

# NAG Library Function Document

## nag\_opt\_handle\_opt\_set (e04zmc)

### 1 Purpose

nag\_opt\_handle\_opt\_set (e04zmc) is an option setting routine for solvers from the NAG optimization modelling suite, namely nag\_opt\_handle\_solve\_ipopt (e04stc) and nag\_opt\_handle\_solve\_pennon (e04svc). It can set a single optional parameter or reset all of them to their default.

### 2 Specification

```
#include <nag.h>
#include <nage04.h>
void nag_opt_handle_opt_set (void *handle, const char *optstr,
                             NagError *fail)
```

### 3 Description

nag\_opt\_handle\_opt\_set (e04zmc) can only be called on handles which have been initialized by nag\_opt\_handle\_init (e04rac) 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 nag\_opt\_handle\_init (e04rac) for a generic description of the suite.

### 4 References

None.

### 5 Arguments

1: **handle** – void \* *Input*

*On entry:* the handle to the problem. It needs to be initialized by nag\_opt\_handle\_init (e04rac) and **must not** be changed.

2: **optstr** – const char \* *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: **fail** – NagError \**Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

**6 Error Indicators and Warnings****NE\_ALLOC\_FAIL**

Dynamic memory allocation failed.

See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

**NE\_BAD\_PARAM**

On entry, argument  $\langle value \rangle$  had an illegal value.

**NE\_HANDLE**

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 `nag_opt_handle_init` (e04rac) or it has been corrupted.

**NE\_INTERNAL\_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

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

See Section 3.6.6 in How to Use the NAG Library and its Documentation for further information.

**NE\_INVALID\_FORMAT**

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* =  $\langle value \rangle$ .

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

**NE\_INVALID\_OPTION**

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

**NE\_INVALID\_VALUE**

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

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

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

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

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*  $> \langle value \rangle$ .

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*  $\geq \langle value \rangle$ .

## NE\_NO\_LICENCE

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

See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

## NE\_PHASE

The options cannot be modified in this phase.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

nag\_opt\_handle\_opt\_set (e04zmc) is not threaded in any implementation.

## 9 Further Comments

None.

## 10 Example

See the example programs associated with the solvers nag\_opt\_handle\_solve\_ipopt (e04stc) and nag\_opt\_handle\_solve\_pennon (e04svc) for a demonstration of how to use nag\_opt\_handle\_opt\_set (e04zmc). See also Section 10 in nag\_opt\_handle\_init (e04rac) for links to all examples in this suite.

---