

Altering Large Numbers Of Cells In Excel A Hundred Times Quicker

Many processes need to write large volumes of data in Excel. The typical method is to loop through each cell and perform the action.

```
Dim CellsDown As Long
CellsAcross As Long
```

```
Dim CurrRow As Long
CurrCol As Long
Dim CurrVal As Long
```

' This can be replaced with the selected range and is just used to illustrate this example.

```
CellsDown = 1000
CellsAcross = 36
```

```
' Loop through cells and insert values
CurrVal = 1
Application.ScreenUpdating = False
For CurrRow = 1 To CellsDown
    For CurrCol = 1 To CellsAcross
        Range("A1").Offset(CurrRow - 1, CurrCol - 1).Value
= CurrVal
        CurrVal = CurrVal + 1
    Next CurrCol
Next CurrRow
```

Rather than writing the values out cell by cell, it is quicker to store the value in an array and write the array to a range of cells at one time.

```
Dim CellsDown As Long
CellsAcross As Long
```

```

Dim CurrRow As Long
CurrCol As Long
Dim CurrVal As Long

Dim TempArray() As Double

' This can be replaced with the selected range and is just
used to illustrate this example.
CellsDown = 1000
CellsAcross = 36

' Update the array
ReDim TempArray(1 To CellsDown, 1 To CellsAcross)
Set TheRange = Range(Cells(1, 1), Cells(CellsDown,
CellsAcross))

' Fill the temporary array
CurrVal = 0
Application.ScreenUpdating = False
For i = 1 To CellsDown
    For j = 1 To CellsAcross
        TempArray(i, j) = CurrVal
        CurrVal = CurrVal + 1
    Next j
Next i

' Transfer temporary array to worksheet
TheRange.Value = TempArray

```

This same method can be used when altering data. By changing the following line

```
TempArray(i, j) = CurrVal
```

To this

```
TempArray(i, j) = TheRange(i, j) * 3
```

By using TheRange(i, j), the existing value can be altered

The process of writing values cell by cell took 3.16 seconds.
Using the array method, it took .08 seconds, nearly 40 times
faster