

NAG Library Function Document

nag_glopt_opt_get (e05zlc)

1 Purpose

nag_glopt_opt_get (e05zlc) is used to query the value of optional parameters available to supported problem solving functions in Chapter e05. Currently the following routines are supported:

```
nag_glopt_bnd_pso (e05sac),
nag_glopt_nlp_pso (e05sbc),
nag_glopt_nlp_multistart_sqp (e05ucc),
nag_glopt_nlp_multistart_sqp_lsq (e05usc).
```

2 Specification

```
#include <nag.h>
#include <nage05.h>

void nag_glopt_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_glopt_opt_get (e05zlc) is used to query the current values of options. It is necessary to initialize optional parameter arrays using nag_glopt_opt_set (e05zkc) before any options are queried.

nag_glopt_opt_get (e05zlc) 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 options also return additional information in **cvalue**. Whether the option 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 12 in nag_glopt_bnd_pso (e05sac), nag_glopt_nlp_pso (e05sbc), nag_glopt_nlp_multistart_sqp (e05ucc) and nag_glopt_nlp_multistart_sqp_lsq (e05usc).

4 References

None.

5 Arguments

1: **optstr** – const char * *Input*

On entry: a string identifying the option whose current value is required. See Section 12 in nag_glopt_bnd_pso (e05sac), nag_glopt_nlp_pso (e05sbc) and nag_glopt_nlp_multistart_sqp (e05ucc) for information on valid options. In addition, the following is a valid option:

Identify

nag_glopt_opt_get (e05zlc) returns in **cvalue** the function name supplied to nag_glopt_opt_set (e05zkc) 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 its current value.
- 3: **rvalue** – double * *Output*
On exit: if the optional parameter supplied in **optstr** is a real valued argument, **rvalue** will hold its current value.
- 4: **cvalue** – char * *Output*
Note: the string returned in **cvalue** will never exceed min(**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 its current 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 **lvalue** – 1 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_glopt_opt_set (e05zkc).
- 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_glopt_opt_set (e05zkc).
- 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 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, **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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_INVALID_OPTION

On entry, the *option* in **optstr** has not been recognized.

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 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_glopt_opt_get (e05zlc) is not threaded in any implementation.

9 Further Comments

Some options have default values which are problem dependent. For example the option **Maximum Iterations Completed** for nag_glopt_bnd_pso (e05sac) has the default value $1000 \times \mathbf{ndim}$. If options such as this are queried before being set, or before the problem solving function has been called, they will return misleading information in **ivalue** or **rvalue**. In some cases, the value of **cvalue** will be set to DEFAULT to indicate that the real or integer valued optional parameter supplied in **optstr** is at its default value.

10 Example

See the example programs associated with the problem solving function you wish to use for a demonstration of how to use `nag_glopt_opt_get` (e05zlc) to query options.
