GNU Make Cheat Sheet

RULES

Short rules can be written as below

target(s): [prerequisites] [; shell-command(s)]

target(s): [prerequisites]

[shell-command]

%.class: %.java; javac \$<</pre>

A target pattern is composed of a "%" between a prefix and a suffix, either or both of which may be empty.

MACROS/VARIABLES

Values assigned to variables with = are expanded when used (like a reference) while values assigned with := are expanded at declaration time. \$(myvar) returns the stored value and **\$(call myvar)** executes the value (like a function).

The target member name, when the target is an archive member. \$%

\$< The name of the first (or only) prerequisite.

Space separated list of all prerequisites newer than than the tar-\$?

Space separated list of all prerequisites. \$^ omits duplicate pre-\$^ | \$+

requisites, while \$+ retains them and preserves their order.

The stem with which an implicit rule matches.

\$(•D) | \$(•F) The path and the file name of a macro. For instance **\$(QD)** returns

the directory-part of \$0.

SPECIAL BUILT-IN TARGET NAMES

- .PHONY prerequisites of a .PHONY target are never considered to be up to date.
- .INTERMEDIATE the targets .INTERMEDIATE depends on are treated as intermediate files and they are deleted once no longer needed for a rule.
- .SECONDARY the targets which .SECONDARY depends on are treated as intermediate files but they are not automatically deleted.
- .PRECIOUS like .SECONDARY, and files will not be deleted if make is aborted.
- .IGNORE errors encountered while executing recipes for the prerequisites of . IGNORE are ignored.
- .EXPORT_ALL_VARIABLES allows ables exported in parent Makefile to be available to rules in child processes. It uses no prerequisites.

INCLUDES

For large projects use several makefiles and use the command include to call them within your 'master' makefile. Use -include Makefile(s)

FUNCTIONS

Replaces each occurrence of from in text by to \$(subst from, to, text) \$(patsubst pattern,replacement,text) Replaces words matching pattern with replacement in text.

Removes excess whitespace characters from string.

Search in for an occurrence of find.

Selects words in text that match one of the pattern words.

Selects words in text that do not match any of the pattern words. Sorts the words in list lexicographically, removing duplicates.

Extracts the directory-part of each file name in names. Extracts the non-directory part of each file name in names.

Returns an absolute name (does not contain ".", ".." or symlinks) for each file name in

Extracts the suffix (everything starting with the last period) of each file name in names.

Extracts all but the suffix of each file name in names.

Appends suffix to each word in names.

Prepends prefix to each word in names.

Join two parallel lists of words.

Extracts the n^{th} word of text.

Counts the number of words in text.

Returns the list of words in text from i to j.

Extracts the first word in names.

Returns the file names matching (a shell file name) pattern (not a "%" pattern). When the function is evaluated a fatal error with the message text is generated. When the function is evaluated a warning with the message text is generated.

Execute a shell command and return its output.

Describes where variable came from. Do not use "\$" or parentheses around variable

unless you want the name not to be constant and provide a variable reference. Evaluate text with var bound to each word in words, and concatenate the results. Evaluate var replacing any references to \$(1), \$(2) with the first, second, etc. param

values.

if you don't want make to abort when the included Makefiles are missing. The minus sign generally forces make to ignore errors.

\$(strip string) \$(findstring find,in) \$(filter pattern_1 pattern_2...,text)

\$(filter-out pattern_1 pattern_2...,text)

\$(sort list) \$(dir names...) \$(notdir names...) \$(realpath names...)

\$(suffix names...)

\$(basename names...)

\$(addsuffix suffix,names...) \$(addprefix prefix,names...) \$(join list_1,list_2)

\$(word n,text) \$(words text)

\$(wordlist i,j,text) \$(firstword names...) \$(wildcard pattern...)

\$(error text...) \$(warning text...) \$(shell command) \$(origin variable)

\$(foreach var,words,text) \$(call var,param,param,...)