NAG Library Routine Document

P01ABF

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

P01ABF either returns the value of IERROR (soft failure), or terminates execution of the program (hard failure). Diagnostic messages may be output.

2 Specification

```
INTEGER FUNCTION P01ABF(IFAIL, IERROR, SRNAME, NREC, REC)
INTEGER                 IFAIL, IERROR, NREC
CHARACTER*(*)           SRNAME, REC(NREC)
```

3 Description

P01ABF is intended for use by other NAG Library routines. If a routine does not terminate normally, P01ABF is called to provide uniform handling of error or warning conditions. Associated with abnormal termination are error or warning indicators, which are listed in Section 6 of the routine documents. P01ABF is called with the indicator value stored in IERROR and the name of the calling library routine in SRNAME. Messages relating to the reason for termination may optionally be supplied in the array REC. The action of P01ABF then depends on the entry values of IFAIL and IERROR.

If IERROR $= 0$ (successful termination), the value 0 is simply returned through the routine name. No message is output.

Nonzero values of IERROR indicate abnormal termination and the action taken depends on the value of IFAIL.

If IFAIL $= -1$ or 1, the value of IERROR is returned through the routine name (soft failure); if IFAIL $= 0$, then execution of your program is terminated without returning to the calling routine (hard failure).

If IFAIL $= 1$, then no output occurs from P01ABF (silent exit). If IFAIL $= 0$ or -1, then P01ABF writes messages to the unit number specified by a call to X04AAF (noisy exit). Any messages supplied by the calling routine in the array REC are output first, followed by a record of the form

```
** ABNORMAL EXIT from NAG Library routine XXXXXX: IFAIL =      n
```

where `XXXXXX` is the value of SRNAME and `n` is the value of IERROR. For soft failure this is followed by the record

```
** NAG soft failure: control returned
```

and for hard failure by

```
** NAG hard failure: execution terminated
```

4 References

None.

5 Parameters

1: IFAIL – INTEGER *Input*

On entry: the value of IFAIL determines the action of the routine as described in Section 3.

2: IERROR – INTEGER *Input*

On entry: the value of the error or warning indicator. Unless IFAIL = 0, the value of IERROR is returned through the routine name.

3: SRNAME – CHARACTER*(*) *Input*

On entry: the name of the routine which has called P01ABF. If this is a NAG Library routine, the length of SRNAME is always 6.

4: NREC – INTEGER *Input*

On entry: the number of elements (records) in the array REC.

Constraint: NREC ≥ 0.

5: REC(NREC) – CHARACTER*(*) array *Input*

Note: the dimension of the array REC must be at least $\max(1, \text{NREC})$.

On entry: an internal file. Unless IFAIL = 1, or NREC = 0, the first NREC elements of REC are written to the unit determined by X04AAF. If NREC = 0, then REC is not referenced.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

In this example the subroutine MYSQRT uses P01ABF in the way in which it might be employed by a NAG Library routine. Within MYSQRT, error number 1 is associated with an attempt to find a real square root of a negative number. The first call illustrates soft failure with a silent exit, the second call soft failure with a noisy exit and the third call hard failure. Output from the main program is all in upper case characters; output from P01ABF uses lower case characters.

9.1 Program Text

```
*     P01ABF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION Y
      INTEGER          IFAIL
*     .. External Subroutines ..
      EXTERNAL         MYSQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'P01ABF Example Program Results'
      WRITE (NOUT,*)
      WRITE (NOUT,*)
     +'Soft failure, silent exit - message output from the main program'
      IFAIL = 1
      CALL MYSQRT(-1.0D0,Y,IFAIL)
      IF (IFAIL.EQ.0) THEN
         WRITE (NOUT,99999) Y
```

```
      ELSE
         WRITE (NOUT,*)
     +     'Attempt to take the square root of a negative number'
      END IF
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Soft failure, noisy exit'
      IFAIL = -1
      CALL MYSQRT(-2.0D0,Y,IFAIL)
      IF (IFAIL.EQ.0) THEN
         WRITE (NOUT,99999) Y
      END IF
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Hard failure, noisy exit'
      IFAIL = 0
      CALL MYSQRT(-3.0D0,Y,IFAIL)
      IF (IFAIL.EQ.0) THEN
         WRITE (NOUT,99999) Y
      END IF
*
99999 FORMAT (1X,F10.4)
      END
*
      SUBROUTINE MYSQRT(X,Y,IFAIL)
*     Simple routine to compute square root
*     .. Parameters ..
      CHARACTER*6      SRNAME
      PARAMETER        (SRNAME='MYSQRT')
*     .. Scalar Arguments ..
      DOUBLE PRECISION X, Y
      INTEGER          IFAIL
*     .. Local Arrays ..
      CHARACTER*51     REC(1)
*     .. External Functions ..
      INTEGER          P01ABF
      EXTERNAL         P01ABF
*     .. Intrinsic Functions ..
      INTRINSIC        SQRT
*     .. Local Scalars ..
      INTEGER          IERROR, NREC
*     .. Executable Statements ..
      IF (X.GE.0.0D0) THEN
         Y = SQRT(X)
         IFAIL = 0
      ELSE
         WRITE (REC,99999) '** Attempt to take the square root of ', X
         IERROR = 1
         NREC = 1
         IFAIL = P01ABF(IFAIL,IERROR,SRNAME,NREC,REC)
      END IF
      RETURN
*
99999 FORMAT (1X,A,1P,E12.5)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 P01ABF Example Program Results

 Soft failure, silent exit - message output from the main program
 Attempt to take the square root of a negative number

 Soft failure, noisy exit
 ** Attempt to take the square root of -2.00000E+00
 ** ABNORMAL EXIT from NAG Library routine MYSQRT: IFAIL =     1
```

```
 ** NAG soft failure - control returned

 Hard failure, noisy exit
 ** Attempt to take the square root of -3.00000E+00
 ** ABNORMAL EXIT from NAG Library routine MYSQRT: IFAIL =     1
 ** NAG hard failure - execution terminated
```