

Chapter 1

A Foundation for Developing Best Practices

IN THIS CHAPTER

- ◆ Understanding alternative ways to represent cells (R1C1 notation style compared to the traditional A1 style)
- ◆ Alternative approaches to computing numbers
- ◆ Getting a firm grounding on absolute, relative and hybrid cell references and when to use each
- ◆ Specifying cell ranges and incorporating those within functions such as SUM
- ◆ Understanding rapid and efficient navigation techniques in large spreadsheets
- ◆ Defining user names (in rapid fire succession)

THIS CHAPTER IS INTENDED TO ARM you with the foundation for best practices including basic components and spreadsheet functions you need to know. The chapter begins with a discussion of important concepts and spreadsheet functions that you will be working with regularly and presents the best way to use these. Included are the gotchas, the tips, cautionary tales, and revelations that uncover new avenues for developing your spreadsheets.

Some important and useful Excel functions are introduced, but focus is on their usage rather than giving a formal treatment.

Preliminaries

This is a book about best practices. In my mind, the notion of *best practices* conjures up an image of working smartly and efficiently when using spreadsheets. To be sure, there is no shortage of “esoterica” or sophistication in this book, but there is plenty of time for that later. Sophisticated techniques that are effective and useful are not generally built upon more esoteric concepts. Rather, they take hold most easily when built on solid foundations.

Moreover, a surprising amount of sophistication comes from simple and basic stuff. This will become evident as you explore the later chapters. If you're curious, thumb ahead and glance at the Chapter 7, "Creating and Using Smart Data," to see what I mean. You'll better appreciate the things you can do, though, if you give me a chance to mold the way you think about spreadsheets and their use. I therefore focus on simple and basic spreadsheet concepts in the early chapters.

Many, if not all of the topics covered in the opening portions of this book may already be well known to you. My purpose for presenting these "foundation" topics is to get you to characterize and think about things a certain way. So, if you're already familiar with the topics, this should be an easy and quick read.

Indulge me. Read through the early chapters even if you know spreadsheets backwards and forwards. Getting into the deeper and more involved discussions, later in the book, will occur more naturally if you do. I'm also willing to bet you'll acquire a thing or two in the early chapters that you'll be happy to carry in your hip pocket.

Enough of the pep talk. Let's get on with Excel best practices.

Working with Different Ways To Compute a Number in Excel

Often, what people are taught about spreadsheets is the lowest common denominator, just enough to squeak by. These skills can be readily acquired through corporate and continuing education courses. Most people who regularly work with spreadsheets tend to adopt, learn and share common practices—which are not always synonymous with best practices. My goal in writing this book is to get you to elevate the bar, to go beyond the common denominator so that your ability to manipulate spreadsheets is on par with your natural abilities. Spreadsheets are intimidating, so many of us are afraid to go beyond what others have taught us. I would like to reconnect your good business instincts with your ability to handle spreadsheets, and I have no doubt that you'll be able to achieve and exceed your expectations.

This chapter introduces you to the various ways you can put to use in your formulas absolute, relative and hybrid expressions. You will also be shown the practical benefits of using one approach over another.

Also, you will find out that the built-in Excel functions are not only about performing arithmetic and mathematical calculations. You will see that Excel introduces specific functions of a spatial nature. Other Excel functions have a temporal nature to them. As I go through this list, it should not surprise you that some Excel functions perform computations specifying precision and fuzziness (and this has nothing to do with "Fuzzy Math"). Did you think I was going to end there? Would it interest you to know that there are Excel functions that can edit text? I'm not talking about functionality, as in features of the Excel product; I am specifically talking about Excel functions that utilize these features in their computations.

The point is, more computational facilities are at your disposal than just straightforward arithmetic. You should be thinking about Excel functions in this purposeful and tantalizing manner, rather than as a boring laundry list of computing functions.

Understanding Alternative Ways to Represent Cells

Spreadsheets organize information into rows and columns like the one contained on your CD-ROM (open the sample spreadsheet, `ch01-01.xls`).

The intersection of a row and column is referred to as a spreadsheet *cell*. In Figure 1-1, a cell has been selected on row 13 and column 2. A shorthand notation for designating this cell position is `R13C2`. Excel understands this notation, so you can use it directly in your formulas. Excel also understands a different way of expressing formulas, which involves having columns that use letters instead of numbers. This section explains that approach and discusses the implications and benefits of both types of notation.

Excel provides two basic ways to display formulas appearing in spreadsheet cells. One of them is referred to as an “A1” style and the other as “R1C1.” In much the same way as a fashion statement, these styles affect only the outer appearance of a spreadsheet. The underlying content remains unaffected no matter how often you switch back and forth between the two modes (yes, you can switch back and forth). What follows is information about these two approaches, what they offer, and how they differ.



Appendix A discusses Excel Options settings that you can adjust, including telling Excel whether you want your fashion garb to be R1C1 or A1. Actually, the CD-ROM with this book contains a spreadsheet tool that will make switching back and forth as often as you like very easy.

The traditional approach: The A1 style

Most people are already familiar with the A1 style of representing spreadsheet cells. In this scheme, columns appear as letters and rows as numbers. Out of the box, Excel is loaded with the default setting switched to A1.

There are some consequences associated with using columns as letters and rows as numbers. When referring to a relative reference (say, two columns over to the right and two rows down), you must specify fixed absolute position in space (the actual column letter and row number). If you copy a cell and paste it to other locations on the spreadsheet, the formula, as it is written, changes. The exact formula is dependent upon the cell you paste to. You’re giving yourself the burden of translating column letters and row numbers in your head.

6 Part I: Best Practice Essentials

	1	2	3	4	5	6
1	[FILL-IN YOUR NAME]	COMPANY				
2	Web Page Statistics					
3	For the prior 7 days					
4						
5		Page Views	Visits			
6	Day 1	55223	17641			
7	Day 2	40694	14156			
8	Day 3	24946	8281			
9	Day 4	52701	16631			
10	Day 5	59495	18902			
11	Day 6	57235	18174			
12	Day 7	55416	18546			
13						
14						
15						

Figure 1-1: A sample spreadsheet

Suppose you see two formulas. The one in cell DC91 is

=CR98+CX66

There's another cell, EG62, which contains the formula:

=DV69+EA36

Now for the pop quiz. Giving these just a quick visual inspection, do you see them as equivalent formulas (that is, could you have done a simple copy-and-paste to get the second formula from the first)? I am willing to bet that most spreadsheet users wouldn't want to be caught thinking about such a question.

