

NAG Library Function Document

nag_quad_opt_set (d01zkc)

1 Purpose

nag_quad_opt_set (d01zkc) either initializes or resets the optional parameter arrays or sets a single optional parameter for supported problem solving functions in Chapter d01.

2 Specification

```
#include <nag.h>
#include <nagd01.h>

void nag_quad_opt_set (const char *optstr, Integer iopts[], Integer liopts,
                      double opts[], Integer lopts, NagError *fail)
```

3 Description

nag_quad_opt_set (d01zkc) has three purposes: to initialize optional parameter arrays; to reset all optional parameters to their default values; or to set a single optional parameter to a user-supplied value.

Optional parameters and their values are, in general, presented as a character string, **optstr**, of the form '*option* = *optval*'; alphabetic characters can be supplied in either upper or lower case. Both *option* and *optval* may consist of one or more tokens separated by white space. The tokens that comprise *optval* will normally be either an integer, real or character value as defined in the description of the specific optional argument. In addition all optional parameters can take an *optval* DEFAULT which resets the optional parameter to its default value.

It is imperative that optional parameter arrays are initialized before any options are set, before the relevant problem solving function is called and before any options are queried using nag_quad_opt_get (d01zlc). To initialize the optional parameter arrays **iopts** and **opts** for a specific problem solving function, the option **Initialize** is used with *optval* identifying the problem solving function to be called, via its short name. For example, to initialize the optional parameter arrays to be passed to nag_quad_ld_gen_vec_multi_rcomm (d01rac) and its associated function nag_quad_ld_gen_vec_multi_dimreq (d01rcc), nag_quad_opt_set (d01zkc) is called as follows:

```
nag_quad_opt_set("Initialize = d01rac", iopts, liopts, opts, lopts, &fail)
```

The available option names and their corresponding valid values are given 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 to be set.
Initialize = *function name*
Initialize the optional parameter arrays **iopts** and **opts** for use with function *function name*, where *function name* is the short name associated with the function of interest.
Defaults
Resets all options to their default values.
option = *optval*
See Section 11 in nag_quad_md_sgq_multi_vec (d01esc) and nag_quad_1d_gen_vec_multi_rcomm (d01rac) for details of valid values for *option* and *optval*. The equals sign (=) delimiter must be used to separate the *option* from its *optval* value.
optstr is case insensitive. Each token in the *option* and *optval* component must be separated by at least one space.
- 2: **iopts**[**liopts**] – Integer *Communication Array*
On entry: optional parameter array.
If **optstr** has the form **Initialize** = *function name*, the contents of **iopts** need not be set.
Otherwise, **iopts** MUST NOT have been altered since the last call to nag_quad_opt_set (d01zkc), nag_quad_opt_get (d01zlc) or the selected problem solving function.
On exit: dependent on the contents of **optstr**, either an initialized, reset or updated version of the optional parameter array.
- 3: **liopts** – Integer *Input*
On entry: the length of the array **iopts**.
Constraint: unless otherwise stated in the documentation for a specific, supported, problem solving function, **liopts** \geq 100.
- 4: **opts**[**lopts**] – double *Communication Array*
On entry: optional parameter array.
If **optstr** has the form **Initialize** = *function name*, the contents of **opts** need not be set.
Otherwise, **opts** MUST NOT have been altered since the last call to nag_quad_opt_set (d01zkc), nag_quad_opt_get (d01zlc) or the selected problem solving function.
On exit: dependent on the contents of **optstr**, either an initialized, reset or updated version of the optional parameter array.
- 5: **lopts** – Integer *Input*
On entry: the length of the array **opts**.
Constraint: unless otherwise stated in the documentation for a specific, supported, problem solving function, **lopts** \geq 100.
- 6: **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 $\langle value \rangle$ had an illegal value.

NE_INT

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

Constraint: **liopts** $\geq \langle value \rangle$.

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

Constraint: **lopts** $\geq \langle value \rangle$.

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_FORMAT

On entry, could not convert the specified *optval* to an integer: **optstr** = $\langle value \rangle$.

On entry, could not convert the specified *optval* to a real: **optstr** = $\langle value \rangle$.

On entry, the expected delimiter '=' was not found in **optstr**: **optstr** = $\langle value \rangle$.

NE_INVALID_OPTION

On entry, either the option arrays have not been initialized or they have been corrupted.

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

NE_INVALID_VALUE

On entry, the *optval* supplied for the character optional parameter is not valid.

optstr = $\langle value \rangle$.

On entry, the *optval* supplied for the integer optional parameter is not valid.

optstr = $\langle value \rangle$.

On entry, the *optval* supplied for the real optional parameter is not valid.

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.

NE_NOT_FUN_NAME

On entry, attempting to initialize the optional parameter arrays but specified function name was not valid: name = $\langle value \rangle$.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_quad_opt_set (d01zkc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the x06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

For suites of functions that share the same option arrays, the option arrays must be initialized using the primary (driver) function name. For example for functions nag_quad_1d_gen_vec_multi_rcomm (d01rac) and nag_quad_1d_gen_vec_multi_dimreq (d01rcc), the option arrays must be initialized for nag_quad_1d_gen_vec_multi_rcomm (d01rac).

When encoding integer valued options in **optstr**, the integer *optval* must be written as an explicit integer. For example, "Maximum Subdivisions = 12" is acceptable, whereas "Maximum Subdivisions = 12.0" and "Maximum Subdivisions = 0.12e2" are not.

When encoding real valued options in **optstr**, the *optval* may be integral if appropriate. For example, "Absolute Tolerance = 10", "Absolute Tolerance = 10.0" and "Absolute Tolerance = 1.0e1" are all acceptable.

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_set (d01zkc) to initialize option arrays and set options.
