

NAG Library Function Document

nag_opt_nlp_revcomm_option_set_string (e04uec)

1 Purpose

To supply individual optional parameters to nag_opt_nlp_revcomm (e04ufc).

2 Specification

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

void nag_opt_nlp_revcomm_option_set_string (const char *str,
      Nag_Boolean lwsav[], Integer iwsav[], double rwsav[], NagError *fail)
```

3 Description

nag_opt_nlp_revcomm_option_set_string (e04uec) may be used to supply values for optional parameters to nag_opt_nlp_revcomm (e04ufc). It is only necessary to call nag_opt_nlp_revcomm_option_set_string (e04uec) for those arguments whose values are to be different from their default values. One call to nag_opt_nlp_revcomm_option_set_string (e04uec) sets one argument value.

Each optional parameter is defined by a single character string, consisting of one or more items. The items associated with a given option must be separated by spaces, or equals signs [=]. Alphabetic characters may be upper or lower case. The string

```
Print Level = 1
```

is an example of a string used to set an optional parameter. For each option the string contains one or more of the following items:

- a mandatory keyword;
- a phrase that qualifies the keyword;
- a number that specifies an Integer or double value. Such numbers may be up to 40 contiguous characters which can be read using C's d or g formats, terminated by a space if this is not the last item on the line.

Blank strings and comments are ignored. A comment begins with an asterisk (*) and all subsequent characters in the string are regarded as part of the comment.

Optional parameter settings are preserved following a call to nag_opt_nlp_revcomm (e04ufc) and so the keyword **Defaults** is provided to allow you to reset all the optional parameters to their default values before a subsequent call to nag_opt_nlp_revcomm (e04ufc).

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in nag_opt_nlp_revcomm (e04ufc).

4 References

None.

5 Arguments

- 1: **str** – const char * *Input*
On entry: a single valid option string (as described in Section 3 and in Section 12 in nag_opt_nlp_revcomm (e04ufc)).

- 2: **lwsav**[120] – Nag_Boolean *Communication Array*
 3: **iwsav**[610] – Integer *Communication Array*
 4: **rwsav**[475] – double *Communication Array*

The arrays **lwsav**, **iwsav** and **rwsav** MUST NOT be altered between calls to any of the functions `nag_opt_nlp_revcomm_option_set_file` (e04udc), `nag_opt_nlp_revcomm_option_set_string` (e04uec), `nag_opt_nlp_revcomm` (e04ufc) or `nag_opt_nlp_revcomm_init` (e04wbc).

- 5: **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 *<value>* had an illegal value.

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

The supplied option string is invalid. Supplied value was: *<value>*.

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.

7 Accuracy

Not applicable.

8 Parallelism and Performance

`nag_opt_nlp_revcomm_option_set_string` (e04uec) is not threaded in any implementation.

9 Further Comments

`nag_opt_nlp_revcomm_option_set_file` (e04udc) may also be used to supply optional parameters to `nag_opt_nlp_revcomm` (e04ufc).

10 Example

See Section 10 in `nag_opt_nlp_revcomm_option_set_file` (e04udc).
