NAG Library Routine Document

e04rxf

1 Purpose

e04rxf is a part of the NAG optimization modelling suite. It allows you to read or write a piece of information to the problem stored in the handle. For example, it may be used to extract the current approximation of the solution during a monitoring step.

2 Specification

Fortran Interface

```
Subroutine e04rxf (handle, cmdstr, ioflag, lrarr, rarr, ifail)
Integer, Intent (In)                :: ioflag
Integer, Intent (Inout)             :: lrarr, ifail
Real (Kind=nag_wp), Intent (Inout) :: rarr(lrarr)
Character (*), Intent (In)          :: cmdstr
Type (c_ptr), Intent (In)           :: handle
```

3 Description

e04rxf adds an additional means of communication to routines within the NAG optimization modelling suite. It allows you to either read or write a piece of information in the handle in the form of a real array. The item is identified by **cmdstr** and the direction of the communication is set by **ioflag**.

The following **cmdstr** are available:

Primal Variables or **X**
: The current value of the primal variables.

Dual Variables or **U**
: The current value of the dual variables (Lagrangian multipliers).

The functionality is limited in this release of the NAG Library to the retrieval of the approximate solution within the monitoring step of **e04mtf** or its final solution.

4 References

None.

5 Arguments

1: **handle** – Type (c_ptr) *Input*

On entry: the handle to the problem. It needs to be initialized by **e04raf** and **must not** be changed between calls to the NAG optimization modelling suite.

2: **cmdstr** – Character(*) *Input*

On entry: a string which identifies the item within the handle to be read or written. The string is case insensitive and space tolerant.

Constraint: **cmdstr** = 'Primal Variables', 'Dual Variables', 'X' or 'U'.

3: **ioflag** – Integer *Input*

On entry: indicates the direction of the communication.

ioflag $\neq 0$
e04rxf will extract the requested information from the handle to **rarr**.

ioflag $= 0$
The writing mode will apply and the content of **rarr** will be copied to the handle.

4: **lrarr** – Integer *Input/Output*

On entry: the dimension of the array **rarr**.

On exit: the correct expected dimension of **rarr** if **lrarr** does not match the item identified by **cmdstr** (in this case **e04rxf** returns **ifail** $= 2$).

5: **rarr**(**lrarr**) – Real (Kind=nag_wp) array *Input/Output*

On entry: if **ioflag** $= 0$ (write mode), **rarr** must contain the information to be written to the handle; otherwise it does not need to be set.

On exit: if **ioflag** $\neq 0$ (read mode), **rarr** contains the information requested by **cmdstr**; otherwise **rarr** is unchanged.

6: **ifail** – Integer *Input/Output*

On entry: **ifail** must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output arguments may be useful even if **ifail** $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of ifail on exit.**

On exit: **ifail** $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry **ifail** $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by **x04aaf**).

Note: **e04rxf** may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

ifail $= 1$

The supplied **handle** does not define a valid handle to the data structure for the NAG optimization modelling suite. It has not been initialized by **e04raf** or it has been corrupted.

ifail $= 2$

On entry, **lrarr** $= \langle value \rangle$, expected value $= \langle value \rangle$.
Constraint: **lrarr** must match the size of the data identified in **cmdstr**.

ifail $= 3$

The provided **cmdstr** is not recognised.

ifail $= 4$

Reading mode is not supported for the given **cmdstr**.

ifail $= 5$

Writing mode is not supported for the given **cmdstr**.

ifail $= 6$

The request cannot be processed at this phase.

The requested information is not available.

ifail $= 7$

The request cannot be processed by the current solver.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

ifail $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

e04rxf is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example demonstrates how to use **e04rxf** to extract the current approximation of the solution when the monitoring routine **monit** is called during the solve by **e04mtf**.

