NAG Library Function Document

nag_implementation_separated_details (a00adc)

1 Purpose

nag_implementation_separated_details (a00adc) prints information about the version of the NAG C Library in use.

2 Specification

```
#include <nag.h>
#include <naga00.h>
```

```
void nag_implementation_separated_details (char *impl, char *prec,
     char *pcode, char *mkmaj, char *mkmin, char *hdware, char *opsys,
     char *ccomp, char *fcomp, char *vend, Nag_Boolean *licval)
```

3 Description

The NAG C Library is available for use on a number of different computer systems. For each distinct system an implementation of the library is prepared and this implementation is given a unique code. The specifics that define the implementation are: the working precision, the major and minor marks of the NAG C Library, the target hardware and operating system, the compiler used, and the vendor library (if any) that is also required to be linked. nag_implementation_separated_details (a00adc) may be called to return, in separate arguments, these specific details of the NAG C Library implementation that is being used; it also returns whether a valid licence has been found for this implementation. This differs from nag_implementation_details (a00aac) which simply outputs the collected information in a readable form directly to the `stdout` (standard output) stream.

4 References

None.

5 Arguments

1: **impl** – char * *Output*

On exit: the implementation title which usually lists the target platform, operating system and compiler.

2: **prec** – char * *Output*

On exit: the working or basic precision of the implementation. Some functions may perform operations in reduced precision or additional precision, but the great majority will perform all operations in basic precision.

3: **pcode** – char * *Output*

On exit: the product code for the NAG C Library implementation that is being used. The code has a discernible structure, but it is not necessary to know the details of this structure. The product code can be used to differentiate between individual product licence codes.

4: **mkmaj** – char * *Output*

On exit: the major mark of the NAG C Library implementation that is being used.

5: **mkmin** – char * *Output*

On exit: the minor mark of the NAG C Library implementation that is being used.

6: **hdware** – char * *Output*

On exit: the target hardware for the NAG C Library implementation that is being used.

7: **opsys** – char * *Output*

On exit: the target operating system for the NAG C Library implementation that is being used.

8: **ccomp** – char * *Output*

On exit: the C compiler used to build the NAG C Library implementation that is being used.

9: **fcomp** – char * *Output*

On exit: the Fortran compiler used to build the NAG C Library implementation that is being used.

10: **vend** – char * *Output*

On exit: the subsidiary library, if any, that must be linked with the NAG C Library implementation that is being used. If the implementation does not require a subsidiary library then the string

```
'(self-contained)'
```

will be returned in **vend**.

11: **licval** – Nag_Boolean * *Output*

On exit: specifies whether or not a valid licence has been found for the NAG C Library implementation that is being used.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example makes a call of nag_implementation_separated_details (a00adc), collects information on the NAG C Library implementation that is being used and prints it out in a form that is similar to the output obtained by a call to nag_implementation_details (a00aac).

10.1 Program Text

```
/* nag_implementation_separated_details (a00adc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 9, 2009.
 */

#include <nag.h>
#include <stdio.h>
#include <string.h>
#include <nag_stdlib.h>
#include <naga00.h>

int main(void)
{
  int         exit_status = 0;
  int         max_char_len = 180;
  char        *impl = 0, *prec = 0, *pcode = 0, *mkmaj = 0, *mkmin = 0,
              *hdware = 0, *opsys = 0, *ccomp = 0, *fcomp = 0, *vend = 0;
  Nag_Boolean licval;

  printf("nag_implementation_separated_details (a00adc)"
          " Example Program Results\n\n");

  if (!(impl = NAG_ALLOC(max_char_len, char)) ||
      !(prec = NAG_ALLOC(max_char_len, char)) ||
      !(pcode = NAG_ALLOC(max_char_len, char)) ||
      !(mkmaj = NAG_ALLOC(max_char_len, char)) ||
      !(mkmin = NAG_ALLOC(max_char_len, char)) ||
      !(hdware = NAG_ALLOC(max_char_len, char)) ||
      !(opsys = NAG_ALLOC(max_char_len, char)) ||
      !(ccomp = NAG_ALLOC(max_char_len, char)) ||
      !(fcomp = NAG_ALLOC(max_char_len, char)) ||
      !(vend = NAG_ALLOC(max_char_len, char)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  nag_implementation_separated_details(impl, prec, pcode, mkmaj, mkmin, hdware,
                                       opsys, ccomp, fcomp, vend, &licval);

  /* Print implementation details. */
  printf("*** Start of NAG C library implementation details ***\n\n");
  printf(" Implementation title: %s\n", impl);
  printf("            Precision: %s\n", prec);
  printf("         Product Code: %s\n", pcode);
  printf("                 Mark: %s.%s\n", mkmaj, mkmin);
  if (!strcmp(vend, "(self-contained)"))
    {
      printf("       Vendor library: None\n");
    }
  else
    {
      printf("       Vendor library: %s\n", vend);
    }
  printf(" Applicable to:\n");
  printf("             hardware: %s\n", hdware);
  printf("     operating system: %s\n", opsys);
  printf("           C compiler: %s\n", ccomp);
  printf("     FORTRAN compiler: %s\n", fcomp);
  printf(" and compatible systems.\n");
  if (!licval)
    {
      printf("        Licence query: %s\n\n", "Unsuccessful");
    }
  else
    {
      printf("        Licence query: %s\n\n", "Successful");
    }
  printf(" *** End of NAG C Library implementation details ***\n");
 END:

  NAG_FREE(impl);
  NAG_FREE(prec);
  NAG_FREE(pcode);
  NAG_FREE(mkmaj);
  NAG_FREE(mkmin);
  NAG_FREE(hdware);
  NAG_FREE(opsys);
  NAG_FREE(ccomp);
  NAG_FREE(fcomp);
  NAG_FREE(vend);

  return exit_status;
}
```

10.2 Program Data

None.

10.3 Program Results

```
nag_implementation_separated_details (a00adc) Example Program Results

*** Start of NAG C library implementation details ***

 Implementation title: NAG C Library
            Precision: double
         Product Code: CLL6A09D9
                 Mark: 9.0
       Vendor library: None
 Applicable to:
             hardware: x86_64
     operating system: Linux 2.6.25.10-47.fc8
           C compiler: gcc (GCC) 4.4.0 20090123 (experimental)
       FORTRAN  compiler:  NAGWare  Fortran  95  compiler  Release  5.1(347,355-
367,375,380-383
                                   389,394,399,401-402,407,431,435,437,446,459-
460,463,472,494,496
                       503,508,511,517,529,555,557,565,595)
 and compatible systems.
        Licence query: Successful

 *** End of NAG C Library implementation details ***
```