

104th MEETING
X3J3 PRE-MEETING DISTRIBUTION
BELLEVUE, WASHINGTON
MAY 11 - 15, 1987

			<u>INDEX</u>	<u>PAGE</u>
1	104(*)	JCA-1	Between Meetings Letter	1
2	104(*)	JCA-2	Part I of the X3 Submittal	3
3	104(*)	JCA-3	Part II, X3Submittal, X3J3 & WG5 Responses	21
4	104(*)	JCA-4	News Releases from X3	108
5	104(*)	JCA-5	DS-4 Projects Manual	111
6	104(*)	JCA-6	SD-6 Membership and Officers	121
7	104(*)	JCA-7	SD-7 Meeting Schedule and Calendar	135
8	104(*)	JCA-8	US Contributions (ATTn.IR)	141
9	104(*)	JCA-9	Coordinating Liaison Responsibility ...	143
10	104(*)	JCA-10	Character Sets New Item	145
11	104(*)	JCA-11	News Item--Data Protection System	147
12	104(*)	MBM-1	Minor Edits	149
13	104(*)	MBM-2	Wearing Letter	151
14	104(*)	MBM-3	Moss Trip Report	153
15	104(*)	MBM-4	Reid Trip Report	159
16	104(*)	MBM-5	Reid Trip Report	163

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

Scientific Computing Division/Advanced Methods Section

P. O. Box 3000 • Boulder, Colorado • 80307

Telephone: (303) 497-1275 • FTS: 320-1275 • Telex: 45 694

104(*)JCA-1

MEMO TO: X3J3

FROM: Jeanne Adams, Chair

DATE: April 1987

SUBJECT: Between Meetings Letter

The meeting in March was a very productive one. We processed the substantive issues remaining from the ballot, and continued with the editorial changes. The committee was divided into two, sometimes three subgroups during the meeting. At some point during the 103rd meeting (either in Torrance or in Albuquerque), each item was processed by some subgroup and coded. We completed all the ** items relating to the main body of the text. During some of the processing, the items were examined by two subgroups before processing by full committee. In continuing with the editorial material, we discovered a number of items that we changed a few meetings ago, and then changed back again. It appears that, like all authors, we process these items again and again. I feel that this kind of editorial processing is reaching the point of very diminishing returns for the effort that we apply.

There were approximately 20 persons in attendance. For every vote, there was a quorum voting from the 17 members who were there. Everyone worked very hard during the meeting because we discovered that typesetting of the document will be possible before the May meeting, and then not be available until late July, too late to have a document before the Liverpool meeting. Lloyd and Walt are working toward having a modified S8 ready before the May meeting so that we can have a role call vote and bring this phase of the work to a close. The role call ballot on S8 will be the wording contained in the letter ballot distributed last December. If you do not plan to attend the meeting in May, please send me, before the meeting, an absentee ballot for the role call vote.

There are two essential proposals to process before the ballot. At the close of the March continuation meeting, we passed a proposal to give Lloyd the freedom to correct any editorial changes to the document and prepare a list for X3J3 at the beginning of the May meeting. One of the proposals is the list of Lloyd's changes, which will already be in S8. The other is the INCLUDE proposal that Kurt prepared based on the straw vote at Torrance. The supplement to the 103rd meeting minutes will contain all of the material that was processed at the March continuation meeting. Please look at the items processed in March very carefully if you did not attend.

Document 207 includes the material that will accompany S8 when it is forwarded for further

processing. This document, Part I and Part II, was distributed at the March meeting, and it will be JCA-2 and JCA-3 in the pre-meeting distribution with some suggested changes already made. Part I is background material, Part II are the responses to the commentary from ballots of X3J3 and WG5. I will not have had time to edit the material that came to me at the meeting before the deadline for the pre-meeting distribution. I will send Dick the comments I have received but these will not necessarily be edited. Part II will contain a proposed response to each ballot. The agenda in May will call for processing these responses. A 2/3 vote is required to approve Part II (no vote is needed for Part I, but I would like your comments). I plan to ballot on each response separately. Subgroups will meet to make the first evaluations; I will have ballot assignments on Monday. These comments and responses must be approved before we can forward S8 to X3 and SPARC. This is an important part of the approval process of the draft standard.

If you have any questions, please feel free to call me or send me electronic mail.

X3J3

X3J3 BALLOT RESULTS

**SUBMISSION TO X3
DOCUMENT NUMBER 207**

Part I

BSR PUBLIC REVIEW TRANSMITTAL FORM

4

FOR
SECRETARY
ACTION

X3TC X3J3

X3 Project # 67-R

Technical Committee Doc. # X3J3/58.104

Formal title as well as "nicknames" which might be used:

~~FORTRAN~~
~~FORTRAN~~ 8X

Results of TC LB to Forward for Further Processing:
(Attach details)

29/7/0
Yes/No/Non Response

Document Content Checklist -- Have you included:

- Foreword? ✓
- Name, address and phone number of Technical Editor? ✓
- Technical Committee List & names of Individual Experts who have contributed? (This is the list which will appear in the approved standard) ✓
- Expository Remarks? ✓

Preferred quote for Public Review Press Release -- geared toward enticing User interest in document:

If you have additional press contacts you would like notified of the availability of the document, please include their names and addresses below (use separate sheet if necessary):

MEMO TO: X3

DATE: May 15, 1987

FROM: Jeanne Adams, X3J3 Chair

SUBJECT: Submission

SECOND X3J3 LETTER BALLOT
December 1, 1986-January 5, 1987
Draft Revision of X3.9-1978 (Fortran 8x)

X3J3 has processed the ballot commentary of the X3J3 letter ballot and the TC97/SC22/WG5 ballot. The final ballots are enclosed. Responses voted by full committee are included with the negative ballots, as well as with some of the affirmative ballots that contain remarks of a general nature. The editorial comments included with ballots have been acted on by X3J3 and either accepted, or considered and rejected. These comments have not been included.

X3J3 requests a compliance review by SPARC. After this review, X3J3 requests that an X3 public review be initiated. At the same time, since the WG5 ballot was affirmative, X3J3 requests that the document be submitted to SC22 members for a review concurrent with the X3 public review.

The technical editor is:

Lloyd Campbell
618 Southgate Road
Aberdeen, MD 21001
Telephone (301) 272-3771

6

SUMMARY OF X3J3 VOTE

Vote	Number
Affirmative (15 with comments)	29
Negative with Reasons	7
No Ballot Received	0
Total	36 members

SUMMARY OF WG5 VOTE

Vote	Number
Affirmative	25
Negative with Reasons	2

In addition to the proposed draft revision for Fortran, information about the history of the development phases and the responses to negative ballots and affirmative ballots with comments have been included. This report contains the following sections.

1. FORTRAN STATEMENT SUMMARY LIST
2. PROJECT PROPOSAL FOR FORTRAN
3. X3J3 DEVELOPMENT PHASES AND HISTORY
4. X3J3 BALLOTS
5. TC97/SC22/WG5 BALLOTS
6. THE DRAFT FORTRAN REVISION (S8)of X3.9-1978

A final tally taken by role call vote at the meeting in Tukwila, Washington on May ?, 1987 was *****affirmative and *****negative.

Please submit the draft revision for public review nationally and internationally as soon as SPARC has completed the compliance review.

FORTRAN STATEMENT SUMMARY

8

ABBREVIATED LIST
FORTRAN STATEMENTS and LINES

Assignment

variable=expr

where variable may be a scalar array or a subobject

IF (scalar-logical-expr) action-stmt

WHERE (array-logical-expr) array-variable=expr

Data Type and Specification

type [[,attribute]... ::] object-list

where type is INTEGER, REAL[(precision)], COMPLEX[(precision)], DOUBLE PRECISION,
LOGICAL, CHARACTER [length selector],
or TYPE (type-name [(type-param-list)])

attribute is PARAMETER, DATA, PUBLIC, PRIVATE, ALIAS, ALLOCATABLE, SAVE,
ARRAY(array-spec), INTENT(intent-spec), OPTIONAL, RANGE[/range-list-name/]

[access] TYPE derived-type-name [(type-parameter-list)]

where access is PUBLIC or PRIVATE

END TYPE [type-name]

IMPLICIT NONE

IMPLICIT implicit-list

EXPONENT LETTER [(precision)] letter

Other Data Specification Statements

DIMENSION array-bounds-list

COMMON [/[block name]/] common-list [[,/][block name]/common-list...

EQUIVALENCE equivalence-list

DATA data-stmt-object-list/value-list/[[,]data-stmt-object-list/value-list/]...

DATA (value-definition-list)

PARAMETER (named-constant-definition-list)

RANGE [/range-list-name/] array-name-list

NAMelist [/namelist-group-name/] namelist-object-list

[[,]/namelist-group-name/namelist-object-list]...

Dynamic Storage Allocation, Ranging, and Aliasing

ALLOCATE (array-allocation-list [,STAT=stat-variable])

DEALLOCATE (array-name-list [,STAT=stat-variable])

SET RANGE ([range]) array-name-list

SET RANGE ([range]) /range-list-name/

IDENTIFY (alias-name=subobject)

IDENTIFY (alias-element=parent-element, alias-range)

Control Constructs

[do-name:] DO [label] [,] do-variable=expr,expr[,expr]
[do-name:] DO [label][[,] (expr) TIMES]
CYCLE [do-name]
EXIT [do-name]
CONTINUE
END DO [do-name]

[if-name:] IF (scalar-logical-expr) THEN
ELSE [IF (scalar-logical-expr) THEN]
END IF [if-name]

[select-name:] SELECT CASE (scalar-expr)
CASE (value-range-list)
CASE DEFAULT
END SELECT [select-name]

GO TO label
GO TO (label-list)[,] scalar-integer-expr
STOP [access code]

WHERE (array-logical-expr)
ELSEWHERE
END WHERE

Statement Function

function-name ((dummy-argument-list)) = expr

Program Units and Subprograms

[PROGRAM program-name]
MODULE module-name
USE module-name [,rename-list]
USE module-name,ONLY:[only-list]
PRIVATE [[::] use-name-list]
PUBLIC [[::] use-name-list]
BLOCK DATA [block-data-name]
EXTERNAL external-name-list
INTRINSIC intrinsic-procedure-name-list

CALL subroutine-name [([actual-argument-list])]
[RECURSIVE] SUBROUTINE subroutine-name [(dummy-argument-list)] [ASSIGNMENT]
[prefix] FUNCTION function-name [(dummy-argument-list)] [suffix]
where prefix is type [RECURSIVE] or [RECURSIVE] type
suffix is [RESULT (result-name)] [OPERATOR(defined-operator)]
or [OPERATOR(defined-operator)] [RESULT(result-name)]

Statements Related to Program Units and Subprograms

INTENT (intent-spec) [::] dummy-argument-list
OPTIONAL [::] dummy-argument-list
SAVE [[::] object-list]
CONTAINS
RETURN
ENTRY entry-name [([dummy-argument-list])]
INTERFACE
END INTERFACE
END [unit [unit-name]]
where unit may be PROGRAM, MODULE, SUBROUTINE, FUNCTION, BLOCK DATA

Input/Output Statements

READ (control-list) [input-list]
WRITE (control-list) [output-list]
READ format [,input-list]
PRINT format [,output-list]

REWIND external-unit or REWIND (position-list)
ENDFILE external-unit or ENDFILE (position-list)
BACKSPACE external-unit or BACKSPACE (position-list)
OPEN (connect-list)
CLOSE (close-list)
INQUIRE (inquire-list) [output-list]
FORMAT ([format-spec])

Note that obsolescent features (PAUSE, ASSIGN, ASSIGNED GO TO, arithmetic IF, alternate RETURN) have not been included in this list.

PROJECT PROPOSAL

FORTRAN

PROPOSAL FOR CONTINUATION OF THE
X3 STANDARDS PROJECT ON FORTRAN

1.0 Identification

- 1.1 Title: Support of and revision to X3.9-1978 Programming Language FORTRAN
- 1.2 Proposer: Jeanne Adams, Chair, X3J3
National Center for Atmospheric Research
P.O. Box 3000
Boulder, Colorado 80307
- 1.3 Date: April, 1978

2.0 Description

2.1 Purpose

The purpose of the standard is to promote portability of FORTRAN programs for use on a variety of data processing systems.

2.2 Nature of the Standard

The standard is a document that specifies the form and establishes the interpretation of programs expressed in the FORTRAN language. The document will be a revision of American National Standard Programming Language FORTRAN, X3.9-1978.

2.3 Scope

1. Provide a mechanism for the solicitation and review of all activities regarding American National Standard Programming Language FORTRAN.
2. Carry out procedures to maintain the continuous responsiveness of the standard to industry needs.
3. Act as a liaison group with other organizations interested in the standardization of FORTRAN.
4. Augment the American National Standards FORTRAN (X3.9) with a comprehensive data base facility using the CODASYL FORTRAN JOD wherever consistent with the general interface features being proposed and wherever appropriate for inclusion in the language.

2.4 Program of work

1. Provide support, interpretation, and maintenance of X3.9-1978, Programming Language FORTRAN.
2. Continue pursuing the work required for the international standardization of FORTRAN.
3. Establish the criteria to be applied in evaluation proposals for the revision of X3.9.
4. Prepare a draft proposed revision of X3.9 for public review and comment.
5. Maintain a close working relationship with other national and international groups interested in FORTRAN standardization such as the CODASYL FORTRAN group.

6. Maintain a close working relationship with groups active in the creation of collateral standards affecting FORTRAN.
7. Keep the public informed of the direction and activities of the committee in order to encourage feedback.

3.0 Expected Benefits

3.1 Intrinsic

The major intrinsic benefit of a FORTRAN standard is to minimize or eliminate potential incompatibilities in different implementations of FORTRAN language processors.

3.2 Interchange

The standard provides for a degree of portability of both programs and programmers between different computers and computer operating systems.

3.3 Educational

The standard provides a language definition that can be used by both authors of FORTRAN documents and teachers of FORTRAN at all levels. Additionally, textbooks, reference manuals, and other documents describing FORTRAN will become more consistent in their terminology and description of the language, thus reducing retraining costs.

3.4 Economics

Interchangeability of programs and programmers has proven to be of major economic advantage within the industry. The continued maintenance and revision of the standard protects the investment in already existing FORTRAN programs and FORTRAN language processors.

4.0 Feasibility of Development

4.1 State of the Art

During development of a new revision to X3.9, it is likely that augmentations will mainly be drawn from functionality that exists in advanced implementations of existing processors. Therefore, development of a new revision of X3.9 is well within the state of the art. During the period of development of X3.9-1978 and during the public comment period on BSR X3.9, a very large body of public comment was received. Many proposed extensions were found to have merit but were not included either because of limited resources available to the committee or because their inclusion would have unduly delayed the issuance of the revision. It is, therefore, highly timely that those proposed extensions be examined for possible inclusion in a subsequent revision.

4.2 Resources

X3J3 has been able to attract sufficient working membership to complete the first revision of the standard. New members have joined the committee since the adoption of the revised standard. There is every indication that the technical committee will have sufficient resources to complete the Program of Work

4.3 Cost

The cost of developing X3.9 has, for the most part, been borne by the sponsors of the membership of X3J3. The committee members and their employers are aware of the expense. The committee intends to meet five times per year for four days. Expected average attendance is 30 persons.

5.0 Feasibility of Implementation

5.1 Supplier Conformance Consideration

In developing the present FORTRAN standard a great deal of effort was expended in describing the language so that it could be interpreted unambiguously by FORTRAN implementors. A large body of public comment confirmed the belief that the existing form of the standard was readily understandable by suppliers. No incompatibilities with emerging technology are foreseen.

5.2 User Operational Considerations

Once a language standard is in force and there are conforming programs, incompatible change, even a small change, impacts the user community. Many of the members of X3J3 are users and additionally, X3J3 has been in continuous contact with various user groups. Any incompatible change to the existing standard would be accepted only after very careful consideration.

5.3 Cost Considerations

One of FORTRAN's most important characteristics is that efficient processors can be implemented at a reasonable cost. One of the most important goals during the next revision will be to retain this characteristic.

6.0 Maintenance

6.1 Extent and Frequency of Anticipated Changes

X3J3 intends to provide interpretation and clarifications of X3.9-1978 as the need arises. The committee also intends to comply with the requirement that X3.9-1978 be reviewed within a five year period.

6.2 Resources

The committee accepts its responsibility to maintain X3.9-1978 and to continue this activity along with its revision efforts.

6.3 Cost

The cost of maintaining the standard is estimated at 10% of the five-year development cost.

7.0 Related Standards Activities

Related standards activities are conducted by other X3J3 committees, X3L5, the CODASYL FORTRAN DML Committee, the Purdue Workshop, ECMA TC8 and TC22, IFIP Working Group 2.5-Numerical Software, and ISO TC97/SC5.

8.0 Recommended Time Frame for Revision of X3.9

- 1978 Jan Establish procedures within X3J3 for processing proposals for revision of X3.9 and begin research on the identification of problem areas that will generate revisions to X3.9.
- 1979 Jan Begin processing proposals for revision of X3.9.
- 1982 Jan Submit draft proposed revision of X3.9 to X3 for public review and comment.
- 1983 Jan Publish X3.9-1983.

X3J3 HISTORY

DEVELOPMENT PHASES X3J3 MEETINGS 1977-1986

1977

October ---> OBJECTIVES

1978

March ---> TUTORIALS

April X3.9-1978

1979

January ---> PROPOSALS

1980

August ---> S6 NEW FEATURES

1981

1982

January ---> S7 F77 + S6

1983

1984

May ---> S8 DRAFT

1985

1986

April ---> FIRST BALLOT

REVISION
PROCESSING

1986

December ---> SECOND BALLOT



ABSTRACT
HISTORY OF X3J3 MEETINGS
1977-1987
Jeanne Adams, Chair, X3J3

1977

1977 marked the conclusion of the public review of Fortran 77 and the conclusion of the responses by the Fortran Standards Committee. A subcommittee was assigned to study the scope and the plans for the future of Fortran. A vote was taken to remain as a 'Committee of the Whole'. The committee was resolved to work with international organizations and the Fortran Experts Group as the draft standard was developed.

1978

X3.9-1978 was approved by BSR. A project proposal was prepared for the next Fortran Standard. SC5 approved Fortran 77 for processing as an International Standard. User needs were surveyed. A Core-Plus-Modules approach to Fortran was adopted. A Tutorial Program was initiated to conclude in January 1979. An international meeting was scheduled. Issues and clarifications of Fortran 77 were processed.

1979

A Core Foundation Document was passed. Task groups were formed to prepare a CODASYL Data Base Standard for Fortran and a standard for Industrial Real Time Fortran. Tutorials continue and the first proposals are processed. The development cycle for Fortran 8x begins. A tutorial on Open Systems was presented. Discussions conclude that too many new features are proposed.

1980

Procedures for Task Groups in X3J3 were formulated. Further attempts were made to place boundaries on new features. A list of allowable technical material was approved. A cutoff date for new material was established. Plans were initiated for the Proposal Document (S6) which was completed and refined. ISO 1529 was accepted as an International Standard.

1981

S6 was formally adopted by X3J3. Plans for a "Fortran Family of Standards" were presented to X3J3 and SPARC. Financial problems threaten the committee and cost cutting methods are discussed. Meetings are limited to four in a year and the milestone charts advance the completion date to 1986. Refinements of major proposals in S6 are the basis of most of the technical proposals processed. A high priority item for X3J3 is the publication of a Fortran Information Bulletin on Clarifications of Fortran 77. A tutorial was held on keeping Fortran a 'naive and universal' language.

1982

The preparation of the draft standard was begun by combining S6, the proposal document, and the Fortran 77 standard. This document (S7) is to be processed by UNICOMP, in Los Alamos, New Mexico. Critical issues were identified for the preparation of S7. A Fortran Information Bulletin was planned summarizing the work toward the Fortran 8x standard. A first draft of S7 was completed.

1983

S7 was reviewed from the point of view of the "ANSI Style Manual". S7 was formally accepted and S6 declared a historical document. A request was sent to X3 to withdraw the data base project on CODASYL. ISO was notified that a draft standard was in the final writing stage. After work progressed on S7 for several meetings, the committee concluded that it needed a basic reorganization from the one based on Fortran 77. After study for several meetings, a reorganization of S7 was begun. This decision was based on the fact that S7 did not present a clear picture of data types (including programmer defined ones, data objects and the relationship of data to an executing program. In spite of the fact that X3J3 knew that this would significantly increase the time needed to complete Fortran 8x, the committee felt that S7 as it stood was unacceptable. Therefore the development of S8, the reorganization of S7, was undertaken.

1984

A Four Meeting Plan was initiated that controls the critical issues needing processing, and establishes longer term committee goals. Project 318 was withdrawn by X3. The Fortran Information Bulletin summarizing Fortran 8x was published. Responses were prepared for the negative votes of IBM and DEC on the Fortran Information Bulletin. Attempts were made to decrease the size of the language; at each meeting candidates for deletion from the new features list were considered. Those that would be acceptable for deletion would not impact the size by much. A completed version of S8 was distributed. Because there were several authors, it became apparent that much editorial work was needed. A Fortran Forums Program was initiated and a user questionnaire was circulated on the new features in Fortran 8x.

1985

S8 was accepted as a base document and S7 became a historical document. The meeting schedule was expanded back to five meetings a year. November 1985 was the goal for forwarding S8 to SPARC for compliance review and balloting on S8 within X3J3. The vote to forward failed, but a straw vote indicated that a vote on S8 at the January 1986 meeting would pass.

1986

In January, X3J3 voted to conduct a letter ballot during March. The document was "frozen" except for editorial fixes and issues to fill holes and repair inconsistencies.

20

The ballot failed (16-20-1-0). Some of the ballots suggested that the language was too large, others did not agree with the style and editorial presentation of the document. Requested deletions were all quite different and it was apparent that further work must be done to resolve issues on the language content. The next three meetings were spent discussing deletions to the language and a reduced language was eventually adopted.

The reduced language was presented to the WG5 meeting in Halifax, where other suggestions to improve the language content were presented. A December ballot was conducted by X3J3 and by WG5. Both ballots passed. There were many editorial comments presented with affirmative votes, as well as with negative votes.

1987

An extension to the February meeting was called for March in Albuquerque to complete processing ballot comments. After the meeting in Albuquerque, all ballot comments had been processed by the full committee. A corrected document was prepared as the draft Fortran standard. The following reviews are requested by X3J3:

1. A compliance review by SPARC
2. Public Review approval by X3
3. Submission to SC22 for concurrent public review with X3

The documents enclosed were prepared for the above review requests.

104 (*) JCA-3²¹

DOCUMENT 207 PART II
X3J3 Letter Ballots

TABLE OF BALLOTS
 December Ballot on S8.103
 Draft Revision of Fortran (X3.9-1978)

NAME	VOTE	PAGE
1. Adamczyk, J. Stephen	Aff	1.0
2. Adams, Jeanne C.	Aff	2.0
3. Barber, Graham	Aff	3.0
4. Bowe, Valerie	*	4.0
5. Brainerd, Walter	Aff	5.0
6. Burch, Carl D.	Aff	6.0
7. Campbell, Lloyd	Affc	7.0
8. Crowley, Ted	Aff	8.0
9. Allison for Chela	Affc	9.0
10. Ellis, T. Miles	Affc	10.0
11. Freeman, Murray	Affc	11.0
12. Harris, Kevin	Neg	12.0
13. Hendrickson, Richard	Neg	13.0
14. Hirschert, Kurt W.	Affc	14.0
15. Hoover, Tracy Ann	Affc	15.0
16. Johnson, Andrew	Affc	16.0
17. Lakwara, Anil	Neg	17.0
18. Marshall, Neldon	Aff	18.0
19. Martin, Bruce	Affc	19.0
20. Martin, Jeanne	Affc	20.0
21. Marusak, Alex	Affc	21.0
22. Matheny, James	Aff	22.0
23. Metcalf, Michael	Aff	23.0
24. Millard, G. G.	Aff	24.0
25. Moss, Leonard	Neg	25.0
26. Phillips, Ivor	Neg	26.0
27. Ragan, Richard	Affc	27.0
28. Reid, J. K.	Affc	28.0
29. Rolison, Lawrence	Neg	29.0
30. Schenk, Werner	Aff	30.0
31. Schonfelder, J. L.	Affc	31.0
32. Smith, Brian	Affc	32.0
33. Swift, Richard C.	Affc	33.0
34. Thompson, Brian	Aff	34.0
35. Wagener, Jerrold	Aff	35.0
36. Weaver, Richard	Neg	36.0
37. Wilson, Alan	Aff	37.0

Note that Aff is affirmative, Affc is affirmative with comments, and Neg is negative with reasons. *Voting with Rolison, one vote only for UNISYS.

Harris is marginally in favor of submitting the draft to X3 for further processing. We have some serious misgivings about the draft, but are willing to see if a public review indicates that the public has similar misgivings. There are some general comments about a few features and we have attached some editorial comments.

SET RANGE is insufficiently defined to be easily understood (if at all). For example, if a module defines a variable A with attribute RANGE but not SAVE, and the variable is USED in procedure B which performs a SET RANGE, and then calls procedure C which also USES variable A, it seems pretty clear that the effective range from procedure B is in effect. And if procedure C also performs a SET RANGE and then exits it seems pretty clear that procedure F is affected by the effective range declared in C. But, if B is called from some other procedure which does not USE A, and B performs a SET RANGE and then exits, and then procedure C is called, does it receive the effective range defined in B? It would seem so, since according to the sentence on page C-11, lines 22-24, the definition only goes away when no one is using the module. This requires an elaborate implementation scheme, which is both slow and large.

The major problem in figuring out SET RANGE is reconciling the statements in the sections on scoping with the description of modules. The section notes for Section 10 give a backhand indication of how the definition status of variables in modules become undefined, but it has nothing to do with local or global scope. Yet, the definition of when effective range reverts back to declared range is in terms of when a variable that is local to a scope becomes undefined (See page 6-3, lines 26-28).

I would be happy to submit a proposal to clarify the intent of these features if only I could figure out what the intent was.

Also, the implications of allowing intrinsic functions with constant expressions as arguments to be constant expressions are far reaching. In general, this requires the compiler to now include the entire runtime library so that it can perform any of the intrinsic functions. As a specific example, cross-compilers are going to have a terrible time doing trigonometric functions to an accuracy which is greater than the host machine's. (Imagine the unusual case of a cross-compiler from a micro or PC to a mainframe. It must emulate floating-point sizes two to four times its floating point size). In terms of compiler size (it is becoming increasingly difficult to imagine how this current draft standard will fit on a micro) and compiler speed (software emulation of a trigonometric function at software emulated precisions), this seems to be an expensive addition for limited additional functionality.

RESPONSE TO ROBERT ALLISON BALLOT

There were 12 editorial comments in your ballot, all of which have been addressed by full committee as changes to the draft, or considered and rejected. You requested comment 1) on the definition of SETRANGE and 2) on certain implications of intrinsic functions with constant expressions.

1) SETRANGE

You are correct. SET RANGE as currently specified in S8 requires an elaborate and slow implementation scheme to reset the effective ranges to the declared ranges. The technical subgroup responsible for SET RANGE is proposing a change in specification that will eliminate this problem. The proposed change will cause the effective range to revert to undefined where it currently reverts to the declared range. This will make SET RANGE work much as COMMON works in Fortran 77 and permit efficient implementations without undue burden on the user or the implementor.

2) Intrinsic Functions with Constant Expressions

The X3J3 committee considered the concerns expressed in the above ballot comments but felt no changes were required for the following reasons.

Fortran 77 currently permits constant expressions to include all of the intrinsic arithmetic, character, and logical operators with constants or constant expressions as operands, except for the exponential operator ** which is limited to integer powers. Therefore, cross compilers even for Fortran 77 currently must address the issue that the operands may have a very different representation on the host than on the destination machine.

On the other hand, Fortran 77 does not require that such expressions be evaluated at compile-time but can be postponed to runtime. Whether it is an advantage or disadvantage to do so depends upon the machine and the properties of the object files and the linker/loader.

The issue, therefore, is not one of precedent but of the number of operations (including those embodied in intrinsic functions) that must be supported in what appears to many to be compile-time expressions. Several models of how this can be implemented are readily apparent. Which one is sensible will depend upon the machine and also on the tradeoffs between storage and execution and on whether the evaluation of such expressions is borne at compile-time, linker/loader-time, or execution-time. (One model that addresses the particular example given in the ballot comment RA 9.1 is one where the expressions are evaluated just prior to the execution of the main program and the values inserted into the appropriate places in the load image.)

With the determination of execution-time determined extents for arrays, the need for

arbitrary constant expressions in specification contexts becomes apparent. However, the use of the entire intrinsic function set appears at first to be excessive. When attempting to formulate a rationale for including some but not all of these functions, it became clear to the committee that the choices were somewhat arbitrary and were dependent on the applications considered. Only a few such functions seemed to be clearly not useful. But providing a small list of excluded functions seemed not to widen the set of feasible implementations and at the same time was user unfriendly.

Case Value Range

Your proposal to delete the case-value range for ":" was accepted.

26

To: Jeanne Adams. Chair - X3J3
From: Miles Ellis
Date: 16 December 1986
Subject: Response to X3J3 Letter Ballot - December 1986

My vote is in favour of submitting X3J3/S8/103 to X3 for further processing as an American National Standard. However, I have two proposals that I would ideally wish to see incorporated in the document before such submission.

RESPONSE TO MILES ELLIS BALLOT

There were two proposals with this ballot.

The first, on conformance, was modified in committee and accepted.
The second was an editorial proposal that was handled in another way.

10.1

30 December 1986

103(*)MFF-1

Page 1 of 32 pages

To: X3J3
From: M.Freeman
Subject: Comments on X3J3/S8.103 attached to letter ballot

General. The comments below are almost entirely in the form of suggested edits to the document. I believe them to be editorial in nature; in those cases where they appear to be substantive, I consider the document to have been incorrectly transcribed or edited. Most of the edits are to improve readability and accuracy.

RESPONSE TO MURRAY FREEMAN BALLOT

Your editorial comments with this ballot have been addressed by full committee and accepted, or considered and rejected.

11.1

Harris 28

Response to Letter Ballot on submission of FORTRAN 8X
to X3 for processing as an American National Standard
Digital Equipment Corporation

Digital Equipment Corporation votes NO on this ballot.

1 TECHNICAL COMMENTS

While Digital believes that an updated Fortran standard is overdue, and that X3J3/S8.103 contains many worthwhile new capabilities, we believe there are still several major problems that need to be addressed before forwarding this document as a draft proposed American National Standard. We appreciate the actions the committee has taken in response to the previous letter ballot. We would appreciate similar attention to the problems described below.

We believe that publishing this document may jeopardize the past successes of Fortran in its attempt to make several major changes to the language. We have several specific areas of concern. In each of these cases, it is apparent that the committee has made an attempt to address an acknowledged weakness in the Fortran 77 standard. However, we believe that the solutions chosen have significant drawbacks. Our major concerns are in the following areas:

1. Users have expressed the need for improved data type support in several areas, particularly the need for new intrinsic types for supporting bit, pointer, and varying length string operations. This proposal attempts to provide these through a set of language extensibility mechanisms rather than new intrinsic types. The implementations resulting from this attempt will be too inefficient.
2. The new source manipulation capabilities are more powerful than necessary, too complex, and untested.
3. The new features intended to enhance portability of numerical software are untested by practice and are not ready to replace the existing default REAL and DOUBLE PRECISION declarations. They are not clearly effective at attaining the desired portability because they do not account for major sources of variability such as round-off error and intrinsic function accuracy.
4. The features chosen for the obsolescent and deprecated features lists are not justifiable based on potential benefits or costs.

These problems and proposed changes are described in more detail below.

RESPONSE TO KEVIN HARRIS BALLOT

X3J3 has considered your ballot and appreciates your continued participation in our effort to complete the draft standard for Fortran. Your points on Language Extensibility, Source Manipulation, Portability of Numerical Software and Deprecated Features have been considered in full committee. Your two editorial comments were handled in another way in other ballot comments. The following notes summarize the committee position on your four technical issues.

1.1 Language Extensibility

The ballot objects to the "Language Extensibility" features proposed for Fortran 8X on the grounds that they are inadequate to provide facilities for bit data, pointers, and varying length characters. The semantic extensibility features were added for entirely different reasons, and criticism is not relevant in an area where the feature is not able to do something it was never intended it should.

That bit data and pointers should be included as intrinsic data types, if they are included at all, has not and never has been seriously in doubt. That a dynamically variable length character would be best done as a new form of aggregate is also fairly widely accepted. Although the semantic extension capabilities with only modest additions could adequately express this functionality, the deficiency with the language for varying character lies more with overly restrictive storage management facilities (one is not permitted to allocate a string) and deficiencies in the area of I/O (the lack of input where the record structure is data determined rather than format). The reason that these three desirable intrinsic facilities are not in the current draft proposals is not because they can easily be provided as language extensions but because adequate proposals have not been put before the committee, or at least no proposal able to command and maintain majority support.

The "Language Extensibility" features, which include the facilities for:

1. Defining and declaring structured data objects,
2. Defining procedures to manipulate such objects,
3. Allowing such procedures to be invoked using a conventional functional notation,
4. Allowing a restricted class of such procedures to be invoked using normal infix operator notations,
5. Allowing the "generic" or "overloading" properties of intrinsic functions and operators to be extended to user defined procedures using either invocation syntax,

were introduced to satisfy strongly expressed user needs for facilities to handle heterogeneous aggregates of data and to allow programs working with such objects as "physical three vectors", relativistic four vectors, rational numbers, tensors, etc. to be written in a notation analogous to that conventionally used in these various fields. The

facilities introduced although probably capable of being improved (all human invention is less than perfect) do allow such functionality in a reasonably economical manner.

Far from derived-types and operator definition being in conflict with intrinsic bit and pointers, both of these intrinsic types, especially the latter, when combined with the existing semantic extensibility features would greatly enhance the language and its ability to express extensions.

1.2 Source Manipulation

Features embodied in modules and USE statements are not new. They exist in Ada as packages and WITH and USE statements. Many believe these features are Ada's greatest contribution to language development because they facilitate reusable software. The Fortran 8X features are simpler than the Ada features since modules contain no directly executable code whereas packages do.

The complexity that you predict is vastly overstated, and presupposes a particular implementation that is not required by the proposed draft standard. In any case, a very good implementation that would check for updated modules and provide efficient linking information is certainly not beyond the capabilities of compiler writers. Document preparation systems provide these capabilities today. The question is not will compilers have to deal with more complexity; the question is will users be given more and better functionality and is that functionality worth the cost. As far as modules are concerned, it certainly is. Modules will be the basic building block in Fortran 8X as subprograms have been in earlier versions of Fortran.

An INCLUDE statement is not needed, since its functionality is subsumed by USE. Admittedly, there is a necessary functionality that has been missing from previous versions of the Fortran standard. Suppliers of Fortran processors have filled this need in the easiest and cheapest way they could devise - by providing for the insertion of text residing outside the current compilation unit. This has usually been done in such a way that there is no source management provided by processors and for this reason it is error prone. It has been error prone in other ways as well, since there has usually been no provision to avoid or escape from name clashes. Vendors have provided a mechanism that is detached from the language itself and that has no interaction with the language. In many cases this has been handled by external macro processors or compiler directives. No one would design a language feature this way. One should look at existing INCLUDE facilities as stopgap measures, not as models for standardization. The USE of a MODULE is an elegantly designed language feature that contains the missing functionality.

Disallowing module procedures and disallowing a USE statement in a module would reduce the feature to little more than an intelligent, but weak, INCLUDE, since even most INCLUDE or macro facilities allow insertions (expansions) within insertions (expansions). A module provides a way of packaging data definitions and subprograms that logically belong together. It is a step beyond the current structure (or lack

thereof) provided by FORTRAN 77. The scope for reusable software will be much greater than the current subprogram libraries. Prohibiting modules from using other modules would be as restrictive to future Fortran 8X users as prohibiting subprograms from calling other subprograms would be to FORTRAN 77 users.

Saying there is little need for this capability is shortsighted. Providing for it is well within the talents of software producers. It is in the best interests of users that the full functionality embodied in modules and the USE statement be standardized.

