Minutes

Meeting of ISO/IEC JTC1/SC22/WG5

6-10 November 1995

Ramada Inn San Diego North, 5550 Kearney Mesa Road
San Diego, California 92111, USA

1. Opening of the Meeting

The meeting was opened by the convenor, Miles Ellis, at 9:05am 6th November 1995.

2. Opening Business

2.1. Introductory Remarks from the Convenor

The convenor began with introductions from all present. He then outlined the primary purpose of the meeting, which was to process the ballots from the CD registration and approval ballots. Secondary purposes of the meeting were to discuss the technical reports that had been proposed at the Tokyo meeting and approved by SC22, finalising the scope of the technical report on data type enhancements, and to take a first look at the Fortran 2000 requirements.

Papers distributed at the meeting were all given “S-numbers”; only those papers that were to be distributed to the full WG5 membership after the meeting received N-numbers. This policy avoids unnecessary distribution of temporary papers used only at the meeting.

2.2. Welcome from the Hosts

The head of the U.S. delegation, Jerry Wagener, welcomed everyone to the meeting.

2.3. Local Arrangements

Ted Terpstra welcomed everyone on behalf of General Atomics and explained the local arrangements.

2.4. Appointments for this Meeting

David Muxworthy, Stan Whitlock, Wolfgang Walter and Masayuki Takata were appointed to the Drafting Committee. Malcolm Cohen was appointed as Secretary. Kurt Hirchert was appointed Librarian.

2.5. Adoption of the Agenda

The convenor proposed breaking into subgroups to expedite processing of the Fortran 95 comments, and noted that we needed to produce a formal document for ballot processing.

Janice Shepard requested that WG5 discuss and approve the interpretation items that have been approved by X3J3, and that if possible WG5 should propose edits for the outstanding items. Miles noted that edits approved by X3J3 next week can be processed by a letter ballot of WG5 for inclusion in Technical Corrigendum 3.

Richard Maine requested that all changes in Technical Corrigendum 3 must include explicit edits for incorporation into the Fortran 95 DIS.

Tutorial sessions on exception handling, data type enhancements, conditional compilation, interoperability and interval arithmetic were scheduled at intervals throughout the week.

The subgroups established for the week were:

- Interp: Janice Shepard with Jon Steidel, Masayuki Takata and Rich Bleikamp.
- Exception: John Reid with Keith Bierman, Dick Hendrickson and Ted Terpstra.
- Interop: Michael Hennecke with Richard Maine, Kurt Hirchert and Stan Whitlock.
- Data: Malcolm Cohen with Jeanne Martin, Steve Morgan and Jeanne Adams.
- CC: David Epstein with Tom Lahey, Minoru Tanaka and Lars Mossberg.
- Interval: Wolfgang Walter with Baker Kearfott, Walt Brainerd and Tony Warnock.
- Policy: Miles Ellis with Jerry Wagener and David Muxworthy.

The agenda was adopted nem con.

2.6. Approval of the Minutes of the Tokyo Meeting [N1118]

The minutes of the Tokyo meeting were approved nem con.

3. Reports

3.1. National Activity Reports (Heads of Delegations)

Wolfgang Walter gave a verbal report on the activities of the German member body.
Masayuki Takata reported on activities on the Japanese member body [N1158].
David Muxworthy reported on activities of the U.K. member body [N1159].
Jerry Wagener reported on activities of the U.S. member body [N1156].

3.2. Report from Primary Development Body

Richard Maine gave a brief status of the editing of the draft document.
Janice Shepard reported on the status of interpretations of Fortran 90 and welcomed input from WG5 members on the (22) outstanding problems.

3.3. Status of Tokyo Resolutions [N1116]

Paper N1157, which contains the X3J3 responses to the Tokyo resolutions, was tabled.

3.4. Liaison Reports

John Reid reported on the activities of SC24/WG4 (GKS) on a draft Fortran 90 binding to GKS. Informal comments he made directly to the editors of the document on improving its quality had been very well received, and it was suggested that we have an official liaison. Fortran 90 will be the first binding to the new GKS standard. It was suggested that John become the official liaison for the time being on the understanding that attendance of SC24/WG4 meetings would not be necessary, and other members were encouraged to examine the draft binding to the new GKS standard and assist in its improvement.

