Subject: First draft of edits for optional dummy default values
From: Van Snyder
Reference: 04-386r2

1 Introduction

Assuming default initial values for optional dummy arguments get onto the J3 work plan, the reason for this paper is to get a running start on the edits.

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.

These are three versions here. The first two assume simpler specs than in 04-386r2: the initializer behaves as an actual argument. They are simpler by imposing fewer constraints, thereby allowing a larger set of valid programs. The specs in 04-386r2 cause optional dummy arguments that have initializers to behave as though they have the VALUE attribute, with most of the C527 restrictions removed, and the initializers are assigned to them.

[Editor: "*initialization-expr*" ⇒ "*expr*".] 72:15

[Editor: Delete "a dummy argument,".] 73:11

C524$\frac{1}{3}$ (R506) If *object-name* is not a dummy argument, *expr* shall be an initialization expression. If *object-name* is a dummy argument, *expr* shall be a restricted expression. 73:14+

[Editor: "does not have" ⇒ "is neither a dummy argument nor has".] 74:24

[Editor: Insert ", an *object-name* that is a dummy argument," after "block".] 74:36

---

(2) An object designator with a base object that is a dummy argument that has initialization, or that has neither the OPTIONAL nor INTENT(OUT) attribute, 125:14-15

[Editor: "specification" ⇒ "restricted" thrice.] 126:7,9,14

If a restricted expression is the initialization for a dummy argument, and it depends upon the value of a dummy argument that is specified in the same *specification-part* and has initialization, the initialization for the dummy argument upon which the restricted expression depends shall be specified in a prior specification of the *specification-part*. The prior specification may be to the left of the restricted expression in the same statement. 126:19+

2.1 Simpler specs

C524$\frac{2}{3}$ (R504) If *initialization* appears and *object-name* is a dummy argument, OPTIONAL shall be specified and *initialization* shall meet the requirements for an actual argument associated with *object-name* (12.4.1). 73:14++

2.1.1 Initialization is an actual argument

If *initialization* is specified for an optional dummy argument that is not present (12.4.1.6), upon entry to the procedure *initialization* becomes associated with the dummy argument as an actual argument and the dummy argument is considered to be present. 83:11+

2.1.2 Initialization is as if an actual argument

If *initialization* is specified for an optional dummy argument that is not present (12.4.1.6), upon entry to the procedure the effect is as if *initialization* were associated with the dummy argument as an actual 83:11+

argument, but *initialization* is not an actual argument and the dummy argument does not become present.

[Editor: Insert "and does not have initialization" after "present".] 272:31

2.2 Specs as in 04-396r2

C524$\frac{2}{3}$ (R504) If *initialization* appears and *object-name* is a dummy argument, OPTIONAL shall be specified, neither INTENT(INOUT) nor INTENT(OUT) shall be specified, and *initialization* shall meet the requirements for an actual argument associated with *object-name* (12.4.1). 73:14++

If *initialization* is specified for an optional dummy argument that is not present (12.4.1.6), *initialization* is evaluated upon entry to the procedure. If the dummy argument has assumed shape or type parameters, they are assumed from *initialization*. If the dummy argument is not a pointer, the value of *initialization* is assigned to the dummy argument as if by intrinsic assignment. If the dummy argument is allocatable, it is assumed not to be allocated before the assignment occurs. If the dummy argument is a pointer, *initialization* is assigned to the dummy argument as if by pointer assignment. The dummy argument is not considered to be present. 83:11+

[Editor: Insert "and does not have initialization" after "present".] 272:31
