

NAME

GETFIL – Attach a file to a given unit for a stated purpose.

SYNOPSIS

CALL GETFIL(WATFOR,LW,SUGNAM,LS,NUNIT,ACCESS)

WATFOR(LW)	is a CHARACTER*1 string describing the use of the file
LW	is the INTEGER*4 length of WATFOR
SUGNAM(LS)	is a CHARACTER*1 default filename to use
LS	is the INTEGER*4 length of SUGNAM
NUNIT	is the INTEGER*4 number of the Fortran I/O unit to be attached
ACCESS	is an INTEGER*4 access code: 1 => read, 2 => write, 3 => both

DESCRIPTION

If unit NUNIT is already attached to a file, the routine returns immediately. If unit NUNIT is assigned on the command line by a parameter string of the form "3=fyle", the routine uses the given filename. Otherwise, it constructs a prompt string giving the use of the file and the default name, if any, prompts for the filename, and reads the filename. If an end-of-file (^D) is entered at the filename prompt, the program stops. If a default name is given and a null (carriage return) or blank response is entered at the prompt, the default name is used. If the file is only to be read but it does not exist, an error message is written and the prompt is repeated. If the file is only to be written but it already exists, permission is requested to unlink it first and if denied the prompt is repeated. When the appropriate conditions have been satisfied, the named file is opened for the given unit. If the access code is given a negative sign, the file is attached for unformatted I/O.

UNITS

0	prompts and error messages
5	keyboard responses to prompts
NUNIT	the unit number to which this routine attaches the given file

DIAGNOSTICS

If NUNIT is not in {1,2,3,4,7...99} an error message is written and the program stops. If ACCESS is not in {1,2,3}, an error message is written and the program stops. If ACCESS=1 but the named file does not exist, an error message is written and the prompt is repeated. If ACCESS=2 and the named file already exists, the routine asks if it is ok to make the assignment and if the answer is no the prompt is repeated. If the return code from OPEN is nonzero, it is reported and the prompt is repeated.

BUGS

No check is made for an illegal filename. No provision is made for opening a file to append. No check is made of file permissions.

LINKAGE

gfortran source.f -L\${HOME}/lib -lmisc

AUTHOR

Michael Kupferschmid

EXAMPLE

```
LOGICAL*4 OK
CALL GETFIL('output', 6, 'answer', 6, 1, 2)
INQUIRE (UNIT=1, OPENED=OK)
WRITE(6, 901) OK
901 FORMAT(L1)
WRITE(1, 902)
902 FORMAT('data')
STOP
END
```

This example produces the following exchange when the files "answer" and "snerd" already exist:

```
unix[1] a.out
Name of output file [answer]:
The file is to be written, but it exists; ok? n
Name of output file [answer]: snerd
The file "snerd" is to be written, but it exists; ok? y
T
unix[2] more snerd
data
unix[3]
```

Here the user presses carriage return in response to the first prompt, so GETFIL attempts to attach the file having the default name of "answer". Finding that the file "answer" already exists, it asks for permission to use it anyway (which will empty the file) and the user declines. Then the prompt is repeated and the user enters the alternate file name "snerd". GETFIL finds that this file also exists, so it asks for permission to use it, and the user approves. When GETFIL returns, the program inquires as to the status of unit 1 and reports that it is opened. When the program finishes executing, the file "snerd" contains the word "data".