Subject: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Fri, 06 Mar 2020 20:22:32 GMT
View Forum Message <> Reply to Message

Hello,

I've been thinking about this for a while.

Markdown format has gained a lot of traction in the last few years.

I think U++ and TheIDE can benefit from this (for example, TheIDE or any U++ app using QTF can display markdown formatted text it its intended form, using a MD -> QTF conversion).

To this end, I propose encapsulating the MD4C ("Markdown for C") library.

It does not take much space, comes as a single header and source file, is multiplaform (Win, POSIX), easy to encapsulate, and very fast.

It has MIT license.

Here is the MD4C's github link: https://github.com/mity/md4c

What do you think?

(I can get it ready for U++ 2020.2 release)

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by zsolt on Mon, 09 Mar 2020 03:17:01 GMT
View Forum Message <> Reply to Message

I would be happy with that.

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Xemuth on Mon, 09 Mar 2020 07:43:52 GMT
View Forum Message <> Reply to Message

That's a good idea Thanks

Subject: Re: [Proposal] Markdown to QTF converter/parser

Posted by forlano on Mon, 09 Mar 2020 10:32:16 GMT
View Forum Message <> Reply to Message

Very good!

Luigi

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by mirek on Mon, 09 Mar 2020 15:07:06 GMT
View Forum Message <> Reply to Message

Good, that will be a nice little plugin/

Mirek

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Sat, 11 Jul 2020 15:09:04 GMT
View Forum Message <> Reply to Message

Ok,

I have started implementing the parser. Hopefully the first official version of it will be available around September if not earlier.

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Didier on Sat, 11 Jul 2020 21:10:14 GMT
View Forum Message <> Reply to Message

Nice addition to Upp

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by koldo on Sun, 12 Jul 2020 11:36:49 GMT
View Forum Message <> Reply to Message

That would be great, as we could have the same help files visible in GitHub and U++ web portals, and in U++ apps.

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by mirek on Tue, 14 Jul 2020 08:40:28 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 12 July 2020 13:36That would be great, as we could have the same help files visible in GitHub and U++ web portals, and in U++ apps.

Yeah, well, but we would need QTF->MD for that I guess instead Which is doable too, but harder.

Mirek

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by koldo on Tue, 14 Jul 2020 18:30:26 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 14 July 2020 10:40koldo wrote on Sun, 12 July 2020 13:36That would be great, as we could have the same help files visible in GitHub and U++ web portals, and in U++ apps.

Yeah, well, but we would need QTF->MD for that I guess instead Which is doable too, but harder.

MirekAt least, the MD->QTF is good for the new helps.

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Fri, 04 Sep 2020 22:12:14 GMT
View Forum Message <> Reply to Message

Hi,

A small update and news:

