Subject: umk (small) improvements
Posted by mirek on Mon, 13 Apr 2020 16:21:34 GMT
View Forum Message <> Reply to Message

It is now possible to

- provide the name of target Makefile (-M=MyMakefile)
- execute created binary: umk examples,uppsrc -sb Bombs ! params for bombs
- default build method, defaults to CLANG
- options now can be placed anywhere on commandline

Mirek

Subject: Re: umk (small) improvements
Posted by amrein on Mon, 13 Apr 2020 22:26:14 GMT
View Forum Message <> Reply to Message

Took me time to do this but my dreamed umk would be:

NAME
umk - Ultimate++ make utility to maintain programs

SYNOPSIS
umk <assembly> <package> [OPTION] [output]

DESCRIPTION
The umk utility will determine automatically which pieces of a program need to be recompiled, and issue the commands to recompile them.
It can also export all files or only used files to another directory for the selected package.

MAIN PARAMETERS
This parameters are the most used by umk users.

<assembly>
Comma separated paths to source directories (nest<,nest<,..>>) or path to .var file.
A nest is a directory containing one or more packages.

<package>
Package name. It's the same name as the directory containing the '<package>.upp' configuration file.

[output]
The name of the final executable, library or the name of the directory where umk will export project files.
If omitted, <package> name will be used for binary name or in export mode, it will use current

path to export project files.

OPTIONS
umk accepts the following options.

!
Start package executable or build it first if it does not exist.

+FLAG<,FLAG<,..>>
Comma separated list of build flags. Default value is 'GUI,MT'.

--
Signal the end of options. This is useful to allow further arguments to the umk program itself to start with a "-".
This provides consistency with the argument parsing convention used by most other POSIX programs.

-b <build_method>, --build-method=<build_method>
Build method name or path to build method description file (file extension: .bm). If not provided, the default value is 'CLANG'.

-C <path>, --directory=<path>
Change to directory path before doing anything else.

--cache=<path>
Path to cache directory (default ./.cache/u++ if it exists or ~/.cache/u++ otherwise).

--config=<path>
Path to configuration files (default ./.config/u++/ if it exists or ~/.config/u++/ otherwise).

-d[=FLAGS], --debug[=FLAGS]
Print debugging information in addition to normal processing. The debugging information says which files are being considered
for remaking, which file-times are being compared and with what results, which files actually need to be remade, which implicit
rules are considered and which are applied e.g everything interesting about how umk decides what to do. If the FLAGS are omitted,
then the behavior is the same as if -d or --debug was specified. FLAGS may be 'a' for all debugging output, 'b' for basic debugging
(same as using -d), 'v' for more verbose basic debugging, 'i' for showing implicit rules, 'j' for details on invocation of
commands, and 'm' for debugging while making the final result. Use 'n' to disable all previous debugging flags.

-e <variable="value">, --environment-overrides <variable="value">
Overwrite an internal build method value.

-f file, --file=file, --makefile=FILE

Use file as name for exported makefile.

-h, --help
Show this message.

-i, --ignore-errors
Ignore all errors in commands executed to remake files.

-j <jobs>, --jobs=<jobs>
Specifies the number of jobs (commands) to run simultaneously. If there is more than one -j option, the last one is effective. If the
-j option is omitted, umk will run simultaneously as many jobs as available processor logical cores.

-n, --just-print, --dry-run
Print the commands that would be executed, but do not execute them.

-p, --print-data-base
Print the data base (rules and variable values) that results from reading package configuration files (extension: .upp) and the build
method file (extension: .bm); then execute as usual or as otherwise specified.

-q, --question
Return an exit status that is zero if the specified targets are already up to date or nonzero otherwise but do not run any commands,
or print anything.

-s, --silent, --quiet
Silent mode; do not print the commands as they are executed.

-o, --outmode=<mode<mode<..>>>
Mode:
a Clean cache and rebuild all.
b Use BLITZ technology to speed the build process.
c Clean cache.
v Print the pourcentage of avancement.
s Use shared libraries.
S Use shared libraries and build as shared libraries.
m Create a makefile (use the -f option to override the default name 'Makefile').
M Create a map file.
x Export project (to current directory or <output> directory if specified), export only files used by project.
X Export project (to current directory or <output> directory if specified), export all files.
k Do not delete target directory <output> before exporting. The target directory will not be deleted if one of the package
nest is inside the output directory but umk will still print a warning.

-v, --version

Print the version of the umk program and its copyright.

EXAMPLES

Build release version of TheIDE using CLANG build method:
umk uppsrc ide -o rs ./theide

Use custom build method using blitz technology and show progress poucentage:
umk uppsrc ide -b ./CPP.bm -o bsv ./theide

Rebuild Bombs example in full debug mode:
umk examples,uppsrc Bombs --debug=a

Build MyApp using its assembly configuration file if it exist in default u++ config directory:
umk myassembly.var MyApp

EXIT STATUS
umk exits with a status of zero if no targets that were built failed. A status of one will be returned if the -q flag was used and umk determines that a target needs to be rebuilt. A status of two will be returned if any errors were encountered.

Subject: Re: umk (small) improvements
Posted by mirek on Tue, 14 Apr 2020 07:54:19 GMT
View Forum Message <> Reply to Message

amrein wrote on Tue, 14 April 2020 00:26
-C <path>, --directory=<path>
Change to directory path before doing anything else.

Well, given that you have "cd" command, not sure that above is necessarry, HOWEVER what is missing is "cd before executing final binary"

Quote:
--cache=<path>
Path to cache directory (default ./.cache/u++ if it exists or ~/.cache/u++ otherwise).

--config=<path>
Path to configuration files (default ./.config/u++/ if it exists or ~/.config/u++/ otherwise).

Yes, I want to add those as well.

Anyway, I agree about functions to be added, however as you might have noticed, many current users depend on current umk commandline definition, so we cannot break backward compatibility just to have it nicer...

That does not mean we cannot add --xxx commands. But I guess it has less priority than adding functionality.

Mirek

Subject: Re: umk (small) improvements
Posted by amrein on Tue, 14 Apr 2020 11:14:35 GMT
View Forum Message <> Reply to Message

Sure.

They could keep the old umk binary in place and only upgrade their application without touching it.

I will create a pool for fun.
This is a lot of work so I can understand why it's an hard step. I'm still working on the new u++ install process and it takes time too (less, but still).

edit: here is the pool https://www.ultimatepp.org/forums/index.php?t=msg&th=109 91

Subject: Re: umk (small) improvements
Posted by Novo on Tue, 14 Apr 2020 14:48:14 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 14 April 2020 03:54
Anyway, I agree about functions to be added, however as you might have noticed, many current users depend on current umk commandline definition, so we cannot break backward compatibility just to have it nicer...

I personally do not depend on it. I'd love to see a better umk ...
I vote for amrein's proposal.

Subject: Re: umk (small) improvements
Posted by Novo on Wed, 29 Apr 2020 18:31:32 GMT
View Forum Message <> Reply to Message

I've posted a patch to generate Clang Compiler Database using umk.
This allows to use different Clang-based tools with UPP.
