

Subject: Miscellaneous editorial(?) comments and questions  
 From: Van Snyder

1 These are in addition to those Malcolm already has in 05-008.

## 2 1 Comments without edits

3 [In **5.1.2.11 Pointer attribute**, The term **pointer association** needs definition.] 83:24+

## 4 2 Edits

5 Edits refer to 04-007. Page and line numbers are displayed in the margin. Absent other instructions, a  
 6 page and line number or line number range implies all of the indicated text is to be replaced by associated  
 7 text, while a page and line number followed by + (-) indicates that associated text is to be inserted after  
 8 (before) the indicated line. Remarks are noted in the margin, or appear between [ and ] in the text.

9 [Editor: (Cannonball polishing) In **4.5.3 Components**, in C441 (the one that begins “If neither the 50:26  
 10 POINTER ...”), delete “attribute” after “POINTER” and “the” before “ALLOCATABLE” (cf. C440).]

11 [Editor: In the last normative paragraph of **4.5.9 Construction of derived-type values**, at the end of 65:13  
 12 the last sentence, after “bounds” insert “, deferred length type parameter values,”]

13 [Editor: In the antepenultimate normative paragraph of **5.1 Type declaration statements** (the one 74:27  
 14 beginning “If an *entity-decl* ...”), insert a space in “=*initialization-expr*.”]

15 [In the penultimate normative paragraph of **5.1 Type declaration statements** (the one beginning “If 74:33  
 16 *initialization* is ...”) the “shall be a pointer” part is covered by C525. Editor: Replace the paragraph  
 17 by “If *initialization* is => *null-init* the initial association status of *object-name* is disassociated.”]

18 [In **5.2.5 DATA statement**, in C563 (the one that begins “If a DATA ...”), the cited rule is *data-* 88:28-29  
 19 *stmt-constant*, so there’s no need to say “If a DATA statement constant...” Editor: replace it by the  
 20 following:]

21 C563 (R532) A named constant or the type of a structure constructor shall have been declared previ-  
 22 ously in the same scoping unit or made accessible by use or host association.

23 [Constraint C715 is on *assignment-stmt* (R734) so there’s no need to say so twice. Editor: Delete “in 138:13  
 24 an *assignment-stmt*”.]

25 [In **7.4.1.2 Intrinsic assignment statement**, add some white space at the bottom of Table 7.8.] 139:2+12

26 [Editor: In **7.4.2 Pointer assignment**, in C727 (the one that begins “A *procedure-name* shall be ...”), 144:5  
 27 insert “specific” before “name”. cf. [271:16], the second paragraph of **12.4.1.3 Actual arguments**  
 28 **associated with dummy procedure entities**.]

29 [Editor: In the numbered list in **8.1.6.4.4 Loop termination**, add a comma at the end of each of items 168:13-19  
 30 (1)-(4) and “, or” at the end of item (5).]

31 [Editor: In **11.2 Main program**, in C1101 (the one that begins “In a *main-program* ...”) “In a *main-* 249:16  
 32 *program*, the *execution-part*” ⇒ “The *execution-part* of a *main-program*”.]

33 [Editor: In the last normative paragraph of **12.3.2.3 Procedure declaration statement** (the one that 265:17-18  
 34 begins “If => *null-init* ...”) “or by ... statement” ⇒ “, by the appearance of *procedure-entity-name* in  
 35 a SAVE statement (5.2.12), or by the appearance of a SAVE statement without a *saved-object-list* in the  
 36 same scoping unit” (cf. [74:35-39], the last normative paragraph of **5.1 Type declaration statements**).  
 37 Also remove the extraneous quotation mark at the end of the sentence.]

38 [Editor: In **12.4 Procedure reference**, in C1229 (the one that begins “A *procedure-name* shall ...”) 267:15  
 39 insert “specific” before “name”. Better yet, replace the text by the text of C727 as revised above (but  
 40 continue to apply to R1221).]

- 1 [Editor: In the first normative paragraph of **12.4.1.6 Restrictions on dummy arguments not present**, 272:26  
2 “A dummy” ⇒ “An optional dummy”; “a dummy” ⇒ “an optional dummy”.]
- 3 [A procedure pointer isn’t an object. Editor: In Table 13.1 in **13.7.88 NULL([MOLD])**, insert “or 341:13+4  
4 procedure pointer” after “object” in both columns of the third row (counting the heading as row 1).]
- 5 [Editor: Move **16.2.3 Restrictions on generic declarations** to be in **12.3.2 Specification of the proce-** 407:24ff  
6 **cedure interface**, say immediately before **12.3.2.2 EXTERNAL statement** at [263:17]. This also entails  
7 either inserting an explicit label `D16:Restrictions on generic declarations`, or finding and revising  
8 references to that label.]
- 9 [Editor: In **16.4.5 Establishing associations**, in the sentence that begins “For host association ...” in 418:17-18  
10 the second paragraph, exchange “associating” and “pre-existing”. Or does this need an interp?]

