

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

1 Edits

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

6 [Editor: Replace the second right-hand side for *component-initialization* (R444):] 50:14
 7 **or** *data-pointer-init*

8 [Editor: Within the second paragraph of **4.5.3.4 Default initialization for components** “*null-init* 53:6
 9 appears for a” \Rightarrow “*data-pointer-init* consisting of *null-init* appears for a data pointer component or
 10 *proc-pointer-init* consisting of *null-init* appears in *proc-decl* for a procedure”.]

11 [Editor: After the second paragraph of **4.5.3.4 Default initialization for components** insert new 53:7+ New ¶’s
 12 paragraphs:]

13 If *data-pointer-init* consisting of *variable* appears for a data pointer component, that component in any
 14 object of the type is initially associated with *variable* or becomes associated with *variable* as specified
 15 in 16.4.2.1.1.

16 If *proc-pointer-init* consisting of *procedure-name* appears in *proc-decl* for a procedure pointer compo-
 17 nent, that component in any object of the type is initially associated with *procedure-name* or becomes
 18 associated with *procedure-name* as specified in 16.4.2.1.1. The component and *procedure-name* shall be
 19 related in the same way as required for *proc-pointer-object* and *proc-target* in 7.4.2.2.

20 [Editor: At the end of **4.5.3.4 Default initialization for components**, immediately before **4.5.3.5** 54:1-
 21 **Component order** insert the following note, which illustrates that we should not require the non-null
 22 initial target of a pointer component to be previously declared.]

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

A pointer component of a derived type may have an initially non-null target, so long as that target is accessible, has the SAVE attribute, does not have the POINTER or ALLOCATABLE attribute, has no expressions that are not initialization expressions (such as a variable subscript), and would be permitted as a target in a pointer assignment.

```

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

TYPE(NODE), SAVE, TARGET :: SENTINEL

```

23 [Editor: Replace the second right-hand side of *initialization* (R506) and insert a new syntax rule:] 72:16
 24 **or** *data-pointer-init*

25 R506 $\frac{1}{2}$ *data-pointer-init* **is** *null-init*
 26 **or** *variable*

27 [Editor: Within the first constraint after *null-init* (R507) “with no arguments” \Rightarrow “that does not have an 72:18-19
 28 argument with a type parameter that is assumed or is defined by an expression that is not an initialization
 29 expression (7.1.7)”. This tiny generalization makes the definition of “initialization expression” easier in
 30 the case of a structure constructor with a pointer component. Otherwise, C525 $\frac{1}{2}$ (immediately below)

