Subject: Discussions of INTRINSIC and EXTERNAL attributes are scattered; discussion of INTRINSIC is contradictory and repetitive
From: Van Snyder

1 Introduction

We have tried to consolidate discussion of attributes into section 5.1.2. The discussion of the INTRINSIC attribute, however, is scattered between 5.1.2.11 and 12.3.2.4 (The INTRINSIC statement), and the discussion of the EXTERNAL attribute is scattered between 5.1.2.10 and 12.3.2.2. Some of the discussion is contradictory – section 12.3.2.4 allows any intrinsic procedure named in an INTRINSIC statement to be used as an actual argument, while section 5.1.2.11 limits this set to those listed in section 13.15 and not marked with a bullet (•). Finally, some of the discussion is repeated.

This paper

(1) Consolidates discussion of the INTRINSIC attribute in section 5.1.2.11, and reduces section 12.3.2.4 to stating that the INTRINSIC statement confers the INTRINSIC attribute, and giving its syntax, and

(2) Consolidates discussion of the EXTERNAL attribute in section 5.1.2.10, and reduces section 12.3.2.2 by deleting material that is redundant and misleading.

2 Edits

Edits refer to 00-007r1. 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 immediately following text, while a page and line number followed by + indicates that immediately following text is to be inserted after the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

A dummy argument that has the EXTERNAL attribute is a dummy procedure or a dummy procedure pointer. A name that has the EXTERNAL attribute and that is not a dummy argument is the name of an external procedure, a procedure pointer, or a block data program unit. 76:39+

Note 5.18$\frac{1}{2}$

The EXTERNAL attribute for a block data program unit can only be specified by an EXTERNAL statement (12.3.2.2); it is not possible to do so in a type declaration statement.

The **INTRINSIC attribute** confirms that a name is the specific name (13.15) or generic name (13.13, 13.14) of an intrinsic procedure. The INTRINSIC attribute allows the specific name of an intrinsic procedure that is listed in section 13.15 and not marked with a bullet (•) to be used as an actual argument (12.4). 76:41-77:2

Declaring explicitly that a generic intrinsic procedure name has the INTRINSIC attribute does not cause that name to lose its generic property.

The following constraint applies to syntax rules R504 and R1214:

Constraint: If the name of a generic intrinsic procedure is explicitly declared to have the INTRINSIC attribute, and it is also the generic name in one or more generic interfaces (12.3.2.1) accessible in the same scoping unit, the procedures in the interfaces and the specific intrinsic procedures shall all be functions or all be subroutines, and the characteristics of the specific intrinsic procedures and the procedures in the interfaces shall differ as specified in section 14.1.2.3.

The INTRINSIC attribute may also be declared by the INTRINSIC statement (12.3.2.4).

[Editor: Delete – moved to 5.1.2.10 and stated in terms of the attribute, not the statement. The present wording implies that the only way to declare a dummy procedure or external procedure is to put its name in an EXTERNAL statement. This isn't necessarily true: The EXTERNAL attribute can be specified by an interface body, or specified for a function in a type declaration statement.] 251:43-252:2

[Editor: Delete misleading note. The correct story is in 5.1.2.10, which is referenced at [251:39].] 252:11-13

[Editor: Insert "(5.1.2.11)" after "attribute" and delete "A name ..." to the end of the paragraph.] 253:44-47

[Editor: Delete.] 254:3-11

Constraint: If an actual argument is a name that is explicitly declared to have the INTRINSIC attribute, it shall not be the specific name of an intrinsic procedure that is listed in section 13.15 and marked with a bullet (•). 255:22+