

Subject: EXIT from any labeled construct
 From: Van Snyder
 Reference: 03-258r1, section 2.1.2, 04-109, 04-156

1 Number

2 TBD

3 Title

4 EXIT from any labeled construct.

5 Submitted By

6 J3

7 Status

8 For consideration.

9 Basic Functionality

10 Allow EXIT from any labeled construct.

11 Rationale

12 Some algorithms cannot be expressed in Fortran 2003 without GOTO statements or extra tests, but
 13 they could be expressed with EXIT if it could be applied to any labeled construct. For example, here's
 14 a routine that says "call R if X is not an element of the set S, which is represented by elements of
 15 A(1:num_in_set)." With an extra test:

```
16   do i = 1, num_in_set
17       if ( x == a(i) ) exit
18   end do ! i
19   if ( i <= num_in_set ) call r
```

20 or, with GOTO:

```
21   do i = 1, num_in_set
22       if ( x == a(i) ) go to 10
23   end do ! i
24   go to 20
25 10 call r
26 20 continue
```

27 or, with a more general EXIT:

```
28 o: if ( .true. ) then
29     do i = 1, num_in_set
30         if ( x == a(i) ) exit o
31     end do ! i
32     call r
33 end if o
```

1 7 Estimated Impact

2 Trivial to minor — a few lines in 8.1.6.4.4. Estimated at meeting 169 to be at 3 on the JKR scale.

3 8 Detailed Specification

4 Allow an EXIT statement with a *do-construct-name* that is the same as the name of any enclosing
5 construct, not just a DO construct. When the EXIT statement is executed, execution of the named
6 construct is terminated. For compatibility, an EXIT statement without a *do-construct-name* continues
7 to refer to the nearest enclosing DO construct.

8 8.1 Suggested editorial changes

9 These suggested editorial changes illustrate the magnitude of the project.

10 Replace *do-construct-name* in R844 with *construct-name*. Allow it to be the name of any construct that
11 encloses the EXIT statement. Add a new subclause 8.1.7 that describes the EXIT statement but not
12 loop termination. Specify there that the EXIT applies to the construct named by the *construct-name*.
13 Do not change the interpretation of an EXIT statement that doesn't mention a *construct-name*.

14 It would be helpful if a construct existed that had no purpose other than to have a construct label.

15 Here's an example of a new 8.1.6.4.4

16 8.1.6.4.4 Loop termination

17 A loop terminates, and the DO construct becomes inactive, when any of the following occurs:

- 18 (1) Determination that the iteration count is zero or the *scalar-logical-expr* is false, when tested
19 during step (1) of the above execution cycle,
- 20 (2) Execution of an EXIT statement that belongs to the DO construct,
- 21 (3) Execution of an EXIT statement or a CYCLE statement that is within the range of the DO
22 construct, but that belongs to an outer construct,
- 23 (4) Transfer of control from a statement within the range of a DO construct to a statement that
24 is neither the *end-do* nor within the range of the same DO construct,
- 25 (5) Execution of a RETURN statement within the range of the DO construct,
- 26 (6) Execution of a STOP statement anywhere in the program, or
- 27 (7) Termination of the program for any other reason.

28 When a DO construct becomes inactive, the DO variable, if any, of the DO construct retains its last
29 defined value.

30 Here's an example of a new subclause about the EXIT statement:

31 8.1.7 EXIT statement

32 The EXIT statement provides one way of terminating a construct.

33 R844 *exit-stmt* is EXIT [*construct-name*]

34 C829 (R844) If an *exit-stmt* refers to a *construct-name*, it shall be within the range of that construct;
35 otherwise, it shall be within the range of at least one *do-construct*.

36 An EXIT statement belongs to a particular construct. If the EXIT statement refers to a construct name,
37 it belongs to that construct; otherwise, it belongs to the innermost DO construct in which it appears.

38 When an EXIT statement that belongs to a DO construct is executed, it terminates the loop (8.1.6.4.4).

39 When an EXIT statement that belongs to a non-DO construct is executed, execution continues with
40 the first executable construct after the END statement for the construct to which the EXIT statement
41 belongs.

42 9 History