- 1 would have to be repeated there.]
-
- 2 [Editor: After the zillionth constraint after *null-init* (R507) — the one that begins “If => appears ...” 73:16+
- 3 — insert a new constraint:]
- 4 C525 $\frac{1}{2}$ (R506 $\frac{1}{2}$) The *variable* shall be declared in the same scoping unit or be accessible therein by
- 5 use or host association, shall have the SAVE attribute or be declared in the main program,
- 6 shall not have the POINTER or ALLOCATABLE attribute, every expression within it shall be
- 7 an initialization expression, and it shall satisfy the requirements for a *data-target* in a pointer
- 8 assignment statement (7.4.2) in which the *data-pointer-object* is the corresponding data entity.
- 9 [This intentionally doesn’t say “previously declared”. That would prohibit initializing a “next” pointer
- 10 component to a “sentinel” object.]
-
- 11 [Editor: Within **5.1 Type declaration statements** replace the zillionth paragraph of ordinary normative 74:33-34
- 12 text — the one that begins “If *initialization* is => ...”]]
- 13 If *initialization* is => *null-init*, the initial association status of *object-name* is disassociated. If *initial-*
- 14 *ization* is => *variable*, *object-name* is initially associated with the variable.
- 15 [An *object-name* is already required to be a pointer by C525. The first sentence of this edit may be
- 16 useful even if this proposal does not proceed.]
-
- 17 [Editor: Replace the *null-init* right-hand side of *data-stmt-constant* (R532):] 88:26
- 18 **or** *data-pointer-init*
-
- 19 [Editor: Within the fifth paragraph of **5.2.5 DATA statement** — the one that begins “The expanded 89:12
- 20 sequence ...” — “*null-init*” \Rightarrow “initial association status”.]
-
- 21 [Editor: Within the sixth and seventh paragraphs of **5.2.5 DATA statement** — the ones that begin “A 89:14,16
- 22 *data-stmt-constant* ...” — “*null-init*” \Rightarrow “*data-pointer-init*” twice.]
-
- 23 [Editor: Within the sixth paragraph of **5.2.5 DATA statement** — the first one that begins “A *data-stmt-* 89:15
- 24 *constant* ...” — “The” \Rightarrow “If *data-pointer-init* is *null-init*, the”. “pointer *data-stmt-object*” \Rightarrow “data
- 25 statement object” because the data statement object is already required to be a pointer and syntax
- 26 terms don’t have association status. Insert a new sentence at the end of the paragraph:]
- 27 If *data-pointer-init* is *variable* the corresponding data statement object is initially associated with the
- 28 variable.
-
- 29 [Editor: Replace the third item in the numbered list in **7.1.7 Initialization expression**:] 126:27-29
- 30 (3) A structure constructor where each *component-spec* corresponding to
- 31 (a) An allocatable component is a *null-init*,
- 32 (b) A data pointer component is a *data-pointer-init*,
- 33 (c) A procedure pointer component is a *proc-pointer-init*,
- 34 (d) Any other component is an initialization expression,
-
- 35 [Editor: Replace the sixth item in the numbered list in **7.1.7 Initialization expression** — the one that 127:4-6
- 36 begins “A reference to the transformational function ...”:]
- 37 (6) A *null-init*,
-
- 38 [Editor: Replace *proc-decl* (R1214):] 264:19
- 39 R1214 *proc-decl* **is** *procedure-entity-name* [=> *proc-pointer-init*]
- 40 R1214 $\frac{1}{2}$ *proc-pointer-init* **is** *null-init*
- 41 **or** *procedure-name*
-
- 42 [Editor: After the fifth constraint after *interface-name* (R1215) — the one that begins “If => appears 264:30+
- 43 ...” — insert a new constraint:]
- 44 C1216 $\frac{1}{2}$ (R1214 $\frac{1}{2}$) A *procedure-name* shall be the name of an accessible external or module procedure,
- 45 or the name of a specific intrinsic function listed in 13.6 and not marked with a bullet (●).

-
- 1 [Editor: Replace the fifth paragraph of **12.3.2.3** — the one that begins “If => appears ...”:] 265:15-18
 2 If => appears in a *proc-decl* in a *procedure-declaration-stmt* it specifies that the initial association status
 3 of the corresponding procedure entity is defined, and implies the SAVE attribute. The SAVE attribute
 4 may be reaffirmed by explicit use of the SAVE attribute in the *procedure-declaration-stmt*, by inclusion
 5 of the procedure entity name in a SAVE statement (5.2.12), or by the appearance of a SAVE statement
 6 without a *saved-entity-list* in the same scoping unit. If => *null-decl* appears, the procedure entity is
 7 initially disassociated. If => *procedure-name* appears, the procedure entity is initially associated with
 8 the procedure specified by *procedure-name*. The characteristics of *procedure-entity-name* and *procedure-*
 9 *name* shall be related in the same way as required for *proc-pointer-object* and *proc-target* in 7.4.2.2.
-
- 10 [Editor: Add an item to the end of the list in **16.4.2.1.1 Events that cause pointers to become** 414:18+
 11 **associated**.]
- 12 (3) The pointer is an ultimate component of an object of a type for which default initialization
 13 consisting of *variable* or *procedure-name* is specified for the component and
 14 [Editor: Copy the three subsidiary items of item (4) in **16.4.2.1.2 Events that cause**
 15 **pointers to become disassociated** at [414:26-30] — the first of which begins “a procedure
 16 is invoked ...” — to here.]
-
- 17 [Editor: Within the fourth item in **16.4.2.1.2 Events that cause pointers to become disassociated** — 414:25
 18 the one that begins “The pointer is an ultimate component...” — insert “consisting of *null-init*” before
 19 “is specified”.]