JOR Response to 01-360r1 - Comments on Section 12

To: J3
From: Craig Dedo
Date: December 4, 2001
Subject: JOR Response to 01-360r1 - Comments on Section 12

JOR reviewed the recommendations in 01-360r1 and recommends as follows. References are to 01-007r4.

1 Edits

[241:4-6] Delete the last sentence in the paragraph. Add the following three paragraphs.
The sequence of computations encapsulated by a procedure has access to entities in the invoking scoping unit by way of argument association (12.4.1). A dummy argument is a name that appears in the SUBROUTINE, FUNCTION, or ENTRY statement in the declaration of a procedure (R1226). Dummy arguments are also specified for intrinsic procedures and procedures in intrinsic modules in Sections 13, 14, and 15.

The entities in the invoking scoping unit are specified by actual arguments. An actual argument is an entity that appears in a procedure reference (R1221).

A procedure may also have access to entities in other scoping units, no necessarily the invoking scoping unit, by use association (16.7.1.2), host association (16.7.1.3), linkage association (16.7.1.4), storage association (16.7.3), or by reference to external procedures (5.1.2.6).

[244:1-2] Yes. Insert “an” before “interface”.

[258:8] Yes. Change “*target*” to “*data-target* or *proc-target*”.

[263:12] Yes. Change “been declared a pointer” to “the pointer attribute”.

[263-264:Note 12.38]
No change is necessary. We should keep the example simple.

[265:18-19] We do not accept the proposed change, but there is a technical defect that needs to be corrected. Remove [] from around *proc-language-binding-spec* in the second specification for the ENTRY statement.

[270:7] Yes. Change “target of” to “*data-target* or *proc-target* in”.

[271:Note 12.46] Yes. This is already corrected in 01-007r4.

[397:8] Yes. Insert “12,” before “12.4.1” in the definition for “actual argument”.

[400:33] Yes. Insert “12,” before “12.5.2.1” in the definition for “dummy argument”.

2 Miscellaneous

[247:10], [248:11] No change is necessary. INTENT(IN) prevents a program from modifying ALLOCATABLE and POINTER arguments but allows modification of POINTER targets. The inclusion of POINTER operands to defined operators is an intentional feature of Fortran 2000.

References

01-007r4, Fortran 2000 Draft

[End of J3 / 01-390]