John Reid reported on the activities of IFIP WG2.5. Bo Einarsson had presented Keith Bierman's paper on a procedural approach to floating-point exception handling to the WG2.5 meeting. WG2.5 were strongly of the opinion that floating-point exception handling was essential in the immediate future and expressed the view that either the ENABLE construct or the procedural approach were acceptable. They further encouraged us to take cognisance of C implementation experience if the procedural approach is used, since it is very similar to the proposed C numerical extensions. Wolfgang Walter noted that WG2.5 were also in favour of adding interval arithmetic to Fortran.

Jerry Wagener reported on the activities of the High Performance Fortran Forum, who are currently developing a revised HPF specification. Proposals to add asynchronous i/o and interoperability with C are currently at an advanced stage in the revised HPF processing.

3.5. Report of SC22 Plenary

Miles Ellis reported on the outcome of the recent SC22 plenary session [N1150].

4. Processing of Fortran 95

In some of the straw votes recorded below, papers referred to by meeting numbers (S-numbers) were later subsumed into N1167.

4.1. Processing of CD Ballot Comments [N1135,N1142]

A number of straw votes of the full committee were taken throughout the week to decide how to resolve specific issues raised by the CD ballot comments that could not be settled by the relevant subgroup. Straw votes are recorded as Yes/No/Undecided.

SV: Whether to accept the German recommendation to always include the optional "::"? 3/12/9.
SV: Whether to accept the German recommendation to make indentation more consistent? 5/14/5.
SV: Whether to accept the German recommendation to add "LEN=" to all CHARACTER declarations? 0/11/13.
SV: Whether to accept the German recommendation to add appropriate names to all END statements? 16/2/5.
SV: Whether to accept the U.S. recommendation not to make assumed-size arrays obsolescent? 6/8/10. This inconclusive vote was retaken later in the week and decided as 19/6/0.
SV: Whether to accept Malcolm Cohen's suggestion that CHARACTER* syntax not be made obsolescent? 1/7/16. This inconclusive vote was retaken later in the week and decided as 17/7/0.
SV: Whether, in US Problem 20, X should be visible only by name Y? 2/7/15. This inconclusive vote was retaken later in the week after an explanatory presentation by Janice Shepard and decided with the four-way straw vote: "Disallow mixing USE ONLY and USE?/X only visible as Y?/X visible as X and Y?/Undecided?/" 5/1/17/2.
SV: Whether to accept the U.K. recommendation to allow all pure intrinsics in specification expressions? 16/2/6.

SV: Whether to accept Richard Maine's recommended edits to make the use of "subprogram" and "procedure" consistent? nem con.

Further processing took place on Friday.
SV: Approve paper S36A as amended (withdrawing 213:11-12, 213:18+, 213:27, 213:30 for passing on to X3J3)? nem con.
SV: Approve paper S42A subject to X3J3 review? 14/4/3.
SV: Approve S28.5 as is?/Accept S51.2 edit?/Pass to X3J3?/undecided? 6/0/16/2.
SV: Remaining items in S51? nem con.
SV: S47 as amended? nem con.
SV: Approve paper S40? nem con.
SV: Approve paper S30a as amended? nem con.
SV: Add an explanatory sentence in each case where DIM is not now an optional argument but a separate form of the intrinsic? 3/16/6.
SV: Approve paper S31a as amended? nem con.
SV: Reject interpretation 194 of N1194 and return the defect item to X3J3? nem con.
SV: Reject interpretation 185 of N1194 and return the defect item to X3J3 as part of S51 referral? nem con.

4.2. Disposition of CD Ballot Comments

Straw votes were taken to settle the disposition of the ballot comments prior to assembly of the disposition document.

SV: Accept S28 without items 1 and 5? nem con.
SV: Accept S30 as amended by the author? nem con.

4.3. Finalising of DIS [N1143]

Miles Ellis requested that explicit instructions necessary for the project editor Richard Maine to finalise the DIS document following from the decisions taken in agenda items 4.1 and 5.1 be provided in machine-readable form to him by Saturday, subgroup heads being responsible for producing the list of edits passed by the full committee as a result of their recommendations.

5. Fortran 90 Defect Management

In some of the straw votes recorded below, papers referred to by meeting numbers (S-numbers) were later subsumed into N1161.

5.1. Fortran 90 Corrigendum 3 [N1140,N1141,N1146]

Straw votes of the full committee were taken to resolve contentious issues as well as to finalise the corrigendum contents.
SV: Whether the POINTER attribute should continue to be allowed for DO variables?
Yes/No/Undecided = 18/5/0.

