

Subject: PASS\_OBJ is ugly syntax, and semantics could be better, too  
 From: Van Snyder

## 1 Introduction

PASS\_OBJ is an ugly keyword, that is unlikely to mean anything to people reading a code who aren't experts in the nuances of Fortran. Furthermore, where a type-bound or object-bound procedure is invoked, if the first dummy argument is of the type to which it is bound, it is likely that the programmer will want to pass the object by which the procedure is invoked to that argument. So the default is upside-down.

This paper proposes to replace PASS\_OBJ with PASS and NOPASS, and with more sensible defaults. There are two possibilities concerning the interface of the bound procedure:

- (1) If the interface has no argument of the type to which it is bound, there is no passed-object dummy argument. It is permitted to confirm this by specifying a NOPASS attribute. It is not permitted to specify the PASS attribute.
- (2) If the interface has an argument of the type to which it is bound, then
  - (a) It is permitted to specify that there is nonetheless no passed-object dummy argument by specifying the NOPASS attribute.
  - (b) If neither PASS nor NOPASS is specified and the first argument has the same type as the type to which the interface is bound, it is the passed-object dummy argument; otherwise there is no passed-object dummy argument.
  - (c) If PASS is specified, it confirms that the first argument shall have the same type as the type to which the interface is bound, and it is the passed-object dummy argument.
  - (d) If PASS(*arg-name*) is specified, *arg-name* is required to be the name of an argument that has the same type as the type to which the interface is bound, and it is the passed-object dummy argument.

Three other problems repaired are that C438 [45:32] and C453 [47:1-5] use "dummy variable" where they ought to use "dummy argument." C458 [47:18-20] largely duplicates C453 [47:2-5].

## 2 Edits

Edits refer to 02-007r1. 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 PASS [ ( <i>arg-name</i> ) ]	45:26
or NOPASS	

---

C438 (R436) If PASS appears without (*arg-name*), the procedure component shall have an explicit interface. The first dummy argument shall meet the requirements for a passed-object dummy argument (4.5.1.6). It shall be polymorphic if and only if type *type-name* is extensible. 45:31-33

C438<sup>1</sup>/<sub>3</sub> (R436) If PASS(*arg-name*) appears, the procedure component shall have an explicit interface with a dummy argument named *arg-name* that meets the requirements for a

1	passed-object dummy argument (4.5.1.6). That dummy argument shall be polymorphic	
2	if and only if type <i>type-name</i> is extensible.	
3	C438 $\frac{2}{3}$ (R436) PASS and NOPASS shall not both appear in the <i>proc-component-attr-spec-list</i> .	
4	R444 <i>binding-attr</i> is PASS [ ( <i>arg-name</i> ) ]	46:38
5	or NOPASS	
6	C453 (R440) If PASS appears without ( <i>arg-name</i> ), the interface specified by <i>abstract-interface-</i>	47:2-5
7	<i>name</i> or each <i>procedure-name</i> specified by <i>binding</i> shall have an explicit interface. The	
8	first dummy argument shall meet the requirements for a passed-object dummy argument	
9	(4.5.1.6). It shall be polymorphic if and only if type <i>type-name</i> is extensible.	
10	C453 $\frac{1}{3}$ (R440) If PASS( <i>arg-name</i> ) appears, the interface specified by <i>abstract-interface-name</i> or	
11	each <i>procedure-name</i> specified by <i>binding</i> shall have an explicit interface with a dummy	
12	argument named <i>arg-name</i> that meets the requirements for a passed-object dummy	
13	argument (4.5.1.6). That dummy argument shall be polymorphic if and only if type	
14	<i>type-name</i> is extensible.	
15	C453 $\frac{2}{3}$ (R440) PASS and NOPASS shall not both appear in the <i>binding-attr-list-list</i> .	
16	[Editor: "PASS_OBJ" $\Rightarrow$ "PASS" twice.]	47:6,9
17	[Editor: Delete.]	47:7-8
18	[Editor: Delete "PASS_OBJ ... extensible" because it's covered by C453 and C453 $\frac{1}{3}$ .]	47:18-20
19	[Editor: "PASS_OBJ" $\Rightarrow$ "PASS".]	52:5+6
20	If a procedure pointer component or type-bound procedure has the PASS attribute, it has	53:2-5
21	a distinguished argument called the <b>passed-object dummy argument</b> . The passed-object	
22	dummy argument affects type-bound procedure overriding (4.5.3.2) and argument association	
23	(12.4.1.1). It shall be a scalar, nonpointer, nonallocatable dummy argument of the type being	
24	defined.	
25	If the PASS attribute is specified without ( <i>arg-name</i> ), the first dummy argument is the passed-	
26	object dummy argument. If the PASS attribute is specified with ( <i>arg-name</i> ), the dummy ar-	
27	gument named <i>arg-name</i> is the passed-object dummy argument. If neither PASS nor NOPASS	
28	is specified and the first dummy argument meets the requirements for a passed-object dummy	
29	argument, it is the passed-object dummy argument and the procedure pointer component or	
30	type-bound procedure implicitly has the PASS attribute. Otherwise there is no passed-object	
31	dummy argument, and the procedure pointer component or type-bound procedure does not	
32	have the PASS attribute.	
33	[Editor: "specify PASS_OBJ" $\Rightarrow$ "have a passed-object dummy argument".]	58:6
34	(4 $\frac{1}{2}$ ) Passed-object dummy arguments, if any, shall correspond.	58:9+
35	[Editor: "PASS_OBJ" $\Rightarrow$ "PASS".]	58:16+6
36	[Editor: "PASS_OBJ is applicable" $\Rightarrow$ "there is a passed-object dummy argument (4.5.1.6)".]	266:11
37	[Editor: "4.5.1" $\Rightarrow$ 4.5.1.6".]	266:12
38	<b>12.4.1.1 The passed-object dummy argument and argument association</b>	266:16
39	[Editor: "with the PASS_OBJ attribute" $\Rightarrow$ "that has a passed-object dummy argument	266:17,20
40	(4.5.1.6)" twice.]	