

From:
Barry Caplin email: infodesk@nag.com web: http://www.nag.com/
Numerical Algorithms Group Response Center, USA Technical Support
phone: 708-971-2345
fax : 708-971-2346

----- Begin Message -----

NAGWare f90 Compiler, Version 2.1

NCSU421N

Sun SPARC (SunOS 4.1)

Users' Note

Contents

1. Introduction

2. General Information

- 2.1. Accessing the Compiler
- 2.2. User Documentation
- 2.3. Random Number Algorithm
- 2.4. Additional Language Extension

3. Specific Information

- 3.1. Debugging Example
- 3.2. Runtime Garbage Collector
- 3.3. 64-bit Data Types

4. Support from NAG

5. NAG Users Association

1. Introduction

This document is essential reading for every user of the NAGWare f90 Compiler implementation specified in the title. It provides implementation-specific detail that augments the information provided in the NAGWare f90 Compiler Users' Manual.

2. General Information

2.1. Accessing the Compiler

Assuming that the compiler has been installed in a sub-directory

in your PATH environment variable, then the compiler can be accessed using the command f90. The run-time libraries and pre-defined modules are located in /usr/local/lib/f90.

2.2. User Documentation

The following material is provided on the compiler release medium. Please contact your local advisory service or NAG Site Contact for details of how you can access this documentation.

(a) general information

Users' Note (this document)

(b) example program material

An example program (f90_util.f90) which outputs the implementation specific information available through Fortran 90 enquiry intrinsics.

(c) man

Information on the NAGWare f90 Compiler has been provided in the form of a man page. For details see

man f90

Each site is supplied with one copy of the following publications, further copies of which can be ordered from NAG:

- Fortran 90 Explained by Metcalf and Reid (ISBN 0-19-853772-7)
- NAGWare f90 Compiler Users' Manual (ISBN 1-85206-105-7)

2.3. Random Number Algorithm

The random number generator supplied as the intrinsic subroutine RANDOM_NUMBER is the "good, minimal standard" generator described in "Random Number Generators: Good Ones Are Hard to Find" [CACM October 1988, Volume 31 Number 10, pp 1192-1201]. This is a parametric multiplicative linear congruential algorithm with the following parameters:

modulus: 2**31-1 (2147483647)
multiplier: 16807

This is a full-period generator. The seed is obtained from the time-of-day clock, viz time*16807+1 where "time" is the number of seconds past midnight.

2.4. Additional Language Extension

The Fortran standard prohibits exporting (making PUBLIC) entities of types which are "declared to be PRIVATE"; this also applies to procedures with dummy arguments of private type and derived type

definitions with components of derived type.

It seems clear from the standard that a type is considered to be PRIVATE if it is so declared in the current scope; however there is currently an interpretation being considered by the standard's maintenance body which could make a type be considered PRIVATE only if it is so declared in its defining scope.

In version 1.x of the NAGWare f90 compiler checking for this condition was performed only for derived type components and not for the other cases. Therefore we have decided to allow this use to continue for the time being as an extension so that our current users will not have their programs rejected. However, we recommend that this practice be discontinued as it is not portable (some other Fortran 90 compilers reject it).

When the official interpretation of this question has been resolved the compiler will be altered to agree with it, either by removing the extension warnings (if they allow it) or by turning the extension warnings into errors (if they disallow it).

3. Specific Information

The following information is implementation specific.

The -O flag on the compiler equates to the -O2 flag on cc.

Details of the interpretation of binary, octal and hex constants may be obtained from the f90_util program provided with the compiler. The following information is also available from f90_util:

INTEGER	Default	int8	int16	int32
KIND number =	3	1	2	3
digits =	31	7	15	31
radix =	2	2	2	2
range =	9	2	4	9
huge =	2147483647	127	32767	2147483647
bit_size =	32	8	16	32

LOGICAL	Default	byte	word
KIND number =	3	1	3

REAL	Default	single	double
KIND number =	1	1	2
digits =	24	24	53
maxexponent =	128	128	1024
minexponent =	-125	-125	-1021
precision =	6	6	15
radix =	2	2	2
range =	37	37	307
epsilon =	0.11920929E-06	0.11920929E-06	0.22204460E-15
tiny =	0.11754944E-37	0.11754944E-37	0.22250739-307
huge =	0.34028235E+39	0.34028235E+39	0.17976931+309

COMPLEX	Default	single	double
KIND number =	1	1	2
precision =	6	6	15
range =	37	37	307

3.1. Debugging Example

This is an example of the use of the vendor-supplied debugger, dbx, in debugging Fortran 90 code which contains COMMON blocks and modules.

