

Subject: Non-null initial targets for pointers
 From: Van Snyder
 References: 03-258r1, section 2.12.2, 04-202, 04-351, 04-387r1, 05-203, WG5/N1626-J3-018

1 Basic Functionality

2 Allow data and procedure pointers to have initial targets that are other than NULL().

2 Detailed Specification

4 Allow the initial target for a data pointer to be an accessible nonpointer nonallocatable variable that
 5 has the TARGET attribute, and has the SAVE attribute or is declared in the main program. Every
 6 subscript, section subscript, substring starting point, substring ending point, and type parameter value
 7 within the *variable* shall be an initialization expression.

8 Allow the initial target for a procedure pointer to be a suitable external, module, or specific intrinsic
 9 procedure.

10 The initial target shall satisfy all the requirements for pointer assignment (e.g. the TARGET attribute,
 11 type conformance, etc.).

12 This feature shall be available both for named pointers and for pointer components. Pointer components
 13 may be default initialized to have an initial target.

14 The target may be accessed by use or host association. If it is declared in the same scoping unit it need
 15 not have been previously declared; this facilitates initialization to a “sentinel” object. (See note 4.36 $\frac{1}{2}$
 16 in section 4 below.)

3 Syntax

18 Allow a *variable* to appear after => as initialization for pointer components or pointer variables. Allow a
 19 *variable* where a *data-stmt-constant* can appear, if it corresponds to a pointer variable. Allow a suitable
 20 procedure name to appear after => as initialization for procedure components.

4 Edits

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

26 [Editor: Add something about non-null initial targets for data and procedure pointers to the introduc- xiii
 27 tion.]

28 [Editor: Delete *component-initialization* (R444); it will reappear in subclause 4.5.3.4.] 50:13-14

29 [Editor: Insert a new paragraph after the first paragraph of 4.5.3.4 **Default initialization for compo- 53:5+
 30 nents:**]

31 A pointer variable or component is **data-pointer-initialization compatible** with a target if the pointer
 32 is type compatible with the target, they have the same rank, and the values of corresponding nondeferred
 33 type parameters are specified by initialization expressions that have the same value.

34 [Then re-create *component-initialization* (R444) with an additional right-hand side and constraints.
 35 Its related constraints are moved here because they need the definition of “data-pointer-initialization
 36 compatible” from the previous paragraph, so the syntax rule is moved here as well.]

37 R446 $\frac{1}{3}$ *component-initialization* **is** => *null-init*
 38 **or** => *initial-data-target*
 39

- 1 R446 $\frac{2}{3}$ *initial-data-target* is *variable*
 2 [Editor: Move C446 and C447 to here.]
 3 C452 $\frac{1}{3}$ (R446 $\frac{1}{3}$) The *component-name* shall be data-pointer-initialization compatible with the *initial-*
 4 *data-target*.
 5 C452 $\frac{2}{3}$ (R446 $\frac{2}{3}$) The *variable* shall be an initialization target.
-
- 6 [Editor: After the second paragraph of **4.5.3.4 Default initialization for components** insert new para- 53:7+ New ¶'s
 7 graphs:]
 8 A *variable* is an **initialization target** if it has the TARGET attribute, either has the SAVE attribute
 9 or is declared in the main program, does not have the ALLOCATABLE attribute, and every subscript,
 10 section subscript, substring starting point, and substring ending point in the variable is an initialization
 11 expression.
 12 If *initial-data-target* appears for a data pointer component, that component in any object of the type is
 13 initially associated with the target or becomes associated with the target as specified in 16.4.2.1.1.
 14 If *initial-proc-target* appears in *proc-decl* for a procedure pointer component, that component in any
 15 object of the type is initially associated with the target or becomes associated with the target as specified
 16 in 16.4.2.1.1.
-
- 17 [Editor: At the end of **4.5.3.4 Default initialization for components**, immediately before **4.5.3.5 Com- 54:1-
 18 ponent order** insert the following note, which illustrates that we should not require the non-null initial
 19 target of a pointer component to be previously declared.]

NOTE 4.36 $\frac{1}{2}$

A pointer component of a derived type may be default-initialized to have an initial target.

```

TYPE NODE
  INTEGER          :: VALUE = 0
  TYPE (NODE), POINTER :: NEXT_NODE => SENTINEL
END TYPE

TYPE(NODE), SAVE, TARGET :: SENTINEL