If you do give it a little thought, you will see that in the first formula, CR98 is 7 rows below DC91 as well as 11 columns to the left of DC91. In the second expression of the formula, CX66 is 25 rows above DC91 and also 5 columns to the left of it.

In the second formula, DV69 is 7 rows below EG62, as well as 11 columns to the left of EG62. So, the first expression lines up. The second expression, EA36, is 26 rows above EG62 as well as 6 columns to the left of EG62.

Whew! That was a bit of mind twisting. Not only did I have to think about this twice, I had the editors breaking their heads when proofing the cell references by hand.

Oh, after all that, you do know whether the formulas were equivalent, don't you?

If you instead used row and column number referencing (that is, the R1C1 style in the Excel Options), you find would it immediately evident that these formulas are not the same. In terms of row and column references, you would see that the first formula is

=R[7]C[-11]+R[-25]C[-5]

and the second formula is

=R[7]C[-11]+R[-26]C[-6]

Although the first expression (`R[7]C[-11]`) matches perfectly, the second expression doesn't. Just looking at the formulas is all you need to do. There is no mental translation.

In this example, using the R1C1 style, it was easy to see that the second expression in the formulas doesn't line up in terms of the number of row offsets or column offsets.

Does the R1C1 approach scale well?

What if the formulas were more complex, having more than two expressions as well as incorporating mathematical functions going well beyond the simple arithmetic of addition and subtraction? The answer is that you would still be fine using the R1C1 style of cell referencing.

Understanding how these two approaches differ

So, why is the alphabetic notation for columns potentially unwieldy when compared to the numeric referencing of columns? There are several reasons:

- ◆ Imagine giving someone driving directions using east/west directions in miles and north/south directions in kilometers. Do you think it's a good way to give directions? Why use numbers for rows and letters for columns? Complicate this by the fact that rows are labeled in Base 10 notation and columns in Base 26 (twenty-six letters in the alphabet). This situation forces you to use two different numbering or labeling systems.
- ◆ Try taking any of your typical spreadsheets and swapping the rows for columns and columns for rows. Compare the formulas between the two spreadsheets. It may not be immediately evident which formulas correspond to which. If Excel didn't provide a "Transpose" capability when copying and pasting cells, you could quickly get lost in comparing and rewriting formulas.
- ◆ In the A1 style of cell referencing, relative references in formulas must specify the *actual* position of a cell relative to the current cell being computed. Does that sound convoluted enough? Other than the fact that you're just used to it; why would you want to use an absolute cell position for a relative reference? Doing so only introduces an artificial artifact.

Take it one step further. When you copy and paste a cell involving relative references, each new formula, although similar in structure, is dependent upon the location it is copied to whenever relative cell references are involved in the formula. Wouldn't it seem preferable and logical to have the replicated formulas remain the same as the original formula they were copied from?

What do you give up by using the Row and Column notation?

Making the switch to using row and column numbers does come with a price, albeit a small one. For example, the R1C1 style forces you to put an *R* in front of the row number and a *C* in front of the column number. There are some other mental adjustments I want you to think about:

- ◆ In the “matrix style” of rows and columns, cells are written with the row followed by the column. The A1 style, by comparison, displays the column first as a letter. If you’re already used to the A1 style, there’s a definite readjustment in thinking “row, column” rather than “column letter, row.”
- ◆ There is another adjustment, though I wouldn’t call it giving up something, but rather an even trade. Absolute references in the A1 style are designated by a \$ symbol—for example, \$B\$23. In R1C1 parlance, this refers to row 23, column 2, hence R23C2. As the example shows, in the R1C1 style you drop the \$ notation, because you don’t need it for absolute references. When a relative reference is involved, you place brackets around an “offset” number. As an example, if you want to specify the cell to your immediate right, you use RC[1]. It doesn’t matter which cell you are in. If you copy and paste cells that use RC[1], the expression RC[1] remains unaltered in the pasted cell regardless of where you paste it.

This way of referencing forces you to start thinking about the position of spreadsheet cells visually. For example, think of R[3]C[1] as one cell over to the right, three cells below.

- ◆ Excel allows you to compute the sum of a whole row or column. Using the A1 style, if you want to take the sum of all cells in row 5, you write SUM(\$5:\$5) in absolute coordinates or SUM(5:5) in relative coordinates for the cell you happen to be in. If you were to switch this to the R1C1 style, you would write SUM(R5) in absolute coordinates and SUM(R[2]) in relative coordinates if you happen to be anywhere in the third row. Understandably, the usage of the R1C1 style for representing a whole row using absolute coordinates can be confusing if you are used to the A1 notation, because R5 looks like an A1 notation. Make no mistake about it. R5 in the R1C1 style refers to all of row 5. How hard can that be to get used to?

In summary, you’re not really giving up anything. You’re just making some mental adjustments to think about things a little differently.

What do you gain by using the Row and Column notation style?

Here’s what you gain by switching to the Row and Column notation:

- ◆ It buys you mental brevity. When you copy and paste formulas, the pasted formulas remain unchanged regardless of where you paste them. This eases the level of spreadsheet complexity. The one thing you don't want to have to be doing is thinking about many different versions of the same formula. You have to think about only one formula; there's no mental translation involved.

You can quickly spot altered formulas, however slight the variations may be. Just use your arrow keys on the keyboard to quickly pass over cells that should be identical. If one of them is different, the formula will not appear the same as the others.

