
Subject: Re: umk static version

Posted by [mirek](#) on Wed, 01 Apr 2020 07:25:11 GMT

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

dolik.rce wrote on Tue, 31 March 2020 21:17mirek wrote on Mon, 30 March 2020 23:28dolik.rce wrote on Mon, 30 March 2020 21:52One thing I'd definitely change about umk is the "UI". Fixed position parameters, no real help, some params prefixed with '-', some with '+', no long options... Explaining this to a common user would be nightmare :) Isn't it time to tweak this a little as well? I'd be willing to do the work. I know it's not hard, but I guess you have more important things to do, right? :)

In any case it would be good to show how do you want to change it before implementing...

I was thinking something like this:Usage:

umk [options] <package> [<output>]

Arguments:

<package> Package name

<output> Output file or directory. If omitted, <package> name will be used.

Options:

-A, --assembly <paths>

Comma separated paths to source directories or path to .var file.

-B, --build-method <build_method>

Build method name or path to .bm file. Default value is 'GCC'.

-f, --flags <FLAGS> Comma separated list of flags. Default value is 'GUI,MT'.

-a, --all Rebuild all.

-b, --blitz Use BLITZ.

-v, --verbose Be verbose.

-l, --silent Silent mode.

-u, --target Use target directory.

-m, --map Create a map file.

-d, --debug=[no|minimal|full]

Select debug mode:

no = release mode, full optimizations (default)

minimal = no debug symbols, no optimizations

full = with debug symbols, no optimizations

-s, --shared Use shared libraries.

-S, --all-shared Use shared libraries and build as shared libraries.

-M, --makefile Create makefile (to file <output>).

-x, --export Export project (to directory <output>), export only files used.

-X, --export-all Export project (to directory <output>), export all files.

-k, --no-delete Do not delete target directory out when exporting.

-H, --threads <N> Build using <N> threads. Default is number of logical cores available.

-h, --help Show this text.

Examples:

Build release version of TheIDE:

```
umk -b -A uppsrc ide ~/theide
```

Use custom build method:

```
umk -b -A uppsrc -B ~/.upp/theide/CLANG.bm ~/theide
```

Rebuild Bombs example in debug mode:

```
umk -ab --debug=full -A examples Bombs
```

Build Bombs example using source paths

```
umk -A upp/examples,upp/uppsrc Bombs
```

What do you think?

Honza

I believe that -A as option does not make much sense. What will umk do if you do not specify -A? Probably the same for -B. I mean, you do not want to write

```
cp --from=sourcefile.txt --to=targetfile.txt
```

either, do you? :)

Also, it would really be good if the new interface was backward compatible, as not to break exiting scripts (not only mine, but I know that there are users that depend on umk). I think making '+' synonyme of '-f', perhaps even hidden, should not cause any problems.

Finally (and that is a problem with current sources too), the commandline parsing should be shared between theide and umk.