1.3 Portability of Numerical Software

Your criticisms of the floating point portability extensions proposed for Fortran 8X confuse the issues of algorithmic portability and numerical stability, with those of linguistic portability. A program implementing an algorithm that is inherently sensitive to the fine detail of numeric precision, exponent-range, round-off mechanism, intrinsic function accuracy, etc. is inherently non-portable. A processor which represents floating-point quantities to some precision, but then proceeds to perform arithmetic or evaluate intrinsic functions to significantly less than this precision is simply an inadequate processor. Neither of these problems are reasonable subjects for program language design, nor do the facilities proposed for Fortran 8X attempt to handle these problems. The extensions proposed aim to provide linguistic portability in a context which assumes that different vendors will provide floating-point approximations of varying precision and range, and that certain unstated basic levels of quality in implementation will be met.

Fortran 77 has three major linguistic portability problems in the floating point area.

1. The language does not provide any standard portable way of exploiting all of the floating point approximation methods supported by any processor. Only REAL and DOUBLE PRECISION, and COMPLEX are supported linguistically. Many processors support three, or more, different floating-point approximation methods. The language needs to support use of these methods for real, complex and any other floating point data type.
2. Since an application that one processor may run adequately using the approximation method implementing REAL, may on another processor require DOUBLE PRECISION, the move of a program from one processor to another frequently requires major and distributed changes to the source codes. Similarly, a program may run adequately on one problem using REAL, but on another with different data values it may need to be run using DOUBLE PRECISION, again requiring major source code modifications. The language needs to be able to specify the floating point approximation characteristics required in a way which is processor independent and can be changed in a way which is at least potentially localizable to a few places in any program unit.

- 3. FORTRAN 77 introduced a major portability enhancement when it defined "generic" intrinsic functions. This allowed a program to be largely changed from REAL to DOUBLE PRECISION without having to search for every reference to an intrinsic function so as to change its name. This facility is required also for user written procedures.

The extensions proposed for Fortran 8X in this area solve these three portability problems. They do not handle, or attempt to handle, the sort of portability problem which could arise if a program were moved between machines with similar approximation methods but where one performed its arithmetic rounded and the other truncated. An application sensitive to such differences may well have numerical problems that are entirely non-linguistic. Provided such a program had been sensibly written the Fortran 8X floating point facilities would make it easy to force the program to use a more precise approximation method but otherwise this is not the type of problem envisaged as solvable by the language extensions proposed.

The optional addition of the parameters PRECISION and EXPONENT_RANGE allow the selection of the floating point approximation method to be done in terms of minimum approximation quality requirements. These parameter values are so constrained to make the selection of approximation method an entirely static choice. These new forms provide a processor independent method of selecting "single precision" on one machine and "double precision" on another. In effect a type attribute such as

```
REAL(PRECISION = 10, EXPONENT_RANGE = 50)
```

is an alternative spelling for REAL on large-word machines and for DOUBLE PRECISION on small-word machines.

Since this method of selecting floating point representation is not dependent on any unique spelling for each supported approximation method, it provides a portable linguistic device for providing access to any number of methods. Two may be selected and used, where appropriate, on a machine which supports only two, but on a machine supporting three or more, three or more may be used. This effectively solves the first two linguistic portability problems. The remaining one of allowing user-defined generic procedures is handled by allowing dummy arguments to be declared to have this generic property.

Declaring a dummy-argument with precision and exponent-range parameters specified as asterisks indicates that it carries this generic property. Although there are many possible implementation models for such procedures the one most commonly used by the committee in discussing the facility has been the "implied overload multiple specific version" model. The nature of the facility is so constrained that all generic objects in the procedure must be implemented by the same approximation method. This means that at most one specific version of any generic procedure is required for each supported approximation method (some versions may be excluded by other internal considerations). The calling program must then arrange to invoke the correct

specific version. Obviously since all the generic dummy arguments employ the same approximation method all the actual arguments associated with them must also. This will be guaranteed to occur and to occur portably if all the actual "generic" arguments have the same declared values for precision and range. This then provides a mechanism which portably allows the writing and the using of procedures which are generic over floating point approximations in a way entirely analogous to that for intrinsic functions. The Fortran 8X floating point facility provides for little or no run time cost and modest compile time cost, a highly portable (linguistically) method of expressing floating point programs, solving one of the major software portability problems of Fortran 77.

1.4 Deprecated Features

Deprecated features are listed in an appendix as possible candidates for the obsolescent category in a future draft standard. They are not marked in the text, and are fully supported features of Fortran 8x, residing compatibly with the new features. Decisions about these features will be made by a future standards committee.

Obsolescent features are marked in the main text, but are fully compatible with new features. They may be deleted by the next Fortran Standards Committee. The rationale for obsolescent features is that these features are available in a better way in FORTRAN 77 and are therefore redundant. Since these features are marked in a special type font in the draft for Fortran 8x, users will be made aware that this feature is redundant and encourage change to a better method. There is no easy way to remove features from a language standard. However, X3J3 believes that the architecture for Fortran will give users at least two standard revision cycles for removing a feature. Older languages, like Fortran, that have had a long and successful history, should not be left to expand beyond a reasonable number of features; nor should they be prevented from adding new modern features that might make the language too large. The method suggested allows Fortran to grow and change and remain within the bounds of size and complexity, if obsolete features are marked and therefore grow into disuse in the Fortran Community.

There are no Deleted features in Fortran 8x:

The Arithmetic IF has long been used with unexpected results due to the round off error possible in never achieving the zero branch. This feature is bad programming practice, and should be discouraged. (If the use persists, this feature may remain in the language, since deletions require that the feature has become little used.) These same arguments apply to shared DO terminations, branching to an END IF from outside the block, alternate RETURN and the PAUSE statement, all of which are declared obsolescent.

FYI
K

34
101

Subject: Comments on Harris Ballot Section 1.1
From: Kurt W. Hirschert

103b(*)KWH-14(Page 1 of 2)

This section was not assigned to me, so this should be considered an informal reaction to Kevin's comments in this section.

Kevin seems to be suggesting that X3J3 has neglected intrinsic types that should be a part of the language on the assumption that they can be implemented as data abstractions. He identifies three specific cases and explains why he believes data abstractions are not suitable replacements for intrinsic capabilities:

1. Character strings varying in length within a given maximum appear to me to be an obvious application for data abstraction. It can be argued that it would be inappropriate to add such a type as standard in a language whose primary emphasis is on numerical computing. The operations of such a type are easily expressed in terms of the operations on the existing fixed length character string type. Kevin claims that a derived-type version of a varying length character type could not take advantage of string manipulation capabilities built into the hardware. I know of no hardware string manipulation capabilities applicable to an intrinsic varying length character string type that would not be equally applicable to the variably sized substring of the fixed length type used in defining a derived varying length character string type. The key to the derived type being as efficient as an intrinsic type is that the processor must be able to expand the operator definitions in-line, so that a varying length character string variable VAR will nearly always be equivalent to VAR%FIXED(1:VAR%LENGTH) with no added procedure overhead. If a processor doesn't have this capability (and we currently don't require it), abstractions such as this may be unacceptably slow.
2. Our failure to adopt a BIT data type has nothing to do with data abstraction. Although it has been demonstrated that the semantics of a BIT type could be simulated using a data abstraction, but it has long been recognized that reasons a BIT data type are desired relate to performance, not semantics. The performance of a data abstraction is compounded from the performance of the intrinsic types and operations from which it is built. To get new performance characteristics, you must introduce new intrinsic types or operators.

The reasons that a BIT type is not part of the proposed standard relate to issues such as the question of whether it is appropriate to standardized a type whose primary applications will be processor dependent and questions relating to the form such a type should take. Should it be a single bit or a bit string? If a bit string, should it be left or right justified? If a single bit, should this be a single bit representation of the LOGICAL type rather than a totally new type? How should input/output and EQUIVALENCE apply to the bit type?

3. Similarly, I believe that the availability of data abstraction is unrelated to our failure to adopt some kind of pointer facility. Again, there have been demonstrations that abstractions built on subscripts may, in many cases, be semantically indistinguishable from abstractions built on pointers. Again the issue is performance. The aspect of performance that is relevant is not the speed issue cited in Kevin's comments. Using one-dimensional subscripts should typically add only one multiply (or shift) per node reference, with that single multiply being sufficient to support references to multiple fields in the node. (To put it another way, absolute pointers and offset pointers typically have the same performance and it takes a single multiply to convert a subscript to an offset pointer.) The real performance disadvantage of subscripts is that elements that it selects among must have been allocated all at the same time, while pointers can select among elements that have been allocated at different times. This added flexibility is necessary to efficiently implement some kinds of algorithms.

F42
K

Subject: Suggested Response to Harris Ballot Section 1.2
From: Kurt W. Hirschert

103b(*)KWH-11(Page 1 of 3)

- 1. "We are concerned because the new packaging features embodied in the MODULE and USE statements are complex, new, untested, and differ so widely from existing practice."

While it is true that the MODULE/USE facility was not based on existing Fortran implementations, similar facilities exist in a wide range of existing languages and language processors, from assembly languages to Ada®. Thus, the concept of allowing a programmer to package definitions or declarations, present them to the processor for conversion to a more efficient representation, and then later reference the converted form can hardly be described as new and untested. The specific details of how this is to be expressed in Fortran are new, but are similar in many ways to the existing practice in other languages.

- 2. "...standardize the spelling of a textual INCLUDE facility."

In including identical program text into a program at different places (typically, in different program units), one can either intend that this text have the same meaning at each point or different meanings. For example, if the text declares the attributes of a variable, it can either be intended that the same variable be declared each time or that different variable are being declared that happen to have the same name and attributes. Although one can imagine applications for textual inclusion in which different meaning are intended, currently existing INCLUDE facilities are used primarily in cases where the same meaning is intended. In other words, INCLUDE is used as a means of conveying static information to multiple program units.

There are a number of disadvantages to conveying such information in textual form. Changes in the implicit typing rules or the presence of additional specifications may cause identical text to have differing interpretations. In a program which makes heavy use of textual inclusion, the cost of reinterpreting the included text in each program unit can be a significant part of the cost of program translation. (Cf. the disadvantages in using text (formatted input/output) in transferring dynamic data between programs or program units.) The ordering constraints imposed in Fortran have the effect that it will often not be possible to package related items (for example, related variables and constants) together in a single text inclusion. Similarly, the ordering constraints may limit the ability to hide implementation decisions in the included text. For example, a text inclusion providing a symbolic name for a unit number may have to be positioned differently depending on whether that symbolic name is a constant or a variable. Because text inclusion offers no flexibility in the names it imports, these names must, in effect, be managed as though they were global names, significantly enlarging the name management problem.

In addition to the disadvantages that apply to all forms of text inclusion, there are disadvantages that apply specifically to the form of text inclusion most commonly implemented. This form places the text to be included in a separate file and has the name of the file as part of the directive requesting the inclusion of the text. Because the use of this facility forces the source for a program to be spread over multiple files, the packaging of such a program for transport to another machine is complicated. Because file names are processor dependent, the program source will probably have to be modified before it can be made to run on another machine. The fact that portions of the program must reside in particular files also complicates the process of maintaining multiple versions of a program. Unlike the use of the processor-dependent name in the OPEN statement, where most references to the file are made through the unit number (thus localizing the processor-dependent code which must be modified in transporting the program to a new processor), the processor-dependent file name

F-12
Subject: Suggested Response to Harris Ballot Section 1.2
From: Kurt W. Hirschert

103b(*)KWH-11(Page 2 of 3)

appears in every reference to an include file and thus tends to be scattered throughout the program.

Because INCLUDE was seen by X3J3 as being primarily a means of transporting static information between program units, X3J3 has, in the past, rejected the standardization of INCLUDE on the grounds that MODULE/USE offered most of the functionality without suffering from the disadvantages. In discussing this ballot comment, members of X3J3 recognized that INCLUDE is relatively easy to implement and already widely implemented, so that its inclusion in the standard would not be an undue burden on implementers, in spite of this overlap in functionality. Accordingly, a proposal for a standard INCLUDE facility was drafted based on the INCLUDE facility in Mil. Std. 1753. This proposal was accepted (rejected) by X3J3 on a (-) vote. (* Fill in last sentence based on actual vote. *)

3. "...eliminate module procedures..."

Although a reduction in the complexity of the syntax of the language was viewed as a secondary benefit of such a change, the primary benefit was seen as being the ability to implement modules purely as information to be used in program translation, without any "object form" to be included as part of the execution of the program. With modules as they are currently described, this object form would be used to specify the translations of the module procedures, the storage for shared variables, the initialization of shared variables, and the memory representation of shared constants. Eliminating module procedures would also eliminate the need to use the object form to specify their translations. Storage for shared variables could be handled by an internal conversion to COMMON. Many processors allow COMMON to be initialized by any program unit that references it, not just BLOCK DATA program units, so it may be possible to handle the initialization of the shared variables by generating the initialization in each program unit that USEs the MODULE, provided that the processor can accept multiple initializations of the same storage without complaint. The memory representations of constants can be handled either by generating them in each program unit that actually uses a given constant (possibly using additional storage because of memory representations for a given constant appearing in multiple program units) or by treating them like initialized shared variables. (On most machines, the translations of module procedures could also be treated like initialized shared variables.) In short, elimination of module procedures allows elimination of the "object form" only if certain other conditions hold or tradeoffs are acceptable, and elimination of the "object form" may be possible even if module procedures are not implemented.

The disadvantages to eliminating module procedures include the enlargement of the global name management problem resulting from using external procedures (with global names) rather than module procedures (with local names), the loss of opportunities for optimizations such as the in-line expansion of these procedures, and the loss of the ability to use MODULEs as a means to package the definition of an abstract data type (use would still be packagable by including interface blocks).

Given the limitations on the benefits of this change, the disadvantages would appear to outweigh the advantages.

4. "...disallow the USE statement inside a module."

It is claimed that this "contributes greatly to the name space management problem." An analysis of the name management problems involved in supporting MODULE/USE suggests

FYI
K

Subject: Suggested Response to Harris Ballot Section 1.2
From: Kurt W. Hirschert

103b(*)KWH-11(Page 3 of 3)

that the only additional name management requirement resulting from allowing the USE statement within a module is the necessity to recognize the difference between the module in which the entity was originally declared as distinct from the module from which it is being USEd (and also the corresponding possible difference in the name of the entity in those two modules). (This information on the original module allows a processor to distinguish between the case of there being two different entities with the same name and the case of a single entity being made accessible by multiple USE statements.)

Building complex packages based on simpler packages is common practice in a number of areas, and there is no reason to believe that this practice is less applicable to definitions and declarations than it is to procedural abstractions. Again, the expected simplification resulting from imposing this restriction does not seem significant enough to warrant the loss of functionality.

- 5. Although this section is titled "Source Manipulation", the MODULE/USE facility is not generally viewed as a source manipulation facility, but rather as a means of transporting static ("compile-time") information. True source manipulation facilities, such as textual inclusion, are merely one alternative for solving the information transfer problem.

Ω

Comments on the NO vote.

Dick Hendrickson

The major reason for my voting NO is the large number of technical errors and holes in the current document. I have serious reservations about some of the features in Fortran 8x and several suggestions for new features, however, I believe the document should be forwarded for public comment as soon as possible. With this in mind I will change my NO vote to YES if the bulk of the technical problems are eliminated, without any action necessarily being taken on the features. The number of flaws in the current document, in my judgment, make it unlikely that we will get any meaningful public comment. We need to make a major effort to clean up IDENTIFY, USE and modules, and clarify the BNF.

I have detailed comments on each chapter. I did not have time to review the Appendix or glossary. The comments for each chapter include page number, starting and ending line number (99 generally means "continued on next page") and a severity level.

Edit: Basically a rewording of something which isn't clear.
 Hole: A missing or contradictory item.
 Minor: What I believe to be a mistake. The text is clear enough, I just believe we should do something else.
 Major: A significant mistake. I strongly believe we should do something else.

Throughout the comments I used the term "depricated" when I should have used "Obsolescent".

Before I would vote YES essentially all of the edit and hole comments and items 1,2,4,6,7 and 13 below must be resolved.

I also have several general comments which didn't seem to fit in to a line by line chapter list.

1) We should replace "assumed shape" with "assumed extent". Shape includes both rank and extent, however, only the extents are assumed. The same applies to "deferred shape". The current terms are correct, but confusing.

RESPONSE TO RICHARD HENDRICKSON BALLOT

X3J3 appreciates your extensive and careful reading of the draft Fortran standard. The individual comments of both an editorial and substantive nature have been handled by full committee and either accepted, or considered and rejected.

Each of the general comments in the introduction to your ballot have been considered and either accepted or rejected, or withdrawn by you. The following notes address these general comments.

1. Assumed Shape and Assumed Extent

The X3J3 committee discussed a suitable alternative descriptive term for the concept of "assumed shape". After considerable discussion, the committee agreed that the term "assumed shape" was slightly misleading but that the suggested term "assumed extent" seemed more confusing as the phrase, taken literally, includes the FORTRAN 77 concept of "assumed size". A more appropriate term might be "assumed extents" but the confusion created by two very different concepts being distinguished by a singular versus plural term was unacceptable.

In support of the current term "assumed shape", the committee felt that a basic principle for the language was that once an array was declared, its rank was fixed. The definition of "assumed shape" currently makes this assumption -- "An assumed-shape array is a dummy argument array that takes its shape from the associated actual argument array". An added sentence to clarify the definition might be helpful, namely "An assumed-shape argument is required to be of the same rank as the associated actual argument array."

2. Ranged Arrays

Currently RANGE is an attribute of an array. Local arrays obtained via a USE statement may not have any attributes changed in the USEing subprogram. Thus, such an array may not be ranged if the module writer neglected to specify RANGE in the module. Although it might be nice to relax this restriction, it does not prevent you from doing what you want. If you IDENTIFY a local array onto the USED array, you can then effectively RANGE the USED array and the range information is local to the subprogram. X3J3 feels that since the basic functionality is provided, a change to relax the restriction is not required.

3,5,8,10,11.

After the initial ballot of X3J3, the committee reviewed the comments that indicated that the language was too large. A compromise was reached within the committee to reduce the size of the draft. Many new features were, with regret, removed in order to satisfy this general complaint. Your ballot asks for new features or a return of the features placed in Appendix F; an executable module and putting BIT data type and

vector valued section selectors back into the language. The reduced language that was approved by the committee should remain. Further work by the committee would be counter-productive and would represent "thrashing" over ideas that have had much committee time and would not necessarily change the votes on these issues.

The CONTAINS statement was reviewed again based on another ballot, and left as it stands. The "%" structure qualifier was reviewed again also. This feature was again left as in the draft. Further progress by X3J3 cannot be made in these areas where repeated discussions result in the feature remaining as it is currently defined.

4. Complexity of Constant Expressions

The X3J3 committee considered the concerns expressed in the above ballot comments but felt no changes were required for the following reasons.

FORTRAN 77 currently permits constant expressions to include all of the intrinsic arithmetic, character, and logical operators with constants or constant expressions as operands, except for the exponential operator ** which is limited to integer powers. Therefore, cross compilers even for FORTRAN 77 currently must address the issue that operands may have a very different representation on the host than on the destination machine.

On the other hand, FORTRAN 77 does not require that such expressions be evaluated at compile-time but can be postponed to runtime. Whether it is an advantage or disadvantage to do so depends upon the machine and the properties of the object files and the linker/loader.

The issue, therefore, is not one of precedent but of the number of operations (including those embodied in intrinsic functions) that must be supported in what appears to many to be compile-time expressions. Several models of how this can be implemented are readily apparent. Which one is sensible will depend upon the machine and also on the tradeoffs between storage and execution and on whether the evaluation of such expressions is borne at compile-time, linker/loader-time, or execution-time. (One model that addresses the particular example given in the ballot comment RA 9.1 is one where the expressions are evaluated just prior to the execution of the main program and the values inserted into the appropriate places in the load image.)

With the determination of execution-time determined extents for arrays, the need for arbitrary constant expressions in specification contexts becomes apparent. However, the use of the entire intrinsic function set appears at first to be excessive. When attempting to formulate a rationale for including some but not all of these functions, it became clear to the committee that the choices were somewhat arbitrary and were dependent on the applications considered. Only a few such functions seemed to be clearly not useful. But providing a small list of excluded functions seemed not to widened the set of feasible implementations and at the same time was user unfriendly.

7,13. Withdrawn Issues

These issues were withdrawn by the commenter.

9, 12. Resolved Issues

These issues were resolved in a proposal contained in another ballot.

Subject: Comment on Balloting Procedures
From: Kurt W. Hirschert

103(1)KWH-1(1 page)
page 1

The ANSI procedures under which this work is being done appear to be based on the expectation that the technical committee preparing a standard will not forward it for public comment until the committee believes it has a document suitable to be the final standard. Under this expectation, the public comment period is really only a safeguard that the committee has done its job properly, and the effects of public comment on the document should normally be only minor technical and editorial clarifications.

This expectation is reasonable for most of the work done under these procedures. If the standard is merely a ratification of existing practice, with few significant technical decisions to be made by the committee, then public comment need only be concerned with whether the existing practice has been correctly described. If the standard is short enough that reviewing it is not a major task, it can be expected that people interested in commenting on the technical decisions made by the committee will be able to do so during the time the document is being prepared, so that no new viewpoints will be revealed during public comment. If the expected users of the standard are sufficiently homogeneous that all relevant viewpoints are adequately represented on the technical committee, one can again expect that the public comment period will reveal no new viewpoints.

This preparation of this document, however, involved the integration of features taken from a great many sources and thus involved a great number of technical decisions. The document itself is so large that even when people outside the committee have been willing to invest the time to review it, they generally have not been able to complete the review of one version of the document before a new version replaces it. There are relevant interest groups that are not represented on the committee, notably the suppliers and users of Fortran on personal computers, for whom the cost of participating on the committee is not economically justifiable. As a result, it is extremely unlikely that public comment on this document will not result in significant technical changes, no matter how long the committee works on the document beforehand.

Given these circumstances, a vote to forward this document for public review does not necessarily mean that one believes that this document should be the revised standard. It does mean that there is nothing in this document that would make it unreasonable for this document to be the revised standard.

I am voting to forward this document for public review in spite of some reservations I have about it. I am concerned that some of the new features (e.g., RANGE) may be a less than optimal choice for solving the problems they are intended to address, but I note that Fortran has survived the suboptimal decisions such as using column-major storage order rather than row-major order. I am concerned that some features (e.g., assumed precision) may have been formulated in ways that make possibly desirable extensions unnecessarily difficult, but again I note that Fortran has survived other roadblocks to extension such as the syntactic difficulties in using expressions to specify the initial values in a DATA statement. I am concerned that there are features (e.g., pointers) that ought to be made a part of the language at this time but that are not included in this document, but once again I note that Fortran has survived such omissions as the failure of Fortran 66 to include end of file testing (corrected in Fortran 77) and the failure of Fortran 77 to make end of file testing work on all sequential files produced by Fortran (corrected in this document). In other words, even if all of my concerns about this document should prove valid, there is no reason to believe that a failure to address them would irreparably harm the language.

All of these concerns are of a nature similar to those that I would expect to be expressed during public comment, even if these particular concerns were addressed beforehand, and some are such that they will be more easily evaluated after a wider sampling of public opinion has been obtained. Thus, considering these concerns as a part of the public review process seems to be the most expeditious way to proceed in the production of a revised standard.

Ω

RESPONSE TO KURT HIRCHERT BALLOT

Each of your specific editorial and minor substantive issues have been reviewed by the committee, and either accepted or rejected.

RANGE Facility

In your general comments, you express a concern that the RANGE facility is less than optimal. Please see the response to the Robert Allison ballot on this issue.

A Pointer Facility

You make a request that pointers should be in the language. Understandably the proposal to reduce the size of the language that was reached after the first ballot did not please everyone. Many favorite features, including pointers, do not appear. The committee felt that new additional facilities should not be included even though many features discussed by the committee including BIT data type and event handling, as well as pointers, have much merit. Taken altogether these features resulted in a language that to many appeared too large. Please review responses to other ballots on these issues.

RESPONSE TO TRACY HOOVER BALLOT

Your first editorial comment was rejected, your ^{second} editorial comment was handled in another ballot comment.

15.1

RESPONSE TO ANDREW JOHNSON BALLOT

Your first two editorial comments were handled in another ballot comment.
Your third was discussed in full committee and resolved.

16.1

Response to Letter Ballot on submission of FORTRAN 8X to X3 for processing as an American National Standard

Anil Lakhwara

Peritus International Inc.

December 31, 1986

Question:

Do you approve the draft Fortran Revision (Fortran 8X) of X3.9-1978 for submission to X3 for further processing as an American National Standard ?

Answer:

Peritus International votes no on this ballot.

Reasons/comments

Peritus's major concerns with the current form of the draft FORTRAN revisions are:

- 1. Non conformance of S8.103 to X3J3 Charter.
- 2. Size and complexity of the language.
- 3. Implementability of the language.

Based on these concerns, we recommend that this document not be forwarded to X3. Given below is a brief discussion of these problems

1. Non conformance of S8.103 to X3J3 Charter.

X3J3's charter was to standardize FORTRAN based on currently implemented language extensions , with exceptions only when there is clear evidence of the need for change. In our opinion this requirement has not been met; instead, S8.103 is a new language with Fortran-77 as a small subset. S8 draft revision reflects little concern for the history or the virtues of the FORTRAN as we know it today. The relationship between the present standard (X3.9-1978) and its predecessor (X3.9-1966) is clearly seen; this cannot be said for the present standard and the proposed S8.103.

Fortran standard affects all areas of science and technology; X3J3 is responsible to both the Fortran community and to those affected by the results of Fortran applications, to

RESPONSE TO ANIL LAKWARA BALLOT

Your ballot contains three main points; 1) that the draft does not conform to the scope and program of work as stated in the project proposal, 2) that the language is too large and complex, and 3) that the language is not implementible.

1) Conformance to Project Proposal

The scope and program of work as stated in the project proposal has been followed in processing the draft standard. One of the statements is that "X3J3 carry our procedures to maintain the continuous responsiveness of the standard to industry needs". The new features, for example arrays and data structures, are in response to these needs. X3J3 has acted as a liaison group (as required) with the international group concerned with Fortran (Working Group 5), as well as maintained other liaisons with data base and graphics committees, and with the engineering and mathematical community.

Members of X3J3 have published technical reports on the interpretation of the current standard and provided maintenance of this standard, as charged. The criteria for evaluating proposals has been published in the minutes, we have reviewed and studied collateral standards, and we have had many forums to publicize our work in the past several years, keeping the user community informed of our activities. The draft document (S8) has been distributed widely several times a year over the past three years.

We (X3J3) believe that we have been in conformity with our charge, that we have kept the user community informed of our progress and been responsive to user, government and industry needs and that the draft standard is in compliance.

2) Size and Complexity of the Language

This comment contains phrases such as "immense size and complexity", "FORTRAN 77 is just a small subset", "radical departure from FORTRAN 77", and two distinct languages glued together". The comment then suggests that derived data types, modular definitions, specified precision, and internal and recursive procedures be removed from Fortran 8x.

Fortran 8x is a major extension of FORTRAN 77, but it is neither "immense" nor "radical". The extensions constitute approximately one-third of Fortran 8x, and great attention has been paid to making them as "Fortran-like" as possible. For example, the new loop control structure is an extension of the FORTRAN 77 DO-loop (though many believe a completely new and different loop structure is "better").

Of the approximately one-third of Fortran 8x that is an extension over FORTRAN 77, about half is due to the array processing facilities and the numerical computation enhancements. Both of these appear to be greatly desired by the scientific user community. They are designed as straightforward extensions of the corresponding base functionality in FORTRAN 77, and improve Fortran considerably in the numerical

computation and array processing areas.

With regard to the specific suggestions for removal:

- a. **Derived Data Types.** Data structures are among the most widely desired features by the Fortran user community. The Fortran 8x derived data type is a very simple data structuring facility, similar to that in Pascal or C. Despite its name, it is not anything like the Ada derived data type facility.
- b. **Modular Definitions.** The MODULE/USE feature of Fortran 8x constitutes only a very small (about 3%) of Fortran 8x. It is far superior to INCLUDE for defining entities (such as global data) that are to be shared by different program units, a functionality that is very important to Fortran. Though implementations may choose to use a dependent compilation model for processing modules, modules may be processed as "filtered INCLUDE". That is, though not as simple to process as the textual INCLUDE facility found as extensions in most FORTRAN 77 implementations, in terms of a dependent compilation model the MODULE/USE facility has neither greater nor less implications than does the typical INCLUDE.
- c. **New Syntax (for REAL).** The numerical community consensus is that the specified precision REAL facilities contained in Fortran 8x are most appropriate means of achieving portable, robust numerical computation capabilities. These facilities have been designed in close coordination with the numerical community, taking care to configure them as Fortran-like extensions to the FORTRAN 77 REAL and COMPLEX facilities. They could hardly blend more with the FORTRAN 77 numerical woodwork.
- d. **Internal and Recursive procedures.** Internal and recursive (especially recursive) procedures are widely implemented extensions to FORTRAN 77. As implementations are simple and the functionality is widely available and wanted, it does not seem advantageous not to standardize internal and recursive procedures in Fortran.

3. Implementability

The issue of "implementability" is a difficult one, involving judgment calls about what is "large" and what is "simple". Your measures imply that Fortran 8x is about twice the "size" of FORTRAN 77. (Your point ii. Also, the current FCVS validation suite is in excess of 100 K lines.) While this may be a large change, it does not seem large for a 10 year evolution.

X3J3 does not see why object code quality or compile time efficiency must suffer for codes which are essentially FORTRAN 77. Perhaps the largest addition to the language is the array processing syntax. Many vendors have implemented all or part of this syntax and none have complained to the committee about compile time requirements nor about the inability to generate efficient object code. The new features perform powerful operations. They might appear to compile "inefficiently", however, they

let a programmer express complex operations concisely. This can allow for good optimization since the processor will not have to decompile basic FORTRAN 77 operations to do large scale array operations. A good example of this would be the FORTRAN 77 INDEX function. This can be written as a nested pair of DO loops with IF statements inside. This is a difficult construct to optimize and it is difficult for a casual programmer to get the end conditions right for the loops.

There is a large degree of separation between the new features in the language. Array processing is essentially independent of derived data types, etc. Users, who are not "professional" programmers, should be able to continue in the FORTRAN 77 mode and begin to use the new features as their applications warrant. The ability to extend the language via derived types and operators is there. However, these extensions can be made by professional programmers and library managers without direct involvement by the normal user. (This is the current situation with commercial libraries such as LINPACK or IMSL.)

The dependent compilation model does not necessarily impose a great burden on a processor, although optimizers may use it extensively. All that is required is a method of accessing the interface information of the USED routines and a method of name management to allow different modules to have routines with the same name. These both are new to Fortran. However, the former doesn't require a compiler to do anything more complex than is currently required to resolve generic function references except that it must work dynamically at compile time. The latter has the processor do something that the user must now consider. Many new processors are implementing multi-subroutine optimizations, so dependent compilation is not a totally new feature.

50

103(*)BAM-0
ballot

GRUMMAN

M e m o r a n d u m

1986 DEC 1+

FROM: Bruce A. Martin
TO: X3J3

SUBJECT: Letter Ballot on draft Proposed
American National Standard
Programming Language Fortran

REFERENCE: X3J3/S8.103 draft Fortran Revision

I am in favor of submitting the draft Fortran Revision X3J3/S8.103 to SPARC and X3, and then for Public Review soon after. I understand that X3J3 will process these Letter Ballot comments before sending it.

It is my opinion that this draft is in far better shape than the previous letter ballot draft, and that the language has improved since then. I believe that Fortran 8x, as defined by this draft, is now more or less acceptable -- although I wish the language could be made even smaller, more regular, and more extensible.

Nevertheless, there are some further improvements that should be made to the language, as well as numerous editorial fixes and readability enhancements to the document.

My comments for this letter ballot consist of discussions, complaints, laments, and rationales, PLUS a set of specific proposals (with text), all of which are attached.

RESPONSE TO BRUCE MARTIN BALLOT

Your ballot consisted of many proposals, a few were accepted, and a few others were handled in another ballot. The remaining issues have already been discussed in the past two years and decided by the full committee. Some of these issues were part of the revision of the draft that resulted after the first ballot.

For example, extensive discussions were held on the nature of the "decremental" features. A change was made to include three categories of these features (deleted, obsolescent and deprecated) in order to soften the impact of changes for Fortran users. The current draft allows a staging process where users have at least 20 years to move to a better feature. The content of the obsolescent and deprecated features was discussed over a period of several meetings. Your ballot would add features to the obsolescent list, while other ballots would remove features from this list. We feel that the current draft should remain as defined.

Another example involves the CONTAINS statement. In a recent meeting, the issue of deleting CONTAINS altogether was revisited. The issue of renaming was also considered. X3J3 decided not to modify its position with regard to CONTAINS.

Your ballot contains many excellent ideas that have been discussed over the past two years. Each of your issues was discussed again in a subgroup during ballot processing, and evaluated. Those issues that were approved were brought to the full committee. It is not possible to revisit issues in a never ending succession of meetings that endangers the possibility of ever getting a draft standard out to the public.

Item 9 -----

Page B-2, lines 5-8. Making the PAUSE statement obsolescent is silly.

Item 10 -----

Page C-14, lines 21-31. There has to be something wrong with the assertion that a standard-conforming program on one processor can become non-standard-conforming on another: "(On that processor, the program would also be viewed as not conforming to the standard because of the references to the nonstandard intrinsic procedure.)"...

Item 11 -----

Proposal: Move Appendix F back to the main body of the Standard.

I believe that the proposed Standard was not improved by moving a small list of items to an Appendix titled "REMOVED EXTENSIONS".

Item 12 -----

Proposal: 1) Move the Bit facilities of pages F-1 through F-12 back to the Standard. 2) Move the Variant Structure facilities of pages F-12 through F-14 back to the Standard. 3) Allow bit arrays in variant structures to be overlaid with other arbitrary data fields. 4) Relax the rules on definition/undefinition to allow simultaneous definition of variants in a variant structure, if one of the variants is a bit field.

Discussion: Many implementations of Fortran contain extensions to allow bit data and bit manipulations. Many of these same implementations allow simultaneous definition of equivalenced variables, even though this is not allowed in the current Standard. The new proposed Standard makes no allowance for these widespread practices.

This proposal would make available the power of bit manipulation facilities (including the replacement as appropriate of logical data with more compact bit data) and the added power of the variant structure. It would legalize a widespread extension of EQUIVALENCE, but confine its use to a specialized, modern data structure.



Item 13 -----

Proposal: Move the facility of Vector-Valued Subscripts, pages F-14 and F-15, back to the Standard.

Discussion: The gather/scatter mechanism in array processing is too important not to be included in a straightforward manner in the language.

III. CONCLUDING OBSERVATIONS

There is much to recommend in this proposed Standard. Array processing, derived types and derived operators, user-specified precision, and dynamic memory management are obvious examples. Some things are missing: bit facilities and pointers are obvious examples. The public will have its own favorites for addition and deletion.

Much has been said of the size of Fortran 8X; many believe it to be "too big." I am concerned not so much with the Standard's size as with its complexity. The document is complex and interrelated; many of the language features are complex and interdependent; and implementations may be complex and tedious to use.

A hallmark of Fortran has been separate and independent compilation. My biggest concern is that with the mandatory usage of interface blocks (to pass assumed-shape or allocatable arrays) and of modules (to define derived-type dummy arguments), independent compilation will be a thing of the past.

RESPONSE TO ALEX MARUSAK BALLOT

Your editorial items have been discussed and accepted, or considered and rejected. Your substantive items 1-8 have been examined and either accepted or rejected by full committee. Items 9-13 and your concluding observations are discussed below.

- **Item 9 Pause Statement.** There is no clear unique functionality provided by the PAUSE statement. Making it obsolescent is consistent with the definition of obsolescent (better ways exist in FORTRAN 77), and makes it possible for PAUSE to be removed from Fortran in the future. It does not mandate removal, however, if upon closer inspection removal seems inadvisable.
- **Item 10 Nonstandard Intrinsic Procedures.** This is a FORTRAN 77 provision, and therefore total conformance of Fortran 8x with FORTRAN 77 requires this statement.
- **Item 11 Appendix F.** The widely supported "post-ballot" draft, of which Appendix F was a major feature, was developed after the March 1986 negative letter ballot. The most common comment of those negative ballots was that certain features should be removed from the draft standard; Appendix F contains those features so agreed upon.
- **Item 12a BIT Data Type.** Moving the BIT data type to the appendix was part of the post-ballot draft. Complicating the entire BIT issue is that there appears to be no consensus on the bit-array versus bit-string functionalities, and that none of the BIT proposals provide for BIT stream I/O.
- **Item 12b Variant Structures.** Variant structures constitute an "equivalencing" mechanism, providing this functionality in the absence of the FORTRAN 77 EQUIVALENCE statement. Part of the post-ballot draft was to restore the EQUIVALENCE statement to unobsolescent form in the draft standard, making it feasible to simplify the standard language somewhat by moving variant structures to Appendix F.
- **Item 12c BIT Arrays in Variant Structures.** This is allowed in the BIT and variant structure facilities described in Appendix F.
- **Item 12c BIT Equivalence.** The functionality of interpreting the bit pattern of a data object as a data value of different type is provided by the TRANSFER intrinsic procedure adopted at X3J3 meeting #101 in Halifax (August 1986).
- **Item 13 Vector-Valued Subscripts.** As part of the post-ballot draft, it was agreed to move vector-valued subscripts to Appendix F. Instrumental in this

decision was the problem of satisfactorily handling the many-to-one problem in assignments involving vector-valued array sections.