SV: Accept N1141 except for defects 125, 183 and 198, and with defect 83 altered by replacing, in the 3rd edit of the EDITS section, "applies to" by "is consistent with"? 19/0/3.

SV: Accept N1141 defect 198 [after discussion]? nem con.

Janice Shepard proposed that the policy for handling the outstanding defects for which WG5 provide a proposed edit be to mark the defect as status "WG5 approved; ready for X3J3" and that these would be included in Technical Corrigendum 3 following an X3J3 letter ballot (no further ballot of WG5 being necessary).

SV: Accept proposed response for defect 148 in paper S15? nem con.
SV: Accept proposed response for defect 154 in paper S16? nem con.
SV: Accept proposed response for defect 201 in paper S17? nem con.

SV: Accept proposed response for defect 176 in paper S22? nem con.
SV: Accept revised response for defect 125 in paper S23? nem con.

Janice Shepard made a presentation on defect 81.
SV: Accept proposed response for defect 81 in paper S25? nem con.
SV: Accept proposed response for defect 187 in paper S34a? nem con.
SV: Accept proposed response for defect 27 in paper S37? nem con.

SV: Accept proposed responses for defects 196, 203, 183, 145? nem con.
SV: Accept proposed reponse for defect 155 (Fortran 90 only, no F95 impact)? nem con.

Conflicts between the previously proposed response for defect item 194 and the US ballot comment editing 119:11 were discussed.
SV: Alter edit for 119:11?/revisit defect 194?/undecided? 0/12/11.
SV: Approve paper S19B as amended, drawing 37:39 and 291:33+ to X3J3's attention? nem con.

Janice Shepard noted at the end of the meeting that the disposition of the current interpretations was that there were 27 WG5-approved ready for SC22, 13 WG5-approved ready for X3J3 and a further 13 that were still open under X3J3 consideration (only two of which needed edits to the standard, and therefore have to wait for Fortran 95; these being defects 180 and 190).

Miles Ellis personally thanked Janice for all the work she has put into masterminding the Fortran 90 maintenance task over the last several years.

5.2. Defect Management for IS 1539-2

Miles Ellis noted that the project editor for the varying string module IS 1539-2, Lawrie Schonfelder, had agreed to handle defect management for this standard for the immediate future.

6. Reports on work for TRs [N1136]

6.1. Exception Handling [N1137,N1149]

John Reid presented the two possible approaches that would provide floating-point exception handling. Stan Whitlock expressed some concern that the procedural approach would unfairly discriminate against particular hardware. Baker Kearfott noted that the intention of the procedural approach was to allow implementation either with flags or with opcodes, depending on the appropriate hardware. Jon Steidel pointed out that some hardware did not allow continuing after overflow detection; John accepted that this meant that the minimal requirements for conformance to the procedural approach would be simply to implement the enquiry functions Janice Shepard expressed some concern that the ENABLE construct approach without user-defined conditions would lead to later problems. Miles Ellis noted that whichever approach is taken it must be definable and implementable in 2-3 years to be useful to the users.

SV: Whether to add the ENABLE construct sometime in the future? 19/0/5.
SV: Continue to develop the TR on ENABLE? 0/9/16.
SV: Adopt the procedural approach for the exception handling TR? 6/1/18?

Dick Hendrickson: But will the vendors implement it? There is no point in imposing an optional requirement.
Keith Bierman: It is possible for a user with no access to proprietary compiler sources to add the procedures themselves.

The "Exception" subgroup met with those concerned to address the issues raised above, as a result of which John Reid reported that the subgroup's recommendations were to adopt the procedural approach because it will allow us to produce the technical report on schedule and does not discriminate against any vendor.

Wolfgang Walter: Is the ENABLE approach not further advanced technically?
Answer: ENABLE is much harder. People have different underlying models and this would need to be resolved. Adopting the procedural approach to floating-point exception handling now does not rule out a more general ENABLE construct in Fortran 2000.
Richard Maine: It is not clear that ENABLE should actually be of such high priority once we have a facility for floating-point exception handling. The procedural approach also has the advantage of fully supporting IEEE arithmetic.

SV: Whether the TR should adopt the procedural approach (vs. ENABLE)? 19/3/3.