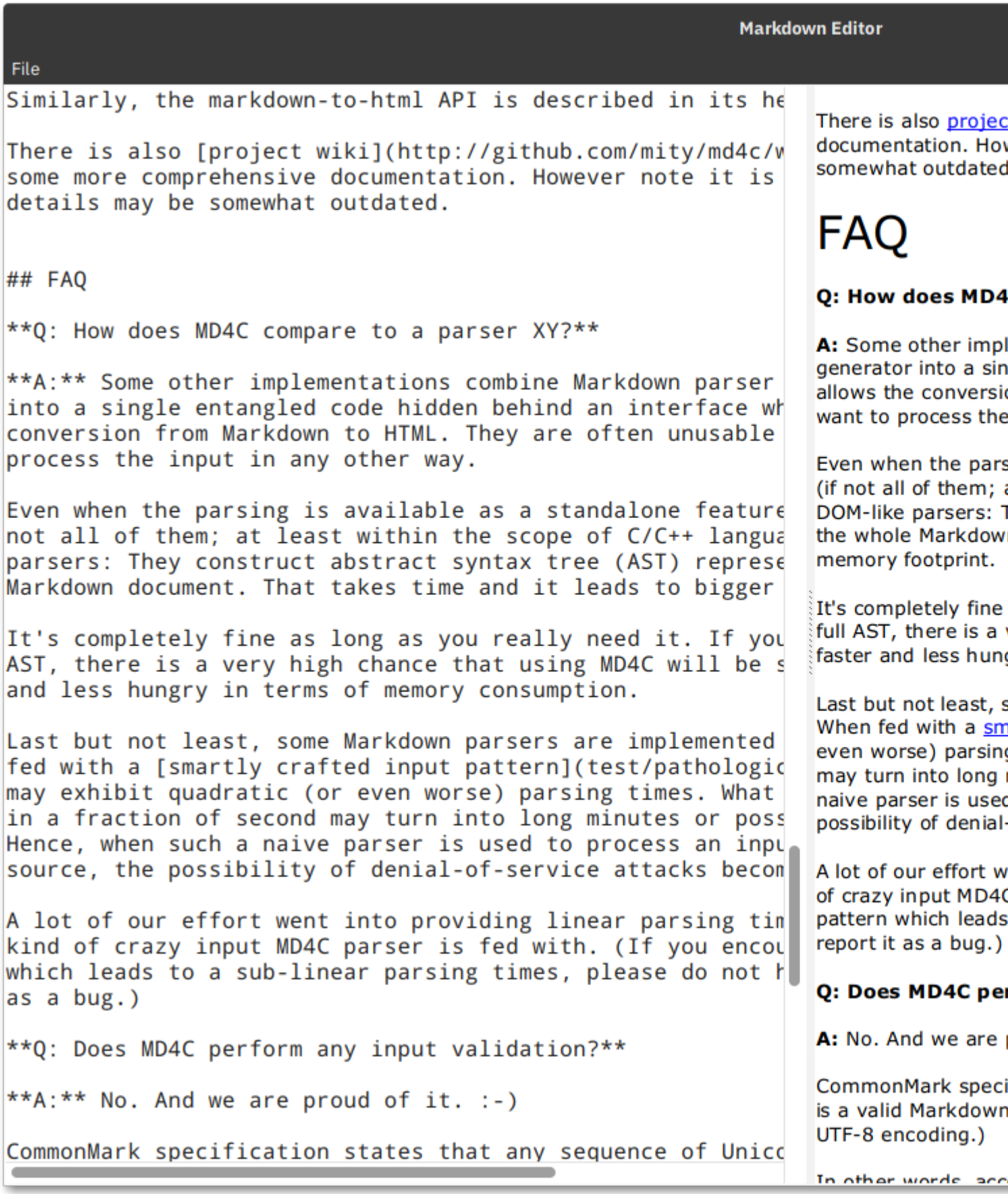
The MD -> QTF converter plugin/wrapper is progressing nicelty.
All the basic stuff (headers, character styles, URL and image links) is done. The more complex stuff (lists & tables) have proven to be somewhat more difficult to convert to qtf. (they can contain more elements than what qtf allows), but there is progress on that side too. So, at least a primitive form of table and list support will hopefully land in first release. Not to mention that I am working on qtf styles supports for md blocks and spans.

A screenshot of the early wersion of -possible- reference example (MarkdownEditor):

Best regards,
Oblivion

File Attachments

1) markdowneditor.png, downloaded 1509 times


Markdown Editor
File
Similarly, the markdown-to-html API is described in its he
There is also [project wiki](http://github.com/mity/md4c/w
some more comprehensive documentation. However note it is
details may be somewhat outdated.
FAQ
Q: How does MD4C compare to a parser XY?
A: Some other implementations combine Markdown parser
into a single entangled code hidden behind an interface wh
conversion from Markdown to HTML. They are often unusable
process the input in any other way.
Even when the parsing is available as a standalone feature
not all of them; at least within the scope of C/C++ langua
parsers: They construct abstract syntax tree (AST) represe
Markdown document. That takes time and it leads to bigger
It's completely fine as long as you really need it. If you
AST, there is a very high chance that using MD4C will be s
and less hungry in terms of memory consumption.
Last but not least, some Markdown parsers are implemented
fed with a [smartly crafted input pattern](test/pathologic
may exhibit quadratic (or even worse) parsing times. What
in a fraction of second may turn into long minutes or poss
Hence, when such a naive parser is used to process an inpu
source, the possibility of denial-of-service attacks becom
A lot of our effort went into providing linear parsing tim
kind of crazy input MD4C parser is fed with. (If you encou
which leads to a sub-linear parsing times, please do not h
as a bug.)
Q: Does MD4C perform any input validation?
A: No. And we are proud of it. :-)
CommonMark specification states that any sequence of Unico
There is also projec
documentation. How
somewhat outdated
FAQ
Q: How does MD4
A: Some other impl
generator into a sin
allows the conversio
want to process the
Even when the pars
(if not all of them; a
DOM-like parsers: T
the whole Markdow
memory footprint.
It's completely fine
full AST, there is a
faster and less hung
Last but not least, s
When fed with a sm
even worse) parsing
may turn into long
naive parser is used
possibility of denial-
A lot of our effort w
of crazy input MD4C
pattern which leads
report it as a bug.)
Q: Does MD4C per
A: No. And we are
CommonMark speci
is a valid Markdown
UTF-8 encoding.)
In other words, acc

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Didier on Sat, 05 Sep 2020 18:19:30 GMT
View Forum Message <> Reply to Message

Starting to look good !