- ◆ You begin to think about the relationship between cells visually. After all, one of the major appeals of using a spreadsheet is to visually spread out your numbers onto rows and columns, in much the same way as you would think about laying out cards and arranging them on a table.
- ◆ You think more clearly about your formulas. Imagine that you are preparing a report that summarizes projected financial data for a ten-year period. Different groups of data exist for each of the ten years. As an example, open the file ch01-02formulacompare.xls (or just look at Figure 1-2). If the categories are arranged in groups of ten columns (one column for each of ten years), your formula could appear as:

=L7+V7+AF7+AP7+A27

when using the A1 Style. If instead, you choose to use the R1C1 Style (if you have the spreadsheet open, try clicking the R1C1 Style button), the formula becomes:

=RC[10]+RC[20]+RC[30]+RC[40]+RC[50]

The screenshot shows a spreadsheet with the following data:

	2004	2005	2006	2007	2008	2009	2010	2011	2012
Product A	24	29	34	39	44	49	54	59	64
Product B	23	28	33	38	43	48	53	58	63
Product C	18	23	28	33	38	43	48	53	58
Product D	15	20	25	30	35	40	45	50	55
Product E	13	18	23	28	33	38	43	48	53
	5	10	15	20	25	30	35	40	45

Below the table, there are two buttons: "R1C1 Style" and "A1 Style". The "A1 Style" button is selected.

A1 Style in use

The Aggregate Sales is the total for the sales for each of the regions.

Each region is located side by side to the immediate right of aggregate sales.

The formula in B7 appears as: **'=L7+V7+AF7+AP7+A27'**

If you were to track an additional region, what would the next expression in your formula be?

Try asking yourself this question using the R1C1 Style (press the R1C1 Style button).

Figure 1-2: Formulas using the "A1 style"

10 Part I: Best Practice Essentials

As you can see, each successive term in the formula points to a cell 10 columns further to the right (Figure 1-3). This formula is a lot easier to understand and visualize than is the A1 style of the same formula.

This is the formula for R7C2. What about the formula for row 7, column 3? The formula is

```
=RC[10]+RC[20]+RC[30]+RC[40]+RC[50]
```

Notice any similarity? Why not check the formula in the cell immediately below R7C3? The formula is

```
=RC[10]+RC[20]+RC[30]+RC[40]+RC[50]
```

In fact, all the formulas that compute the aggregate sales for any product during any given year are identical. They are also easy to read and understand.

1	YOUR COMPANY NAME								
2	Sales Summary and Details								
3	for the 10 year period starting from 2004								
4									
5									
6									
7	Product A	24	29	34	39	44	49	54	59
8	Product B	23	28	33	38	43	48	53	58
9	Product B	18	23	28	33	38	43	48	53
10	Product C	15	20	25	30	35	40	45	50
11	Product D	13	18	23	28	33	38	43	48
12	Product E	5	10	15	20	25	30	35	40
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									

Figure 1-3: Formulas using the "R1C1 style"

Try pressing the A1 style button. Inspect the formulas in the different cells. Do the formulas appear identical? Is it clear that you're shifting over in columns of 10 with each additional expression in the formula? Would you know what to type in for the next expression (that is, what column letter to use) if you were to add yet another category?

If you never used spreadsheets before in your life, and you looked at this example, which approach do you think would make more sense and enable you to better manage increasingly complex spreadsheets? What would the kid who just graduated from college, and is not wedded to any particular style, think? If he or she would pick the R1C1 style, would that gained simplicity give that person a competitive edge?

The correct answer in any scenario is that you decide what's best for you. All I can do is lay the cards on the table and present options and ways to think about these things.

Is there any happy medium between these choices?

Yes, there is a happy medium. There is no reason for you to deny yourself the availability of both approaches. You have seen that you can switch between one mode and the other at a press of a button (see Figures 1-2 and 1-3).



I've provided a SwitchTool spreadsheet (ch00-02switchtool.xls on the CD-ROM) for your use. You can switch back and forth at the click of a button.

You should be aware that Excel really doesn't care which way you represent cells. The underlying representation of cells has nothing to do with the way you represent and label the data. So, you can switch the mode or context from the R1C1 to A1 (and vice versa) as often as you like, and save your spreadsheet files in either mode.

Even better: Using names instead of coordinates for cell referencing

To make matters better, there is yet a cleaner and more elegant approach to complex spreadsheets that definitely promotes best practices. The approach is to define descriptive names and use the names instead of cell references directly in your formula. To see for yourself, try this: Which of the following three formulas would you prefer to use to compute current assets?

```
=B$24+B$35+B$40+B$45+B$50
```

```
=R24C-R35C+R40C+R45C+R50C
```

```
=Cash+MktbleSecurities+Receivables+Inventory+PrepaidExpenses
```

Although an extra step is required in terms of defining named ranges, the payoff is definitely there. To begin with, your formulas incorporate meaningful words rather than use cryptic cell references. The formula not only is readable to you and third parties but also is equally well understood by the computer. Getting everyone (the computer included) to be using the same referencing scheme is a definite plus.

As if this is not enough justification, you have the added benefit of getting the computer to validate that you are using properly defined expressions. If you accidentally misspell a word such as *Inventory* by spelling it, for example, *Inventorx*,

12 Part I: Best Practice Essentials

Excel will evaluate the cell to be `#Name?`. You'll know right away that it's wrong. This wouldn't happen if you had used a cell reference such as `R39$C` instead of `R40$C` or `R39C` instead of `R40C`.

The use of named references does not involve any context mode switching, and no Excel setting is required. It's just there for you to use.

Using named references is definitely the way to go!

Computing a Number in Excel

Spreadsheet cells have a variety of properties. In terms of plain-vanilla spreadsheet cells, one of three basic options is allowed for any spreadsheet cell:

- ◆ **Option 1:** The cell can be blank.
- ◆ **Option 2:** The cell can contain a fixed value. This fixed value can be a number, a date, or some text label. If this value is fixed, the cell has no formula. Also, the value of the cell will be visually displayed on the spreadsheet.
- ◆ **Option 3:** A cell can possess a formula that is used to dynamically compute a value that will be displayed in the cell. Formulas always start with an equal (=) symbol.

