NAG Library Function Document

nag_quad_opt_get (d01zlc)

1 Purpose

nag_quad_opt_get (d01zlc) is used to query the current value associated with an optional argument for nag_quad_1d_gen_vec_multi_rcomm (d01rac).

2 Specification

```
#include <nag.h>
#include <nagd01.h>
void nag_quad_opt_get (const char *optstr, Integer *ivalue, double *rvalue,
     char *cvalue, Integer lcvalue, Nag_VariableType *optype,
     const Integer iopts[], const double opts[], NagError *fail)
```

3 Description

nag_quad_opt_get (d01zlc) is used to query the current value associated with optional arguments. It is necessary to initialize optional argument arrays, **iopts** and **opts**, using nag_quad_opt_set (d01zkc) before any optional arguments are queried.

nag_quad_opt_get (d01zlc) will normally return either an integer, real or character value dependent upon the type associated with the optional argument being queried. Some real and integer optional arguments also return additional information in **cvalue**. Whether the optional argument queried is of integer, real or character type, and whether additional information is returned in **cvalue**, is indicated by the returned value of **optype**.

Information on optional argument names and whether these options are real, integer or character can be found in Section 11 in nag_quad_1d_gen_vec_multi_rcomm (d01rac).

4 References

None.

5 Arguments

1: **optstr** – const char * *Input*

On entry: a string identifying the option whose current value is required. See Section 11 in nag_quad_1d_gen_vec_multi_rcomm (d01rac) for information on valid optional arguments. In addition, the following is a valid option:

Identify

In which case nag_quad_opt_get (d01zlc) returns in **cvalue** the 6 character function name supplied to nag_quad_opt_set (d01zkc) when the optional argument arrays **iopts** and **opts** were initialized.

2: **ivalue** – Integer * *Output*

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

3: **rvalue** – double * *Output*

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

4: **cvalue** – char * *Output*

Note: the string returned in **cvalue** will never exceed $\min(\mathbf{lcvalue}, 41)$ characters in length (including the null terminator).

On exit: if the optional argument supplied in **optstr** is a character valued argument, **cvalue** will hold that value. **cvalue** will also contain additional information for some integer and real valued arguments, as indicated by **optype**.

5: **lcvalue** – Integer *Input*

On entry: length of **cvalue**. At most $\min(\mathbf{lcvalue} - 1, 40)$ non-null characters will be written into **cvalue**.

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

6: **optype** – Nag_VariableType * *Output*

On exit: indicates whether the optional argument 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 argument; its current value has been returned in **ivalue**.

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

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

optype = Nag_Integer_Additional
: **optstr** is an integer valued optional argument; its current value has been returned in **ivalue**. Additional information has been returned in **cvalue**.

optype = Nag_Real_Additional
: **optstr** is a real valued optional argument; its current value has been returned in **rvalue**. Additional information has been returned in **cvalue**.

7: **iopts**[*dim*] – const Integer *Communication Array*

Note: the dimension, *dim*, of this array is dictated by the requirements of associated functions that must have been previously called. This array MUST be the same array passed as argument **iopts** in the previous call to nag_quad_opt_set (d01zkc).

8: **opts**[*dim*] – const double *Communication Array*

Note: the dimension, *dim*, of this array is dictated by the requirements of associated functions that must have been previously called. This array MUST be the same array passed as argument **opts** in the previous call to nag_quad_opt_set (d01zkc).

9: **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.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

On entry, **lcvalue** $= \langle value \rangle$.
Constraint: **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.

NE_INVALID_OPTION

On entry, the optional argument in **optstr** was not recognized: **optstr** = "⟨*value*⟩".

The arrays **iopts** and **opts** have either not been initialized, have become corrupted, or are not compatible with this option setting function.

NW_TRUNCATED

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

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

See the example programs associated with the problem solving function you wish to use for a demonstration of how to use nag_quad_opt_get (d01zlc).