Subject: Comments on Clause 9
From: Van Snyder

1 Edits — and comments without editorial suggestions

Edits refer to 06-007. 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.

[The POS= specifier in an INQUIRE statement is not restricted to use with files connected for formatted stream access. Editor: Replace ", the" by ", the file position can be set to an arbitrary positive integer or to a position that was previously identified by the POS= specifier in an INQUIRE statement. The".] 210:38

[If a file storage unit contains a record marker, might it contain something else as well? Formatted sequential access files consist of records, that in turn consist of characters, not arbitrary file storage units. Are the file storage units other than record markers of files connected for formatted stream access characters or not? Assuming the answer to the first question is no, and the answer to the second is "characters" ... Editor: Replace "contain" by "be", and before "There" insert "All other file storage units of a file connected for formatted stream access are characters."] 211:4-5

[If a file is connected for formatted stream access its position cannot be set to something other than a position previously identified by a POS= specifier in an INQUIRE statement. Editor: Insert "only" after "set".] 211:16

[So what is the file position after formatted stream output or input? It can't be as explained by [213:9-10] because that paragraph clearly applies to non-stream access.] 213:1-3

[The assertion that every value in a stream file shall occupy an integer number of file storage units is inconsistent with the possibility that some file storage units in a stream file connected for formatted input might contain record markers, at least where record markers occupy more than one character. The Fortran runtime library provides that illusion, but the values "in" the file might occupy more than one file storage unit. Perhaps the sentence should begin "Every value read from or written to a stream file. . . ."] 213:15-16

[The syntax permits only *file-unit-number* so the item says nothing that is not said more precisely by the syntax rules. Editor: Delete item (10).] 214:28-29

[This sentence prohibits internal file I/O. Editor: Insert "that does not refer to an internal file" or "that refers to an external unit" after the second "statement". Alternatively, move [215:28-217:1] to [216:6], inserting it before "The property. . . ."] 216:8

[One needs to refer three paragraphs back to determine that an OPEN statement that specifies an open unit and no FILE= specifier refers to the file to which it is connected. Editor: Move the paragraph at [217:12-14] to be before this paragraph.] 217:20-

---

[Editor: Add "10.6" to the cross references.] 219:29

[Editor: The subclause heading for NEWUNIT= is at the wrong level. Probably should be `\subsubsection`.] 220:16

[The clearest explanation of what PAD does is in item (1) in 9.10.3. Editor: Add 9.10.3 to the list of cross references.] 220:27

[I can't find any support for the assertion in Note 9.20. What does it mean, anyway? What is the default character type for G15.8 format?] 220:29+1-2

[Editor: Delete the extraneous quote.] 222:4

[There still isn't anything wrong with sequential input for team-connected files. Say what it does instead 222:12+

of prohibiting it.]	
[Editor: Insert “or connection” after “file”.]	223:10
[Did we intend for there to be no UNIT= specifier subclause?]	226:25-
[What is the point of “said to be”? Editor: Delete it twice.]	227:17,19
[Editor: Add “10.6” to the cross references.]	228:19
[Editor: For stylistic consistency with [228:32] replace “this” with “the POS=”.]	229:4
[Editor: “*input-item* shall not” ⇒ “effective list item shall neither”. Then move the paragraph to [232:0+].]	230:16
[The term “following” has indefinite extent — it could go to page 600. Editor: “following rules” ⇒ “rules in the remainder of this subclause”.]	230:26-27
[The term “effective list item” is used before it is defined. Editor: Move [231:16-19] to be within this paragraph at the end, and move Note 9.39 to be after the result.]	
[Editor: “is” ⇒ “are”.]	232:1
[“occurred” is the only past-tense verb in the lists; the rest of the list is in the present tense. Editor: “occurred” ⇒ “occurs” at [232:24], [232:28], [233:5] and [233:13].]	232:24-233:13
[The terminology used elsewhere is “pending data transfer operation” (in which “data transfer” ought to be hyphenated because it’s an adjective). Editor: “an asynchronous data transfer” ⇒ a pending data-transfer operation”.]	233:1
[“The . . . operation.” is the subject of the paragraph at [233:15-17]; there’s no point to saying it twice, and subtly differently. Editor: Delete “The . . . operation.”]	233:2-4
[“The conditions . . . statement.” (at the end of the second paragraph of the item) is the subject of the paragraph at [233:23-30]; there’s no point to saying it twice, and subtly differently. Editor: Delete “The conditions . . . statement.” (at the end of the second paragraph of the item).]	233:5-10
[Editor: To incorporate a subtle difference deleted by the edit for [233:5-10], insert “but not both” after “wait operation”.]	233:24
[Editor: So as not to appear to support a conflict between [232:9-10] and [234:4-5], replace the body of Note 9.42 by “The unit may be preconnected. For an output statement, the file need not exist.”]	234:5+2
[READ statements without control lists and PRINT statements have no FMT= specifier; instead, they have only *format*. Editor: “specification . . . specifier” ⇒ “specified by *format*”.]	234:9-10
[Editor: “has been” ⇒ “is”.]	234:11
[Does “entity specified more than once” cover the case of an element of an array specified more than once by a vector subscript?]	234:27
[There are no types other than intrinsic or derived. The paragraph is vacuous. Editor: Delete it.]	235:9-10
[It’s not clear that the real values are the same kind as the complex list entity, as opposed to being the same kind as each other. Editor: “of the same kind stored in the file” ⇒ “stored in the file, of the same kind as the list entity,”]	235:20
[Editor: “The” ⇒ “In an unformatted input/output statement, the”. Then move the paragraph to [234:3+] where it belongs.]	235:34
[There are no types other than intrinsic or derived. The first sentence is vacuous. The second sentence repeats [231:9-12]. Editor: Delete the paragraph.]	236:2-5
[Editor: If Note 9.20 at [220:29+1-2] is true (there appears to be no normative text supporting it), it belongs here.]	236:11+ or 19+

