J3 / 99-221r1

NCITS J3 ANSI Fortran Standards Committee   Craig T. Dedo
J3 Responses to Interpretation Requests from Japan   August 25, 1999
Page 1 of 12

J3 Responses to Interpretation Requests from Japan

J3 has looked at the interpretation requests of Japan in J3 paper 99-208. We have put these requests into the following classes:

- "Fortran 95 Interpretation" - JOR subgroup will submit this item as a formal interpretation.
- "Recommended for Fortran 2000" - These items are problems, but they are not serious enough to warrant a formal Fortran 95 Interpretation.
- "Deferred to Editor" - We will recommend that the editor of Fortran 2000 include these changes into the text of the Fortran 2000 draft.
- "Not Recommended" - We believe that no action is required on these items. An explanation is provided.

Here are the responses of J3, which are indicated by a ">>" symbol at the beginning of the line.

All references are to Fortran 95.

J3 thanks the ITSCJ for their careful reading of Fortran 95. Please feel free to contact J3 if you disagree with any of these recommendations.

JP-1)
 In 2.5.7 Intrinsic, the third sentence:

 "All intrinsic data types, procedures, and operators may be used in
  any scoping unit without further definition or specification."

 Insert 'assignments' after 'procedures,'.

>> Recommended for Fortran 2000.

JP-2)
 In the sentence immediately after NOTE 4.18:

 "If initialization-expr appears for a nonpointer component, that
  component in any object of the type is initially defined or becomes
  defined as specified in (14.7.5) with the value determined from
  initialization-expr."

 Change '(14.7.5)' to '14.7.5'.

>> Recommended for Fortran 2000.

JP-3)
 4.4.4 Construction of derived-type values
 Before NOTE 4.33:

 "A structure constructor whose component values are all constant
  expression is a derived-type constant expression."

 In the above, 'derived-type constant expression' should be bold,
 because the term is defined here.

>> Deferred to editor.

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

J3 / 99-221r1
Craig T. Dedo
August 25, 1999
Page 2 of 12

1 JP-4)
2 4.4.4 Construction of derived-type values
3 Before NOTE 4.34:
4 "Where a component in the derived type is a pointer, the
5 corresponding constructor expression shall evaluate to an object that
6 would be an allowable target for such a pointer in a pointer
7 assignment statement(7.5.2)."

8 Change 'an object' to 'a result value'. A value of an expression can
9 not be an object, by definition.

10 >> Fortran 95 Interpretation.

11 JP-5)
12 4.5 Construction of array values,

13 "The ac-do-variable of an ac-implied-do that is in another
14 ac-implied-do shall not appear as the ac-do-variable of the containing
15 ac-implied-do."

16 This sentence should be a Constraint.

17 >> Fortran 95 Interpretation.

18 JP-6)
19 5.1 Type declaration statements

20 As for the 16th and 19th constraint after R506:

21 the 16th:
22 "Constraint: The function-name shall be the name of an external
23 function, an intrinsic function, a function dummy procedure, or a
24 statement function."

25 Because the syntactic class 'object-name' is only defined as a
26 'name' in the standard, the following constraint should be added here:
27 Constraint: The object-name shall be the name of a data object.

28 After that, in the 19th:
29 "Constraint: initialization shall not appear if object-name is dummy
30 argument, a dummy argument, a function result, an object in a named
31 common block unless the type declaration is in a block data program
32 unit, an object in blank common, an allocatable array, an external
33 name, an intrinsic name, or an automatic object."

34 In the above, 'a function result,' should be removed.

35 If we can not add the constraint above, 'a statement function' should
36 be added in the 19th constraint.

37 >> Fortran 95 Interpretation.

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

J3 / 99-221r1
Craig T. Dedo
August 25, 1999
Page 3 of 12

1   JP-7) is absent

2   JP-8)
3   5.1 Type declaration statements

4   After NOTE 5.3:

5   "If a length-selector (5.1.1.5) is a nonconstant expression, ..."