```

- 20 [Editor: Add a third right-hand side of *initialization* (R506):] 72:16+
 21 **or** => *initial-data-target*
 22 C505 $\frac{1}{3}$ (R506) If an *initial-data-target* is specified, the *object-name* shall be data-pointer-initialization
 23 compatible with it. (4.5.3.4).
-
- 24 [Editor: Within C525, “=> appears in *initialization*” => “*pointer-initialization* appears”.] 73:15
-
- 25 [Editor: Within **5.1 Type declaration statements** replace the paragraph that begins “If *initialization* is 74:33-34
 26 => ...”]
 27 If *null-init* appears, the initial association status of the object is disassociated. If *initial-data-target*
 28 appears, the object is initially associated with the target.
-
- 29 [Editor: Add a right-hand side after the *null-init* right-hand side of *data-stmt-constant* (R532):] 88:26+
 30 **or** *initial-data-target*
-
- 31 [Editor: Within the fifth paragraph of **5.2.5 DATA statement** — the one that begins “The expanded 89:12
 32 sequence ...” — “or *null-init*” => “, initial data target, or *null-init*” after “*null-init*”.]
-
- 33 [Editor: Within the sixth and seventh paragraphs of **5.2.5 DATA statement** — the ones that begin “A 89:14,16
 34 *data-stmt-constant* ...” — insert “or *initial-data-target*” after “*null-init*” twice.]
-
- 35 [Editor: Within the sixth paragraph of **5.2.5 DATA statement** — the first one that begins “A *data-* 89:15

1 *stmt-constant ...* — “The” ⇒ “If *data-stmt-constant* is *null-init*, the”. “pointer *data-stmt-object*”
 2 ⇒ “data statement object” because the data statement object is already required to be a pointer and
 3 syntax terms don’t have association status. Insert a new sentence at the end of the paragraph:]

4 If *data-stmt-constant* is *initial-data-target* the corresponding *data-stmt-object* shall be data-pointer-
 5 initialization compatible with the initial data target; the data statement object is initially associated
 6 with the target.

7 [Editor: Replace the third item in the numbered list in **7.1.7 Initialization expression:**] 126:27-29

- 8 (3) A structure constructor where each *component-spec* corresponding to
- 9 (a) An allocatable component is a reference to the intrinsic function NULL,
 - 10 (b) A pointer component is a reference to the intrinsic function NULL or an initialization
11 target, and
 - 12 (c) Any other component is an initialization expression,

13 [Editor: Replace *proc-decl* (R1214):] 264:19

14 R1214 *proc-decl* is *procedure-entity-name* [= > *proc-pointer-init*]
 15 R1214¹/₃ *proc-pointer-init* is *null-init*
 16 or *initial-proc-target*
 17 R1214³/₃ *initial-proc-target* is *procedure-name*

18 [Editor: After the fifth constraint after *interface-name* (R1215) — the one that begins “If => appears 264:30+
 19 ...” — insert a new constraint:]

20 C1216¹/₂ (R1214²/₃) The *procedure-name* shall be the name of an initialization target.

21 [Insert new paragraph] 265:15-

22 A procedure is an initialization target if it is a nonelemental external or module procedure, or a specific
 23 intrinsic function listed in 13.6 and not marked with a bullet (●).

24 [Editor: Replace the fifth paragraph of **12.3.2.3** — the one that begins “If => appears ...”:] 265:15-18

25 If => appears in a *proc-decl* in a *procedure-declaration-stmt* it specifies the initial association status
 26 of the corresponding procedure entity, and implies the SAVE attribute. The SAVE attribute may be
 27 confirmed by explicit use of the SAVE attribute in the *procedure-declaration-stmt*, by inclusion of the
 28 procedure entity name in a SAVE statement (5.2.12), or by the appearance of a SAVE statement without
 29 a *saved-entity-list* in the same scoping unit. If => *null-init* appears, the procedure entity is initially
 30 disassociated. If => *initial-proc-target* appears, the procedure entity is initially associated with the
 31 target.

32 If *proc-entity-name* has an explicit interface, its characteristics shall be the same as *initial-proc-target*
 33 except that *initial-proc-target* may be pure even if *proc-entity-name* is not pure and *initial-proc-target*
 34 may be an elemental intrinsic procedure.

35 If the characteristics of *proc-entity-name* or *initial-proc-target* are such that an explicit interface is
 36 required, both *proc-entity-name* and *initial-proc-target* shall have an explicit interface.

37 If *proc-entity-name* has an implicit interface and is explicitly typed or referenced as a function, *initial-*
 38 *proc-target* shall be a function. If *proc-entity-name* has an implicit interface and is referenced as a
 39 subroutine, *initial-proc-target* shall be a subroutine.

40 If *initial-proc-target* and *proc-entity-name* are functions, they shall have the same type; corresponding
 41 type parameters shall either both be deferred or both have the same value.

42 [Editor: Add an item to the end of the list in **16.4.2.1.1 Events that cause pointers to become 414:18+
 43 associated:**]

- 44 (3) The pointer is an ultimate component of an object of a type for which default initialization
 45 is specified for the component, and the corresponding initializer is an initialization target,
 46 and

1 [Editor: Copy the three subsidiary items of item (4) in **16.4.2.1.2 Events that cause**
2 **pointers to become disassociated** at [414:26-30] — the first of which begins “a procedure
3 is invoked ...” — to here.]

4 [Editor: Within the fourth item in **16.4.2.1.2 Events that cause pointers to become disassociated** 414:25
5 — the one that begins “The pointer is an ultimate component...” — insert “, and the corresponding
6 initializer is a reference to the intrinsic function NULL,” before “is specified”.]