Date: 5 May 1997
To: J3
From: Van Snyder
Subject: Comments about J3/97-102, "Syntax and edits for asynchronous I/O"

On page 6 in paragraphs 2–3, 97-102 proposes

In rule 912 (io-control-spec) (page 144), add:

> or ASYNCHRONOUS

I prefer ASYNCHRONOUS=*scalar-default-logical-expr*. This would allow

```
open (unit=u, file=f, ASYNCHRONOUS=async)
```

instead of requiring

```
if (async) then
  open (unit=u, file=f, ASYNCHRONOUS, ...)
else
  open (unit=u, file=f, ...)
endif
```

On page 7, in the first paragraph for *9.4.1.10 Asynchronous specifier* 97-102 proposes

> ... Records read or written by asynchronous data transfer statements will be read, written and processed in the same order they would have been if the data transfer statement did not contain the ASYNCHRONOUS specifier.

In the case of direct access, this could be relaxed a little bit, requiring only that writes be processed in the order initiated, and that the relative order of reads and writes be preserved. This would require a facility for a separate "user id" for each data transfer statement, so the user could know which transfer has completed upon execution of a WAIT statement. This has other uses described below.

In paragraph 2 on page 8, 97-102 proposes

> ... variables in the namelist group not actually written by the data transfer statement may be redefined before the corresponding wait operation.

Maybe I'm mistaken, but I always thought that a namelist WRITE wrote everything in the namelist. That is, the set to which the above remark applies is empty. If so, it should be deleted. The same remark applies to paragraph 4 on page 10.

In paragraph 7 on page 9, 97-102 asks

> Question: Should we allow scalars? Can they be passed by copyin/out? Any other ways to force pass by address or descriptor?

Can't we just allow anything to be passed to a procedure for which the interface is explicit and the corresponding dummy argument is ASYNCHRONOUS?

Concerning the WAIT statement (described beginning on page 11):

I have used systems that include a "wait for any I/O to complete" request. This is very useful in interactive and real-time applications. It requires a facility to attach a separate user-id to each data transfer request, so that when WAIT continues, the program can know which data transfer has completed.

In paragraph -3 on page 12, I would like to add

or ASYNCHRONOUS = *scalar-default-logical-variable*

so that one can find out if ASYNCHRONOUS=.TRUE. was specified in the OPEN statement.

At the bottom of page 12, there appears to be no requirement that a unit number is specified if ID= and PENDING= specifiers are present (it seems to be implied by the context in which the proposed change will be inserted).

If we don't allow "wait for any request to finish" (and maybe even if we do) it would be useful to allow "INQUIRE (PENDING=foo)" to answer the question "are there *any* asynchronous data transfers in progress?"

Nit Picking:

In paragraph 3 on page 12, "IOSTAT specifier" should be "IOSTAT= specifier" (just for stylistic consistency).

At the end of line 5 of paragraph 7 on page 12, "an" should be "a".

In line 5 of paragraph 2 on page 15, “simplifys” should be “simplifies”.

In the last line of “Model 2” (paragraph -2 or -3, depending on how you count) on page 15, “free wait” should be “free to wait”.

In line 2 of paragraph -1 on page 15, “an” should be “a”.

In line 4 of paragraph -1 on page 15, “to then” should be “then to”.

In line 5 of paragraph 2 on page 16, “to only” should be “only to”.

In line -2 of paragraph 2 on page 16, “encumbent” should be “incumbent” (My Webster says “resting on, as a duty”). “only pass in” should be “pass in only”.

In line -1 of paragraph 2 on page 16, “detetct” should be “detect”.

In line 2 of paragraph 3 on page 16, “ERROR conditions” should be “ERR condition”.