

# NAG Library Routine Document

## X04ADF

**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

X04ADF closes a file associated with a given Fortran unit number.

### 2 Specification

```
SUBROUTINE X04ADF ( IOUNIT, IFAIL )
```

```
INTEGER IOUNIT, IFAIL
```

### 3 Description

X04ADF is especially useful if the calling language is not Fortran. It closes a file associated with a given Fortran unit number.

### 4 References

None.

### 5 Parameters

1: IOUNIT – INTEGER *Input*

*On entry:* the Fortran unit number which identifies the file to be closed.

2: IFAIL – INTEGER *Input/Output*

*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 parameter, 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

Failure to close the file.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

This example program simply illustrates how to close a file once it has been opened for writing followed by how to close a file once it has been opened for reading.

### 9.1 Program Text

```

Program x04adfe

!      X04ADF Example Program Text

!      Mark 24 Release. NAG Copyright 2012.

!      .. Use Statements ..
      Use nag_library, Only: x04acf, x04adf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: iounit = 4, nout = 6
      Character (*), Parameter    :: fname = 'x04adfe_success.res'
!      .. Local Scalars ..
      Integer                     :: ifail
!      .. Executable Statements ..
      Write (nout,*) 'X04ADF Example Program Results'

!      Test successful open and close for write

      ifail = 0
      Call x04acf(iounit,fname,1,ifail)

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

      ifail = 0
      Call x04adf(iounit,ifail)

      Write (nout,99998)

!      Test successful open and close for read

      ifail = 0
      Call x04acf(iounit,fname,0,ifail)

      Write (nout,99997)

      ifail = 0
      Call x04adf(iounit,ifail)

      Write (nout,99998)

99999 Format (' OK file successfully opened for writing')
99998 Format (' OK file successfully closed')
99997 Format (' OK file successfully opened for reading')
      End Program x04adfe

```

### 9.2 Program Data

None.

### **9.3 Program Results**

```
X04ADF Example Program Results
OK file successfully opened for writing
OK file successfully closed
OK file successfully opened for reading
OK file successfully closed
```

---