We solve the following linear programming problem:

$$-0.02x_1 - 0.2x_2 - 0.2x_3 - 0.2x_4 - 0.2x_5 + 0.04x_6 + 0.04x_7$$

subject to the bounds

$$\begin{aligned} -0.01 &\leq x_1 \leq 0.01 \\ -0.1 &\leq x_2 \leq 0.15 \\ -0.01 &\leq x_3 \leq 0.03 \\ -0.04 &\leq x_4 \leq 0.02 \\ -0.1 &\leq x_5 \leq 0.05 \\ -0.01 &\leq x_6 \\ -0.01 &\leq x_7 \end{aligned}$$

and the general constraints

$$
\begin{array}{rcrcrcrcrcrcrcl}
 & & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & = & -0.13 \\
 & & 0.15x_1 & + & 0.04x_2 & + & 0.02x_3 & + & 0.04x_4 & + & 0.02x_5 & + & 0.01x_6 & + & 0.03x_7 & \leq & -0.0049 \\
 & & 0.03x_1 & + & 0.05x_2 & + & 0.08x_3 & + & 0.02x_4 & + & 0.06x_5 & + & 0.01x_6 & & & \leq & -0.0064 \\
 & & 0.02x_1 & + & 0.04x_2 & + & 0.01x_3 & + & 0.02x_4 & + & 0.02x_5 & & & & & \leq & -0.0037 \\
 & & 0.02x_1 & + & 0.03x_2 & & & & & + & 0.01x_5 & & & & & \leq & -0.0012 \\
-0.0992 & \leq & 0.70x_1 & + & 0.75x_2 & + & 0.80x_3 & + & 0.75x_4 & + & 0.80x_5 & + & 0.97x_6 & & & & \\
-0.003 & \leq & 0.02x_1 & + & 0.06x_2 & + & 0.08x_3 & + & 0.12x_4 & + & 0.02x_5 & + & 0.01x_6 & + & 0.97x_7 & \leq & 0.002
\end{array}
$$

During the monitoring step of **e04mtf**, if the three convergence measures are below an acceptable threshold, the approximate solution is extracted with **e04rxf** and printed on the standard output.

10.1 Program Text

```
!   e04rxf Example Program Text

!   Call during the monitoring of the solve of a small linear programming
!   problem

!   Mark 26.1 Release. NAG Copyright 2017.

    Module e04rxfe_mod

!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                             :: monit
    Contains
      Subroutine monit(handle,rinfo,stats,iuser,ruser,cpuser,inform)

!       Monitoring function

!       .. Use Statements ..
        Use iso_c_binding, Only: c_ptr
        Use nag_library, Only: e04rxf, nag_wp
!       .. Scalar Arguments ..
        Type (c_ptr), Intent (In)      :: cpuser, handle
        Integer, Intent (Inout)        :: inform
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: rinfo(100), stats(100)
        Real (Kind=nag_wp), Intent (Inout) :: ruser(*)
        Integer, Intent (Inout)        :: iuser(*)
!       .. Local Scalars ..
        Real (Kind=nag_wp)             :: tol
        Integer                        :: i, ifail, n, nout
!       .. Local Arrays ..
        Real (Kind=nag_wp), Allocatable :: x(:)
!       .. Executable Statements ..

        nout = iuser(1)
        n = iuser(2)
        tol = 1.0E-03_nag_wp
!       x is close to the solution, extract the values with e04rxf and print it
        If (rinfo(5)<tol .And. rinfo(6)<tol .And. rinfo(7)<tol) Then
          Allocate (x(n))
          ifail = 0
          Call e04rxf(handle,'Primal Variables',1,n,x,ifail)
          Write (nout,*)
          Write (nout,99999)                                                  &
            'monit() reports good approximate solution (tol =', tol, '):'
          Do i = 1, n
            Write (nout,99997) i, x(i)
          End Do
          Write (nout,99998) 'End of monit()'
        End If

        Return
```

```
99999   Format (3X,A,Es9.2,A)
99998   Format (3X,A)
99997   Format (5X,'X',I1,': ',Es9.2)
      End Subroutine monit

    End Module e04rxfe_mod

    Program e04rxfe

!     .. Use Statements ..
      Use e04rxfe_mod, Only: monit
      Use iso_c_binding, Only: c_null_ptr, c_ptr
      Use nag_library, Only: e04mtf, e04raf, e04rff, e04rhf, e04rjf, e04rzf,   &
                             e04zmf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Type (c_ptr)                     :: cpuser, handle
      Integer                          :: idlc, ifail, m, n, nnza, nnzc, nnzu
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), bla(:), bua(:), c(:), u(:),    &
                                          x(:), xl(:), xu(:)
      Real (Kind=nag_wp)               :: h(1), rinfo(100), ruser(1),          &
                                          stats(100)
      Integer, Allocatable             :: cindex(:), icola(:), irowa(:)
      Integer                          :: icolh(1), irowh(1), iuser(2)
!     .. Executable Statements ..

      Write (nout,*) 'E04RXF Example Program Results'

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

!     read dimensions of the problem
      Read (nin,*) m, n, nnza, nnzc
      nnzu = 2*n + 2*m

!     Allocate memory
      Allocate (cindex(nnzc),icola(nnza),irowa(nnza),a(nnza),bla(m),bua(m),    &
        xl(n),xu(n),c(nnzc),x(n),u(nnzu))

!     Read problem data
      Read (nin,*) cindex(1:nnzc)
      Read (nin,*) c(1:nnzc)
      Read (nin,*) irowa(1:nnza)
      Read (nin,*) icola(1:nnza)
      Read (nin,*) a(1:nnza)
      Read (nin,*) bla(1:m)
      Read (nin,*) bua(1:m)
      Read (nin,*) xl(1:n)
      Read (nin,*) xu(1:n)

!     Create the problem handle
!     Initialize handle
      ifail = 0
      Call e04raf(handle,n,ifail)

!     set objective function
      Call e04rff(handle,nnzc,cindex,c,0,irowh,icolh,h,ifail)

!     Set box constraints
      Call e04rhf(handle,n,xl,xu,ifail)

!     Set linear constraints.
      idlc = 0
      Call e04rjf(handle,m,bla,bua,nnza,irowa,icola,a,idlc,ifail)

!     Turn on monitoring
      Call e04zmf(handle,'LPIPM Monitor Frequency = 1',ifail)
```

```
!     Print the solution at the end of the solve
      Call e04zmf(handle,'Print Solution = X',ifail)

!     Call LP interior point solver
!     The monitoring function monit calls e04rxf to extract the primal
!     and dual variables.
      cpuser = c_null_ptr
      iuser(1) = nout
      iuser(2) = n
      ifail = -1
      Call e04mtf(handle,n,x,nnzu,u,rinfo,stats,monit,iuser,ruser,cpuser,      &
        ifail)

!     Free the handle memory
      ifail = -1
      Call e04rzf(handle,ifail)


    End Program e04rxfe
```

