

NAG Library Function Document

nag_implementation_details (a00aac)

1 Purpose

nag_implementation_details (a00aac) prints information about the version of the NAG C Library in use.

2 Specification

```
#include <nag.h>
#include <naga00.h>
void nag_implementation_details ()
```

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. This includes tested compiled libraries and any necessary system-specific support material. nag_implementation_details (a00aac) may be called to print the implementation details and Mark (i.e., maintenance level) of the NAG C Library implementation that is being used.

4 References

None.

5 Arguments

None.

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_details (a00aac) sending output to the current advisory message unit.

10.1 Program Text

```

/* nag_implementation_details (a00aac) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

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

int main(void)
{
    Integer exit_status = 0;
    unsigned int sizeofpointer = sizeof(void *);
    unsigned int sizeofInteger = sizeof(Integer);
    unsigned int sp, si;

    /* Get the expected sizes of pointers and integers (in bytes) */
    a00aay(&sp, &si);

    printf("nag_implementation_details (a00aac) Example Program Results\n\n");

    /* Check that the pointer and integer sizes are as expected, and
       issue a warning if not */
    if (sp != sizeofpointer || si != sizeofInteger) {
        printf("
        ~~~~~~\n\n");
        if (sp != sizeofpointer)
            printf(" Incorrect value of sizeof(void *)\n"
                   "   expected %u, returned %u.\n\n", sp, sizeofpointer);
        if (si != sizeofInteger)
            printf(" Incorrect value of sizeof(Integer)\n"
                   "   expected %u, returned %u.\n\n", si, sizeofInteger);
        printf(" The NAG C Library header files are "
               " incompatible with the NAG Library.\n\n");
        printf(" Please check the location of your "
               " NAG C Library include files.\n\n");
        printf("
        ~~~~~~\n\n");
        exit_status = 1;
    }

    nag_implementation_details();

    return exit_status;
}

```

10.2 Program Data

None.

10.3 Program Results

```

nag_implementation_details (a00aac) Example Program Results

*** Start of NAG C Library implementation details ***

Implementation title: ?OS?, ?x?-bit, ?C compiler? (?y?-bit integers)
    Precision: double precision
    Product Code: ?CL?
    Mark: ?z? (self-contained)

```

Type sizes: sizeof(Pointer) = ?P?, sizeof(Integer) = ?I?

This is a ?x?-bit library using ?y?-bit integers.

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