NAG Library Function Document

nag_ip_mps_free (h02bvc)

1 Purpose

nag_ip_mps_free (h02bvc) frees the memory allocated by nag_ip_mps_read (h02buc).

2 Specification

```
#include <nag.h>
#include <nagh.h>
void nag_ip_mps_free (double **a, double **bl, double **bu,
     Nag_Boolean **intvar, double **cvec, double **x)
```

3 Description

nag_ip_mps_free (h02bvc) should be used in conjunction with nag_ip_mps_read (h02buc), which reads data for an integer programming problem from an MPSX file, allocates several arrays, and populates them with the data contained in the file. nag_ip_mps_free (h02bvc) is a utility provided for the convenient freeing of this memory. It should be called in order to conserve memory which is no longer required, e.g., following a call to nag_ip_bb (h02bbc) (which may be used to solve the problem defined by the MPSX file). Any memory not freed will, of course, be freed when your program terminates.

nag_ip_mps_free (h02bvc) can be used to free a subset of the allocated arrays by passing null pointers for those arguments which you do not wish to free.

4 References

None.

5 Arguments

1: **a** – double ** *Input/Output*

On entry: the nonzeros of the constraint matrix A, to be freed. If **a** or ***a** is a null pointer, no action is taken.

On exit: if **a** is not null, ***a** is set to the null pointer.

2: **bl** – double ** *Input/Output*

On entry: the lower bounds of the problem variables and general constraints, to be freed. If **bl** or ***bl** is a null pointer, no action is taken.

On exit: if **bl** is not null, ***bl** is set to the null pointer.

3: **bu** – double ** *Input/Output*

On entry: the upper bounds of the problem variables and general constraints, to be freed. If **bu** or ***bu** is a null pointer, no action is taken.

On exit: if **bu** is not null, ***bu** is set to the null pointer.

4: **intvar** – Nag_Boolean ** *Input/Output*

On entry: the indicators as to which are the integer variables in the problem, to be freed. If **intvar** or ***intvar** is a null pointer, no action is taken.

On exit: if **intvar** is not null, ***intvar** is set to the null pointer.

5: **cvec** – double ** *Input/Output*

On entry: the coefficients, c, of the linear term of the objective function, to be freed. If **cvec** or ***cvec** is a null pointer, no action is taken.

On exit: if **cvec** is not null, ***cvec** is set to the null pointer.

6: **x** – double ** *Input/Output*

On entry: a set of initial values for the variables, to be freed. If **x** or ***x** is a null pointer no action is taken.

On exit: if **x** is not null, ***x** is set to the null pointer.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

In addition to allocating the memory freed by this function, nag_ip_mps_read (h02buc) also allocates memory to the **crnames** member of the **options** structure (if the structure is supplied as an argument). The function nag_ip_free (h02xzc) should be used to free this memory. You should **not** use the standard C function `free()` for this purpose.

10 Example

For an example of the use of nag_ip_mps_free (h02bvc) see the documentation for nag_ip_mps_read (h02buc).