Development of A1 and R1C1 Referencing Styles

- VisiCalc

The A1 style of referencing, with columns lettered and rows numbered, started with VisiCalc where Dan Bricklin and Bob Frankston used A1 to refer to the upper left corner cell of a spreadsheet.

- Lotus 1-2-3

The same system was adopted by Mitch Kapor who authored Lotus.

- Multiplan

Microsoft's Multiplan went against this trend and used the R1C1 system where rows and columns were both numbered.

Lotus in the 80 and early 90s dominated and A1 became the accepted standard.

Excel uses A1 as the default however R1C1 is still supported and indeed is used internally in the Macro Recorder and other formulations.

Switching to R1C1 from A1

- File → Options → Formulas →
- check R1C1 Referencing Style

Example

E7 becomes in R1C1 → R7C5

// remember the column row format switches to row column in R1C1
Toggling Reference Style Via a Macro

// code from Mr. Excel's VBA forum, offered by MVP, most valuable player, pgc01

Sub ToggleNotation( )
    If Application.ReferenceStyle = xlA1 Then
        Application.ReferenceStyle = xlR1C1
    Else
        Application.ReferenceStyle = xlA1
    End If
End Sub

Excel Formula Application Power

'On the fly', recalculation of thousand of cells is a great power of Excel as is copying a formula once, and applying it to thousands of cells.

To demonstrate find the sample used in Fig. 6.3 of the text or copy it. Double click the AutoFill handle and the formula is changed to be applied to successive values in the column.

// See text page 129 to 130. the Excel 2010 text or page 101 in Excel 2013 text

- Enter a formula such as =C4*B4 in cell D4
- Double click autofill handle
  ◦ (drag lower right corner of highlight box)
- Ctrl + ' switches between normal and formula view

Formulas in R1C1 are Constant and Apply to Whole Sets of Calculations.

Switch between A1 and R1C1 styles, noticing that the formulas change for each calculation in A1, however in R1C1 the formulas do not change.

This is one aspect of R1C1 that improves efficiency.

// Use the support files to illustrate points in the Case Study

Before we look at the case study on page 131 we go out of order and look at R1C1 Referencing Rules as explained on page 132-134 of the 2010 text or page 104 of the 2013 text.

R1C1 Relative References

The R1C1 relative referencing system works as follows.
Basic Formula

R – row #  C - col #

// where the row number follows R and the column number follows C

R1C1 Relative Referencing

<table>
<thead>
<tr>
<th>Rows</th>
<th>Examples</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>a positive number</td>
<td>R[ 1]C</td>
</tr>
<tr>
<td>-</td>
<td>a positive number</td>
<td>R[-1]C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Columns</th>
<th>Examples</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>a positive number</td>
<td>RC[1]</td>
</tr>
<tr>
<td>-</td>
<td>a positive number</td>
<td>RC[-1]</td>
</tr>
</tbody>
</table>

Using R1C1 Notation in Ranges

You cannot specify a cell in R1C1 notation as a string value as follows.

Wrong R1C1 Form Example  // doesn't work

Range("R1C1")

However, a little advertised fact is, you can use R1C1 referencing in the square brace Range form that works with cell references that do not use string form.

Range Style Using R1C1 notation Example  // does work

[R1C1, R2C2, R3C3]

Other Points Regarding R1C1 Referencing

Leaving off the bracket after R means you are referencing a cell in the same row as the active cell.

Leaving off the bracket after C means you are referring to a cell in the same column as the active cell.

RC itself is a circular reference, pointing to the active cell itself as it is pointing to the intersection of a cell in the same row and a cell in the
same column. Typically, RC as a reference would not be used.

Absolute Referencing

An absolute reference is a fixed location not relative to any other point.

A1 Absolute Referencing

In A1 style an absolute reference is achieved using the $ sign.

Example

$A$10

This is the absolute cell location, A10

R1C1 Absolute Referencing

In R1C1 absolute referencing the square braces are left out of the formulation.

Example

R10C1 // R10C1 is the same as $A$10

Mixed References

Absolute and relative referencing can be mixed. In A1 style the following has fixed columns with relative rows.

Example

$A1

// relative columns with fixed rows would look like A$1

In R1C1 style the absolute value has no square brace. The example in the text pegs the start value of a summation anchored absolutely to the second row, current column and extends the one less the current row, wherever that is, and again current column.