Concluding Observations

Interface blocks, while in many cases most logically placed in modules, need never be so configured. The presence of an interface block in a procedure subprogram does not constitute "dependent compilation" any more than does, say, the specification `INTEGER X ;` in both cases, specification information is contained in the program using that information. When modules are used, specification information is located in a different program unit, which requires the availability of the module (or the equivalent information) during compilation of the using program unit.

Though implementations may choose to use a dependent compilation model for processing modules, modules may be processed as "filtered INCLUDE". That is, though not as simple to process as the textual INCLUDE facility found as extensions in most FORTRAN 77 implementations, in terms of a dependent compilation model the MODULE/USE facility has neither greater nor less implications than does the typical INCLUDE.

SLAC MEMORANDUM

December 17, 1986

To: X3J3

From: Len Moss
Sunnie Sund

Subject: Comments accompanying X3J3 Letter Ballot of 21 November 1986

We have discussed the current draft of Fortran 8x (X3J3/S8/103) with a number of concerned Fortran users at SLAC. Based on these discussions, we are voting "NO" on the letter ballot dated 21 November 1986.

The primary reason for this vote is the large number of technical and editorial errors still remaining in the document. While it is true that the document has improved considerably since the previous letter ballot in March, we still find a great many mistakes. Most of these problems are relatively minor and would probably not obstruct a careful reviewer from understanding the draft standard; however, a substantial number remain which are potential sources of great confusion. In addition, there are a number of cases where the current text is quite clear but also wrong, that is, disagrees with what has been passed by the committee. For example, R220 says that a module subprogram may not contain even the one level of internal procedures allowed for external subprograms and main programs. This makes it impossible to simply embed an external subprogram in a module in order to make the interface explicit, and is not, I believe, what was intended when the nesting of internal procedures was prohibited as part of the compromise plan.

Most, if not all, of these problems are non-controversial, so if their total number were small we might consider voting YES with comments, in the expectation that they would all get fixed during letter ballot processing. However, the number is not small, and our rate of convergence on a readable document has been quite slow. We feel very strongly that X3J3 should not publish a document until the whole committee is convinced that it is understandable, and this document does not yet meet that criterion.

We are enclosing separate lists of problems and proposed fixes, including text where possible. This is not a complete list, however, because the errors are very numerous and the time we have available for reviewing the draft is limited.

In addition to the technical and editorial problems discussed above, we have several additional concerns about major features, or their lack, in the current draft. At recent meetings, it has become clear that X3J3 is not willing to consider such major modifications at this time, and so we will withdraw our previous objections to publishing a document without

25.1 + 25.2, 25.3

RESPONSE TO LEONARD MOSS BALLOT

Your ballot contains two principal concerns, and in addition expresses need for a number of new features. You say that 1) the document is not consistent with what we passed, and 2) that there is an expectation that new features may be added or deleted as a result of public review. A response to the request for each of the new features that you would like in the language follows addressing your general comments.

Document Correctness

As to your point about document correctness, X3J3 believes that after a year of editorial work including two ballots that the document is as correct as is reasonable. The main body of the standard is 161 pages with 65 pages in appendices not including a glossary, or the indices. It is difficult for forty or more persons to agree to an editorial style or method of description for a technical document that long. The committee has done an excellent job following the ANSI style manual and cooperating with the editor. There comes a point of diminishing returns in processing edits and small substantive changes that we have reached. It is time for a public review document.

New Features during Public Review

The purpose of a public review is to determine if the reduced language that X3J3 reached after the first committee ballot is a viable one. A number of features were removed from the draft since the ballot indicated that the language was too large. (Many members now do not believe that the language is too large, nor was it too large at the time of the first ballot.) The present document represents the position approved by the committee. It is a viable standard as presented. Further processing of what should be in the language by X3J3 would represent "thrashing" over ideas that have been discussed repeatedly. It is time that the public review the draft. Further work by the committee without statements from the user community in a formal public review process would be counter-productive. (We have consistently reached users in public forums and in large distributions of our working document. Many interested persons are waiting for the formal review before carefully examining the draft.)

If the standard as it currently exists passes all further processing within ANSI and ISO, the committee feels that it would be a good standard for Fortran. The 2/3 requirement for voting ensures that arbitrary changes would be difficult to make. There are many implementations of new features in the current draft. A new standard is needed as soon as possible. The current ballot indicates that most of the members believe that this is so.

We appreciate your editorial comments on the draft. All of them have been considered by the full committee and either accepted or rejected. Your specific items for addition to Fortran are addressed below.

Request for Various New Features

X3J3 has spent considerable time discussing pointers and bits.

Many versions of a pointer facility have been discussed. However, none seemed to meet all of the needs. The need for pointers as a memory management tool is met by automatic and allocatable arrays. Pointers for recursive data structures can be simulated by using subscripts into an array of derived type. This does require that a fixed (or at least allocated) array be defined and limits the number of structures available.

The bit facility was removed from the draft in an attempt to simplify the language and reduce its size and complexity. Since the bit facility was removed there have been no formal proposals to implement either the IRTF or MIL-STD bit functions.

Intrinsic operator overloading permits, but does not require, obscure coding. However, operators are currently "overloaded" in the sense that type coercions take place. Array references look like function references, character variable references or simple variable references, depending on the subscripts. Overloading of intrinsic operators allows a precedence to be associated with the operator. Without this precedence it is necessary to use parentheses to force evaluation order. This is different from ordinary Fortran notation and seems to be as potentially error prone as "excessive" overloading could be.

The decision to release the draft for public review is a difficult one. The draft is a workable compromise that most members can support. As a compromise it is necessarily imperfect. While many hope that there will be a ground swell of support for their particular point of view, an affirmative vote does mean that the draft is an acceptable standard.

I am voting no on the current version of the standard as defined in the February 1987 version of CC. The reasons are given, in detail, later. I consider the standard, in its present form, unlikely to be accepted by the public. Nevertheless, in order to make progress, it is time the public is permitted to make its comments on the proposed standard. I have, therefore, divided the changes I advocate into two classes: recommended and mandatory. I believe both classes of changes are necessary to make the standard acceptable to the public. However, if the changes that I have designated mandatory are made (they are given in detail below), I will change my no vote to a yes vote. This yes vote will, however, be a yes vote with comments. The comments will be a repeat of the reasons why I believe the proposed standard should be modified.

Reasons for Changing the Proposed Standard

There are several reasons why I recommend changing the standard. They are:

- (1) The standard has deviated too far from FORTRAN 77.
- (2) Features have been added to "modernize" the language that now result in many duplicative and confusing features. This is demonstrated by the existence of 16 obsolescent and deprecated features. Because of this, the language will, at the Fortran 9X stage, face an unattractive choice: either the duplicated features will be removed, resulting in upward incompatibility; or they will remain, causing the language to remain clumsy indefinitely.
- (3) The language is considerably more complex than its predecessor. The language can no longer be readily "contained in the head". Because of this complexity, the language will change from one that is readily usable by nonprofessional programmers into one that is not. Nonprofessional users make up a significant part of the Fortran programmer community—it has been estimated that the typical aerospace engineer spends a quarter of the workday programming (in FORTRAN).
- (4) Because of the large increase in the size and complexity of the language, vendors are unlikely to implement the whole language in their initial releases. A quite probable scenario is that the standard will be implemented piecemeal, with different vendors implementing it, in differing orders, over a lengthy period of time. This will cause considerable portability problems.

Recommended Changes

For the preceding reasons, I recommend that this committee (X3J3) implement two standards. This has been exhaustively discussed before. The first one would be upward compatible with FORTRAN 77, containing no obsolete, obsolescent, or deprecated features. I will refer to this language as Fortran A. The second one would remove, immediately, all the obsolescent and deprecated features of FORTRAN 77 and would incorporate most of the new features proposed for Fortran 8X. I say most, because, for example, some features are largely present to make existing FORTRAN 77 code execute efficiently. Thus the interface statement could be almost completely eliminated. I will refer to this language as Fortran B.

RESPONSE TO IVOR PHILLIPS BALLOT

The main points in your ballot are 1) dividing the language into two separate languages, and 2) adding two new features to the language, as in your BIT String Data proposal and the OVERLOAD statement. These are addressed in the following notes. Your 11 editorial comments have been considered by the committee and either rejected or accepted.

Divide Fortran into Two Separate Languages

X3J3 has considered at great length and in great depth, the issue of a "partitioned" Fortran. One form of partitioning considered (during 1985-1986) is that of two separate languages. Earlier another form, that of a "core + modules" architecture, was considered, adopted, and subsequently rejected. In both cases multiple versions of standard Fortran appeared to impede program portability, the very thing that standardization is supposed to enhance. In the case of two languages (separately standardized and hence officially unrelated), the question of which is the "real" Fortran appeared to be a further frustration to the user community. As a result of these factors, X3J3's carefully considered conclusion is that Fortran 8x should be a single language.

Add BIT String Data

Bit level processing has been discussed by X3J3 for many years. There are two major options for providing bit processing: bit strings and bit arrays. To date, and after considerable discussion, there has not been a consensus for a bit string facility. The current bit array facility in Appendix F was adopted on the basis of maximum compatibility with the array facilities, and, conversely, operations on bit arrays provide much bit-string type capability. Two additional factors appear to prevent complete consensus on any comprehensive bit processing facility in Fortran. First is that data representation at the bit level is inherently variable and hence corresponding software dependence is extremely nonportable. Second is that none of the bit proposals presented thus far have been able to define bit stream I/O satisfactorily, which seems to be required in order to support true bit data processing. For these reasons, X3J3 has not considered the investment in a bit string facility (which is more complex than a bit array facility) to be advisable.

60

Add an OVERLOAD Statement

An OVERLOAD statement is not necessary to disambiguate any instance of operator or procedure overloading. As proposed in this comment, any use of FORTRAN 77 overloaded intrinsic operators (e.g., "*", as it is overloaded for integer and complex multiplication) or generic intrinsic functions (e.g., INT) would require use of an overload statement. This would make Fortran 8x not upward compatible with FORTRAN 77, and thus conflict with an important principle of Fortran 8x. Requiring an OVERLOAD statement for program-defined overloading, but not for intrinsic overloading, would constitute an unnecessary language irregularity, one that X3J3 has not thought advisable.

RESPONSE TO RICHARD RAGAN BALLOT

All your suggested changes to the document to improve readability and consistency have been considered by X3J3 and either accepted or rejected.

02

RESPONSE TO JOHN REID BALLOT

All the editorial comments have been accepted or considered and rejected by the full committee. Your glossary which was included as a ballot commentary has been accepted for inclusion as an appendix to the draft.

Unisys has cast a NO vote after much deliberation because we feel that the aggregate addition of the "modern" programming language features has been a disservice to the language. We can not in good conscience vote YES just to let the public have a look at the proposed standard. A YES vote, far more importantly, would be a statement that we are satisfied with the proposed standard in its current form and with the language it describes. We are not.

We feel that the addition of the entire package of modern programming language features has warped the language nearly beyond recognition. It is our opinion that the committee has spent too much time in language design at the expense of standardizing current practice. We fear the proposed language will no longer serve the community for which it was originally designed. The promoters of the modern language additions, on the other hand, contend that the language already fails to serve that community and that if the committee does not make drastic changes to the language, users will abandon it in favor of a modern language such as Ada. Our reply is: Let them use Ada. To paraphrase Mark Twain, we believe the predicted demise of FORTRAN has been greatly exaggerated.

The hallmark of FORTRAN has been that it is a reasonably compact language that may be implemented via fairly small compilation/run-time systems, and that its generated code, particularly numeric code, runs extremely fast. We are concerned that the committee has not given sufficient consideration to the impact of the new 8x features on compilation speed, compiler size, and run-time execution speed. There were presentations and long discussions at the January, 1986, X3J3 meeting about the size of the language that eventually influenced the development of the compromise. However, the features that are the greatest impact on the compilation and execution environments are still in the language. Such features are as narrowly focused as the implementation of modules and as general to the language as the addition of a significant number of semantic rules.

Our language design/implementation concerns include, but are not limited to, the following:

- * The concept of modules and the USE statement introduce dependent compilation into the language. Dependent compilation makes it difficult, if not impossible, to implement a one-pass compiler. We believe that modules may be simply a case of overkill and that standardization of the current common practice of using an INCLUDE statement would have been satisfactory.
- * The 8x document describes a language that is quite complex. Other complex languages, like Ada for example, benefit from grammars that lend themselves to be parsed by some form of parser generator. Because of the context-sensitive and ambiguous syntactic constructs in 8x, parsing will be difficult. For example, 8x introduces the derived-type constructor which could have the same syntactic structure as a function reference or an array reference. The increased semantic checking to resolve this and other new constructs will place further demands on a compiler. Parsing is also made more complex by the addition of a second source form.
- * In many ways 8x describes a modern language that attempts to provide a virtual machine to the programmer. Yet in many other ways, it is still tied to the hardware (and probably rightfully so if one of the goals of this standard is to maintain upward compatibility with FORTRAN 77). 8x can therefore never provide a complete virtual machine model. Complex languages like 8x have a better chance of acceptance and usage as a modern language if they are based on a consistent formal model. It may be the absence of this model that accounts for critics stating that the language has no focus.
- * The readability (and therefore maintainability) of FORTRAN programs is degraded by some new features such as intrinsic

64

RESPONSE TO ROLISON/BOWE BALLOT

Your arguments that "modern" Fortran is a disservice to the user community was very persuasively presented. Your quotation from Mark Twain was very apt. The committee disagrees with you, however, and believes that the Fortran user community needs the modernizing features included in the draft. The power and memory size of very common computers even in the microprocessor field are easily capable of processing much larger compilers than were used only on the largest main frames five to ten years ago. This draft standard may be five years during processing and review, during which time even more computer resources will commonly be available for compilers the size of Fortran. The architecture presented suggests a modest obsolescent category while users learn that some features will not be available forever. We believe that the only way for Fortran to serve its users best is to move toward a modern language suited to today's scientific and engineering research problems. There are implementations currently available as extensions on a number of compilers for many of the features suggested.

You have been very responsive to the serious technical work and goals of X3J3. This is evidenced by your careful reading of the document with many very useful suggested changes. All these changes have been either accepted or considered and rejected by the full committee. Your principal technical issues are addressed in the following notes. With your help and the help of others on X3J3, we will have an excellent document to present to X3 for public review.

Concept of Modules and Complexity Issues

Modules. The MODULE/USE facility of Fortran 8x is far superior to INCLUDE for defining entities (such as global data) that are to be shared by different program units, a functionality that is very important to Fortran. Though implementations may choose to use a dependent compilation model for processing modules, modules may be processed as "filtered INCLUDE". That is, though not as simple to process as the textual INCLUDE facility found as extensions in most FORTRAN 77 implementations, in terms of a dependent compilation model, the MODULE/USE facility has neither greater nor less implications than does the typical INCLUDE.

Complexity. Fortran has always been a context-sensitive language. For example, array elements and function references have always (in FORTRAN 66 and FORTRAN 77) had the same syntax. Any language with overloaded operators and/or generic procedures is context-sensitive. The fact that Fortran 8x is context-sensitive, with the attendant parsing difficulties, is neither new to Fortran nor undesirable in current language technology. With advanced state-of-the-art compiling techniques, somewhat increased parsing complexity is a good price to pay for improved user functionality; a one-time investment in parsing will pay off millions of times in increased user productivity and improved software reliability.

Virtual Machine Model

The X3J3 committee considered the concerns expressed in the above ballot comment but felt no changes were required for the following reasons.

FORTRAN 66 was indeed a programming language rather tightly coupled to a particular view of hardware (called a virtual machine in the ballot comment). FORTRAN 77, because it was nearly upward compatible with FORTRAN 66, maintained many of these "machine-like" properties. Where Fortran 77 was incompatible with Fortran 66, there was a move in the direction of creating a more formal (machine-independent) model -- for example, the replacement of Hollerith data with a character string capability introduced a formal model for character handling. Fortran 8x, because it is completely upward compatible with FORTRAN 77 (except when free-form source is used), inherits the remaining parts of the virtual machine model that are machine dependent but has introduced a more formal model, particularly for such features as arrays and global data sharing via modules.

Thus, the "lack of focus" is a consequence of the evolutionary development of Fortran rather than a revolutionary change, say to a new language with a complete formal model. However, Fortran 8x can be considered to have a very definite model, one of coexistence with past versions of the language enhanced by facilities that promote portability and utility. The problem is that this model is not regular as would be ideal. Fortran 8x is thus neither a modern language nor an ossified one. The X3J3 committee is aware that the language required by upward compatibility has yielded a language that requires both a knowledge of modern programming concepts and an awareness of its history to fully understand the language and its model.

Array Valued Expressions and Descriptors

Array-valued Expressions. Array-valued expressions certainly require the use of temporary arrays. However, it is not obvious that the management problem is as difficult as you suggest. Methods to minimize the total number of temporaries are well known for the scalar case. The number of temporaries needed for a comparable array expression is the same except that each temporary is many words long rather than one. Effective reuse of larger temporaries is desirable if only because of their larger sizes.

However, even this is not expected to represent a significant problem. In most cases, the size of the temporary is determined, not by the total array size, but rather by the strip-mined length of a hardware vector (which is 1 in the scalar machine case). Such lengths vary, but are commonly on the order of 64-512 words long. Given that current memory sizes of processors that provide hardware vectors seem to be on the order of 32-128 megabytes, the amount of space used to provide storage for temporaries does not seem to be a problem.

Array Descriptors. You are correct that array descriptors will be used in most implementations of array argument interfaces and alias arrays. A descriptor is also

required for an allocatable array but it is simpler than that for array dummy arguments and alias arrays. In fact, an allocatable array descriptor needs exactly the same information that a FORTRAN 77 compiler had to manage for an adjustable array dummy argument of the same number of dimensions (the base address and the size of each dimension as a minimum). Thus, allocatable arrays should not impact compile or run-time performance (especially for those who do not use them).

Since dummy argument arrays and alias arrays permit non-contiguous array slices, some form of descriptor is required to provide this additional functionality. The new descriptors contain one more component per dimension than the comparable FORTRAN 77 adjustable array (namely the stride between non-contiguous array slices). From a performance standpoint this adds one more multiply per array dimension to the subscript computation formula. In the case of machines that provide only byte addressing the multiply already existed to get from bytes to words and so only the multiplier will change.

The proposed standard recognizes that Fortran performance must be retained. Therefore, the INTERFACE block provides a mechanism such that FORTRAN 77 style dummy argument arrays (fixed, adaptable, and assumed-size) can be implemented with no descriptor and hence with no performance degradation. In essence, the Fortran 8x user can decide to use the extended functionality and to pay a slightly higher cost for the more general capability.

REAL (*,*)

The X3J3 committee considered the concerns expressed in the above ballot comment but felt no changes were required for the following reasons.

The REAL(*,*) facility has been designed to permit the preparation of portable software using specified precision real and complex data types. If a programmer is not interested in portability, there are many alternative ways to write the program to avoid the costs (if any) of creating portable source code. These include the use of precision specifications using explicit integer constants, globally specified integer parameter values for the precision and exponent range, or even the use of default real and double precision types.

The cost of this portability can vary from negligible to serious and is entirely up to the implementation. The cost will depend upon how much effort is made and inventiveness is introduced by the implementor into the implementation to reduce the cost. For example, the cost of generation of "a few" copies of a large program may only appear in the code generation phase of the compiler for when the linker/loader resolves the procedure references, only one copy of the compiled procedure may be needed and therefore present in the load image. If more copies are present in such an implementation, they are required and the "surprise" is possibly a misconception of the properties of the program. Another example might be an implementation that specifies in the job control statements how many versions are to be supported.

Of course, an implementation that introduces no new compilation or linker/loader techniques for handling REAL(*,*) will behave in ways that are surprising and not very useful. The issue therefore is the tradeoff between the cost of an inventive implementation and the wider availability and portability of software. For the price of a more costly implementation, carefully written high quality software can be more readily transferred between machines. Since enhanced portability is one of the stated goals for Fortran 8x, the committee believes that the REAL(*,*) facility is an appropriate enhancement for Fortran 8x.

Keyword Argument Mapping

Keyword argument mapping is the result of three user requests/needs: order-independent actual argument lists, optional arguments, and improved procedure call readability. These requirements are all met with keyworded actual arguments. This feature is quite simple, and should therefore provide the user with the desired optional functionality without adding much complexity to the implementation of Fortran 8x.

Summary

Clearly, Fortran 8x implementations will be "larger than" FORTRAN 77 implementations, as FORTRAN 77 is a subset of Fortran 8x. Presumably such larger implementations will have somewhat lower compile-time performance. Run-time performance need not be lower, however, and performance improvements (over corresponding FORTRAN 77 provisions) can certainly be expected in things like array processing (or vector and parallel architectures) and internal procedures (if expanded in-line). To suggest that users do not want Fortran 8x is to ignore

- a. the results of numerous user surveys and
- b. the fact that Fortran 8x is the result of what may be the most open language standards process ever.

X3J3 has tried very hard to obtain widespread user input and develop a Fortran 8x that most satisfies the needs of Fortran programmers. Though top performance is a very important part of those needs, higher level language, added functionality and improved portability are increasingly important factors. Increasing volumes of application backlogs, increasing maintenance costs, and greater portability demands are widely affecting the Fortran user community and are not alleviated by the continuing dramatic advances in hardware capabilities. These hardware advances (together with improving compiler techniques) will help to make Fortran 8x compiler performance acceptable to the user community. It is the features of Fortran 8x, however, that will affect the users' abilities to develop applications more quickly, reduce maintenance costs, and obtain improved portability with acceptable performance. Evolution of Fortran, though not without costs, is the most cost effective way to achieve these benefits, not only for the user, but almost certainly for the vendor as well. Fortran 8x represents such an evolution, shaped primarily by Fortran user input and needs.

To: X3J3
From: Lawrie Schonfelder
Subject: Letter Ballot Comments
References: 88.103 February 1987

Introduction: This set of comments which is attached to an affirmative ballot is divided into two parts. The first is a set of trivial editorial corrections. These are concentrated in the sections for which I have had subgroup responsibility. The second part includes some more substantive commentary regarding areas where I think there are still a number of defects in either the functionality or its description. In each case correcting the defect would make a substantive change to the language which in my opinion would improve it. However none of these defects do I consider of sufficient importance to warrant rejecting the sending of the document for public comment at this stage.

RESPONSE TO LAWRIE SCHONFELDER BALLOT

All your items have been either accepted, or considered and rejected by X3J3. The substantive issues mentioned in your ballot have been considered, or have been rejected in the compromise plan developed after the first letter ballot.

Comments Accompanying The Letter Ballot on Document S8.103

Brian T. Smith
Brian T. Smith

I am very much in favor of having the document S8.103 available for public review as soon as possible. The following comments represent editorial improvements to the document and its exposition. It is my opinion that the technical content of this document, particularly after the enclosed comments are addressed, is sufficiently clear and unambiguous to be reviewed by the public as a draft proposed standard. Indeed, it is most appropriate at this time to have this draft reviewed by the users of Fortran before any further technical changes are made.

In reviewing this version of S8, its accompanying glossary (S11.102) and addenda, I recommend that the following corrections be made to improve the presentation and readability of the document. I found the glossary a considerable help in establishing the terminology required to read the document and recommend that it be included at least as an appendix to the draft document. As we all know, the index is incomplete but I recommend that we work to improve it for the public review.

RESPONSE TO BRIAN SMITH BALLOT

Your ballot included many editorial suggestions that will improve the readability and consistency of the document. The proposal on host association was approved by full committee for inclusion. All your comments were either accepted or considered and rejected.

RESPONSE TO RICHARD SWIFT ~~SECRET~~ BALLOT

Your editorial comments were either accepted or considered by the full committee and rejected.

IBM Corporation Santa Teresa Laboratory San Jose, Ca

Response to X3J3 Letter Ballot on Forwarding 8x to X3 for Further Processing. January 5, 1987

X3J3, attempting to extend Fortran with many new features while maintaining compatibility with Fortran-77, has produced a draft document describing a proposed 8x language. Language design is a major effort for any organization and is especially difficult for a standards organization. X3J3 is to be admired for its continued labor over the many years required for the difficult task of defining a large and complex programming language.

X3J3 now proposes that this language, 8x, be submitted to the X3 parent committee for further processing, with the goal of 8x eventually becoming a new standard, replacing Fortran-77.

The IBM response to this proposal is "no".

The difficulty with 8x lies not in the quality of the work that X3J3 has done (indeed we will here not make any comment on the quality of 8x as a language) but rather in the somewhat surprising observation that the result of X3J3's work is a new language and not a revision of Fortran.

Why isn't 8x a new Fortran? What is Fortran? Or better, "what are the attributes of Fortran that should remain constant across different versions of a Fortran standard"?

Each new 8x feature is, in some sense, "a good idea"; at some time each feature had sufficient support in X3J3 to be added to the draft. Notice, however, that the questions above, and the criteria we are about to develop, pertain to Fortran in total, rather than to individual features. The overriding concern, which we and every reviewer must address, is the total mass of the 8x language.

In reviewing the Fortran literature and in the experience of the many IBM'ers who produce, market, sell and use Fortran compilers, we have found that Fortran:

- * is a small language, used by full-time programmers as well as being suitable for occasional use by professionals who are not necessarily computer programmers. Fortran is described as a language where, the occasional user can write programs without the extensive use of a reference manual.
- * is efficient. Fortran statements have a straight-forward relationship to the machine. Fortran users understand what Fortran does and there is a simple run-time environment.

36.1 - 36.2 - 36.6

RESPONSE TO RICHARD WEAVER BALLOT

Your ballot contains two central issues. One is the size and complexity of Fortran as a language in competition with ADA. The other is your proposal for two languages. Your concerns relating to small language, efficiency, inexpensive implementations, current inventories of application code, widespread implementation, user training, and current compiler investments do not support your claim that two standard languages are advisable. Specific comments on these concerns follow.

Small Language

Fortran 8x is a major extension of FORTRAN 77. However, the major part of Fortran 8x is FORTRAN 77, which makes Fortran 8x, though larger and more complex than FORTRAN 77, in the same general size category as FORTRAN 77. Since all of FORTRAN 77 is retained in Fortran 8x, FORTRAN 77 users, or those using only a portion of FORTRAN 77, will be able to use Fortran 8x in exactly the same way without learning anything more.

The language comparisons on page 36.2 are not valid comparisons. For example, the FORTRAN 77 statement count (53) on pages 36.5 and 36.6, makes FORTRAN 77 appear to be ten percent larger than Ada. In a similar vein, the Fortran 8x syntax rules define a number of nonterminals that are not necessary to the syntax description. If these intermediate terms were eliminated, the 280 syntax rules would be roughly reduced by half, making Fortran 8x appear not only much smaller than Ada, but also smaller than Pascal. The number of intrinsic functions does not affect the size of the language (e.g., in terms of syntactic description); the number of intrinsic functions (or the size of any library) does affect the size of an "implementation", but not the size of the "language".

Fortran 8x is not an "immense language". Its extensions over FORTRAN 77 constitute approximately one-third of Fortran 8x, and great attention has been paid to making them as "Fortran-like" as possible. Fortran users will have more functionality available to them if they want to use it, but no Fortran programmer need change anything about his/her view of Fortran or how he/she uses it.

Efficiency

As FORTRAN 77 is a subset of Fortran 8x, there is no reason why the bulk of FORTRAN 77 optimizations cannot be carried forward in Fortran 8x implementations. The efficiency of the new extensions will be directly related to the vendors' efforts in their optimization.

Inexpensive Implementations

As Fortran 8x is a larger language than FORTRAN 77, implementations can be expected to be correspondingly more expensive. The MODULE/USE facility does not imply a dependent compilation model, however, at least not an Ada-like one. Though

implementations may choose to use a dependent compilation model for processing modules, modules may be processed as "filtered INCLUDE". That is, though not as simple to process as the textual INCLUDE facility found as extensions in most FORTRAN 77 implementations, in terms of a dependent compilation model, the MODULE/USE facility has neither greater nor less implications than does the typical INCLUDE. The impact on other operating system services depends upon the implementation strategy. There need be no impact (on the link editor, for example) if name management is placed in the compiler. A valid implementation decision could, of course, be to have the compiler pass some such processing on to other system services. These trade-offs will have to be assessed by the vendor for each implementation.

Inventory of Application Code

As all of FORTRAN 77 is a subset of Fortran 8x, all investment in FORTRAN 77 code is preserved in Fortran 8x. None of the features cited (COMMON, EQUIVALENCE, ENTRY) are obsolescent in Fortran 8x, and therefore could not possibly be considered for at least two more revisions of the standard beyond Fortran 8x -- after the year 2010 at the earliest. Fortran should therefore strengthen its position as the language of choice for scientific applications, while preserving all investment in FORTRAN 77 code, thereby providing considerable motivation for widespread implementation.

User Training

All of FORTRAN 77 is contained in Fortran 8x. All user investment in FORTRAN 77 training is therefore preserved, and no additional training is needed for effective use of Fortran 8x. In order to use the new features of Fortran 8x, only incremental training is needed, as the new enhancements are compatible extensions of FORTRAN 77. Similarly, only minor incremental training would be associated with the non-use of obsolescent features in application software; there are very few of these features (arithmetic if, assign and assigned go to statements, pause statement, alternate return, real and double precision DO variables), they are not widely used in FORTRAN 77 code, and the alternatives to their use are already generally well known in the Fortran user community.

Vendor Compiler Investments

A major advance of Fortran 8x over FORTRAN 77 is in the area of dynamic storage allocations. Whereas FORTRAN 77 was based upon a completely static storage model, several areas of Fortran 8x involve dynamic storage allocation. These include the allocate/free statements, adjustable local arrays, elimination of restrictions in character string processing, certain array operations, and recursive procedures. The need to accommodate dynamic storage allocation could have significant impact on existing FORTRAN 77 implementations. Compiler designs tied very tightly to the Fortran 77 static storage model may require significant redesign to be extended to Fortran 8x. Not all existing FORTRAN 77 compilers are drastically affected, however, as many already support recursion and other features involving dynamic storage

allocation. A major part of many FORTRAN 77 compilers is that part devoted to optimization. Most investment in FORTRAN 77 optimization could be preserved in Fortran 8x implementations.

Two Languages

X3J3 has considered at great length and in great depth the issue of a "partitioned" Fortran. One form of partitioning considered (during 1985-1986) is that of two separate languages. Earlier, another form, that of a "core + modules" architecture, was considered, adopted, and subsequently rejected. In both instances massive user preference for a single monolithic language persuaded the committee away from any form of partitioning. In both cases, multiple versions of standard Fortran appeared to impede program portability, the very thing that standardization is supposed to enhance. In the case of two languages (separately standardized and hence officially unrelated), the question of which is the "real" Fortran appeared to be a further frustration to the user community. As a result of these factors X3J3's carefully considered conclusion is that Fortran 8x should be a single language.

WG5 BALLOTS

TO: ISO/TC97/SC22/WG5 - Fortran
FROM: J. T. Martin
SUBJECT: Informal Ballot Results on X3J3/S8.103 (February, 1987)

An informal ballot on the adoption of X3J3/S8.103 (February 1987) as the draft proposed new Fortran standard was sent to WG5 members. The ballot period extended through January 12, 1987. Twenty-four ballots were received prior to the deadline. The vote by individuals was:

Yes 22.25
No 2

Four country ballots were received. The vote by countries was:

Yes 4
No 0

The next page is a combined Table of Contents and ballot tally.

CONTENTS

URSTIN

Cellarozzo

Louauer

	Page	Vote
← Adams, Jeanne	3	Yes <i>yes</i>
Buckley, Albert	4	Yes
Delves, L. M.	5	Yes
Du Croz, J. J.	6	Yes
Hammarling, S. J.	8	Yes
Kan, Tadayoshi	12	Yes
Kneis, Wilfried	17	No
Luttermann, Hermann	18	Yes
Marshall, Neldon H.	19	Yes
Martin, Jeanne T.	21	Yes
Meek, Brian L.	31	No
← Meier, Bruno	33	Yes
← Munchhausen, Meinolf	34	Yes <i>yes</i>
Muxworthy, David T.	35	Yes
Plassner, Klaus	43	Yes
Pollicini, Aurelio	44	Yes
Rotthausen, K.	45	Yes
Schmitt, Gerhard	46	Yes
← Shen, M. K.	59	Yes <i>yes</i>
Vallance, David M.	61	Yes
Wagener, Jerrold L.	66	Yes
Warren, Graham	67	Yes
Wilson, John D.	72	Yes
Wu, Qing-bao	75	Yes
Austria	76	Yes
Canada	77	Yes
Germany	78	Yes
United Kingdom	79	Yes

Japanese Comments

We approve to process S8.103 for submission to SC22 provided that the following proposals should be considered.

(1) Extension of the intrinsic character type

that necessary

(see Attachment A) :

to insert [(KIND=n)] between the keyword CHARACTER and [length selector] in the syntax R502,

(2) Additional intrinsic functions (see Attachment B) :

5

to add the important functions shown in the Attachment B to the intrinsic functions of S8.103.

We strongly propose either the above (1) and (2) are included in the standard or in Appendix F of S8.103.

Then, we would like to prepare the detail works caused by adopting our proposal, if necessary.

RESPONSE TO TADAYOSHI KAN BALLOT

Your ballot contained requests for two extensions to the draft proposal. The first was your proposal to extend the CHARACTER data type. It was not adopted for the reasons given below. The second request, for additional intrinsic functions, was rejected as it would increase the size of the language. The first letter ballot failed in X3J3 because many X3J3 members felt the language was too large. Prior to the second ballot, a number of features were removed from the language. They are described in Appendix F for reference purposes only. It was further agreed that new features would not be added before the document is forwarded as a draft proposed standard.

Extensions to CHARACTER Data Type for "multi-byte" Characters

1. **Functionality Available.** The desired functionality is already available in the FORTRAN 77 CHARACTER data type. FORTRAN 77 does not specify the size of a character storage unit. Though most implementations use one byte, the standard equally allows, for example, two byte character storage units. It does not allow both one byte and two byte character storage units in the same implementation, however. Though this capability is provided by the proposed extension, it does not seem to be either essential or desirable for the functionality. Multiple character sets can be incorporated into an implementation of the FORTRAN 77 CHARACTER data type, and distinguished via the ICHAR intrinsic function. For example, an implementation using two byte character storage units could accommodate both ASCII and Kangi, where ICHAR returns values 0-127 for ASCII and greater than 256 for Kangi.

2. **New Data Types Desirable.** If a new and different data type is desired, Fortran 8x provides the capability to define it and make it available to the programmer. The relevant facilities for this are the derived data type, operator definition and overloading, and module encapsulation. For example, a new data type KANGI could be defined that was two character storage units and the CHAR and ICHAR functions, etc. If encapsulated in a module called KANGI and placed in a public library, any programmer would have access to the KANGI data type simply by including USE KANGI in all routines requiring it.

3. **Storage Unit De-emphasis.** X3J3 believes that the concept of storage units should ultimately be removed from the Fortran standard. The proposed extension to CHARACTER utilizes the storage unit concept, and hence is somewhat in conflict with this X3J3 objective.

RESPONSE TO WILFRIED KNEIS BALLOT

X3J3 notes that the reasons for your negative ballot are that exception handling and bit data type are missing. Both of these features were in the language at the time of the first letter ballot; however, that ballot failed in X3J3 because many X3J3 members felt the language was too large. Prior to the second ballot, several features including these two were removed from the language. They are described in Appendix F for reference purposes only. With these features removed, the second ballot was successful. More specific reasons for not reinstating these features are as follows.