Wolfgang Walter: It is not clear that rounding modes as in this proposal are suitable for implementing interval arithmetic.
Stan Whitlock: Rounding modes should not interact with interval arithmetic, they are in this proposal because they are defined with IEEE and needed for full support.
Janice Shepard: We should not delay the TR just to add all the IEEE facilities that are not needed for exception handling.

SV: Whether the TR should include other IEEE functions (as in the draft)? 8/4/12.

6.2. Interoperability with C [N1147]

Michael Hennecke presented the draft TR on interoperability with C.

- name binding with BIND clause and character string, in interface body, EXTERNAL statements and COMMON blocks. Possibility of allowing a BIND clause on procedure definitions as well. Potential for conflict between procedure with BIND and another procedure without BIND.
- data type mappings in a standard module (e.g. ISO_C_KINDS); no remapping (as in HPF2 proposal), C's "void*" as an integer, address-of operator. Assumed-size/adjustable arrays match C's arrays (except for row versus column layout).
- procedure arguments: existing practice is directives or wrappers ("jacket" routines), but we could do this with dummy attributes or use HPF2's conversion to the "obvious" convention.
- header files; these provide (mostly) manifest constants (can be turned into Fortran PARAMETER) and macro definitions (would need to be function calls in Fortran). They are too hard for automatic conversion.

Richard Maine: Some of the suggestions could help calling Fortran from C as well (e.g. name-binding on procedure definitions). I did not like the idea of binding COMMON blocks to C external variables.
Jerry Wagener: We should be compatible with HPF2 - either feed back advised changes or pick up their approach. External C variables can be manipulated by C routines so we might not need to map COMMON to them. Recommend the effects of EXTRINSIC(C). Runtime conversion (in the HPF2 proposal) only occurs with the MAP_TO_ATTR directive, not by default.
Keith Bierman: Automatic conversion of C structures would be user-friendly (the padding rules might differ). Macro definitions in C header files are a real problem - no vendor has sufficient manpower to convert header files, and if a user did it it would be likely to go out-of-date quickly (next release of the C compiler).
Jerry Wagener: HPF2 only provides the MAP_TO directive, rather than functions at call sites.
Richard Maine: Automatic treatment of header files is all but impossible, because anything can appear there, even non-C code.
Kurt Hirchert: The "perl" community has been reaonably successful in transforming most C header files to perl header files with the h2ph tool. (This is an auxiliary tool, not in the language).
Richard Maine: So the standard need not address the issue if we had a tool that would make it feasible for the user to do the conversion themselves.
Stan Whitlock: The Fortran compiler needs to understand what the particular C compiler uses for name mapping, data type conversion, etc. Most efforts at mixing these are very low-level and compiler-specific. We should be very general and high-level and not make the user embed name mappings etc. in their programs.
Michael Hennecke: HPF2 does not provide mappings of basic types or say what happens if Fortran and C types are not actually the same. The "BIND" name mappings are the C name not the actual linkage editor name.

Janice Shepard: C pointer argument mapping opens up the ability to actually have the C pointer and walk through memory. This kills optimisation. The address-of function (LOC) must only be allowed in actual argument lists.
Dick Hendrickson: Why have C data kinds in a Fortran program - it seems awkward not to use the normal Fortran kinds for solving the user's program.
Richard Maine: We could forbid applying LOC to anything that is not a TARGET.
Jerry Wagener: The symbol for address-of in HPF2 (the at sign "@") was chosen instead of LOC because this character is not otherwise used in Fortran, and also because if LOC were an intrinsic function it would break Fortran semantics for INTENT(OUT) arguments.
Stan Whitlock: It is common in calls to C routines to have data returned through the argument list. The MAP_TO directive could really hurt here. VAX Fortran has always had the %LOC function that is only usable at the call site and users do not like to use it; we should do all this in the interface block.
Miles Ellis: How important is it to ensure that the Fortran TR and HPF2 are compatible?
Jerry Wagener: Maybe we should at least start with HPFF's proposal and see where we go?
Janice Shepard: It is better for HPFF to be following us; HPFF meeting votes do not always reflect opinions as many delegates abstain rather than vote against a proposal they are not in favour of.

Further discussion on the proposal occurred later in the meeting.

Michael Hennecke: Everyone is aware of HPFF's proposal and realises that we should take some notice of it, without being bound by it.
Jerry Wagener: We need to decide whether interoperability is both ways or only Fortran calling C.
Michael Hennecke: The proposal currently is only for Fortran calling C except where the features can be made symmetric.
Keith Bierman: The general case is more the scope of the C standard rather than Fortran.
Kurt Hirchert: We do actually want Fortran to be callable from C, not just for numerical libraries but also for callback functions. This should come free with the appropriate syntax.
Jon Steidel: The C standards committee are interested in this and are watching what we are doing.

