Subject: The PROCEDURE() syntax of interface blocks is unnecessary
From: Van Snyder
References: 02-165, 02-106r1, 02-107r1, 02-177

1 Introduction

The PROCEDURE() syntax to create abstract interfaces are unnecessary. You may recall that abstract interfaces were introduced to give explicit interfaces to procedure pointers, and later used to give explicit interfaces to deferred type-bound procedure bindings.

Presumably, the reason to invent abstract interfaces was so as not to require a procedure of the same name as the interface to be a part of the program. It is, however, explicitly contemplated in the second paragraph of C.9.3 that one should be allowed to have interfaces for procedures that are not part of the program.

Therefore, it would be possible simply to use any entity with an explicit interface, including ordinary interface bodies in ordinary interface blocks, to provide the explicit interfaces needed for procedure pointers and deferred type-bound procedure bindings.

This proposal changes the definition of "abstract interface" and deletes the syntactic entity of that name; it incorporates 02-165 to make interface bodies local entities, and 02-177 to correct several defects in the discussion of the EXTERNAL attribute.

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.

---

(6) Procedure pointers, xiii, item(6)

[Editor: Insert ", abstract interface" after "pointer".] 80:23

[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 the *external-name* in an *external-stmt* (12.3.2.2), the *object-name* in a *type-declaration-stmt* in which the EXTERNAL *attr-spec* (5.1) appears, the *proc-entity-name* in a *proc-declaration-stmt* (12.3.2.3), or it is the name of an interface body.

If a name has the EXTERNAL attribute and the POINTER attribute, then it is the name of a procedure pointer. If it is a dummy argument it is a dummy procedure. If the name does not have the POINTER attribute, is not a dummy procedure, and it is 80:36-38

(1) An *object-name* in a *type-declaration-stmt*,
(2) The *proc-entity-name* in a *proc-declaration-stmt*,
(3) The *procedure-name* in a *function-reference*, *call-stmt*, or *pointer-assignment-stmt*,
(4) An actual argument, or
(5) The name of an interface body and there is a reference to a generic interface that resolves to that interface body (9.5.3.7.3, 12.3.2.1.1, 12.3.2.1.2, 12.4.4.1),

then it is the name of an external procedure.

If the name appears only as the *external-name* in an *external-stmt* it is the name of an external procedure or a block data program unit.

If none of the above apply it is the name of a local entity.

[Editor: Delete ", the value of its binding label" (because this shouldn't be a characteristic).] 254:13-14

An **abstract interface** consists of procedure characteristics and the names of dummy arguments. 255:5+ New ¶

[Editor: "interface … argument" ⇒ procedure's interface consists of its abstract interface, its name, its binding label if any" (this is where the binding label belongs).] 255:7-8

[Editor: Delete.] 256:13

[Editor: Delete "An interface ... **interface**.".] 257:15-16

[Editor: Delete "INTRODUCED … specification)" ⇒ "without a generic specification".] 257:17-18

[Editor: Delete "in a generic … block".] 257:18-19

[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 its name has the explicit interface given by the interface body."] 257:20-24

[Editor: Delete "or abstract interface".] 257:25

[Editor: Delete 12.3.2.1.4.] 261:4-6+

C1212 (R1215) The *name* shall be the name of a dummy procedure, external procedure, interface body, internal procedure, intrinsic procedure, module procedure, or procedure pointer. The interface of the entity specified by *name* shall be explicit. If it is declared by a *procedure-declaration-stmt* it shall be previously declared. If it is an intrinsic procedure the *name* shall be the specific name of an intrinsic procedure that is listed in 13.6 and not marked with a bullet (•). 262:23

C1212$\frac{1}{2}$(R1215) The *name* shall not be the same as a keyword that specifies an intrinsic type.

[Editor: "All … *proc-interface*" ⇒ "The abstract interface (12.3) is that specified by the interface named by *abstract-interface-name*".] 263:2-3

[Editor: Replace line 1 of Note 12.15 by "`INTERFACE`", followed by the interface bodies, associated comments, and `END INTERFACE` statement from lines 3-12 of Note 12.11 to here, with "abstract" deleted twice (at third and eighth lines of Note 12.11).] 263:11+5

[Editor: Delete "an abstract … or".] 277:12

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

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

[Editor: "blocks" ⇒ "bodies"; delete ", abstract interfaces" (because "blocks" is just plain wrong).] 401:2

---

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

[Editor: Delete ", abstract interface".] 419:6

3 If we don't do this...

[Editor: Delete ", the value of its binding label" (because this shouldn't be a characteristic).] 254:13-14

[Editor: Insert ", the value of its binding label, if any" after the second "procedure" (because this is where it belongs).] 255:7

[Editor: "blocks" ⇒ "bodies" (because "blocks" is just plain wrong).] 401:2

abstract interface (12.3.2.1) : An interface declared in an interface block introduced by INTERFACE PROCEDURE(). It may be used to specify the interface of an external or dummy procedure, a procedure pointer, or a deferred type-bound procedure binding. 415:6+