6   Change 'length-selector' to 'char-selector', 'char-len-selector' or
7   'character-length'.

8   >> Fortran 95 Interpretation.

9   JP-9)
10   5.1.1.5 4th Constraint after R510 (Page 51 Line 1) states that:

11   "Constraint: A function name declared with an asterisk
12   char-len-param-value shall not be array-valued,
13   recursive, or pure."

14   The word "Constraint" should be shown in distinguishing type font
15   because this is a constraint for an obsolescent feature.

16   >> Recommended for Fortran 2000.

17   JP-10)
18   5.1.1.5 After the NOTE 5.6 (Page 51 Line 32,33) states that:

19   "The length specified for a character-valued statement function or
20   statement function dummy argument of type character shall be a
21   constant specification expression."

22   This should be shown as a constraint because it is a restriction
23   for character length.

24   Note that this is an obsolescent feature.

25   >> Recommended for Fortran 2000.

26   JP-11)
27   5.1.2.4.1 After R516 (Page 54 Line 29-33) states that:

28   "Constraint: An explicit-shape array whose bounds depend on the
29   values of nonconstant expressions shall be a dummy
30   argument, a function result, or an automatic array
31   of a procedure.

32   An automatic array is an explicit-shape array that is declared in
33   a subprogram, is not a dummy argument, and has bounds that are
34   nonconstant specification expressions."

1 The constraint seems meaningless because the following paragraph
2 which defines the automatic array duplicates it.
3 Is this constraint necessary ?

4 >> Not Recommended. Yes, we believe that the constraint is necessary because it belongs to three
5 things, not just automatic arrays. No change is necessary.

6 JP-12)
7 5.1.2.4.3 (2) after R518 (Page 55 Line 41) states that:

8 "(2) They are specified in a pointer assignment statement. ..."

9 In this description, the term "pointer assignment statement" should
10 be changed to "pointer assignment".

11 Reason : The bounds of each dimension of an array pointer may be
12 specified not only in a pointer assignment statement but also in a
13 derived-type intrinsic assignment statement with a component of an
14 array pointer.

15 >> Fortran 95 Interpretation.

16 JP-13)
17 5.2.10 5th and 9th constraints after R537 (Page 62 Line 2 and 10)
18 state that:

19 "Constraint: A scalar-int-expr of a data-implied-do shall involve
20 as primaries only constants, ... , or
21 DO variables of the containing data-implied-dos, ..."

22 "Constraint: In an array-element or a scalar-structure-component
23 that is a data-i-do-object, any subscript shall be an
24 ... ,or DO variables of the containing
25 data-implied-dos, ..."

26 In the latter constraint, the phrase :
27 "DO variables of the containing data-implied-dos"
28 should be changed to :
29 "DO variables of this data-implied-do and the containing
30 data-implied-dos".

31 Consider the following program:

32 INTEGER, DIMENSION (3,3) :: IARY
33 DATA ((IARY(IA,JA), JA=IA,3), IA=1,3) /1,2,3,4,5,6/

34 The "IA" in "JA=IA,3" is a scalar-int-expr described in the former
35 constraint stated above. In this case, "IA" is a DO variable of the
36 containing data-implied-do.
37 This is the meaning of the phrase "containing data-implied-do".
38 In another word, "containing" should mean "outer".

The "IA" and "JA" in "IARY(IA,JA)" are subscripts described in the
latter constraint stated above. In this case, "IA" is a DO variable
of the containing data-implied-do. However, "JA" is a DO variable
of not "containing" but "this" data-implied-do.

If the "containing data-implied-do" means both "this" and "outer"
data-implied-do, the former constraint is incorrect because
it allows the DO variable of this data-implied-do
(JA of "JA=IA,3" in this example)
to be involved in scalar-int-expr
(IA and 3 of "JA=IA,3" in this example).