6.3. Data Type Enhancements

Malcolm Cohen and Steve Morgan presented the proposal for a technical report on data type enhancements that extended the allocatable component scope to include allocatable dummy arguments and function results, and that included parameterized derived types.

Janice Shepard: At the Edinburgh meeting WG5 had decided that the interaction of parameterized derived types with object-oriented Fortran should be pursued, and that they were not ready for standardisation until this was resolved. There were many technical flaws in the previous version of this proposal.
Jerry Wagener: The proposal is reasonably well-developed but there are several areas with problems.. Perhaps we can split the parameterised derived type into two smaller stages, one for KIND type parameters, the other being the (perhaps more important) LEN-type parameters.
Wolfgang Walter: The KIND type parameter is more important for the user, so that the type can be generic over all representation methods.
Miles Ellis: Kind type parameters are certainly more important for library and module vendors.
Keith Bierman: The lack of parameterized derived types is certainly a "hole" in the language, but the lack of derived-type i/o is a far bigger and more important one.
Stan Whitlock: It is hard to be convinced of the pressing need for parameterized derived types; we need examples of "killer apps".
John Reid: Perhaps a simplified proposal with only a single kind type parameter could be completed in a reasonable time.

The discussion was resumed later in the meeting.

Jerry Wagener: The allocatable section of the proposal is further along, and since parameterized derived types is controversial it would unduly delay the whole technical report should it be included. The scope of this technical report should be limited to allocatable components (but including

allocatable dummy arguments and function results) and perhaps another group should investigate how to pursue parameterized derived types and report back to the next meeting.
Miles Ellis: I agree that these should probably be decoupled. Steve Morgan could head a group to either produce a further technical report or report on the correct course for furtherance of parameterized derived types.
Richard Maine: SC22 seem to want a decision now as to whether this technical report should include parameterized derived types. It would not be responsive to SC22 to put off that decision until the next meeting.
Steve Morgan: It is not necessary to split the technical report at this time. We should pursue parameterized derived types as long as it is possible. The changes necessary for parameterized derived types are no more extensive in the standard than those for allocatable components.
Wolfgang Walter: Allocatable components is too trivial for the title “Data Type Enhancements”.
Jerry Wagener: The parameterized derived type proposal would not be dead if it is removed from this technical report. Not only is it a U.S. requirement for Fortran 2000 but there is also the middle route of a fourth technical report. The U.S. is happy to have this possibility examined at the next meeting.
Kurt Hirchert: Both allocatable components and parameterized derived types are necessary, but the latter will take much longer. Parameterized derived types will not be materially delayed by deferring action on them until after our next meeting and the SC22 plenary next year, but allocatable components can go ahead much sooner.
Miles Ellis: Parameterized derived types do not seem to fit our technical report criteria at the moment.
Richard Maine: The issue is not whether there is a hole in the language - most of the proposed Fortran 2000 requirements are to fix holes in the language. The issue is whether parameterized derived types should be in this particular report.
Michael Hennecke: Allocatable components seem ready to go, we should not delay them for another year.
Steve Morgan: The whole technical report as it stands now should be proceeded with.

SV: PDT should be included in this report?/excluded, investigate new TR in Dresden?/excluded, F2000 requirement only?/undecided? = 3/18/1/2.

SV: Whether allocatable dummy arguments and function results should be in this report? 23/0/0.

7. Requirements for Fortran 2000 [N1144,N1145]

Janice Shepard said that we should not even consider items that are not already in our repository of requirements. Jerry Wagener noted that there was insufficient time to discuss any item in depth, and proposed that a straw vote of the repository items below be taken to establish which items currently had widespread support for inclusion in Fortran 2000 and which items had little support; this would establish a priority ordering for the development body to begin work on and also reduce the number of requirements to a more manageable level. The convenor concurred and the results of the straw vote, where on each item every member either abstained or ranked it as high priority, low priority or undesirable, is recorded below. The “score” column reflects the chosen scoring metric that counted +2 for a “high” vote, +1 for a “low” vote, -1 for an “undesirable” vote and 0 for an abstention.