Exception Handling

The CONDITION/ENABLE facility of Fortran 8x has been one of the most controversial features of the draft standard. Among the reasons for this controversy are:

- a. concern about the possible effects on optimization,
- b. questions concerning propagation of conditions up the call chain in search of a handler, and
- c. the infeasibility of sensible resumption of execution in the case of some conditions.

The ENABLE block, which syntactically resembles a control construct, is intended to limit the impact on optimization to programmer-specified portions of the code; programmers must explicitly so identify each portion of code in which exception handling is activated. There seems to be a considerable amount of "discomfort" in X3J3 about the provisions in this exception handling facility for propagating conditions back up the call tree in search of a handler; the implications for procedure interfaces may not be entirely clear. Finally, there is considerable controversy over the "granularity" of resuming execution after handling a condition; the current facility has statement level granularity, with execution resuming at the next statement.

BIT Data Type

Bit level processing has been discussed by X3J3 for many years. The current bit array facility in Appendix F was adopted, rather than a bit string facility. On the basis of maximum compatibility with the array facilities, and, conversely, operations on bit arrays, provide much bit string type capability. Nevertheless, there remains significant preference for a bit string, rather than a bit array, facility. Two additional factors appear to prevent complete consensus on any comprehensive bit processing facility in Fortran. First is that data representation at the bit level is inherently variable among implementations and hence corresponding software dependence is extremely nonportable. Second is that none of the bit proposals presented thus far have been able to satisfactorily define bit stream I/O, which seems to be required in order to support true bit data processing.

I vote NO for two reasons, either of which would have been sufficient in itself. If both are met I shall be happy for it to be issued as a draft for public comment though there are many deficiencies (e.g. its inconsistency of presentation and its looseness of wording) which would subsequently need to be corrected before it could become a DIS.

3*

Reasons for NO vote:

1. Failure to include the enhanced "Conformance of processors" section (including detection and reporting of extensions, and documentation) agreed at the Halifax WG5 meeting. If there are any difficulties in phrasing, let these be allowed to come out at public comment stage, along with everything else. What deficiencies there are seem to me to stem from the vagueness of much of the rest of this section (e.g. "forms and relationships"), not from the Halifax wording.

3*

2. Failure to reinstate Appendix F into the main body of the standard. I am willing to accept two classes of entry: "to be included unless strong reasoned argument is received against" and "to be excluded unless strong reasoned argument is received in favour." Not this.

5

P32

Brian Meek

RESPONSE TO BRIAN MEEK BALLOT

Of the two reasons that you cited in your negative response (processor conformance requirement, reinstatement of Appendix F), one (processor conformance) has been resolved by action at the February 1987 X3J3 meeting. However, a substantial part of Appendix F has not been reinstated. It is the intent of X3J3 that Appendix F constitute precisely those features, in the words of the text of this reason, "to be excluded unless strong reasoned argument is received in favor". More specific comments on these two reasons are as follows:

Processor Conformance. The wording for this requirement was adopted at X3J3 at its February meeting in Los Angeles. It is essentially the wording proposed in Miles Ellis' ballot.

Reinstatement of Appendix F. The WG5 Halifax resolutions specified reinstatement of several specific Appendix F features. Most of these (e.g., structure constructors, overloaded operators) have been reinstated. The WG5 Halifax resolutions did not request reinstatement of a large majority of the Appendix F. features. The current status of Fortran 8x is in very substantial conformance with the Halifax resolutions.

RESPONSE TO BRUNO MEIER BALLOT

Your ballot comments contain two suggestions for additions to S8.103: pointers and syntax charts.

You suggest the addition of pointers, at least in Appendix F. The first ballot failed in X3J3 primarily because many X3J3 members felt the proposed language was too large and complex. Prior to the second ballot, several features were removed from the language. One of the considerations used in selecting features to be removed was the necessity for promoting the efficient execution of Fortran programs. Thus features that might interfere with optimization were prime targets for removal. One of these was exception handling; another was the identification of arrays on multiple hosts. The latter feature provided the functionality of pointers for arrays. Since it was considered as merely a simplification of the IDENTIFY statement, it was not described as a feature in Appendix F. A similar functionality for scalar entities has never been described by X3J3, and would seem unlikely to gain favor at this time.

You suggest the addition of an Appendix containing syntax charts similar to Appendix F of ANSI X3.9-1978. When X3J3 adopted the form for the syntax rules used throughout the document to describe the language, it was felt that syntax charts would not be needed. The syntax rules define the language more precisely than the word descriptions used in ANSI X3.9-1978 and can be included in the body of the proposed standard, whereas the syntax charts were relegated to an Appendix. However, the syntax rules are extracted from the text of S8 to form a concise description of the language in Appendix D.

General Editorial Comments.

The document is found difficult to read by those new to it. It would benefit from editing:

- to have a short introductory paragraph at the start of each section
- to cut out unnecessary sentences and such tutorial matter which has no place in a standard
- to eliminate text which is duplicated in different sections
- to have a short introduction before a long set of syntax rules
- to give section numbers in the case of important forward references
- to reduce verbosity, thus exposing the meaning of the text.

I support the use of "Fortran" as a proper noun, not as an acronym.

Detailed Editorial Comments.

These are arranged in page order, not order of importance.

- 1A. p 1-1 l 7. This conflicts with p i l 6-7. Replace "is standard conforming" by "is intended to be standard conforming".
- 1B. In accordance with Halifax WGS resolution 10, and subsequent discussion in X3J3, make the following changes:
 - p 1-1 l 34. Add "generally" before "refer". This was in the corresponding text in ANS X3.9-1978 (p 1-2 l 9).
 - p 1-1 l 37. Add "usually" after "must". This was in the corresponding text in ANS X3.9-1978 (p 1-2 l 14).
 - p 1-2 l 5-6. Delete the sentence contained in these lines and replace by:

"A processor conforms to this standard if:

 - (a) it executes standard-conforming programs in a manner that fulfills the interpretations prescribed herein;
 - (b) it contains a capability to detect and report the use within a submitted program of permitted additional forms of Fortran statements and constructs (see below), or of forms designated herein as obsolete, obsolescent or deprecated;
 - (c) it is accompanied by documentation which defines the additional forms

RESPONSE TO DAVID MUXWORTHY BALLOT

Your detailed editorial comments were very helpful to X3J3 in refining S8. They have been considered along with many others and accepted or rejected.

You made some good suggestions for improving the readability of the document. X3J3 agrees that the document should read well. However, in making it precise, certain sections become verbose. Standards documents are a legal and technical specification. They should not be descriptive or tutorial, but used primarily for reference. Redundancy serves the purpose of providing information in all relevant parts of the document. X3J3 feels that redundancy can contribute to understanding and actually make the document more readable.

23753
L300

SECRET

JAN 21 3 03 AM '87

LLNL LVMR

LLNL LVMR

AFNOR Z 611974F

ZCZC

JFA TX 814 87.01.21

ATT : MRS JEANNE MARTIN ~~XXXXXXXXXX~~

I APOLOGISE FOR THE DELAY OF AFNOR ANSWER RELATING TO DP FORTRAN-EX FRENCH VOTE IS AS FOLLOWING

YES WITH COMMENTS

1 - TO CHECK THAT THE IMPLEMENTATIONS COULD AVOID CONFLICTS BETWEEN NEW AND DEPRECATED FEATURES

2 - THE FORTRAN AFNOR GROUP IS SURPRISED THAT X3J3 HAS NOT TAKEN INTO ACCOUNT SOME OF THE RESOLUTIONS (NOTABLY POINTERS-SIGNIFICANT BLANKS) VOTED WITH A VERY LARGE MAJORITY AT THE HALIFAX MEETING.

RELATING TO FF WG 5 PARIS MEETING, I WILL COMMUNICATE YOU THE FINAL DATE AFTER DISCUSSION WITH THE AFNOR FORTRAN GROUP.

SINCERELY YOURS AND BETTER LATER THAN NEVER - BEST WISHES FOR 1987

BOURSTIN

AFNOR

LLNL LVMR

AFNOR Z 611974F

TO REPLY FROM TELEX I OR II (TWX) DIAL 100 FROM EASYLINK USE /WUW.
EST 1127 JAN/21/1987

RESPONSE TO BOURSTIN BALLOT

The request for pointers has appeared in a number of ballots, including yours. The resolution of comments from the first ballot of X3J3 was reached in order to decrease the size of the language. The committee feels that further work on new features in the draft revision especially those that were not included based on the first ballot results would slow down the progress toward a new Fortran that the user community needs as soon as possible.

RESPONSE TO GERHARD SCHMITT BALLOT

Your detailed editorial comments were very helpful to X3J3 in refining S8. They have been considered along with many others and accepted or rejected.

You say you would like to see more deprecated features, including those that rely on storage association. As you know, the first S8 ballot failed in X3J3. One of the changes prior to the second ballot was to introduce obsolescent features that appear in S8 in a small type font. These are features that can be omitted from current Fortran programs. The deprecated features still exist, but they are not differentiated in S8 by a different type font. They are listed in Appendix B which also describes which new features can be used instead of the deprecated ones. In a subsequent revision of the language, some of the deprecated features could become obsolescent. Thus a path still exists for the migration of features out of the language, but it will depend on whether the features actually fall into widespread disuse.

Another change prior to the second ballot was the removal of several features from the language in order to simplify and reduce the size of the language. One of the removed features was exception handling. With these features removed, the second ballot was successful.

RESPONSE TO SCHONAUER BALLOT

Your ballot comments indicate you regret the removal of vector-valued subscripts from the proposed language. The first letter ballot failed in X3J3 because several members felt the language was too large and complex. Prior to the second ballot, the descriptions of some features (including vector-valued subscripts) were moved to an Appendix that is not part of the proposed standard. It was felt that vector-valued subscripts contributed to the complexity and irregularity of the language. Note that in the Appendix F description, there is a restriction that the left-hand side of an assignment statement must not include an array element more than once. This restriction was required to promote portability of Fortran programs since it was not in the best interests of optimization to standardize the order of array assignment. Without this restriction, efficient execution would interfere with portability, and portability is the main purpose of standardization. The rule does introduce an irregularity since there is no similar restriction for the right-hand side of an assignment statement. With such features removed, the second ballot succeeded.

7D

Comments accompanying the informal letter ballot

For an average reader of the document in the forthcoming public review period the first sentence of Appendix F is unlikely to be convincing/sufficient. In order to reduce the language size effectively, one would argue, the whole of array operations and the corresponding intrinsic procedures should have belonged to Appendix F.

Since it appears questionable whether the majority of those WG5 members who have not attended X3J3 meetings possess sufficient information on the genuine arguments supporting the exclusion of the features presently in Appendix F from the standard proper, I suggest that WG5 request X3J3 to supply, for reference purposes, an abstract of concrete and convincing specific/non-specific arguments that have been successfully put forward in this connexion.

During the public review period WG5 members are quite likely to be questioned by interested persons of the international community concerning that matter. It would be very embarrassing to have to admit one's ignorance.

Munich, 30th December 1986

W. Pitts Shen
" letter
certification for
Appendix F

M. K. Shen

M. K. Shen

RESPONSE TO J K SHEN BALLOT

Rationale for Appendix F

Your ballot requests a brief rationale for the contents of Appendix F. The following summary provides some of the reasons for creating Appendix F and including it in this first draft copy. X3J3 hopes that this will be helpful to you.

In March of 1986 a letter ballot was conducted relative to forwarding the Fortran 8x draft to X3 for public review. This ballot resulted in a 16-20 negative vote and failed resoundingly to achieve the necessary two-thirds needed to release the draft to X3 for public review. The ballot comments represented the best available description of a substantial disagreement with the contents of the draft at that time.

The predominant comment was that Fortran 8x was too large, and that some of the new features should be removed to make it smaller. All such comments were accompanied by a list of those features recommended for removal. Rather than formally vote feature by feature on these recommendations, an attempt was made to incorporate those features most requested for removal into an overall "removal package", one that left the language not only smaller but also coherent as a whole. This was successfully accomplished, with the resulting reduced language receiving wide support in X3J3.

The features removed by this process, and placed in Appendix F, were, almost without exception, those most often cited by the comments. It is difficult to accurately summarize the points in the comments calling for these adjustments. The reasons for removing some of the items are in the following list.

- **Remove CONDITION/ENABLE.** There was concern about the possible effects on optimization; there were questions concerning propagation of conditions up the call chain in search of a handler; the infeasibility of sensible resumption of execution in the case of some conditions that might be flagged.
- **Remove BIT data type.** Bit strings preferable bit arrays in the opinion of some users. Bit processing is inherently nonportable. There was no bit stream I/O.
- **Simplify internal procedures.** X3J3 decided to disallow ENTRY statements in internal procedures and disallow passing internal procedures as arguments. Nesting internal procedures was also disallowed.
- **Simplify MODULE/USE.** USE statements were restricted to using modules. The ALL EXCEPT use option was removed. Module procedures were made like external procedures rather than like internal procedures.

- **Remove some array features.** Vector-valued subscripts were removed because of the many-to-one problem. The FORALL statement (redundant with vectorized DO loops) was removed. Several fairly specialized array intrinsic functions were removed.
- **Remove variant structures.** The importance of "equivalence functionality" of variant structures was reduced considerably with the removal of the EQUIVALENCE statement from obsolescent status. There was continuing disagreement over how best to handle variant components. With the advent of much larger memories, even on microcomputers, the space savings of variant components is of much less importance.
- **Make blanks significant.** The significance of blanks was not used to make the Fortran 8x syntax simpler; significant blanks places additional restrictions on the programmer (without concomitant benefit in the form of simpler syntax).

Although these points, and others, may have been cited by the commentators, there is the relatively high frequency of ballot comments that advised X3J3 to reduce the size of the draft Fortran.

Comments on S8 v103 from the Canadian Fortran Group

Technical Comments

1. We support the reduction in the size of the language which has been effected by the compromise solution (by simplifying some of the features or moving parts of the language to the "Removed Extensions" appendix), in particular:
 - removal of vector-valued subscripts
 - simplified REAL(*,*)
 - simplified IDENTIFY
 - simplified internal procedures
 - reduced number of array intrinsic functions
2. One removed feature which we would prefer to see reinstated in the language is the significant blank in free-form source. We feel that if this feature is ever to be introduced into Fortran, then now must be the time (Halifax WGS resolution #14).
3. We would like to see a pointer facility in the proposed standard (Halifax WGS resolution #11).

Editorial Comments

- | | |
|-----------------------|--|
| page ii, line 45 | Change "to be removed" to "to be removed from future standards". |
| page iii, line 31 | Change "(user defined)" to "(user-defined)". |
| page iv, lines 15-16 | Change "describes the approximately one hundred intrinsic functions and two intrinsic subroutines" to "describes the 93 intrinsic functions and four intrinsic subroutines". |
| page v, line 3 | Change "Section 14 (Entity Scope, Association, and Definition)" to "Section 14 (Scope, Association, and Definition)". |
| page v, line 12 | Change "since that time" to "since April 1978". |
| page 1-3, lines 38,41 | Change "-stmt" to "-stmt". |
| page 1-5 | Why is the nature of deleted and obsolescent features defined in Chapter 1, yet the nature of deprecated features defined in Appendix B? It would be better to have all 3 definitions together in one place. |
| page 1-5, line 11-12 | The term "core conforming" is used without being previously defined. If it is not to be previously defined (in Section 1.4 "Conformance") there should at least be a reference here to the section of the document where it is defined (Appendix A, though even there the term is not explicitly defined). |
| page 2-8, line 28 | Change "a rank one array" to "a rank-one array". |



RESPONSE TO CANADIAN FORTRAN GROUP

We appreciate your support for the compromise proposal developed after the first X3J3 ballot. Your editorial comments were considered by X3J3 and either accepted or rejected. The following notes address items 2 and 3 of your ballot.

Item 2 Significant Blanks

The issue of blank significance has been debated in the committee for many years. At times they were voted in, and then later voted out. During this time, the use of blank significance in the syntax was ruled out, since no one seemed to feel certain what the final disposition would be. After discussion, especially with the members who voted "no" the first time, it appeared that this was one issue in the draft that we would have to give way on. It seems that this is something that long-term users of Fortran like and want to retain.

Item 3 Pointers

The committee studied the need for pointers and reviewed several proposals from members during the past several years. There was no consensus on the best approach for this facility. There were a number of proposals that provided a pointer facility for arrays. But when the need for a pointer facility for data structures as well as arrays was determined, there was no simple and straightforward way to introduce this facility into the language.

5

My "yes" vote indicates I am in favour of submitting the Fortran 8X document for public comment. I do not think the draft standard, nor especially the document itself, is yet suitable as a new Fortran standard. However it is high time the general computing public was given the opportunity to give detailed appraisal of the proposal: any further delay is likely to cause the proposed standard to lose credibility. In my experience of presenting Fortran 8X to existing Fortran users, there was a fair amount of support for the draft proposals as outlined at the WG5 meeting in Geneva in April 1984, but there is growing impatience over the delay in completing the drafting process.

I shall postpone detailed and technical comments until the public review: the following are mainly general observations and impressions.

1. The document is a considerable improvement over earlier versions but still appears lacking in structure and consistency of style. Terms are not defined in a logical sequence but appear randomly throughout the text. There are still too many forward references and definitions which appear in more than one place. Every term should be defined completely and only once. The general text is too verbose, tautological and repetitive: it should be concise and precise. There is a logical structure to the document as a whole, but there is no consistent structure within a chapter.

2. The examples within the text are helpful although in some cases more explanation is needed (eg lines 30 - 40 on page 4-7). The Section Notes should be read with the text, which involves continual flipping back and forth while reading the document. I suggest the text and syntax definitions be printed on odd numbered pages only with the facing even numbered pages used for the Section Notes and examples.

3. I find the form of BNF used for the Syntax Rules extremely tedious and difficult to follow. A more diagrammatic form is needed: what was wrong with the "railroad" diagrams of FORTRAN 77 which proved very popular with users? I understand there were some discrepancies with the text but that is no reason to abandon it. Also, surely the word "help" in line 32 of page 1-2 should be deleted?

4. Appendix F should be deleted totally. I have changed my mind since the Halifax meeting; I now feel it only confuses the reader and adds unnecessary bulk.

5. I take issue with the second sentence of section 1.1 page 1-1. I do not feel this standard does anything to improve portability or reliability of Fortran programs! The statement on conformance (section 1.4) adds nothing to that in the Fortran 77 document. As a minimum, it is essential to have a much stronger statement on conformance along the lines of Halifax Resolution 10. Compiler writers must not be allowed to introduce extensions without at least giving the user the means to detect their use.

6. There seems to be an unnecessarily large number of similar terms used in Chapter 2: Program Unit, Subprogram, Procedure, some can be internal and external, some can't - all very confusing! Do we really need all these different terms, and even if we do how are they inter-related? A diagram is needed. From reading page 2-4 it is not easy to see what the difference is between a Procedure and a Subprogram: it is only upon reaching Chapter 12 that Procedures are explained more fully.

7. I don't like the term "module". It is frequently used elsewhere for other purposes - eg a group of subprograms. Also "modularity" is something different. The term "module" does not convey the sense of globality. I suggest changing the word used in this context to "globule".

8. A few minor errors:

- line 21 page 1-4 replace "protects" by "attempts to protect".
- lines 11-13 page 1-5 "core conforming" appears twice, once with a hyphen and once without. What is meant by "core"?
- line 1 page 2-8 replace "must" by "does".
- line 18 page 2-8. It is bad practice to use definitions like "A scalar is a datum that is not an array" especially when "array" has not yet been defined!
- line 40 page 4-4 gives 2.1 as an example of a signed constant. This is true according to rule R404 but why do we need two rules, R404 and R405, when the only difference is the presence of a sign which is optional anyway!
- line 25 page 5-2 replace "4.3.1.1" by "4.4.1.1"

J.D. Wilson
December 17, 1986

RESPONSE TO J D WILSON BALLOT

It is time that the Fortran draft revision to X3.9-1978 is submitted to X3 and SC22 for further processing. The draft that resulted from after the first ballot is a complete and well thought out draft. It is sometimes difficult to bring the national and international Fortran experts into agreement on some of the issues, as we discovered at the Halifax meeting in August of 1986. There was considerable support at Working Group 5 that all of Appendix F should have remained in the draft standard. However, it is important that due process is carried out in producing a standard and that an effort to achieve consensus is made. A careful effort has been made in the past year to do this. X3J3 has depended on the interest and support shown by the international community throughout the development of this standard.

You have seven observations and some minor editorial corrections. The corrections have been considered by full committee and either accepted or rejected. Comments on your seven points follow.

1. **Logical Structure of Sections.** The logical structure of the chapters emphasizes data types and the use of data in assignments, followed by executable constructs in Fortran, and I/O explained for various data objects. You may feel that it is not the best arrangement to describe program units at the end of the standard, rather than at the beginning when global concepts are under discussion. The draft standard is not a tutorial, but a specification, and the reader, in a sense, needs to know everything at once. This makes it necessary to have redundant definitions. The arrangement points out, we felt, the importance of data objects and their manipulation in today's Fortran.
2. **Examples and Section Notes.** There were a number of editorial comments that improve the examples in the text. The Section Notes are not allowed in the text of the standard according to the "ANSI Style Manual".
3. **Syntax Rules (BNF).** The railroad syntax did not define the language well enough. It was not possible to introduce important constraints and restrictions. The BNF follows commonly used rules for a more precise definition. The word "help" on page 1-2 is important because the text of the standard always takes precedence over the syntax rules if there is a discrepancy.
4. **Appendix F.** It is true that many people are ambivalent about including Appendix F. It is placed in the draft for review only. It will be removed from the final standard.
5. **Conformance.** At the 103rd meeting, we passed a stronger statement on conformance introduced by your colleague, Miles Ellis. This was in response to the resolutions at Halifax.

6. **Section 2 Terminology.** Program units are explained in detail in Sections 11 and 12. Again, I suggest that it is necessary to know everything at once in order to read a specification. Initial definitions for program units are necessary early in the text to explain the global concepts required to understand a Fortran program.
7. **Module.** The committee spent many meetings debating this term. Initially, it was called a "bundle" with the assumption that at some future time the term would be changed. It is different from an Ada package, and "package" would not be appropriate. Since we had introduced the concept of "core + modules" and rejected the initial definitions, we finally agreed to use the term "module".

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

X3/87/03-053-X, S
Project 507-D

NEWS RELEASE

104(*) JCA-4

For more information contact:

Eli Szklanka, X3J12 Chair
617-964-3890

Date: March 5, 1987

*** * * NEW DIBOL STANDARD EXPECTED TO STABILIZE CURRENT IMPLEMENTATION * * ***

**X3 ANNOUNCES PUBLIC REVIEW AND COMMENT PERIOD ON
DRAFT AMERICAN NATIONAL STANDARD X3.165-198x,
PROGRAMMING LANGUAGE DIBOL**

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed American National Standard, X3.165-198x. The public review period extends from April 10, 1987 to August 10, 1987.

The DIBOL programming language is a general purpose business programming language that is well suited for writing financial applications for small business computers. It was originally introduced 14 years ago and underwent major feature enhancements in 1975 and 1983.

Due to the expanding use and the variety of implementations, the need has arisen for a clearly defined standard. The current implementations are similar, but incompatible. The purpose of this standard is to provide an unambiguous, machine and operating system independent definition of the DIBOL language.

This draft standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$ 30.00

#

**Operating under the procedures of The American National Standards Institute.*

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

X3/87/03-052-X, S
Project 405-D

NEWS RELEASE

104 (*) JCA -
For more information contact:

Richard Steinbrenner, X3B5 Chair
201-386-7053

Date: March 5, 1987

*** * * NEW STANDARD TO SATISFY INDUSTRY NEED * * ***

X3 ANNOUNCES PUBLIC REVIEW AND COMMENT PERIOD ON
DRAFT AMERICAN NATIONAL STANDARD X3.164-198X,
UNRECORDED MAGNETIC TAPE CASSETTE FOR INFORMATION
INTERCHANGE, 3.81 MM (0.150 IN) TAPE, 394 FTPMM
(10,000 FTPI) "HIGH DENSITY CASSETTE"

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed American National Standard, X3.164-198x. The public review period extends from April 10, 1987 to August 10, 1987.

This standard for an unrecorded cassette containing 3.81 mm (0.150 in) wide magnetic tape presents the minimum requirements for the mechanical and magnetic interchangeability of the cassette between information processing systems, using the physical recording density of 394 ftpmm (10 000 ftpi).

The cassette is of the twin hub coplanar type, loaded with 3.81 mm (0.150 in) wide magnetic tape. Access holes are provided for, and the tape is transported between hubs by external reel motors. Features are provided for the use of external tape capstan(s), tape position sensing and external tape guiding.

This standard applies to the cassettes used for data interchange. Where it applies to cassette testing only, it is specifically stated.

This draft standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$20.00

*Operating under the procedures of The American National Standards Institute.

#

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

X3/87/03-051-X, S
Project 356-D

NEWS RELEASE

104(*) JCA-4

For more information contact:

Bill Carlson, X3B7 Chair
408-496-0322

Date: March 5, 1987

*** * * NEW STANDARD TO PERMIT THE DESIGN & USE OF ADVANCED DISK DRIVES * * ***

**X3 ANNOUNCES PUBLIC REVIEW AND COMMENT PERIOD ON
DRAFT AMERICAN NATIONAL STANDARD X3.163-198X,
CONTACT START/STOP METALLIC THIN FILM
STORAGE DISK**

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed American National Standard, X3.163-198x. The public review period extends from April 10, 1987 to August 10, 1987.

The goal of this standard is to establish interchange standards and set mechanical and magnetic criterion permitting the design and use of advanced disk drives. The standard is similar in nature to the current ANSI standards for rigid media, but expands some current restrictions and addresses parameters ignored in those documents.

This standard provides the mechanical, physical and magnetic properties of a low friction 130 mm (5.118 inch) diameter magnetic disk, with a typical recording density of 38 tracks/mm (960 TPI) and 402 flux transitions per mm (10200 flux transitions per inch) at 33,02 mm (1.300) inch radius.

This draft standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$25.00

#

**Operating under the procedures of The American National Standards Institute.*

X3 Secretariat: Computer and Business Equipment Manufacturers Association
311 First Street, N.W., Suite 500, Washington, DC 20001-2178

Tel: 202/737-8888
Fax: 202/638-4922

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

NEWS RELEASE

104 (*) JCA -11

For more information contact:
William Kenworthy, X3L8 Chair
202-694-3361

Date: March 5, 1987

*** NO LIMIT TO LOCATIONS ***

**X3 ANNOUNCES PUBLIC REVIEW AND COMMENT PERIOD ON
DRAFT REVISED AMERICAN NATIONAL STANDARD X3.47-198X,
STRUCTURE AND DATA REQUIREMENTS FOR THE IDENTIFICATION OF
NAMED POPULATED PLACES, PRIMARY COUNTY DIVISIONS,
AND OTHER LOCATIONAL ENTITIES OF THE U.S.**

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed revised American National Standard, X3.47-198x. The public review period extends from April 10, 1987 to August 10, 1987.

This standard establishes a structure for the assignment of identifying data codes to locational entities in the United States and its outlying areas, for the purpose of information interchange among data processing systems. Types of locational entities for which this structure is intended include (but are not limited to) populated places, primary county divisions such as townships and New England towns, Indian reservations, and facilities such as airports and military bases. This standard also establishes requirements for associated data that improve the utility of the primary data codes.

This draft revised standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$10.00

#

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

NEWS RELEASE

104 (*) JCA-4

For more information contact:

William Kenworthy, X3L8 Chair
202-694-3361

Date: March 5, 1987

*** * * X3 ANNOUNCES A REVISED STANDARD * * *
THAT MAKES A BROADER STATEMENT**

DRAFT REVISED AMERICAN NATIONAL STANDARD X3.38-198X,
IDENTIFICATION OF THE STATES, THE DISTRICT OF COLUMBIA,
AND THE OUTLYING AREAS OF THE UNITED STATES
FOR INFORMATION INTERCHANGE

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed revised American National Standard, X3.38-198x. The public review period extends from April 10, 1987 to August 10, 1987.

The current American National Standard provides codes for the fifty states of the United States and the District of Columbia. Only four of the outlying areas are included in the standard. It does not include codes for the remaining outlying areas. These include Northern Mariana Islands, Trust Territory of the Pacific Islands, and nine minor islands. The revised standard is intended to specify codes for States, the District of Columbia, and all U.S. outlying territories.

This draft revised standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$

#

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

NEWS RELEASE

104 (X) JCA - C

For more information contact:

William Kenworthy, X3L8 Chair
202-694-3361

Date: March 5, 1987

*** * * CAN YOU IDENTIFY YOUR COUNTY? * * ***

**X3 ANNOUNCES PUBLIC REVIEW AND COMMENT PERIOD ON
DRAFT REVISED AMERICAN NATIONAL STANDARD X3.31-198X,
STRUCTURE FOR THE IDENTIFICATION OF THE COUNTIES OF THE U.S.
FOR INFORMATION INTERCHANGE**

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed revised American National Standard, X3.31-198x. The public review period extends from April 10, 1987 to August 10, 1987.

This standard establishes a structure for the assignment of identifying data codes to counties and county equivalents of the United States and its outlying areas, for the purpose of information interchange among data processing systems.

This draft revised standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$10.00

#

105

X3/87/03-047-X, S
Project 585-D

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

NEWS RELEASE

104 (X) JCA-4

For more information contact:

Donald Schricker, X3J4 Chairman
617-967-7628

Date: March 5, 1987

***** ADDITION OF INTRINSIC FUNCTION MODULE ***
TO PROGRAMMING LANGUAGE COBOL (X3.23-1985)**

X3.23A-198X NOW AVAILABLE FOR PUBLIC REVIEW AND COMMENT

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a four-month public review and comment period on draft proposed American National Standard, X3.23A-198x. The public review period extends from April 10, 1987 to August 10, 1987.

Significant additions to COBOL since the base standard was approved have resulted in this public review of the draft COBOL addendum. This addendum addresses the Intrinsic Function Module. The Intrinsic Function module provides the capability to reference a data item whose value is derived automatically at the time of reference during the execution of the object program.

Data processing problems frequently require the use of values which are not directly accessible in the data storage associated with the object program. These data values must be derived through performing some operations on other data. Functions provide the capability to reference a data item whose value is derived automatically at the time of reference during the execution of the object program.

***** MKK *****

**Operating under the procedures of The American National Standards Institute.*
X3 Secretariat: Computer and Business Equipment Manufacturers Association
311 First Street, N.W., Suite 500, Washington, DC 20001-2178

106

ANNOUNCEMENT OF PUBLIC REVIEW AND COMMENT PERIOD
ON X3.23A-198X, ADDENDUM TO COBOL STANDARD (CONTINUED)

The value returned by a function is considered to be a data value. A mechanism is provided at object time to assign a data value to a function when it is referenced. In order to determine the function's value, the evaluation mechanism may require access to data values provided by the referencing program. These data values are provided by specifying parameters, known as arguments, when referencing the function. Specific functions may place constraints on these arguments such as range, etc.

An implementation of Standard COBOL must recognize as reserved words all of the COBOL reserved words occurring in the specification of the seven required modules and the four optional modules of Report Writer, Communication, Debug, and Segmentation. An implementation of Standard COBOL need not recognize any new reserved words introduced by the optional module of Intrinsic Function until that module is included in an implementation.

This draft standard is available for public review and comment for a four-month period ending August 10, 1987. Copies may be obtained from GLOBAL ENGINEERING DOCUMENTS, INC. by calling 800-854-7179.

Single Copy Price: \$30.00

#

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS*****NEWS RELEASE**

104(*)JCA-4

For more information contact:

Release: Immediate

March 13, 1987

Catherine A. Kachurik
202-737-8888**VICE CHAIRMAN OF ACCREDITED STANDARDS COMMITTEE, X3 NAMED**

Washington, D.C. -- Donald C. Loughry, Standards Manager for the Information Networks Group at Hewlett-Packard, has been named the Vice Chair of the Accredited Standards Committee, X3, Information Processing Systems. The vacancy was created with the appointment of Richard Gibson, AT&T, as Chair.

X3 develops the standards on which information processing companies rely for increased markets, both domestically and internationally.

Loughry, who has been with H-P since 1956, has participated in standards development in both in the X3 arena and in IEEE. He has been actively involved with codes, interfaces, data communications and networking standards.

"The user community has voiced a great deal of confidence in Don's ability," says Vico E. Henriques, President of CBEMA. "I am sure that the team of Gibson and Loughry will take X3 to new heights in the field of information processing systems."

#

*Operating under the procedures of The American National Standards Institute.

X3 Secretariat: Computer and Business Equipment Manufacturers Association
311 First Street, N.W., Suite 500, Washington, DC 20001-2178

Tel: 202/737-8888
Fax: 202/638-4922



american national standards institute, inc
1430 broadway, new york, n.y. 10018

news release

104 (*) JCA-4

For Additional Information
Contact: Cindy Swanteck
212 642-4922

FOR IMMEDIATE RELEASE

February 24, 1987

INTERNATIONAL STANDARD GENERALIZED
MARKUP LANGUAGE PUBLISHED

The world now has an internationally accepted standard for a generalized markup language for programming computer-assisted text processors widely used by the publishing industry and companies that generate in-house publications. ISO International Standard 8879 is available in the U.S from the American National Standards Institute.

The markup language specified in ISO 8879 makes processing easier and provides for flexibility in output by separating a document's text from the instructions that dictate style for such features as titles, subheadings, paragraphing, and lists. Separation is achieved by applying general identifiers and generic coding to document elements requiring special treatment.

The new international standard includes an abstract syntax for markup of document elements. To accommodate the many environments in which the standard will be used, provision is made for customizing the language and taking into account different user preferences in text entry conventions and different requirements imposed by a variety of keyboards and displays.

Use of the standard is expected to reduce markup costs and production lead time and allow interchange among word processors with different

-more-

ANSI 1987 Public Conference on

INDUSTRY SELF-REGULATION

March 25, 1987

Marriott Crystal Gateway Hotel

Arlington, Virginia

U 109

functional capabilities. Another expected advantage is the ability to create auxiliary documents like mailing lists from original text.

Appendixes to ISO 8879 provide basic concepts, an outline of implementation issues, examples of application, and identifiers for special alphabet (Greek, Cyrillic) characters, graphic and publishing character sets, and technical and mathematical symbols.

The standard was developed by Technical Committee 97, Information Processing Systems, of the International Organization for Standardization (ISO). ANSI administers the TC 97 secretariat.

Copies of ISO 8879-1986--Standard Generalized Markup Language (SGML)--are available from ANSI's International Sales Department at \$50 each.

The American National Standards Institute is a private, nonprofit organization that coordinates the development of voluntary national standards, approves American National Standards, and represents U.S. interests in the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

R731
0140H

X3/SD-4

FEBRUARY 1987

104 (X) JCA-5

ACCREDITED STANDARDS COMMITTEE *

X3-INFORMATION PROCESSING SYSTEMS

PROJECTS MANUAL

* Operating under the procedures of the
American National Standards Institute

secretariat:



Computer & Business Equipment Manufacturers Association

X3 PROJECTS MANUAL

FOREWORD

X3 administers its responsibilities for consideration and development of standards within its scope by means of a Project Management System.

- New X3 work is initiated by a Proposal, which, if sufficient interest is found, causes initiation of a STUDY project to determine the feasibility and need for standards on that subject.
- When the Study conclusions are affirmative, a project Recommendation is submitted to X3 letter ballot. If approved by at least 2/3 of the X3 membership, a DEVELOPMENT project is established to produce a standard.
- The project is converted to MAINTENANCE type when the proposed draft is approved by ANSI as an American National Standard.
- It is converted to REVISION type when a substantive change is proposed to and approved by X3, as a result of experience with and comments on the standard.
- The project is converted to REAFFIRMATION when, five years after publication the standard is reviewed and found to require no modification.
- LIAISON projects identify work of an industry, government, professional or international standards body, in which X3 has an interest but for which it has no directly related project.
- An INTERNATIONAL DEVELOPMENT project is one with an approved New Work Item which X3 has committed to support, and which is intended to result in an International Standard.

The Project Management System provides X3 the means used to identify, catalog, monitor and report its activities, and for filing its technical papers. A project may be terminated by X3 decision at any time prior to completion of a standard. However, once an American National Standard is published, the project remains, going through cyclic Maintenance, Revision and/or Reaffirmation stages as required until the standard is withdrawn.