### 11 3 Questions

- 12 Should we revise the section macros and page footer macro to put the current subclause number — the  
13 last one that begins on the page if any begin on the page — into page footers? If we do, I prefer that  
14 the page number remain on the outer edge, with the subclause number on the inner edge.
- 15 Do we need both the term “conformable” and “conform”? Both are used in many places.
- 16 Should “common association” be mentioned in Section 16? 99:29
- 17 What if *expr* is scalar? 148:42
- 18 Should kind type parameters be involved in the definition of the result value of `EXTENDS_TYPE_OF`? 316:19-22
- 19 What are identifiers of statement labels? 406:7

### 20 4 The index

- 21 Why are some index items bold and some not? There is no consistent pattern.
- 22 Should we search for index terms, and add occurrences of them that are not indexed to the index?
- 23 Index 12.0 [255:6,10] and 12.4.1 [268:1] for “argument.” Make “argument association” subsidiary to  
24 “argument”, i.e., `\mindex{argument!association}`. Combine “argument keywords” into “argument  
25 keyword”, and make it subsidiary to “argument”.
- 26 Make “array constructor”, “array intrinsic assignment statement”, “array pointer” and “array section”  
27 subsidiary to “array”. Combine “array elements” into “array element” and make it subsidiary to “array”.  
28 ‘Make “array element order” subsidiary to “array!element”.
- 29 Make “ASCII character set”, “ASCII character type” and “ASCII collaring sequence” subsidiary to  
30 “ASCII”.
- 31 Make “assignment statement” subsidiary to “assignment”.
- 32 Combine “associate names” into “associate name”.
- 33 Make “pointer association status” subsidiary to “association”.
- 34 Do we need “assumed-shape array”, “assumed-size array” and “automatic array” in addition to “array!as-  
35 sumed-shape”, “array!assumed-size” and “array!automatic”?
- 36 Combine “attributes” into “attribute”. Make “attribute specification statements” subsidiary to “at-  
37 tribute”. Index the `KIND` attribute.
- 38 If we do need both “array!automatic” and “automatic array”, make “automatic array” and “automatic  
39 data object” both subsidiary to “automatic”.
- 40 Combine “belongs” into “belong”.
- 41 Do we need both “branch target statement” and “Branching”, since they’re both on the same page?  
42 Replace both by “branching”.
- 43 Make “character context”, “character intrinsic assignment statement”, character intrinsic operation”,