Description (Item no. in N1144)	Score	High Priority	Low Priority	Undesirable	Abstentions
Condition Handling (5)	31	10	12	1	2
Command line arguments (20)	25	11	6	3	3
Proc dependent features list (51)	12	5	6	4	10
Asynchronous I/O (52)	21	9	6	3	7
Derived-type I/O (17)	43	21	2	1	1
INTENT for POINTER args (44a)	12	3	11	5	6
Nesting of internal procedures (33)	11	3	11	6	5
Bit data type - string (21)	10	3	8	4	10
Compiler directives (19)	5	2	8	7	8
Modules with non-std intrinsics (46)	-5	0	4	9	12
Varying-length char up to max (34)	10	4	8	6	6
POSIX binding to Fortran 90 (47)	12	1	12	2	10
Object-oriented Fortran (18)	17	5	10	3	7
Variable repeat format-specs (48)	7	2	10	7	6
Specifying default precision (49)	22	9	5	1	9
Unsigned integer data type (37)	9	5	3	4	12
Handling of pointer arguments (44)	-2	0	1	3	20
Pointers to procedures (43)	23	9	8	3	5
More than 7 array dimensions (24)	9	5	6	7	7
Unlimited statement length (50)	14	6	8	6	5
Private/shared data in parallel proc.s (53)	14	5	7	3	10
Internal procedures as actual args (42)	-2	4	2	12	7
Renaming of defined operators (41)	9	2	8	3	12

8. Update of WG5 Strategic Plan

The convenor lead a discussion of the strategic plan and the revision thereof proposed by the “policy” subgroup.

SV: Whether new parts of the Fortran standard should be optional by default? 21/0/3.

SV: Whether we should invite X3J3 to be the development body for F2000? 10/0/1 (11 abstentions).

9. Any Other Technical Items

9.1. Conditional Compilation [N1138,N1139]

David Epstein presented proposals for a conditional compilation facility.

Miles Ellis: I have never seen this requested before, why do we need it now?

Keith Bierman: This ought to be brought forward as a requirement.

Michael Hennecke: It is needed for KIND values.

Richard Bleikamp: It is existing practice and should be standardised.

Stan Whitlock: Users are not using complicated expressions, so the C syntax of cpp is not an issue.

David Muxworthy: The demand in the U.K. is for the facility to be provided by cpp.

Jeanne Martin: A separate part of the standard may be a mandatory requirement - this is the usual case in other standards. Conditional compilation is not difficult to implement and support. Users would like Fortran syntax.

Stan Whitlock: In the PC world, Microsoft have a facility that is not based on cpp. Thus in a mixed environment, users need to change their programs (that use conditional compilation) when moving between mainframes, workstations and PCs. The proliferation of differing facilities is painful for users and vendors alike. There is no need to put it in the Fortran standard, just point to cpp.

Janice Shepard: We should standardise a facility that is suited to the Fortran programmer.

Kurt Hirchert: One reasonable approach would be to start with cpp and alter it not to mangle Fortran, alternatively we could start with a clean slate and build something that is Fortran specific.

Miles Ellis: If we follow the cpp route, we must settle the issue of reference versus inclusion: i.e. if cpp changes in a new revision of the C standard, should we follow? We ought not to make a separate part of the Fortran standard, that is produced at a later time than the main part, mandatory; we should put everything mandatory in the main part.
Keith Bierman: There are tens of millions of lines of Fortran that use a subset of cpp. It is a disservice to the users to require them to rewrite their code - or worse, mix different preprocessors.
Michael Hennecke: A conditional compilation facility must be mandatory so that it will be portable.
Tony Warnock: We should do what users want, not what they are currently forced to use. The use of cpp is not very portable - they only use it because there is no alternative. Users would prefer something they can understand.
Jerry Wagener: The U.S. proposal is that there should be an optional part of the standard that defines a preprocessor.

SV: Do we want to do something ourselves about this? 23/1/1.
SV: Optional part of standard/requirement for F2000? 21/1/4.
SV: cpp/fcc/other? 1/13/10.

Some reservations were expressed over the meaning of the final straw vote above, and the discussion was continued later in the meeting.

Janice Shepard: We agreed yesterday that all additional parts of the Fortran standard would not be mandatory until Fortran 2000. The facility as proposed does not handle Fortran INCLUDE lines.
David Epstein: This is meant to be mandatory, and according to our straw vote earlier, not based on cpp.
Stan Whitlock: We could ask for two approaches to be prepared, one cpp-like, the other different.
Richard Maine: We do not have consensus to the proposal in S46 or to any other proposal.
Jeanne Martin: Normally separate parts of a standard are mandatory.