[Editor: If Note 9.20 at [220:29+1-2] is true (there appears to be no normative text supporting it), something similar belongs here.] 236:23+

[For the DTIO read routines, the actual argument associated with the `dtv` dummy argument has to be a variable. Editor: Delete "value/" twice.] 238:10,25

[I could not find any normative support for the POS= and REC= assertions in Note 9.51 at [241:13+1-3]. Editor: Add the following sentences to the end of the paragraph.] 241:9,13+1-3

A child data-transfer statement shall not be a direct-access data-transfer statement. If `unit` is connected for stream access, a POS= specifier shall not appear in a child data-transfer statement.

[Then delete Note 9.51.]

[The type definition in Note 9.52 is not standard conforming. Editor: Insert the following statement after the `CONTAINS` statement:] 241:13+13

```
PROCEDURE :: pwf
```

[The type definition in Note 9.53 is not standard conforming. Editor: Insert the following statement after the `CONTAINS` statement:] 242:Note 9.53

```
PROCEDURE :: pwf
```

[In "data transfer operation", "data transfer" is an adjective, so it ought to be hyphenated. Editor: "data transfer" ⇒ "data-transfer" four times.] 244:2-4

[Item (3) in the list in Subclause 9.5.3 at [232:15-17] says that a synchronous input/output statement performs a wait operation for the unit. A FLUSH statement can't possibly do what it says it does, at least for output, unless it also does a wait operation. Editor: Insert "' FLUSH" after "INQUIRE" (to preserve the reverse alphabetical order) and insert ", synchronous data-transfer" after "CLOSE". Then move the sentence to be the first sentence of the first paragraph of Subclause 9.6.2 at [244:32], which subclause is then moved to be immediately after this paragraph.] 244:5-6

[The sentence appears to include a unit that was opened for asynchronous input/output and then closed and then opened.... Everything else in the paragraph is in present tense. Editor: "was not opened" ⇒ "is not open",] 244:28-29

[It would be clearer if a wait operation were defined before the WAIT statement is described. Editor: Move Subclause 9.6.2 to be before Subclause 9.6.1.] 244:31-245:9

[It would be clearer to describe the restrictions in terms of units instead of files. Editor: "file" ⇒ "unit" thrice, "referred to by" ⇒ "specified in" thrice.] 245:17-20

[I just noticed that it seems to be OK to rewind a unit connected for direct access. Is this really OK? What does it do?] 246:9.7.3

[Why does a FLUSH statement cause data to be made available to other processes, but cause data written by means other than Fortran to be available to a READ statement. What if another Fortran process is writing the file and the present Fortran program wants to read it? Shouldn't it work both ways? Editor: "means other than Fortran" ⇒ "other processes".] 247:24

[The previous paragraph is posed primarily in terms of the unit, not the file. Editor: "file that is connected but" ⇒ "unit that is connected to a file that".] 247:26

[A FLUSH statement can't possibly do what it says it does, at least for output, unless it also performs a wait operation. Editor: Add the following paragraph:] 247:27+

Execution of a FLUSH statement performs a wait operation for all pending asynchronous data-transfer operations for the specified unit.

[By saying "about the file" the impression is conveyed that the specifier is available only for inquiry by file. Rather than say verbosely "specified file if the inquiry is by file or the file connected to the unit if 249:24+

the inquiry is by unit and the unit is connected to a file" for every specifier, spell it out in one place with the following paragraphs.]

All specifiers other than FILE=, IOLENGTH=, and NUMBER= are allowed for inquiry by unit. All specifiers other than UNIT=, ID=, and IOLENGTH= are allowed for inquiry by file.