Arbitrarily pick any cell, say row 14, column 1. If this cell holds the formula

`=2+3`

it maintains an internal formula but displays the value 5. There is a fundamental distinction between fixed values (for example, 5) and equivalent formulas (such as `=2+3` or `=5`). In a formula, Excel will try to compute formulas and then display the value rather than just simply retrieve the value. You will see that formulas can and do get more complicated.

Excel formulas can use mathematical functions. For instance, you can obtain the square root of 200 using the following formula:

`=SQRT(200)`

The value that Excel displays will be 14.14213562. Certainly, you can combine mathematical expressions to build more complicated formulas. As an example, you can add 10 to the square root of 200 with the following formula:

`=SQRT(200)+10`

The value displayed will be 24.14213562. Clearly, this is neither a very complicated nor interesting formula. Probably, some of you are thinking “I’m not interested in square roots. That’s what mathematicians use. I don’t have the time to get involved with stuff like that.” Basically, you’re right. Throughout most of this book you will hardly touch upon square roots and other similar mathematical constructs except where it’s really appropriate. When you do, you’ll be supplied with all the information you need.

In addition to conventional mathematical expressions, Excel formulas can incorporate values already computed in other spreadsheet cells. Here, things start to get interesting. The real goal is to get you to understand the options available to you when entering Excel formulas and the subtleties involved with cell references.

Cell references in Excel are generally not well understood. This is why people are easily stymied when it comes to understanding and using formulas in Excel. It’s worth the time to really nail down this topic. It will pay off in handsome dividends.

As you build formulas, you can refer to the contents of other spreadsheet cells. You can do so by identifying which cell you want based on its *specific* row and column. This is known as an *absolute* cell reference.

You can also do so by identifying, in terms of the number of rows and columns, *how far away* the cell is. This is known as a *Relative* cell reference.

You can also make use of a *hybrid* cell reference, in which the row is fixed but the column offset is relative (or vice versa).

For an illustration of the contrast among these three types, look at Figure 1-4.

1	2	3	4
Description	actual computation	formula used in computation	Comments
ten	10	10	plain value
square root of 200	14.14213562	=SQRT(200)	simple formula
different ways to compute	24.14213562	=R5C2+R4C2	Absolute rows, columns
ten + square root of 200	24.14213562	=R[-2]C+R[-3]C	Relative rows, columns
	24.14213562	=R5C+R4C	Abs. rows, Rel. columns
	24.14213562	=R[-4]C2+R[-5]C2	Rel. rows, Abs. columns

Figure 1-4: Absolute vs. relative vs. hybrid cell reference usage in Excel formulas

Rows six through nine of column 2 show alternative ways to compute the value of ten plus the square root of 200. They all produce the identical results, but each one is computed in a slightly different manner. Column 3 displays the formulas used for column 2 so that you can see what they look like; column 4 provides some explanation.

The value of 10 is occupied in the cell R4C2. It is a plain value.

In R5C2, the square root of 200 is computed using this formula:

```
=SQRT(200)
```

The cell displays the value that results from the computation.

14 Part I: Best Practice Essentials

The number in row 6 is computed using the sum of pure absolute cell references R5C2 and R4C2:

```
=R5C2+R4C2
```

Row 7 contains a slightly different kind of formula, which employs “relative” references. The formula used is

```
=R[-2]C+R[-3]C
```

Basically, the formula in row 7 column 2 is saying: “Give me the value of the cell two rows directly above me and in the same column (this happens to be 14.14213562) and add to this the value of the cell three rows directly above me and in the same column as I am (which happens to be the value 10).”

The bracketed numbers immediately after the row reference indicates a “row offset.” So, R[-1]C would refer to the neighboring cell to the immediate north. By comparison, R[1]C refers to the neighboring cell directly below or to the south. By the same token, R[2]C refers to the cell two rows directly below (to the south).

The relative reference in row 7 column 2 could explicitly state the “column offset” of zero in addition to the row offsets. This would make the formula more verbose than it needs to be. In this case, the formula would appear as

```
=R[-2]C[0]+R[-3]C[0]
```

The column offset happens to be zero, since the cells being referred to by the formulas are in the same column as the formula for R7C2. Excel is happy to drop off the bracketed zeros, because doing so will make formulas easier to read. Had the formula been referring to a cell to the immediate left (west), the column offset would be a negative number. Had the formula been referring to a cell to the immediate right (east), the column offset would be a positive number.

Thus, the way to refer to a cell directly above and to the immediate right (that is, northeast) would be referenced by R[-1]C[1]. Had there been a greater separation between the cells, the offsets of rows and columns would have had to be increased accordingly.

This way of thinking about cell references takes a little getting used to. When you get comfortable using offsets, you’ll find yourself thinking about formulas visually rather than in terms of abstract formulas.

At this point you might be asking, “Can I mix relative and absolute cell references?” The answer is a resounding yes. When a formula is composed of a compound formula involving multiple references, it uses the cell value and doesn’t care how the cell was referenced.

As an example, take a look at the formula in row 8.

```
=R5C+R4C
```

Here you have such a mixture. The rows are listed in absolute references (notice that the numbers 5 and 4 do not have brackets around them, so they must refer to rows 5 and 4, respectively). The columns are relative references. In relative references, you don't have to show the brackets if your offsets are zero. Had you decided to provide the column offsets using bracketed expressions, you would have:

```
=R5C[0]+R4C[0]
```

Another possible formula appears in row 9. In this case, the row references are kept relative and the columns are absolute. It takes the form:

```
=R[-4]C2+R[-5]C2
```

Most of the time, you will be working with pure relative references. There are a couple of reasons for this.

1. Excel will let you build spreadsheet formulas by “clicking” the cells you want to reference. As you click cells, Excel inserts them into the edit line of your formula as relative cell references. So unless you want to change these references, the formulas you build will tend to be based on relative references.
2. Relative cell references are mostly a good thing because they make replicating formulas to other portions of your spreadsheet easy.



When entering and editing an Excel formula, you can switch selected portions of your formula from relative to absolute to hybrid by pressing the <F4> key.

