

11 Parsing Alpha Fields for numbers

Written by Dave Dell'Aquila

Published date: August 1, 1987

Revised date: March 3, 1988

Formatting alpha fields which contain numbers.

Syntax ParsePhone(strexp)

The following code shows how to format a phone number field so that it always appears as ###-###-#### or ###-####. It can accept numbers entered in *any* format, with any number of additional unrelated characters, as long as it contains 7 or 10 numeric characters. The final formatting can easily be modified, and can be extended to format social security numbers, charge card numbers, various account numbers, etc.. You would typically call the procedure from from the During phase of an Input Layout procedure.

```

If (Length($1)#0)                                `if the length is not zero
$1:=ReplaceChar ($1,"-","")                        `replace all dashes with nothing
$1:=ReplaceChar ($1,"e","")                        `replace 'e' with nothing
$1:=String(Num($1))                                `strip out all non-numeric
$L:=Length(String(Num($1)))                        `calculate the length once
Case of                                             `format what's leftover
: ($L=7)
  $0:=String(Num($1),"###-####")                  `format for 7 digits
: ($L=10)
  If ((Substring($1;2;1)#"0")&(Substring($1;2;1)#"1"))
    Alert("Not a valid area code. Please re-enter")
    $0:=String(Num($1),"###-####-####")            `format for 10 digits
  Else
    $0:=String(Num($1),"###-###-####")              `format for 10 digits
  End if
Else                                             `not 7 or 10 characters
  Alert("Not a valid phone number. Please re-enter.")
  If (Length($1)<10)                               `guess the formatting
    $0:=String(Num($1),"#####-####")
  Else
    $0:=String(Num($1),"#####-###-####")
  End if
End case
End if

```

where the *ReplaceChar* function looks like:

```

$pos:=Position($2;$1)                             `check the position of the dash character
While (($pos#0)&($pos#Length($1))                `if it exists and isn't the last char, replace it
$1:=Substring($1;1;$pos-1)+$3+Substring($1;$pos+1;Length($1)-$pos)
$pos:=Position($2;$1)                             `check if it still exists
End while
$0:=$1

```



Example**Case of**: **(Modified**(Phone1))[File1]Phone1:=*ParsePhone*([File1]Phone1)**End Case**