

# NAG Library Routine Document

## E04NGF/E04NGA

**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

To supply optional parameters to E04NFF/E04NFA from an external file. More precisely, E04NGF must be used to supply optional parameters to E04NFF and E04NGA must be used to supply optional parameters to E04NFA.

E04NGA is a version of E04NGF that has additional arguments in order to make it safe for use in multithreaded applications (see Section 5). The initialization routine E04WBF **must** have been called before calling E04NGA.

### 2 Specification

#### 2.1 Specification for E04NGF

```
SUBROUTINE E04NGF (IOPTNS, INFORM)
  INTEGER IOPTNS, INFORM
```

#### 2.2 Specification for E04NGA

```
SUBROUTINE E04NGA (IOPTNS, LWSAV, IWSAV, RWSAV, INFORM)
  INTEGER          IOPTNS, IWSAV(610), INFORM
  REAL (KIND=nag_wp) RWSAV(475)
  LOGICAL          LWSAV(120)
```

### 3 Description

E04NGF/E04NGA may be used to supply values for optional parameters to E04NFF/E04NFA. E04NGF/E04NGA reads an external file and each line of the file defines a single optional parameter. It is only necessary to supply values for those arguments whose values are to be different from their default values.

Each optional parameter is defined by a single character string, of up to 72 characters, consisting of one or more items. The items associated with a given option must be separated by spaces, or equals signs [=]. Alphabetic characters may be upper or lower case. The string

```
Print Level = 1
```

is an example of a string used to set an optional parameter. For each option the string contains one or more of the following items:

- a mandatory keyword;
- a phrase that qualifies the keyword;
- a number that specifies an integer or real value. Such numbers may be up to 40 contiguous characters in Fortran's I, F, E or D formats, terminated by a space if this is not the last item on the line.

Blank strings and comments are ignored. A comment begins with an asterisk (\*) and all subsequent characters in the string are regarded as part of the comment.

The file containing the options must start with `Begin` and must finish with `End`. An example of a valid options file is:

```
Begin * Example options file
  Print level = 5
End
```

For E04NGF each line of the file is normally printed as it is read, on the current advisory message unit (see X04ABF), but printing may be suppressed using the keyword **Nolist**. To suppress printing of Begin, **Nolist** must be the first option supplied as in the file:

```
Begin
  Nolist
  Print level = 5
End
```

Printing will automatically be turned on again after a call to E04NFF or E04NGF and may be turned on again at any time using the keyword **List**.

For E04NGA printing is turned off by default, but may be turned on at any time using the keyword **List**.

Optional parameter settings are preserved following a call to E04NFF/E04NFA and so the keyword **Defaults** is provided to allow you to reset all the optional parameters to their default values before a subsequent call to E04NFF/E04NFA.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in E04NFF/E04NFA.

## 4 References

None.

## 5 Arguments

1: IOPTNS – INTEGER *Input*

*On entry:* the unit number of the options file to be read.

*Constraint:*  $0 \leq \text{IOPTNS} \leq 99$ .

2: INFORM – INTEGER *Output*

**Note:** for E04NGA, *INFORM* does not occur in this position in the argument list. See the additional arguments described below.

*On exit:* contains zero if the options file has been successfully read and a value  $> 0$  otherwise (see Section 6).

**Note:** the following are additional arguments for specific use with E04NGA. Users of E04NGF therefore need not read the remainder of this description.

3: LWSAV(120) – LOGICAL array *Communication Array*

4: IWSAV(610) – INTEGER array *Communication Array*

5: RWSAV(475) – REAL (KIND=nag\_wp) array *Communication Array*

The arrays LWSAV, IWSAV and RWSAV **must not** be altered between calls to any of the routines E04NGA, E04NFA, E04NHA or E04WBF.

6: INFORM – INTEGER *Output*

**Note:** see the argument description for INFORM above.

## 6 Error Indicators and Warnings

INFORM = 1

IOPTNS is not in the range  $[0, 99]$ .

INFORM = 2

Begin was found, but end-of-file was found before End was found.

