

# NAG Library Function Document

## nag\_fit\_opt\_set (e02zkc)

### 1 Purpose

nag\_fit\_opt\_set (e02zkc) either initializes or resets the optional parameter arrays or sets a single optional parameter for supported problem solving functions in Chapter e02. Currently, only nag\_2d\_spline\_fit\_ts\_scatter (e02jdc) is supported.

### 2 Specification

```
#include <nag.h>
#include <nage02.h>
void nag_fit_opt_set (const char *optstr, Integer iopts[], Integer liopts,
                    double opts[], Integer lopts, NagError *fail)
```

### 3 Description

nag\_fit\_opt\_set (e02zkc) 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\_fit\_opt\_get (e02zlc). 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 optional parameter arrays to be passed to nag\_2d\_spline\_fit\_ts\_scatter (e02jdc), nag\_fit\_opt\_set (e02zkc) is called as follows:

```
nag_fit_opt_set("Initialize = e02jdc", iopts, liopts, opts, lopts, &fail);
```

Information relating to available option names and their corresponding valid values is given in Section 11 in nag\_2d\_spline\_fit\_ts\_scatter (e02jdc).

### 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 of the problem solving function you wish to use.  
**Defaults**  
 Resets all options to their default values.  
*option* = *optval*  
 See Section 11 in nag\_2d\_spline\_fit\_ts\_scatter (e02jdc) for details of valid values for *option* and *optval*. The equals sign (=) delimiter must be used to separate the *option* from its *optval*.  
 The processing of **optstr** does not depend on its case. 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\_fit\_opt\_set (e02zkc), nag\_fit\_opt\_get (e02zlc) or the selected problem solving function or suite of functions.  
*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\_fit\_opt\_set (e02zkc), nag\_fit\_opt\_get (e02zlc) or the selected problem solving function or suite of functions.  
*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 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, **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 3.6.6 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 3.6.5 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\_fit\_opt\_set (e02zkc) 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

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\_fit\_opt\_set (e02zkc) to initialize option arrays and set options.

---