Subject: Interface bodies ought to be local entities
From: Van Snyder
References: 02-107r1

1 Introduction

At meeting 160, Subgroup C agreed in principle with the proposition set forth in 02-107r1, that interface bodies ought to be local entities. This is in keeping with the intent explicitly stated in C.9.3 that a name declared by an interface body need not imply that an external procedure of that name is part of the program. Many processors behave in the way therein described.

Unfortunately, 02-107r1 was incomplete in not discussing an interface body for a specific procedure in a generic interface block, in the case that that specific interface is never referenced. It was therefore withdrawn.

While studying the revision, I noticed that the assertion at [80:23-24], *viz.* "This attribute may be specified for any of these entities by an EXTERNAL statement (12.3.2.2)" is contradicted by 12.3.2.2 [262:4-5]. Further, the next sentence at [80:24-26], *viz.* "It may be specified for any of these entities other than a block data subprogram by a type declaration statement, a procedure declaration statement (12.3.2.3), or by the appearance of the name as a specific procedure name in an interface body (12.3.2.1)" is contradicted by C512 [70:36-37].

2 Edits

Edits refer to 02-007r1. 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 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.

[Editor: Replace "This ... (12.3.2.1)" by the following:] 80:23-26

This attribute may be explicitly specified by the appearance of the name as

(1) The *external-name* in an *external-stmt* (12.3.2.2),

(2) The *object-name* in a *type-declaration-stmt* in which the EXTERNAL *attr-spec* (5.1) appears,

(3) The *proc-entity-name* in a *proc-declaration-stmt* (12.3.2.3),

(4) The *function-name* in the *function-stmt* or the *subroutine-name* in the *subroutine-stmt* that introduces an interface body (12.3.2.1) provided the interface body is in a generic interface block, there is a reference to the generic interface, and the reference is resolved to the name of the interface body (9.5.3.7.3, 12.3.2.1.1, 12.3.2.1.2, 12.4.4.1), or

(5) The *function-name* in the *function-stmt* or the *subroutine-name* in the *subroutine-stmt* that introduces an interface body (12.3.2.1) provided the name is referenced in a scoping unit in which the interface is accessible, and the reference is either

  (a) An actual or dummy argument,

  (b) A *procedure-name* in a *function-reference* or *call-stmt* in a scoping unit in which the reference is established to be specific (12.4.4.2), or

  (c) A *procedure-name* in a *pointer-assignment-stmt* (7.5.2).

[Editor: Delete “for an existing external procedure or a dummy procedure” – the “existing external procedure” part appears to contradict the second paragraph of C.9.3.] 257:19-20

[Editor: “If … interface” ⇒ “The name of an interface body`\index{name!interface body}` is the *function-name* in the *function-stmt* or the *subroutine-name* in the *subroutine-stmt* that introduces the interface body. If an interface body is accessible in a scoping unit and its name has the EXTERNAL attribute, its name has the explicit interface given by the interface body.”] 257:20-24

[Editor: “abstract interfaces” ⇒ “interface bodies”.] 396:6

[Editor: Insert “or interface body name” before “(12.3.2.1)”.] 396:12

(12) The name of an interface body; 401:22