Module 1.1: nag_lib_support
Library Support Facilities

`nag_lib_support` provides support facilities for the Library.

Contents

Procedures

Procedure: nag_lib_ident

1 Description

nag_lib_ident prints details of the implementation of the Library.

2 Usage

```
USE nag_lib_support
CALL nag_lib_ident
```

3 Arguments

None.

4 Error Codes

None.

5 Examples of Usage

The program

```
PROGRAM nag_lib_support_ex01

  ! Example Program Text for nag_lib_support
  ! NAG f190, Release 4. NAG Copyright 2000.

  ! .. Use Statements ..
  USE nag_lib_support, ONLY : nag_lib_ident
  ! .. Implicit None Statement ..
  IMPLICIT NONE
  ! .. Executable Statements ..
  CALL nag_lib_ident

END PROGRAM nag_lib_support_ex01
```

produces details of the Library implementation. A typical example of the output from this program might be:

```
*** Start of NAG Fortran 90 Library implementation details ***

Implementation title: Generalised Base Version
        Product Code: FNBAS04D9
             Release: 4
           Precision: double (KIND=  2)

*** End of NAG Fortran 90 Library implementation details ***
```

Procedure: nag_deallocate

1 Description

nag_deallocate deallocates storage from the pointer components of structures with types defined by the Library.

2 Usage

```
USE nag_lib_support
CALL nag_deallocate(comm)
```

2.1 Interfaces

Distinct interfaces exist, allowing the procedure to be used for an argument comm of any of the following derived data types:

nag_sparse_mat_real_*wp*:	defined by module nag_sparse_mat (4.3)
nag_sparse_mat_cmplx_*wp*:	defined by module nag_sparse_mat (4.3)
nag_pch_comm_*wp*:	defined by module nag_pch_interp (8.1)
nag_spline_1d_comm_*wp*:	defined by module nag_spline_1d (8.2)
nag_spline_2d_comm_*wp*:	defined by module nag_spline_2d (8.3)
nag_scat_comm_*wp*:	defined by module nag_scat_interp (8.4)
nag_rk_comm_*wp*:	defined by module nag_ivp_ode_rk (12.1)
nag_ref_vec_*wp*:	defined by module nag_rand_discrete (21.3)

3 Arguments

3.1 Mandatory Argument

comm — type(any of the derived types listed in Section 2.1), intent(in)

Input: the structure whose pointer components are to be deallocated.

4 Error Codes

None.

5 Examples of Usage

The module documents in which the relevant derived types are defined contain illustrations of the use of this procedure.

6 Further Comments

6.1 Access to the Procedure

This procedure is also available through the USE statement for the module which defines the type of the structure to be deallocated.