Example // 'VBA & Macros' Microsoft Excel 2010, Bill Jelen & Tracy Syrstad

“=SUM(R2C:R[-1]C)”
Referencing An Entire Column or Row

Rows and columns of unknown length can be referenced using short form notations. This is probably not a great thing to do unless the column is huge and all it's values need to be evaluated. A column with a hundred values will be compared to all the zeros inhabiting the remaining 1048476 rows!

Put a few values into a column and then test the MAX or MIN function to demonstrate how these notations work.

A1 Absolute Example  // probably not a best practice to adopt

$D:$D

The R1C1 Equivalent is as follows.

C4  // where C refers to Column

// Note for this to work you need to be in R1C1 Referencing mode

Using Similar Notation in a Macro

R1C1 short forms for absolute references that parallel those shown for A1 work fine using functions accessed from the spreadsheet.

For example, enter some values in the fourth column and call MAX using C4 to describe the column and the expected result appears.

Sub That Outputs Row Count For an Entire Column

The following code reveals the number of rows in the spreadsheet via the short form R1C1 notation for a column. It shows some odd behavior but is still logical as C referencing 'this row' will yield the same value as C1, C2, C3 etc.

Sub ShortCuts( )
    ' have to have spreadsheet in R1C1 mode
    Dim rg As Range
    Dim l As Long
    Set rg = [C] 'doesn't work with range set to C1, does work with C
    l = rg.Rows.Count
    Debug.Print ("Rows: " & l)
End Sub

OUTPUT  // immediate window  → Rows: 1048576
I have looked far and wide and am only aware, at the time of this writing of one form of the Range object that can take R1C1 addressing style directly and that is the square brace form of the Range.

Generally the Cells object suffices to accommodate the numerical values that parallel R1C1 style notation. If though you wanted to use relative or absolute R1C1 notation directly in a Range I think the square braces are the 'only game in town.”

Example Using R1C1 Notation in Square Brace Range Form

Sub RC( )
Dim r As Range
Dim j As Integer
j = 1001
Set r = [R1C1:R1C7]
For Each cell In r
cell.Value = j
j = j + 1001
Next cell
End Sub

The Case Study // Page 131

Mr. Excel in his Case Study on page 131 tries to expose how some of the Excel magic is implemented in A1 style macros by showing how it is implemented behind the scenes in R1C1 code.

The Code and Data are available in the supplemental download.

What is Going On in The Case Study Code

• formulas are entered in cells D4, F4 and G4
• the formulas are copied and pasted for the remaining column cells
• the values are finally totaled

A1 Style Macro // from 'VBA & Macros' Microsoft Excel 2010, Bill Jelen & Tracy Syrstad

Sub BookA1Style()
' Locate the FinalRow
FinalRow = Cells(Rows.Count, 2).End(xlUp).Row
' Enter the first formula
' Watch for a typo error in the next line in book
Range("D4").Formula = "=B4*C4"
Range("F4").Formula = "=IF(E4,ROUND(D4*$B$1,2),0)"
Range("G4").Formula = "=F4+D4"
' Copy the formulas from Row 4 down to the other cells
Range("D4").Copy Destination:=Range("D5:D" & FinalRow)
Range("F4:G4").Copy Destination:=Range("F5:G" & FinalRow)

' Enter the Total Row
Cells(FinalRow + 1, 1).Value = "Total"
Cells(FinalRow + 1, 6).Formula = "=" & SUM(G4:G & FinalRow & "")

End Sub

The Copy Method

The above code introduces the Copy Method. In the above example the destination
range is specified by a string with the start range concatenated to the Final Row
derived by the use of the End method used near the top of the macro. The Excel
Documentation supplies the following for the Copy method.

Note we see later Copy used without an argument, in which case the range is being
copied to the Clipboard.

Example // MS Excel Documentation

Worksheets("Sheet1").Range("A1:D4").Copy _
destination:=Worksheets("Sheet2").Range("E5")

Example

Sub Copier()
    Range("A1").Value = "One"
    Range("A2").Value = "Two"
    Range("A3").Value = "Three"
    Range("A1:A3").Copy Range("C1:C3")
End Sub

OUTPUT

Notice in the following RIC1 Notation that Range( “D4:D” & FinalRow) signifies
that that Final Row value will be concatenated to D4:D perhaps as in  D4:D11.
So the formula is being applied down the column from start to final row.
Sub BookR1C1Style()
    ' Page 131, middle
    ' Locate the FinalRow
    FinalRow = Cells(Rows.Count, 2).End(xlUp).Row
    ' Enter the first formula
    Range("D4:D" & FinalRow).FormulaR1C1 = "=RC[-1]*RC[-2]"
    Range("F4:F" & FinalRow).FormulaR1C1 = "=IF(RC[-1],ROUND(RC[-2]*R1C2,2),0)"
    Range("G4:G" & FinalRow).FormulaR1C1 = "=+RC[-1]+RC[-3]"
    ' Enter the Total Row
    Cells(FinalRow + 1, 1).Value = "Total"
    Cells(FinalRow + 1, 6).FormulaR1C1 = "=SUM(R4C:R[-1]C)"
End Sub

Truth Be Told!

