Lower Case and Mixed Case Syntax Elements

by Craig T. Dedo
May 15, 1997

1. Rationale

The need to restrict syntax elements to upper case letters for guaranteed portability is a relic from the 1960s. Up until the early 1980s, some machines did not support lower case letters in their character sets. Now, such a restriction is no longer necessary; all platforms with a Fortran 90 processor include support for lower case letters.

Requiring support for lower case letters in Fortran syntax elements would allow programmers to write more visually appealing code. All other major programming languages already support lower case letters in their syntax elements.

Removing this restriction would help remove the image that Fortran still has of being a "Stone Age" relic.

2. Technical Specification

This proposal would require the inclusion of lower case letters in the Fortran character set. It also would require a Fortran processor to make lower case or mixed case syntax elements equivalent to upper case syntax elements. There are two exceptions:

(1) in a character context, i.e., within a character literal constant or a character string edit descriptor; and

(2) eventually, as required by features needed to implement interoperability with C.

It is the intent of this proposal to construe exception (2) as narrowly as possible. The exception for a character string is modified slightly so that character expressions that equate to keywords are case insensitive unless there is a specific constraint that specifies that a character expression for a particular keyword is either case insensitive or processor dependent.

In a Fortran 90 standard-conforming program, symbolic names are case insensitive. Some processors offer, as an extension, a user-selectable option that allows symbolic names to be case sensitive. Under this proposal, symbolic names in a standard-conforming program would still be case insensitive. A processor could continue to offer the user-selectable option as an extension to the standard and not be in violation of the standard.

Example:

If this proposal is adopted, the following constructs would all be equivalent and standard conforming on all processors. In Fortran 90 and Fortran 95, only the first option is standard-conforming on all processors.

```
1:  IF (NDATE < NLAST_DATE) THEN
        NDATE = NDATE + 1
        CALL BIG_COMPLEX_OPERATION (NDATE)
    END IF

2:  if (ndate < nlast_date) then
        ndate = ndate + 1
        call big_complex_operation (ndate)
    end if
```

```
3:   If (NDate < NLast_Date) Then
         NDate = NDate + 1
         Call Big_Complex_Operation (NDate)
     End If
```

3. Edits to the Standard

These edits are with respect to the Fortran 95 draft, 96-007r1.

Section 3.1.1
[21:24-26+] Change the text to read:
"The set of letters defines the syntactic class *letter* . The process or character set shall include lower-case and uppe r-case letters. The lower-case letters are equivalent to the corresponding upper-case letters in program units except in a character context (3.3)."

Example:
The following statements are equivalent.

```
CALL BIG_COMPLEX_OPERATION (NDATE)
call big_complex_operation (ndate)
Call Big_Complex_Operation (NDate)
```

[End of Proposed Edit]

[End of J3 / 97-161r1]