structures that we want to document above the standard control which is used to input the single document pages that are used right now.

PS: What dictionary formats can RichEdit handle?

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 18 Jun 2008 09:20:56 GMT

mdelfede wrote on Wed, 18 June 2008 03:55Or, even better, a page for each function, or class or whathever.
A dialog with fixed fields for input, like this :

Title : xxxxxxxxxxxxxxxxxxxxxx

Description :
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx

Usage :
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx

Parameters :
xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxx

Result :
xxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx

Caveats :
xxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx

Example :
xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx

See also :
xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx

that's all. Click OK and you have it autoformatted in richedit or whathever you like. A button to insert links would be good also. Quick to enter documentation, not loosing time to format it.
Of course, a button to get it back into dialog for editing.
BTW, I'd prefere a doc page for one item, too... putting 30-40 items with short description in one page is not enough and time consuming to look for info, IMHO.

Max

This is too long, if applied to each method.

Mirek

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 18 Jun 2008 09:23:31 GMT
View Forum Message <> Reply to Message

Heh, it is sort of funny how everybody apparently sees priority elsewhere

For me, reference documentation is unavoidable prerequisite to anything else. Hence my fixation on improving C++ parser first...

Also, in the process, I think that the priority is to document things first and care about the "quality" later. Bad docs are better than no docs... (of course, we do not need to get to extremes here

Mirek

Subject: Re: Documentation and Topic++
Posted by mr_ped on Wed, 18 Jun 2008 09:39:32 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 18 June 2008 11:23For me, reference documentation is unavoidable prerequisite to anything else. Hence my fixation on improving C++ parser first...

I completely agree with you.

Subject: Re: Documentation and Topic++
Posted by mdelfede on Wed, 18 Jun 2008 10:05:37 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 18 June 2008 11:20

This is too long, if applied to each method.

Mirek

I'd say that it forces to document extensively each method... which is what we need
BTW, usually when I look for help, I look for a cathegory in some index (graphics, strings, path, threading....), then inside it for the function I need.
So, a good topics "group" index, pointing to a group of related stuffs and so on. The alphabetical index is for reverse search.
Usually my problem with upp is that I don't know the name of a function that I'd like to use.... well, usually I don't ever know it it exists. It's quite less often that I know function name and I search for the purpose... that happens only when I dig inside upp code. Then, I'd like to have also some example of usage... and maybe a link to a reference project using it.

I think that we don't need all the fields filled immediately... that can be done on time while using it. We just need the skeleton to do it.
For example :

```
string
  string::cat()   <short description>
    string::cat()
      long desc    -\
      parameters     \ this is one page
      usage          /
      sample      -/
      .....
    string::Find()  <short desc>
    ............
image
  .....
```

My idea is :
1) Set up skeleton like this
2) Set up a public svn repository
3) When my svn class would be integrated on ide, use it to
   extend documentation.

So, if I need something, and I don't find doc, I can add it on the fly (after searching inside upp code, maybe...).
That would be quite few effort to do like this... and docs could grow fast. After adding, just an <update svn doc> button and that would be available to all.

Your parser stuff it is important but also can be separated from it. It's needed for completion, to update doc skeleton and to quick locate stuffs from code inside docs, but it can be made independent from docs itself.

Ciao

Max

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 18 Jun 2008 11:22:50 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Wed, 18 June 2008 06:05luzr wrote on Wed, 18 June 2008 11:20

This is too long, if applied to each method.

Mirek

I'd say that it forces to document extensively each method... which is what we need
BTW, usually when I look for help, I look for a cathegory in some index (graphics, strings, path, threading....), then inside it for the function I need.

Well, I think some nice fulltext works here pretty well. Actually, if I look for help (about anything), I always use google...