[Perhaps the previous paragraph should be a constraint.]

The specifiers ACCESS=, ACTION=, ASYNCHRONOUS=, BLANK=, DECIMAL=, DELIM=, ENCODING=, FORM=, NEXTREC=, NUMBER=, OPENED, PAD=, POS=, POSITION=, RECL=, ROUND=, and SIGN= inquire about a property of a connection.

The specifiers DIRECT=, EXIST=, FORMATTED=, NAME=, NAMED=, READ=, READWRITE=, SEQUENTIAL=, SIZE=, STREAM=, UNFORMATTED= and WRITE= inquire about a property of a file. If the inquiry is by file and the file does not exist, or if the inquiry is by unit and either the unit is not connected to a file or is connected to a file that does not exist, the processor is unable to determine the property.

The specifiers ID= and PENDING= inquire about the progress of an asynchronous input/output operation. If there is no connection, an error condition occurs.

[Saying "file is connected" gives the appearance that the ACCESS= specifier is available only for inquiry by file. Editor: "file is connected" ⇒ "connection is" thrice.] 249:38-39

[Saying "file is connected" gives the appearance that the ACTION= specifier is available only for inquiry by file. Editor: "file is connected" ⇒ "connection is" twice, "it is connected" ⇒ "the connection is".] 249:41-250:1

[Saying "file is connected" gives the appearance that the ASYNCHRONOUS= specifier is available only for inquiry by file. Editor: "file is connected" ⇒ "connection is" twice.] 250:5-6

[Editor: Delete "or not".] 250:26-27

[Saying "file is connected" gives the appearance that the ENCODING= specifier is available only for inquiry by file. Editor: "file is connected" ⇒ "connection is" twice.] 250:29-31

[Saying "file is connected" gives the appearance that the FORM= specifier is available only for inquiry by file. Editor: "file is connected" ⇒ "connection is" twice.] 251:5-6

[Editor: Delete "or not".] 251:10

[Is NAME= available for inquiry by unit? If so, what happens if the unit is not connected, or is connected to a file that does not exist?] 251:17+

[Is NAMED= available for inquiry by unit? If so, what happens if the unit is not connected, or is connected to a file that does not exist?] 251:21+

[Saying "the file connected" gives the appearance that the NEXTREC= specifier is available only for inquiry by file. Editor: "file connected" ⇒ "connection", "the file is connected" ⇒ "there is a connection", "file is not connected" ⇒ "there is no connection, the connection is not", insert a comma between "direct access" and "or".] 251:24-26

[Editor: Delete "or not".] 252:15

[Is the PENDING= specifier's result well defined for a team connection?] 252:27+

[Saying "file is connected" gives the appearance that the POSITION= specifier is available only for inquiry by file. Editor: "file is connected by an OPEN statement" [253:4] ⇒ "connection was opened", remaining three "file is connected" [253:5-7] ⇒ "connection was opened".] 253:4-7

[Editor: Delete "or not" twice.] 253:15,20

[Saying "the file" gives the appearance that RECL= is available only for inquiry by file. Editor: "of a file connected" ⇒ "if the connection is" twice, "file is connected" ⇒ "connection is for" twice.] 253:23-26

[Editor: Delete "or not".] 253:40

173	[Editor: Delete “or not” thrice.]	254:16-17,22,26
174 175 176	[The first two sentences duplicate C949. Editor: Replace them by “If a FILE= specifier appears the inquiry is by file. If a UNIT= specifier appears the inquiry is by unit.” Then “The” ⇒ “For inquiry by unit, the”.]	254:29-31
177	[The first paragraph duplicates the syntax, which says it more precisely. Editor: Delete it.]	254:35-36
178 179	[Editor: “An” ⇒ “If an end-of-file condition does not occur, an”. Then delete “, unless. . . ” to the end of the paragraph. Then move the paragraph to [255:17+].]	255:7-9
180 181	[It’s impossible for item (3) at [255:13] to occur if end-of-file cannot occur during a stream input statement. Editor: Delete the sentence.]	255:16-17
182 183	[Editor: For consistency with Subclause 8.2, “execution continues with” ⇒ “a branch to the” and insert “occurs” at the end of the sentence.]	255:36-37
184 185	[Editor: For consistency with Subclause 8.2, “execution continues with” ⇒ “a branch to the” and insert “occurs” at the end of the sentence.]	256:13-14
186	[Editor: “error” ⇒ “end-of-record condition”.]	256:26
187 188	[Editor: For consistency with Subclause 8.2, “execution continues with” ⇒ “a branch to the” and insert “occurs” at the end of the sentence.]	256:34-35
189	[Editor: overfull hbox.]	256:41