NAG Library Routine Document

X04AAF

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

X04AAF returns the value of the current error message unit number, or sets the current error message unit number to a new value.

2 Specification

```
SUBROUTINE X04AAF (IFLAG, NERR)
INTEGER IFLAG, NERR
```

3 Description

X04AAF enables those library routines which output error messages, to determine the number of the output unit to which the error messages are to be sent; in this case X04AAF is called with IFLAG = 0. X04AAF may also be called with IFLAG = 1 to set the unit number to a specified value. Otherwise a default value (stated in the Users' Note for your implementation) is returned.

Records written to this output unit by other library routines are at most 80 characters long (including a line-printer carriage control character).

Note that if the unit number is set < 0, no messages will be output.

4 References

None.

5 Arguments

1: IFLAG – INTEGER *Input*

On entry: the action to be taken (see NERR).

Constraint: IFLAG = 0 or 1.

2: NERR – INTEGER *Input/Output*

On entry: if IFLAG = 0, NERR need not be set.

If IFLAG = 1, NERR must specify the new error message unit number.

On exit: if IFLAG = 0, NERR is set to the current error message unit number.

If IFLAG = 1, NERR is unchanged.

Note that Fortran unit numbers must be positive or zero. If NERR is set < 0, output of error messages is totally suppressed. It is important to note that if you supply an illegal value for NERR (such as a unit number associated with a file opened for reading instead of writing) then X04AAF cannot detect that fact, but any output sent to the unit by later calls of NAG routines may have undesirable consequences, such as program crashes.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

X04AAF is not thread safe and should not be called from a multithreaded user program. Please see Section 3.12.1 in How to Use the NAG Library and its Documentation for more information on thread safety.

X04AAF is not threaded in any implementation.

9 Further Comments

The time taken by X04AAF is negligible.

10 Example

In this example X04AAF is called by your main program to make the error message from the routine DUMMY appear on the same unit as the rest of the output (unit 6). Normally a NAG Library routine with an IFAIL argument (see Section 3.4.2 in How to Use the NAG Library and its Documentation) would take the place of DUMMY.

10.1 Program Text

```
!   X04AAF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module x04aafe_mod

!     X04AAF Example Program Module:
!            Parameters and User-defined Routines

!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                           :: dummy
!     .. Parameters ..
      Integer, Parameter, Public       :: nout = 6
    Contains
      Subroutine dummy

!       .. Use Statements ..
        Use nag_library, Only: x04aaf
!       .. Local Scalars ..
        Integer                        :: nerr
!       .. Executable Statements ..
        Call x04aaf(0,nerr)

        Write (nerr,*)
        Write (nerr,*) 'This is a dummy error message'

        Return

      End Subroutine dummy
    End Module x04aafe_mod
    Program x04aafe

!     X04AAF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: x04aaf
      Use x04aafe_mod, Only: dummy, nout
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Local Scalars ..
      Integer                          :: outchn
!     .. Executable Statements ..
      Write (nout,*) 'X04AAF Example Program Results'

      outchn = nout

      Call x04aaf(1,outchn)

      Call dummy

    End Program x04aafe
```

10.2 Program Data

None.

10.3 Program Results

```
 X04AAF Example Program Results

 This is a dummy error message
```