Subject: Test DO constructs at the end
From: Van Snyder
Reference: 03-258r1, section 2.1.7

## Number

TBD

## Title

Test DO constructs at the end.

## Submitted By

J3

## Status

For consideration.

## Basic Functionality

Test DO constructs at the end, independently of whether they are tested at the beginning.

## Rationale

One sometimes needs to test a DO construct at the end. One can do this with an IF statement and an EXIT statement, but these aren't quite as clear as statements specifically designed for this purpose. That a majority faction of the Fortran 90 committee agreed in principle that this is clearer is obvious from the existence of the DO WHILE statement to introduce a loop.

## Estimated Impact

Minor.

## Detailed Specification

Allow a new statement (or maybe two) that can be used in place of the END DO statement, with a *scalar-logical-expr* that controls whether the loop continues or terminates. Examples:

DO [ *loop-control* ]
    *do-block*
UNTIL ( *scalar-logical-expr* )

is equivalent to but a little clearer than

DO [ *loop-control* ]
    *do-block*
    IF ( *scalar-logical-expr* ) EXIT
END DO

Similarly,

DO [ *loop-control* ]
    *do-block*
WHILE ( *scalar-logical-expr* )

is equivalent to but a little clearer than

```
DO [ loop-control ]
    do-block
    IF ( .not. scalar-logical-expr ) EXIT
END DO
```

For symmetry, allow

```
DO UNTIL ( scalar-logical-expr )
    do-block
END DO
```

which is equivalent to but a little clearer than

```
DO
    IF ( scalar-logical-expr ) EXIT
    do-block
END DO
```

The tests implied by *loop-control* and an UNTIL or WHILE statement at the end of the loop are independent. That is, one can have both *loop-control* at the beginning of the loop, and an unrelated UNTIL or WHILE statement at the end.

## History