The A1 code 'knows' R1C1 and is translating the R1C1 behavior into it's actions. Thus the following shorter A1 form demonstrates the apparent 'magic' that comes courtesy of the underlying underlying R1C1 referencing system notation.

Sub BookA1Modified()
    ' Page 131, bottom
    ' Locate the FinalRow
    FinalRow = Cells(Rows.Count, 2).End(xlUp).Row
    ' Enter the first formula
    Range("D4:D" & FinalRow).Formula = "=B4*C4"
    Range("F4:F" & FinalRow).Formula = "=IF(E4,ROUND(D4*$B$1,2),0)"
    Range("G4:G" & FinalRow).Formula = "=F4+D4"
    ' Enter the Total Row
    Cells(FinalRow + 1, 1).Value = "Total"
    Cells(FinalRow + 1, 6).Formula = "=SUM(G4:G & FinalRow & ")"
End Sub

Array Preview

Mr. Excel sneaks a new data structure, the Array into the next example. Arrays are a topic in their own which we cover later.

Building a Multiplication Table Using an R1C1 Formulation

- enter numbers 1 to 12 in range B1:M1
- enter or past the same numbers in A2:A13
Using R1C1 Formula To Create a Multiplication Table
// from 'VBA & Macros' Microsoft Excel 2010, Bill Jelen & Tracy Syrstad

Sub MultiplicationTable()

'Build a multiplication table using a single formula
Range("B1:M1").Value = Array (1,2,3,4,5,6,7,8,9,10,11,12)
Range("B1:M1").Font.Bold=True
Range("B1:M1").Copy
Range("A2:A13").PasteSpecial Transpose := True
Range ("B2:M13").FormulaR1C1 = "=RC1*R1C"
Cells.EntireColumn.AutoFit
End Sub

=RC1*R1C in plain speak says “Take this row's Column 1 and multiply it by Row 1 of this column.

Note the economy expressed in using the R1C1 Formula to fill a multiplication table through the use of a single R1C1 formula.

// Slight variation

**Multiplication Table** // adapted from Mr. Excel's code

Sub MTA()
Dim ary1 As Variant
ary1 = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)

Range("B1:M1") = ary1
Range("B1:M1").Font.Bold = True
Range("B1:M1").Interior.Color = RGB(200, 222, 170)
Range("B1:M1").Copy ' to the clipboard
Range("A2:A13").PasteSpecial Transpose:=True

Range("B2:M13").FormulaR1C1 = "=RC1*R1C"

'the following clears the clip board and exits cut/copy mode
Application.CutCopyMode = False

End Sub

**Copy Method Without A Parameter** // copying to the clipboard

Note in Mr. Excel's code the Copy method is used without a parameter. In this case the range is being copied to the 'clipboard'. The PasteSpecial function with 'Transpose' set to 'True' implies something extra has to be done to paste a horizontal range to a vertical range. (Try changing Transpose to false.)
Evident in Excel here, are orientation features, visa a vi, rows and columns that need to be handled by a special transposition process.

As a review of our last note, we can imitate what was done with the R1C1 formula using loops, shown in next table.

```
Sub MT()
    Dim j As Integer
    Dim rg1, rg2 As Range

    Set rg1 = Range("B1:M1")
    Set rg2 = Range("A2:A13")

    j = 1
    For Each cell In rg1
        cell.Value = j
        j = j + 1
    Next cell

    j = 1
    For Each cell In rg2
        cell.Value = j
        j = j + 1
    Next cell
End Sub
```

Arrays Sneak in Again Ahead Of Schedule

The next example shows how effectively arrays can be used to provide values to ranges. Consider non-consecutive values like 'resistor', 'capacitor' and 'inductor' cannot be created with loops. So arrays are an important instrument for use in VBA coding.

R1C1 Formula Values Wrap Around the Spreadsheet

The following steps shows another interesting feature of Excel' Spreadsheet model.

