Subject: Correct C1107 and definition of ultimate components

From: Van Snyder

1 1 Introduction

- 2 C1107 [250:23-25] refers to "... of a type for which component-initialization is specified...." This doesn't
- 3 cover the case of a component of a component of ..., but almost certainly should. The term "subcompo-
- 4 nent" is wrong because it extends to components of allocatable components, and "ultimate component"
- 5 is wrong because it excludes components of derived type. What's needed is to resurrect the definition
- 6 direct component from Fortran 95 but done correctly. By "direct" the Fortran 95 standard clearly
- 7 didn't mean "within a single type definition." Rather, it meant "the processor doesn't have to chase a
- 8 pointer, even one disguised as allocatable."
- 9 The definition of "ultimate components" is defective by not carrying through its recursion correctly;
- 10 indeed, since ultimate components are of intrinsic type, there is no hope for it to do so. That is an
- 11 additional reason to resurrect "direct components." Having a correct definition for direct components,
- 12 constructing a correct definition for ultimate components is trivial.
- 13 Should this be an interp?

14 2 Edits

20

21

22

- 15 Edits refer to 04-007. Page and line numbers are displayed in the margin. Absent other instructions, a
- 16 page and line number or line number range implies all of the indicated text is to be replaced by associated
- 17 text, while a page and line number followed by + (-) indicates that associated text is to be inserted after
- 18 (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

19 The direct components of an object of derived type are

44:24-26

- (1) components declared in its type definition, and
- (2) components declared in the type definitions of its nonpointer nonallocatable direct components of derived type.
- The ultimate components of an object are its direct components that are of intrinsic type or that have the POINTER or ALLOCATABLE attribute.
- 25 C1107 (R1104) If a direct component (4.5) of a variable declared in the specification part of a module 250:23-25 has component initialization, the variable shall have the SAVE attribute.

27 **As an interp request**

```
28 NUMBER: TBD
```

- 29 TITLE: Default initialization of module variables and SAVE, ultimate components
- 30 KEYWORDS: DEFAULT INITIALIZATION, MODULE VARIABLES, SAVE, ultimate components
- 31 DEFECT TYPE: Erratum
- 2 STATUS: TBD
- 33 QUESTION 1a:
- 34 Does the following program conform to the Fortran 2003 standard?

```
module M
35
36
      type :: T1
37
        integer :: A = 0
      end type T1
38
      type :: T2
39
40
        type(t1) :: B
      end type T2
41
42
      type(t2) :: X
```

3 April 2006 Page 1 of 5

```
1 end module M
```

- 2 QUESTION 1b:
- 3 Alternative question: Add the ALLOCATABLE attribute to the B component of type T2.
- 4 ASWER 1:
- 5 The answer depends upon which way one interprets C1107: either "type for which component initializa-
- 6 tion is specified" applies only to components declared in the type of the object (1a), or to components
- 7 "at all levels of component selection" (1b).
- 8 ANSWER 1a:
- 9 Unfortunately, this program conforms, although it was intended that it not conform. The problem is
- 10 that C1107 [250:23-25] is defective: it applies only to components declared within the object's type
- 1 definition; therefore component initialization is not specified for type T2. Edits are provided to correct
- 12 this defect.
- 13 ANSWER 1b:
- 14 Unfortunately, this program does not conform, although it was intended that it conform. The problem
- 15 is that C1107 [250:23-25] is defective: it applies to components at all levels of component selection;
- 16 therefore, component initialization is specified for type T2. Edits are provided to correct this defect.
- 17 QUESTION 2:
- 18 Does the following program conform to the Fortran 2003 standard?

```
program P
19
20
      type :: T1
        integer, pointer :: A1
21
      end type T1
22
23
      type :: T2
24
        type(t1) :: A2
        integer :: B = 2
25
26
      end type T2
      type :: T3
27
        type(t2) :: B2
28
        integer :: C = 3
29
      end type T3
30
      type(t3) :: X
31
32
      integer, target :: Y
      x\%b2\%a2\%a1 => y
33
34
      print *, x
   end program P
35
```

- 36 ANSWER 2:
- 37 Unfortunately, this program conforms, although it was intended that it not conform. The problem is
- 38 the use of "ultimate components" in the second paragraph after NOTE 9.35 in subclause 9.5.2, together
- 39 with a defective definition for "ultimate components". Edits are provided to correct this defect.
- 40 QUESTION 3:
- 41 Does the following program conform to the 2003 standard:

```
42 program E
43    type :: T1
44    sequence
45    integer, allocatable :: A1
46    end type T1
47    type :: T2
```

