

# NAG Library Function Document

## nag\_code\_to\_error\_name (x04ndc)

### 1 Purpose

nag\_code\_to\_error\_name (x04ndc) converts an integer error code to its corresponding string error name. The integer error code is typically returned as part of the NAG error structure by a prior call to a NAG C Library function.

### 2 Specification

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

char * nag_code_to_error_name (int code)
```

### 3 Description

nag\_code\_to\_error\_name (x04ndc) takes as input the value of the integer error code returned in the NAG error structure and returns the corresponding string error name. This can be useful when you want to print the returned error code as a name. For example the **NE\_BAD\_PARAM** code has the value 70. A call to nag\_code\_to\_error\_name (x04ndc)(70) returns the string **'NE\_BAD\_PARAM'**.

nag\_code\_to\_error\_name (x04ndc) returns 0 (null pointer) if the input value is not recognized as a valid NAG error code.

### 4 References

None.

### 5 Arguments

1: **code** – int *Input*  
*On entry:* the value of a NAG error code.

### 6 Error Indicators and Warnings

### 7 Accuracy

Not applicable.

### 8 Further Comments

None.

### 9 Example

This example produces error name strings corresponding to a number of integer error codes.

## 9.1 Program Text

```

/* nag_code_to_error_name (x04ndc) Example Program.
 *
 * Copyright 2009 Numerical Algorithms Group.
 *
 * Mark 9, 2009.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nag_string.h>
#include <nagx04.h>

int main(int argc, char *argv[])
{
    FILE          *fpout;
    /* Scalars */
    Integer       exit_status = 0;
    unsigned      i;
    /* Pointers */
    const char    *str_error = 0;
    NagError      fail;

    int           fail_code[] = { NE_BAD_PARAM, NE_INCOMPAT_PARAM, NE_COMPLEX_ZERO,
                                  NE_NOT_MONOTONIC, NE_TOO_MANY_ITER, NE_SINGULAR,
                                  NE_INITIALIZATION };

    INIT_FAIL(fail);

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

    fprintf(fpout, "nag_code_to_error_name (x04ndc) Example Program Results\n\n");

    /* Print header. */
    fprintf(fpout, "   Code value           Error name\n");
    fprintf(fpout, "   -----           -----\n");

    /* Convert the error code to strings and print. */

    /* nag_code_to_error_name (x04ndc).
     * Converts NAG error code to its string value
     */
    for (i = 0; i < sizeof(fail_code)/sizeof(int); i++)
    {
        str_error = nag_code_to_error_name(fail_code[i]);
        if (str_error)
            fprintf(fpout, "%7d           %-s\n", fail_code[i], str_error);
        else
        {
            exit_status = 1;
            goto END;
        }
    }
    END:
    if (fpout != stdout) fclose(fpout);
    return exit_status;
}

```

## 9.2 Program Data

None.

### 9.3 Program Results

nag\_code\_to\_error\_name (x04ndc) Example Program Results

Code value	Error name
-----	-----
70	NE_BAD_PARAM
88	NE_INCOMPAT_PARAM
133	NE_COMPLEX_ZERO
245	NE_NOT_MONOTONIC
357	NE_TOO_MANY_ITER
366	NE_SINGULAR
2049	NE_INITIALIZATION

---