The file to be debugged is called fh4.f90 and contains:

```

MODULE fh4
  REAL r
END MODULE fh4

PROGRAM fh4_prog
  USE fh4
  COMMON/fh4com/i
  i = 2
  CALL sub
  PRINT *,i,r
END PROGRAM fh4_prog

SUBROUTINE sub
  USE fh4
  COMMON/fh4com/i
  r = 0.5*i
  i = i*3
END SUBROUTINE sub

```

It is first compiled with the -g option and then run under dbx:

```

% f90 -g -o fh4 fh4
% dbx fh4
Reading symbolic information...
Read 379 symbols

```

Setting a breakpoint in routine SUB; the 'C' version of the name must be used, in this case it becomes lowercase with an appended underscore.

```

(dbx) stop in sub_
(2) stop in sub_
(dbx) run
Running: fh4
stopped in sub_ at line 16 in file "fh4.f90"
   16      r = 0.5*i

```

Printing the value of a variable in a COMMON block; if the COMMON block is "simple" (e.g. not extended with EQUIVALENCE) and consistent among program-units, access is also simple.

```
(dbx) print fh4com_.i_  
fh4com_.i_ = 2  
(dbx) next  
stopped in sub_ at line 17 in file "fh4.f90"  
  17      i = i*3
```

To access a variable in a module (or to set a breakpoint in a module procedure), the module name and the variable name are joined by the string "_MP_".

```
(dbx) print fh4_MP_r  
fh4_MP_r = 1.0  
(dbx) next  
stopped in sub_ at line 18 in file "fh4.f90"  
  18      END SUBROUTINE sub  
(dbx) print fh4com_.i_  
fh4com_.i_ = 6
```

The value of variables can also be set.

```
(dbx) set fh4com_.i_ = 7  
(dbx) cont  
  7      1.0000000  
  
execution completed, exit code is 0  
program exited with 0  
(dbx) quit  
%
```

3.2. Runtime Garbage Collector

The runtime garbage collector (-gc option) is supported on this machine by version 2.1 of the NAGWare f90 compiler. See the NAGWare f90 Compiler Users' Manual for full details on usage.

3.3. 64-bit Data Types

Version 2.1 of the NAGWare f90 compiler on this machine does not support 64-bit INTEGER or LOGICAL data types.

4. Support from NAG

(a) Contact with NAG

Queries concerning this document or the implementation generally should be directed initially to your local Advisory Service. If you have difficulty in making contact locally, you can write to NAG directly, at one of the supplied addresses. Users subscribing to the support service are encouraged to contact one of the NAG Response Centres (see below).

(b) NAG Response Centres

The NAG Response Centres are available for general enquiries from all users and also for technical queries from sites with an annually licensed product or support service.

The Response Centres are open during office hours, but contact is possible by fax, email and phone (answering machine) at all times.

When contacting a Response Centre please quote your NAG user reference and NAG product code.

(c) Network

Network, NAG's newsletter, is produced quarterly and sent free of charge to sites with a supported product or service.

(d) NAG Bulletin Board

The NAG Bulletin Board is an information service providing items of interest to users and prospective users of NAG products and services. The information is regularly updated and reviewed and includes implementation availability, descriptions of products, downloadable software and technical reports. The bulletin board can be contacted using Gopher

Host=www.nag.co.uk, Port=70, Type=1, Path=1/

or WWW

<http://www.nag.co.uk:70/>

5. NAG Users Association

NAGUA, the NAG Users Association, is a self-financing, non-profitmaking body. It exists to promote communications between NAG and users of its products and services. It provides information to NAG on the requirements of users, who are in turn kept informed of developments in services. Membership is available to any institution or individual which holds a licence for any NAG product or service.

Members receive discounts on the registration fees at conferences and workshops. Members also receive 'NAGUA News', NAGUA's own newsletter.

For an information pack and membership application form, please contact NAGUA at the supplied address.

Appendix - supplied addresses

NAG Ltd
Wilkinson House
Jordan Hill Road
OXFORD OX2 8DR
United Kingdom

NAG Ltd Response Centre
email: infodesk@nag.co.uk

Tel: +44 1865 511245
Fax: +44 1865 310139

Tel: +44 1865 311744
Fax: +44 1865 311755

NAG Inc
1400 Opus Place, Suite 200
Downers Grove
IL 60515-5702
USA

NAG Inc Response Center
email: infodesk@nag.com

Tel: +1 708 971 2337
Fax: +1 708 971 2706

Tel: +1 708 971 2345
Fax: +1 708 971 2346

NAG GmbH
Schleissheimerstrasse 5
85748 Garching
Germany

Tel: +49 89 3207395
Fax: +49 89 3207396

NAG Users Association
PO Box 426
OXFORD OX2 8SD
United Kingdom
email: nagua@nag.co.uk

Tel: +44 1865 311102