X3/SD-4 provides a listing of the current X3 projects, arranged by technical discipline and cross-referenced to the related ISO/TC97 projects, proposals and approved standards.

<u>X3 PROJ. NO/TYPE</u>	<u>TITLE</u>	<u>STD. DESIG.</u>	<u>EST. Cmpl. DATE</u>	<u>ISO PROJ. DESIG.</u>	<u>ISO. DOC. NO.</u>	<u>SD-3 REF. NUMBER</u>
X3A1 - OCR & MICR						
0017-R	Print Specifications for Magnetic Ink Character Recognition Incorporates X3 Proj. 314	X3.2-1976	1986	97.03.07	ISO 1004 -77	SPARC/82-477
0018-L	Bank Check Specifications for Magnetic Ink Character Recognition	X9.13-1983				
0057-M	Character Set for Optical Character Recognition (OCR-A)	X3.17-1981	1986	97.03.01	ISO 1073 /I-76	
0061-M	Character Set for Optical Character Recognition (OCR-B)	X3.49-1982	1987	97.03.01	ISO 1073 /II-76	
0062-M	Character Set for Handprinting	X3.45-1982	1987			
0069-M	Optical Character Recognition (OCR) Character Positioning	X3.93M-1981	1986	97.03.02	ISO 1831 -80	
0227-M	Matrix Character Sets for Optical Character Recognition	X3.111-1986	1984			SPARC/854
0228-L	OCR Font for 7 and 9 Matrix Printers (Liaison with ECMA/TC4)	-				
0254-R	Paper Used In Optical Character Recognition (OCR) Systems	X3.62-1979	1986			X3/85-31R
0274-M	Design of OCR Forms	X3/TR-5-1982	1987			
0284-RF	Optical Character Recognition (OCR) Inks	X3.86-198x	1985			
0285-M	Guideline for Optical Character Recognition (OCR) Print Quality	X3.99-1983	1988		ISO 1831 -80	
0312-DT	Basic Information on OCR	X3/TR-XX-	-1984			SPARC/78-136
0313-S	Bar Code Standards	-				
0477-D	ANS for Guideline for Bar Code Print Quality	-	1986			SPARC/84-320
X3B5 - DIGITAL MAGNETIC TAPE						
0038-R	Magnetic Tape Labels and File Structure for Information Interchange	X3.27-198X	1986	97.15.1	ISO 1001 -79	X3/85-1204
0070-M	Unrecorded Magnetic Tape For Information Interchange (9-track 200 and 800 cpi, nrzi, and 1600 cpi, pe)	X3.40-1983	1988	97.11.04	ISO 1864 -85	
0071-M	Recorded Magnetic Tape for Information Interchange (200 CPI, NRZI) (With X3L2, see Proj. 237-M)	X3.14-1983	1988	97.11.02	ISO 1862 -75	
0072-M	Recorded Magnetic Tape for Information Interchange (800 CPI, NRZI)	X3.22-1983	1988	97.11.03	ISO 1863 -76	
0073-M	Recorded Magnetic Tape for Information Interchange (1600 CPI, PE)	X3.39-1986	1986	97.11.05	ISO 3788 -76	
0213-M	Magnetic Tape Cassettes for Information Interchange (3.81 mm, 0.150 inch) Tape at 32 bps (800 BPI), PE	X3.48-1986	1986	97.11.06	ISO 3407 -76	SPARC/82-1055
0217-M	Magnetic Tape Cassette Label	ISO 4341-'84	1989	97.15.3	ISO 4341 -78	
0221-R	Unrecorded Magnetic Tape Cartridge for Information Interchange (0.250 Inch, 1600 BPI, Phase Encoded)	X3.55-1982	1985	97.11.10	ISO 4057 -79	X3/85-1219
0233-M	Recorded Magnetic Tape for Information Interchange (6250 CPI, Group Encoded Recording)	X3.54-1986	1986	97.11.11	ISO 5652 -83	SPARC/573
0236-L	Recorded Magnetic Tape (7-Track, 200 CPI NRZI)	-		97.11.01	ISO 1861 -75	
0250-RF	One-Half Inch Magnetic Tape Interchange Using a Self-Loading Cartridge	X3.85-1981	1986	97.11.12	ISO 6098 -82	SPARC/591
0255-M	Recorded Magnetic Tape Cartridge for Info. Interchange, 4-Track, 0.250 Inch 630mm, 1600 BPI, 63 BPMM Phase Encoded	X3.56-1986	1985	97.11.10	ISO 4057 -79	SPARC/82-421
0256-W	Magnetic Tape Cassette for Info. Interchange, Dual Track Complementary Return-To-Bias, Four-States Recording	X3.59-1981	1986	97.11.08		
0271-RF	Parallel Rec. Mag. Tape Cartridge for Info. Interchange, 4-Track, 0.250 Inch, 6.30 mm, 1600 BPI, 63BPMM, Phase Encoded	X3.72-1981	1986			SPARC/831
0282-M	Unrecorded Magnetic Tape Minicassette For Information Interchange, Coplanar 3.81 mm (0.150 Inch)	X3.103-1983	1988			SPARC/754
0350-M	Recorded Magnetic Tape Cartridge, 1/4 Inch 6400 BPI, 4-Track	X3.116-1986	1986	97.11.10	DIS 8063 /2	SPARC/80-564
0366-M	Unrecorded 1/4 Inch Recorded Magnetic Tape Cartridge (6400-10000 BPI)	X3.127-1987	1986	97.11.10	DIS 8063 /1	SPARC/82-420
0391-D	Recorded Magnetic Tape for Information Interchange, 0.5 Inch (12.7 mm) Tape, Nine Track, 3200 CPI (126 CPMM)	X3.157-198x	1985			SPARC/83-517

<u>X3 PROJ.</u> <u>NO./TYPE</u>	<u>TITLE</u>	<u>STD.</u> <u>DESIG.</u>	<u>EST.</u> <u>CMPL.</u> <u>DATE</u>	<u>ISO</u> <u>PROJ.</u> <u>DESIG.</u>	<u>ISO.</u> <u>DOC.</u> <u>NO.</u>	<u>SD-3 REF.</u> <u>NUMBER</u>
<u>X387 - MAGNETIC DISKS (CONTINUED)</u>						
0123-L	Track Format for Six Disk Pack	ISO LIAISON		97.10.1 .2	ISO 3561 -76	
0124-L	Track Format for Single Disk Cartridge (Top Loaded)	ISO LIAISON		97.10.2 .3	ISO 3563 -76	
0224-M	Unrecorded Single Disk Cartridge (Front Loading, 2200 BPI)	X3.52-1981	1987			
0225-M	Interchangeable Magnetic Twelve-Disk Pack (100 Megabytes)	X3.63-1981	1986	97.10.4	ISO 4337 -77	
0251-M	Interchangeable Magnetic Twelve-Disk Pack (200 Megabytes)	X3.84-1981	1987	97.10.6	DIS 5653	SPARC/598
0275-M	Unrecorded Single-Disk, Double-Density Cartridge (Front Loading 2200 BPI, 200 TPI)	X3.89-1981	1987			SPARC/750-A
0277-M	Unformatted Single-Disk Cartridge (Top Loading, 200 TPI, 4400 BPI)	X3.76-1981	1987			SPARC/750-C
0321-M	Physical, Mechanical & Magnetic Characteristics of an Unformatted 80 Megabyte Trident Pack for Use at 370 TPI, 6000BPI	X3.115-1984	1989			SPARC/78-137
0328-M	14-Inch (356 mm) Diameter Low Surface Friction Magnetic Storage Disk	X3.112-1984	1989	97.10.7	ISO 6901	SPARC/79-55R
0345-M	Contact Start/Stop Storage Disk, 158361 Flux Transitions Per Track, 8.268 Inch Outer and 3.937 Inch Inner Diameters	X3.119-1984	1989		DIS 7298	SPARC/80-234 REV.
0344-M	Contact Start/Stop Storage Disk, 95840 Flux Transitions Per Track, 7.874 Inch Outer & 2.50 Inch Inner Diameters	X3.120-1984	1989		ISO 7297	SPARC/80-235 R. 10/10/80
0353-M	Contact Start/Stop Storage Disk: 83000 Flux Transition Per Track 130 mm (5.118 In) Outer Diam. 40 mm (1.575 In) Inner	X3.128-1985	1990		DP 7929	SPARC/81-52 REV. 3/18/81
0356-D	A Contact Start/Stop Metallic Thin Film Storage Disk, 83,333 Flux Transition Per Track, 130MM Outer Dia. & 40MM Inner Dia	-	1987			X3/85-607
0360-D	5 1/4 Inch Rigid Disk Removable Cartridge	X3.155-198X	1987	97.10.11	DP 8679	SPARC/81-893R 1/6/82
0369-D	Nominal 8 inch Rigid Media Removable Cartridge	X3.156-198x	1987	97.10.12	DP 8680	SPARC/81-265R 5/14/81
0479-D	95 MM Rigid Digital Recording Disk	-	1987			SPARC/84-527
0489-L	Liaison with the Semiconductor Equipment Manufacturers Institute (SEMI) for the Development of Substrate Standards	LIAISON		N/A		X3/84-916
0492-D	100 mm Rigid Digital Recording Disc for Cartridge Applications	-	1987			SPARC/84-471
<u>X387.1 - TEST METHODS AND PROCEDURES</u>						
0582-S	Study Project on Test Methods and Procedures	-				X3/86-1027
<u>X388 - FLEXIBLE DISK CARTRIDGES (FDC)</u>						
0231-M	Single-Sided Unformatted Flexible Disk Cartridge for 6631 BPR Use	X3.73-1986	1986	97.11.9 .1	ISO 5654 /1-84	SPARC/541
0232-D	Flexible Disks - Recorded Characteristics	-	1986	97.11.9	ISO 5654 /2-82	SPARC/818
0272-D	Flexible Disk Labels and File Structures	ISO 7665-	1987	97.15.8	ISO 7665	SPARC/80-818
0286-M	Two-Sided Unformatted 8-Inch (200 MM) Double Density Flexible Disk Cartridge (For 13262 FTFR Two-headed Application)	X3.121-1984	1989	97.11.9 .3.1	ISO 7065 /1-2	SPARC/77-28
0287-M	Two Sided Unformatted (200 mm) Double Density Flexible Disk Cartridge, General Physical and Magnetic Requirements	X3.121-1984	1989			SPARC/77-20.
0306-R	One-Sided Single-Density Unformatted 5.25 Inch Flexible Disk Cartridge	X3.82-1980	1986	97.11.9 .4.1	ISO 6596 /1-2	SPARC/77-27
0322-M	Two-Sided, Double-Density, Unformatted 5.25 Inch (130 mm) 48-tpi (1.9 tpm) Flexible Disk Cartridge for 7958 BPR Use	X3.125-1984	1989	97.11.9 .5.1	ISO 7487 /1	SPARC/79-73-A
0354-M	One or Two-Sided Double Density Unformatted 5.25 Inch (130 mm) 96 Tracks Per Inch Flexible Disk Cartridge	X3.126-1986	1990	97.11.15		SPARC/81-313R 8/3/81
0373-D	3.5 Inch Flexible Disk Cartridge	X3.137-198x	1986			SPARC/83-423
0453-D	Unformatted 72 mm (3 Inch Nominal) Two-Sided Double Density Flexible Disk Cartridge for Information Interchange	X3.142-198X	1985			SPARC/83-731
0475-I	NWI TC97 N1368, Std. of Flexible Disk Cartridges for Data Interchange Having a Diameter Smaller than 100 MM	-		97.11.15	TC97 N1368	SPARC/84-429
0494-D	5.25 Inch High Density (130 mm) Flexible Disk Cartridge	-	1986		DP 8630 /1,/2	X3/84-1065R

<u>X3 PROJ.</u> <u>NO./TYPE</u>	<u>TITLE</u>	<u>STD.</u> <u>DESIG.</u>	<u>EST.</u> <u>CMPL.</u> <u>DATE</u>	<u>ISO</u> <u>PROJ.</u> <u>DESIG.</u>	<u>ISO.</u> <u>DOC.</u> <u>NO.</u>	<u>SD-3 REF.</u> <u>NUMBER</u>
<u>X3B11 - OPTICAL DIGITAL DATA DISKS (CONTINUED)</u>						
0409-D	Unrecorded Optical Media Unit for Digital Information Interchange, Nominal 120 mm (4.72 Inch) Diameter	-	1986			SPARC/83-540R
0456-D	Unrecorded Optical Media Unit for Digital Information Interchange, Nominal 356 mm (14.00 Inch) Diameter	-	1986	97.23.1	NWI TC97 N1419	SPARC/83-538R
0457-D	Unrecorded Optical Media Unit for Digital Information Interchange, Nominal 130 mm (5.25 Inch) Diameter	-	1986	97.23.4	NWI TC97 N1422	SPARC/83-539R
0480-D	Recorded Characteristics of Optical Media Units for Digital Info. Interchange, Nominal 120 mm (4.72 Inch)	-	1986			SPARC/84-600
0481-D	Recorded Characteristics of Optical Media Units for Digital Info. Interchange, Nominal 130 mm (5.25 Inch)	-	1986	97.23.4	NWI TC97 N1422	SPARC/84-601
0482-D	Recorded Characteristics of Optical Media Units for Digital Info. Interchange, Nominal 200 mm (8 Inch)	-	1986	97.23.3	NWI TC97 N1421	SPARC/84-602
0483-D	Recorded Characteristics of Optical Media Units for Digital Info. Interchange, Nominal 300 mm (12 Inch)	-	1986	97.23.2	NWI TC97 N1420	SPARC/84-603
0484-D	Recorded Characteristics of Optical Media Units for Digital Info. Interchange, Nominal 356 mm (14 Inch)	-	1986	97.23.1	NWI TC97 N1419	SPARC/84-604
0524-D	File Structure and Labelling of Optical Digital Data Disks for Information Interchange	-	1988			X3/85-356
0581-D	Unrecorded Optical Media Unit for Digital Information Interchange - Nominal 90 mm (3.5 Inch) Diameter	-198X	1988			X3/86-1588
<u>X3B2 - DATA BASE</u>						
0355-M	Database Language NDL	X3.133-1986	1986	97.21.3	SC21 N174	SPARC/81-191R
0363-M	Database Language SQL	X3.135-1986	1986	97.21.3	SC21 N173	SPARC/81-689R .2
0525-D	ANS for Extended Database Language SQL	-	1986			X3/85-657R
0571-D	Embedding of SQL Statements into Programming Languages	-	1986			X3/85-658R
0583-L	Remote Data Access (RDA) Service and Protocol	-				X3/86-1760
0594-D	Database Language SQL/Addendum 1 (Integrity Enhancement Feature)	-				X3/86-1761
<u>X3B3 - COMPUTER GRAPHICS</u>						
<u>X3B3.1 - PROGRAMMER'S HIERARCHICAL INTERACTIVE GRAPHICS SYSTEM (PHIGS)</u>						
0460-D (PHIGS)	Programmer's Hierarchical Interactive	X3.144-198x	1987	97.21.24	SC21 N819 1/18/84	SPARC/83-832 Rev. Graph1
<u>X3B3.3 - VIRTUAL DEVICE INTERFACE</u>						
0346-D	Computer Graphics Interface (CGI)	X3.161-198x	1987	97.21.26	SC21/2 N1179	SPARC/80-420R
0347-M	Computer Graphics Metafile (CGM) (Formerly VDM)	X3.122-1986	1986	97.21.5	DIS 8632	SPARC/80-557
<u>X3B3.4 - LANGUAGE BINDING</u>						
0529-D	ANS for Ada Language Binding of the Graphical Kernel System (GKS)	X3.124.3-198X	1987	.3	97.21.7 /3	DP 8651X3/85-1166R
0530-D	ANS for the Ada Language Binding of the Programmers Hierarchical Interactive Graphics Standard (PHIGS)	-	1987	97.21.27 .3	SC21 N668	X3/85-1168R
0531-D	ANS for the Pascal Language Binding of the Graphical Kernel System (GKS)	X3.124.2-198x	1987	.2	97.21.7 /2	DIS 8651X3/85-1169R
0532-D	ANS for the Fortran Language Binding of the Programmers Hierarchical Interactive Graphics Standard (PHIGS)	X3.144.1-198x	1987	.1	97.21.27	SC21 N667X3/85-1172R
0533-D	C Language Binding of the Graphical Kernel System (GKS)	-	1987	97.21.7 .7	SC21 N669	X3/85-1173R
0534-D	C Language Binding of the Programmers Hierarchical Interactive Graphics System (PHIGS)	-	1988	97.21.27 .4		X3/85-1175R
0535-M	Graphical Kernel System (GKS) FORTRAN Binding	X3.124.1-1985	1990	.1	97.21.7	DP 8651/1
0543-D	ANS for the Ada* Language Binding of the 3-D Extensions to GKS (*Ada is a Registered trademark of the U.S. Government)	-	1988	97.21.7 .6		X3/85-1167R

<u>X3 PROJ.</u> <u>NO./TYPE</u>	<u>TITLE</u>	<u>STD.</u> <u>DESIG.</u>	<u>EST.</u> <u>CHPL.</u> <u>DATE</u>	<u>ISO</u> <u>PROJ.</u> <u>DESIG.</u>	<u>ISO.</u> <u>DOC.</u> <u>NO.</u>	<u>SD-3 REF.</u> <u>NUMBER</u>
	<u>X3J7.1 - PROCESSOR LANGUAGES</u>					
	<u>X3J7.2 - POSTPROCESSOR LANGUAGES</u>					
	<u>X3J7.3 - LATHE LANGUAGE</u>					
	<u>X3J7.4 - ROBOTICS LANGUAGE</u>					
	<u>X3J9 - PASCAL</u>					
0317-M	Programming Language PASCAL (Note: Complete Designation is ANSI/IEEE770X3.97-1983)	X3.97-1983	1988	97.05.10	DIS 7185	SPARC/79-111
	<u>X3J9.1 - PASCAL EXTENSIONS</u>					
0345-D	Extended Programming Language PASCAL	X3.160-198x	1986			X3/85-1900
	<u>X3J10 - APL</u>					
0331-D	Programming Language APL	X3.123-198X	1986	97.05.11	DP 8485	SPARC/79-349
0577-D	ANS for Advancements in the APL Language	-	1991			X3/86-527
	<u>X3J11 - C LANGUAGE</u>					
0381-D	Programming Language C 3/2/83	X3.159-198x	1985			SPARC/83-79R
	<u>X3J12 - DIBOL</u>					
0507-D	Programming Language DIBOL	-	1987			X3/84-994R
	<u>X3J13 - COMMON LISP</u>					
0574-D	COMMON LISP	-	1988		86-344	
	<u>X3K1. - COMPUTER DOCUMENTATION</u>					
0016-M	Guide For Technical Documentation of Computer Projects	X3/TR-6-1982	1987			
0264-L	Computer Configuration Charts	-		97.5.5		
0266-L	Specification of Single-Hit Decision Tables	-		97.7.7	DIS 5806	
0299-L	Symbols & Conventions for Program Flow, Program Networking, Data Flow & Computer Configuration	-		97.7.8	DIS 5807	
0506-L	Guidelines for the Documentation of Computer-Based Systems	-		97.7.3	DP 6592	N/A
0516-D	Documentation Standard for Small Computer Applications	-	1987	97.07.03	SC7 N537	X3/85-112
0517-D	Logical Flow of Activities in the Life of an Automated System	-	1986			X3/85-111
	<u>X3K5 - VOCABULARY FOR INFORMATION PROCESSING SYSTEMS</u>					
0026-D	American National Dictionary for Information Processing Systems (ANDIPS)	X3/TR-1-1982	1987			SPARC/84-581
0027-L	ISO Vocabulary of Data Processing	-		97.1.1 ..20	ISO 2382 /I-XVI	
	Section 01: Fundamental Terms			97.1.1	ISO 2382/1-84	
	Section 02: Mathematics & Logic. Arith. & Logic Oper.			97.1.2	ISO 2382/2-76	
	Section 03: Equipment Technology			97.1.3	ISO 2382/3-76	
	Section 04: Organization of Data			97.1.4	ISO 2382/4-74	
	Section 05: Representation of Data			97.1.5	ISO 2382/5-74	
	Section 06: Preparation of Handling Data			97.1.6	ISO 2382/6-74	
	Section 07: Digital Computer Programs			97.1.7	ISO 2382/7-	
	Section 08: Control, Integrity, and Security			97.1.8	ISO 2382/8-	
	Section 09: Data Communication			97.1.9	ISO 2382/9-79	
	Section 10: Operating Techniques & Facilities			97.1.10	ISO 2382/10-76	
	Section 11: Control, Input-Output & Arithmetic Equipment			97.1.11	ISO 2382/11-76	
	Section 12: Storage Techniques & Data Media			97.1.12	ISO 2382/12-78	
	Section 13: Computer Graphics and Micrographics			97.1.13	ISO 2382/13-84	
	Section 14: Reliability, Maintainability & Availability			97.1.14	DIS 2382/14-	
	Section 15: Programming Languages			97.1.15	DIS 2382/15-	
	Section 16: Information Theory			97.1.16	ISO 2382/16-78	
	Section 17: Data Bases			97.1.17		
	Section 18: Open Systems & Distributed Data Networks			97.1.18	DP 2382/18	
	Section 19: Analogue Computing			97.1.19	ISO 2382/19-80	
	Section 20: Systems Development			97.1.20	DP 2382/20	

<u>X3 PROJ.</u> <u>NO./TYPE</u>	<u>TITLE</u>	<u>STD.</u> <u>DESIG.</u>	<u>EST.</u> <u>CMPL.</u> <u>DATE</u>	<u>ISO</u> <u>PROJ.</u> <u>DESIG.</u>	<u>ISO.</u> <u>DOC.</u> <u>NO.</u>	<u>SD-3 REF.</u> <u>NUMBER</u>
<u>X3L2 - CODES & CHARACTER SET (CONTINUED)</u>						
0514-D	ANS for Alternate Controls for Character Imaging Devices	-	1985			X3/84-1691
0536-S	Identifying Videotex Requirements to be Met by Message Handling Systems Under Dev. in TC97/SC18 & X3V1 (MOTIS) & CCITT	-	1986			X3/85-1207
0537-S	Draft Joint X3L2/X3V1 Project for Identifying Data Link Protocol Requirements for Videotex	-	1986			X3/85-1208
0538-S	Joint X3L2/X3V1 Project for Identifying Session Layer Protocol Requirements for Videotex	-	1986			X3/85-1209
0539-S	Joint X3L2/X3V1 Project for Identifying Presentation Layer Protocol Requirements for Videotex	-	1986			X3/85-1210
0540-S	Joint X3L2/X3V1 Project for Identifying Common Application Layer Protocol Requirements for Videotex	-	1986			X3/85-1211
0554-D	Extending ANS X3.110-1983, NAPLPS, to Accommodate Office Systems Requirements For Videotex; see also X3 Proj. 359	-	1986			X3/85-1206
0555-DT	Technical Report for Specifying Guidelines for Implementors of X3.110-1983	-	1986			X3/85-1214
0564-D	Extending ANS X3.110-1983, Videotex/ Teletext Presen. Level Protocol Syntax, to Include Photographic Image Coding	-	1986			X3/85-1212
<u>X3L2.1 - VIDEOTEK/TELETEXT</u>						
0359-M	Videotex/Teletext Presentation Layer Protocol Syntax (North American PLPS)	X3.110-1983	1988	97.02.14	ISO 6937 /1-2	SPARC/81-684
0572-DT	Technical Report for the NAPLPS Verification Test Package	-	1986			X3/85-1213
<u>X3L2.2 - ADDITIONAL CONTROL FUNCTIONS FOR X3.64</u>						
0004-R	Control Codes for 8-Bit Sets	X3.64-1979	1986	97.02.08	ISO 6429 -83	SPARC83-543,544
<u>X3L2.3 - TWO-BYTE GRAPHIC CHARACTER SET</u>						
0396-D	Two-Byte Graphic Character Set for Processing and Interchange	-	1985	97.02.18		SPARC/83-545
<u>X3L8 - DATA REPRESENTATION</u>						
0045-M	Representation for U.S. Customary, SI and other Units to be used in Systems with Limited Character Sets	X3.50-1986	1986	97.14.16	ISO 2955 -74	X3/84-1418
0083-M	Representation of Calendar Date and Ordinal Date for Information Interchange	X3.30-1985	1990			
0084-M	Representation of Local Time of Day for Information Interchange	X3.43-1986	1986	97.14.3.1	ISO 3307 -75	X3/84-1414
0085-M	Representations of Universal Time, Local Time Differentials, and U.S. Time Zone References for Information Interchange	X3.51-1986	1986	97.4.3.2	ISO 4031 -78	X3/84-1419
0087-D	Structure for Identification of Organizations	-	1986	97.14.5	DIS 6523	
0090-R	Identification of States of the U.S. (Including the District of Columbia) for Information Interchange	X3.38-1972	1986			X3/85-806
0091-R	Structure for the Identification of the Counties of the U.S. for Information Interchange	X3.31-1973	1986			X3/84-1413
0092-R	Structure for the Identification of Named Populated Places and Related Entities of the States of the U.S.	X3.47-1977	1986			X3/84-1417
0093-M	Representation of Geographic Point Locations for Information Interchange	X3.61-1986	1986	97.14.6		X3/84-1420
0096-L	Identification of Countries (ANSC 239/SC27)	239.27-1976	N/A			
0097-D	Identification of Sub-Divisions of Countries	-	1986			
0242-L	Representation of Human Sexes (Liaison with TC97/SC14)	-	N/A	97.14.10	ISO 5218 -77	
0243-L	Representation of Human Blood Type (Liaison with TC97/SC14)	-	N/A	97.14.11		
0259-L	Check Characters (Liaison with TC97/SC14)	-	N/A	97.14.15	ISO 7064 -83	

<u>X3 PROJ. NO/TYPE</u>	<u>TITLE</u>	<u>STD. DESIG.</u>	<u>EST. Cmpl. DATE</u>	<u>ISO PROJ. DESIG.</u>	<u>ISO. DOC. NO.</u>	<u>SD-3 REF. NUMBER</u>
<u>X3S3 - DATA COMMUNICATIONS (CONTINUED)</u>						
0508-I	Connectionless-mode Data Transmission NWI TC97 N1433, Design of Procedures for Testing & Conformance to Data Com- munications Protocol Based on CCITT X.25	-	N/A	97.06.38	TC97 N1433	X3/84-1167
0598-I	NWI TC97 N1606, Procedures for Testing Conformance to ISO 8073, Transport Protocol	-	N/A		TC97 N1606	
<u>X3S3.1 - DATA COMMUNICATIONS PLANNING</u>						
0248-L	Data Transmission Vocabulary	-	N/A	97.06.14	ISO 2382 /09	
<u>X3S3.3 - NETWORK LAYER</u>						
0047-M	Structure for Formatting Message Headings for Information Interchange Using ASCII for Data Comm. Sys. Control	X3.57-1986		1986		
0111-M	Bit Sequencing of ASCII in Serial-By-Bit Data Transmission	X3.15-1983	1988	97.06.01	ISO 1177	
0112-M	Character Structure and Character Parity Sense for Serial-By-Bit Data Communication in ASCII	X3.16-1983	1988	97.06.01	ISO 1155 /1177	
0114-M	Character Structure and Character Parity Sense for Parallel-By-Bit Data Communication in ASCII	X3.25-1983	1988	97.06.01		
0281-D	Code Independent Message Heacing Format (From 47)	-		1986		
0326-S	OSI Reference Model, Network Layer	-		1986 97.06.32	DIS 8348	
0365-D	ANS for Open Systems Interconnection-- Reference Model Internetwork Protocol of the Network Layer	-		1987 97.06.16	DP 7777 & 8473	SPARC/82-134
0549-D	Information Processing Systems - Data Communications - Network Layer Addressing	-		1987 97.06.32 .05		X3/85-375R
0550-D	Information Processing Systems - Data Communications - Network Layer Service Definition	-		1986 97.06.32 .01,.02		X3/85-377R
0551-DT	Information Processing Systems - Data Communications - Technical Report on Network Layer Architecture	-		1988 97.06.32 .03		X3/85-376R
<u>X3S3.4 - CONTROL PROCEDURES</u>						
0048-M	Procedures for Use of the Communication Control Characters of the ASCII Code in Specified Data Communication Links	X3.28-1986	1986	97.06.01	ISO 1745 -75	X3/84-1470
0049-R	Advanced Data Communication Control Procedures (ADCCP)	X3.66-1979	1986	97.06.16	ISO 3309 & 4335	X3/84-1127R
<u>X3S3.5 - COMMUNICATION SYSTEMS PERFORMANCE</u>						
0325-S	OSI Reference Model, Data Link Layer	-		1987 97.06.31		
0028-RF	Determination of the Performance of Data Communication Systems	X3.44-198X		1986		
0319-M	Data Communications Systems and Services -Measurement Methods for User Oriented Performance Evaluation	X3.141-1986		1986		SPARC/78-121
0320-M	Data Communication User Oriented Performance Parameters	X3.102-1983		1988		SPARC/78-120
0223-D	General Purpose Interface Between DTE & DCE for Synchronous Operation on Public Data Networks (Liaison with CCITT X.21)	X3.69-198X	N/A	97.06.11 .21		
0278-D	Network Characteristics Including User Classes of Service Facilities for Public Data Networks	-		1986 97.06.13		
0279-M	Interface Between DTE & DCE for Packet Mode Operation with Packet Switch Data Communications Networks (CCITT X.25)	X3.100-1983		1988 97.06.11		
0364-D	Standard Interface for Data Terminal Equipment Operating in the Packet Mode (See also CCITT X.25)	-		1986 97.06.11	DP 8208	SPARC/82-274
<u>X3T1 - DATA ENCRYPTION</u>						
0293-M	Data Encryption Algorithm	X3.92-1987		1986		
0295-M	Data Link Encryption	X3.105-1983		1988		SPARC/77-16
0308-S	Data Encryption	-				SPARC/77-96
0323-M	Modes of Operation for the Data Encryption Algorithm	X3.106-1983		1988		

<u>X3 PROJ.</u> <u>NO./TYPE</u>	<u>TITLE</u>	<u>STD.</u> <u>DESIG.</u>	<u>EST.</u> <u>DATE</u>	<u>ISO</u> <u>PROJ.</u> <u>DESIG.</u>	<u>ISO.</u> <u>DOC.</u> <u>NO.</u>	<u>SD-3 REF.</u> <u>NUMBER</u>
<u>X3T9 - I/O INTERFACE</u>						
0496-M	Logical Device Generic Command Set Intelligent Peripheral Interface - Logical Device Generic Command Set for Optical and Magnetic Disks	X3.132-1987	1986			X3/84-1090
0503-D	Fiber Distributed Data Interface (FDDI) Station Management (SMT) Standard	-	1987			X3/84-1093
0504-D	Intelligent Peripheral Interface - Logical Device Generic Command Set for Communications	-	1986			X3/84-1092
0505-M	Intelligent Peripheral Interface -- Logical Device Generic Command Set for Magnetic Tape	X3.147-1986	1986			X3/84-1091
0541-D	ANS for Fiber Distributed Data Interface (FDDI) Physical Layer, Medium Dependent (PMD)	-	1987			X3/85-985
0587-D	Enhanced Small Device Interface (ESDI)	-				X3/86-1673
0591-D	Intelligent Peripheral Interface, Logical Device Specific Command Set for Magnetic Tapes	-				X3/86-1687
<u>X3T9.2 - LOWER LEVEL INTERFACE</u>						
0375-R	Small Computer Systems Interface (SCSI)	X3.131-198x	1986	97.13.10	TC97 N1431	SPARC/82-878
<u>X3T9.3 - DEVICE LEVEL INTERFACE</u>						
0052-R	Interfaces Between Flexible Disk Cartridges and Their Host Controllers	X3.80-198X	1986	97.13.10	TC97 N1431	SPARC/83-360, Rev.
0053-M	Storage Module Interfaces	X3.91M-1987	1986	97.13.7, .10	TC97 N1431	X3/84-1449
0054-L	Small Computer to Peripheral Bus Interface, Data Transfer Between Computer and Peripheral	-		97.13.9	DP 7069	
0329-M	Interfaces Between Rigid Disk Drives and Hosts	X3.101-1984	1989	97.13.10	TC97 N1431	SPARC/79-342
0370-R	Intelligent Peripheral Interface Physical Level	X3.129-198x		97.13.10	TC97 N1431	X3/86-1688
<u>X3T9.5 - LOCAL DISTRIBUTED DATA INTERFACE</u>						
0337-D	Physical Layer Interface for Local Data Interfaces to a Non- Branching Coaxial Cable Bus	X3.108-198x	1986			SPARC/80-145 REV. Distri 3/19/80
0338-D	Data Link Layer Protocol for Local Data Interfaces	-	1986			SPARC/80-146 REV. Distri 3/19/80
0357-D	Physical Layer Protocol for Local Distributed Data Interface	-	1986			SPARC/81-704
0377-D	Local Distributed Data Interface (LDDI) Network Layer Protocol	-	1987			SPARC/82-1056
0379-D	Fiber Distributed Data Interface (FDDI) Physical Layer	X3.148-198x	1987			SPARC/82-1058
0380-M	Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC)	X3.139-1987	1986			SPARC/82-1059
0382-D	Fiber Distributed Data Interface (FDDI) Network Layer Protocol	-	1987			SPARC/82-1060
0556-D	ANS for Local Distributed Data Interface (LDDI) Star-Wired Physical Interface Sublayer	-	1987	97.13.01		X3/85-986
0573-D	Fiber Distributed Data Interface (FDDI) Hybrid Ring Control (HRC)	-	1987	97.13.01		X3/85-1903R
<u>X3T9.6 - CARTRIDGE TAPE DRIVES</u>						
0378-M	Device Level Interface for Streaming Cartridge and Cassette Tape Drives	X3.146-1986	1987			X3/85-0029
0401-D	Enhanced Device Level Interface for Cartridge Tape Drives	-	1986			X3/85-0030
<u>X3V1 - TEXT: OFFICE & PUBLISHING SYSTEMS</u>						
0522-D	ANS for Text Interchange on Magnetic Media	-	1986			X3/85-557
0576-D	SGML Document Interchange Format (SDIF)	-	1986		DIS 8879	X3/86-715R
0578-S	User System Interfaces for Standards Related to Text Processing: Office and Publishing Systems	-				X3/86-888R
0580-	Print Image for Interchange - Office Systems	-	1987			X3/86-1035R