10.2 Program Data

```
E04RXF Example Program Data
  7  7  41  7                                        : Problem dimensions
  1  2  3  4  5  6  7                               : Objective index
  -0.02  -0.20  -0.20  -0.20  -0.20   0.04   0.04  : Objective values
  1  1  1  1  1  1  1
  2  2  2  2  2  2  2
  3  3  3  3  3  3
  4  4  4  4  4
  5  5  5
  6  6  6  6  6  6
  7  7  7  7  7  7  7                               : End of irowa
  1  2  3  4  5  6  7
  1  2  3  4  5  6  7
  1  2  3  4  5  6
  1  2  3  4  5
  1  2  5
  1  2  3  4  5  6
  1  2  3  4  5  6  7                               : End of icola
  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
  0.02   0.04   0.01   0.02   0.02
  0.02   0.03   0.01
  0.70   0.75   0.80   0.75   0.80   0.97
  0.02   0.06   0.08   0.12   0.02   0.01   0.97   : End of a
 -0.13 -1.0e20 -1.0e20 -1.0e20 -1.0e20 -0.0992 -0.003  : bla
 -0.13 -0.0049 -0.0064 -0.0037 -0.0012  1.0e20  0.002  : bua
 -0.01 -0.1 -0.01 -0.04 -0.1 -0.01 -0.01               : xl
  0.01  0.15 0.03  0.02 0.05 1.0e20 1.0e20             : xu
```

