

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,
    Integer *mkmaj, Integer *mkmin, char *hardware, 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** – Integer * *Output*
On exit: the major mark of the NAG C Library implementation that is being used.

- 5: **mkmin** – Integer * *Output*
On exit: the minor mark of the NAG C Library implementation that is being used.
- 6: **hware** – 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 Further Comments

None.

9 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).

9.1 Program Text

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

#include <nag.h>
#include <nagx04.h>
#include <stdio.h>
```

```

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

int main(int argc, char *argv[])
{
    FILE          *fpout;
    int           exit_status = 0;
    int           max_char_len = 120;
    char          *impl, *prec, *pcode, *mkmaj, *mkmin, *hware, *opsys, *ccomp;
    char          *fcomp, *vend;
    Nag_Boolean   licval;
    NagError      fail;

    INIT_FAIL(fail);

    /* Check for command-line IO options */
    fpout = nag_example_file_io(argc, argv, "-results", NULL);

    fprintf(fpout, "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)) ||
        !(hware = 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)))
    {
        fprintf(fpout, "Allocation failure\n");
        exit_status = -1;
        goto END;
    }

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

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

```

```
if (fpout != stdout) fclose(fpout);

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

return exit_status;
}
```

9.2 Program Data

None.

9.3 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: 09
      Vendor library: (self-contained)
Applicable to:
      hardware: x86_64
      operating system: Linux 2.6.11.4-20a-smp
      C compiler: gcc (GCC) 4.2.0
      FORTRAN compiler: NAG Fortran Compiler Release 5.2(692)
and compatible systems.
      Licence query: Successful

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