Best Practice Topic: Evolving a strategy toward Absolute vs. Relative vs. Hybrid cell references

Rather than just state a pre-established set of guidelines regarding the use of alternative types of cell references, I want to present some scenarios and have you formulate your own judgments concerning the merits of the various choices among absolute, relative and hybrid cell references.

Consider the following: You have a variety of items. For argument's sake, let's say there are ten such items and you may be purchasing these items. Although each of these has a listed cost, you are entitled to a 20% discount. The items are arranged (see Figure 1-5) in a vertical column.

16 Part I: Best Practice Essentials

	1	2	3	4	5
1	[FILL-IN YOUR NAME] COMPANY				
2	Combined Relative and Absolute References				
3	In the computation of discounted amounts				
4					
5			discount		
6	Item No.	Cost	20%		
7	1	100			
8	2	125			
9	3	300			
10	4	235			
11	5	195			
12	6	540			
13	7	1325			
14	8	1200			
15	9	900			
16	10	760			
17					

Figure 1-5: Preparing a schedule to compute discounted amounts

A natural step in the cost analysis is to compute the discount amount for each of the ten items. In the cells to the immediate right of each of the ten items, you can write a formula in column 3 that looks like:

`=RC[-1]*0.2`

or

`=RC[-1]*20%`

or

`RC[-1]*20/100`

They'll all work. However, they are hard-wired to 20%. This may be correct for your discount percentage today. Next week you might need to adjust your discount percentage. That would mean that you would have to rework your formulas every time the discount changes. You don't want to have to change this by hand, for two good reasons:

1. In the first place, it is time consuming to hunt everywhere to change the 20% discount factor to something else.
2. Even if you have the time to make manual changes, chances are that as the complexity and size of your spreadsheet increases, the likelihood of missing a number you need to change, or making a mistake when hand editing, increases.

So a better way to approach this is to reference the item cost using a relative reference, and reference the discount rate using an absolute cell reference. The discount rate could be parked into a place like R6C3 (see Figure 1-6).



Where possible, try to use global references. Single global numbers in isolated cells are easily identified and maintained. It will serve your purpose even better if you assign a name to use in place of cell coordinates.

Then you could use a formula like:

Formula 1-1: First Attempt at Computing Discounted Amounts

=RC[-1]*R6C3

	1	2	3	4	5
1	[FILL-IN YOUR NAME] COMPANY				
2	Combined Relative and Absolute References				
3	In the computation of discounted amounts				
4					
5			discount		
6	Item No.	Cost	20%		
7	1	100	20 =RC[-1]*R6C3		
8	2	125	25 =RC[-1]*R6C3		
9	3	300	60 =RC[-1]*R6C3		
10	4	235	47 =RC[-1]*R6C3		
11	5	195	39 =RC[-1]*R6C3		
12	6	540	108 =RC[-1]*R6C3		
13	7	1325	265 =RC[-1]*R6C3		
14	8	1200	240 =RC[-1]*R6C3		
15	9	900	180 =RC[-1]*R6C3		
16	10	760	152 =RC[-1]*R6C3		
17					

Figure 1-6: Combined relative and absolute references

The formula for computing discounted amounts is more easily maintainable. The discount percentage is now:

- ◆ A single global number. (If you were to update/alter the number, every reference to the discount percentage would be immediately updated.)
- ◆ The number is visible and accessible for easy updating.

This way has a decisive advantage. When the number has changed, you know it has changed and you know what the new value is.

So far, so good. It's time to throw in a little more complexity. The likelihood is that you might need to evaluate the financial impact of more than one discount rate. You have two strategies available:

1. Compute your values using the first rate (and if necessary, print it). Compute your values again with the alternative rate. After all, the formulas are set up so that the discount rate only appears in one (globally referenced) cell. Changing it in this one location is easy. Spreadsheets are great for this purpose. You can interactively try different values and see what their impact would be. This is often referred to as a "what if?" scenario.

18 Part I: Best Practice Essentials

What I describe is a kind of manual approach to a what-if scenario. Excel sports a special-purpose facility called a Scenario Manager. To get an idea of what you can do with Scenario Manager, look at Figure 1-7.

Scenario Summary				
Current Values:		Scenario 21	Scenario 22	Scenario 23
When you create a scenario, Excel would insert your comments here. Created by Loren Abdulazer on 4/8/2003. don't specify something.				
Changing Cells:				
R6C3	20%	21%	22%	23%
Result Cells:				
R7C3	20	21	22	23
R8C3	25	26.25	27.5	28.75
R9C3	60	63	66	69
R10C3	47	49.35	51.7	54.05
R11C3	39	40.95	42.9	44.65
R12C3	108	113.4	118.8	124.2
R13C3	265	278.25	291.5	304.75
R14C3	240	252	264	276
R15C3	180	189	198	207
R16C3	152	159.6	167.2	174.8

Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

Figure 1-7: Scenario Summary produced by the Excel Scenario Manager

There are too many caveats on the usage of this facility. My chief goal right now is to keep you focused on the use of absolute, relative, and hybrid cell references in Excel formulas.

- 2., The Scenario Summary, shown in Figure 1-7, is actually food for thought. Instead of trying to experiment with different values for the discount percentage one at a time (which is what you would do when you generate each scenario anyway), think about how you would write the formulas to compute discount amounts for multiple discount rates placed side by side.

To make this accommodation for the second strategy, you need to make some alterations to Formula 1-1. Think of doing a side-by-side comparison of discounted amounts based on differing percentages. The one that already incorporates the 20% baseline is a series of numbers running down column 3 from rows 7 through 16. Imagine placing an alternate discount percentage on row 6 column 4 and having the computed numbers running down column 4 from rows 7 through 16.

These sets of computations share two features:

1. The discounted amount resides on the same row as the item to be discounted. Specifically, the item to be discounted resides in column 2. You'll want to think of the item to be discount in terms of its absolute column reference.