10.3 Program Results

```
 E04RXF Example Program Results

 ---------------------------------------------
  E04MT, Interior point method for LP problems
 ---------------------------------------------

 Begin of Options
     Infinite Bound Size          =         1.00000E+20     * d
     Print File                   =                   6     * d
     Print Level                  =                   2     * d
     Print Options                =                 Yes     * d
     Print Solution               =                   X     * U
     Monitoring File              =                  -1     * d
     Monitoring Level             =                   4     * d
     Stats Time                   =                  No     * d
```

```
    Task                          =            Minimize     * d
    Lpipm Centrality Correctors   =                   6     * d
    Lp Presolve                   =                 Yes     * d
    Lpipm Scaling                 =          Arithmetic     * d
    Lpipm System Formulation      =                Auto     * d
    Lpipm Algorithm               =         Primal-dual     * d
    Lpipm Stop Tolerance          =         1.05367E-08     * d
    Lpipm Monitor Frequency       =                   1     * U
    Lpipm Stop Tolerance 2        =         2.67452E-10     * d
    Lpipm Max Iterative Refinement=                   5     * d
    Lpipm Iteration Limit         =                 100     * d
End of Options

Original Problem Statistics

  Number of variables          7
  Number of constraints        7
  Free variables               0
  Number of nonzeros          41


Presolved Problem Statistics

  Number of variables         13
  Number of constraints        7
  Free variables               0
  Number of nonzeros          47


-------------------------------------------------------------------------------
 it|    pobj    |    dobj    |  optim  |  feas   |  compl  |   mu    | mcc | I
-------------------------------------------------------------------------------
  0 -7.86591E-02  1.71637E-02  1.27E+00  1.06E+00  8.89E-02  1.5E-01
  1  5.74135E-03 -2.24369E-02  6.11E-16  1.75E-01  2.25E-02  2.8E-02   0
  2  1.96803E-02  1.37067E-02  5.06E-16  2.28E-02  2.91E-03  3.4E-03   0
  3  2.15232E-02  1.96162E-02  7.00E-15  9.24E-03  1.44E-03  1.7E-03   0
  4  2.30321E-02  2.28676E-02  1.15E-15  2.21E-03  2.97E-04  3.4E-04   0

  monit() reports good approximate solution (tol = 1.00E-03):
    X1: -9.99E-03
    X2: -1.00E-01
    X3:  3.00E-02
    X4:  2.00E-02
    X5: -6.73E-02
    X6: -2.35E-03
    X7: -2.27E-04
  End of monit()
  5  2.35658E-02  2.35803E-02  1.32E-15  1.02E-04  8.41E-06  9.6E-06   0

  monit() reports good approximate solution (tol = 1.00E-03):
    X1: -1.00E-02
    X2: -1.00E-01
    X3:  3.00E-02
    X4:  2.00E-02
    X5: -6.75E-02
    X6: -2.28E-03
    X7: -2.35E-04
  End of monit()
  6  2.35965E-02  2.35965E-02  1.64E-15  7.02E-08  6.35E-09  7.2E-09   0

  monit() reports good approximate solution (tol = 1.00E-03):
    X1: -1.00E-02
    X2: -1.00E-01
    X3:  3.00E-02
    X4:  2.00E-02
    X5: -6.75E-02
    X6: -2.28E-03
    X7: -2.35E-04
  End of monit()
  7  2.35965E-02  2.35965E-02  1.35E-15  3.52E-11  3.18E-12  3.6E-12   0
-------------------------------------------------------------------------------
```

```
Status: converged, an optimal solution found
------------------------------------------------------------------------------
Final primal objective value         2.359648E-02
Final dual objective value           2.359648E-02
Absolute primal infeasibility        4.168797E-15
Relative primal infeasibility        1.350467E-15
Absolute dual infeasibility          5.084353E-11
Relative dual infeasibility          3.518607E-11
Absolute complementarity gap         2.685778E-11
Relative complementarity gap         3.175366E-12
Iterations                                      7

Primal variables:
  idx   Lower bound        Value      Upper bound
    1  -1.00000E-02   -1.00000E-02    1.00000E-02
    2  -1.00000E-01   -1.00000E-01    1.50000E-01
    3  -1.00000E-02    3.00000E-02    3.00000E-02
    4  -4.00000E-02    2.00000E-02    2.00000E-02
    5  -1.00000E-01   -6.74853E-02    5.00000E-02
    6  -1.00000E-02   -2.28013E-03         inf
    7  -1.00000E-02   -2.34528E-04         inf
```