Representation of EOR in Formatted Stream I/O

To: J3
From: Craig Dedo
Date: June 5, 2001
Subject: Representation of EOR in Formatted Stream I/O

Should the character represented by the linefeed character, ACHAR(10), or any other character, be used to delimit the end-of-record (EOR) in formatted stream I/O?

There is a proposal in section 2.2 of 01-193r1 to replace the current specification in [221:7-13] with a named constant specified in the ISO_FORTRAN_ENV module. Another alternative is to simply depend on the existing practice of using the slash (/) edit descriptor to specify EOR.

This paper presents the various alternatives, their advantages and disadvantages, and proposes a straw vote on which alternative to implement in Fortran 2000.

Analysis

Stream I/O was originally proposed by WG5 work items 63 and 63a as a Fortran 2000 feature so that Fortran programs could read from and write to files in formats that are commonly used by C language processors and in binary formats that have little or no internal record structure [98-209r2].

It is clear from the history of the stream I/O feature that compatibility with files read from and written to by C language processors is mandatory. Other design principles and assumptions are listed in 01-208.

It is commonly believed that the linefeed character, represented by ACHAR(10) in the ASCII collating sequence, is the way that C language processors universally represent the end of a line of text. Although very widespread, the C language standard does not mandate this practice. Section 5.2.1 of the C standard says, in part, “In source files, there shall be some way of indicating the end of each line of text; this International Standard treats such an end-of-line indicator as if it were a single new-line character. In the basic execution character set, there shall be control characters representing alert, backspace, carriage return, and new line.”

Section 7.19.2 specifies the properties of C file streams. It says, in part, “A text stream is an ordered sequence of characters composed into lines, each line consisting of zero or more characters plus a terminating new-line character. Whether the last line requires a terminating new-line character is implementation-defined. Characters may have to be added, altered, or deleted on input and output to conform to differing conventions for representing text in the host environment. Thus, there need not be a one-to-one correspondence between the characters in a stream and those in the external representation.”

It is highly unlikely that vendors of Fortran processors will develop an implementation of formatted stream I/O that is incompatible with the needs of C language processors that use the same host operating system and file system. The history of the stream I/O feature makes it clear that compatibility with files written to and read from C is the top priority. Customers will demand that Fortran processors achieve this goal.

Right now, there are three possible methods under consideration for representing EOR in formatted stream I/O.

1. Use ACHAR(10) as the EOR character.

2. Use a character designated by a named constant in the ISO_FORTRAN_ENV module.
3. Use only the slash (/) edit descriptor.

Following is a survey of the advantages and disadvantages of each alternative.

ACHAR(10) as EOR.

Advantages.

1. Provides a plausible form of portability between different Fortran processors on the same platform.

Disadvantages.

1. Possible conflicts with standards and conventions of some host operating systems and file systems. They may allow ACHAR(10) as a legal character within a record. Paper 01-210 discusses these possible conflicts in some detail.
2. Redundant with respect to slash edit descriptor.

Named Constant as EOR

Advantages.

1. Allows processor freedom of implementation and ability to adapt to the peculiar conditions of the host environment.
2. Does not assign peculiar magic properties to any articular character.

Disadvantages.

1. May introduce incompatibilities between different processors in the same host environment.
2. May create files that cannot be read from C.
3. May not be able to read files created and written to by C.
4. Redundant with respect to slash edit descriptor.

Slash (/) Edit Descriptor Only as EOR

Advantages.

1. Already part of Fortran, hence its behavior is already well understood.
2. Portable between different Fortran processors on the same host operating system and file system.
3. Portable between different host operating systems and file systems.
4. Can be implemented in stream I/O in a way that is compatible with the host operating system and host file system.
5. Can be implemented in stream I/O in a way that is compatible with the conventions used by C processors in the host environment.

Disadvantages.

1. May create files that cannot be read from C.
2. May not be able to read files created and written to by C.

Straw Vote: How should we represent EOR in formatted stream I/O?

Option 1: The linefeed character, ACHAR(10), represents EOR.
Option 2: Use a character represented by a named constant in the ISO_FORTRAN_ENV module.
Option 3: Use only the slash (/) edit descriptor.
Undecided

Current Language

Here is the current language in section 10.6.3 [221:7-13].

> If the file is connected for stream access, the output may be split across more than one record if it contains newline characters. A newline character is the character returned by the intrinsic function reference ACHAR(10). Beginning with the first character of the output field, each character that is not a newline is written to the current record in successive positions; each newline character causes file positioning at that point as if by slash editing (the current record is terminated at that point, a new empty record is created following the current record, this new record becomes the last and current record of the file, and the file is positioned at the beginning of this new record).

References

01-007r1, Fortran 2000 Draft
01-193r1, Miscellaneous Remarks
01-208, Design Considerations for Stream I/O
01-210, Issue 128 - Empty Incomplete Record
98-209r2, Specs and Syntax for M.25, Stream I/O
98-211r2, Edits for M.25, Stream I/O
99-110r1, Stream I/O - Suggested Changes (Unresolved Issue 68)
Compaq Computer Corporation, *Guide to OpenVMS File Applications*, Chapter 2, “Choosing a File Organization” (Web site: www.openvms.compaq.com:8000/72final/4506/4506_pro)
ISO/IEC 9899:1999, International Standard - Programming Languages - C

[End of J3 / 01-240]