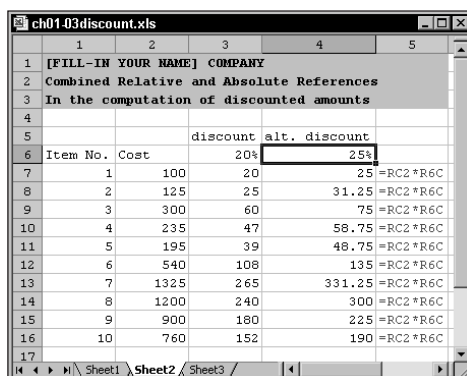
2. The discounted amount resides on the same column as the discount percentage. Specifically, the discount percentage in this layout (see Figure 1-8) is always placed on row 6; this is an absolute.

The resulting formula to compute the discounted amount should be

Formula 1-2: Hybrid Formula

=RC2*R6C

Figure 1-8 shows how the spreadsheet with the revised formulas should appear. Note that the formulas (appearing in column 4) used for computing the discounted amount based on the alternate discount percentage are completely identical to the formulas (appearing in column 3) based on the original discount percentage. Additionally, if you want to do a side-by-side comparison with three or more discount rates, you need only to keep putting your discount rates side by side across row 6. Then replicate your formula (=RC2*R6C) across all your discounted amounts.



	1	2	3	4	5
1	[FILL-IN YOUR NAME] COMPANY				
2	Combined Relative and Absolute References				
3	In the computation of discounted amounts				
4					
5			discount	alt. discount	
6	Item No.	Cost	20%	25%	
7	1	100	20	25 =RC2*R6C	
8	2	125	25	31.25 =RC2*R6C	
9	3	300	60	75 =RC2*R6C	
10	4	235	47	58.75 =RC2*R6C	
11	5	195	39	48.75 =RC2*R6C	
12	6	540	108	135 =RC2*R6C	
13	7	1325	265	331.25 =RC2*R6C	
14	8	1200	240	300 =RC2*R6C	
15	9	900	180	225 =RC2*R6C	
16	10	760	152	190 =RC2*R6C	
17					

Figure 1-8: Correct use of replicating hybrid formulas

Clearly, if the spreadsheet you build is arranged differently than the one in this chapter, the formulas will be different. However, the logic remains identical.

More useful information about working with formulas

Shortly I discuss the functions SUM, SUMPRODUCT, and others. First, I explain some details on entering formulas, changing cell references, and designating any range of cell names of your choosing so that formulas can be easier to write, read, and validate.

20 Part I: Best Practice Essentials

ENTERING FORMULAS

You may have noticed that the mathematical functions such as `SQRT` and cell references have been appearing in uppercase format. This is no accident. Excel likes to convert and store its worksheet functions and cell references that way. What happens if you enter them in lowercase? Go ahead and try it. Enter something like the following set of formulas, each in separate rows, one below the other:

```
=now()  
=row()  
=row(r[1]c)  
=sqrt(18)  
=power(2,3)  
=rand()  
=sum(r[-5]c:r[-1]c)  
=max(r[-6]c:r[-2]c)  
=average(r[-7]c:r[-3]c)  
=average(10,20,30,70,80,90)
```

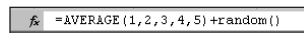
After you've entered these expressions, go back and click on each of these cells to see whether these spreadsheet functions and cell references appear in uppercase format.

Okay, Excel doesn't really care how you enter formulas; it converts them to uppercase regardless. What's the big deal? You can use this to your advantage. Consider typing in the formula:

```
=average(1,2,3,4,5)+random()
```

The result you get is `#NAME?`. Obviously, something is wrong with this formula; otherwise you would get a computed number, not a quizzical error message. You have two functions in your formula: `average` and `random`. Which of these two is the offending function? Maybe it's both.

If you select the cell where you just entered this formula, you will see the text shown in the Excel Formula Bar in Figure 1-9.



`=AVERAGE(1,2,3,4,5)+random()`

Figure 1-9: The function with the error is not automatically converted to uppercase.

Notice that Excel converted the `AVERAGE` function to uppercase. However, it didn't do anything with "random." This is because Excel does not have a "random" function. Had you written the function as `RANDOM`, Excel would have kept it that way, in which case you'd be staring at two expressions in uppercase form, not knowing which one is the offender.



Enter the Excel functions in your formula in lowercase form. If the functions are understood by Excel, they will be converted to uppercase. Those functions not understood by Excel will remain in lowercase. This practice enables you to spot naming errors in your formulas when they occur.

If you want to go ahead and fix the formula, you can edit the formula to

```
=AVERAGE(1,2,3,4,5)+rand()
```

ENTERING AND EDITING FORMULAS

Perhaps it's time to begin discussing easier ways to enter formulas. Open the sample spreadsheet, ch01-01.xls (Figure 1-10).

	1	2	3	4	5	6
1	[FILL-IN YOUR NAME]	COMPANY				
2	Web Page Statistics					
3	For the prior 7 days					
4						
5		Page Views	Visits			
6	Day 1	55223	17641			
7	Day 2	40694	14156			
8	Day 3	24946	8281			
9	Day 4	52701	16631			
10	Day 5	59495	18902			
11	Day 6	57235	18174			
12	Day 7	55416	18546			
13						
14						
15						

Figure 1-10: Sample spreadsheet with Web log data

If you want, you can enter a formula into R13C2 such as the total Page Views over the seven-day interval. You could enter the following (see Figure 1-11):



The following formula is included in your Take-Away text file takeaway.txt:

```
=r6c2+r7c2+r8c2+r9c2+r10c2+r11c2+r12c2
```

This formula produces a count of 345,710 for Page Views over the seven-day period.

22 Part I: Best Practice Essentials

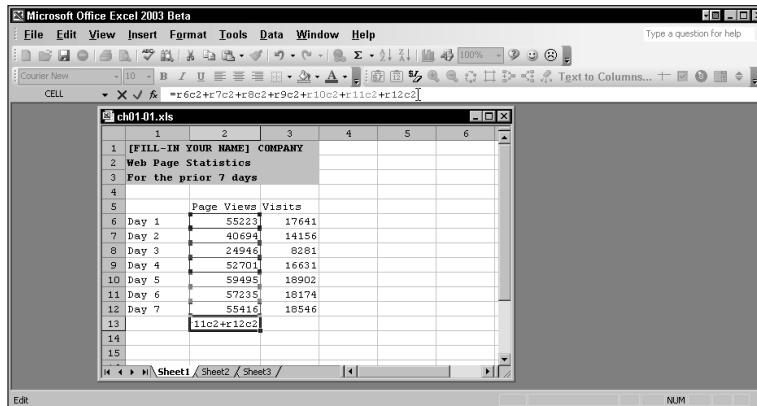


Figure 1-11: Cells in the formula line are visually highlighted.

