

# NAG Library Routine Document

## E04NTF

**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

E04NTF may be used to supply individual integer optional parameters to E04NQF. The initialization routine E04NPF **must** have been called before calling E04NTF.

### 2 Specification

```
SUBROUTINE E04NTF (STRING, IVALUE, CW, IW, RW, IFAIL)
INTEGER           IVALUE, IW(*), IFAIL
REAL (KIND=nag_wp) RW(*)
CHARACTER(*)      STRING
CHARACTER(8)      CW(*)
```

### 3 Description

E04NTF may be used to supply values for integer optional parameters to E04NQF. It is only necessary to call E04NTF for those arguments whose values are to be different from their default values. One call to E04NTF sets one argument value.

Each integer optional parameter is defined by a single character string in STRING and the corresponding value in IVALUE. For example, the following allows the iteration limit to be defined:

```
ITNLIM = 1000
IF (M > 500) ITNLIM = 500
CALL E04NTF ('Iterations', ITNLIM, CW, IW, RW, IFAIL)
```

Optional parameter settings are preserved following a call to E04NQF 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 E04NQF.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in E04NQF.

### 4 References

None.

### 5 Arguments

- 1: STRING – CHARACTER(\*) *Input*  
*On entry:* a single valid keyword of an integer optional parameter (as described in Section 12 in E04NQF).
- 2: IVALUE – INTEGER *Input*  
*On entry:* an integer value associated with the keyword in STRING.
- 3: CW(\*) – CHARACTER(8) array *Communication Array*  
**Note:** the dimension of the array CW must be at least LENCW (see E04NPF).

4: IW(\*) – INTEGER array *Communication Array*  
**Note:** the dimension of the array IW must be at least LENIW (see E04NPF).

5: RW(\*) – REAL (KIND=nag\_wp) array *Communication Array*  
**Note:** the dimension of the array RW must be at least LENRW (see E04NPF).

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, 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 initialization routine E04NPF has not been called.

IFAIL = 2

The supplied option is invalid. Check that the keywords are neither ambiguous nor misspelt.

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

E04NTF is not threaded in any implementation.

## **9 Further Comments**

E04NRF or E04NSF may also be used to supply integer optional parameters to E04NQF.

## **10 Example**

See Section 10 in E04NQF and E04NRF.

---