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

1 Submodules

A new program unit, the submodule, is defined. It is surrounded by a statement that declares its name, and a corresponding END statement.

R202 *program-unit* **is** *main-program*
or *external-subprogram*
or *module*
or *submodule*
or *block-data*

R1108a *submodule* **is** *submodule-stmt*
[*specification-part*]
[*module-subprogram-part*]
end-submodule-stmt

R1108b *submodule-stmt* **is** SUBMODULE (*parent-name*) *submodule-name*

R1108c *end-submodule-stmt* **is** END [SUBMODULE [*submodule-name*]]

C1107a (R1108c) If the *submodule-name* is specified in the *end-submodule-stmt*, it shall be identical to the *submodule-name* specified in the *submodule-stmt*.

2 Procedures that have separate interface and implementation

2.1 Interface body for a module procedure

The interface body for a module procedure that has a separately-defined body has exactly the same syntax as any other interface body, but it is declared within an interface block that has a FORWARD prefix on its initial INTERFACE statement:

R1203 *interface-stmt* **is** [FORWARD] INTERFACE [*generic-spec*]
or ABSTRACT INTERFACE

A **forward interface** is declared by an interface body in an interface block that is introduced by a FORWARD INTERFACE statement.

C1209a (R1206) A *procedure-stmt* shall not appear in an interface block introduced by a FORWARD INTERFACE statement.

2.2 Separate procedure body

The implementation of a module procedure that is separate from its interface body shall be immediately surrounded by an IMPLEMENTATION statement and an END IMPLEMENTATION statement. It is optional whether the interface is redeclared, unless it is necessary to specify RECURSIVE or RESULT, or it is desired to specify that the procedure is pure even though the interface does not require it. If the interface is redeclared the body of the procedure is immediately surrounded by a subprogram header and a subprogram END statement.

R1108 *module-subprogram* **is** *function-subprogram*
or *subroutine-subprogram*
or *implementation*

R1233a *implementation* **is** *implementation-stmt*
[*implementation-body*]
end-implementation-stmt

R1233b *implementation-stmt* **is** IMPLEMENTATION *subprogram-name*

C1252b (R1233b) The *subprogram-name* shall be identical to the name of a forward interface that is accessible by host association from the ancestor module of the scoping unit in which the *implementation* appears.

R1233c *end-implementation-stmt* **is** END [IMPLEMENTATION [*subprogram-name*]]

C1107a (R1233c) If a *subprogram-name* appears in the *end-implementation-stmt*, it shall be identical to the *subprogram-name* specified in the *implementation-stmt*.

R1233d *implementation-body* **is** *function-impl*
or *subroutine-impl*

R1233e *function-impl* **is** *function-subprogram*
or *subprogram-body*

R1233f *subprogram-body* **is** [*specification-part*]
[*execution-part*]
[*internal-subprogram-part*]

C1252c (R1233e) If *function-impl* is *function-subprogram* the *function-name* shall be identical to the *subprogram-name* specified in the *implementation-stmt*.

C1252d (R1233e) If *function-impl* is *function-subprogram* interface declared by *function-impl* shall be identical to the interface declared by the interface body for the *subprogram-name*, except that it may specify PURE even if the interface declared by the interface body does not.

R1233g *subroutine-impl* **is** *subroutine-subprogram*
or *subprogram-body*

C1252g (R1233g) If *subroutine-impl* is *subroutine-subprogram* the *subroutine-name* shall be identical to the *subprogram-name* specified in the *implementation-stmt*.

C1252h (R1233g) If *subroutine-impl* is *subroutine-subprogram* the interface declared by *subroutine-impl* shall be identical to the interface declared by the interface body for the *subprogram-name*, except that it may specify PURE even if the the interface declared by the interface body does not.

C1258a (R1234) An *entry-stmt* shall not appear in an *implementation-body*.