Subject: Oh No! Not more left hand functions!
From: Van Snyder

1 Introduction

Meeting 176 paper 06-154r4 changed the definition of *variable* from *designator* to *designator* or *expr*. This allows `A+B=C+D` so long `A+B` is a reference to a function that returns a pointer result. Similarly, it allows `READ(1,IOSTAT=A+B)` so long as A+B is a reference to a function that has a scalar integer pointer result, etc.

I didn't intend this when I proposed the functionality, but others may have seen this as a natural and desirable consequence.

Is it what J3 wants?

1.1 Malcolm responds

It was certainly envisaged by some for the argument association case at least (it was a U.K. proposal – I know at least one of us thinks that a function reference is a function reference regardless of syntax).

I agree that `A+B=C` looks pretty strange. However, more realistic examples like

```
    table.HASH.key = new_value
```

or

```
    .REF.swizzled_pointer = value
```

look reasonable. Many people like using infix notation for this kind of thing.

2 Edits — Assuming it's not what J3 wants

Edits refer to 06-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

or *function-reference* 115:7

C602 (R601) The *function-reference* shall be a reference to a function that has a pointer result. 115:9

3 On the other hand ...

If we wish to allow `READ(1,IOSTAT=A+B)` so long as A+B is a reference to a function that has a scalar integer pointer result, etc., but not `A+B=C+D`:

C716a (R734) If *variable* is *expr*, *expr* shall be *function-reference*. 157:3+

4 More stuff

[Note 9.35 isn't quite right any more. Editor: After "references" insert "that does not have a pointer result", and delete "may . . . but" so as not to give the impression by omission that an expression whose result isn't a pointer cannot appear in an output list.] 230:17+2-3