

Re: advance="no" not portable

Source: <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2005-05/msg00837.html>

- *From:* Richard E Maine <nospam@xxxxxxxxxxxxxx>
 - *Date:* Fri, 27 May 2005 13:51:59 -0700
-

In article <1117224920.753751.3200@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, "Eric Lavigne" <lavigne.eric@xxxxxxxx> wrote:

```
> WRITE(2,*,ADVANCE="NO") '*'
```

Your problem is not with the advance='no'. It is with the list-directed formatting (indicated by the first "*" above).

The standard doesn't even allow advance='no' with list-directed formatting, and you have found **EXACTLY** the reason why not. Using the two together is misleading. It makes you think that you have specified something that you haven't really done.

The standard allows a compiler is allowed to start new records quite freely in list-directed output. Thus, if the standard allowed the above combination at all, a compiler could validly accept it and do nothing differently than if you had specified advance='yes'. In particular, the vendor could claim that a new record was started as part of the list-directed formatting instead of as part of an advance='yes'.

The "universal" advice relating to list-directed output applies here. List-directed output is simple and handy, but it does not allow you precise control of the formatting. In using list-directed formatting, you abdicate detailed decisions to the compiler, and the results **WILL** vary among compilers.

In your case above, it is simple to just use the explicit format '(a)' instead of list-directed formatting. I would expect that to do what you wanted. In other cases, it can be a bit more bother, sometimes best handled by doing a list-directed write to an internal file (character variable), and then writing out that character variable with an explicit format; this wasn't standard-conforming in f77, which made some things quite a pain.

Note, by the way, that the prohibition against list-directed output with advance='no' is a constraint in the standard. This means that the compiler is in violation of the standard if it is not able to diagnose this as an error (though such diagnosis does not have to be enabled by default). Did you use the compiler options for diagnosing violations of

Re: advance="no" not portable

the standard? If not, I recommend getting in the habit of doing so, not necessarily on every compilation, but on occasion, and particularly when diagnosing problems.

If you did use such options and the compilers failed to diagnose this, then I suggest submitting a bug report.

As long as the compilers have the capability of diagnosing the violation of the constraint, what they otherwise do with it would be an extension and thus any behavior could be considered as "right". But even without resorting to calling it an extension, I would interpret the standard as allowing either behavior if the constraint against this were deleted/

--

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

• **References:**

- ◆ [advance="no" not portable](#)
◇ From: Eric Lavigne

- Prev by Date: [Re: static function ? how ?](#)
- Next by Date: [Re: static function ? how ?](#)
- Previous by thread: [advance="no" not portable](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)