

# NAG Library Routine Document

## D01ZKF

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

### 1 Purpose

D01ZKF either initializes or resets the optional parameter arrays or sets a single optional parameter for supported problem solving routines in Chapter D01.

### 2 Specification

```
SUBROUTINE D01ZKF (OPTSTR, IOPTS, LIOPTS, OPTS, LOPTS, IFAIL)
INTEGER          IOPTS(LIOPTS), LIOPTS, LOPTS, IFAIL
REAL (KIND=nag_wp) OPTS(LOPTS)
CHARACTER(*)     OPTSTR
```

### 3 Description

D01ZKF 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 routine is called and before any options are queried using D01ZLF. To initialize the optional parameter arrays IOPTS and OPTS for a specific problem solving routine, the option **Initialize** is used with *optval* identifying the problem solving routine to be called, via its short name. For example, to initialize the optional parameter arrays to be passed to D01RAF, and its associated routines D01RBF and D01RCF, D01ZKF is called as follows:

```
call D01ZKF('Initialize = d01raf', IOPTS, LIOPTS, OPTS, LOPTS, IFAIL)
```

The available option names and their corresponding valid values are given in Section 10 in D01RAF.

### 4 References

None.

### 5 Parameters

1: OPTSTR – CHARACTER(\*) *Input*

*On entry:* a string identifying the option to be set.

**Initialize** = *routine name*

Initialize the optional parameter arrays IOPTS and OPTS for use with routine *routine name*, where *routine name* is the short name associated with the routine of interest. The associated long name may also be used, however only the short name will be returned if D01ZLF is called with OPTSTR = **Identify**.

**Defaults**

Resets all options to their default values.

**Option = optval**

See Section 10 in D01RAF 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 array *Communication Array*

*On entry:* optional parameter array.

If OPTSTR has the form **Initialize** = *routine name*, the contents of IOPTS need not be set.

Otherwise, IOPTS **must not** have been altered since the last call to D01ZKF, D01ZLF or the selected problem solving routine.

*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 routine,  $LIOPTS \geq 100$ .

- 4: OPTS(LOPTS) – REAL (KIND=nag\_wp) array *Communication Array*

*On entry:* optional parameter array.

If OPTSTR has the form **Initialize** = *routine name*, the contents of OPTS need not be set.

Otherwise, OPTS **must not** have been altered since the last call to D01ZKF, D01ZLF or the selected problem solving routine.

*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 routine,  $LOPTS \geq 100$ .

- 6: IFAIL – INTEGER *Input/Output*

*On entry:* IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

*On exit:* IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

## 6 Error Indicators and Warnings

If on entry  $IFAIL = 0$  or  $-1$ , explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$IFAIL = 11$

On entry, the **option** in OPTSTR has not been recognized. OPTSTR =  $\langle value \rangle$ .

$IFAIL = 12$

On entry, OPTSTR did not contain the delimiter '='.

$IFAIL = 13$

On entry, the *optval* present in OPTSTR could not be converted into the required numerical value. One of the following may apply:

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 *optval*,  $\langle value \rangle$ , associated with OPTSTR was partially recognized as a numerical value. However, it could not be fully interpreted.

The **option** in OPTSTR is associated with a numerical value. However, the *optval*,  $\langle value \rangle$ , present in OPTSTR is non-numerical.

$IFAIL = 14$

On entry, the **option** in OPTSTR has been detected as **Initialize**, however the *optval*,  $\langle value \rangle$ , associated with OPTSTR has not been recognized as a valid routine name.

$IFAIL = 15$

On entry, the integer *optval* associated with the optional parameter is not valid. One of the following may apply:

On entry, the absolute value of the integer present in OPTSTR is above the maximum allowable value.

$|ivalue| = \langle value \rangle$ ,  $|ivalue| \leq \langle value \rangle$  required.

On entry, the absolute value of the integer present in OPTSTR is below the minimum required value.

$|ivalue| = \langle value \rangle$ ,  $|ivalue| \geq \langle value \rangle$  required.

On entry, the integer value present in OPTSTR is above the maximum allowable value.

$ivalue = \langle value \rangle$ ,  $ivalue \leq \langle value \rangle$  required.

On entry, the integer value present in OPTSTR is below the minimum allowable value.

$ivalue = \langle value \rangle$ ,  $ivalue \geq \langle value \rangle$  required.

On entry, the integer value present in OPTSTR is outside of allowable bounds.

$ivalue = \langle value \rangle$ ,  $\langle value \rangle \leq ivalue \leq \langle value \rangle$  required.

On entry, the integer value present in OPTSTR is outside the allowable set.

$ivalue = \langle value \rangle$ .

$IFAIL = 16$

On entry, the real *optval* associated with the optional parameter is not valid. One of the following may apply:

On entry, the magnitude of the real value present in OPTSTR is above the maximum allowable value.

$|rvalue| = \langle value \rangle$ ,  $|rvalue| \leq \langle value \rangle$  required.

On entry, the magnitude of the real value present in OPTSTR is below the minimum required.

$|rvalue| = \langle value \rangle$ ,  $|rvalue| \geq \langle value \rangle$  required.

On entry, the real value present in OPTSTR is above the maximum allowable value.

$rvalue = \langle value \rangle$  and  $rvalue < \langle value \rangle$  required.

On entry, the real value present in OPTSTR is below the minimum required value.

$rvalue = \langle value \rangle$ ,  $rvalue > \langle value \rangle$  required.

On entry, the real value present in OPTSTR is outside allowable bounds.

$rvalue = \langle value \rangle$ ,  $\langle value \rangle \leq rvalue \leq \langle value \rangle$  required.

On entry, the real value present in OPTSTR is outside the allowable set.

$rvalue = \langle value \rangle$ .

#### IFAIL = 17

On entry, the character *optval* associated with the optional parameter is not valid. One of the following may apply:

On entry, the *optval* in OPTSTR is not compatible with the **option** in OPTSTR.  
OPTSTR =  $\langle value \rangle$ .

The **option** = *optval* pairing present in OPTSTR is not compatible with the routine for which the option arrays were initialized.

#### IFAIL = 21

Either the option arrays have not been initialized, have been corrupted, or are not compatible with this option setting routine.

On entry, the optional parameter arrays IOPTS and OPTS have not been initialized, or have been corrupted.

#### IFAIL = 31

LIOPTS is insufficient for the chosen routine name.

On entry, LIOPTS =  $\langle value \rangle$ .

Constraint: LIOPTS  $\geq \langle value \rangle$ .

#### IFAIL = 51

LOPTS is insufficient for the chosen routine name.

On entry, LOPTS =  $\langle value \rangle$ .

Constraint: LOPTS  $\geq \langle value \rangle$ .

#### IFAIL = -999

Dynamic memory allocation failed.

## 7 Accuracy

Not applicable.

## 8 Further Comments

For suites of routines that share the same option arrays, the option arrays must be initialized using the primary (driver) routine name.

For example for the suite of routines D01RAF, D01RBF and D01RCF, the option arrays must be initialized for D01RAF.

## **9 Example**

See the example programs associated with the problem solving routine you wish to use for a demonstration of how to use D01ZKF to initialize option arrays and set options.

---