

Date: 29 June 1998
 To: J3
 From: Van Snyder
 Subject: Rationale, Specs, Edits for nonadvancing unformatted read

The following is distantly related to Stream I/O. I wrote it up to show how trivial the edits are. It ought not to cause a headache for implementors, or performance problems.

1 Rationale

It would save a little space and time to write variable-length unformatted records without explicit length information. Sometimes one is presented with files containing such records, that have been written by programs over which one has no control; sometimes those programs are written in different languages with a “feature” that allows reading and writing files having the same format as unformatted Fortran files, but the semantics of the language allow reading such records.

2 Specs

To allow reading unformatted records of indeterminate length, allow nonadvancing read for unformatted records.

3 Edits

Edits refer to 98-007r2. 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 immediately following text, while a page and line number followed by + indicates that immediately following text is to be inserted after the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

[Editor: Replace the sentence spanning these lines by the following.] [148:29-30]

A nonadvancing formatted input/output statement, or a nonadvancing unformatted read statement, may position the file within the current record, or at a subsequent record (10.6.2).

[Editor: Add “, or in an unformatted read statement” before the end of the sentence.] [157:16]

[Editor: Delete “advancing formatted”.] [160:1]

[Editor: Change title to “Input count”.] [160:3]

[Editor: Insert “formatted” before “input”.] [160:4]

[Editor: New paragraph] [160:7+]

When a synchronous unformatted input statement terminates the variable specified in the SIZE= specifier becomes defined with the count of the number of items transferred. Each real, integer or logical entity counts as a single item. Each complex entity counts as two items. Each character in a character entity counts as a separate item. For derived type input, each ultimate component is counted separately.

[Editor: Insert “or items” after “characters”.] [160:9]