- Turn on Relative Referencing on the Macro Recorder
- Turn on the Recorder
- Place cursor in B2  // Mr.Excel uses F6
- Enter the formula =A1
- Press Ctrl-Enter to stay in F6
- Stop the Recorder
- View the Macro Created
Macro Created

Sub Macro2( )
    Range("B2").Select
    Selection.FormulaR1C1 = "+=R[-1]C[-1]"
End Sub

Now run the sub on A1.

The cursor moves. Place the cursor back in A1 and you see the the following.

Wrap Around Value

date = XFD1048576

// the last cell in spreadsheet, Quickly get to the end cell using Ctrl + right arrow key
// and then again Ctrl + down arrow key.

Rather Than Going Out of Range, Cell References Wrap Around To End of Sheet

Rather than yielding an error, something like 'out of bounds', in fact the value referenced is a 'wrap around value', and is the cell at the very end of the spreadsheet.

Mr. Excel is just pointing out this feature, and points out there are likely few practical applications.

Memorizing Your Letter to Number Correspondence

Mr. Excel likes using R1C1 formulas but prefers working in the regular A1 format for the spreadsheet. He has an exercise you can do to help memorize your number letter correspondence.

But, If You Don't Want to Memorize

- Move Pointer to Cell A1
- Hold Down Shift + press right arrow key // shows in upper left text field
- After the first sheet a small tool tip to the right of cell translates to R1C1

Using FormulaArray, the Array Formula, Needs CSE or Ctrl+Shift+Enter

Mr. Excel calls array formulas 'super formulas'. To use these formulas at the spreadsheet requires super measures, or a specially prepared entry form, CSE or Ctrl + Shift + Enter. As well, array formulas require R1C1 notation.
Open the last spreadsheet in the Chapter 6 resource file called 'Array Formula'.

**Screenshot of the Array Formula Spreadsheet**

<table>
<thead>
<tr>
<th>Region</th>
<th>Product</th>
<th>Date</th>
<th>Quantity</th>
<th>Unit Price</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>XYZ</td>
<td>1/1/2001</td>
<td>1000</td>
<td>22.81</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>DEF</td>
<td>1/2/2001</td>
<td>100</td>
<td>22.57</td>
<td>9.84</td>
</tr>
<tr>
<td>East</td>
<td>ABC</td>
<td>1/2/2001</td>
<td>500</td>
<td>20.49</td>
<td>8.47</td>
</tr>
<tr>
<td>Central</td>
<td>XYZ</td>
<td>1/3/2001</td>
<td>500</td>
<td>22.48</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>XYZ</td>
<td>1/4/2001</td>
<td>400</td>
<td>23.01</td>
<td>10.22</td>
</tr>
<tr>
<td>East</td>
<td>DEF</td>
<td>1/4/2001</td>
<td>800</td>
<td>23.19</td>
<td>9.84</td>
</tr>
<tr>
<td>East</td>
<td>XYZ</td>
<td>1/4/2001</td>
<td>400</td>
<td>22.88</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>ABC</td>
<td>1/5/2001</td>
<td>400</td>
<td>17.15</td>
<td>8.47</td>
</tr>
<tr>
<td>East</td>
<td>ABC</td>
<td>1/7/2001</td>
<td>400</td>
<td>21.14</td>
<td>8.47</td>
</tr>
<tr>
<td>East</td>
<td>DEF</td>
<td>1/7/2001</td>
<td>1000</td>
<td>21.73</td>
<td>9.64</td>
</tr>
<tr>
<td>West</td>
<td>XYZ</td>
<td>1/7/2001</td>
<td>600</td>
<td>23.01</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>ABC</td>
<td>1/9/2001</td>
<td>800</td>
<td>20.52</td>
<td>8.47</td>
</tr>
<tr>
<td>East</td>
<td>XYZ</td>
<td>1/9/2001</td>
<td>900</td>
<td>23.35</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>XYZ</td>
<td>1/10/2001</td>
<td>900</td>
<td>23.82</td>
<td>10.22</td>
</tr>
<tr>
<td>East</td>
<td>XYZ</td>
<td>1/10/2001</td>
<td>900</td>
<td>23.85</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>ABC</td>
<td>1/12/2001</td>
<td>300</td>
<td>20.89</td>
<td>8.47</td>
</tr>
<tr>
<td>West</td>
<td>XYZ</td>
<td>1/12/2001</td>
<td>400</td>
<td>22.86</td>
<td>10.22</td>
</tr>
<tr>
<td>Central</td>
<td>ABC</td>
<td>1/14/2001</td>
<td>100</td>
<td>17.4</td>
<td>8.47</td>
</tr>
<tr>
<td>East</td>
<td>XYZ</td>
<td>1/14/2001</td>
<td>100</td>
<td>24.01</td>
<td>10.22</td>
</tr>
</tbody>
</table>

Note extra measures taken to have the effect of pressing Ctrl+Shift+Enter are not needed in calls to array formulas in VBA macros, as the next example shows.

