NAG Library Function Document

nag_opt_handle_opt_get (e04znc)

1   Purpose

nag_opt_handle_opt_get (e04znc) is an option getting routine for solvers from the NAG optimization modelling suite, namely nag_opt_handle_solve_ipopt (e04stc) and nag_opt_handle_solve_pennon (e04svc). It is used to query the value of optional parameters.

2   Specification

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

void nag_opt_handle_opt_get (void *handle, const char *optstr,
     Integer *ivalue, double *rvalue, char *cvalue, Integer lcvalue,
     Nag_VariableType *optype, NagError *fail)
```

3   Description

nag_opt_handle_opt_get (e04znc) is used to query the current values of options. It can be especially useful to retrieve the optional parameters left for automatic choice by the solver.

This function will normally return either an integer, real or character value dependent upon the type associated with the optional parameter being queried. This is indicated by the returned value of **optype**.

Information relating to available option names 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 whose current value is required. See the documentation of the particular solver for information on valid options.

3:     **ivalue** – Integer *   *Output*

   On exit: if the optional parameter supplied in **optstr** is an integer-valued argument, **ivalue** will hold its current value.

4:     **rvalue** – double *   *Output*

   On exit: if the optional parameter supplied in **optstr** is a real-valued argument, **rvalue** will hold its current value.

5:    **cvalue** – char *   *Output*

Note: the string returned in **cvalue** will never exceed $\min(\textbf{lcvalue}, 41)$ characters in length, including the terminating **NULL** character.

On exit: if the optional parameter supplied in **optstr** is a character-valued argument, **cvalue** will hold its current value.

6:    **lcvalue** – Integer   *Input*

On entry: length of the string **cvalue**. At most $\textbf{lcvalue} - 1$ non-null characters will be written into **cvalue**.

Constraint: $\textbf{lcvalue} > 1$.

7:    **optype** – Nag_VariableType *   *Output*

On exit: indicates whether the optional parameter supplied in **optstr** is an integer, real or character-valued argument and hence which of **ivalue**, **rvalue** or **cvalue** holds the current value.

optype = Nag_Integer
> **optstr** is an integer-valued optional parameter; its current value has been returned in **ivalue**.

optype = Nag_Real
> **optstr** is a real-valued optional parameter; its current value has been returned in **rvalue**.

optype = Nag_Character
> **optstr** is a character-valued optional parameter; its current value has been returned in **cvalue**.

8:    **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 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_BAD_PARAM

> On entry, argument ⟨*value*⟩ 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_INT

> On entry, **lcvalue** = ⟨*value*⟩.
> Constraint: $\textbf{lcvalue} > 1$.

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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_INVALID_OPTION

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

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NW_TRUNCATED

On entry, **optstr** indicates a character optional parameter, but **cvalue** is too short to hold the stored value. The returned value will be truncated.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_opt_handle_opt_get (e04znc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

See the example program associated with the solver nag_opt_handle_solve_pennon (e04svc) for a demonstration of how to use nag_opt_handle_opt_get (e04znc) to query options. See also Section 10 in nag_opt_handle_init (e04rac) for links to all examples in the suite.