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Sun, 27 Sep 2020 20:47:00 GMT
View Forum Message <> Reply to Message

Hi,

Another small update/nws.

Markdown to qtf converter is progressing nicely.
After some wrestling with "loose" lists, I have managet to map complex DOM model to qtf.

A screenshot:

Unfortunately the package won't be available in the upcoming upp 2020.2 release. But I'll probably send the initial version in couple of weeks.

Best regards,

Oblivion

File Attachments
1) markdownblocks.png, downloaded 1554 times

File

```
## Blockquotes

> Blockquotes can also be nested...
>> ...by using additional greater-than signs right next to each othe
> > > ...or with spaces between arrows.


## Lists

Unordered

+ Create a list by starting a line with `+`, `-`, or `*`
+ Sub-lists are made by indenting 2 spaces:
  - Marker character change forces new list start:
    * Ac tristique libero volutpat at
    + As with other markdown blocks, all list items can contain `COD
    - Nested list items can also contain other markdown blocks, such

    ```
    void main()
    {
        printf("Hello, world!");
    }
    ```
    Cool, isn't it?

+ Very easy!

Ordered

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Integer molestie lorem at massa


1. You can use sequential numbers...
1. ...or keep all the numbers as `1.`
```

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by zsolt on Sun, 27 Sep 2020 21:52:19 GMT
View Forum Message <> Reply to Message

Seems to be very cool! Thanks for working on it!

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by mirek on Fri, 18 Dec 2020 09:54:26 GMT
View Forum Message <> Reply to Message

What is the status? I feel like this would be useful with UppHub.... (we could display github readme in theide)

Even partially working solution would be fine for now.

Mirek

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Fri, 18 Dec 2020 09:59:14 GMT
View Forum Message <> Reply to Message

Quote:What is the status? I feel like this would be useful with UppHub.... (we could display github readme in theide)

Even partially working solution would be fine for now.

Mostyly done, except image loading mechanism, styling api and some cleanup. If it's OK, I'll upload it with the editor/converter example on next monday or tuesday (21-22), after I cleanup the code this weekend.
Then I'll start working on completing the missing parts.

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by mirek on Fri, 18 Dec 2020 10:11:14 GMT
View Forum Message <> Reply to Message

OK, thank you.

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Wed, 23 Dec 2020 19:40:58 GMT
View Forum Message <> Reply to Message

Hello Mirek,

FYI, I have delayed the initial release several more days.
The reason is I couldn't find time to finalize it earlier and I don't want to publish a sphagetti-code just because "it works".
I'm in the process of streamlining it, and I'll release it, this weekend.

I am really sorry for the delay. And thank you for your patience.

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by zsolt on Wed, 23 Dec 2020 21:06:41 GMT
View Forum Message <> Reply to Message

Good news!
It would be useful for me also. I'm almost at the point starting to code my own parser.
BTW, your OAuth code is extremely useful in one of my a projects also. I added some extra to it.
Are you open to get my paches to that?

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Wed, 23 Dec 2020 21:34:39 GMT
View Forum Message <> Reply to Message

Hello zsolt,

Quote:
I'm almost at the point starting to code my own parser.

Don't bother, It is almost ready. You don't want to dive into that hell.

Quote:
BTW, your OAuth code is extremely useful in one of my a projects also. I added some extra to it.
Are you open to get my paches to that?

Sure, I'd be more than happy to enhance the OAuth code and receive contributions (which you will be credited for)!

Best regards,

Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by zsolt on Sat, 26 Dec 2020 00:22:32 GMT
View Forum Message <> Reply to Message

OK, I palan registering to github to be able to send you pull requests. Or is it enough for you if I save my commits to files and send them to you in PM here?
The changes are made to be able to use it with Google in a desktop app, signing in to send emails.

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Sat, 26 Dec 2020 08:32:07 GMT
View Forum Message <> Reply to Message

Hello Zsolt,

Quote:OK, I palan registering to github to be able to send you pull requests. Or is it enough for you if I save my commits to files and send them to you in PM here?
The changes are made to be able to use it with Google in a desktop app, signing in to send emails.

