

# Lawrence Berkeley National Laboratory

## LBL Publications

### Title

Fortran Newsletter Vol 7 No 4

### Permalink

<https://escholarship.org/uc/item/7s54p2qj>

### Author

Lawrence Berkeley National Laboratory

### Publication Date

1981-10-01

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

## **DISCLAIMER**

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

= F O R - W O R D = &gt;

NAMELIST Proposal Adopted at August X3J3 Meeting

Actions taken by the ANSI Fortran Standards Committee (X3J3) at its August 1981 meeting in Los Alamos NM included the adoption of a NAMELIST input-output proposal.

The feature, titled in the proposal "Name Directed Input and Output", is somewhat simpler than commonly available NAMELIST facilities. The feature is invoked by a special character in the Format position of the control part of the READ, WRITE, or PRINT statement; the exact form of this special character is not specified in this proposal.

For input, each item on the external record(s) is identified with a name followed by an equals sign. The name may be any name in the input list of the READ statement, and such items may appear on the external record(s) in any order. Corresponding to an array name in the input list, the input item may consist of the array name followed by one or more data items. The last name-directed item on the external record(s) is followed by a slash.

Name directed output produces records suitable for name-directed input, except for character constants.

Other X3J3 Actions

Refinement of Precision proposals. Proposals were adopted to pin down default precision attributes of real data objects and expressions, precision coercion in expression evaluation and assignment, and use of certain "environmental" intrinsic functions in precision declarations.

Precision matching rules for argument association were discussed: the "declared" precision values must agree. Inasmuch as a given "declared" (minimum required) precision may produce different "actual" (hardware implementation) precisions on different machines, matching of declared precision values is required for portability.

Refinement of Internal Procedure proposals. Implicit type specification rules are not "inherited" by an internal procedure from its host, according to a proposal that was adopted. In the absence of any declaration in the internal procedure, the "global" implicit typing rules of the language apply. Also discussed was an extension of the "INHERIT" statement (previously adopted) to an "INHERIT ALL" statement, which would give an internal procedure access to the data environment of its host.

Array extensions. The committee voted to eliminate the intrinsic functions (previously adopted) for computing the inverse or the determinant of a matrix. It was conceded that intrinsic functions might be better than subprograms coded individually by every untrained user; nevertheless, the numerical community favors deleting these intrinsic functions. A great deal of sophistication is required to develop proper linear algebra subroutines, as most users now realize; the use of well-written library programs (from IMSL, NAG, etc.) is becoming universally accepted.

--(continued on page 16)--

Loren P. Meissner  
CSAM - 50B  
Lawrence Berkeley Laboratory  
University of California  
Berkeley CA 94720

F I R S T C L A S S

RECEIVED

OCT 15 1981

ref

L.B.L. LIBRARY

For Reference

Not to be taken from this room

PUB-390  
[7:4 10/81]  
2500

Lawrence Berkeley Laboratory is an equal opportunity employer.

Future Meetings of X3J3

2 to 6 Nov 1981, Yorktown Hts NY  
 8 to 12 Feb 1982, San Diego CA  
 10 to 14 May 1982, Minneapolis MN  
 9 to 13 Aug 1982  
 8 to 12 Nov 1982

Meetings are open to the public, but facilities are limited. Further information is available from the X3J3 Vice Chairman,

Martin Greenfield  
 MS 844a  
 Honeywell Information Systems  
 300 Concord Rd  
 Billerica MA 01821  
 (617) 671-2912

International Fortran Meetings. The next meeting of the Fortran Experts Group under ISO/TC 97/SC 5 will be held in Vienna, 14 to 17 June 1982. A meeting of the X3J3 Task Group on Real Time Fortran will be held in connection with the Vienna meeting.

CORE AND MODULES (editorial comments):What does 'obsolete' really mean?

According to the projections of the latest X3J3 "milestone chart", final processing of the next Fortran Standard will occur in 1986. What does this mean to Fortran users?

• Fortran 77 is just now (3 years after its formal adoption as an ANSI Standard) becoming widely recognized and used. Thus we cannot expect implementations of the next Fortran standard much before 1990. Just as Fortran 77 is becoming "the Fortran of the 1980's", the next version will most likely be "the Fortran of the 1990's".

How will the Fortran of the 1990's be implemented?

The "minimal" Fortran implementation will include all of Core Fortran plus the "obsolete features module", and will accept all Fortran 77 programs -- at worst, a few options such as "source form" may have to be specified at the beginning of a source file. More complete implementations will include the Core and the obsolete features module, plus certain of the "extension modules" and "application area support modules" (the latter being the subject of related but separate Standards).