>> Recommended for Fortran 2000.

JP-14)
5.2.10 R539 (Page 62 Line 13,14) states that:

"R539 data-stmt-repeat is scalar-int-constant
                    or scalar-int-constant-subobject"

The syntactic definition of scalar-int-constant-subobject can be
derived from int-constant-subobject (1.6.3).
Add the definition of int-constant-subobject.

>> A *constant-subobject* is a subobject of a named constant. We will add clarifying text to Fortran 2000.

JP-15)
5.2.10 3rd paragraph after R540 and constraints (Page 62
Line 43-45) states that:

"A zero-sized array or an implied-DO list with an iteration count
 of zero contributes no variables to the expanded sequence of
 variables, but a zero-length scalar character variable does
 contribute a variable to the list."

The word "list" at the end of above statement should be replaced
with "sequence" or "expanded sequence".

Note that the words "list" and "sequence" are used for the
separate meanings in this section as follows :

"The data-stmt-object-list is expanded to form a sequence of
 pointers and ..."
"The data-stmt-value-list is expanded to form a sequence of
 data-stmt-constants."

Also refer to 2.5.9 for the definition of "sequence".

>> Recommended for Fortran 2000.

JP-16)
5.4 2nd paragraph after R545 and constraints (Page 66 Line 11)

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

J3 / 99-221r1
Craig T. Dedo
August 25, 1999
Page 6 of 12

1   states that:

2   "Any namelist-group-name may occur in more than one NAMELIST
3   statement in a scoping unit."

4   Can a namelist-group-name occur more than once in one NAMELIST
5   statement ?

6   Is the following NAMELIST statement standard conforming ?

7   NAMELIST /NLIST/ A, B   /NLIST/ C, D

8   If this is standard conforming, is it the same as the
9   following ?

10   NAMELIST /NLIST/ A, B, C, D

11   >> Fortran 95 Interpretation.

12   JP-17)
13   5.4 3rd paragraph after R545 and constraints states that:

14   "A namelist group object may be a member of more than one
15   namelist group."

16   Can a namelist group object occur more than once in one
17   namelist group?

18   Is the following NAMELIST statement standard conforming?

19   NAMELIST /NLIST/A,B,A

20   >> Fortran 95 Interpretation.

21   JP-18)
22   In 7.1.1.1, a constraint below R702 states that:

23   "subobject shall be a subobject designator whose parent is a constant.
24   A variable that is a primary shall not be a whole assumed-size array."

25   The second sentence,'A variable that ...',should be
26   'Constraint: A variable that...'.

27   >> Recommended for Fortran 2000.

28   JP-19)
29   The third paragraph of 7.1.4.1 has the following:

30   "If a pointer appears as one of the following, the associated target
31   object is referenced:
32   (1) A primary in an intrinsic or defined operation,
33   (2) As the expr of a parenthesized primary, or

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

J3 / 99-221r1
Craig T. Dedo
August 25, 1999
Page 7 of 12

1   (3) As the only primary on the right-hand side of an intrinsic
2   assignment statement."

3   The first word "As" in (2) and (3) should be removed.

4   >> Deferred to editor.

5   JP-20)
6   In 7.5.3.2, the paragraph before NOTE 7.49 states that

7   "A statement that is part of a where-body-construct shall not be a
8   branch target statement."

9   The term 'branch target statement' is defined to be one of a specific
10   set of statements, and is not appropriate here. Replace this with a
11   constraint such as:

12   "Constraint: A statement label of any statement that is part of a
13   where-body-construct shall not be referred to from outside of the
14   construct."

15   The same applies to 7.5.4.1 (FORALL).

16   >> Recommended for Fortran 2000.

17   JP-21)
18   NOTE 7.49 has the following example:

19   WHERE (A > 0.0)
20   A = LOG (A)
21   A = A / SUM (LOG (A))

22   END WHERE

23   But SUM(LOG(A)) causes an error when the array variable A has
24   negative values. So the example program should be changed to
25   something like the following.

