NAG Library Routine Document

X04ABF

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

X04ABF returns the value of the current advisory message unit number, or sets the current advisory message unit number to a new value.

2 Specification

```
SUBROUTINE X04ABF (IFLAG, NADV)
INTEGER IFLAG, NADV
```

3 Description

X04ABF enables those library routines which output advisory messages to determine the number of the output unit to which the advisory messages are to be sent; in this case X04ABF is called with IFLAG $= 0$. X04ABF 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 120 characters long (including a line-printer carriage control character), unless those library routines allow you to specify longer records.

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 NADV).

Constraint: IFLAG $= 0$ or 1.

2: NADV – INTEGER *Input/Output*

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

If IFLAG $= 1$, NADV must specify the new advisory message unit number.

On exit: if IFLAG $= 0$, NADV is set to the current advisory message unit number.

If IFLAG $= 1$, NADV is unchanged.

Note that Fortran unit numbers must be positive or zero. If NADV is set < 0, output of advisory messages is totally suppressed. It is important to note that if you supply an illegal value for NADV (such as a unit number associated with a file opened for reading instead of writing) then X04ABF 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

X04ABF 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.

X04ABF is not threaded in any implementation.

9 Further Comments

The time taken by X04ABF is negligible.

10 Example

In this example X04ABF is called by your main program to make the advisory 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

```
!   X04ABF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module x04abfe_mod

!     X04ABF Example Program Module:
!            Parameters and User-defined Routines

!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                           :: dummy
!     .. Parameters ..
      Integer, Parameter               :: iget = 0
      Integer, Parameter, Public       :: iset = 1, nout = 6
    Contains
      Subroutine dummy

!       .. Use Statements ..
        Use nag_library, Only: x04abf
!       .. Local Scalars ..
        Integer                        :: nadv
!       .. Executable Statements ..
        Call x04abf(iget,nadv)

        Write (nadv,*)
        Write (nadv,*) 'This is a dummy advisory message'

        Return

      End Subroutine dummy
    End Module x04abfe_mod
    Program x04abfe

!     X04ABF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: x04abf
      Use x04abfe_mod, Only: dummy, iset, nout
!     .. Implicit None Statement ..
```

```
      Implicit None
!     .. Local Scalars ..
      Integer                          :: outchn
!     .. Executable Statements ..
      Write (nout,*) 'X04ABF Example Program Results'

      outchn = nout

      Call x04abf(iset,outchn)

      Call dummy

    End Program x04abfe
```

10.2 Program Data

None.

10.3 Program Results

```
 X04ABF Example Program Results

 This is a dummy advisory message
```