<u>X3 PROJ.</u> <u>NO/TYPE</u>	<u>TITLE</u>	<u>STD.</u> <u>DESIG.</u>	<u>EST.</u> <u>CMPL.</u> <u>DATE</u>	<u>ISO</u> <u>PROJ.</u> <u>DESIG.</u>	<u>ISO.</u> <u>DOC.</u> <u>NO.</u>	<u>SD-3 REF.</u> <u>NUMBER</u>
<u>X3V1.9 - USER SYSTEMS INTERFACE & SYMBOLS (CONTINUED)</u>						
0418-I	NWI TC97 N1215, Layout and Operation of the Keyboard for Multiple Latin Alphabet Languages	-		97.18.24	SC18 N414	SPARC/84-053
0424-R	Keyboard Arrangement for Alphanumeric Machines	X3.154-198x	1986			X3/85-1272
0425-M	Alternate Keyboard Arrangement for Alphanumeric Machines	X4.22-1985	1988			
0451-D	Classification of Vocabulary of Word Processing for Document Preparation	-	1985	97.18.08, .09	SC18 N55, 56	
0515-I	TC97 NWI N1450, Functionality and Layout of Function Keys	-		97.18.25	NWI TC97 N1450	X3/84-1572
0601-I	NWI TC97 N1607, Document Interchange Format for Storage Media	-	N/A		NWI TC97 N1607	
<u>SPARC/DBSSG - DATA BASE SYSTEMS STUDY GROUP</u>						
0226-S	Data Base Management Systems	-	1987	97.21.25		
0528-I	NWI TC97 N1526, Reference Model for DBMS Standards	-		97.21.30	TC97 N1526	X3/85-724
0599-I	Software Systems Interface	-	N/A			
<u>SPARC - STANDARDS PLANNING AND REQUIREMENTS COMMITTEE</u>						
0076-M	One-Inch Perforated Paper Tape for Information Interchange	X3.18-1982	1987	97.0.3	ISC 1154 -75	
0077-M	Eleven-Sixteenths Inch Perforated Paper Tape for Information Interchange	X3.19-1982	1987			
0078-M	Take-up Reels for One Inch Perforated Tape for Information Interchange	X3.20-1982	1987	97.0.6	ISO 3692 -76	
0079-M	Specifications for Properties of Unpunched Oiled Paper Perforator Tape	X3.29-1984	1989			
0080-M	Interchange Rolls of Perforated Tape for Information Interchange	X3.34-1984	1989			
0081-M	Flow Chart Symbols and Their Usage in Information Processing	X3.5-1970	1985	97.7.1-2,4	ISC 1026 & 263c	
0211-M	Computer Program Abstracts	X3.66-1987	1986			
0216-M	Representation of Vertical Carriage Positioning Characters in Information Interchange	X3.76-1987	1986			
0253-M	Programming Language - Programming Aid for Numerically Controlled Manufacturing	X3.94-1985	1990			
0263-L	Programming Language for Industrial Process (PLIP)	-		97.5.5		SPARC/849
0265-L	Program Design	-		97.7.3		
0267-S	Long Range Planning for Programming Language Standards	-				
0269-L	Programming Language MUMPS (MUMPS Development Committee)	-				
0342-L	SPARC Liaison Project with DOD High Order Language Group on Ada*	-				X3/85-307
0422-M	Minimum Markings to Appear on Containers Used for Printing Ribbons	X4.19-1985	1990			
0423-M	Office Machines & Printing Machines Used for Information Processing, Widths of Fabric Ribbons on Spools	X4.20-1985	1990			
0426-M	10-Key Keyboard for Adding and Calculating Machines	X4.6-1984	1989			
0427-M	Minimum Requirements for Office-Type Dictating Equipment	X4.9-1984	1989			
0428-M	Remote Dictation Through an Intercommunications Switching System	X4.10-1984	1989			
0452-S	Programming Languages Study	-				SPARC/83-55R
0558-D	Presentation Transfer Syntax and Notation	-	1986			X3/85-954
<u>SC7 TAG - U.S. TAG TO TC97/SC7</u>						
0469-I	NWI TC97 N1246, Guidelines for Software Development Methods	-		97.07.16	TC97 N1246, 1445	SPARC/83-710
<u>SC21 TAG - U.S. TAG TO TC97/SC21</u>						
0513-I	TC97 NWI N1448, A Survey of DBMS Related Standardization Activities	-	N/A	97.21.25	NWI N1448	X3/84-1550
0546-I	TC97 NWI N1555, Information Processing - Remote Database Access Service and Protocol	-		97.21.31	TC97 N1555	X3/85-1146
0602-I	NWI TC97 N1629, Reference Model for Software Development	-	N/A		NWI TC97 N1629	

X3/SD-6 101

FEBRUARY 1987

104(*) JCA-6

ACCREDITED STANDARDS COMMITTEE*

X3-INFORMATION PROCESSING SYSTEMS

MEMBERSHIP & OFFICERS

**Operating under the procedures of the
American National Standards Institute**

Secretariat:

Computer & Business Equipment Manufacturers Association



TABLE OF CONTENTS

Accredited Standards Committee X3

Producer Members	1
Consumer Members	2
General Interest Members	3
Observers	4

X3 Standing Committees

Secretariat Management Committee	5
International Advisory Committee (X3/IAC)	6
Standards Planning and Requirements Committee (X3/SPARC)	7

X3 Technical Committee Officers

X3-Information Processing Systems	8
A - Character Recognition	8
B - Media	8-9
H & J - Programming Languages	9-10
K - Documentation	10
L - Data Representation	11
S - Data Communication	11
T & V - Systems Technology	12
Alphabetical Listing of Officers' Mailing Addresses	13-17
Technical Committee Officer Appointment Dates	18-19
SPARC and IAC Liaisons	20
X3 - General X3 Organizational Chart	21
Detailed X3 Technical Committee Organizational Chart	22
ISO/TC97 - International Organizational Chart	23
Names and Addresses of Chairmen of U.S.-Held Secretariats for ISO/TC97 Subcommittees	24
Names and Addresses of Working Group Convenors	24

CONSUMERS

CUBE

THOMAS EASTERDAY (P)
CUBE
10 W. 106TH STREET
P.O. BOX 527
INDIANAPOLIS, IN 46206
317/846-4211

DONALD MILLER (A)
CUBE
WOLPOFF & ABRAMSON
11140 ROCKVILLE PIKE
ROCKVILLE MD 20852-3164
301/468-0303 X 41

DECUS

DENNIS PERRY (P)
DECUS
C/O DARPA/ISTO
1400 WILSON BLVD.
ARLINGTON, VA 22209-2308
202-694-4002

DVORAK INTERNATIONAL

VIRGINIA RUSSELL (P)
DVORAK INTERNATIONAL
BOX 128
BRANDON, VT 05733
802-247-6020

EASTMAN KODAK

GARY HAINES (P)
EASTMAN KODAK
PHOTOGRAPHIC TECHNOLOGY DIV.
1700 DEWEY AVENUE, BLDG. 69
ROCHESTER, NY 14650
716-477-3597

CHARLETON C. BARD (A)
EASTMAN KODAK
C/O 74 CORNWALL LANE
ROCHESTER, NY 14617
716/722-5452

GENERAL ELECTRIC COMPANY

RICHARD W. SIGNOR (P)
GENERAL ELECTRIC COMPANY
BLDG. 30EW
1285 BOSTON AVENUE
BRIDGEPORT, CT 06601-2385
203-382-3610

WILLIAM R. KRUESI (A)
GENERAL ELECTRIC COMPANY
MAIL DROP W1E
3135 EASTON TURNPIKE
FAIRFIELD CT 06432-1008
203/373-2402

GENERAL SERVICES ADMN.

WILLIAM C. RINEHOLS (P)
GENERAL SERVICES ADMN.
ADTS
8457 RUSHING CREEK COURT
SPRINGFIELD VA 22153-2532
202/566-1180

LARRY L. JACKSON (A)
GENERAL SERVICES ADMN.
ADTS
8457 RUSHING CREEK COURT
SPRINGFIELD, VA 22153-2532
202/566-0194

GUIDE INTERNATIONAL

FRANK KIRSHENBAUM (P)
GUIDE INTERNATIONAL
AMERICAN MANAGEMENT SYSTEMS
31 MEADOWOOD DRIVE
JERICHO NY 11753-2833
212/618-0300

SANDRA SCHWARTZ ABRAHAM (A)
GUIDE INTERNATIONAL
25 HORNOR LANE
PRINCETON, NJ 08540

LAWRENCE BERKELEY LABORATORY

DAVID F. STEVENS (P)
LAWRENCE BERKELEY LABORATORY
BUILDING 50B, ROOM 2258
1 CYCLOTRON ROAD
BERKELEY CA 94720
415/486-7344

ROBERT L. FINK (A)
LAWRENCE BERKELEY LABORATORY
UCLBL
M/S 50B-2258
BERKELEY, CA 94720
415-486-5692

MAP/TOP

JAMES D. CONVERSE (P)
MAP/TOP
C/O EASTMAN KODAK COMPANY
901 ELMGROVE ROAD
ROCHESTER, NY 14650
716-726-1957

MIKE KAMINSKI (A)

MAP/TOP
C/O G.M. ADVANCED ENGINEERING
30300 MOUND ROAD
WARREN, MI 48090-9040

NATIONAL COMMUNICATIONS SYSTEM

GEORGE W. WHITE (P)
NATIONAL COMMUNICATIONS SYSTEM
8TH & SO. COURTHOUSE ROAD
ARLINGTON VA 22204-2198
202/692-2124

RAILINC CORPORATION

R. A. PETRASH (P)
RAILINC CORPORATION
50 F STREET, N.W.
WASHINGTON, DC 20001
202-639-5541

RECOGNITION TECH. USERS ASSN.

HERBERT F. SCHANTZ (P)
RECOGNITION TECH. USERS ASSN.
GRAHAM MAGNETICS
6625 INDUSTRIAL PARK BLVD.
N. RICHLAND HILLS TX 76118-2001
817/281-9450

G. W. WETZEL (A)
RECOGNITION TECH. USERS ASSN.
GRAHAM MAGNETICS
6625 INDUSTRIAL PARK BLVD.
NO. RICHLAND HILLS TX 76118
817/281-9450

SHARE, INC.

THOMAS B. STEEL (P)
SHARE, INC.
BELL LABORATORIES
CRAWFORD CORNER RD., RM 1M-355
HOLMDEL, NJ 07733
201/949-8644

ROBERT P. RANNIE (A)
SHARE, INC.
DEPT. OF COMPUTER SCIENCES
NORTHERN ILLINOIS UNIVERSITY
DE KALB, IL 60115
815-753-0423

TRAVELERS INSURANCE COMPANIES

JOSEPH T. BROPHY (P)
TRAVELERS INSURANCE COMPANIES
1 TOWER SQUARE
HARTFORD CT 06115-1599
203/277-3122

U.S. DEPT. OF DEFENSE

FRED VIRTUE (P)
U.S. DEPT. OF DEFENSE
HQ USAF/SCTT
ROOM 5E1085, PENTAGON
WASHINGTON DC 20330-5190
202/695-0499

BELKIS LEONG-HONG (A)
U.S. DEPT. OF DEFENSE
OASD (C), MS, IRMS.
ROOM 1C 535 PENTAGON
WASHINGTON, DC 20301-1100
202/695-4470

VIM INCORPORATED

CHRIS TANNER (P)
VIM INCORPORATED
CHALK RIVER NUCLEAR LABS
CHALK RIVER, ONT KOJ 1J0
CANADA
613-584-3311 X2865

M. SPARKS (A)
VIM INCORPORATED
C/O SYSTEMS DEVELOPMENT CORP.
4810 BRADFORD BLVD., N. W.
HUNTSVILLE AL 35805-1994
205/837-7610 X 247

X3 MEMBERSHIP

EX-OFFICIO MEMBERS - WITHOUT VOTE

CHAIRPERSON X3/IAC
CHAIRPERSON X3/SPARC
CHAIRPERSON SMC
CHAIRPERSON OF X3/TC'S

EX-OFFICIO OBSERVERS

MEMBERS OF SMC
MEMBERS OF X3/IAC
MEMBERS OF X3/SPARC
OFFICERS OF X3/TC'S, SC'S AND SPARC/SG'S

CHAIRPERSON OF ANSI ISSB

JAMES H. BURROWS
DIRECTOR, ICST
NATIONAL BUREAU OF STANDARDS
TECHNOLOGY BLDG. 225, RM. B154
GAITHERSBURG, MD 20899
301-975-2822

LIAISON OBSERVERS

AMERICAN NATIONAL STANDARDS INSTITUTE

FRANCES SCHROTTER (L)
ANSI
1450 BROADWAY
NEW YORK NY 10018-3308
212-642-4554

CODASYL/PROGRAMMING LANGUAGE COMMITTEE

JAN PROKOP (L)
MCGRAW-HILL, INC.
29 HARTWELL AVE.
LEXINGTON, MA 02173
617/863-5100

ASC T1 - TELECOMMUNICATIONS

ALVIN LAI (L)
T1 SECRETARIAT
EXCHANGE CARRIERS STDS. ASSOC.
FOUR CENTURY DRIVE, THIRD FLOOR
PARSIPPANY, NJ 07054
201-538-6111

X9 - FINANCIAL SERVICES

CYNTHIA L. FULLER (L)
AMERICAN BANKERS ASSOCIATION
1120 CONNECTICUT AVENUE, N. W.
WASHINGTON DC 20036-3973
202-663-5000

X12 - BUSINESS DATA INTERCHANGE

TOM JONES (L)
INFORMATION ACCESS
424 INDIANOLA ROAD
YOUNGSTOWN, OH 44512

ANSC Z39 - LIBRARY AND INFORMATION SCIENCES AND RELATED PUBLISHING PRACTICES

RAY DENNENBERG (L)
LIBRARY OF CONGRESS
NETWORK DEVELOPMENT OFFICE
WASHINGTON DC 20540-0001
202/287-5894

HUMAN FACTORS SOCIETY

PAUL REED (L)
HUMAN FACTORS SOCIETY
C/O AT&T BELL LABORATORIES
184 LIBERTY CORNER ROAD
WARREN, NJ 07060
201-580-5618

X3 LIAISONS TO ANSI PLANNING PANEL ON INDUSTRIAL AUTOMATION

LEROY RODGEFS
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET, ML05-5/T58
MAYNARD, MA 01454-2572
617-493-3163

X3 LIAISON TO IEEE COMPUTER SOCIETY STANDARDS ACTIVITIES BOARD

GARY S. ROBINSON
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET ML03-2/E51
MAYNARD, MA 01754-2572
617-493-4094

X3 LIAISON TO CCITT USMC

EDWARD KELLY
AMP INCORPORATED
M/S 210-20
P.O. BOX 3608
HARRISBURG, PA 17105-3608
717-561-6153

RICHARD GIBSON

AT&T
ROOM 5A 211
ROUTE 202 & 206N
BEDMINSTER, NJ 07921
201-234-3795

OBSERVERS

WALTER G. FREDRICKSON (O)
HARRIS CORPORATION
1025 WEST NASA BLDV. MS/14
MELBOURNE, FL 32919
305-727-9100

JOSEPH H. WUJEK (O)
APPLE COMPUTER, INC.
20525 MARIANI AVENUE M/S 22Y
CUPERTINO, CA 95014
408-973-3084

DAVID L. SEIGAL (O)
AMERICAN EXPRESS
TRAVEL RELATED SERVICES
P.O. BOX 13809
PHOENIX, AZ 85002
602-994-7837

GERARD A. RAINVILLE, JR. (O)
NATIONAL SECURITY AGENCY
P.O. BOX 11
ANNAPOLIS JUNCTION, MD 20701
301-776-1684

SELMA ZINKER (O)
TANDEM COMPUTERS
CORPORATE INFORMATION CENTER
19333 VALLCO PARKWAY
CUPERTINO CA 95014-2599
408-725-6343

PIERRE L'ALLIER (O)
CONCURRENT COMPUTER CORP.
227 BATH ROAD
SLOUGH
BERKSHIRE SL1 4AX ENGLAND

KEY TO ABBREVIATIONS

P = PRINCIPAL (VOTING MEMBER)
A = ALTERNATE MEMBER
O = OBSERVER
L = LIAISON
XO = EX-OFFICIO

I A C

International Advisory Committee

CHAIRMAN

JOSEPH S. DEBLASI (A)
IBM CORPORATION
2000 PURCHASE STREET
PURCHASE NY 10577-2597
914/697-7280

VICE CHAIRMAN

GARY S. ROBINSON (P)
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET MLO12-B/E51
MAYNARD MA 01754-2572
617/493-4094

SECRETARY

CATHERINE A. KACHURIK
X3 SECRETARIAT/CBEMA
311 FIRST STREET, NW
SUITE 500
WASHINGTON, DC 20001-2178
202-737-8866

CHARLETON C. BARD (A)
EASTMAN KODAK
C/O 74 CORNWALL LANE
ROCHESTER, NY 14617
716/722-5432

MARVIN W. BASS (A)
UNISYS
M/S 2G4
P. O. BOX 500
BLUE BELL, PA 19424-0001
215/542-3319

THOMAS EASTERDAY (P)
CUBE
10 W. 106TH STREET
P.O. BOX 527
INDIANAPOLIS, IN 46206
317/846-4211

STANLEY FENNER (P)
UNISYS
MCCORD SUITE 200
P. O. BOX 33052
DETROIT MI 48232-3052
313-972-9140

RICHARD GIBSON (P)
AT&T COMMUNICATIONS
ROOM 5A 211
ROUTE 202 & 206K
BEDMINSTER NJ 07921
201-234-3795

MARY ANNE GRAY (P)
IBM CORPORATION
2000 PURCHASE STREET
PURCHASE NY 10577-2597
914/697-7224

GARY HAINES (P)
EASTMAN KODAK
PHOTOGRAPHIC TECHNOLOGY DIV.
1700 DEWEY AVENUE, BLDG. 69
ROCHESTER, NY 14650
716-477-3597

EDWARD KELLY (P)
AMP INCORPORATED
M/S 210-01
P. O. BOX 3608
HARRISBURG PA 17105-3608
717-561-6153

FRANK KIRSHENBAUM (P)
GUIDE INTERNATIONAL
AMERICAN MANAGEMENT SYSTEMS
31 MEADOWOOD DRIVE
JERICHO NY 11753-2833
212/616-0300

PATRICK E. LANNAN (A)
AMP INCORPORATED
6435 CLEARWATER DRIVE
SPRING HILL, FL 33526-0703
904-686-4476

ROBERT E. ROUNTREE (P)
NATIONAL BUREAU OF STANDARDS
BUILDING 225, ROOM B168
GAITHERSBURG MD 20899-0999
301-975-2827

SAVA I. SHERR (P)
IEEE
345 E. 47TH STREET
NEW YORK NY 10017-2330
212-705-7966

DELBERT L. SHOEMAKER (A)
DIGITAL EQUIPMENT CORPORATION
1331 PENNSYLVANIA AVE. N.W.
SIXTH FLOOR
WASHINGTON, DC 20004
202-383-5622

DOROTHY STAPLETON (A)
U.S. DEPT. OF DEFENSE
HQ USAF/SCTT
ROOM 5E1085, PENTAGON
WASHINGTON DC 20330-5190
202/695-0499

FRED VIRTUE (P)
U.S. DEPT. OF DEFENSE
HQ USAF/SCTT
ROOM 5E1085, PENTAGON
WASHINGTON DC 20330-5190
202/695-0499

JOHN L. WHEELER (P)
XEROX CORPORATION
CARRIAGE HOUSE COMMONS
SUITE 347, 159 WEST MAIN ST.
WEBSTER, NY 14580

OBSERVER
ARTHUR G. NORTON (O)
PRIME COMPUTER, INC.
500 OLD CONNECTICUT PATH
M/S 10B-10-2
FRAMINGHAM, MA 01701
617-879-2960 X4342

EX-OFFICIO

WILLIAM C. RINEHULS (XO)
GENERAL SERVICES ADMN.
ADTS
8457 RUSHING CREEK COURT
SPRINGFIELD VA 22153-2532
202/566-1180

FRANCES SCHROTTER (XO)
ANSI
1430 BROADWAY
NEW YORK NY 10018-3308
212-642-4934

136

TECHNICAL COMMITTEE OFFICERS CHART

<u>X3TC</u>	<u>**CHAIR</u>	<u>**VICE CHAIR</u>	<u>**INT'L REP.</u>	<u>VOCAB. REP.</u>	<u>SECRETARY</u>	<u>**INT'L</u>
<u>X3 - INFORMATION PROCESSING SYSTEMS</u>						
X3	VACANT	RICHARD GIBSON AT&T COMM. 201-254-3795	N/A	N/A	CATHIE KACHURIK X3 SECRETARIAT 202-737-8688	TC97
<u>SPARC - STANDARDS PLANNING AND REQUIREMENTS COMMITTEE</u>						
SPARC	W.C. RINEHULS GEN. SER. ADMN. 202-566-1180	LEROY RODGERS DEC 617-493-3163	N/A	N/A	GWENDY PHILLIPS X3 SECRETARIAT 202-737-8888	
<u>SPARC STUDY GROUPS</u>						
DBSSG Database Sys. Study Group	EDWARD STULL GTE GVT. SYS. 301-294-8649	NANCY MCDONALD COMP. TECH. PLNG. 813-968-2660	ELIZABETH FONG NBS 301-975-3250	EDWARD STULL GTE GVT. SYS. 301-294-8649	ELIZABETH FONG NBS 301-975-3250	SC21
<u>IAC - INTERNATIONAL ADVISORY COMMITTEE</u>						
IAC	JOSEPH DEBLASI IBM CORP. 914-697-7280	GARY ROBINSON DEC 617-493-4094	N/A	N/A	CATHIE KACHURIK X3 SECRETARIAT 202-737-8688	
<u>SMC - SECRETARIAT MANAGEMENT COMMITTEE</u>						
SMC	VACANT	DELBERT SHOEMAKER DEC 202-383-5622	N/A	N/A	STANLEY ALLEN X3 SECRETARIAT 202-737-8688	
<u>X3 SUBCOMMITTEES</u>						
SC7 TAG	A. J. NEUMANN NBS 301-975-3338	RICHARD WERLING 703-820-4034	N/A	N/A	VACANT	SC7
SC21 TAG	J. ASCHENBRENNER IBM CORP. 919-254-0299	HELEN WOOD NBS 301-975-3241	N/A	N/A	VACANT	SC21
SC22 TAG	HELEN WOOD NBS 301-975-3241	ROBERT POLLETT IBM CORPORATION 301-564-2108	N/A	N/A	VACANT	SC22
<u>X3 TECHNICAL COMMITTEES</u>						
<u>A - RECOGNITION</u>						
X3A1 OCR & MICR	ROGER MONELL MONARCH MARKING 714-893-0511	ROBERT BLOSS UNISYS 313-451-4298	JOHN MCDONNELL RECOG. EQUIP. 214-579-5800	CARL KNOEDEL STD. REG. CO. 513-443-1072	CARL KNOEDEL STD. REG. CO. 513-443-1072	
X3A1.1 FONT DESIGN	PATRICK TRAGLIA IBM CORP. 607-752-3357	VACANT	N/A	N/A	VACANT	
X3A1.2 OCR SUPPL. & FORMS	CARL KNOEDEL STD. REG. CO. 513-443-1072	VACANT	N/A	N/A	VACANT	
X3A1.3 IMAGE DEF. & MEASUREMENT	C.E. BISS PHO. SCI. CORP. 716-265-1600	VACANT	N/A	N/A	VACANT	
<u>B - MEDIA</u>						
X3B5 DIGITAL MAGNETIC TAPE	R. STEINBRENNER AT&T TECH. 201-386-7053	SAM CHEATHAM STOR. TECH. CORP. 303-673-3359	MICHAEL HOGAN NBS 301-975-2926	ARNOLD ROCCATI EG&T WASH. ANAL. 301-840-3277	VACANT	97/11
X3B6 INSTRUMENTA. TAPE	BILL POLAND NASA/GODDARD 301-286-8592	VACANT	STAN REYNOLDS SAND. MATL LABS.	R.F. DAVIS THORN-EMI	VACANT	97/WG2
X3B7 MAG. DISKS	BILL CARLSON PROQUIP 408-496-0322	VACANT	BILL CARLSON PROQUIP 408-496-0322	PAUL JAHNKE 3M COMPANY 612-736-0117	A. SARATORA BASF SYS., INC. 617-271-4255	97/10

NOTE: WHERE THE SMC APPOINTMENT BALLOT IS NOT COMPLETE OR VOLUNTEERS HAVE NOT SUBMITTED THE NECESSARY DOCUMENTATION, THE OFFICER POSITION HAS BEEN LISTED AS ACTING (*) OR "VACANT"
 ** SMC ELECTED OFFICER
 *** CORRESPONDING INTERNATIONAL STANDARDS COMMITTEE

127

<u>X3TC</u>	<u>**CHAIR</u>	<u>**VICE CHAIR</u>	<u>**INT'L REP.</u>	<u>VOCAB. REP.</u>	<u>SECRETARY</u>	<u>**INT'L</u>
<u>H & J - PROGRAMMING LANGUAGES (CONTINUED)</u>						
X3B4.2 IRDS EXTER. SOFTWARE INTPC.	VACANT	VACANT	N/A	N/A	VACANT	
X3J1 PL/I	JOHN KLENSIN MIT 617-253-1355	CHARLES NYLANDER DEC 603-881-2081	*K.R. LUND IBM CORP.	ARTHUR COSTON APPLD. INFO. SYS. 919-942-7801	VACANT	97/22
X3J1.3 GENERAL PURPOSE SUBSET	VACANT	VACANT	N/A	N/A	VACANT	
X3J2 BASIC	JAMES TOTTON DEC 603-881-2228	ANDREW KLOSSNER TEKTRONIX, INC.	VACANT	JOHN CUGINI NBS 301-975-3248	VACANT	97/22
X3J3 FORTRAN	JEANNE ADAMS NATL CTR. AT. RES 303-497-1275	JERROLD WAGENER AMOCO PROD. CO. 918-664-3415	VACANT	JAMES MATHENY COMP. SCI. CORP. 213-615-0311	JEANNE MARTIN LAWRENCE LIV. 415-422-3753	97/22
X3J4 COBOL	DON SCHRICKER WANG LABS. 617-967-7628	LEMUEL SKIDMORE SKIDMORE RES. 203-669-7126	PEGGY BEARD NCR CORP. 619-693-5730	VICKI ECKELS TEXAS INST. 617-967-7628	PEGGY BEARD NCR CORP. 619-693-5730	97/22
X3J7 APT	WICKHAM LOH IBM CORP. 213-312-5926	C. W. WILSON MARTIN MARIETTA	RICHARD CAMPBELL APPL. SCHLUMBERGER 214-386-0157	ELLIOT BREBNER SPERRY CORP.	VACANT	184/SC3
X3J7.1 PROCESSOR LANGUAGES	VACANT	VACANT	N/A	N/A	VACANT	
X3J7.2 POSTPROCESSOR LANGUAGES	VACANT	VACANT	N/A	N/A	VACANT	
X3J7.3 LATHE LANG.	VACANT	VACANT	N/A	N/A	VACANT	
X3J7.4 ROBOTICS LANGUAGE	VACANT	VACANT	N/A	N/A	VACANT	
X3J9 PASCAL	K. ZEMROWSKI TRW 703-876-8014	MICHAEL HAGERTY RADIONICS 408-757-8877	PAULA SCHWARTZ RES. LIBR. GRP. 415-329-3534 X234	DAVID ROBBINS GTE CORP.	V. FOLWARCZNY NCR CORP. 614-439-0516	97/22
X3J10 APL	VACANT	G. H. FOSTER SYRACUSE UNIV. 315-423-4375	VACANT	PETER LUSTER 703-525-8001	VACANT	97/22
X3J11 C LANGUAGE	JIM BRODIE MOTOROLA 602-438-3456	THOMAS PLUM PLUM HALL INC. 609-927-3770	*STEVE HERSEE LATTICE, INC. 312-858-7950	ANDREW JOHNSON PRIME 617-879-2960 X4045	P.J. PLAUGHER WHITESMITHS, LTD. 617-369-8499	
X3J12 DIBOL	ELI SZKLANKA TEC COMPUTER SYS. 617-964-3890	KENNETH LIDSTER D.I.S.C. INC. 916-635-7300	VACANT	K. SCHILLING MCBA 818-242-9600	KENNETH BEERS DEC 603-884-0889	
X3J13 COMMON LISP	*ROBERT MATHIS 703-425-5923	VACANT	VACANT	VACANT	VACANT	97/21
<u>K - DOCUMENTATION</u>						
X3K1 COMPUTER DOCUMENTATION	VALERIE WINKLER U.S. NAVY	JOHN HACKNEY INTSTATE. ELEC. 714-635-7210	ALBRECHT NEUMANN NBS 301-975-3338	ALBRECHT NEUMANN NBS 301-975-3338	VACANT	97/7
X3K5 VOCAB. FOR INFO PROC. SYS	JOHN WOOD IBM CORP. 919-254-0182	HELMUT THIESS 202-265-9439	RICHARD BATEY SPERRY CORP. 215-542-2709	VACANT	ROY MULLINAX HOUS. & URBAN 703-755-5607	97/1

NOTE: WHERE THE SMC APPOINTMENT BALLOT IS NOT COMPLETE OR VOLUNTEERS HAVE NOT SUBMITTED THE NECESSARY DOCUMENTATION, THE OFFICER POSITION HAS BEEN LISTED AS ACTING (*) OR "VACANT"

** SMC ELECTED OFFICER

*** CORRESPONDING INTERNATIONAL STANDARDS COMMITTEE

128

<u>X3TC</u>	<u>**CHAIR</u>	<u>**VICE CHAIR</u>	<u>**INT'L REP.</u>	<u>VOCAB. REP.</u>	<u>SECRETARY</u>	<u>***INT'L</u>
<u>T & V - SYSTEMS TECHNOLOGY</u>						
X3T1 DATA ENCRYPTION	STEPHEN LEVIN SOFTSEL INC. 609-683-1150	GEOFFREY TURNER BANK OF AMERICA	ROBERT ELANDER IBM CORP. 914-385-6692	STEPHEN LEVIN SOFTSEL INC. 609-683-1150	VACANT	97/20
X3T2 DATA INTER- CHANGE	L. J. GALLAGHER NBS 301-975-3251	VACANT	VACANT	VACANT	VACANT	
X3T5 OSI	JERROLD FOLEY EDS CORP. 313-524-8416	J. ASCHENBRENNER IBM CORP. 919-254-0299	J. ASCHENBRENNER IBM CORP. 919-254-0299	WILLIAM MATHEWS NCR CORP. 619-452-1020	FRAN NIELSEN NBS 301-975-3365	97/21
X3T5.1 OSI ARCHI.	JOHN DAY CODEX CORP. 617-364-2000 X7111	JOHN GURZICK M/A-COM DCC, INC. 301-258-8940	N/A	N/A	VACANT	97/21/1
X3T5.4 OSI MGMT. PROTOCOLS	WILL COLLINS CODEX CORP. 617-364-2000	MARK KLERER AT&T BELL LABS. 201-949-8645	N/A	N/A	VACANT	
X3T5.5 APPLI. & PRES. LAYERS	L. L. HOLLIS IBM CORP. 919-254-0292	*PAUL BARTOLI AT&T TECH. 201-949-5965	N/A	N/A	PAT MELZER NCR CORP. 619-693-5591	97/21
X3T9 I/O INTERFACE	DEL SHOEMAKER DEC 202-383-5622	BILL BURR NBS 301-975-2914	PATRICK LANNAN AMP INC. 904-686-4476	ARNOLD ROCCATI EG&G WASH. AN. 301-840-3277	BILL BURR NBS 301-975-2914	97/13
X3T9.2 LOWER LEVEL INTERFACE	BILL BURR NBS 301-975-2914	JOHN LOHMEYER NCR CORP. 316-688-8000	N/A	N/A	VACANT	
X3T9.3 DEVICE LEVEL INTERFACE	GARY ROBINSON DEC 617-493-4954	VACANT	N/A	N/A	VACANT	
X3T9.5 LOCAL DIST. DATA INTERFACE	GENE MILLIGAN MAG. PERIPH. 405-524-3070	FLOYD ROSS UNISYS 215-542-2518	N/A	N/A	VACANT	
X3T9.6 CARTRIDGE TAPE DRIVES	LOUIS DOMSHY ARCHIVE CORP. 714-641-0279	VACANT	N/A	N/A	VACANT	
X3V1 TEXT: OFFICE & PUBLISHING SYS.	L. M. COLLINS IBM CORP. 214-556-7690	JAMES MASON OAK RID. LAB. 615-574-6973	VACANT	JONATHAN GRUDIN WANG LABS. 617-967-6135	CHARLES REEVES ASSOC. OF INFO. 615-574-2342	97/18
X3V1.1 USER REQ.: M.S.T.	VACANT	W. S. HOBGOOD IBM CORP.	N/A	N/A	JONATHAN GRUDIN MCC 512-338-3615	
X3V1.3 DOCUMENT ARCHITECTURE	ROY PIERCE XEROX CORP. 214-420-4326	HERMAN SILBIGER AT&T TECH. 201-898-2360	N/A	N/A	JAMES MASON OAK RIDGE LAB. 615-574-6973	
X3V1.4 TEXT INTERCHNG	R. H. CHRISTIE CONTROL DATA 612-482-6689	CAROL MOLLEN NCR CORP. 612-638-7777	RONALD HARVEY HONEYWELL 602-861-4711	CAROL MOLLEN NCR CORP. 612-638-7777	ROBERT SAMUELL BELLSOUTH SER. 404-529-7246	
X3V1.5 CONTENT ARCHI.	VACANT	VACANT	N/A	N/A	LYNNE ROSENTHAL NBS 301-975-3353	
X3V1.8 TEXT DESCR. & PROC. LANGS.	BILL DAVIS INTERNAL REV. 202-566-6533	LAWRENCE BECK GRUMMAN DATA 516-682-8478	N/A	N/A	VACANT	
X3V1.9 USER SYS. INTF. & SYMBOLS	VACANT	HELEN WALKER IBM CORP. 512-838-7861	N/A	N/A	VACANT	

NOTE: WHERE THE SMC APPOINTMENT BALLOT IS NOT COMPLETE OR VOLUNTEERS HAVE NOT SUBMITTED THE NECESSARY DOCUMENTATION, THE OFFICER POSITION HAS BEEN LISTED AS ACTING (*) OR "VACANT"
 ** SMC ELECTED OFFICER
 *** CORRESPONDING INTERNATIONAL STANDARDS COMMITTEE

MICHAEL DEESE, X3B11-IR
CHEROKEE DATA SYSTEMS
SUITE H
1880 SOUTH FLATIRONS COURT
BOULDER CO 80301
303-530-0618

DONALD DEUTSCH, X3H2-CH
GENERAL ELECTRIC COMPANY
GEISCO
278 FRANKLIN ROAD, SUITE 300
BRENTWOOD, TN 37027
615-371-6038

PHILLIP H. DIAMOND, X3L8.5-CH
U.S. VETERANS ADMINISTRATION
810 VERMONT AVE., N.W.
ROOM 202B
WASHINGTON DC 20420
202-233-2815

LOUIS C. DOMSHY, X3T9.6-CH
ARCHIVE CORPORATION
1650 SUNFLOWER AVENUE
COSTA MESA, CA 92626
714-641-0279

ALICE DROOGAN, X5B10-VC
MASTERCARD INTERNATIONAL
886 7TH AVENUE
NEW YORK, NY 10106
212-957-2498

VICKI ECKELS, X3J4-VR
TEXAS INSTRUMENTS
P.O. BOX 2909, M/S 2076
AUSTIN, TX 78769

ROBERT ELANDER, X3T1-IR
IBM CORPORATION
KINGSTON, NY 12401
914-385-6692

W. F. EMMONS, X3S3-CH
IBM CORPORATION X3S3.1-CH
SYSTEMS COMMUNICATIONS DIV.
P.O. BOX 12195 C71/664
RES. TRIANGLE PK NC 27709-2195
919-254-0294

JERROLD S. FOLEY, X3T5-CH
ELECTRONIC DATA SYSTEMS CORP.
300 EAST BIG BEAVER ROAD
P.O. BOX 7019, CUBE 5176
TROY, MI 48083
313-524-8416

ROBERT H. FOLLETT, SC22-VC
IBM CORPORATION
643/RCTR/9E
6705 ROCKLEDGE DRIVE
BETHESDA, MD 20817
301-564-2108

VICTOR POLWARCZNY, X3J9-SY
MCR CORPORATION
E & M CAMBRIDGE
800 COCHRAN AVENUE
CAMBRIDGE, OH 43725-0523
614-439-0516

ELIZABETH FONG, DBSSG-SY & IR
NATIONAL BUREAU OF STANDARDS
BUILDING 225, ROOM A258
GAITHERSBURG MD 20899
301-975-3250

G. H. FOSTER, X3J10-VC
SYRACUSE UNIVERSITY
DEPT. OF ELEC. & COMPUTER ENG.
111 LINK HALL
SYRACUSE, NY 13244-1240
315/423-4375

JEAN B. FREEDMAN, X3B11-VR
NATIONAL BUREAU OF STANDARDS
BUILDING 225, ROOM A49
GAITHERSBURG MD 20899
301-975-2922

LEONARD J. GALLAGHER, X3H2-IR
NAT'L BUR. OF STDS. X3T2-CH
BUILDING 225, ROOM A156
GAITHERSBURG, MD 20899
301-975-3251

RICHARD GIBSON, X3-VC
AT&T COMMUNICATIONS
ROOM 5A 211
ROUTE 202 & 206N
BEDMINSTER NJ 07921
201-234-3795

MICHAEL M. GORMAN, X3H2-SY
WHITEMARSH INFO. SYSTEMS
2008 ALTHEA LANE
BOWIE MD 20716
301/245-1142

GEORGE GRINSTEIN, X3H3.6-CH
UNIVERSITY OF LOWELL
GRAPHICS RESEARCH LABORATORY
ONE UNIVERSITY AVENUE
LOWELL, MA 01854
617-452-5000