3 April 2006 Page 2 of 5

```
sequence
1
2
        type(t1) :: A2
3
        integer :: B = 2
      end type T2
4
5
      type :: T3
        sequence
6
7
        type(t2) :: B2
8
        integer :: C = 3
      end type T3
9
10
      type(t3) :: X, Y
      equivalence ( X, Y )
11
    end program E
   ANSWER 3:
13
    Unfortunately, this program conforms, although it was intended that it not conform. The problem is the
    use of "ultimate components" in C576 [96:7-8]. The subobjects X%B2%A2%A1 and Y%B2%A2%A1
15
    are not ultimate components. If the A1 component of type T1 were declared within T2, the allocatable
16
    subobjects X%B2%A1 and Y%B2%A1 would be ultimate components of X and Y, and the program
17
    would not conform. Edits are provided to correct this defect.
18
    QUESTION 4:
19
   In the following program
20
    program P
21
22
      type :: T1
23
        integer, allocatable :: A1
24
      end type T1
      type :: T2
25
        type(t1) :: A2
26
27
        integer :: B = 2
      end type T2
28
      type :: T3
29
        type(t2) :: B2
30
        integer :: C = 3
31
      end type T3
32
      type(t3), allocatable :: X
33
34
      allocate (x)
35
    end program P
    what is the allocation status of X%B2%A2%A1 after execution of the allocate statement?
36
37
38
    Unfortunately, the standard does not provide an interpretation for this program. The last paragraph of
   6.3.1.1 [113:20-21] probably intended that X%B2%A2%A1 is unallocated after execution of the allocate
39
   statement, but because the definition of "ultimate components" extends through only two levels of
   component selection, it provides no interpretation. Edits are provided to correct this defect.
41
    QUESTION 5:
42
    Does the following program conform to the 2003 standard?
    program P
44
45
      type :: T1
        integer, allocatable :: A1
46
47
      end type T1
      type :: T2
48
```

3 April 2006 Page 3 of 5

```
type(t1) :: A2
1
        integer :: B = 2
2
3
      end type T2
      type :: T3
4
5
        type(t2) :: B2
        integer :: C = 3
6
7
      end type T3
8
      type(t3) :: X
      namelist /out/ x
9
10
      write ( *, out )
    end program P
11
   ANSWER 5:
    One might think that the program does not conform because because it violates the second paragraph
13
   of 9.5.3.6 Namelist formatting (allocatable components shall be processed by user-defined derived-
   type input/output subroutines). Unfortunately, it does not conform because the standard provides no
15
   interpretation for it (how does one output an unallocated variable?), because the definition of "ultimate
   components" extends through only two levels of component selection. Edits are provided to correct this
   defect.
18
    QUESTION 6:
19
   In the following program
20
21
    program P
      type :: T1
22
23
        integer, pointer :: A1 => NULL()
24
      end type T1
25
      type :: T2
        type(t1) :: A2
26
27
        integer :: B = 2
28
      end type T2
      type :: T3
29
        type(t2) :: B2
30
        integer :: C = 3
31
      end type T3
32
33
      type(t3) :: X
      call sub (x)
34
35
   contains
      subroutine SUB ( Y )
36
        type(t3), intent(out) :: Y
37
      end subroutine SUB
38
    end program P
39
    what is the pointer association status of X%B2%A2%A1 after execution of the CALL SUB statement?
40
    ANSWER 6:
41
    The standard does not provide an interpretation because the definition of "ultimate components" extends
42
    through only two levels of component selection. Edits are provided to correct this defect.
43
    QUESTION 7:
44
    In the following program
45
    program P
46
47
      type :: T1
48
        integer, pointer :: A1
```

3 April 2006 Page 4 of 5

```
end type T1
1
2
      type :: T2
        type(t1) :: A2
3
        integer :: B = 2
4
      end type T2
5
      type :: T3
6
7
        type(t2) :: B2
8
        integer :: C = 3
      end type T3
9
10
      type(t3) :: X
      nullify ( x%b2%a2%a1 )
11
      call sub (x)
12
   contains
13
14
      subroutine SUB (Y)
        type(t3), intent(out) :: Y
15
      end subroutine SUB
16
    end program P
17
18
    what is the pointer association status of X%B2%A2%A1 after execution of the CALL SUB statement?
    ANSWER 7:
19
    X%B2%A2%A1 remains disassociated because the definition of "ultimate components" extends through
20
    only two levels of component selection. It is clear, however, that the intent is that the pointer association
   status of X%B2%A2%A1 becomes undefined. Edits are provided to correct this defect.
22
   EDITS:
23
    The definition of "ultimate components" cannot be corrected in its current form because recursion via
24
    derived types is required, while the fundamental property of "ultimate components" is that they are
25
   of intrinsic type. As a result, the definition of "ultimate components" encompasses only two levels of
26
    derived-type components. Instead, the definition of "direct components" is resurrected from Fortran 95,
27
28
    after having been removed in Fortran 2003, (and corrected because it was defective in Fortran 95) and
    that definition used to correct both problems.
29
    The direct components of an object of derived type are
                                                                                                         44:24-26
30
                components declared in its type definition, and
31
          (2)
                components declared in the type definitions of its nonpointer nonallocatable direct compo-
32
33
                nents of derived type.
    The ultimate components of an object are its direct components that are of intrinsic type or that
34
    have the POINTER or ALLOCATABLE attribute.
35
    C1107 (R1104) If a direct component (4.5) of a variable declared in the specification part of a module 250:23-25
36
```

3 April 2006 Page 5 of 5

has component initialization, the variable shall have the SAVE attribute.

37

38

HISTORY:

SUBMITTED BY: Van Snyder