- 1 “character intrinsic operator”, character length parameter”, “character literal constant”, “character rela-
- 2 tional intrinsic operation”, “character sequence type”, character set“, “character storage unit”, character
- 3 string”, “character string edit descriptor” and “character type” subsidiary to “character”.
- 4 Make “common association” (if it continues to exist as a term), “common block” and “common block
- 5 storage sequence” subsidiary to “common”.
- 6 Combine “components” into “component”. Make “component” and “component keyword” subsidiary
- 7 to “component”. Combine “Component order” into “component order” and it them subsidiary to
- 8 “component”.
- 9 Delete “connected files” (since “connected” is indexed on the same page).
- 10 Make “constant subobject” subsidiary to “constant”.
- 11 Combine “Construct association” into “construct association”. Make it and “construct entity” subsidiary
- 12 to “construct”.
- 13 Combine “control edit descriptors” into “control edit descriptor”.
- 14 Remove “conversion!numeric“ since “numeric!conversion” is indexed. “numeric!editing” is indexed but
- 15 “editing!numeric” is not, etc.
- 16 Combine “data edit descriptors” into “data edit descriptor”. Make it and “data entity”, “data object”,
- 17 “data object reference”, “data pointer”, “data transfer” and “data type” subsidiary to “data”. Combine
- 18 “data transfer input statements”, “data transfer output statements” and “data transfer statements” into
- 19 “data transfer input/output statement” and make it subsidiary to “data!transfer”.
- 20 Combine “declarations” into “declaration”.
- 21 Combine “default-initialized” into “default initialization”. Make “default character”, “default initializa-
- 22 tion”, “default integer”, “default logical” and “default real” subsidiary to “default”.
- 23 Make “defined assignment”, “defined binary operation”, defined elemental assignment statement”, “de-
- 24 fined elemental operation” and “defined operation” subsidiary to “defined”. Make “defined assigned
- 25 statement” subsidiary to “defined!assignment”. Delete “defined unary operation”.
- 26 Decapitalize “Delimiters”.
- 27 Combine “derived types” into “derived type”. Make “derived type determination” subsidiary to “derived
- 28 type”.
- 29 Make “direct access input/output statement” subsidiary to “direct access”.
- 30 Combine “dummy arguments” into “dummy argument”. Make it and “dummy array”, “dummy data
- 31 object” and “dummy procedure” subsidiary to “dummy”.
- 32 Combine “edit descriptors” into “edit descriptor”. Refer “edit descriptor” to “format descriptor” (which
- 33 is made singular below).
- 34 Change “element array assignment (FORALL)” to “elemental array assignment (FORALL)”. Make
- 35 it and “elemental intrinsic function”, “elemental operation” and “elemental procedure” subsidiary to
- 36 “elemental”.
- 37 Combine “executable constructs” into “executable construct”. Make it and “executable statement”
- 38 subsidiary to “executable”.
- 39 Make “explicit formatting”, “explicit initialization” and “explicit interface” subsidiary to “explicit”.
- 40 Combine “expressions” into “expression”.
- 41 Make “extension operations”, “extension operator” and “extension type” subsidiary to “extension”.
- 42 Make “external file”, “external linkage”, “external procedure”, “external procedure”, “external subpro-
- 43 gram” and “external unit” subsidiary to “external”.
- 44 Make “field width” subsidiary to “field”.
- 45 Combine “files” into “file”. Make “file access”, “file connection”, “file inquiry”, “file position” and “file
- 46 storage unit” subsidiary to “file”. Make “file connection statements” subsidiary to “file connection”.
- 47 Make “file inquiry statement” subsidiary to “file inquiry”. Make “file positioning statements” subsidiary
- 48 to “file position”.

- 1 Combine “final subroutines” into “final subroutine”.
- 2 Combine “finalizable”, “finalization” and “finalized”?
- 3 Replace “format descriptors” by “format descriptor”.
- 4 Make “formatted data transfer”, “formatted input/output statement” and “formatted record” subsidiary
- 5 to “formatted”.
- 6 Make “function reference”, “function result” and “function statement” subsidiary to “function”.
- 7 Combine “Generic names” into “generic name”. Replace “generic procedure references” by “generic
- 8 procedure reference”. Make them and “generic identifier”, “generic interface”, “generic interface block”
- 9 and “generic procedure references” subsidiary to “generic”.
- 10 Combine “global entities” into “global entity”. Make it and “global identifier” subsidiary to “global”.
- 11 Replace “Graphic characters” by “graphic character”.
- 12 Make “host association” and “host scoping unit” subsidiary to “host”.
- 13 Combine “inheritance associated” into “inheritance association”.
- 14 Make “initialization expression” subsidiary to “initialization”.
- 15 Replace “Input statements” by “input statement”.
- 16 Make “input/output list” and “input/output statement” subsidiary to “input/output”.
- 17 Make “inquiry function” and “inquiry, type parameter” subsidiary to “inquiry”.
- 18 Make “instance of a subprogram” subsidiary to “instance”.
- 19 Make “integer constant”, “integer editing”, “integer model” and “integer type” subsidiary to “integer”.
- 20 Make “interface body”, “interface block” and “interface of a procedure” subsidiary to “interface”.
- 21 Combine “internal files” into “internal file”. Make it and “internal procedure”, “internal subprogram”
- 22 and “internal unit” subsidiary to “internal”.
- 23 Combine “intrinsic operations” into “intrinsic operation”. Combine “intrinsic procedures” into “intrinsic
- 24 procedure”. Combine “intrinsic types” into “intrinsic type”. Make them and “intrinsic assignment
- 25 statement”, “intrinsic binary operation”, “intrinsic module” and “intrinsic unary operation” subsidiary
- 26 to “intrinsic”. Index “intrinsic function” and “intrinsic subroutine”?
- 27 Make “ISO 10646 character set” and “ISO 10646 character type” subsidiary to “ISO 10646”.
- 28 Make “length of a character string” and “length type parameter” subsidiary to “length”.
- 29 Replace “letters” by “letter”.
- 30 Combine “Lexical tokens” into “lexical token”.
- 31 Combine “local identifiers” into “local identifier”. Make it, “local entity” and “local variable” subsidiary
- 32 to “local”.
- 33 Combine “logical intrinsic operations” into “logical intrinsic operation”. Make it and “logical intrinsic
- 34 assignment statement”, “logical intrinsic operator” and “logical type” subsidiary to “logical”.
- 35 Combine “masked array assignment” and “masked array assignment (WHERE)”.
- 36 Make “module procedure”, “module reference” and “module subprogram” subsidiary to “module”.
- 37 Make “named common block”, “named constant” and “named file” subsidiary to “named”.
- 38 Combine “Nonexecutable statements” into “nonexecutable statement”.
- 39 Combine “numeric intrinsic operations” into “numeric intrinsic operation”. Combine “numeric types”
- 40 into “numeric type”. Make them and “numeric conversion”, “numeric intrinsic assignment statement”,
- 41 “numeric intrinsic operator”, “numeric relational intrinsic operation” and “numeric sequence type” sub-
- 42 subsidiary to “numeric”.
- 43 Combine “obsolescent features” into “obsolescent feature”.
- 44 Combine “operands” into “operand”.
- 45 Combine “operations” into “operation”.
- 46 Combine “operators” into “operator”. Make “operator precedence” subsidiary to “operator”.