Possibly very late in the decade, after users have had several years of exposure to the "replacements" for the obsolete features, some daring (though non-standard) implementations may omit certain of those features.

In any case, it must be expected that all standard Fortran 77 programs will continue to be accepted during this millennium (i.e., through the year 2000) by standard-conforming processors.

After that, another revision of the Fortran standard may begin to be implemented. After the year 2000, those "obsolete features" that are in fact no longer widely used may disappear from most Fortran implementations.

A major example: Storage association

X3J3 currently has a major philosophical commitment to remove the concept of storage association from the next Fortran standard, if possible. It may prove impossible to do this while meeting the other criteria considered important by the Committee.

But even if storage association is removed from Core Fortran, there remains a commitment to support Fortran 77 programs that rely on the storage association facilities (such as COMMON, EQUIVALENCE, reconfiguration of arrays used as arguments, etc.) that have been relegated to the obsolete features module. The next Fortran Standard will have to indicate how such programs are to be accommodated, especially when they also incorporate data structures or other "new" features.

Thus storage association, along with all of the other "obsolete" features, will be with us throughout the remainder of this millennium.

Then what is the point of it all?

The advantage is that new programs can be written using only the "cleaner" facilities of Core Fortran (plus the extensions), and new programmers can learn to write only in the new, "more elegant" language (although they may be required to read programs that contain "obsolete" features as well).

Is this better than starting over entirely (with Pascal, for example, or Ada)? Only insofar as the new language remains compatible with the old one, for at least one revision cycle of the Standard.

The cost, of course, is that all compilers during the 1990's decade will have to support both the old and the new features. This cost will be acceptable only if redundancy is kept to a minimum.

If the core is kept "small", and the replacements for obsolete features are carefully designed, the next standard Fortran language should come into increasing use during the 1990's. By the beginning of the next millennium, it may be possible for many implementations to support the improved features and to ignore most of the obsolete features.

QUESTIONS WE KEEP HEARING1. How can I get a copy of the official ANSI Fortran 77 standard?

Send \$19.50 (\$16.50 + \$3.00 handling charge) and a request for "Programming Language FORTRAN X3.9-1978" to American National Standards Institute 1430 Broadway, New York NY 10018

Orders must be accompanied by remittance. (Special prices are available for large quantity orders, and discounts are available in certain cases: for information write to Bruno Bombardi at the address given above.)

(Note: An extensive paraphrase of the Standard appears as an Appendix to the textbook by Meissner and Organick, listed below.)

2. What Fortran 77 textbooks are available?

- a) A. Balfour and D.H. Marwick, "Programming in Standard Fortran 77", Heinemann/American Elsevier 1979
- b) W.S. Brainerd, C.H. Goldberg, and J.L. Gross, "Fortran 77 Programming", Harper and Row, 1978
- c) L.P. Meissner and E.I. Organick, "Fortran 77 Featuring Structured Programming", Addison Wesley, 1980
- d) R. Page and R. Didday, "Fortran 77 for Humans", West Publishing Co, 1980
- e) J.L. Wagener, "Fortran 77: Principles of Programming", Wiley, 1980

3. Who are the officers of the Fortran Standards Committee (X3J3)?

Chairman: Jeanne Adams, National Center for Atmospheric Research, Boulder CO.

Vice-Chairman: Martin Greenfield, Honeywell Information Systems, Billerica MA.

International Representative: Betty Holberton, National Bureau of Standards, Gaithersburg MD.

Secretary: Loren Meissner, Lawrence Laboratory, Berkeley CA.

Draft Standard for Real-Time Fortran

A Draft Standard on Real-Time Fortran is now available for public review and comment. Copies may be obtained free of charge from

Dr. Wilfried Kneis  
Mergenthaler Linotype -- Entwicklung  
Frankfurter Allee 55-75  
D-6238 Eschborn, Frankfurt, W Germany

Comments on the Draft are to be submitted to the above address, prior to 31 December 1981.

INPUT: Array Handling in Fortran

As the current X3J3 proposals on array handling for future versions of Fortran are distributed more widely, some feedback is being received. While the need for array handling in Fortran is conceded, there are some objections to the particular form of the features being considered in proposals before X3J3.

We have received two papers which are a bit too long to reprint in For-Word. Both of these have been distributed to the X3J3 mailing list.

John R Rice of Purdue University notes the need for a "parallel assignment operator". An array name with "dummy subscripts" would appear on the left, and an expression using the same dummy subscripts would appear on the right. The assignment would be subject to two constraints on the dummy subscripts:

- (1) a range (or upper and lower bound, for each dimension) constraint, and
- (2) a mask (logical expression involving the subscripts or the array elements) constraint.

