

## #12

## Input Data validation

Written by Ron Dell'Aquila

October, 1987

Input data validation on a character by character basis.

Sometimes it's necessary to validate data as it's being entered into an input layout. For example, part codes must fall between the range "AAA000" and "MMM555".

This validation procedure will evaluate that the first character entered is greater than "A", and less than "M". Similarly, the sixth character entered must lie between the range of "0" and "5".

The following procedure will pass the contents of the field being modified as well as the low and high parameter values to a user defined function. Each field is evaluated using a left to right character precedence. If any of the input characters do not fit the specified range, the errant character will be highlighted, enabling corrections.

In your layout procedure, add the following code:

```
` Local Proc
Case of
  : (During)
    Case of
      : (Modified(Field1))
        vLow:="AAA000"
        vHigh:="MMM555"
        vPointer:=Validate (Field1;vLow;vHigh) `call user defined function
        If (vPointer#0)
          REJECT(Field1)
          HIGHLIGHT TEXT(Field1;vPointer;vPointer+1)
        End if

      : (Modified(Field2))'Second field to be evaluated...
        vLow:="MMM000"
        vHigh:="ZZZ555"
        vPointer:=Validate (Field2;vLow;vHigh)
        If (vPointer#0)
          REJECT(Field2)
          HIGHLIGHT TEXT(Field2;vPointer;vPointer+1)
        End if
    End case
End case
```

The following global procedure is used as a function. The values passed from the above local procedure are received in variables \$1, \$2, and \$3, where \$1 will contain the contents of the field, \$2 will contain the lower range limit and \$3 will contain the upper limit. The While loop will iterate until the pointer reaches the last character in the field, or until an out of range character is encountered.



```
`Global proc: Validate
vPointer:=1      `the sequential pointer
VFlag:=False
While ((vPointer<=Length($1))&(Not(vFlag)))
    vLowChar:=Substring($2;vPointer;1)      `get pointer character from vLow
    vHighChar:=Substring($3;vPointer;1)      `get pointer character from vHigh
    vActChar:=Substring($1;vPointer;1)      `get pointer character from Field1
    If ((vActChar<vLowChar)|(vActChar>vHighChar))      `evaluate input character
        BEEP(5)
        $0:=vPointer
        vFlag:=True
    End if
    vPointer:=vPointer+1      `prepare to evaluate next character
End while
```

The beauty of the above procedures is that the global is used as a function. This means that each field to be validated can use the same global procedure by merely passing the necessary parameters (field, LowValue, and HighValue). If you need to evaluate many fields one function can be used, making your application more efficient

