

NAME

fed — edit associative memory for form letter

SYNOPSIS

fed

DESCRIPTION

Fed is used to edit a form letter associative memory file, **form.m**, which consists of named strings. Commands consist of single letters followed by a list of string names separated by a single space and ending with a new line. The conventions of the Shell with respect to '*' and '?' hold for all commands but **m**. The commands are:

e name ...

Fed writes the string whose name is *name* onto a temporary file and executes *ed*. On exit from the *ed* the temporary file is copied back into the associative memory. Each argument is operated on separately. Be sure to give an *ed w* command (without a filename) to rewrite *fed's* temporary file before quitting out of *ed*.

d [name ...]

deletes a string and its name from the memory. When called with no arguments **d** operates in a verbose mode typing each string name and deleting only if a **y** is typed. A **q** response returns to *fed's* command level. Any other response does nothing.

m name1 name2 ...

(move) changes the name of *name1* to *name2* and removes previous string *name2* if one exists. Several pairs of arguments may be given. Literal strings are expected for the names.

n [name ...]

(names) lists the string names in the memory. If called with the optional arguments, it just lists those requested.

p name ...

prints the contents of the strings with names given by the arguments.

q

returns to the system.

c [**p**] [**f**]

checks the associative memory file for consistency and reports the number of free headers and blocks. The optional arguments do the following:

p causes any unaccounted-for string to be printed.

f fixes broken memories by adding unaccounted-for headers to free storage and removing references to released headers from associative memory.

FILES

/tmp/ftmp?	temporary
form.m	associative memory

SEE ALSO

form(1), ed(1), sh(1)

WARNING

It is legal but unwise to have string names with blanks, ':' or '?' in them.