Subject: Edits to allow internal subprograms as actual arguments and procedure pointer targets
From: Van Snyder
Reference: 03-258r1, section 1.7, 04-148, 04-382r1, WG5/N1626-J3-013

1 Introduction

Internal procedures as actual arguments and pointer targets were put onto the J3 work plan at the Delft WG5 meeting. It was decided to allow internal procedures of recursive hosts, after Malcolm insisted there are ways to do so that have no additional performance penalty attributable to the recursive nature of the host.

2 Edits

Edits refer to 04-007. 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. Two numbers in the margin separated by a point are the chapter and line number in the LaTeX source for 04-007.

[Editor: Within the penultimate constraint (C727) of **7.4.2 Pointer assignment**, after "external" insert ", internal".] 144:5 7.2150

[Editor: Within the fifth constraint (C1229) after *alt-return-spec* (R1222) — the one that begins "A *procedure-name* shall . . . " — after "external" insert ", internal". Better yet, replace C1229 with exactly the same text as C727, as modified by the edit for 144:5 above — but applying to R1221 of course.] 267:15 12.896

[Editor: Delete Note 12.16 — the one that begins "This standard does not allow...."] 267:17+1-7 12.908

[Editor: Within the second paragraph of **12.4.1.3 Actual arguments associated with dummy data objects**, after "external" insert ", internal".] 271:16 12.1220

[Editor: Before item (6) in the numbered list in **16.4.2.1.3 Events that cause the association status of pointers to become undefined** — the item that begins "The pointer is an ultimate . . . " — insert a new item:] 415:17+ 16.804+

(5½) Execution of a procedure is completed and the subprogram that defines that procedure is the host of an internal procedure that is the target of the pointer, even if the pointer has the SAVE attribute.

[Editor: Replace the second paragraph of 16.4.5. The first three sentences of the replacement paragraph are the first, last and second sentences of the existing paragraph — except that the second sentence appeared to be backward, so it's reversed here. The rest are new.] 418:16-419:3 16.1066

For argument association, the associating entity is the dummy argument and the pre-existing entity is the actual argument. For construct association, the associating entity is identified by the associate name and the pre-existing entity is the selector. For host association, the pre-existing entity is the entity in the host scoping unit and the associating entity is the entity in the contained scoping unit. If the host scoping unit of an internal subprogram is a recursive procedure, the pre-existing entity that participates in the association is the one from the innermost procedure instance at the instant

(1) the internal procedure is invoked if it is not invoked by way of a procedure pointer or dummy procedure,
(2) the internal procedure was an actual argument in a procedure reference that ultimately resulted in the internal procedure being associated with a dummy procedure if it is invoked by way of that dummy procedure, or
(3) the internal procedure's name was a *proc-target* in a pointer assignment statement that ultimately resulted in the internal procedure being associated with a procedure pointer if it

1 is invoked by way of that procedure pointer.

NOTE 16.16$\frac{1}{2}$

If an internal procedure is used as an actual argument, and the associated dummy argument is in turn used as an actual argument, and the internal procedure is invoked by way of the second dummy argument, the instance of the host is the one executing at the instant the internal procedure was first used as an actual argument. If an internal procedure's name is a *proc-target* in a pointer assignment, and the pointer in that pointer assignment is in turn used to provide the target for a second pointer, and the internal procedure is invoked by way of the second pointer, the instance of the host is the one executing at the instant the internal procedure was associated with the first pointer.