You don't need to sign in to GitHub for that, you can simply upload the modified package via a message to Bazaar OAuth topic: https://www.ultimatepp.org/forums/index.php?t=msg&th=10412&start=0&
(Let's continue there)

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Sun, 27 Dec 2020 19:34:54 GMT
View Forum Message <> Reply to Message

Hello Mirek,

Please find attached the "technical preview version" of Markdown package. (Let's call it that until we fix the remaining issues (before 2021.1 release))
Package contains:

- Markdown package

- Markdown Editor Example

Markdown package contains the latest version of md4c library.
It lacks documentation, for the time being. I will add it in the following days. However, it is so simple to use, documentation is hardly required.

Current issues and todo:

TODO:

- Image handling is yet to be implemented.
- Task lists are yet to be implemented.

Remaining issues:

- Block elements don't have configurable style.
- Page properties (margins, indentation, etc.) are not configurable.
- Soft break is not handled very well

The Markdown editor example is a so-called "real-time" editor in which you can design/test/load/save/export markdown documents in real-time. (it reacts to key-strokes)

It is what I use to test markdown package.

If you can grant me SVN write permissions to package location after adopting the package, I'll fix the remaining issues and maintain the package.

Best regards and a happy, productive new year!

Oblivion

File Attachments

1) Markdown_Package.zip, downloaded 379 times

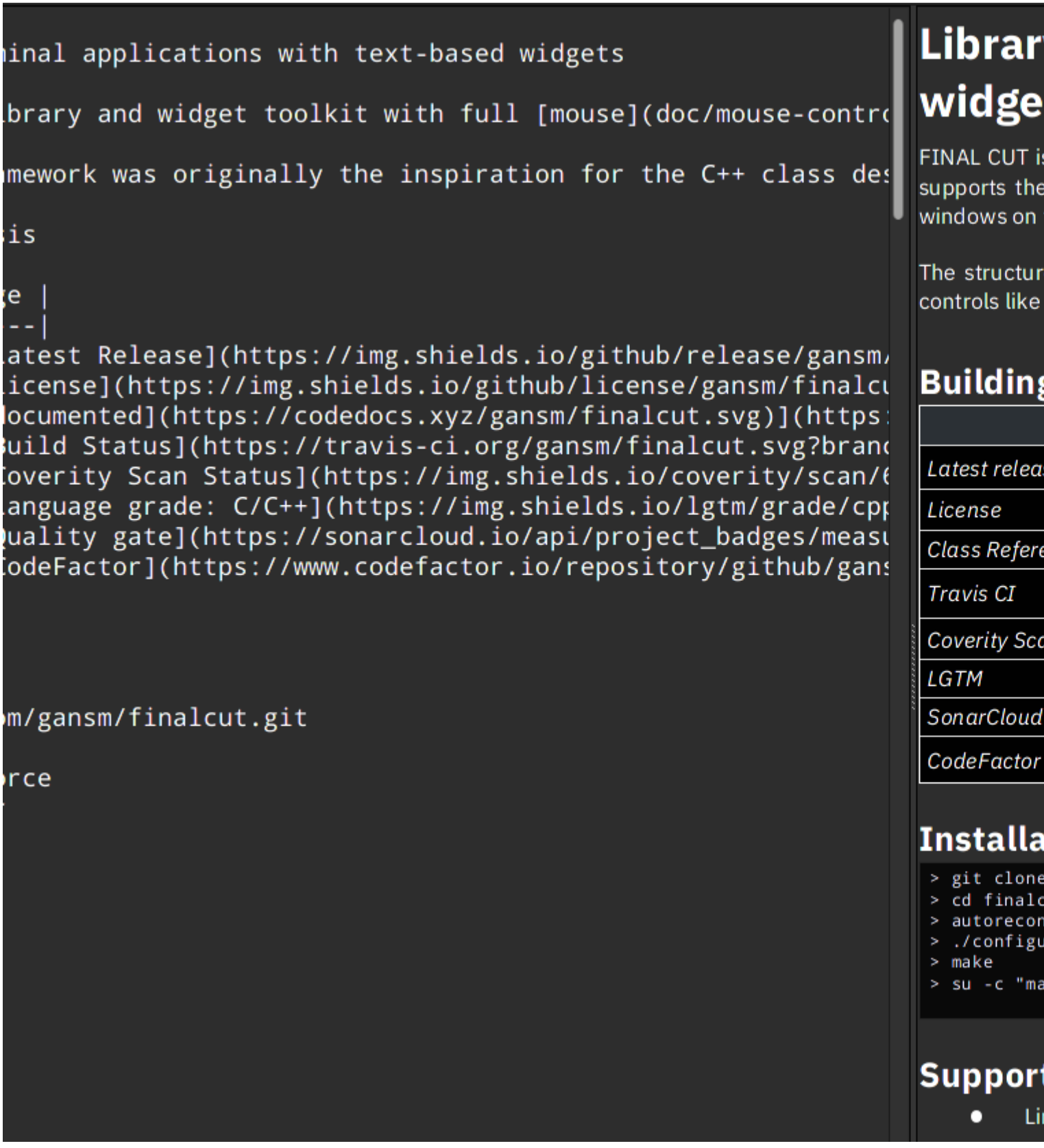
Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Fri, 01 Jan 2021 13:39:30 GMT
View Forum Message <> Reply to Message

