

Subject: Syntax for the Modules TR  
 From: Van Snyder  
 References: 02-277, 02-324, 02-325

## 1 Submodules

A new program unit, the submodule, is defined. It is introduced by a statement that declares its name, and the name of its parent module or submodule:

```

SUBMODULE ( parent-name ) submodule-name
and terminated by an end statement of the usual kind:
END [ SUBMODULE [ submodule-name ] ]

```

## 2 Procedures that have separate interface and implementation

Substantial controversy remains within subgroup concerning the syntax for interface bodies for procedures that have a separate implementation, and for the implementations of those procedures. A generally desirable feature of the syntax, but not a requirement, is that the syntax of declaration of the interface of a procedure in an interface body should be the same as the syntax of the declaration of its interface in the implementation of the procedure body (provided the interface is redeclared in the implementation, which is optional).

The syntax options are presented separately below, but they should be considered together, for reasons stated in the previous paragraph.

### 2.1 Interface body for a module procedure

#### Interface block for procedure with separate body

Shall the interface body for a module procedure have exactly the same syntax as any other interface body, but be declared within a distinguished interface block:

```

SEPARATE INTERFACE [ generic-spec ]
! interface bodies for procedures with separate implementations
END INTERFACE [ generic-spec ]

```

This is analogous to ABSTRACT INTERFACE.

#### Interface body for procedure with separate body

Or shall the syntax of an interface block remain as described in the current draft, but the syntax of the interface body shall have a prefix to indicate that the implementation of the procedure body is separate:

```

INTERFACE [ generic-spec ]
SEPARATE subroutine-or-function-statement
! declaration of characteristics
END subroutine-or-function name
END INTERFACE [ generic-spec ]

```

33 **Issues**

34 The prefix is negotiable; subgroup prefers SEPARATE. Subgroup believes MODULE INTER-  
35 FACE would be misleading, as this does not declare the interface for a module; it declares the  
1 interface for a procedure that has a separately-defined body.

2 If the interface block indicates that its contained interface bodies are for procedures with sepa-  
3 rate implementations, the fact of this context might be declared some distance from the interface  
4 body.

5 If the interface body indicates that it is for a procedure with a separate implementation, the  
6 syntax for declaration of the implementation body ought to be the same (but this is only  
7 desirable – it's not a requirement).

8 **2.2 Separate procedure body**

9 Subgroup agrees that an ordinary subprogram declaration, in an ordinary context, is the decla-  
10 ration of a complete subprogram, not the declaration of the implementation of a body that has  
11 a separately-declared interface. If it happens to have the same name as an interface body for a  
12 procedure that has a separately-implemented body, and the interface body is accessed by host  
13 association, it overrides that name. If it happens to have the same name as such an interface  
14 body but it appears in the same program unit as that interface body, it is an error (just like it  
15 is now).

16 Subgroup almost agrees that if any characteristic other than PURE is declared in the definition  
17 of the implementation of the procedure body, they shall all be declared, and the interface  
18 declared in the procedure body shall be identical to the interface declared by the interface body,  
19 except that the procedure body can specify that the procedure is pure even if the interface does  
20 not (but not vice-versa). The only point of disagreement is whether specifying whether the  
21 declaration that the implementation is for a subroutine or function (which is a characteristic)  
22 requires redeclaration of all characteristics. This affects the controversy about syntax described  
23 below.

24 If a procedure is to be recursive, it is necessary so to specify on the procedure header for the  
25 separate body.

26 Controversy remains concerning the syntax to indicate the definition of the body of a procedure  
27 that has a separately-declared interface.

28 **A separate statement**

29 The implementation of a module procedure that is separate from its interface body shall be  
30 immediately preceded by a statement that indicates it is the implementation of a separate  
31 procedure:

32 `IMPLEMENTATION procedure-name`

33 The keyword for this statement is negotiable; subgroup prefers IMPLEMENTATION.

34 If one wishes to redeclare the characteristics of the procedure, to declare that the body is pure  
35 even though the interface doesn't require it, or to declare a property of the procedure that  
36 is not a characteristic but that can only be declared on the *subroutine-stmt* or *function-stmt*  
37 (i.e., RECURSIVE or RESULT), a *subroutine-stmt* or *function-stmt* shall immediately follow  
38 the IMPLEMENTATION statement, and all of the characteristics of the procedure shall be  
39 respecified.

40 **Prefix of a procedure header**

41 The body of a module procedure that is separate from its interface body shall be introduced by  
 42 a *subroutine-stmt* or a *function-stmt* that has a prefix that indicates it is the implementation  
 43 of the body of a procedure that has a separately-declared interface. If the dummy argument  
 44 list is replaced by the usual way that we indicate “assumed”, *viz.* by a colon in parentheses,  
 1 the interface is assumed from the separately-declared interface body, except that PURE can  
 2 be specified independently of whether it is specified in the separately-declared interface body.  
 3 Otherwise the interface shall be completely specified, and shall agree with the interface declared  
 4 by the separately-declared interface body.

```
5  prefix SUBROUTINE subroutine-name (:)
6      ...
7  END SUBROUTINE subroutine-name

8  prefix FUNCTION function-name (:) [ RESULT ( result-name ) ]
9      ...
10 END FUNCTION function-name
```

11 The *prefix* shall include the word that indicates that it introduces the body of a procedure that  
 12 has a separately-declared interface. Subgroup prefers SEPARATE, but the word is negotiable.  
 13 It may include RECURSIVE or PURE without requiring that all characteristics be specified.  
 14 If any other characteristic is specified, all characteristics shall be specified.

15 **Issues**

16 There isn't a comfortable place to put RECURSIVE, PURE or RESULT on the IMPLEMENTATION  
 17 statement. If an IMPLEMENTATION statement is to be used, subgroup believes  
 18 there is not choice but to respecify the entire interface if one of the non-characteristic declara-  
 19 tions, or PURE, is needed.

20 It's not obvious what the END statement that goes with the IMPLEMENTATION statement  
 21 should be. Should it be END IMPLEMENTATION always, or only when a SUBROUTINE or  
 22 FUNCTION statement does not appear? Should it be END SUBROUTINE or END FUNC-  
 23 TION always or only when a SUBROUTINE or FUNCTION statement appears?

24 The IMPLEMENTATION statement has to immediately precede the SUBROUTINE or FUNC-  
 25 TION statement. Although we have statement ordering requirements, we have only two other  
 26 requirements for one statement to immediately precede another: SELECT CASE and SELECT  
 27 TYPE.

107 Subgroup could not agree whether (:) was acceptable.