**Converting Numbers to Words**

There are times when you need to spell numbers out. For instance, you may want to spell out "1234" as "one thousand two hundred thirty-four.” Word has no built-in function that will do the conversion for you, so you are left to create a macro that will handle the conversion.

The following macro, BigCardText, will convert any number between 0 and 999,999,999. To use it, simply place the insertion point either within the number you want to convert or just to the right of the number (if it is a single digit). It will check that the selection is a valid integer and does not contain comma separators. If the number is preceded by a $ then the word dollars is added (it will give plural for $1). It also gives sentence case to the whole sentence.

Sub BigCardText()  
    Dim sDigits As String  
    Dim sBigStuff As String  
    Dim rDollar As Range  
    Dim rNextChar As Range  
    sBigStuff = ""  
  
    ' Select the full number in which the insertion point is located  
    Selection.MoveLeft unit:=wdWord, Count:=1, Extend:=wdMove  
    Set rNextChar = Selection.Range  
    rNextChar.End = rNextChar.Start + 1  
    If rNextChar.Text = "$" Then Selection.MoveRight , Count:=1, Extend:=wdMove ' covers case where number including $ is selected  
      
    Selection.MoveRight unit:=wdWord, Count:=1, Extend:=wdExtend  
    If Selection.Range.Text = "." Then GoTo nERR ' insertion point precedes a dot  
          
    If Not IsNumeric(Selection.Text) Then GoTo nERR  
          
    Set rDollar = Selection.Range  
    Set rNextChar = Selection.Range  
    rNextChar.Start = rNextChar.End  
    rNextChar.MoveEnd unit:=wdCharacter, Count:=1  
    If rNextChar.Text = "." Then ' there is a decimal point as next char. It may be an end of sentence  
      rNextChar.Start = rNextChar.End  
      rNextChar.MoveEnd unit:=wdCharacter, Count:=1  
      If rNextChar.Text <> Chr(13) And rNextChar.Text <> " " Then GoTo nERR  
    End If  
    rDollar.MoveStart unit:=wdCharacter, Count:=-1  
    rDollar.End = rDollar.Start + 1  
    If rDollar.Text = "." Or rDollar = "," Then GoTo nERR ' leading decimal point  
  
    ' Store the digits in a variable  
    sDigits = Trim(Selection.Text)  
  
    If Val(sDigits) > 999999 Then  
        If Val(sDigits) <= 999999999 Then  
            sBigStuff = Trim(Int(str(Val(sDigits) / 1000000)))  
            ' Create a field containing the big digits and  
            ' the cardtext format flag  
            Selection.Fields.Add Range:=Selection.Range, \_  
              Type:=wdFieldEmpty, Text:="= " + sBigStuff + " \\* CardText", \_  
              PreserveFormatting:=True  
  
            ' Select the field and copy it  
            Selection.MoveLeft unit:=wdWord, Count:=1, Extend:=wdExtend  
            sBigStuff = Selection.Text & " million "  
            sDigits = Right(sDigits, 6)  
        End If  
    End If  
    If Val(sDigits) <= 999999 Then  
        ' Create a field containing the digits and the cardtext format flag  
        Selection.Fields.Add Range:=Selection.Range, \_  
          Type:=wdFieldEmpty, Text:="= " + sDigits + " \\* CardText", \_  
          PreserveFormatting:=True  
        ' Select the field and copy it  
        Selection.MoveLeft unit:=wdWord, Count:=1, Extend:=wdExtend  
        sDigits = sBigStuff & Selection.Text  
        If rDollar.Text = "$" Then  
          sDigits = sDigits & " dollars"  
          rDollar.Text = ""  
        End If  
        ' Now put the words in the document  
        Selection.TypeText Text:=sDigits  
        Selection.TypeText Text:=" "  
        Selection.Range.Sentences(1).Select  
        Selection.Range.Case = wdTitleSentence  
        Selection.Collapse direction:=wdCollapseEnd  
    Else  
        MsgBox "Number too large", vbOKOnly  
    End If  
  Exit Sub  
    
nERR:  
  MsgBox "Is not an integer"  
End Sub