
Subject: Topic browser wishes [FEATURE REQUEST]

Posted by [hojtsy](#) on Fri, 06 Jan 2006 21:42:21 GMT

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

These missing features would be essential for a documentation browser:

- scrolling the text panel by keyboard
 - copying selected text regions from the text panel
 - search on current page
 - search on all pages
 - search by topic name
-

Subject: Re: Topic browser wishes

Posted by [mirek](#) on Fri, 06 Jan 2006 22:05:31 GMT

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

I agree. I will try to fix that ASAP.

Subject: Re: Topic browser wishes

Posted by [hojtsy](#) on Sat, 07 Jan 2006 10:54:11 GMT

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

Could you tell the benefits of using a proprietary tpp format for documentation instead of html? I think html docs would be much more accessible to users. They could choose their favourite web browser, which already has all the fancy features such as for example multiple tabs or mouse gestures. A docs browser could still be provided as an alternative, which has multi-page search.

Subject: Re: Topic browser wishes

Posted by [mirek](#) on Sat, 07 Jan 2006 11:16:47 GMT

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

Well, there of course is a lot of truth there...

However, whole thing begins with the QTF issue. In the beginning, we needed some rich-text format mainly to create reports of our database applications.

Well, real choices back then were:

- HTML
- RTF
- develop the format

Now:

HTML - problem is that HTML lacks complete typography, so it cannot be used there. It is also quite verbose and parsing is not trivial.

RTF - totally unusable if you are going to use it in code to create things (like programmatically create the table). Also, complete implementation is not trivial.

That left us (back then) with "develop the format".

Later we started to use the same format for e.g. dialog boxes (right now, almost any text in U++ GUI is QTF enabled - just prefix it with \1).

Even later we have created wordprocessor package and while it can in fact work with any format, QTF was the natural choice for the default one.

It is non-standard, but does its job very well.

QTF allows using Topic++ for creating application rich-text resources. Topic++ is multi-purpose tool; one of purposes is to create the help for the application or other document resources (like copleft text in the About.. box of TheIDE). .tpp files can be directly compiled into the application, not external files are needed to ship with it (self-containment is one of goals we always tried to achieve with U++). See "reference/Topic" example.

As for HTML, the QTF->HTML conversion is trivial. The whole U++ website is generated by exporting .tpp files and all code documentation topics are there as well, so if you prefer HTML, just read the docs at the website.

Subject: Re: Topic browser wishes
Posted by [hojtsy](#) on Sun, 08 Jan 2006 22:36:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

I have some wishes for the topic++ editor:

- provide the widespread alternative copy-paste keys: shift-del, shift-ins, and ctrl-ins
- multi step undo
- find & replace

These would be a big help for editing the documentation. For example when a single file contains multiple occurrences of the same error, a find & replace would speed up the work of fixing them.

Subject: Re: Topic browser wishes
Posted by [unodgs](#) on Mon, 09 Jan 2006 07:24:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

>provide the widespread alternative copy-paste keys: shift-del, shift-ins, and ctrl-ins

absolutely!

Subject: Re: Topic browser wishes
Posted by [mirek](#) on Mon, 09 Jan 2006 08:38:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

hojtsy wrote on Sun, 08 January 2006 17:36I have some wishes for the topic++ editor:
- provide the widespread alternative copy-paste keys: shift-del, shift-ins, and ctrl-ins

Done - on uvs now (sorry, those we just forgot to add)

Quote:
- multi step undo
- find & replace

These would be a big help for editing the documentation. For example when a single file contains multiple occurrences of the same error, a find & replace would speed up the work of fixing them.

If I understand you well, both are already there.... Undo IS multistep and find&replace is on Ctrl+F. Set selection to activate "replace in selection".

Subject: Re: Topic browser wishes
Posted by [hojtsy](#) on Mon, 09 Jan 2006 08:51:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Sorry, I thought that undo and find & replace is missing. The reason was that there is no menu entry for them. Now I see that toolbar icons are provided, but those are usually duplicated in menus, so I tend to check menus only. Another problem was that I use the Alt-Backspace for undo, which works in most applications, but not in Topic++ editor.

