Issue 128 - Empty Incomplete Record

To: J3
From: Craig Dedo
Date: May 20, 2001
Subject: Issue 128 - Empty Incomplete Record

Issue

The editor writes:
The words in 10.5.3 about linefeeds in A output imply to me that a nonadvancing formatted stream output statement that writes a linefeed as the last character in a stream file will cause there to be an empty and incomplete record at the end of the file. Is this empty incomplete record supposed to be distinguishable from having no record? If so, I wonder how Unix-like systems are supposed to distinguish it. If not, I wonder whether we have it described correctly. Same with / editing, where this was just copied from. These holes leave me unconvinced that the description of record handling "just works" with formatted stream I/O. This related to unresolved issue 127 about handling of incomplete records.

Analysis

This analysis makes use of the ideas developed in paper 01-208, "Design Considerations for Stream I/O".

There are several separate but closely related issues.

1. Does a newline character terminate the current record and start a new record, just like the slash edit descriptor?
2. What happens if a program attempts to write an EOR on a file structure that has no EOR?
3. Can a processor distinguish an empty incomplete record in formatted stream I/O from no record at all?

Whether a newline character is a record delimiter in formatted stream I/O is dependent upon the properties of the host operating system and file system. It is conceivable that a file system could treat a newline character in a number of different ways. Here are some possibilities.

1. A newline character is the only means of delimiting records.
2. A newline character does not have any special properties but is a character just like any other, and records are delimited in some other fashion.
3. Whether the newline character is a record delimiter depends on the file and record structure, and therefore is a property of the file in question.

Therefore, we cannot to assign special properties to any particular character in the character set, whether newline or anything else. The only solution that works well on widely differing file systems is to leave the behavior of an embedded newline character up to the host operating system and file system. Thus, this behavior is processor dependent.

If a file is connected for formatted stream I/O, the only way that a program can write a record delimiter is to use the slash edit descriptor. Even using a slash does not always guarantee a new record. If the host file system defines the file and record structure as having no record markers at all, trying to write a record marker will have no effect. One could plausibly argue that trying to write a record marker in a file that is defined as having no such thing is an error and should produce an error condition. I believe that the "no effect" behavior is more user-friendly. If there is significant disagreement over this issue, we can have a straw vote on it.

Likewise, it may or may not be possible to identify an empty incomplete record at the end of a file. If a file system supports both end-of-record markers and end-of-file markers for the file under

consideration, then it will be possible to distinguish an empty incomplete record from no record at all. If the file system does not support either end-of-record markers or end-of-file markers for some or all kinds of files, then it will not be possible to make such distinctions. Thus, this issue is necessarily processor-dependent.

Edits

Edits are with respect to the 01-007r1.

[221:7-13] Delete the paragraph.

[221:14-23] Delete the J3 note.

[221:24-26] Delete Note 10.15.

References

01-007r1, Fortran 2000 Draft
99-110r1, Stream I/O - Suggested Changes (Unresolved Issue 68)
01-191, Changes to List of Unresolved Issues
01-192, Open Unresolved Issues
01-208, Design Considerations for Stream I/O
01-209, Issue 127 - End-of-File in Formatted Stream Files

[End of J3 / 01-210]