

NAG Library Routine Document

E04ZMF

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

E04ZMF is an option setting routine for solvers from the NAG optimization modelling suite, namely E04STF and E04SVF. It can set a single optional parameter or reset all of them to their default.

2 Specification

```
SUBROUTINE E04ZMF (HANDLE, OPTSTR, IFAIL)
```

```
INTEGER          IFAIL
CHARACTER(*)     OPTSTR
TYPE (C_PTR)     HANDLE
```

3 Description

E04ZMF can only be called on handles which have been initialized by E04RAF and not during the call to the solver. It has two purposes: to reset all optional parameters to their default values; or to set a single optional parameter to a user-supplied value.

Optional parameters and their values are, in general, presented as a character string, OPTSTR, of the form '*option = optval*'; alphabetic characters can be supplied in either upper or lower case. Both *option* and *optval* may consist of one or more tokens separated by white space. The tokens that comprise *optval* will normally be either an integer, real or character value as defined in the description of the specific optional argument. In addition all optional parameters can take an *optval* DEFAULT which resets the optional parameter to its default value.

Information relating to available option names and their corresponding valid values is given in the documentation of the particular solver. See also E04RAF for a generic description of the suite.

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.
- 2: OPTSTR – CHARACTER(*) *Input*
On entry: a string identifying the option and its value to be set.

Defaults

Resets all options to their default values.

Option = optval

See the documentation of the particular solver for details of valid values for *option* and *optval*. The equals sign (=) delimiter must be used to separate the *option* from its *optval* value.

Option = Default

Resets the given option back to its default value.

OPTSTR is case insensitive. Each token in the *option* and *optval* component must be separated by at least one space.

3: 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, if you are not familiar with this argument, the recommended value is 0. **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).

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

The options cannot be modified in this phase.

IFAIL = 11

On entry, the *option* supplied in OPTSTR was not recognized: OPTSTR = $\langle value \rangle$.

IFAIL = 12

On entry, the expected delimiter '=' was not found in OPTSTR: OPTSTR = $\langle value \rangle$.

IFAIL = 13

On entry, could not convert the specified *optval* to an integer: *optval* = $\langle value \rangle$.

On entry, could not convert the specified *optval* to a real: *optval* = $\langle value \rangle$.

IFAIL = 15

On entry, the *optval* supplied for the integer optional parameter is not valid.

option = $\langle value \rangle$, *optval* = $\langle value \rangle$.

Constraint: *optval* < $\langle value \rangle$.

On entry, the *optval* supplied for the integer optional parameter is not valid.

option = $\langle value \rangle$, *optval* = $\langle value \rangle$.

Constraint: *optval* > $\langle value \rangle$.

On entry, the *optval* supplied for the integer optional parameter is not valid.

option = $\langle value \rangle$, *optval* = $\langle value \rangle$.

Constraint: *optval* ≤ $\langle value \rangle$.

On entry, the *optval* supplied for the integer optional parameter is not valid.

option = $\langle value \rangle$, *optval* = $\langle value \rangle$.

Constraint: *optval* ≥ $\langle value \rangle$.

IFAIL = 16

On entry, the *optval* supplied for the real optional parameter is not valid.
option = $\langle value \rangle$, *optval* = $\langle value \rangle$.
Constraint: *optval* < $\langle value \rangle$.

On entry, the *optval* supplied for the real optional parameter is not valid.
option = $\langle value \rangle$, *optval* = $\langle value \rangle$.
Constraint: *optval* > $\langle value \rangle$.

On entry, the *optval* supplied for the real optional parameter is not valid.
option = $\langle value \rangle$, *optval* = $\langle value \rangle$.
Constraint: *optval* \leq $\langle value \rangle$.

On entry, the *optval* supplied for the real optional parameter is not valid.
option = $\langle value \rangle$, *optval* = $\langle value \rangle$.
Constraint: *optval* \geq $\langle value \rangle$.

IFAIL = 17

On entry, the *optval* supplied for the character optional parameter is not valid.
option = $\langle value \rangle$, *optval* = $\langle value \rangle$.

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

E04ZMF is not threaded in any implementation.

9 Further Comments

None.

10 Example

See the example programs associated with the solvers E04STF and E04SVF for a demonstration of how to use E04ZMF. See also Section 10 in E04RAF for links to all examples in this suite.
