NAG Fortran 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(*)
```

3 Description

P01ABF is intended for use by other NAG Fortran 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.

Non-zero 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 the user's 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 Fortran 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
```

See also Section 8 concerning compatibility with error-handling mechanisms in the NAG Fortran Library before Mark 12.

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 Fortran 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(*) – 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

P01ABF was introduced at Mark 12 of the NAG Fortran Library, to supersede the earlier error-handling P01AAF. P01ABF is compatible with P01AAF except that the details of the messages printed on hard failure have changed.

Compatibility includes even those routines (introduced mainly at Marks 7 and 8) which used IFAIL in a different way to control the output of messages. In those routines IFAIL is regarded as a decimal integer whose least significant two digits are denoted ba with the following meanings:

$a = 0$: hard failure $\quad a = 1$: soft failure

$b = 0$: silent exit $\quad b = 1$: noisy exit

However this aspect of P01ABF will not, and should not, be used in future.

9 Example

In the simple program below, the subroutine MYSQRT uses P01ABF in the way in which it is usually employed by a NAG Fortran 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

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     P01ABF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             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(-1.0e0,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(-2.0e0,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(-3.0e0,Y,IFAIL)
      WRITE (NOUT,99999) Y
      STOP
*
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 ..
      real              X, Y
      INTEGER           IFAIL
*     .. Local Arrays ..
      CHARACTER*51      REC(1)
*     .. External Functions ..
      INTEGER           P01ABF
      EXTERNAL          P01ABF
*     .. Intrinsic Functions ..
      INTRINSIC         SQRT
*     .. Executable Statements ..
      IF (X.GE.0.0e0) THEN
         Y = SQRT(X)
         IFAIL = 0
      ELSE
         WRITE (REC,99999) '** Attempt to take the square root of ', X
         IFAIL = P01ABF(IFAIL,1,SRNAME,1,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
```