JONATHAN GRUDIN, X3V1.1-SY
MCC
P.O. BOX 200195
AUSTIN, TX 78720
512-338-3615

JOHN GURZICK, X3T5.1-VC
M/A-COM DCC, INC.
1350 PICCARD DRIVE, SUITE 400
ROCKVILLE MD 20850
301/258-8940

JOHN HACKNEY, X3K1-VC
INTERSTATE ELECTRONICS
P. O. BOX 3117
ANAHEIM CA 92803
714/635-7210

MICHAEL HAGERTY, X3J9-VC
RADIONICS
27911 BERNICK DRIVE
CARMEL, CA 93923
408-757-8877

RONALD B. HARVEY, X3V1.4-IR
HONEYWELL INFORMATION SYSTEMS
P.O. BOX 8000, M/S AZ13-H32
PHOENIX, AZ 85066-8000
602-861-4711

STEVE HERSEE, X3J11-IR
LATTICE, INC.
P. O. BOX 3072
GLEN ELLYN IL 60138
312-858-7950

W. S. HOBGOOL, X3V1.1-VC
IBM CORPORATION
P.O. BOX 12195
J15/662
RES. TRIANGLE PARK, NC 27709

MICHAEL HOGAN, X3B5-IR
NAT'L BUR. OF STDS. X3B6-IR
BUILDING 225, ROOM A59
GAITHERSBURG MD 20899
301-975-2926

L. L. HOLLIS, X3T5.5-CH
IBM CORPORATION
E98/002-1W
P. O. BOX 12195
RES. TRIANGLE PARK NC 27709
919-254-0292

PAUL D. JAHNKE, X3B7-VR
3M COMPANY
3M CENTER BLDG. 236-1
ST. PAUL MN 55144
612/736-0117

ANDREW JOHNSON, X3J11-VR
PRIME COMPUTER, INC.
MS 10C17-3
500 OLD CONNECTICUT PATH
FRAMINGHAM MA 01701
617/879-2960 X 4045

JIM JOHNSON, X3B10.1-SY
DATA CARD CORP.
11111 BREN ROAD WEST
MINNETONKA, MN 55440
612-933-1223

CAROL D. JOYCE, X3H2-VC
RELATIONAL TECHNOLOGY INC.
1080 MARINA VILLAGE PARKWAY
ALAMEDA, CA 94501
415-765-1400

CATHERINE A. KACHURIK, IAC-SY
X3 SECRETARIAT/CBEMA X3-SY
311 FIRST STREET, NW
SUITE 500
WASHINGTON, DC 20001-2178
202-737-8888

WILLIAM H. KENWORTHY, JR., X3L6-CH & VR
U.S. DEPT. OF DEFENSE
C/O 420 KIMBLEWICK DRIVE
SILVER SPRING, MD 20904
202/694-3361

KAREN KIRKBRIDE, X3L8-IR
WASH. HDQRTS. SER. X3L8.6-CH
DIR. OF INFO. OPER. & REPORTS
C/O 4204 SANDHURST COURT
ANNANDALE, VA 22003
202/746-0797

ROBERT A. KITCHENER, X3B10-CH
LOGICARD SYSTEMS, INC.
P.O. BOX 637
ARMONK, NY 10504
914-273-8734

JOHN C. KLENSIN, X3J1-CH
MIT
77 MASSACHUSETTS AVENUE
ROOM 20A-226
CAMBRIDGE, MA 02139
617/253-1355

MARK KLERER, X3T5.4-VC
AT&T BELL LABORATORIES
NETWORK MANAGEMENT STANDARDS
CRAWFORDS CORNER ROAD
HOLMDEL, NJ 07733
201-949-8645

DAVID M. PISCITELLO, X3S3.3-VC
BURROUGHS CORPORATION
SDG/DEVON ENGINEERING
P. O. BOX 1874
SOUTHEASTERN PA 19398
215-341-4642

P.J. PLAUGER, X3J11-SY
WHITESMITHS, LTD.
59 POWER ROAD
WESTFORD, MA 01886
617/369-8499

THOMAS PLUM, X3J11-VC
PLUM HALL INC.
1 SPRUCE AVENUE
CARDIFF NJ 08252
609/927-3770

JOSEPH PODVOJSKY, X3S3.7-SY
MITRE CORPORATION
MS B270
P. O. BOX 208
BEDFORD MA 01730
617-271-2155

WILLIAM B. POLAND, JR., X3B6-CH
NASA/GODDARD SPACE FLIGHT CTR.
CODE 730.4
GREENBELT MD 20771
301-286-8592

THOMAS POWERS, X3H3.3-VC
DIGITAL EQUIPMENT CORPORATION
ML5-3/E12
146 MAIN STREET
MAYNARD MA 01754
617-493-2704

BILL PROCTOR, X3B5-VC
MEMOREX CORPORATION
2400 CONDENSE WAY MS 23-74
SANTA CLARA CA 95052
408-987-0684

CHARLES REEVES, JR., X3V1-SY
ASSOC. OF INFORMATION
SYSTEMS PROFESSIONALS
10812 DINEEN DRIVE
KNOXVILLE TN 37922
615-574-2342

STAN REYNOLDS, X3B6-IF
SANDIA NATIONAL LABORATORIES
DIV. 7535
P.O. BOX 5800
ALBUQUERQUE, NM 87185

WILLIAM C. RINEHULS, SPARC-CH
GENERAL SERVICES ADMN.
ADTS
8457 RUSHING CREEK COURT
SPRINGFIELD VA 22153-2532
202/566-1180

DAVID C. ROBBINS, X3J9-VR
GTE CORP.
GTE LABORATORIES
40 SYLVAN ROAD
WALTHAM, MA 02254

GARY S. ROBINSON, IAC-VC
DIG. EQUIP. CORP. X3T9.3-CH
146 MAIN STREET MLO12-B/E51
MAYNARD MA 01754-2572
617/493-4094

ARNOLD J. ROCCATI, X3B5-VR
EG&G WASH. ANALY. X3T9-VR
SERVICES CENTER, INC.
1396 PICCARD DRIVE
ROCKVILLE, MD 20850-4308
301-840-3277

LEROY RODGERS, SPARC-VC
DIGITAL EQUIPMENT CORPORATION
146 MAIN STREET MLO5-5/T58
MAYNARD, MA 01754-2572
617/493-3163

LYNNE ROSENTHAL, X3V1.5-SY
NATIONAL BUREAU OF STANDARDS
BUILDING 225, ROOM B257
GAITHERSBURG, MD 20899
301-975-3353

FLOYD E. ROSS, X3T9.5-VC
SPERRY CORP.
P.O. BOX 500
M/S E8-106
BLUE BELL, PA 19424
215/542-2518

JOHN RUSSELL, X3L2-VR
CONTROL DATA CORPORATION
8100 34TH AVE., SO. BMO3M
MINNEAPOLIS MN 55440
612/853-5414

ROBERT L. SAMUELL, III, X3V1.4-SY
BELLSOUTH SERVICES, INC.
40V93 SOUTHERN BELL CENTER
675 WEST PEACHTREE ST., N.E.
ATLANTA, GA 30375
404-529-7246

ANTHONY J. SARATORA, X3B7-SY
BASF SYSTEMS, INC.
CROSBY DRIVE
BEDFORD MA 01730
617/271-4255

KENNETH M. SCHILLING, X3J12-VR
MCBA
425 WEST BROADWAY
GLENDALE, CA 91204
818-242-9600

DON SCHRICKER, X3J4-CH
WANG LABORATORIES, INC.
M/S 013-790
ONE INDUSTRIAL AVENUE
LOWELL MA 01851
617-967-7628

STEVEN A. SCHRIER, X3V1-VR
BOOZ, ALLEN & HAMILTON, INC.
4330 EAST-WEST HIGHWAY
BETHESDA, MD 20814
202-767-4975

PAULA SCHWARTZ, X3J9-IR
RESEARCH LIBRARIES GROUP INC.
852 LA PARA AVENUE
PALO ALTO, CA 94306
415-329-3534

DUNHAM B. SEELEY, X3B9-SY
THE STANDARD REGISTER COMPANY
1150 UNIVERSITY AVE., BOX 910
ROCHESTER, NY 14603
716-271-3400 X 2407

D. L. SEIGAL, X3B10-IR
AMERICAN EXPRESS
TRAVEL RELATED SERVICES
1647 E. MORTEN AVENUE
PHOENIX, AZ 85020
602-995-7837

NEAL B. SEITZ, X3S3.5-CH
U.S. DEPT. OF COMMERCE
NTIA/ITS-N3
325 BROADWAY
BOULDER CO 80303
303/497-3106

DELBERT L. SHOEMAKER, SMC-VC
DIG. EQUIP. CORP. X3T9-CH
1331 PENNSYLVANIA AVE. N.W.
SIXTH FLOOR
WASHINGTON, DC 20004
202-383-5622

HERMAN R. SILBIGER, X3V1.3-VC
AT&T TECHNOLOGIES
ROOM 3L-603
CRAWFORD CORNERS ROAD
HOLMDEL, NJ 07733
201/896-2360

LEMUEL SKIDMORE, X3J4-VC
SKIDMORE RESOURCE MNGMT. CO.
20 PEPPERBUSH DRIVE
CLINTON CT 06413
203-669-7126

MADELEINE SPARKS, X3H3-VR
UNISYS X3H3.4-CH
4810 BRADFORD BLVD., N. W.
HUNTSVILLE AL 35805
205/837-7610 X 247

JOHN STEARNS, X3B10-SY
DATA CARD CORP.
11111 BREN ROAD WEST
MINNETONKA, MN 55440
612-933-1223

RICHARD STEINBRENNER, X3B5-CH
AT&T TECHNOLOGIES
C/O AT&T BELL LABORATORIES
ONE WHIPPANY DRIVE
WHIPPANY, NJ 07931
201/386-7053

EDWARD L. STULL, DBSSG-CH & VR
GTE GOVERNMENT SYSTEMS
1700 RESEARCH BOULEVARD
ROOM 3095
ROCKVILLE, MD 20850
301-294-8649

MARY SUMMERS, X3L8.7-VC
DEFENSE INTELLIGENCE AGENCY
ATTN: RSE-1A
WASHINGTON, DC 20340-3466
202-373-3007

ELI SZKLANKA, X3J12-CH
TEC COMPUTER SYSTEMS
30 TOWER ROAD, M/S MK02-1/H10
NEWTON, MA 02164
617-964-3890

DONALD J. THELEN, X3L2-CH
AT&T TECHNOLOGIES X3L2.1-CH
C/O AT&T INFORMATION SYSTEMS
60 COLUMBIA TURNPIKE, A-B210
MORRISTOWN, NJ 07960
201/644-3728

OFFICER APPOINTMENT & TERM EXPIRATION DATE

XJTC	OFF. POS.	NAME	DATE EFFECTV. YY/MM/DD	TICKLER NOTIF. YY/MM/DD	TERM EXPIRATION YY/MM/DD	XJTC	OFF. POS.	NAME	DATE EFFECTV. YY/MM/DD	TICKLER NOTIF. YY/MM/DD	TERM EXPIRATION YY/MM/DD
X3	VC	RICHARD GIBSON	85/09/27	N/A	N/A	X3B11	CH	JOSEPH ZAJACZKOWSKI	84/11/30	87/08/30	87/11/30
SMC	VC	DELBERT SHOEMAKER	85/06/10	88/01/01	88/06/10	X3B11	VC	ROBERT BENDER	84/12/10	87/09/10	87/12/10
IAC	CH	JOSEPH DEBLASI	84/12/10	87/06/10	87/12/10	X3B11	IR	MICHAEL UESE	84/12/15	87/09/15	87/12/15
IAC	VC	GARY ROBINSON	86/01/06	88/09/06	89/01/06	X3H2	CH	DONALD DEUTSCH	84/05/20	87/01/20	87/05/20
SPARC	CH	WILLIAM RINGHOLS	85/04/12	88/01/01	88/04/12	X3H2	VC	CAROL JOYCE	86/10/21	89/05/21	89/10/10
SPARC	VC	LEROY RODGERS	82/08/10	85/04/08	85/08/10	X3H2	IR	LEONARD GALLAGHER	00/00/00	00/00/00	00/00/00
DBSSG	CH	EDWARD STULL	00/00/00	00/00/00	00/00/00	X3H3	CH	PETER BONO	84/05/04	87/01/04	87/05/04
DBSSG	VC	NANCY McDONALD	85/06/25	88/03/25	88/06/25	X3H3	VC	JOHN MCCONNELL	86/11/15	89/07/15	89/11/15
DBSSG	IR	ELIZABETH FONG	85/09/18	88/04/18	88/09/18	X3H3	IR	JANET CHIN	82/12/18	85/07/18	85/12/18
SC7 TAG	CH	ALBRECHT NEUMANN	86/04/15	89/01/15	89/04/15	X3H3.1	CH	DAVID BAILEY	86/01/06	88/09/06	89/01/06
SC7 TAG	VC	RICHARD WERLING	86/07/01	89/04/01	89/07/01	X3H3.3	VC	THOMAS POWERS	85/06/07	88/01/01	88/06/07
SC21 TAG	CH	J. ASCHENBRENNER	84/11/30	87/06/30	87/11/30	X3H3.4	CH	MADELINE SPARKS	84/12/15	87/09/15	87/12/15
SC21 TAG	VC	HELEN WOOD	84/12/15	87/06/15	87/12/15	X3H3.4	VC	GERALDINE CUTHBERT	85/06/07	88/01/01	88/06/07
SC22 TAG	CH	HELEN WOOD	85/04/15	88/04/15	88/04/15	X3H3.5	CH	MARY MILLER	87/01/22	89/10/22	90/01/22
SC22 TAG	VC	ROBERT FOLLETT	86/03/05	89/01/05	89/03/05	X3H3.6	CH	GEORGE GRINSTEIN	87/02/13	89/12/13	90/02/13
X3A1	CH	ROGER MONELL	84/06/05	87/01/05	87/06/05	X3H4	CH	ANTHONY WINKLER	84/05/30	87/03/30	87/05/30
X3A1	VC	ROBERT BLOSS	84/09/06	87/04/05	87/09/06	X3H4	IR	DAVID THOMAS	84/05/04	87/03/04	87/05/04
X3A1	IR	JOHN McDONNELL	84/09/01	87/04/01	87/09/01	X3J1	CH	JOHN KLENSIN	82/03/22	85/01/22	85/03/22
X3A1.1	CH	PATRICK TRAGLIA	85/03/20	88/01/20	88/03/20	X3J1	VC	CHARLES NYLANDER	83/10/15	86/07/15	86/10/15
X3A1.2	CH	CARL KNOEDEL	84/09/10	87/04/10	87/09/10	X3J2	CH	JAMES TOTTON	84/05/04	87/02/04	87/05/04
X3A1.3	CH	C. BISS	84/05/04	87/01/04	87/05/04	X3J2	VC	ANDREW KLOSSNER	86/11/20	89/07/20	89/11/20
X3B5	CH	RICHARD STEINBRENNER	00/00/00	00/00/00	00/00/00	X3J3	CH	JEANNE ADAMS	84/05/04	87/02/04	87/05/04
X3B5	VC	SAM CHEATHAM	87/01/29	90/01/29	90/01/29	X3J3	VC	JERROLD WAGENER	86/06/13	89/03/13	89/06/13
X3B5	IR	MICHAEL HOGAN	00/00/00	00/00/00	00/00/00	X3J4	CH	DON SCHRICKER	85/09/23	88/06/23	88/09/23
X3B6	CH	WILLIAM POLAND, JR.	84/05/04	87/02/04	87/05/04	X3J4	VC	LEMUEL SKIDMORE	84/07/01	87/04/01	87/07/01
X3B6	IR	STAN REYNOLDS	00/00/00	00/00/00	00/00/00	X3J4	IR	PEGGY BEARD	84/09/01	87/04/01	87/09/01
X3B7	CH	BILL CARLSON	85/07/20	88/03/20	88/07/20	X3J7	CH	WICKHAM LOH	87/02/13	89/12/13	90/02/13
X3B7	IR	BILL CARLSON	85/06/20	88/03/20	88/06/20	X3J7	VC	C. WILSON	87/02/13	89/12/13	90/02/13
X3B8	CH	JAMES BARNES	85/09/09	88/06/06	88/09/09	X3J7	IR	RICHARD CAMPBELL	82/09/13	85/06/13	85/09/13
X3B8	VC	BILL PROCTOR	86/03/31	89/01/01	89/03/31	X3J9	CH	KENNETH ZEHROWSKI	85/11/02	86/08/12	86/11/02
X3B8	IR	MICHAEL HOGAN	84/05/04	87/01/04	87/05/04	X3J9	VC	MICHAEL HAGERTY	85/06/10	88/01/01	88/06/10
X3B9	CH	D. ODDY	85/06/07	88/01/01	88/06/07	X3J9	IR	PAULA SCHWARTZ	86/01/06	88/09/06	89/01/06
X3B10	CH	ROBERT KITCHNER	86/01/06	88/09/06	89/01/06	X3J10	VC	G. FOSTER	00/00/00	00/00/00	00/00/00
X3B10	VC	ALICE DRUGGAN	86/01/06	88/09/06	89/01/06	X3J11	CH	JIM BRODIE	84/01/02	86/09/02	87/01/02
X3B10	IR	D. SEIGAL	86/01/06	88/09/06	89/01/06	X3J11	VC	THOMAS PLUM	84/01/02	86/09/02	87/01/02
X3B10.4	CH	ROBERT CALLEN	86/06/25	89/03/25	89/06/25	X3J11	IR	STEVE HERSEE	86/01/06	88/09/06	89/01/06

12

SPARC TC/SG NATIONAL AND INTERNATIONAL MONITORING
REPORTING DATE - 86

<u>COMMITTEE</u>	<u>MEMBER</u>	<u>REPORTING DATE - 86</u>
X3A1	Mr. Fenner	January
*X3B5/SC11	Mr. Fogle	July
X3B6/WG2	Mr. Bass	November
X3B7	Mr. Rountree	January
*X3B8/SC11	Mr. Kurihara	January
X3B9	Mr. Bass	July
X3B10/SC17	Mr. Bass	July
X3B11/SC23	Mr. Follett	July
X3H2	Mr. LaPlant	March
X3H3	Mr. Fogle	November
X3H4	Mr. McNamara	March
X3J1	Mr. Ryland	September
X3J2	Mr. Kurihara	September
X3J3	Mr. LaPlant	May
X3J4	Mr. McNamara	September
X3J7	Mr. Ryland	January
X3J9	Mr. Kurihara	March
X3J10	Mr. Fogle	July
X3J11	Mr. Kurihara	May
X3J12	Mr. LaPlant	September
X3J13	Mr. LaPlant	September
X3K1	Ms. Butler	November
X3K5/SC1	Ms. Butler	November
X3L2/SC2	Mr. Haines	July
X3L5/SC15	Mr. Follett	November
X3L8/SC14	Mr. McNamara	November
X3S3/SC6	Mr. Follett	March
X3T1/SC20	Mr. Rountree	July
X3T2	Mr. Follett	
X3T5	Mr. Fenner	March
X3T9/SC13	Mr. Ryland	March
X3V1/SC18	Mr. McNamara	September
SPARC/DBSSG	Ms. Butler	March
PLIP	(Inactive)	
U.S. TAG TO SC7	Ms. Butler	November
U.S. TAG TO SC21	Mr. Follett	March
U.S. TAG TO SC22	Ms. Butler	March

*SC11 liaison is conducted through reporting to both TC's.

PROJECT LIAISON

<u>PROJECT</u>	<u>MEMBER</u>	<u>LIAISON WITH</u>
Ada	Mr. Virtue	DOD HOLWG
MUMPS	Mr. LaPlant	MDC

SPARC LIAISON ACTIVITIES WITH EXTERNAL ORGANIZATIONS

<u>ORGANIZATION</u>	<u>MEMBER</u>
Information Systems Standards Board (ISSB)	Mr. Rinehuls
ANSI Planning Panel on Industrial Automation	Mr. Rodgers
IEEE Computer Society Standards Activities Board	Mr. LaPlant
X9 - Financial Services	Mr. Fenner
X12 - Electronic Business Data Interchange	Mr. Rountree
IEC/TC83 Working Group on Planning and Requirements	Mr. McNamara

IAC International Liaison Assignments

ISO/TC97 - Chair	ISO/TC65 Banking - Marv Bass
WG2 - Marv Bass	TC145 Graphics - Mary Anne Gray
SC1 - Gary Haines	TC154 Data Elements - Bob Rountree
2 - Bob Rountree	TC159 Ergonomics - Gary Robinson
6 - John Wheeler	
7 - Bob Rountree*	ISO/TC184 Ind'l Automation - Pat Lannan
11 - Bob Rountree	IEC/TC3 Graphics - Sava Sherr
13 - Patrick Lannan	SC47B Microprocessors - Sava Sherr
14 - Bob Rountree	TC74 Product Safety - Marv Bass
15 - Mary Anne Gray	TC77 EMI - Pat Lannan
17 - Marv Bass	TC86 Fiber Optics - Pat Lannan
18 - Mary Anne Gray	ACOS Sava Sherr
20 - Gary Robinson	CCITT/SGVII Networks - Richard Gibson
21 - Gary Robinson	VIII Terminals - Mary Anne Gray
22 - Bob Rountree	XVII Data Com. - Richard Gibson
23 - Gary Haines	XVIII Dig. Networks - Bob Rountree
	IEC/TC83 Info. Tech. Eq't. - Sava Sherr

- ECMA - Gary Robinson
- IEC - Sava Sherr
- CCITT - Richard Gibson
- ISO - Joseph DeBlasi

DETAILED X3 TECHNICAL COMMITTEE ORGANIZATIONAL CHART

X3 Technical Committee	Technical Advisory Group (TAG) Responsibility	X3 Technical Committee	Technical Advisory Group (TAG) Responsibility
U.S. TAG to SC7	ISO/TC97 SC7	H & J - LANGUAGES (CONTINUED)	
U.S. TAG to SC21	ISO/TC97 SC21	X3J12 DIBOL	
U.S. TAG to SC22	ISO/TC97 SC22	X3J13 COMMON LISF	
A - RECOGNITION		K - DOCUMENTATION	
X3A1 OCR & MICF		X3K1 COMPUTER DOCUMENTATION	ISO/TC97 SC7
X3A1.1 FONT DESIGN		X3K5 VOCAB. FOR INFO. PROC. SYS.	ISO/TC97 SC1
X3A1.2 OCR SUPPLIES & FORMS			
X3A1.3 IMAGE DEF. & MEASUREMENT		L - DATA REPRESENTATION	
B - MEDIA		X3L2 CODES & CHARACTER SETS	ISO/TC97 SC2
X3B5 DIGITAL MAGNETIC TAPE	ISO/TC97 SC11	X3L2.1 VIDEOTEX/TELETEXT	
X3B6 INSTRUMENTATION TAPE	ISO/TC97 WG2	X3L2.2 ADD. CONTR. FUNCTS. FOR X3.64	
X3B7 MAGNETIC DISKS		X3L2.3 TWO-BYTE GRA. CHARACTER SET	ISO/TC97 SC2 WG2
X3B8 FLEXIBLE DISK CARTRIDGES	ISO/TC97 SC11	**X3L5 LABELS & FILE STRUCTURE	ISO/TC97 SC15
X3B8.1 TRACK FORMATS FOR FDC'S		X3L8 DATA REPRESENTATION	ISO/TC97 SC14
X3B5 PAPER FORMS/LAYOUTS		X3L8.4 GEOGRAPHIC UNITS	
X3B10 CREDIT/ID CARDS	ISO/TC97 SC17	X3L8.5 ATTRIBUTES OF DATA ELEMENTS	
X3B10.1 INTEGRATED CIRCUIT CARDS	ISO/TC97 SC17 WG4	X3L8.6 CLASSIF. OF DATA ELEMENTS	
X3B10.2 REVISION OF X4.18		X3L8.7 MNEM. CODES FOR DATA ELEM.	
X3B10.3 MIN. PHYS. REQ'TS/SAVINGSBOOKS		S - COMMUNICATION	
X3B10.4 OPTICALLY ENCODED CARD MEDIA		X3S3 DATA COMMUNICATION	ISO/TC97 SC6
X3B11 OPTICAL DIGITAL DATA DISKS	ISO/TC97 SC23	X3S3.1 DATA COMM. PLANNING	
H & J - LANGUAGES		X3S3.2 DATA COMM. VOCABULARY	
X3H2 DATABASE	ISO/TC97 SC21 WG3	X3S3.3 NETWORK LAYER	ISO/TC97 SC6 WG2
X3H3 COMPUTER GRAPHICS	ISO/TC97 SC21 WG2	X3S3.4 CONTROL PROCEDURES	
X3H3.1 CORE GRAPHICS SYSTEMS		X3S3.5 COMM. SYS. PERFORMANCE	
X3H3.2 REFERENCE MODELS		X3S3.7 PUB. DATA NETWORK ACCESS	
X3H3.3 VIRTUAL DEVICE INTERFACE		T & V - SYSTEMS TECHNOLOGY	
X3H3.4 CONFORMANCE & BINDING		X3T1 DATA ENCRYPTION	ISO/TC97 SC20
X3H3.5 MIN. INTERFACE TO GRPHCS.		X3T2 DATA INTERCHANGE	
X3H3.6 DISF. MGMT. GRPHICL DEV.		X3T5 OSI	ISO/TC97 SC21
X3H4 INFO. RESOURCE & DICT.	ISO/TC97 SC21	X3T5.1 OSI ARCHITECTURE	ISO/TC97 SC21 WG1
X3H4.1 IRDS REFERENCE MODEL		X3T5.4 OSI MNGMT. PROTOCOLS	ISO/TC97 SC21 WG4
X3H4.2 IRDS EXTER. SFTWRE INTFCE		X3T5.5 APPLIC. & PRES. LAYERS	ISO/TC97 SC21 WG6
X3J1 PL/I	ISO/TC97 SC22	X3T9 I/O INTERFACE	ISO/TC97 SC13
X3J1.3 GENERAL PURPOSE SUBSET		X3T9.2 LOWER LEVEL INTERFACE	ISO/TC97 SC13 WG3
X3J2 BASIC	ISO/TC97 SC22 WG8	X3T9.3 DEVICE LEVEL INTERFACE	
X3J3 FORTRAN	ISO/TC97 SC22 WG5	X3T9.5 LOCAL DIST. DATA INTFC.	
X3J4 COBOL	ISO/TC97 SC22 WG4	X3T9.6 CARTRIDGE TAPE DRIVES	
X3J7 APT	ISO/TC97 SC3	X3V1 TEXT: OFF. & PUB. SYS.	ISO/TC97 SC18
X3J7.1 PROCESSOR LANGUAGES		X3V1.1 USER REQ. M.S.T.	ISO/TC97 SC18 WG1
X3J7.2 POSTPROCESSOR LANGUAGES		X3V1.3 DOCUMENT ARCHITECTURE	ISO/TC97 SC18 WG2
X3J7.3 LATHE LANGUAGES		X3V1.4 TEXT INTERCHANGE	ISO/TC97 SC18 WG4
X3J7.4 ROBOTICS LANGUAGE		X3V1.5 CONTENT ARCHITECTURE	ISO/TC97 SC18 WG5
X3J9 PASCAL	ISO/TC97 SC22 WG2	X3V1.8 TXT DESCR. & PROC. LANGS.	ISO/TC97 SC18 WG8
X3J10 APL	ISO/TC97 SC22 WG3	X3V1.9 USER SYS. INTFC & SYMBOLS	ISO/TC97 SC18 WG9
X3J11 C LANGUAGE			

** Recommendation for Disbandment is Under Consideration at this Time

This Chart is Organizational Only and Does not Reflect the Standards Approval Process

134

ISO/TC97/SC CHAIRS FOR US-HELD SECRETARIATS

TC97
Mr. L. John Rankine
Director, Standards & Data Security
IBM Corporation
2000 Purchase Street
Purchase, NY 10577-2597
914-697-7230

TC97/SC18
Mary Anne Gray
IBM Corporation
2000 Purchase Street
Purchase, NY 10577-2597
914-697-7224

TC97/SC66
Mr. Edward Lohse
Burroughs Corporation
240 Radnor Chester Road
Suite 100 A
215-964-5625

TC97/SC11
Mr. Joseph Zajaczkowski
Cherokee Data Systems
Suite H
1880 South Flatirons Court
Boulder, CO 80301
303-449-1239

TC97/SC21
Mr. Richard desJardins
ARPA/IPTO
P.O. Box 12368
Arlington, VA 22209
202-694-4001

WORKING GROUP CONVENORS

SC1/WG9 RICHARD W. BATEY
SPERRY CORPORATION
P.O. BOX 500 1C1-NE6
BLUE BELL, PA 19424
215-542-2709

SC2/WG6 MACKLIN W. BISHOP
IBM CORPORATION
11400 BURNET ROAD
BLDG. 803, DEPT. D34
AUSTIN, TX 78758
512-823-2005

SC6/WG1 DAVID E. CARLSON
AT&T BELL LABORATORIES
ROOM 3N 610
CRAWFORDS CORNER ROAD
HOLMDEL, NJ 07733
201-949-7503

SC17/WG3 CHARLES B. FISCHER
AIR TRANSPORT ASSOC.
1709 NEW YORK AVE., NW
WASHINGTON, DC 20006
202-626-4222

SC17/WG5 BERYL ANN BARBER
CHASE MANHATTAN BANK, N.A.
3 SHORTLANDS HAMMERSMITH
LONDON W68RZ UK

SC18/WG8 JAMES D. MASON
OAK RIDGE NAT'L LABORATORY
P.O. BOX X, 4500-S
OAK RIDGE, TN 37831
615-574-6973

SC20/WG3 JOSEPH TARDO
DIGITAL EQUIP. CORP.
550 KING STREET
LITTLETON, MA 01460-1289
617-486-7660

SC21/WG6 PAUL D. BARTOLI
AT&T INFO. SYSTEMS
ROOM 3M-622
CRAWFORDS CORNER ROAD
HOLMDEL, NJ 07733
201-949-5965

SC22/WG4 MABEL VICKERS
NATIONAL BUREAU OF STANDARDS
TECH. BLDG. 225, RM. A-367
GAITHERSBURG, MD 20899
301-975-3277

SC22/WG3 JEANNE MARTIN
LAWRENCE LIVERMORE LAB.
P.O. BOX 808L-300
LIVERMORE, CA 94550
415-422-3753

SC22/WG9 VIRGINIA CASTOR
AJPO, DIRECTOR
COMPUTER SOFTWARE & SYS.
1211 SOUTH FERN STREET
ROOM C107
ARLINGTON, VA 22202
202-694-0209

SC22/WG11 DONALD F. NELSON
TANDEM COMPUTERS, INC.
10555 RIDGEVIEW COURT
CUPERTINO, CA 95014
408-996-6340

SC22/WG14 STEVE HERSEE
LATTICE, INC.
P.O. BOX 3072
GLEN ELLYN, IL 60138
312-858-7950

X3/SD-7
FEBRUARY 1987
104(X)JCA-7

ACCREDITED STANDARDS COMMITTEE*

X3-INFORMATION PROCESSING SYSTEMS

MEETING SCHEDULE AND CALENDAR

* Operating under the procedures of the
American National Standards Institute

Secretariat:



Computer & Business Equipment

Manufacturers Association

TC97 & TC97 SC MEETINGS

157

<u>Meeting</u>	<u>YeMonDay</u>	<u>Place</u>	<u>City</u>	<u>Ref</u>
<u>ISO/TC 97 - INFORMATION PROCESSING SYSTEMS</u>				
TC97 Plenary	88/05/00	to be confirmed	Tokyo	97 N 1125
TC97 Advisory Grp	87/01/20-22	BSI	London	97 N 1763
TC97 SWG or Strategic Plan	87/01/19-20	BSI	London	97 N 1772
TC97 SWG Functional Std	87/03/09-13		Europe	97 N 1760
<u>SC 1 - VOCABULARY</u>				
TC97 SC01 Plenary	87/05/11-15	Quality Inn	Washington	97/1 N 1021
TC97 SC01 Plenary	88/06/00	to be determined	Berlin	TX 6/19/86
<u>SC 2 - CHARACTER SETS AND INFORMATION CODING</u>				
TC97 SC02 Plenary	87/03/16-20	DIN	Berlin	97/2 N 1700
TC97 SC02 Plenary	89/Sept	to be determined	Washington, DC	TX 8/29/86
TC97 SC02 WG02	87/03/Week 09-13	BSI (probably)	London	TX 8/29/86
TC97 SC02 WG04	87/03/Week 09-13	BSI (probably)	London	TX 8/29/86
<u>SC 6 - TELECOMMUNICATIONS AND INFORMATION EXCHANGE BETWEEN SYSTEMS</u>				
TC97 SC06 Plenary	88/01 or 88/02	to be determined	UK	97/6 N 4294
TC97 SC06 WG01	87/06/01-05	AFNOR	Paris	97/6 N 4294
TC97 SC06 WG01	88/01 or 88/02	to be determined	UK	97/6 N 4294
TC97 SC06 WG02	87/05/18-27	DIN	Berlin	97/6 N 4294
TC97 SC06 WG02	88/01 or 88/02	to be determined	UK	97/6 N 4294
TC97 SC06 WG03	87/04/23-29	Fragrant Hill Hotel	Beijing	97/6 N 4294
TC97 SC06 WG03	88/01 or 88/02	to be determined	UK	97/6 N 4294
TC97 SC06 WG04	87/05/11-14	Eigtveds Pakhus	Copenhagen	97/6 N 4294
TC97 SC06 WG04 LOTOS	87/01/19-21	Twente University of Tech	Enschede, NL	97/6/4 N185
TC97 SC06 WG04 Conf Test	87/02/11	Hotel Plaza	Nice	97/6/4 N 191
TC97 SC06 WG04	88/01 or 88/02	to be determined	UK	97/6 N 4294

134

TC97 SC17	WG4	87/07/01-03	Integrated Circuits: Cards	Tokyo, Japan	Letter d/d 11/24/86
TC97 SC17	WG5	87/04/07/08		Wash., U.S.A.	Letter d/d 11/24/86
TC97 SC17	WG7	87/04/09/10		Wash., U.S.A.	Letter d/d 11/24/86

SC 18 - TEXT AND OFFICE SYSTEMS

TC97 SC18	Plenary & HOD/C	87/05/04-08	Eigtveds Pakhus	Copenhagen	97/18 N 862
TC97 SC18	Plenary	86/05/01	to be determined		
TC97 SC18	WG01	87/01/12-16	Sheraton Sand Resort	Clearwater bch, FL	97/18 N 868
TC97 SC18	WG03,05,08	87/01/07/09	Sheraton Sand Key Resort	Clearwater Bch, FL	97/18 N 860
TC97 SC18	WG03	87/00/00	IBM	Austin, TX	97/18 N 694
TC97 SC18	WG03	87/07/06-10	to be determined	Paris	
TC97 SC18	WG03 SWG Format Spec	87/02/18-20	BSI	London	97/18/3 N 742
TC97 SC18	WG03 SWG Extensions	87/02/9-13	BSI	London	97/18 N 870
TC97 SC18	WG03 SWG Example	87/02/16-19	BSI	London	97/18 N 869
TC97 SC18	WG03-05 Comm Assessment ECR	87/03/24-27	to be determined	Austin, TX	
TC97 SC18	WG03-05 B613 Editors	87/04/15-16	to be determined	Austin, TX	
TC97 SC18	WG04	87/05/11-14	to be determined	Copenhagen	97/18/4 N705
TC97 SC18	WG05	87/03/30-87/04/03	IBM	Austin, TX	
TC97 SC18	WG05	87/05/11-15	to be determined	Europe	97/18 N 703
TC97 SC18	WG05	87/09/21-25	to be determined	Europe	97/18 N 703
TC97 SC18	WG08	87/04/27-30	tentative	Hamburg	
TC97 SC18	WG08	87/08/24-28	tentative	USA	
TC97 SC18	WG08 SWGFonts	87/01/05-09		Honolulu	97/18 N 872
TC97 SC18	WG08 SWGFonts	87/03/09-13	tentative	Boulder, Co	

SC 20 - DATA CRYPTOGRAPHIC TECHNIQUES

TC97 SC20	Plenary	87/04/09-10	Govmt Conference Ctr	Canada	97/20 N 161
TC97 SC20	WG01	87/04/13-14		Ottawa	TX d/d 12/14/86

134

TC97 SC21	WG02 SMS-30	87/03/00	Valcorune, France or	Wash, DC , USA	97/21N 1559R
TC97 SC21	WG03 Ref Mod1	87/01/19-23	DCE	Amsterdam, NL	97/21/3N 266
TC97 SC21	WG03 IRDS	87/01/26-30	Bull Systems	Paris, France	97/21/3N 263
TC97 SC21	WG03 Database Language	87/02/09-13	Bally's Hotel	Reno, Nevada	97/21/3N255
TC97 SC21	WG03 RemoteDB Access	87/02/17-19	BSI	London	97/21/3N 270
TC97 SC21	WG03 IRDS	87/05/28-29	Kikai Shinko Bldg	Tokyo, Japan	97/21N 1559R
TC97 SC21	WG03	87/06/01-05	Kikai Shinko Bldg	Toykyo, Japan	97/21 N 1578
TC97 SC21	WG04 Flt Mgmt	87/02/18-25	IBM Italia	Rome, Italy	97/21 N 1597
TC97 SC21	WG04 Config Mgmt	87/02/18-25	IBM Italia	Rome, Italy	97/21 N 1597
TC97 SC21	WG04 Secur. Mgmt	87/02/18-25	IBM Italia	Rome, Italy	97/21 N 1597
TC97 SC21	WG04	87/06/02-10	Shiba Park Hotel	Tokyo, Japan	97/21 N 1578
TC97 SC21	WG05 OSCRL	87/02/02-05	Fernuniversitaet	Hagen, Germany	97/21/5N 189
TC97 SC21	WG05 V.T.	87/03/25-04/03	BSI	London, UK	97/21N
TC97 SC21	WG05 Trans-mode	87/02/18-25	AFNOR	Paris, France	97/21/5N 187
TC97 SC21	WG05	87/06/08-12	Kikai Shinko Bldg	Tokyo, Japan	97/21 N 1578
TC97 SC21	WG06 ACSE/JLA	87/02/17-25	Holiday Inn	Palo Alto, CA	97/21/6 N240
TC97 SC21	WG06 Con Test Cas	87/04/27-05/01		Phoenix AZ	97/21N 1559R
TC97 SC21	WG06	87/06/03-11	Kikai Shinko Bldg	Tokyo, Japan	97/21 N 1578

SC 22 - APPLICATION SYSTEMS ENVIROMENTS AND PROGRAMMING LANGUAGES

TC97 SC22	Plenary & HCP	87/09/07-11		Washington	97 N 1666
TC97 SC22	Basic	87/01/22-23	Dearfield Beach	Florida	TX 12/11/86
TC97 SC22	WG02	87/06/08-12		Andover, USA	TX 12/11/86
TC97 SC22	WG02	87/04/22/24		Teeside, UK	TX 12/11/86
TC97 SC22	WG03	87/05/04/08		Dallas, USA	TX 12/11/86
TC97 SC22	WG04	87/04/27/29		Nice, France	TX 12/11/86
TC97 SC22	WG04	87/04/25/27		New York, USA	TX 12/11/86

1987

JANUARY						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

JULY						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

FEBRUARY						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

IAC: 02-03
SD-2 Tut.: 10
X3: 11-12

AUGUST						
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

MARCH						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

SPARC: 03-06
SMC: 17-18

SEPTEMBER						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

IAC: 28-29

APRIL						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

IAC: 06-07

OCTOBER						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

SD-2 Tut.: 05
X3: 06-07

MAY						
S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

SPARC: 05-07
SMC: 19-20

NOVEMBER						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

SPARC: 03-06
SMC: 17-18

JUNE						
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

IAC: 01-02
SD-2 Tut.: 08
X3: 09-10

DECEMBER						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

IAC: 01-02

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

104 (*) JCA-8
141
Doc. No.: X3/87-02-169X,I,S,T
February 20, 1987

Date:
Project:
Ref. Doc.:
Reply to:

TO: I. R.'s, X3 Community
cc: Officers, X3 TC's, SC's and SPARC/SG's

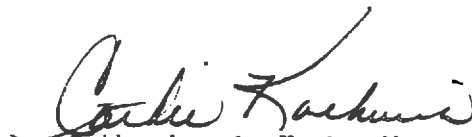
SUBJECT: NEED TO COMPLETE STAFF WORK ON NWI LETTER BALLOTS AND ALL U. S.
CONTRIBUTIONS TO THE INTERNATIONAL COMMUNITY

Printed on reverse is a letter faxed to my office yesterday. It was a desperate attempt to get some work done on the part of the TC97 Secretariat, which should have been completed prior to submittal to her. For that, this office takes full responsibility. However, we need everyone in the system to cooperate and produce the necessary documentation at the first step so that nothing is held up when time becomes a real factor.

