Date: 04/05/2012
Procedure: Excel - VBA - Referring To Ranges In Your Code
Source: LINK
Permalink: LINK
Created by: HeelpBook Staff
Document Version: 1.0

EXCEL – VBA – REFERRING TO RANGES IN YOUR CODE

Newcomers to **VBA** are often confused about how to refer to ranges on a worksheet. This confusion is somewhat justified, because **VBA** offers several different ways to refer to ranges. In this document I provide an overview of these techniques:

- Referring to ranges directly
- Referring to ranges by using the Cells method
- Referring to ranges by using the Offset method

Referring to ranges directly

Perhaps the most common way to refer to a range on a worksheet is to specify the range directly. Here's an example that assigns the value 1 to range **A1:C12** on **Sheet1** in a workbook named MyBook:

```
Workbooks("MyBook").Sheets("Sheet1").Range("A1:C12").Value = 1
```

Notice that this is a fully qualified reference. This statement will work regardless of which sheet is active. If **Sheet1** is the active sheet, the statement can be simplified as:

```
Range("A1:C12").Value = 1
```

If the range has a name, you can use the name in your statement:

```
Range("myrange").Value = 1
```

You can also refer to a range by specifying the upper left and the lower right cell. Here's an example that assigns a value to all cells in the range **A1:D12** on the active worksheet.

```
Range(Range("A1"), Range("D12")).Value = 99
```

Referring to ranges by using the Cells method

In **Excel**, the **Range** object has a method called **Cells**. Note that Cells is a *method* - not an object. When the **Cells** method is evaluated, it returns an object (*specifically, a **Range** object*).

The **Cells** method takes two arguments: the row and the column. The following statement assigns the value 1 to cell **C2** on **Sheet1**:

```
Worksheets("Sheet1").Cells(2,3).Value = 1
```

You can also use the **Cells** method to refer to a larger range. The following statement assigns the value 1 to **A1:J12** on the active worksheet:

```
Range(Cells(1,1), Cells(12,10)).Value = 1
```

In the preceding examples, the arguments for **Cells** were actual numbers. The advantage of using the **Cells** method becomes apparent when you use variables as the arguments. The subroutine below fills a 10X10 range (*rowwise*) with consecutive numbers from 1 to 100.

```
Sub FillRange()
    Num = 1
    For Row = 1 To 10
        For Col = 1 To 10
            Sheets("Sheet1").Cells(Row, Col).Value = Num
            Num = Num + 1
        Next Col
     Next Row
End Sub
```

Referring to ranges by using the Offset method

The **Offset** method is another useful way to refer to ranges. The **Offset** method returns a **Range** object, and takes two arguments. The first argument represents the number of rows to offset; the second represents the number of columns to offset.

The following statement assigns the value 1 to the cell that is one row below cell **C2** and two cells to the right of **C2** (i.e., cell E3):

```
  Range("C2").Offset(1,2).Value = 1
```

The **Offset** method is most useful when the arguments are variables, rather than numbers. The subroutine below fills a 10X10 range (*rowwise*) with consecutive numbers from 1 to 100.

```
Sub FillRange2()
      Num = 1
      For Row = 0 To 9
          For Col = 0 To 9
              Sheets("Sheet1").Range("A1").Offset(Row,Col).Value = Num
              Num = Num + 1
          Next Col
      Next Row
End Sub
```