Subject: Re: Topic browser wishes
Posted by [hojtsy](#) on Thu, 12 Jan 2006 20:46:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

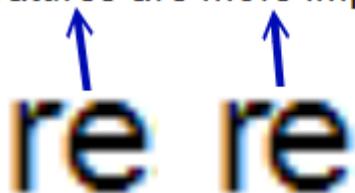
Here is an other problem with the topic++ editor.
The character layout within a line is uneven and very strange to look at. See in the partial screenshot I am attaching, that in the same line the distance between characters are changing. This makes it hard to read the text, because it needs more concentration to identify the real spaces. Could you fix it to make the distances between neighbouring characters visually equal, and not changing on a page?

By the way how do I hide all those dots which are displayed for spaces? And could you please also make Alt-Backspace hotkey for Undo, just like in the code editor?

File Attachments

1) [topic_screenshot2.png](#), downloaded 2579 times

features are more important.



Subject: Re: Topic browser wishes

Posted by [mirek](#) on Thu, 12 Jan 2006 21:42:01 GMT

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

Actually, this is quite hard to fix, this is the result of "true wysiwyg".

The problem is that RichEdit (and RichText in this mode) are printing the text by downscaling it from its physical (printer) appearance. This inevitably leads to "uneven" layout of characters (it could be in theory resolved by subpixel resolution, but that would make the text blurry, which is even worse I think).

Hm, maybe keeping "pixel" distances for words and fix them at spaces.... I will think about it.

(Just a note: Both MS Word and OpenOffice Writer share exactly the same problem. If you are doing wysiwyg, some characters will be one pixel off their ideal positions for given zoom factor).

Alt-Backspace was added on Tuesday (I think). Sync uvs

Subject: Re: Topic browser wishes

Posted by [hojtsy](#) on Thu, 12 Jan 2006 23:08:52 GMT

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

I copied together the same text & font as rendered by 1) RichText, 2) Ms Word, and 3) PhotoImpact.

First I thought that this example will show that RichText is uneven and MS Word is even. Then I noticed that the Ms Word also rendered the distance between "r" and "e" smaller in the "are". Still this seems much less noticeable in Ms Word. For example I don't remember seeing anything in Word like the big distance between "n" and "t" in the "important".

File Attachments

1) [topic_screenshot3.png](#), downloaded 2453 times

features are more important
features are more important
features are more important

Subject: Re: Topic browser wishes
Posted by [hojtsy](#) on Fri, 13 Jan 2006 07:42:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

I investigated further the character spacing. It seems that Word and RichText renders the shape of each character differently given the same font & size. This is independent from the position in the line. The first line was rendered by RichText, the second by Word.

Notice for example that RichText renders the letter "c" narrower than the same curve in "d", which gives an uneven look. Also notice the big space after the letter "n". Since RichText renders the letter "n" narrower there are usually this space on either the left or the right side of the "n"-s. I suppose that the difference could be in the rounding the fraction parts of the font sizes. Maybe if the character sizes would be somehow corrected, the spacing issues would not be that noticable.

File Attachments

1) [topic_screenshot4.png](#), downloaded 2529 times
abcdefghijklmnopqrstuvwxy~~z~~.
abcdefghijklmnopqrstuvwxy~~z~~

Subject: Re: Topic browser wishes
Posted by [mirek](#) on Fri, 13 Jan 2006 08:33:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

Well, I am just using regular font rendering of the platform.

In the example you present, size of fonts are not equal (both Word and RichEdit scale them, so setting both to the same point size does not result in the same screen size). RichEdit scaling just results in smaller text (there are more differences than just "c").

However, it is possible that my downscaling algorithm has quirks. You are welcome to comment, the code is easily accessible, see

`RichText/ParaPaint.cpp`

The actual rescaling is performed around the line 170.