- 1 Replace “Output statements” by “output statement”.
- 2 Combine “PARAMETER” into “PARAMETER attribute”.
- 3 Make “parent component”, “parent data transfer statement” and “parent type” subsidiary to “parent”.
- 4 Combine “pointer associated” into “pointer association”. Make it and “pointer assignment” subsidiary
- 5 to “pointer”. Make “pointer assignment statement” subsidiary to “pointer assignment”. Make “pointer
- 6 association status” subsidiary to “pointer association”.
- 7 Combine “Preconnection” into “preconnected”. Make “preconnected files” subsidiary to “preconnected”.
- 8 Combine “procedure references” into “procedure reference”. Make it and “procedure designator”, “pro-
- 9 cedure interface” and “procedure pointer” subsidiary to “procedure”.
- 10 Make “real model”, “real part” and “real type” subsidiary to “real”.
- 11 Replace “record lengths” by “record length”. Make it, “record file” and “record number” subsidiary to
- 12 “record”.
- 13 Combine “relational intrinsic operations” into “relational intrinsic operation”. Make it and “relational
- 14 intrinsic operator” subsidiary to “relational”.
- 15 Combine “representation methods” into “representation method”.
- 16 Make “sequence association”, “sequence structure” and “sequence type” subsidiary to “sequence”.
- 17 Make “sequential access input/output statement” subsidiary to “sequential access”.
- 18 Make “shape conformance” subsidiary to “shape”.
- 19 Make “size of a common block” and “size of a storage sequence” subsidiary to “size”.
- 20 Remove “specific interface block”.
- 21 Replace “Specific names” by “specific name”.
- 22 Combine “specifications” into “specification”. Make “specification expression”, “specification function”
- 23 and “specification inquiry” subsidiary to “specification”.
- 24 Combine “Statement functions” into “statement function”. Replace “statements” by “statement”. Make
- 25 “statement entity”, “statement function”, “statement label” and “statement order” subsidiary to “state-
- 26 ment”.
- 27 Index “statement!ASSOCIATE”, “statement!CLASS”, “statement!SELECT TYPE”, “statement!TYPE”
- 28 and “statement!TYPE IS”.
- 29 Combine “storage associated” and “Storage association into “storage association”. Make them, “storage
- 30 sequence” and “storage” unit subsidiary to “storage”.
- 31 Make “structure component” and “structure constructor” subsidiary to “structure”.
- 32 Combine “subobjects” into “subobject”.
- 33 Make “subroutine reference” and “subroutine subprogram” subsidiary to “subroutine”.
- 34 Make “subscript triplet” subsidiary to “subscript”.
- 35 Combine “transformational functions” into “transformational function”.
- 36 Combine “type declaration statements” into “type declaration statement”. Make it, “type compati-
- 37 ble”, “type conformance”, “type equality”, “type incompatible”, “type parameter” and “type specifier”
- 38 subsidiary to “type”. Combine “Type parameter order” into “type parameter order”. Make it, “type
- 39 parameter inquiry”, and “type parameter keyword” subsidiary to “type parameter”.
- 40 Combine “ultimate components” into “ultimate component”.
- 41 Make “unformatted input/output statement” subsidiary to “unformatted data transfer”.
- 42 Combine “use associated” and “Use association” into “use association”.
- 43 Combine “variables” into “variable”.