Subject: SAME_TYPE_AS is inconsistent with SELECT TYPE
From: Van Snyder

1 The problems

A *type-spec* in a *type-guard-stmt* in a SELECT TYPE construct is allowed to be of an intrinsic type, but neither argument of the SAME_TYPE_AS intrinsic procedure is allowed to be of intrinsic type. This is inconsistent.

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

An obscure point relevant to the SAME_TYPE_AS intrinsic procedure would benefit from clarification.

2 The solutions

Allow either argument of the SAME_TYPE_AS intrinsic procedure to be of intrinsic type.

Add a sentence to Note 13.17.

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

3 Edits

Edits refer to 03-007r2. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

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

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

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

4 Malcolm has proposed...

Malcolm has proposed that neither the EXTENDS_TYPE_OF nor the SAME_TYPE_AS intrinsic procedure are useful because they provide the same functionality as the SELECT TYPE constuct; they could therefore be deleted without loss. They don't provide quite the same functionality, however: The intrinsics test the relation between objects, while the construct tests the relation between an object and a *type-spec*. The functionality of the intrinsics is actually impossible to get by using the construct. The functionality of EXTENDS_TYPE_OF is probably not very useful, but the functionality of SAME_TYPE_AS probably is useful, for example to inquire whether two dummy arguments have the same dynamic type — most likely to produce error messages. Therefore, it's probably OK to delete EXTENDS_TYPE_OF but not SAME_TYPE_AS.

[Editor: Delete.] 296:15

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

On the other hand, EXTENDS_TYPE_OF doesn't add much complexity to the standard.