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Wed, 18 Jun 2008 12:42:33 GMT
View Forum Message <> Reply to Message

Fulltext is of course a highly required thing (as well as text search within text page) - this is first approach. But in my opinion fulltext search is rather ineffective in teaching new techniques. It requires different approach - something like manual docs.
First approach is experienced users - they already know HOW to do things but may forget some details from time to time. Second approach is for everyone else - from novice to those almost expert users who may uncover for themselves new parts of library. In U++, knowing methods of class doesn`t guarantee that you use them in a good way. Because, you know, library is very advanced in many ways comparing to mainstream toolkits. Even comparing to QT.
It is like a bying new shiny hi-tech car, where you first should discover how to use it`s abilities, then use it.

Subject: Re: Documentation and Topic++
Posted by mdelfede on Wed, 18 Jun 2008 14:29:57 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 18 June 2008 13:22

Well, I think some nice fulltext works here pretty well. Actually, if I look for help (about anything), I always use google...

Me too, for public APIs and so. But for upp I have to dig inside code... and it's a pity that what I find get lost because we haven't a quick way to add document.

Max

Subject: Re: Documentation and Topic++
Posted by mirek on Wed, 18 Jun 2008 15:43:48 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Wed, 18 June 2008 10:29luzr wrote on Wed, 18 June 2008 13:22

Well, I think some nice fulltext works here pretty well. Actually, if I look for help (about anything), I always use google...

Me too, for public APIs and so. But for upp I have to dig inside code... and it's a pity that what I find get lost because we haven't a quick way to add document.

Max

Well, IMO, we are sort of wasting the time, as all suggested approaches are compatible and valuable.

In fact, if you have noticed, there always was an idea of "src" T++ group for reference and "srcdoc" for tutorials and other docs.

Anyway, if you are about to write those tutorial/conceptual docs, I think there is nothing to prevent you doing so now.

Meanwhile, I think it is my responsibility to aim for parser/browser/reference...

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Thu, 19 Jun 2008 11:26:39 GMT

The idea of converting "src" and "srcdoc" into something more comfortable for user didn`t survive through all these discussions?

Subject: Re: Documentation and Topic++
Posted by mirek on Thu, 19 Jun 2008 13:00:48 GMT

src and srcdoc are just low-level facilities. User does not have to be bothered (unless he wants to).

I guess the more serious issue is T++ modularity - I mean, each package has its T++ themes. That might cause some issues, OTOH I think it has its advantages too...

Mirek

Subject: Re: Documentation and Topic++
Posted by Mindtraveller on Sun, 06 Jul 2008 07:45:15 GMT

luzr wrote on Wed, 18 June 2008 19:43Anyway, if you are about to write those tutorial/conceptual docs, I think there is nothing to prevent you doing so now. Actually there is. For now it is quite slow and complicated process imo. It is funny that the functionality itself is enough, what we need now is a little "service" items for our help engine. They are small and simple but save a lot of time to docs developers.

a)While editing help docs we need additional toolbar with predefined styles like "common text", "comment", "important", "code", etc.
b) Newly created article must be created from template with all main items ("heading", "see also", etc). This is base help page template, which would be much easier to edit than creating new standard page each time from the start.
c) Better links support. For now creating new link is rather complicated process. Ideally this operation should take 2-3 mouse clicks and no more.
d) Adding new article shouldn`t be a nightmare. 1 click to articles tree (newly creating article root item), then 1 click to "Add new article" button from toolbar. This should generate new article with predefined text (see (b)) and predefined name (just "New article"). That`s it.

