

Re: backspace error (reading file from the end)

Source: <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2006-08/msg00786.html>

- *From:* Richard Maine <nospam@xxxxxxxxxxxxxx>
 - *Date:* Mon, 7 Aug 2006 09:22:52 -0700
-

On Sun, 6 Aug 2006 23:03:51 -0700, leaf wrote
(in article <1154930631.268787.138040@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>):

BACKSPACE cannot be used to skip over records that have been written using list-directed or namelist formatting.

That's an "interesting" restriction in the standard. I don't think it is ever enforced because the records written by list-directed and namelist formatting are indistinguishable from those written with explicit formatting (and then there is the matter of records written with a mix of formatting methods with `advance='no'`). This restriction is basically about the fact that you don't generally know how many records are written by a list-directed or namelist write. I'm not sure why the standard bothers with the restriction as there are things that are otherwise perfectly safe that you could do. Seems to me that the standard is just prohibiting this in case you might do something wrong with it.

The I/O unit number must specify an open file on disk or magnetic tape.

Not necessarily so. The standard certainly says no such thing; it does not mention disk or tape. However, many implementations have restrictions like this. The details of the restrictions vary.

The fact that some implementations disallow backspace on some files (and yes, the standard allows an implementation to place such limitations) is one reason why I recommend avoiding the use of backspace. It is annoying when you find that your program doesn't work when it takes input from a pipe instead of a disk file, for example.

Backspacing from the current record `n` is performed by rewinding to the start of the file and then performing `n-1` successive READs to reach the previous record.

Re: backspace error (reading file from the end)

The standard says nothing like this. I didn't even think it was a particularly common implementation choice. Why would it be for formatted files in most current systems? I'd expect the system to be able to search backwards for the previous record terminator pretty easily. Since I tend to avoid BACKSPACE, I guess I'm not sure on this – I probably wouldn't have noticed. But still, such an implementation would moderately surprise me.

A BACKSPACE statement must not be specified for a file that is open for direct or append access, because n is not available to the Fortran I/O system.

There is no such thing as append access (anyway, not in standard Fortran). There is position='append', but not access='append'. There is nothing wrong with doing backspace with position='append'. I would also argue that the issues here have little to do with having n available to the Fortran I/O system.

For the OP, I second the advice of Jan and dpb.

--

Richard Maine | Good judgment comes from
experience;
email: my first.last at org.domain| experience comes from bad judgment.
org: nasa, domain: gov | -- Mark Twain

.