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

```c
#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:

```c
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.

Mark 25
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 \(<value>\) 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 \('<value>\)' and \(i\text{value} = <value>\).

**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).