MAKE(I)   **MAKE(I)**

NAME
        make — make a program

SYNOPSIS
        make [−f descfile] [−p] [−i] [−k] [−s] [−r] [−n] [−t] file ...

DESCRIPTION
Make may be used to mechanize program creation and maintenance, while ensuring that all constituents are current. A graph of dependencies is specified in the descfile(s). The standard input will be read if a descfile named − is given. If no −f options are present, the file named **makefile** or, if absent, the file named **Makefile** in the current directory is used. The −p option prints out a version of that graph. Each file name argument is 'made', as described below.

The file consists of a sequence of entries that describe the prerequisites and operations for creating an object (usually a file). The first line of each entry contains the names of the objects to be made, followed by a colon, optionally followed by a list of other files that must be available and current in order to remake it. Text following a semicolon on the first line, and all immediately following lines that begin with a tab, are fed to the Shell to make the object. Each line is fed to a separate instance of the Shell. All text following a sharp is taken to be a comment. For example:

pgm: x.o y.o ; cc x.o y.o -lp
        mv a.out pgm   # command to be done
x.o: dcls

Make walks the graph of dependencies. If a needed object depends on another that is not present or is younger than itself, it is remade. If an object's name ends in '.o', the description file, and then the current directory, are searched for a corresponding name ending in '.r', '.e', '.f', '.c', '.s', '.l' (Lex), '.y' (Yacc-C), '.yr' (Yacc-Ratfor), '.ye' (Yacc-EFL). (These default rules are not applied if the −r option is specified). If such a file is found and is younger than the object, a compilation command is generated. In the example above, if 'dcls' has been changed since 'pgm' was last made, 'x.c' will be recompiled and 'pgm' will be reloaded.

Each command line is printed before it is executed unless the −s option is specified on the command line or the special name '.SILENT' appears in the description. The command lines are printed but not executed if the −n option is specified. The date last modified is updated but the commands given are not executed for each file if the −t option is specified. (This option is useful when a source change is known to be incremental or benign).

The handling of errors in commands (nonzero status returns) depends on a number of flags. If the −i flag is specified on the command line, if the special name '.IGNORE' appears in the description files, or if the line specifying the command begins with a hyphen, the error is ignored. Normally, *make* aborts upon detecting an error. However, if the −k flag is specified on the command line, *make* ceases issuing commands to update the current node, but continues updating other branches of the graph that do not depend on the failed node.

If an interrupt or quit is detected during the execution of a command, *make* will delete the file being updated unless that file name is a dependent of the special name '.PRECIOUS'.

DIAGNOSTICS
        No description file argument ; Cannot find description file ; Syntax error ; Compilation error

BUGS
        Many UNIX commands return random status, which will cause Make to assume tha the command failed. In case of trouble, use the −i option or a minus sign on the command line.