You should notice that as you type in the formulas, the cells or ranges of cells you specified in your formula become highlighted in colored borders and the colors are matched with the cell references as you are editing or typing in the formulas. This is a useful aid for two reasons:

- ◆ It can give you easy visual feedback of what you've just typed in.
- ◆ If you have a typographic error in your formula, strange things will happen to the highlighting (Figure 1-12); or what you expect to be highlighted will not be highlighted at all.

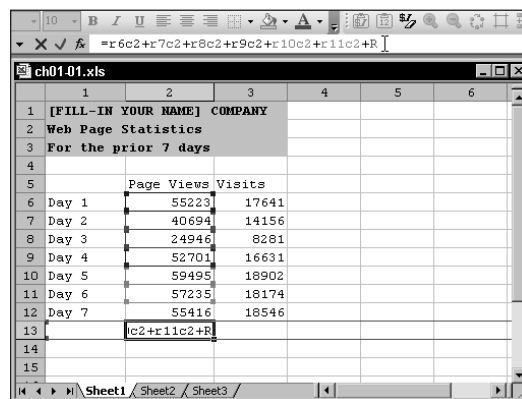


Figure 1-12: A formula entered with an error is highlighted differently.

This technique of clicking in the edit line and seeing the corresponding highlighted cells within the worksheet will prove helpful when you prepare and analyze more complex spreadsheets.



The technique of visually highlighting cell references in a formula (as shown in Figures 1-11 and 1-12) need not be applied only to new formulas you are entering. You can also click the Excel Formula Bar for any cell. This is especially useful when you receive a spreadsheet from a third party and want to quickly inspect where it is gathering its numbers.

While you're at it, you're likely to have the occasion to write some nested formulas. As you type away in the edit line, when you enter a closing parenthesis, Excel momentarily highlights the corresponding opening one (see Figure 1-13). This feature helps you both to see what your expressions contain as well as to balance the parentheses (that is, ensure that the number of opening parentheses matches the number of closing ones).



Figure 1-13: As you enter the closing parenthesis on the right, the corresponding opening parenthesis is momentarily highlighted (in this case, between IF and RC[1]).

Unfortunately, the corresponding balanced parenthesis is highlighted for only a brief moment. Sometimes Excel formulas can contain heavily nested expressions. A technique I often use is to go to the edit line, backspace or delete one of the closing parentheses, and immediately re-enter it. This way, I can see where the opening parenthesis is located.

Excel sports another feature to aid in the editing of formulas. The logical formula structure is highlighted in a hovering text message (see Figure 1-14).



Be sure to enable the Function tooltips feature on your Excel Options... General tab. For further information, consult Appendix A.

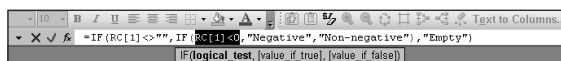


Figure 1-14: Hovering text helps to identify formula structure.

24 Part I: Best Practice Essentials

This use of the hovering text is definitely helpful in keeping your bearings when entering formulas. It zeroes in on the selected portion of the Excel formula, freeing you from being distracted by the rest of the line.

SELECTING CELLS AND SPECIFYING CELL RANGES IN EXCEL FORMULAS

Many Excel functions are capable of performing computations on ranges of cells. Most prevalent of these is the familiar `SUM` worksheet function. Ranges of cells in such formulas are separated by a colon (:) symbol. Examples of this usage include:

```
=SUM(R1C1:R10C4)
=SUM($A$1:$D$10)
```

The preceding examples include equivalent formulas in both the R1C1 and A1 notation (see the section “Understanding Alternative Ways to Represent Cells,” earlier in this chapter). Where possible, I present both these notations or else use named ranges, as appropriate. For cases in which the context is clear and has little or no dependency on the cell-referencing notation, I may use only one notation style.

With regard to screen shots, I use only one of these notations. The choice is based on whatever makes most sense. Quite often, the screen shot illustrates something independent of the notation.



You have the SwitchTool spreadsheet to change modes on your spreadsheets at the press of a button. Therefore, don't worry about the format.

I discuss the Excel `SUM` function in other places and in different contexts throughout this book. Here, I'm just using it as a representative function that accepts multiple cell ranges for its input.

Take a look at the three examples in Listing 1-1.

Listing 1-1: The same formula shown three different ways

```
=SUM(R1C1:R10C4,R4C6:R14C6)
=SUM($A$1:$D$10,$F$4:$F$14)
=SUM(OverHeadAndAdminExpenses,OperatingExpenses)
```

As you can see, Excel functions such as `SUM` can accommodate multiple cell ranges, each separated by a comma. The following representative formulas (see Formulas 1-3 and 1-4) show that the `SUM` function can accommodate a variety of input arguments. I provide the same examples in both the R1C1 notation and A1 notation:

Chapter 1: A Foundation for Developing Best Practices

25

Formula 1-3: Example of Input Arguments for the SUM function in R1C1 Notation

```
=SUM(R21C1:R30C1)
=SUM(R[-10]C:R[-1]C)
=SUM(R21C1:R30C1,R[-10]C:R[-1]C)
=SUM(R[-10]C:R[-1]C,C[1],R4,R5,R6)
=SUM(OneUserDefinedRange,AnotherDefinedRange,StillAnotherUserDefinedRange)
=SUM(OneUserDefinedRange,R[1]C[1],3,4,5,R1C1:R10C3,3.14159)
```

Formula 1-4: Same Formulas as that in Formula 1-3, represented here in A1 Notation

```
=SUM($A$21:$A$30)
=SUM(A11:A20)
=SUM($A$21:$A$30,A12:A21)
=SUM(A13:A22,B:B,$4:$4,$5:$5,$6:$6)
=SUM(OneUserDefinedRange,AnotherDefinedRange,StillAnotherUserDefinedRange)
=SUM(OneUserDefinedRange,B25,3,4,5,$A$1:$C$10,3.14159)
```

Notice that valid cell ranges can include literal values (such as a number), user defined named ranges, cell references (whether relative, absolute or hybrid). Note that a cell range can reference a rectangular swatch; not just only a single row or column.