INFORM = 3

end-of-file was found before Begin was found.

INFORM = 4

Not used.

INFORM = 5

One or more lines of the options file is invalid. Check that all keywords are neither ambiguous nor misspelt.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

E04NGF/E04NGA is not threaded in any implementation.

## 9 Further Comments

E04NHF/E04NHA may also be used to supply optional parameters to E04NFF/E04NFA.

## 10 Example

This example solves the same problem as the example for E04NFF/E04NFA, but in addition illustrates the use of E04NGF/E04NGA and E04NHF/E04NHA to set optional parameters for E04NFF/E04NFA.

In this example the options file read by E04NGF/E04NGA is appended to the data file for the program (see Section 10.2). It would usually be more convenient in practice to keep the data file and the options file separate.

### 10.1 Program Text

*the following program illustrates the use of E04NGF. An equivalent program illustrating the use of E04NGA is available with the supplied Library and is also available from the NAG web site.*

```
!   E04NGF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

Module e04ngfe_mod

!       E04NGF Example Program Module:
!       Parameters and User-defined Routines

!       .. Use Statements ..
!       Use nag_library, Only: nag_wp
!       .. Implicit None Statement ..
!       Implicit None
!       .. Accessibility Statements ..
!       Private
!       Public                                :: qphess
!       .. Parameters ..
!       Integer, Parameter, Public           :: iset = 1, nin = 5, ninopt = 7,      &
!                                              nout = 6

Contains
!       Subroutine qphess(n,jthcol,h,ldh,x,hx)
!       In this version of QPHESS, the lower triangle of matrix H is
```

```

!      stored in packed form (by columns) in array H.
!      More precisely, the lower triangle of matrix H must be stored with
!      matrix element H(i,j) in array element H(i+(2*N-j)*(j-1)/2,1),
!      for i.ge. j.
!      Note that storing the lower triangle of matrix H in packed form (by
!      columns) is equivalent to storing the upper triangle of matrix H in
!      packed form (by rows).
!      Note also that LDH is used to define the size of array H, and
!      must therefore be at least N*(N+1)/2.

!      .. Scalar Arguments ..
Integer, Intent (In)          :: jthcol, ldh, n
!      .. Array Arguments ..
Real (Kind=nag_wp), Intent (In) :: h(ldh,*), x(n)
Real (Kind=nag_wp), Intent (Out) :: hx(n)
!      .. Local Scalars ..
Real (Kind=nag_wp)           :: s
Integer                       :: i, inc, j, l, lp1
!      .. Executable Statements ..
If (jthcol/=0) Then

!      Special case -- extract one column of H.

      l = jthcol
      inc = n

      Do i = 1, jthcol
        hx(i) = h(l,1)
        inc = inc - 1
        l = l + inc
      End Do

      l = l - inc + 1

      If (jthcol<n) Then
        lp1 = l

        Do i = jthcol + 1, n
          hx(i) = h(lp1,1)
          lp1 = lp1 + 1
        End Do

      End If

Else

!      Normal case.

      l = 0

      Do i = 1, n
        s = 0.0E0_nag_wp

        Do j = i, n
          l = l + 1
          s = s + h(l,1)*x(j)
        End Do

        hx(i) = s
      End Do

      l = 0

      Do j = 1, n - 1
        l = l + 1

        Do i = j + 1, n
          l = l + 1
          hx(i) = hx(i) + h(l,1)*x(j)
        End Do
      End Do