**Example** // from 'VBA & Macros' Microsoft Excel 2010, Bill Jelen & Tracy Syrstad

```vba
Sub EnterArrayFormulas()
    ' Page 137 in 2010 book, page 109 in 2013 text
    ' Add a formula to multiply unit price x quantity

    FinalRow = Cells(Rows.Count, 1).End(xlUp).Row
    Cells(FinalRow + 1, 5).FormulaArray = ";SUM(R2C[-1]:R[-1]C[-1]*R2C:R[-1]C)"
End Sub
```

What makes this formula 'array-like' is that two columns of data are being specified. The two sets of values are probably being stored each in its own array. Then the arrays are being iterated, doing the multiplication successively on each row.

Following is a close scrutiny of the rows and columns that are being referenced in the formula.
R1C1 Formula Described in Detail  // each pair of cells are multiplied and the results summed

"=SUM(R2C[-1]:R[-1]C[-1]*R2C:R[-1]C)"

|     |          |        |     |    |  |        |_in this column
|     |          |        |     |    |  |_ to one row up from current row
|     |          |        |     |    |_this column
|     |          |        |     |_times second row from top
|     |          |        |_one column back
|     |          |_to one row up from current row, the last row being summed
|     |_one column back
|_second row from the top in absolute R1C1 terms, a common used reference

Breaking Down the Count That Adds To The Maximum Columns in an Excel Spreadsheet

There are XFD or 16,384 columns in an Excel Spreadsheet.

The breakdown is as follows:

A    to    Z    =    26
    to    ZZ    =    676      // 26 x 26

// you can think it 26 x 26 + 26 = 702
// or as 26 x 27 with an offset so A to Z becomes 0Z to 0Z
AAA    to    XAA    =    15548

// Note this is one less than 24, BAA is 1, or 26 x 26 x 23
    to    XEZ    =    130

// again that is one less than 6 or 26 x 5
    to    XFD    =    4

_______________________
Total                                 16384

OR

26 + (26 x 26) + (26 x 26 x 23) + (26 x 5) + 4    = 16384
Assignment  // exercises using of R1C1 notation

1. Here is an Excel based word puzzle which exercises R1C1 relative referencing.

Given the following spreadsheet of values and starting at the active cell marked START what does the following spell out? (Think of an R or C as 'same row or col'.)

[R[C][1], R[-2][C][-2], R[-2][C], R[1][C][1], R[1][C][-2], R[C][1], R[-2][C][2], R[-2][C][1], R[-1][C][1], R[-2][C][-2]]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
</tr>
<tr>
<td>3</td>
<td>K</td>
<td>L</td>
<td><strong>START</strong></td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
</tr>
<tr>
<td>5</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>XYZ</td>
</tr>
</tbody>
</table>

2. Using Shift + right arrow key, which column (letter or number) does the left field stop showing the R1C1 notation for the column and instead the tool tip takes over.

3. Using Shift + right arrow key what column number is CZ equivalent to?.

4. Given a number system based on a base of 26, show the arithmetic to arrive at the value for column BF.

Example

For EC,

E = 5 , 5 x 26 = 130  
C = 3 + 3  
____________________  
133

// double checking with Shift + right arrow shows it is correct

5. a ) Using Ctrl + Right arrow, what is the A1 representation for the final column
(on the platform you are running Excel on)?

b) Use arithmetic to convert the last column A1 notation to a number.

c) Confirm your answer by changing to R1C1 reference style via Options → Formula → check R1C1 Reference Style.

d) Double confirm by putting the Address formula into first cell of the last column while in R1C1 reference style and evaluate the address.

6. Create three columns titled D1, D2 and D3. Enter 5 simple numerical values in each column and a final row holding the text 'Total' as shown in the example below.

Example

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>D3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total</td>
</tr>
</tbody>
</table>

Now, using the Array Formula Sub shown on the bottom of page 137, modify the R1C1 notation in the SUM formula to multiply the three values in each row and then Sum results, putting the value in the D3 column after the word 'Total'.