

# NAG Library Routine Document

## X04ACF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

### 1 Purpose

X04ACF opens a Fortran unit number for reading, writing or appending, and associates the unit with a named file.

### 2 Specification

```
SUBROUTINE X04ACF (IOUNIT, FILE, MODE, IFAIL)
INTEGER          IOUNIT, MODE, IFAIL
CHARACTER(*)    FILE
```

### 3 Description

X04ACF is especially useful if the calling language is not Fortran. It opens a Fortran unit number for reading, writing or appending, and associates the unit with a filename specified by the argument FILE.

### 4 References

None.

### 5 Arguments

- 1: IOUNIT – INTEGER *Input*  
*On entry:* the Fortran unit number which identifies the file to be read from, written to or appended to. Note that this may be system dependent. Values in the range 7 to 1000 should however be safe on most systems.
- 2: FILE – CHARACTER(\*) *Input*  
*On entry:* the name of the file to be opened.  
*Constraint:* must contain a valid filename for the computer system being used.
- 3: MODE – INTEGER *Input*  
*On entry:* specifies whether the file is to be opened for reading, writing or appending.  
MODE = 0  
    The file is to be opened for reading.  
MODE = 1  
    The file is to be opened for writing.  
MODE = 2  
    The file is to be opened for appending.  
*Constraint:*  $0 \leq \text{MODE} \leq 2$ .
- 4: IFAIL – INTEGER *Input/Output*  
*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value  $-1$  or  $1$  is recommended. If the output of error messages is undesirable, then the value  $1$  is recommended. Otherwise, if you are not familiar with this argument, the recommended value is  $0$ . **When the value  $-1$  or  $1$  is used it is essential to test the value of IFAIL on exit.**

*On exit:* IFAIL =  $0$  unless the routine detects an error or a warning has been flagged (see Section 6).

## 6 Error Indicators and Warnings

If on entry IFAIL =  $0$  or  $-1$ , explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL =  $1$

On entry, MODE is invalid.

IFAIL =  $2$

Failure to open the file for reading.

IFAIL =  $3$

Failure to open the file for writing.

IFAIL =  $4$

Failure to open the file for appending.

IFAIL =  $-99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL =  $-399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL =  $-999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

X04ACF is not threaded in any implementation.

## 9 Further Comments

None.

## 10 Example

This example illustrates how to open a file for writing.

### 10.1 Program Text

```

Program x04acfe

!      X04ACF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: x04acf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: iounit = 4, nout = 6
Character (*), Parameter   :: file = 'x04acfe_success.res'
!      .. Local Scalars ..
Integer                    :: ifail, mode
!      .. Executable Statements ..
Write (nout,*) 'X04ACF Example Program Results'

!      Test successful open for write

mode = 1

ifail = 0
Call x04acf(iounit,file,mode,ifail)

Write (nout,99999)
Write (iounit,99999)

99999 Format (' OK file successfully opened for writing')
End Program x04acfe

```

### 10.2 Program Data

None.

### 10.3 Program Results

```

X04ACF Example Program Results
OK file successfully opened for writing

```

---