

# 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 parameter for nag\_quad\_md\_sgq\_multi\_vec (d01esc) and nag\_quad\_ld\_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 parameters. It is necessary to initialize optional parameter arrays, **iopts** and **opts**, using nag\_quad\_opt\_set (d01zkc) before any optional parameters 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 parameter being queried. Some real and integer optional parameters also return additional information in **cvalue**. Whether the optional parameter 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 parameter names and whether these options are real, integer or character can be found in Section 11 in nag\_quad\_md\_sgq\_multi\_vec (d01esc) and nag\_quad\_ld\_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\_md\_sgq\_multi\_vec (d01esc) and nag\_quad\_ld\_gen\_vec\_multi\_rcomm (d01rac) for information on valid optional parameters. 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 parameter arrays **iopts** and **opts** were initialized.

2: **ivalue** – Integer \* *Output*

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

- 3: **rvalue** – double \* *Output*  
*On exit:* if the optional parameter 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  $\min(\mathbf{lvalue}, 41)$  characters in length (including the null terminator).  
*On exit:* if the optional parameter 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: **lvalue** – Integer *Input*  
*On entry:* length of **cvalue**. At most  $\min(\mathbf{lvalue} - 1, 40)$  non-null characters will be written into **cvalue**.  
*Constraint:* **lvalue** > 1.
- 6: **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**.  
**optype** = Nag\_Integer\_Additional  
**optstr** is an integer valued optional parameter; 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 parameter; 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 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\_INT

On entry, **lvalue** =  $\langle value \rangle$ .

Constraint: **lvalue** > 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 3.6.6 in How to Use the NAG Library and its Documentation for further information.

### NE\_INVALID\_OPTION

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

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

### 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.

### 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\_quad\_opt\_get (d01zlc) is not threaded in any implementation.

## 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).

---