SV: Decide now on direction (cpp/other)?/Decide at next meeting?/undecided? 12/11/3.

The convenor noted that this vote meant that we did not have consensus to reach a decision on direction now, and that we would therefore be deferring the issue until the next meeting.

SV: Whether it should be a separate part of the standard? 23/0/2.

9.2. Interval Arithmetic [N1148]

Baker Kearfott presented the case for interval arithmetic and proposed that it should be defined in a separate part of the Fortran standard.

Stan Whitlock: Should this be defined for single precision, double precision, or all (real) precisions? The interaction with COMPLEX also needs to be decided.
Baker Kearfott: It would be acceptable if the interval data type were only defined for double precision.
Keith Bierman: Just like COMPLEX, it should be defined for all precisions.
Wolfgang Walter: COMPLEX intervals would be useful in the future but there is no real need for them now.
Richard Maine: Is a representation of infinity required (e.g. for handling division of an interval containing zero)?
Answer: No, but can be very useful in some contexts.
Jerry Wagener: Could we provide intervals as a KIND of REAL?
Answer: No, because some of the rules of arithmetic are different.
Wolfgang Walter: If we provide intervals via a derived type, kind type parameterization of derived types is required.

SV: Whether interval arithmetic should (eventually) be added in some way? 7/4/13.
SV: If so, requirement for F2000?/separate part?/TR?/undecided? 4/17/0/3.
SV: Provide an interval arithmetic module as an informative annex? 13/3/6.

Wolfgang Walter: We have been waiting for interval arithmetic for 20 years now and it would be very useful. There are problems being solved right now that are impossible with normal floating-point arithmetic.
Janice Shepard: We should investigate this further and bring a more concrete proposal to the next meeting.

SV: Begin preparing an additional part of the standard now? 12/4/11.

John Reid: We should address what is needed for interval arithmetic and not put it into the language itself.
Keith Bierman: To get it sufficiently fast to be useful it needs to be in the language, it cannot be done by the user.
Janice Shepard: It is not clear that putting it into the language is necessary, user pressure will get the performance (e.g. from a standard module) because the vendors want to make money.
Tom Lahey: We should see what it would look like as an extension module. It the module runs fast enough we then do not need to put it into the language. If the module does not run fast enough it should show us how we need to extend the language.
Steve Morgan: There is clear evidence of user demand for this - a petition was circulated to this meeting.
Stan Whitlock: On the contrary, no user has come to a vendor with money and asked for the facility.
Jerry Wagener: We have not seen this proposal for long enough to make such a significant decision for the future of Fortran. We should not act precipitously - national bodies have not had time to discuss and settle a position before the meeting.
Michael Hennecke: I would like to see a more mature document at the next meeting before deciding what to do with it.
Miles Ellis: The proposal is indeed only very recent, and only just made the deadline for the agenda. We cannot expect people to read and understand such a complicated proposal and make an informed decision without more time.

SV: Whether to invite a body to produce proposals for the next meeting? 24/0/2.

10. Closing Business

10.1. Future Meetings

Wolfgang Walter requested a straw vote on what dates would be most acceptable for the next meeting.
SV: End of July?/Beginning of September?/Any? = 8/2/14.
(Only one person would be unable to attend a July meeting, and two people would be unable to attend a September meeting).

Wolfgang Walter invited WG5 on behalf of Dresden University and DIN to the next meeting which in accordance to the wishes expressed above would be held on 22nd-26th July 1996 in Dresden.

10.2. Electronic Distribution

Paper S44 (N1154) by Miles Ellis on future electronic distribution of WG5 documents was discussed.

Keith Bierman: The text form of a document is useful since Acrobat won't run on a palmtop.
Kurt Hirchert: It would be nice if documents had the document number on each page and if each page were numbered, preferably in the style "Page N of M".
Miles Ellis: Authors submitting PostScript documents should format them in this way; document numbers are available on request.

10.3. Any other business

Kurt Hirchert offered to set up email reflectors for the subgroups working on the Technical Reports.

11. Adoption of San Diego Resolutions

All resolutions of the meeting were adopted unanimously.

12. Adjournment

The meeting was adjourned at 5pm on Friday 10th November 1995.