Variable Format Expressions

by Craig T. Dedo
April 24, 1996

1. Rationale

A variable format expression (VFE) allows a programmer to use an arithmetic expression at any place in a format edit descriptor where an integer can be used. The current Fortran 90 standard and the Fortran 95 DIS only allow format edit descriptors which are either character expressions or hard coded into the source code.

A VFE provides the application programmer a convenient and straightforward method of varying the format edit descriptor at run time, as conditions require. Many application programmers consider VFEs to be easier to use and less error prone than the only current alternative of building format specifications in character expressions. Thus, VFEs offer the application developer an opportunity to save on programming time and effort, with a consequent increase in productivity.

It is the intent of this proposal to ratify existing practice. A survey of existing compilers indicates that most implementations which offer this feature do so in a similar, if not identical, manner.

2. Technical Specification

A variable format expression is a scalar numeric expression enclosed inside angle brackets, <>. A VFE can be used in a format edit descriptor any place that an integer can. During program execution, the VFE is re-evaluated each time it is encountered in the normal format scan.

If the VFE is not of integer data type, it is converted to integer type before it is used, according to the usual rules for data type conversion. A VFE may be of any KIND type that is defined on the processor.

The VFE can be any valid scalar numeric expression. It can include dummy arguments, function references, and named constants. It shall not include any array valued data item or any variable with the POINTER attribute. Any function referenced in a VFE shall not execute any I/O operation.

The value of a VFE must obey the restrictions on magnitude applying to its use in the format, or an error occurs.

VFEs shall not be used in a character format specification (10.1.2).

VFEs are evaluated each time they are encountered in the scan of the format. If the value of the variable used in the expression changes during the execution of the I/O statement, the new value is used the next time the format item containing the expression is processed.

Example
Consider the following statements:

```
        INTEGER ::    IWIDTH, IDIGITS, IEXPONENT, NF, NE
        REAL, ALLOCATABLE       ::    FREAL (:), EREAL(:)
!       Intervening statements defining the values of NF and NE
        ALLOCATE     FREAL (NF)
        ALLOCATE     EREAL (NE)
```

```
!        Intervening statements defining the elements of FREAL, EREAL, IWIDTH, IDIGITS,
!            and IEXPONENT.
         WRITE (*, 601)  FREAL, EREAL
601      FORMAT (5X, <NF>F<IWIDTH>.<IDIGITS>, &
         & 5X, <NE>E<IWIDTH+6>.<IDIGITS>E<IEXPONENT>)
```

If, on one execution of the WRITE statement, NF is 5, NE is 3, IWIDTH is 12, IDIGITS is 7, and IEXPONENT is 2, then statement 601 would be equivalent to:

```
601      FORMAT (5X, 5F12.7, 5X, 3E18.7E2)
```

If, on a subsequent execution of the WRITE statement, NF is 6, NE is 2, IWIDTH is 10, IDIGITS is 5, and IEXPONENT is 3, then statement 601 would be equivalent to:

```
601      FORMAT (5X, 6F10.5, 5X, 2E16.5E3)
```

3. Edits to the Standard

These edits are preliminary and are offered primarily as a basis for discussion.

Section 10.2

Change rule R1004 and the following constraints to read:
R1004 *r* **is** *int-literal-constant* or *vfe*
Constraint: *r* shall be positive.
Constraint: *r* shall not have a kind parameter specified for it if *r* is a constant.
The expression *r* is called a **repeat specification.**

Section 10.2.1

Change rules R1006, R1007, R1008, R1009 and the last constraint following them to read:
R1006 *w* **is** *int-literal-constant* or *vfe*
R1007 *m* **is** *int-literal-constant* or *vfe*
R1008 *d* **is** *int-literal-constant* or *vfe*
R1009 *e* **is** *int-literal-constant* or *vfe*

Constraint: *w, m, d,* and *e* shall not have kind parameters specified for them if they are constants.

Change rule R1011 and the constraint following it to read:
R1011 *k* **is** *signed-int-literal-constant* or *vfe*
Constraint: *k* shall not have a kind parameter specified for it if it is a constant.

Change rule R1013 and the constraint following it to read:
R1013 *n* **is** *int-literal-constant* or *vfe*
Constraint: *n* shall not have a kind parameter specified for it if it is a constant.

Section 10.2.3

Add a section 10.2.3 to read:

10.2.3 Variable Format Expressions

A variable format expression is a scalar numeric expression enclosed inside angle brackets, <>. A variable format expression can be used in a format edit descriptor any place that an integer can. During program execution, the variable format expression is re-evaluated each time it is encountered in the normal format scan.

R1017 *vfe* **is** <*numeric-expr*>
Constraint: The *numeric-expr* shall not be array valued.
Constraint: The *numeric-expr* shall not have the POINTER attribute.
Constraint: A *vfe* shall not appear in a character format specification (10.1.2).

Any function referenced in the *numeric-expr* shall not execute any input-output statement.

If the *vfe* is not of integer data type, it is converted to integer type after it is evaluated and before it is used. A *vfe* may be of any KIND type that is defined on the processor.

The value of a *vfe* shall obey the restrictions on magnitude applying to its use in the format.

A *vfe* shall be evaluated each time it is encountered in the scan of the format. If the value of the *vfe* changes during the execution of the I/O statement, the new value is used the next time the format item containing the expression is processed.

4. References

Digital Equipment Corporation. February 1995. *DEC Fortran Language Reference Manual*. Maynard, MA: Digital Equipment Corporation. Section 8.3.7, pp. 8-26 thru 8-28.

Digital Equipment Corporation. April 1995. *DEC Fortran 90 Language Reference Manual*. Maynard, MA: Digital Equipment Corporation. Section 11.6, pp. 11-44 thru 11-45.

International Business Machines Corporation. December 1993. *AIX/XL Fortran Compiler/6000 Language Reference Version 3 Release 1*. North York, ON: International Business Machines Corporation. p. 272.

ISO/IEC 1539-1:1991(E) *International Standard Programming Language Fortran 90*, 10.1.2.
[End of 96-nnn]