26   WHERE (D > 0.0)
27   X = -B + SQRT (D)
28   R = D / SUM (SQRT (ABS (D)))

29   END WHERE

30   >> Not Recommended. As indicated in the comments in the example, it is to show that LOG(A) will be
31   referenced even if that was not the intent of the programmer.

32   JP-22) is absent

33   JP-23)
34   In NOTE 7.55, the "END FORALL" statement should be supplied.

J3 / 99-221r1

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

Craig T. Dedo
August 25, 1999
Page 8 of 12

1 >> Not Recommended. The purpose of the example was to illustrate the FORALL mask and the ellipses
2 indicate the continuation of the rest of the block.

3 JP-24)
4   In 8.1.4.1.2, second constraint below R833 states that:

5   "The do-term-shared-stmt shall be identified with a label
6   and all of the label-do-stmts of the shared-term-do-construct
7   shall refer to the same label."

8   This implies a label-do-stmts of the outer-most outer-shared-
9   do-construct will permit not to refer to the same label, because
10   shared-term-do-construct does not include outer-most outer-
11   shared-do-construct.

12   So the term "shared-term-do-construct" should be changed to
13   "inner-share-do-construct and outer-shared-do-construct."

14 >> Fortran 95 Interpretation.

15 JP-25)
16   In the second sentence of 8.1.4.3:

17   "Once active, the DO construct becomes inactive only when the
18   construct it specifies is terminated(8.1.4.4.4)."

19   Remove "it specifies".

20 >> Deferred to editor.

21 JP-26) is absent

22 JP-27)
23   In the second sentence from the bottom of 8.2:

24   "It is permissible to branch to an end-do-stmt or a do-term-action-
25   stmt only from within its DO construct."

26   "end-do-stmt" should be "end-do".

27 >> Recommended for Fortran 2000.

28 JP-28)
29   The last sentence of 8.2.1:

30   "Only branch target statements (8.2), FORMAT statements, and DO
31   terminations shall be referred to by the use of statement labels
32   (3.2.4)."

33   "(3.2.4)" should be moved to the end of the previous sentence.

34 >> Recommended for Fortran 2000.

J3 / 99-221r1

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

Craig T. Dedo
August 25, 1999
Page 9 of 12

1 JP-29)
2   The last sentence of 8.2.1:

3   "Only branch target statements (8.2), FORMAT statements, and DO
4   terminations shall be referred to by the use of statement labels
5   (3.2.4)."

6   Change ", FORMAT statements, and DO terminations" to "and FORMAT
7   statements". The reasons are as follows:

8   1. All cases are covered without "DO terminations".
9   2. A DO termination can be a shared-term-do-construct. A statement
10   label is intended to refer to a statement, and the notion of
11   referring to a construct by a statement label is not defined.

12 >> Recommended for Fortran 2000.

13 JP-30)
14   The first sentence of 9.4.1.7:

15   "If an end-of-record condition (9.4.3) occurs and no error condition
16   (9.4.3) occurs during execution of an input/output statement that
17   contains an EOR= specifier ..."

18   "input/output statement" should be "input statement".

19 >> Recommended for Fortran 2000.

20 JP-31)
21   The fourth sentence of 10.8 and sixth sentence of 10.9:

22   "Each value is either a null value or one of the forms:
23   c
24   r*c
25   r*
26   where c is a literal constant or a nondelimited character constant
27   and r is an unsigned, nonzero, integer literal constant."

28   "a literal constant" should be "an optionally signed literal constant"

29 >> Fortran 95 Interpretation. Add the phrase, "if integer or real" to that last phrase above.

30 JP-32)
31   The first sentence of NOTE 10.25:

32   "List-directed input/output allows data editing according to the type of
33   the list item instead of by a format specifier."

34   "format specifier" should be "format specification".