```

```

        End Do

        End If

        Return

    End Subroutine qphess
End Module e04ngfe_mod
Program e04ngfe

!      E04NGF Example Main Program

!      .. Use Statements ..
Use nag_library, Only: e04nff, e04ngf, e04nhf, nag_wp, x04abf, x04acf,    &
                        x04baf
Use e04ngfe_mod, Only: iset, nin, ninopt, nout, qphess
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Character (*), Parameter      :: fname = 'e04ngfe.opt'
!      .. Local Scalars ..
Real (Kind=nag_wp)           :: obj
Integer                      :: i, ifail, inform, iter, j, lda, ldh, &
                                liwork, lwork, mode, n, nclin,      &
                                outchn, sda
Character (80)                :: rec
Character (1)                 :: uplo
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: a(:,,:), ax(:), bl(:), bu(:),      &
                                clamda(:), cvec(:), h(:,,:), work(:), &
                                x(:)
Integer, Allocatable          :: istate(:), iwork(:)
!      .. Intrinsic Procedures ..
Intrinsic                    :: max
!      .. Executable Statements ..
Write (rec,99998) 'E04NGF Example Program Results'
Call x04baf(nout,rec)

!      Skip heading in data file
Read (nin,*)

Read (nin,*) n, nclin
liwork = 2*n + 3
lda = max(1,nclin)

If (nclin>0) Then
    sda = n
Else
    sda = 1
End If

!      This particular example problem is of type QP2 with a nondefault QPHESS,
!      so we allocate CVEC(N) and H(LDH,1), and define LDH and LWORK as below

ldh = n*(n+1)/2

If (nclin>0) Then
    lwork = 2*n**2 + 8*n + 5*nclin
Else
    lwork = n**2 + 8*n
End If

Allocate (istate(n+nclin),ax(max(1,nclin)),iwork(liwork),h(ldh,1),bl(n+ &
    nclin),bu(n+nclin),cvec(n),x(n),a(lda,sda),clamda(n+nclin), &
    work(lwork))

Read (nin,*) cvec(1:n)
Read (nin,*)(a(i,1:n),i=1,nclin)
Read (nin,*) bl(1:(n+nclin))
Read (nin,*) bu(1:(n+nclin))
Read (nin,*) x(1:n)

```

```

      Read (nin,*) uplo
      If (uplo=='U') Then
!        Read the upper triangle of H
          Read (nin,*)((h(j+(2*n-i)*(i-1)/2,1),j=i,n),i=1,n)
      Else If (uplo=='L') Then
!        Read the lower triangle of H
          Read (nin,*)((h(i+(2*n-j)*(j-1)/2,1),j=1,i),i=1,n)
      End If

      ldh = n*(n+1)/2

!      Set the unit number for advisory messages to OUTCHN
      outchn = nout
      Call x04abf(iset,outchn)

!      Set four options using E04NHF
      Call e04nhf(' Print Level = 1 ')
      Call e04nhf(' Check Frequency = 10 ')
      Call e04nhf(' Crash Tolerance = 0.05 ')
      Call e04nhf(' Infinite Bound Size = 1.0D+25 ')

!      Open the options file for reading
      mode = 0

      ifail = 0
      Call x04acf(ninopt,fname,mode,ifail)

!      Read the options file for the remaining options
      Call e04ngf(ninopt,inform)

      If (inform/=0) Then
        Write (rec,99999) 'E04NGF terminated with INFORM =', inform
        Call x04baf(nout,rec)
        Go To 100
      End If

!      Solve the problem

      ifail = 0
      Call e04nff(n,nclin,a,lda,bl,bu,cvec,h,ldh,qphess,istate,x,iter,obj,ax, &
        clamda,iwork,liwork,work,lwork,ifail)

100    Continue

99999 Format (1X,A,I5)
99998 Format (1X,A)
      End Program e04ngfe