As you know, TC97 has instituted a new letter ballot with SEVEN QUESTIONS regarding new work items. All of those must be answered when the recommended position is transmitted to this office for final X3 letter ballot. We have pointed out the additional work, but few people are paying attention. Unless we receive the full complement, we will not be able to process.

In addition, you all know that the U. S. contributions going to ISO must be written in ISO style. No first person, or X3 references, may be included. At the time of development of those comments and contributions, you are acting as ANSI's agent, and you must, in fact, present your statements in that mode.

As a result of our failure to do the above, the U. S. positions on the two referenced areas could have been discounted since they were late and incomplete. We have an advantage over the rest of the international community in that our mail system gives us additional time to do the work, we speak the same language, and we can operate under a very informal phone contact system. Let us not take our good fortune for granted.


Catherine A. Kachurik
TC97 TAG Administrator

Attachment (printed on reverse)



142

American National Standards Institute, Inc
1430 Broadway, New York, N.Y. 10018
(212) 354-3300

Cable: Standards, New York • International Telex: 42 42 96 ANSI UI • Facsimile: (212) 302-1288

February 19, 1987

Ms. Catherine Kachurik
Computer and Business Equipment
Manufacturers Association
311 First Street, N.W. - Suite 500
Washington, D.C. 20001

Dear Cathie:

I need your help immediatly. I have recently received recommended USA votes on the following documents:

- 97 N 1761
(Due Feb. 16, 1987) - Proposal for a NWI for a Register of Encipherment Algorithms
- 97 N 1773
(Due Feb. 20, 1987) - Proposal for a NWI on Portable Operating System Interface for Computer Environments - POSIX*

In neither case has the appropriate ballot form been completed - 7 questions need to be answered. Please can we straighten this out ASAP for these cases and may I request that in future you or your staff actually complete the TC 97 ballot form prior to submitting any vote on a NWI. As you know we are usually right down to the wire and there is no time left to conduct telephone inquiries.

One other problem, we are still receiving comments (USA vote on N 1761, see attached) which are written in the first person - "I recommend," etc. Can we try to do something about this as well. We are re-keying this particular one, but cannot do so in future.

Thanks for your help with these requests.

Sincerely,

Frances E. Schrotter
Secretariat
ISO/TC 97

FES:rtg
Encls.

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

104 (*) JCA-7
Doc. No.: X3/87-03-163-S P.1 of 2
Date: March 23, 1987
Project:
Ref. Doc.:
Reply to:
8457 Pushing Creek Court
Springfield, VA 22153
143

TO: Chairs, X3 Technical Committees

SUBJECT: Coordinating Liaison Responsibilities

It has come to SPARC's attention that there is still some confusion regarding the coordinating liaison method of operation. This confusion stems from two areas: 1) the responsibilities of the organization developing the standard in the matter of document distribution to organizations with coordinating liaison responsibilities for that standard; and 2) the scope of the coordinating liaison's responsibilities as it applies to that standard.

The organization developing the draft standard is responsible for providing distribution of all Committee documentation relevant to that standards development activity to each organization with which coordinating liaison has been directed. This distribution should go either to the Chair of the coordinating liaison organization or the individual he designates as the recipient. SPARC has witnessed several cases where the processing of the draft standard at the SPARC level and above has been delayed because the coordinating liaison organization was not made aware of the deliberations in the development process at an early stage, thereby causing a delay in their review and approval of the draft standard for further processing.

Close liaison, including document distribution, can rectify this problem and lead to more timely coordination by coordinating liaison organizations when the draft document is completed.

If your organization has a development project which requires coordinating liaison with another TC, please ensure that that TC receives copies of all of your documents which have a bearing on that development process. If your TC has coordinating liaison responsibilities with the development project of another TC and you are not receiving documentation on that project, please contact the Chair of the TC developing the standard to begin this distribution.

On the second area of confusion, any TC designated as a coordinating liaison organization in the development of a standard is also a coordinating liaison organization for any interpretation documents released by the TC on that standard. In this regard, SPARC is developing a change to the SD-2 to clarify this point.

When a Project Proposal is approved by X3, it automatically covers the development of a standard and any subsequent interpretations issued for that standard; therefore, any organization with coordinating liaison responsibilities at the development stage are also coordinating liaison organizations in the interpretation process. When a Project Proposal is developed for a

JCA-9
2 of 2
144

Ltr to X3 TC Chairs
March 23, 1987
Page 2

revision of that standard, the coordinating liaison issue will be addressed again and any coordinating liaison established for that revision will also apply to interpretations of the revision.

I hope this has clarified these issues. However, if there are any questions regarding this letter or any other coordinating liaison issues, please advise me and SPARC will address them. As with any new process, the coordinating liaison procedure is experiencing "birth pains" and your cooperation is appreciated while we resolve any problems pertaining thereto.

Sincerely



WILLIAM C. RINEHULS
Chairman, SPARC

**Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS***

AS/01/03-144-A, 15
Project 013-RF *1014(X)KAT*
NEWS RELEASE

JCA-10
For more information contact:
Donald Thelen, X3L2 Chair
201-644-3728 *145*

Date: March 18, 1987

*** * * PROPOSED REAFFIRMATION OF CHARACTER SETS REGISTRATION STANDARD * * ***

X3 ANNOUNCES PUBLIC REVIEW AND COMMENT PERIOD ON
THE PROPOSED REAFFIRMATION OF AMERICAN NATIONAL STANDARD X3.83-1980,
USA SPONSORSHIP PROCEDURES FOR ISO REGISTRATION ACCORDING TO ISO 2375

Washington, D. C. -- X3, the Accredited Standards Committee on Information Processing Systems, announces a two-month public review and comment period on the proposed reaffirmation of American National Standard, X3.83-1980. The public review period extends from April 24, 1987 to June 24, 1987.

X3 Technical Committee, X3L2, has reviewed this standard and concluded that the document should be reaffirmed in its entirety. This standard specifies the procedure to be followed in submitting proposals for character sets for ANSI sponsorship for submission to the ISO Registration Authority for processing in accordance with the ISO procedure for registration.

This proposed reaffirmation is available for public review and comment for a two-month period ending June 24, 1987. Copies may be obtained from the American National Standards Institute, Sales Department, 1430 Broadway, New York, New York 10018. For more information, please call (212) 642-4900.

Single Copy Price: \$5.00 plus \$2.00 postage & handling

#



AMERICAN
BANKERS
ASSOCIATION

X3/87-03-114-X, I, S
104 (*) JCH-11
NEWS 147

PUBLIC RELATIONS □ 1120 CONNECTICUT AVENUE, N.W., WASHINGTON, D.C. 20036 □ (202) 663-5000

CONTACT: Victoria Street
202/663-5469

FOR IMMEDIATE RELEASE
(87)

**CURRENT FINANCIAL DATA PROTECTION SYSTEM ADEQUATE,
NATIONAL SECURITY AGENCY TELLS ABA**

WASHINGTON, March 10 -- After discussions with the American Bankers Association, the National Security Agency (NSA) has decided to continue to support the current Data Encryption Standard (DES) program for the banking community until suitable replacement for financial data encryption and authentication is found.

The ABA began a series of meetings with the NSA in June, 1986, following the NSA's announcement to discontinue its endorsement program for DES equipment, according to Cheryl W. Helsing, chairman of ABA's Data Security Committee. "This issue has been a top priority for ABA because NSA's plans for replacing DES jeopardized the millions of dollars our industry has invested in security technology," said Helsing, who is also vice president, BankAmerica Corporation, San Francisco. "This agreement is a signal to the banking industry to continue its progress in improving electronic funds transfer security and privacy protection of computer data," she said.

DES encryption is commonly used to securely scramble passwords, Personal Identification Numbers, and other sensitive information for computer storage and data communications. For example, DES is commonly used in automated tellers and point-of-sale systems, and is the basis for new industry standards for assuring the integrity and authenticity of electronic funds transfers.

-more-

The NSA reported to the ABA that the current DES algorithm will for some time to come provide adequate security to protect the confidentiality and integrity for financial applications. Although the NSA feels the public DES algorithm should eventually be succeeded by a secret algorithm under the Commercial COMSEC Endorsement Program (CCEP), they have recognized through the seven months of discussions with the ABA, that current CCEP products having export control may have limited application for many banks. Therefore, the NSA will continue to support endorsement of DES equipment for banking applications under FED Standard 1027 until a suitable CCEP replacement is available.

The NSA also considered that the DES-based security techniques have been widely accepted by banks and their customers, resulting in major investments in DES equipment. Their decision to endorse DES products for the banking community, either within CCEP or other NSA-supported programs, is in direct support of the industry.

"Clearly, the development of CCEP products for banks will require a continuing dialogue with the NSA," said Eddie L. Zeitler, vice president, Security Pacific National Bank, Los Angeles and project manager for the NSA liason with the ABA Data Security Committee.

"Our initial meetings have been mutually beneficial and have resulted in a better understanding of the need to support DES implementations. We look forward to a continued working relationship with the NSA in the development of new cryptographic tools for the banking industry," Zeitler said.

The American Bankers Association is the national trade and professional association for America's commercial banks of all sizes and types. The assets of its member banks comprise about 95 percent of the industry total.

MEMORANDUM

To: X3J3
From: Mike Metcalf
Subject: Minor Edits (C. 25.1034)

In the course of the work of editing during the 103rd. meeting, I noted the following other small points which I consider would improve the document. Only the change #7 is of any substance.

1. P. 3-1, l. 25-38:
Box the Table, (better layout).
2. P. 4-5, l. 39:
After "The length is" add "specified by", (clearer wording).
3. P. 5-1 l. 11-12:
Change "EXPEND" to "EXPENDITURE", (better use of long name).
4. P. 5-7, l. 6:
Add "nor" before "be", (better wording).
5. P. 5-10, l. 13 and 15:
Add "value" after "default", ("default" is an adjective).
6. P. 5-13, l. 23 and 26:
Delete (superfluous).
7. P. 5-14, l. 25:
Replace
is RANGE [/range-list-name/]array-name-list
by
is RANGE/ range-list-name/array-name-list
or RANGE[::] array-name-list

(make use of double colon permissible, in order to be more regular with respect to other attribute definitions).
8. P. 8-4, l. 26-28:
Move these two sentences to be two separate constraints following 1. 2. Set "case-value" as "case-value" and "Case-value" as "The case-value".
9. P. 8-6, l. 15:
Change "nor" to "or";
l. 16:
change the first "the" to "a", (better grammar, and consistency of wording).
10. P. 8-9, l. 24-39:
Delete, or set in obsolescent font, (obsolescent code).
11. P. 10-2, l. 10:
Change "It" to "The integer literal constant r"; move this sentence out of the constraint onto the following line, (a definition does not belong in a constraint).

151

NCC
NATIONAL CENTRE
FOR INFORMATION
TECHNOLOGY

The National Computing Centre Limited Oxford Road Manchester M1 7ED
Telephone 061-228 6333 Telex 668962

Telecom Gold 81:NCT 008; Fax (Groups 2 & 3): 061-228 2579;

25th March

To X3J3 Members,

I would like to thank the members of the X3J3 committee for their hospitality during the recent meeting in Los Angeles. I am especially grateful to the subgroup members who ensured that I understood the point under discussion, as this has helped my appreciation of the new standard.

I am a little concerned, as I said on the Friday, that the public review period will cause detrimental comments to be made because of misunderstandings gained whilst reading the document. I feel that one cause of the problem is because there are two styles in the document, a tutorial style and a strict formal style, and these are intermixed in what appears to be an arbitrary fashion. It can be frustrating to encounter the formal style for a feature when the previous feature was described in more depth, but less precisely, i.e. in a tutorial style.

In discussions during a break in the full committee session I proposed that maybe consideration should be given to providing two documents, a formal standard with an associated rationale, to be published together. This would not cause as much new work as it may seem because there would not be additional technical work, the editor may disagree. My suggestion was also disagreed with during our discussion based upon the experience of the PL/1 standard, of which I unfortunately have no knowledge, although I do not think there was a rationale document.

I apologise for sounding negative, it was not my intention, I just wanted to give a reaction to the document from a maybe typical user of the language (prior to my employment at NCC) who was seeing the document for the first time. The document is undergoing a fairly extensive review whilst the ballot papers are being processed and I have found that my particular thoughts about certain sections are reflected in member's comments. The next version of the document will therefore undoubtedly be more readable as a result of all these efforts.

I hope that the work of the committee continues to be successful.

Yours sincerely

Alison Wearing

Alison Wearing

Languages and Validation Services

Registered Office
The National Computing Centre Limited Oxford Road Manchester M1 7ED Registered in England Number 881195

Regional Offices

Belfast Birmingham Bristol Glasgow London Manchester



L A C M E M O R A N D U M

February 27, 1987

To: Interested FORTRAN users

From: L. Moss

Subject: Trip Report on 103rd X3J3 Meeting, 9-13 February 1987

Note: This is a personal report of these meetings and in no sense does it constitute an official record.

SUMMARY

The result of the recent letter ballot was 29 affirmative or affirmative with comment, and 7 negative. This is sufficient to forward S8 to X3 for public review (which requires one more than half the members and one more than 2/3 of the members voting). WG5 conducted a parallel, but informal, ballot at the same time, with the result: 25 yes, 2 no.

This meeting was devoted entirely to processing the comments from the letter ballot (including those received from WG5 members). Nearly all of this work was editorial in nature, and I will not report it in detail. Instead, I will explain how we attacked the work, and touch on a few of the items that were more technical in nature.

Late in January, the editorial and steering committees met in Albuquerque to sort out the individual items from the various ballots (many of them duplicates of one another). The result of this meeting was a cross reference list which identified each ballot "item" with a sequence number, contained pointers both to the appropriate page and line of S8 and to the relevant ballot page, and assigned a preliminary classification code to the item, according to the following scheme:

- 1 Typos, formatting, etc.
- 2A Changes in style and taste
- 2B Changes that might be controversial, but not substantive
- 3 Substantive changes
- 4 Substantive changes requiring subgroup study
- 5 Additions to the language that did not seem appropriate at this time (these were essentially items that had been discussed and rejected following the previous ballot)
- R Unclassified, rejection recommended

We then proceeded through the ballots in alphabetical order. Each member was given a chance to change the classification assigned to the items in his ballot, to defend individual items, and/or to make general comments about his ballot. Items classified as 1 or 2 were then

154

assigned to the editorial subgroup, while items classified as 4 or 5 were assigned to the appropriate technical subgroup. Some of the technical subgroups also reviewed the items classified as 3 or simply "R" (i.e., rejection with undefined classification) within their sections of S8. Some of the items classified as 3 were voted on in full committee as we got to them, while others were deferred in the interest of making a first pass through all the ballots by the end of the week. As items came back from the technical subgroups, they were voted on in full committee. As usual, the recommendations from the editorial subgroup were lumped together (subject to requests from the floor to separate individual items) and voted on all at once.

By late Friday afternoon, we managed to complete the first pass through the ballot, and had even resolved a large number of the items. However, there were still a considerable number of items unresolved. A special meeting has been called in Albuquerque for March 23-27. This will be run as a continuation of the 103rd meeting, and will try to resolve all remaining items classified as 1's or 2's. There will also be subgroup time to make recommendations for 4's and 5's, to be voted at the regular May meeting. I believe the intent is to look also at the remaining 3's, and possibly vote on those that seem less controversial (by our usual rules, of course, anything decided in March can be reopened in May).

At the May meeting, we will attempt to resolve the remaining substantive items, and will draft and vote on the actual committee response to each item. Members will be given an opportunity to change their letter ballot votes, then, providing we still have the necessary 2/3 majority, the document will be forwarded to X3.

OPENING BUSINESS

Administrative Business

There is a new call for candidates for International Representative for X3J3 (the pending applications for this position have been deemed out of date).

Jeanne Adams' current term as chairman of X3J3 ends in May, and there is a call for candidates for this office.

Related X3 Activities

SPARC is considering a proposed change to the SD-2 procedures. At present, if the vote of the technical committee (e.g., X3J3) on forwarding a draft standard for public review is not unanimous, X3 conducts a 30-day letter ballot (and possibly a 15-day follow-on reconsideration ballot) on whether to release the draft. If the technical committee vote is unanimous, the public review is automatic. The proposed change would allow the automatic public review when the technical committee vote is less than unanimous (say, 90% or some other algorithm). Comments on this proposal are solicited before SPARC acts on it at its May 5-7 meeting.

Membership

At the beginning of this meeting there were 37 members, with none on provisional status. Unisys is giving up one of its memberships, thus leaving us with 36 voting members. A quorum is 13.

Minutes

The minutes of both the 101st meeting (August 1986, Halifax, Nova Scotia) and the 102nd meeting (November 1986, Albuquerque, NM) were approved unanimously.

Liaison Reports

Mike Metcalf and I reported briefly on the discussion of Fortran 8x during the recent Asilomar conference on Computing in High Energy Physics.

SPECIFIC PROPOSALS

All contributed documents are given numbers of the form, "mm(wg)aaa-n", where:

- mm is the meeting number (the February 1987 meeting was number 103).
- wg is the number of the working group assigned responsibility for the proposal, or "*" for individual proposals.
- aaa are the initials of the author.
- n is a small number to distinguish different documents from a single author at one meeting.

The results of straw votes (SV) are, unless otherwise noted, given as: (yes-no-undecided), with an asterisk next to my vote; formal votes (FV) are always strictly (yes-no). "WITHDRAWN" means that, following some discussion, there was no motion or that the motion or second was withdrawn.

103(16)CDB-3 PROMPT= Specifier

(Burch ballot.) The current draft states that the processor will write the prompt string on the unit which is about to be read, which could be a disaster if one is reading from a disk file. This proposal changes the wording to say that the string is "displayed in a processor-dependent manner, if possible". FV (27*-0) -- PASSES.

103(*)RCA-1 Deleting the Case-Value Range of the form ":"
(Allison ballot.) This proposal removes the degenerate case selector,
"CASE (:)". FV (24*-0) -- PASSES.

103(*)THRE-2 Processor Conformance

(Related to Ellis ballot.) One of the WG5 resolutions at the Halifax meeting last August requested that standard-conforming processors be required to flag the use of "forms" from the deleted, obsolescent, and deprecated lists, and/or any processor extensions. An ad hoc subgroup at this meeting met to formulate such a conformance requirement for processors. The term "forms" was generally interpreted to mean syntax checking only, with no requirement for run-time checks nor for extensive compile-time data flow analysis. It was decided to limit the required checking to a single program unit, and to those requirements expressed in the syntax rules along with the formal constraints. In other words, the BNF and the constraints define the things that must be checked; violations of any restrictions contained in ordinary text need not be detected by the processor. Part of this proposal included moving about half a dozen constraints, which did not seem to be easily checkable at compile time, to ordinary text. FV (24*-0) -- PASSES.

103(*)RAH-7, Proposal 1

(Related to Hendrickson ballot.) This is a proposal to restrict which intrinsic functions may be used in constant and specification expressions. A four way straw vote was taken:

Allow only the inquiry functions	-- 4
Allow inquiry plus a few math functions	-- 10
S8 as is (i.e., all intrinsic functions)	-- 10*
Undecided	-- 6

A second straw vote was taken, eliminating the first option above:

S8 as is (i.e., all intrinsic functions)	-- 13*
Allow inquiry plus a few math functions	-- 10
Undecided	-- 6

At this point the proposal was tabled; it will, presumably, come up again at either the March or May meeting.

Harris Ballot

Harris voted "NO" on the letter ballot, citing objections to a number of major features of the language. We took straw votes on several of his specific objections:

- Remove MODULE subprograms: SV (4-18*-7).
- Disallow USE inside MODULEs: SV (3-19*-8).
- Add a simple textual include: SV (15-12*-5).

I believe one of the subgroups will look into adding a textual include.

103(*)BTS-2 Host Association

(Related to a number of ballots including B. Smith's.) As a result of some changes made as part of the compromise, some language was inadvertently lost which explained that an internal or module procedure had access to the objects defined in its host. This proposal attempts to fix this problem by defining a new concept of "host association", which is analogous to "use association". The subgroup made a number of minor amendments, then a formal vote was taken: FV (27*-0) -- PASSES.

103(16)CDB-4 NULLS= and VALUES= specifiers in NAMELIST I/O

(Related to Hendrickson and Ragan ballots.) A couple of ballots pointed out ambiguities in the handling of NULLS= and VALUES= specifiers in NAMELIST I/O. After looking at the problem, the subgroup concluded that these specifiers are not particularly useful for NAMELIST I/O, especially in the case of input because of the random order of the data. The subgroup therefore made this proposal, which forbids these specifiers in NAMELIST I/O statements. FV (25*-0) -- PASSES.

Bit Data Type

Several ballots, including those of Harris, Metcalf, Moss, Philips, and Weaver, mention the need to restore bit data type to the language. A straw vote was taken on bits: SV (12*-5-8).

103(*)MBM-1/item 2.6 Fix IOLENGTH

(Metcalf ballot.) The original proposal for the IOLENGTH= specifier in the INQUIRE statement disallowed any other specifiers in the same statement. This restriction was inadvertently lost, and the current draft allows IOLENGTH= to be freely mixed with other INQUIRE specifiers. This proposal restores the restriction: FV (13*-5) -- PASSES.

Incorporate the Glossary (S11) as an Appendix

(B. Smith ballot.) Initial straw vote: SV (19*-2-4). Brian Smith argued that reading through the glossary made it much easier to understand the main text, even for someone on the committee. Including it in the document as an appendix, would insure that the public has access to it which would, in turn, help us to obtain more meaningful comments. The argument that the potential for conflicting definitions makes this dangerous was rejected by a number of people, since the appendix is clearly subordinate to the main text. FV (14*-4) -- PASSES.

103(16)LJH-2 Optional Comma in the Do-construct

Two ballots (B. Martin and L. Rolison) point out that the current BNF for the DO-construct allows an optional comma to appear in some places that may be undesirable. For example:

```
DO 1, (5 TIMES)
DO , (5 TIMES)
DO , I=1,5
```

The subgroup recommended proposal 1, which would disallow the first two examples above, but permit the third: FV (16*-1) -- PASSES. The

subgroup also recommended requiring the comma in cases like the third example, since this syntax seemed a little less error prone (in fact, the subgroup would probably have recommended requiring the comma also when a label is present, except that this violates compatibility with Fortran 77). A straw vote was taken: SV (5*-11-4) -- WITHDRAWN.

CLOSING BUSINESS

The registration fee for each of the next two meetings (Albuquerque and Seattle) will be \$70.

Future Meetings

- 103.5 23-27 March, 1987, Albuquerque, NM (host: Walt Brainerd).
104th 11-15 May 1987, Tukila, WA (host: Ivor Philips).
105th 10-14 August 1987, Liverpool, England (host: Lawrie
 Schonfelder). WG5 will meet the preceding week (3-7 August)
 at the same place.
106th 9-13 November 1987, Ft. Lauderdale, FL (host: Graham Barber).

Next Distribution

The closing date for the next regular pre-meeting distribution is 6 April 1987. To get an item into the distribution it should be sent to (NB: this is a new address and telephone number):

Richard A. Hendrickson
Cray Research Incorporated
1345 Northland Drive
Mendota Heights MN 55120
(612-681-5804)

There will not be time for a regular distribution for the special March meeting. However, anyone wishing to put documents on the table for that meeting should get them to Jeanne Adams by 16 March. She will get them copied and take them in her car to the meeting. Her address is:

Jeanne C. Adams
Scientific Computing Div.
NCAR
P.O. Box 3000
Boulder, CO 80307

To: BCS, NAG, Fortran Forum, etc.

From: John Reid

Date: 19 February 1987

Subject: X3J3 meeting in Los Angeles

Note: This is a personal report of the meeting and in no sense does it constitute an official record of it.

1. Summary

A letter ballot of all X3J3 members was conducted prior to this meeting, asking the question "Do you approve the draft Fortran Revision...enclosed with this ballot for submission to X3 for further processing as an American National Standard?" The result was Yes: 9; Yes with comments, 20; No: 7. Since less than 1/3 voted "no", the ballot passed. Short summaries of the reasons given for each no vote appear in Section 2. It looks of 2 or 3 of these may change to yes votes once all the comments have been processed, but the rest are asking for drastic changes that are quite unacceptable to a majority of the Committee.

The meeting was concerned with processing the huge number of comments received with the ballots. A preliminary meeting (attended by Jeanne Adams, Jeanne Martin, Walt Brainerd, Jerry Wagener, Brian Smith and Dick Hendrickson) collated the comments and categorized them (1 for minor edits, 2a for editorial changes in style and taste, 2b for editorial changes that might be controversial, 3 for substantive changes, 4 for substantive changes requiring subgroup study, and 5 for changes to the language that would be appropriate in the public comment period). Most of them were labelled as recommended for acceptance (**), covered by another recommended change (*), or rejected (R).

Each committee member in turn was given the opportunity to introduce his or her ballot and discuss items other than those in categories 1 and 2, which were considered by the Editorial Subgroup and moved in blocks. The changes in categories 3 and 4, and the outright rejections, were passed to the technical subgroups to be studied and moved singly. The only technical change made (other than corrections of errors) was over requirements for processor conformance (see Section 3), Substantial rewriting of text for host association (the term adopted for access by internal procedures to host entities), array constructors, and the IDENTIFY statement were adopted. It was decided that the glossary should be included as an appendix and that there should be an index of the bnf terms. There was some sentiment in favour of limiting the intrinsic functions permitted in specification statements, but the proposal was withdrawn for further work. It was confirmed that entry statements are not permitted in interface blocks (each entry has its own interface).

2. The no votes in the ballot

The no votes were as follows:-

1. Kevin Harris (DEC) wants

(a) derived data types to be removed and new intrinsic types to be added for bit strings, general pointers, and varying length character strings;

(b) to disallow modules to contain procedures or access other modules; to add an INCLUDE facility;

(c) to remove generalized precision; to require minimal accuracy standards for operations and intrinsic functions; and

(d) to reduce the obsolescent features list to noninteger DO variables, ASSIGN, assigned GOTO, and assigned FORMAT specifiers; to reduce the deprecated list to one item, the old source form.

2. Dick Hendrickson (Cray) wishes to see the large number of technical errors and holes that he has found corrected.

3. Anil Lakwara (Peritus) believes that the document does not conform with X3J3's Charter, that the language is too large and complex, and that implementations will be too large and inefficient.

4. Len Moss (SLAC) feels strongly that X3J3 should not publish the document until the whole committee is convinced that it is understandable, and is concerned about the large number of editorial and technical errors that remain in the document.

5. Ivor Philips (Boeing) wishes to change the Appendix F bit facility to be based on bit strings and to add an OVERLOAD statement that flags any overloading of procedures or defined operators.

6. Larry Rolison (Unisys) has found virtually no new feature that will enhance either compiler or run-time performance and is concerned that the committee has created a negative incentive for the average user to move to an 8x implementation.

7. Dick Weaver (IBM) wants Fortran 8x to be reorganized into two languages. One would be called Fortran and would consist of Fortran 77 plus data structures, array language as an option, bit string data type, dynamic allocation, environmental inquiry functions, precision specification, keyword and optional arguments on CALL statements, IMPLICIT NONE, and the new form of DO. The rest would be processed as a technical report, to allow implementation experience, testing and evaluation.

8. Processor conformance

ISO Working Group 5 requested that processor conformance requirements be extended, essentially to require static checking of the syntax of a program for violations of the standard and for the use of extensions. The committee accepted the spirit of the proposal, but had some difficulty with the exact wording. After discussions in full committee and ad hoc subgroups, the following was adopted (24-0):

1. Add to the body of S8:

A processor conforms to this standard if

(1) It executes any standard conforming program in a manner that fulfils the interpretations herein, subject to any limits that the processor may impose on the size and complexity of the program.

(2) It contains the capability to detect and report the use within a submitted program unit of a form designated herein as deleted, obsolescent, or deprecated, insofar as such use can be detected by reference to the numbered syntax rules herein and their associated constraints.

(3) It contains the capability to detect and report the use within a submitted program unit of an additional form or relationship which is not permitted by the numbered syntax rules or their associated constraints.

However, a processor is not required to detect or report the use of deleted, obsolescent, or deprecated features, nor the use of additional forms of relationships occurring in a format specification which is not part of a format statement (10.1.1).

2. Add to Section Notes (Appendix C):

The standard requires a standard-conforming processor to be capable of detecting and reporting the use within a program unit of forms designated as deleted, obsolescent, or deprecated, or of additional forms or relationships, where such use can be detected by reference to the numbered syntax rules and their associated constraints. It is recommended that the processor be accompanied by documentation which specifies the limits it imposes on the size and complexity of the program and the means of reporting when these limits are exceeded, which defines the additional forms and relationships it allows, and which defines the means of reporting the use of additional forms and relationships or of the deleted, obsolescent, or deprecated forms. Note that in this context, the use of a deleted form is the use of an additional form.

4. Further meetings

Because there is much work still to be done on the ballot proposals, the Chairman has called for an extra meeting to be held at Albuquerque, 23-27 March. This will be treated as a continuation of the February meeting, without separate minutes. It should ensure that processing of the ballots is completed at the meeting in Seattle, 11-15 May. The premeeting deadline is April 6.

To: ISO, Fortran Forum, BCS, NAG, etc.

From: John Reid

Date: 30 March 1987

Subject: X3J3 meeting in Albuquerque

Note: This is a personal report of the meeting and in no sense does it constitute an official record of it.

1. Summary

This was technically a continuation of the February meeting, called because of the volume of comments received with the ballots. About half the members were present, including almost all those who suggested large numbers of changes. It was a very hard-working and productive meeting and at its end everyone seemed satisfied that the comments had been considered adequately. The plan is to incorporate all the accepted changes into the draft Standard and vote on the resulting document at the beginning of the May meeting. I am expecting to find it ready for release for public comment and am hoping that the necessary two-thirds of the Committee will too. We need also to respond formally to all those who submitted comments. Constructing and approving these responses will be a significant task for the May meeting.

The bulk of the changes were editorial and the details are not interesting to readers of this report. We did, however, make a change to the default implicit rules (Section 2), that I find very welcome; change the syntax of DO () TIMES (Section 3), again a change that I welcome; and there were various minor technical changes and clarifications (Section 4).

2. The default IMPLICIT rules

The default IMPLICIT rules were complicated and several members commented adversely in their ballots. I was not one of them, but was keen to see a change, so I participated in a small working party and co-authored a paper during the meeting. We proposed that in a nested set of scoping units, each should take for its default the IMPLICIT rules of its host, with the Fortran 77 default in the outermost one. In each scoping unit, IMPLICIT rules might be used to override the defaults for any or all of the letters. Safe practice is to use IMPLICIT NONE, particularly in inner scopes, but we felt that it is wrong that sometimes this should be the default and sometimes the Fortran 77 rules should be the default. Also it should be possible to define the IMPLICIT rules throughout the whole of a program unit by a single statement at its head. We were encouraged by very favourable straw votes and the proposal passed (15-0).

3. The syntax of the DO statement

Larry Rolison suggested that the syntax of the statement

DO (scalar-int-expr TIMES)

be replaced by

DO (scalar-int-expr) TIMES

in order to make parsing easier. The Committee was rather undecided about whether such a change was really necessary, but eventually approved it (10-5). I have been asked for this change, and supported it.

4. Miscellaneous minor changes and clarifications

It was decided to allow a specification expression to refer to the result of another specification expression, for example

INTEGER B(2*SIZE(A))

only if the expression referenced comes earlier in the specification sequence. This disallows such a case as

INTEGER A(2*SIZE(B)), B(2*SIZE(A))

It was agreed that constant arrays may have the RANGE attribute.

It was decided to allow construct names to be used on ELSE IF, ELSE, and CASE statements.

Automatic arrays are arrays that are not dummy arguments but have variable extents. It was decided to use the term "automatic object" for both these and structures with varying type parameters since these have similar properties. For example, matrix arithmetic might be implemented by a type MATRIX that had parameters for the numbers of rows and columns. Automatic objects are created on entry to a procedure and lost on return.

It was decided (8-5) not to permit statement functions in modules.

Wording was accepted (13-0) to clarify the rules that an alias may have only one parent. It is acceptable for an alias to be associated with two subobjects of the same named object.

5. Next meeting of X3J3

The next meeting of X3J3 will be in Seattle, 11-15 May. The premeeting deadline is April 6.