Subject: Re: Documentation and Topic++
Posted by mirek on Sun, 06 Jul 2008 18:06:01 GMT

Mindtraveller wrote on Sun, 06 July 2008 03:45luzr wrote on Wed, 18 June 2008 19:43Anyway, if you are about to write those tutorial/conceptual docs, I think there is nothing to prevent you doing

---

so now. Actually there is. For now it is quite slow and complicated process imo. It is funny that the functionality itself is enough, what we need now is a little "service" items for our help engine. They are small and simple but save a lot of time to docs developers.

a)While editing help docs we need additional toolbar with predefined styles like "common text", "comment", "important", "code", etc.
b) Newly created article must be created from template with all main items ("heading", "see also", etc). This is base help page template, which would be much easier to edit than creating new standard page each time from the start.
c) Better links support. For now creating new link is rather complicated process. Ideally this operation should take 2-3 mouse clicks and no more.
d) Adding new article shouldn`t be a nightmare. 1 click to articles tree (newly creating article root item), then 1 click to "Add new article" button from toolbar. This should generate new article with predefined text (see (b)) and predefined name (just "New article"). That`s it.

Add c). I believe it actually IS 2-3 mouse clicks now.

Add d). Do not forget that topics are still bound to packages. Creating new article is IMO quite simple from package. In other words, I believe you are gona spend much more time writing the article than creating it either way...

Mirek

Mirek

Subject: Re: Documentation and Topic++
Posted by captainc on Mon, 07 Jul 2008 17:37:28 GMT
View Forum Message <> Reply to Message

One issue I have with topic++ is that when I add custom Styles, it only shows/applies to the current topic. I would love a way to save styles and use them globally across any/all topics for any project. Maybe an import/export feature or global/local setting for the style.

Subject: Re: Documentation and Topic++
Posted by mirek on Mon, 07 Jul 2008 20:15:54 GMT
View Forum Message <> Reply to Message

captainc wrote on Mon, 07 July 2008 13:37One issue I have with topic++ is that when I add custom Styles, it only shows/applies to the current topic. I would love a way to save styles and use them globally across any/all topics for any project. Maybe an import/export feature or global/local setting for the style.

It is there since beginning, via "Stylesheet" ("Store stylesheet", "Apply stylesheet").

Mirek

Subject: Re: Documentation and Topic++
Posted by captainc on Tue, 08 Jul 2008 12:06:57 GMT

View Forum Message <> Reply to Message

Here's what I did...
Clicked the "..." button to open the style manager.
Right-Clicked empty space and clicked "Create New Style"
Used the style in topic.
Created 2nd topic.
Looked for style in list, but its not there.
Went back to 1st topic, selected the style from the list.
Clicked "<" store button, clicked ok to keep the name.
Switched to the second topic.
Looked for style, still not there.

Was this the correct way to go about this?

Note: If I copy and paste text into the new topic that has the new style from the first topic, then the style is imported into the new topic.

On a side note, I'm still getting that pesky mouse highlight text offset problem when I decrease the zoom. Screenshot of this attached...