```

## 10.2 Program Data

```

Begin   Example options file for  E04NGF
  Feasibility Phase Iteration Limit = 5      * (Default = 70)
  Optimality   Phase Iteration Limit = 10     * (Default = 70)
End

E04NGF Example Program Data
  7 7                                     :Values of N and NCLIN
-0.02 -0.20 -0.20 -0.20 -0.20 0.04 0.04 :End of CVEC
 1.00 1.00 1.00 1.00 1.00 1.00 1.00
 0.15 0.04 0.02 0.04 0.02 0.01 0.03
 0.03 0.05 0.08 0.02 0.06 0.01 0.00
 0.02 0.04 0.01 0.02 0.02 0.00 0.00
 0.02 0.03 0.00 0.00 0.01 0.00 0.00
 0.70 0.75 0.80 0.75 0.80 0.97 0.00
 0.02 0.06 0.08 0.12 0.02 0.01 0.97 :End of matrix A
-0.01 -0.10 -0.01 -0.04 -0.10 -0.01 -0.01
-0.13 -1.0D+25 -1.0D+25 -1.0D+25 -1.0D+25 -9.92D-02 -3.0D-03 :End of BL
 0.01 0.15 0.03 0.02 0.05 1.0D+25 1.0D+25
-0.13 -4.9D-03 -6.4D-03 -3.7D-03 -1.2D-03 1.0D+25 2.0D-03 :End of BU
-0.01 -0.03 0.00 -0.01 -0.10 0.02 0.01 :End of X
'L'                                     :End of UPLO
 2.00
 0.00 2.00
 0.00 0.00 2.00
 0.00 0.00 2.00 2.00
 0.00 0.00 0.00 0.00 2.00
 0.00 0.00 0.00 0.00 0.00 -2.00
 0.00 0.00 0.00 0.00 0.00 -2.00 -2.00 :End of matrix H

```

## 10.3 Program Results

E04NGF Example Program Results

Calls to E04NHF  
-----

```

Print Level = 1
Check Frequency = 10
Crash Tolerance = 0.05
Infinite Bound Size = 1.0D+25

```

OPTIONS file  
-----

```

Begin   Example options file for  E04NGF
  Feasibility Phase Iteration Limit = 5      * (Default = 70)
  Optimality   Phase Iteration Limit = 10     * (Default = 70)
End

```

\*\*\* E04NFF

Parameters  
-----

```

Problem type..... QP2

Linear constraints..... 7      Feasibility tolerance.. 1.05E-08
Variables..... 7      Optimality tolerance... 1.05E-08
Hessian rows..... 7      Rank tolerance..... 1.11E-14

Infinite bound size.... 1.00E+25      COLD start.....
Infinite step size..... 1.00E+25      EPS (machine precision) 1.11E-16

Check frequency..... 10      Expand frequency..... 5
Minimum sum of infeas.. NO      Crash tolerance..... 5.00E-02

Max degrees of freedom. 7      Print level..... 1

```

```

Feasibility phase itns.      5      Monitoring file..... -1
Optimality phase itns.     10

```

```

Workspace provided is      IWORK(      17),  WORK(      189).
To solve problem we need  IWORK(      17),  WORK(      189).

```

Varbl	State	Value	Lower Bound	Upper Bound	Lagr Mult	Slack
V 1	LL	-1.000000E-02	-1.000000E-02	1.000000E-02	0.4700	.
V 2	FR	-6.986465E-02	-0.100000	0.150000	.	3.0135E-02
V 3	FR	1.825915E-02	-1.000000E-02	3.000000E-02	.	1.1741E-02
V 4	FR	-2.426081E-02	-4.000000E-02	2.000000E-02	.	1.5739E-02
V 5	FR	-6.200564E-02	-0.100000	5.000000E-02	.	3.7994E-02
V 6	FR	1.380544E-02	-1.000000E-02	None	.	2.3805E-02
V 7	FR	4.066496E-03	-1.000000E-02	None	.	1.4066E-02

L Con	State	Value	Lower Bound	Upper Bound	Lagr Mult	Slack
L 1	EQ	-0.130000	-0.130000	-0.130000	-1.908	.
L 2	FR	-5.879898E-03	None	-4.900000E-03	.	9.7990E-04
L 3	UL	-6.400000E-03	None	-6.400000E-03	-0.3144	-2.6021E-18
L 4	FR	-4.537323E-03	None	-3.700000E-03	.	8.3732E-04
L 5	FR	-2.915996E-03	None	-1.200000E-03	.	1.7160E-03
L 6	LL	-9.920000E-02	-9.920000E-02	None	1.955	4.1633E-17
L 7	LL	-3.000000E-03	-3.000000E-03	2.000000E-03	1.972	1.7347E-18

Exit E04NFF - Optimal QP solution.

Final QP objective value = 0.3703165E-01

Exit from QP problem after 8 iterations.

---