A screenshot demonstrating the image handling progress update (This is not avaiable yet):

Best regards,
Oblivion

File Attachments

1) Ekran Görüntüsü - 2021-01-01 16-36-09.png , downloaded 1111 times


inal applications with text-based widgets
brary and widget toolkit with full [mouse](doc/mouse-contr
mework was originally the inspiration for the C++ class des
is
e |
--|
atest Release](https://img.shields.io/github/release/gansm/
icense](https://img.shields.io/github/license/gansm/finalcu
ocumented](https://codedocs.xyz/gansm/finalcut.svg)](https:
uild Status](https://travis-ci.org/gansm/finalcut.svg?branc
overity Scan Status](https://img.shields.io/coverity/scan/6
anguage grade: C/C++](https://img.shields.io/lgtm/grade/cpp
uality gate](https://sonarcloud.io/api/project_badges/measu
odeFactor](https://www.codefactor.io/repository/github/gans
m/gansm/finalcut.git
rce
Librar
widge
FINAL CUT is
supports the
windows on
The structur
controls like
Building
Latest relea
License
Class Refere
Travis CI
Coverity Sca
LGTM
SonarCloud
CodeFactor
Installa
> git clone
> cd finalc
> autorecon
> ./configu
> make
> su -c "ma
Support
Li

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by jjacksonRIAB on Mon, 04 Jan 2021 08:49:53 GMT
View Forum Message <> Reply to Message

Very nice. I did a simple markdown to RichCtrl a year or so ago using PEGTL, so it would be nice to see something more complete.
https://github.com/BornTactical/MarkdownFormatter

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by mirek on Thu, 04 Feb 2021 16:18:26 GMT
View Forum Message <> Reply to Message

It is now used in UppHub to display README.md. Unfortunately, to do that, I needed it in uppsrc (so that theide compiles from uppsrc), so moved it to plugin/md....

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Thu, 04 Feb 2021 16:22:06 GMT
View Forum Message <> Reply to Message

Quote:It is now used in UppHub to display README.md. Unfortunately, to do that, I needed it in uppsrc (so that theide compiles from uppsrc), so moved it to plugin/md....

OK, Do I have commit access? I'll commit the missing bits (styles, at least.) in this weekend.

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by mirek on Thu, 04 Feb 2021 19:44:37 GMT
View Forum Message <> Reply to Message

You should have commit access now.

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Thu, 04 Feb 2021 20:03:44 GMT
View Forum Message <> Reply to Message

OK, thanks!

Best regards,
Oblivion

Subject: Re: [Proposal] Markdown to QTF converter/parser
Posted by Oblivion on Wed, 21 Apr 2021 08:13:59 GMT
View Forum Message <> Reply to Message

Hi,

I have added image span handling to plugin/md. It should be available with U++ nightly builds.

Summary:

U++ md plugin does not render images or recognize any image format. It is up to client code to render and manipulate images. This is done via an event:

Event <VectorMap<String, String>& > WhenImages;

The md plugin will parse/collect the image data of a given md document and hand them to the client code as a string map via the WhenImages event.
The keys of this map are image URLs whereas the values are meant to be filled with some QTF/Richobject data (as qtf).

In this way, the client code can [down]load and manipulate the image data as it sees fit (e.g. it can simply substitute a qtf text on image loading/retrieval failures, or blacklist images from certain domains, or put decorative frames around images, using tables etc...)

This method also allows MT or async calls for image [down]loading and rendering, as it passes to client a complete list of available image span info at once.

Now, the map values can certainly be filled with the "whole" image strings, but this is usually not desireable, as it makes the document larger after a couple of images.
Instead you can use an array of RichObjects to keep the real image data in a cache and only pass the serial/pointer as the map value, using the QtfRichObject() function, for example.

If you have any questions, suggestions etc., let me know.

Best regards,
Oblivion