The assignment would take effect only for those dummy subscript values satisfying both of these constraints.

According to Rice, "It is essential that a programmer be aware of the working size ... of any arrays in his program," and it is important to explicitly specify upper and lower bounds for the portion of an array currently in use. Thus an explicit "range constraint" is a necessary feature in array operations.

Ken Kennedy of Rice University addresses the "virtual array" concept introduced by the IDENTIFY statement. He notes especially the difficulty of implementing such arrays as actual arguments.

He perceives that the IDENTIFY statement "attempts to do too many things at once. In a single statement it

- (1) defines a subscript transformation,
- (2) defines a shorthand name for the base array with subscripts transformed, and
- (3) sets implicit upper and lower ... bounds for the new array name."

Accordingly, he suggests that the facility be subdivided into three separate features. "The function of making shorthand names should be relegated to a simple macro feature, the setting of iteration ranges should be handled by a special executable statement, and a new statement should be introduced to define subscript transformations."

August X3J3 Meeting (continued).

Use of "array sections" in arrays of structures was discussed at length. If A is an array of structures, each of which contains a component S which is an array, then is the identifier "A(\*)" the same as the identifier "A(\*)S(\*)", where "\*" is the whole array section selector symbol? The "canonical element order" for the former would hold fixed an element of A while taking all elements of S; while for the latter it would hold fixed an element of S while taking components across all elements of A. It was pointed out that, if "storage association" is removed from Fortran in the future, "canonical order" will have no meaning except when an unsubscripted array (or section) name appears in an input or output list. Thus the canonical storage sequence of Fortran 77 could be maintained for compatibility, and an option could be specified in the control part of an input or output statement to designate left-to-right or right-to-left subscript sequencing for input or output purposes.

Refinement of Enhanced Call. An intrinsic function "PRESENT" was defined, with an argument which is a dummy argument name, and a result which is "true" if there is a corresponding actual argument, and is false otherwise.

Character delimiter file (connection) property. A proposal was adopted to add to the control list for an OPEN or INQUIRE statement a specifier indicating whether character data items are to be enclosed in single or double quotes, or neither.

Content of Core Fortran. Discussion of Core Fortran content proposals centered on the extent of array processing facilities needed in the Core language. What facilities are needed for operating on a portion of a declared array, such as the upper left-hand corner of a large working-space matrix?

This question interacts with the implementation of character facilities. With extensive full-array facilities, it might be possible to handle character strings simply as one-dimensional arrays of single characters. (However, the operations needed for strings differ in many respects from those needed for numeric arrays.)

More MANUAL MADNESS

Sergio Bernstein of Albuquerque NM comments on For-Word, April 1981, page 8: "Mr Bill Weisman (wisely!) takes exception to 'common GHIA'. We agree; such things are enough to cause 'DIMENSION PRECOX (N)'."

French text of Fortran Standard

French standard NF Z 65-110, a French language version of the Fortran 77 Standard, has been adopted as the French version of ISO International Standard 1539-1980 (Programming Language FORTRAN). Copies may be obtained through the Association française de normalisation, Tour Europe, Cedex 7, 92080 Paris La Defense, France.

PL/1 Subset Standard

American National Standard X3.74-1981, "General Purpose Subset Programming Language PL/1", is available from ANSI (see page 15 for the address). The price (including handling charge) is \$10.00.

FORTRAN PIONEER ANECDOTES WANTED

Pioneer Day activities at the National Computer Conference, to be held in Houston, 7 - 10 Jun 1982, will commemorate the 25th anniversary of the first Fortran compiler. A History Committee headed by Dr. J.A.N. Lee is collecting anecdotes relating to early Fortran usage. Of special interest is the history of Fortran in non-IBM environments, from the late 1950's through approximately 1967.

A Fortran archive is also being developed at the Charles Babbage Institute in Minnesota.

Please forward stories, anecdotes, and references to potentially interesting papers and persons (please use some imagination) to Dr. Lee at the following address:

M43 / D22, IBM Corporation;  
555 Bailey Ave;  
San Jose CA 95150.

CONCERNING FOR-WORD

This Newsletter is prepared for the U.S. Dept. of Energy, Office of Energy Research, Applied Mathematical Sciences Research Program, under Contract W-7405-ENG-48.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Dept. of Energy, to the exclusion of others that may be suitable.

The editor's name and address appear in the Mailing area on the first page of each issue. Requests for additions or corrections to the mailing list should be directed to the editor.

Correspondence on all Fortran-related topics is welcomed. Especially solicited are reviews of recent Fortran textbooks, software products, literature, etc.