File Attachments
1) upp-topicpp-highlightproblem.jpg, downloaded 507 times
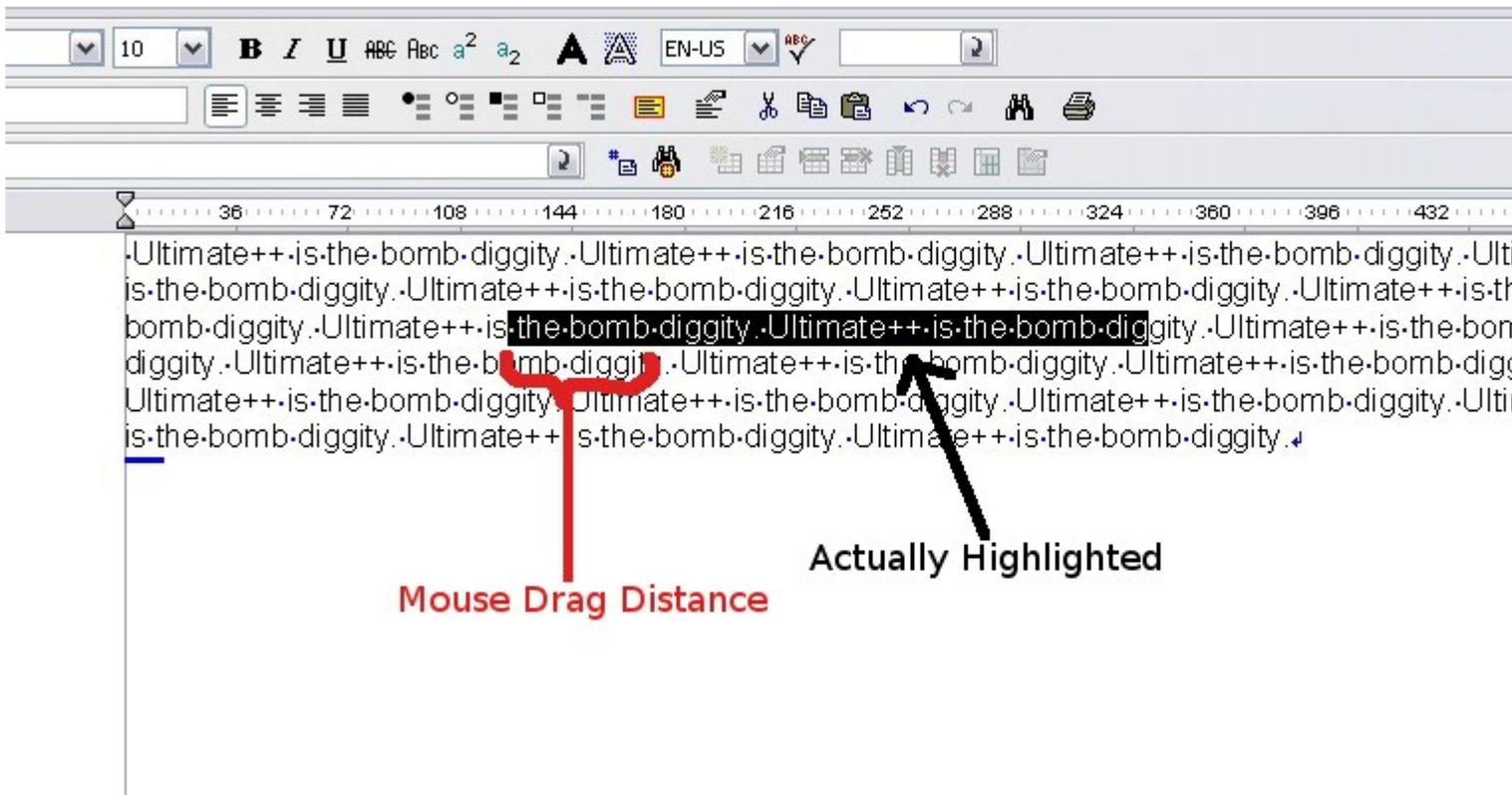
Subject: Re: Documentation and Topic++
Posted by mirek on Tue, 08 Jul 2008 14:11:05 GMT

View Forum Message <> Reply to Message

captainc wrote on Tue, 08 July 2008 08:06Here's what I did...
Clicked the "..." button to open the style manager.
Right-Clicked empty space and clicked "Create New Style"
Used the style in topic.
Created 2nd topic.
Looked for style in list, but its not there.
Went back to 1st topic, selected the style from the list.
Clicked "<" store button, clicked ok to keep the name.
Switched to the second topic.
Looked for style, still not there.

Was this the correct way to go about this?

No. Each document has its own stylesheet. You need to save this stylesheet to global stylesheets
and then choose it when creating the document.

Alternatively, you can also "apply" the stylesheet - adding all global stylesheet styles to current
document.

Mirek

Subject: Re: Documentation and Topic++
Posted by mirek on Tue, 08 Jul 2008 16:11:06 GMT

View Forum Message <> Reply to Message

captainc wrote on Tue, 08 July 2008 08:06
On a side note, I'm still getting that pesky mouse highlight text offset problem when I decrease the
zoom. Screenshot of this attached...

Fixed, thanks.

Mirek