

# NAG Library Routine Document

## E04MGF/E04MGA

**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 E04MFF/E04MFA from an external file. More precisely, E04MGF must be used to supply optional parameters to E04MFF and E04MGA must be used to supply optional parameters to E04MFA.

E04MGA is a version of E04MGF that has additional parameters in order to make it safe for use in multithreaded applications (see Section 5). The initialization routine E04WBF **must** have been called before calling E04MGA.

### 2 Specification

#### 2.1 Specification for E04MGF

```
SUBROUTINE E04MGF (IOPTNS, INFORM)
INTEGER IOPTNS, INFORM
```

#### 2.2 Specification for E04MGA

```
SUBROUTINE E04MGA (IOPTNS, LWSAV, IWSAV, RWSAV, INFORM)
INTEGER          IOPTNS, IWSAV(610), INFORM
REAL (KIND=nag_wp) RWSAV(475)
LOGICAL         LWSAV(120)
```

### 3 Description

E04MGF/E04MGA may be used to supply values for optional parameters to E04MFF/E04MFA. E04MGF/E04MGA reads an external file and each line of the file defines a single optional parameter. It is only necessary to supply values for those parameters 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 16 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 E04MGF 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 E04MFF or E04MGF and may be turned on again at any time using the keyword **List**.

For E04MGA 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 E04MFF/E04MFA 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 E04MFF/E04MFA.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in E04MFF/E04MFA.

## 4 References

None.

## 5 Parameters

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 E04MGA, *INFORM* does not occur in this position in the parameter list. See the additional parameters 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 parameters for specific use with E04MGA. Users of E04MGF 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 E04MGA, E04MFA, E04MHA or E04WBF.

6: INFORM – INTEGER *Output*

**Note:** see the parameter 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

Not applicable.

## 9 Further Comments

E04MHF/E04MHA may also be used to supply optional parameters to E04MFF/E04MFA.

## 10 Example

This example solves the same problem as the example for E04MFF/E04MFA, but in addition illustrates the use of E04MGF/E04MGA and E04MHF/E04MHA to set optional parameters for E04MFF/E04MFA.

In this example the options file read by E04MGF/E04MGA 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 E04MGF. An equivalent program illustrating the use of E04MGA is available with the supplied Library and is also available from the NAG web site.*

```

Program e04mgfe
!      E04MGF Example Program Text
!
!      Mark 25 Release. NAG Copyright 2014.
!
!      .. Use Statements ..
!      Use nag_library, Only: e04mff, e04mgf, e04mhf, nag_wp, x04abf, x04acf, &
!                               x04baf
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: iset = 1, nin = 5, ninopt = 7,      &
!                                   nout = 6
!      Character (*), Parameter   :: fname = 'e04mgfe.opt'
!      .. Local Scalars ..
!      Real (Kind=nag_wp)         :: obj
!      Integer                    :: i, ifail, inform, iter, lda, liwork, &
!                                   lwork, mode, n, nclin, outchn, sda
!      Character (80)             :: rec
!      .. Local Arrays ..
!      Real (Kind=nag_wp), Allocatable :: a(:,,:), ax(:), bl(:), bu(:),      &
!                                   clamda(:), cvec(:), work(:), x(:)
!      Integer, Allocatable       :: istate(:), iwork(:)
!      .. Intrinsic Procedures ..
!      Intrinsic                  :: max

```

```

!      .. Executable Statements ..
      Write (rec,*) 'E04MGF Example Program Results'
      Call x04baf(nout,rec)

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

      Read (nin,*) n, nclin

      liwork = 2*n + 3

!      The minimum LWORK for an LP problem:

      If (0<nclin .And. nclin<n) Then
        lwork = 2*(nclin+1)**2 + 7*n + 5*nclin
      Else If (nclin>=n) Then
        lwork = 2*n**2 + 7*n + 5*nclin
      Else
        lwork = 7*n + 1
      End If

      lda = max(1,nclin)

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

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

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

!      Set the unit number for advisory messages to OUTCHN

      outchn = nout
      Call x04abf(iset,outchn)

!      Set three options using E04MHF

      Call e04mhf(' Print Level = 1 ')
      Call e04mhf(' Check Frequency = 10 ')
      Call e04mhf(' 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 e04mgf(ninopt,inform)

      If (inform/=0) Then
        Write (rec,99999) 'E04MGF terminated with INFORM = ', inform
        Call x04baf(nout,rec)
        Go To 100
      End If

!      Solve the problem

      ifail = 0

```

```

      Call e04mff(n,nclin,a,lda,bl,bu,cvec,istate,x,iter,obj,ax,clamda,iwork, &
        liwork,work,lwork,ifail)

100  Continue

99999 Format (1X,A,I5)
      End Program e04mgfe

```

## 10.2 Program Data

```

Begin  Example options file for  E04MGF
      Crash Tolerance = 0.05  * (Default = 0.01)
      Iteration Limit = 25    * (Default = 70)
End

E04MGF 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

```

## 10.3 Program Results

E04MGF Example Program Results

Calls to E04MHF

-----

```

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

```

OPTIONS file

-----

```

Begin  Example options file for  E04MGF
      Crash Tolerance = 0.05  * (Default = 0.01)
      Iteration Limit = 25    * (Default = 70)
End

```

\*\*\* E04MFF

Parameters

-----

```

Problem type.....          LP

Linear constraints.....      7      Feasibility tolerance..  1.05E-08
Variables.....              7      Optimality tolerance...  1.05E-08

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

Print level.....            1      Iteration limit.....      25
Monitoring file.....       -1

```

Workspace provided is IWORK( 17), WORK( 182).  
 To solve problem we need IWORK( 17), WORK( 182).

Varbl	State	Value	Lower Bound	Upper Bound	Lagr Mult	Slack
V	1	LL	-1.000000E-02	1.000000E-02	0.3301	.
V	2	LL	-0.100000	0.150000	1.4384E-02	.
V	3	UL	3.000000E-02	3.000000E-02	-9.0997E-02	.
V	4	UL	2.000000E-02	2.000000E-02	-7.6612E-02	.
V	5	FR	-6.748534E-02	5.000000E-02	.	3.2515E-02
V	6	FR	-2.280130E-03	None	.	7.7199E-03
V	7	FR	-2.345277E-04	None	.	9.7655E-03

L Con	State	Value	Lower Bound	Upper Bound	Lagr Mult	Slack
L	1	EQ	-0.130000	-0.130000	-1.431	2.7756E-17
L	2	FR	-5.479544E-03	None	.	5.7954E-04
L	3	FR	-6.571922E-03	None	.	1.7192E-04
L	4	FR	-4.849707E-03	None	.	1.1497E-03
L	5	FR	-3.874853E-03	None	.	2.6749E-03
L	6	LL	-9.920000E-02	None	1.501	.
L	7	LL	-3.000000E-03	2.000000E-03	1.517	-8.6736E-19

Exit E04MFF - Optimal LP solution.

Final LP objective value = 0.2359648E-01

Exit from LP problem after 5 iterations.

---