NAG Library Routine Document

A00ADF

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

A00ADF prints information about the version of the NAG Library in use.

2 Specification

```
SUBROUTINE A00ADF (IMPL, PREC, PCODE, MKMAJ, MKMIN, HDWARE, OPSYS,        &
                   FCOMP, VEND, LICVAL)

INTEGER        MKMAJ, MKMIN
LOGICAL        LICVAL
CHARACTER(*)   IMPL, PREC, PCODE, HDWARE, OPSYS, FCOMP, VEND
```

3 Description

The NAG Library is available for use on a number of different computer systems. For each distinct system an implementation of the library is prepared and this implementation is given a unique code. The specifics that define the implementation are: the working precision, the major and minor marks of the NAG Library, the target hardware and operating system, the compiler used, and the vendor library (if any) that is also required to be linked. A00ADF may be called to return, in separate parameters, these specific details of the NAG Library implementation that is being used; it also returns whether a valid licence has been found for this implementation. This differs from A00AAF which simply outputs the collected information in a readable form directly to the current advisory message unit (see X04ABF).

4 References

None.

5 Parameters

1: IMPL – CHARACTER(*)   *Output*

   Note: it is recommended that IMPL be at least 57 characters in length.

   On exit: the implementation title which usually lists the target platform, operating system and compiler.

2: PREC – CHARACTER(*)   *Output*

   Note: it is recommended that PREC be at least 57 characters in length.

   On exit: the working or basic precision of the implementation. Some routines may perform operations in reduced precision or additional precision, but the great majority will perform all operations in basic precision. See Section 4.4 in the Essential Introduction for definitions of these precisions.

3: PCODE – CHARACTER(*)   *Output*

   Note: it is recommended that PCODE be at least 20 characters in length.

   On exit: the product code for the NAG Library implementation that is being used. The code has a discernible structure, but it is not necessary to know the details of this structure. The product code can be used to differentiate between individual product licence codes.

4:    MKMAJ – INTEGER   *Output*

   On exit: the major mark of the NAG Library implementation that is being used.

5:    MKMIN – INTEGER   *Output*

   On exit: the minor mark of the NAG Library implementation that is being used.

6:    HDWARE – CHARACTER(*)   *Output*

   Note: it is recommended that HDWARE be at least 64 characters in length.

   On exit: the target hardware for the NAG Library implementation that is being used.

7:    OPSYS – CHARACTER(*)   *Output*

   Note: it is recommended that OPSYS be at least 64 characters in length.

   On exit: the target operating system for the NAG Library implementation that is being used.

8:    FCOMP – CHARACTER(*)   *Output*

   Note: it is recommended that FCOMP be at least 64 characters in length.

   On exit: the compiler used to build the NAG Library implementation that is being used.

9:    VEND – CHARACTER(*)   *Output*

   Note: it is recommended that VEND be at least 64 characters in length.

   On exit: the subsidiary library, if any, that must be linked with the NAG Library implementation that is being used. If the implementation does not require a subsidiary library then the string

```
'(self-contained)'
```

   will be returned in VEND.

10:   LICVAL – LOGICAL   *Output*

   On exit: specifies whether or not a valid licence has been found for the NAG Library implementation that is being used.

6   Error Indicators and Warnings

None.

7   Accuracy

Not applicable.

8   Parallelism and Performance

Not applicable.

9   Further Comments

None.

10   Example

This example makes a call of A00ADF, collects information on the NAG Library implementation that is being used and prints it out in a form that is similar to the output obtained by a call to A00AAF. Additionally the time is also printed in a readable form. The output from running this example program provides information that is very useful when contacting NAG with a support query.

10.1 Program Text

```
      Program a00adfe

!     A00ADF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: a00adf, x05aaf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: msglen = 14, nout = 6
!     .. Local Scalars ..
      Integer                           :: i, mkmaj, mkmin
      Logical                           :: licval
      Character (80)                    :: fcomp, hdware, impl, opsys, pcode,   &
                                           prec, vend
!     .. Local Arrays ..
      Integer                           :: itime(7)
      Character (102)                   :: msg(msglen)
!     .. Intrinsic Procedures ..
      Intrinsic                         :: trim
!     .. Executable Statements ..
      Write (nout,*) 'A00ADF Example Program Results'

      Call a00adf(impl,prec,pcode,mkmaj,mkmin,hdware,opsys,fcomp,vend,licval)

!     Print implementation details.

      Write (nout,*)

      msg(1) = '*** Start of NAG Library implementation details ***'
      msg(2) = ''
      msg(3) = 'Implementation title: ' // impl
      msg(4) = '            Precision: ' // prec
      msg(5) = '         Product Code: ' // pcode

      If (mkmin<10) Then
        Write (msg(6),99999) mkmaj, mkmin
      Else
        Write (msg(6),99998) mkmaj, mkmin
      End If

      If (vend=='(self-contained)') Then
        msg(7) = '       Vendor Library: None'
      Else
        msg(7) = '       Vendor Library: ' // vend
      End If

      msg(8) = 'Applicable to:'
      msg(9) = '    hardware - ' // hdware
      msg(10) = '    op. sys. - ' // opsys
      msg(11) = '    compiler - ' // fcomp
      msg(12) = 'and compatible systems.'
      msg(13) = ''
      msg(14) = '*** End of NAG Library implementation details ***'

      Do i = 1, msglen
        Write (nout,'(A)') trim(msg(i))
      End Do

!     Print whether valid licence was found for this product.

      Write (nout,*)

      If (licval) Then
        Write (nout,*) 'A valid licence was found for ' // trim(pcode)
      Else
        Write (nout,*) '** A valid licence was not found for ' // trim(pcode)
```

```
      End If

!     Print the date.

      Write (nout,*)
      Write (nout,*) 'This program was run on the following date:'

      Call x05aaf(itime)

      Write (nout,99997) '        Year : ', itime(1)
      Write (nout,99997) '       Month : ', itime(2)
      Write (nout,99997) '         Day : ', itime(3)

      Write (nout,*) '*** ------------------------------------- ***'

99999 Format ('                  Mark: ',I2,'.',I1,1X,A)
99998 Format ('                  Mark: ',I2,'.',I2,1X,A)
99997 Format (1X,A,I4)
    End Program a00adfe
```

10.2 Program Data

None.

10.3 Program Results

```
 A00ADF Example Program Results

*** Start of NAG Library implementation details ***

Implementation title: Linux/NAG nagfor
           Precision: FORTRAN double precision
        Product Code: FLL6A24D9
                Mark: 24.0
      Vendor Library: None
Applicable to:
    hardware - x86_64
    op. sys. - Linux 3.9.4-200.fc18.x86_64
    compiler - NAG Fortran Compiler Release 5.3.2(947)
and compatible systems.

*** End of NAG Library implementation details ***

 A valid licence was found for FLL6A24D9

 This program was run on the following date:
        Year : 2013
       Month :    7
         Day :   24
 *** ------------------------------------- ***
```