

Subject: SAME\_TYPE\_AS is inconsistent with SELECT TYPE  
 From: Van Snyder

1 The disposition of edits in this paper is in 03-270.

## 2 1 The problems

3 A *type-spec* in a *type-guard-stmt* in a SELECT TYPE construct is allowed to be of an intrinsic type, but  
 4 neither argument of the SAME\_TYPE\_AS intrinsic procedure is allowed to be of intrinsic type. This is  
 5 inconsistent.

6 The description of the effect of the CLASS IS statement doesn't work if the *type-spec* specifies an intrinsic  
 7 type, because the term "is an extension type of" doesn't apply to intrinsic types. The MOLD argument  
 8 of the EXTENDS\_TYPE\_OF intrinsic procedure isn't allowed to be of intrinsic type, so this is OK.

9 An obscure point relevant to the SAME\_TYPE\_AS intrinsic procedure would benefit from clarification.

## 10 2 The solutions

11 Allow either argument of the SAME\_TYPE\_AS intrinsic procedure to be of intrinsic type.

12 Add a sentence to Note 13.17.

13 Prohibit the *type-spec* in a CLASS IS *type-guard-stmt* from specifying an intrinsic type.

## 14 3 Edits

15 Edits refer to 03-007r2. 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 C815 $\frac{1}{2}$  (R823) The *type-spec* in a CLASS IS *type-guard-stmt* shall not specify an intrinsic type. 162:20+

---

20 [Editor: Insert "intrinsic or" before "extensible" twice.] 347:30, 348:1

---

21 [Editor: Add a sentence in Note 13.17 "An unlimited polymorphic object has no declared type; therefore, 348:4+2  
 22 if either A or B is an unlimited polymorphic object that is a disassociated pointer or an unallocated  
 23 allocatable, the result is false."]

## 24 4 Malcolm has proposed...

25 Malcolm has proposed that neither the EXTENDS\_TYPE\_OF nor the SAME\_TYPE\_AS intrinsic pro-  
 26 cedure are useful because they provide the same functionality as the SELECT TYPE constuct; they  
 27 could therefore be deleted without loss. They don't provide quite the same functionality, however: The  
 28 intrinsics test the relation between objects, while the construct tests the relation between an object  
 29 and a *type-spec*. The functionality of the intrinsics is actually impossible to get by using the con-  
 30 struct. The functionality of EXTENDS\_TYPE\_OF is probably not very useful, but the functionality of  
 31 SAME\_TYPE\_AS probably is useful, for example to inquire whether two dummy arguments have the  
 32 same dynamic type — most likely to produce error messages. Therefore, it's probably OK to delete  
 33 EXTENDS\_TYPE\_OF but not SAME\_TYPE\_AS.

---

34 [Editor: Delete.] 296:15

---

35 [Editor: Delete (including Note 13.9).] 316:11-22+2

36 On the other hand, EXTENDS\_TYPE\_OF doesn't add much complexity to the standard.