35 >> Recommended for Fortran 2000.

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

J3 / 99-221r1
Craig T. Dedo
August 25, 1999
Page 10 of 12

JP-33)
Page 176/ line 16 (2nd paragraph before 10.8.1.1):

"(3) The first nonblank character"
should be
"(3) The first character".

REASON:
Since (1) says that "the character sequence does not contain the value
separators blank, comma, or slash," the first character should not be
a blank. So the word "nonblank" is confusing.

>> Recommended for Fortran 2000.

JP-34)
Page 177/ line 35 (6th line of 10.8.2):

The word "sequence" in "not occur within a constant or sequence"
should be "character sequence".

>> Recommended for Fortran 2000. Rewrite [177:34-36] to read, "The processor may begin new records
as necessary, but the end of record shall not occur within a constant except for complex constants and
character sequences. The processor shall not insert blanks within a constant or character sequence."
Check NAMELIST output in section 10.9.2.

JP-35) is absent

JP-36)
Page 192/ line 17 (The last line of 12.2.1.1):

In dummy data objects, the size is allowed to be assumed.
However the characters of "size," is still small.
It should be changed to normal size.

>> Deferred to editor.

JP-37)
Page 196/ line 3 (The first statement of 12.3.2.1.1):

The referring section "(12.4)" is wrong, and should be
"(7.1.3, 7.3)".

>> Recommended for Fortran 2000.

JP-38) is absent

JP-39)
Page 204/ line 17 (NOTE 12.20):

In 6.3.3.2, "If a pointer is currently associated with an
allocatable array, the pointer shall not be deallocated".
So "DEALLOCATE (B)" would NOT be permitted.

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

1 >> Fortran 95 Interpretation.

2 JP-40) is absent

3 JP-41)
4 p.278, in the item (3) of 14.1.2.4:

5 (3) A procedure is not established in a scoping unit if it is neither
6 established to be generic nor established to be specific.

7 Change "procedure" to "procedure name".

8 >> Deferred to editor.

9 JP-42)
10 p.289, in the item (6) of 14.7.5:

11 (6) A reference to a procedure causes the entire dummy argument
12 data object to become defined if the entire corresponding actual
13 argument is defined <obsolescent>with a value that is not a
14 statement label.</obsolescent>

15 Delete the obsolescent-font part.

16 >> Recommended for Fortran 2000.

17 JP-43)
18 p.304, in the following part of B.1.1:

19 "R901 io-implied-do-control is do-variable = scalar-numeric-expr ,
20 ..."

21 Change "R901" to "R918".

22 >> Deferred to editor.

23 JP-44)
24 p.307, in the following part of B.2.6:

25 ... keyboards with screen displays, it is an unnecessary overhead, and is
26 potentially error-prone, to have to locate positions 6, 7, or 72 on a line.

27 Change "72" to "73".

28 >> Deferred to editor.

29 JP-45)
30 p.311, in the following part of C.1.2:

31 ... an exponent range from 10**-50 to 10**50.

32 Change "exponent range" to "range" or "value range".

NCITS J3 ANSI Fortran Standards Committee
J3 Responses to Interpretation Requests from Japan

1   >> Deferred to editor.

2   JP-46)
3   p.311, in the following line of the program in C.1.3:

4   CURRENT => CURRENT % NEXT_CELL

5   Change "CELL" to "NODE".

6   >> Deferred to editor.

7   JP-47)
8   p.317, in the following line of the program in C.4.6:

9   X(2:N-1) = (X(1:N-2) + 2*X(2:N-1) + X(3:N+1)) / 4

10 Change "X(3:N+1)" to "X(3:N)".

11 >> Deferred to editor.

12 JP-48)
13 p.342, in the following sentence in C.11.2.3.2:

14 This decision will be recorded as the true elements of an array FLIP.

15 Change "FLIP" to "FLIPS".

16 >> Deferred to editor.