

## NAG Library Function Document

### nag\_opt\_sparse\_convex\_qp\_option\_set\_integer (e04ntc)

#### 1 Purpose

nag\_opt\_sparse\_convex\_qp\_option\_set\_integer (e04ntc) may be used to supply individual Integer optional arguments to nag\_opt\_sparse\_convex\_qp\_solve (e04nqc). The initialization function nag\_opt\_sparse\_convex\_qp\_init (e04npc) **must** have been called before calling nag\_opt\_sparse\_convex\_qp\_option\_set\_integer (e04ntc).

#### 2 Specification

```
#include <nag.h>
#include <nage04.h>

void nag_opt_sparse_convex_qp_option_set_integer (const char *string,
        Integer ivalue, Nag_E04State *state, NagError *fail)
```

#### 3 Description

nag\_opt\_sparse\_convex\_qp\_option\_set\_integer (e04ntc) may be used to supply values for Integer optional arguments to nag\_opt\_sparse\_convex\_qp\_solve (e04nqc). It is only necessary to call nag\_opt\_sparse\_convex\_qp\_option\_set\_integer (e04ntc) for those arguments whose values are to be different from their default values. One call to nag\_opt\_sparse\_convex\_qp\_option\_set\_integer (e04ntc) sets one argument value.

Each Integer optional argument 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;
e04ntc ("Iterations", itnlim, &state, &fail);
```

Optional argument settings are preserved following a call to nag\_opt\_sparse\_convex\_qp\_solve (e04nqc) and so the keyword **Defaults** is provided to allow you to reset all the optional arguments to their default values before a subsequent call to nag\_opt\_sparse\_convex\_qp\_solve (e04nqc).

A complete list of optional arguments, their abbreviations, synonyms and default values is given in Section 12 in nag\_opt\_sparse\_convex\_qp\_solve (e04nqc).

#### 4 References

None.

#### 5 Arguments

- 1: **string** – const char \* *Input*  
*On entry:* a single valid keyword of an Integer optional argument (as described in Section 12 in nag\_opt\_sparse\_convex\_qp\_solve (e04nqc)).
- 2: **ivalue** – Integer *Input*  
*On entry:* an Integer value associated with the keyword in **string**.
- 3: **state** – Nag\_E04State \* *Communication Structure*  
**state** contains internal information required for functions in this suite. It must not be modified in any way.

4: **fail** – NagError \*

*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_ALLOC\_FAIL

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

### NE\_BAD\_PARAM

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

### NE\_E04\_OPTION\_INVALID

The supplied option is invalid. Check that the keywords are neither ambiguous nor misspelt. The option string is ' $\langle value \rangle$ ' and **ivalue** =  $\langle value \rangle$ .

### NE\_E04NPC\_NOT\_INIT

The initialization function `nag_opt_sparse_convex_qp_init` (e04npc) has not been called.

### 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 the Essential Introduction for further information.

### NE\_NO\_LICENCE

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

See Section 3.6.5 in the Essential Introduction for further information.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

Not applicable.

## 9 Further Comments

`nag_opt_sparse_convex_qp_option_set_file` (e04nrc) or `nag_opt_sparse_convex_qp_option_set_string` (e04nsc) may also be used to supply Integer optional arguments to `nag_opt_sparse_convex_qp_solve` (e04nqc).

## 10 Example

See Section 10 in `nag_opt_sparse_convex_qp_solve` (e04nqc) and `nag_opt_sparse_convex_qp_option_set_file` (e04nrc).

---