Keep in mind that named values in Excel need not be restricted to individual cells, but can be indicative of a range of cells for any individual name. As indicated in Listing 1-1 (appearing earlier in the chapter), `OverHeadAndAdminExpenses` could be defined to range over the cells A1 through D10. `OperatingExpenses` might correlate to F4:F14; making the third formula equivalent to the first two. There are certainly other possible combinations of valid formulas equivalent to the formulas shown in Listing 1-1. Consider:

```
=SUM(R1C1:R10C4, OperatingExpenses)
=SUM($A$1:$D$10,OperatingExpenses)
```

OVERLAPPING RANGES

What happens if the ranges separated by commas happen to overlap? Would those overlapping regions be counted more than once? While you to ponder this question, consider another one, which may help: What amount would be computed for the following?

```
=SUM(1,3,1,2)
```

The computed sum in this case doesn't involve cell positions; it just involves literal values. There is no confusion. The result is 7. Ask yourself what SUM is doing. The answer: It is separately evaluating each of the expressions separated by commas and adding up the numeric value of each of the results.

26 Part I: Best Practice Essentials



If you're summing regions that overlap, the overlapped regions can get counted more than once.

The situation is no different when cell references are brought into the picture. Look at Figure 1-15. The formula:

```
=SUM(B3:E12,D9:H20)
```

would first evaluate the sum of B3:E12 (which is 10500), and then evaluate the sum of D9:H20 (which is 10003), and then add these together to get a total of 20503. So, if the numbers are represented more than once, they will be counted more than once.

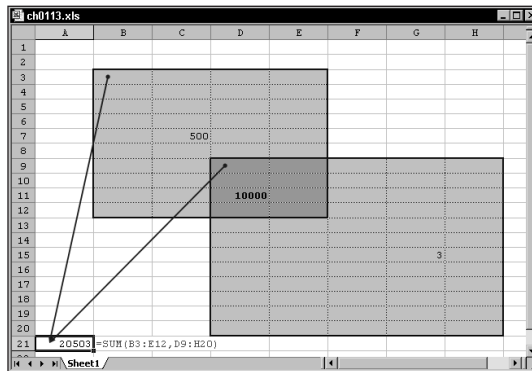


Figure 1-15: Overlapped regions can be double counted.

Keyboard and cursor navigation tips

Although the use of spreadsheets and the windowing environment is intuitive, people sometimes find it not so easy to manage large spreadsheets that may have many thousands of rows of data. There are some things you can do to facilitate moving about on a large spreadsheet and manage formulas. These are essentially trivial, but practical nonetheless.

Open the file ch01-05navigation.xls (see Figure 1-16).

The screen shot shows the beginning and end portions of data that spans 1,000 rows. When you open the file on your CD that accompanies the book, it appears without the horizontal split pane shown in the figure. The file contains data that's actually used in a later chapter (Chapter 9, "How Not to Get Stuck in the M.U.D."), where you'll be addressing some very thorny issues on data ambiguity. The data, as you can see, is a string of numbers. The numbers actually represent dates without a

delimiter to separate days, months, and years. Take a look at the first two entries. If you knew the dates follow the sequence of day-month-year, you could unambiguously interpret 122952 as 12/29/1952. Likewise, you could see that 3643 would be 3/6/1943. The way you reason this out is dependent, in part, upon the length of the string of characters that make up the date string. Does this problem seem intriguing? If you want, you can jump ahead to Chapter 9 to get a sneak peak.

	1	2	3	4	5	6	7	8
1								
2	string	length						
3	122952	6						
4	3643							
5	122750							
6	92047							
7	92552							
8	92749							
9	22948							
10	121851							
994	1259							
995	92149							
996	111563							
997	10947							
998	5148							
999	82045							
1000	5850							
1001	3853							
1002	42349							
1003								
1004								

Figure 1-16: Spreadsheet with continuous stream of data for 1,000 rows

Right now, I want you to deal with the much more mundane task of quickly filling in the needed first steps, the first of which is to compute the length of each of the 1,000 strings. The focus here is in nailing down a rudimentary mechanical technique so that when you turn to large, complex, and challenging problems, you'll be slicing through the data as easily as a hot knife cuts through butter.

The worksheet function used to compute the length of a string of characters is called `LEN`. With the string data in the first column, the formula to compute the string length from the second column is

```
=LEN(RC[-1])
```

This is obviously a simple formula. The key is to replicate it a thousand rows down. One way of doing so is to select the cell in the second row containing your length formula and scroll all the way down until you reach the last piece of data, which happens to be on row 1002. Press Shift and click Select and Fill Down (Ctrl+D). Not particularly hard, right? But ask yourself some questions:

- ◆ How many mouse clicks or key presses did you need to scroll down?
- ◆ What if the number of lines of data were closer to fifteen thousand than one thousand?

Think of some of the shorter steps you could have done. You could grab the vertical scroll bar and pull it all the way down to get to the last row of data. The “last row” happens to be the last row of the worksheet. In any given situation, what guarantee do you have that the specific data you are working on extends all the way down to the bottom of the spreadsheet?

THE PROBLEM: TRYING TO KEEP TRACK OF YOUR DATA

The real spreadsheets that you work with, as opposed to the simplified example from this book, might be chock full of data spread over many columns. And when you scroll down to the bottom the last piece of data you’re looking for, you might not find it there. You think it’s very near the last row of the spreadsheet, so you press PgUp several times. To no avail; it’s not there. Now you go back to manually scanning by moving the vertical scroll bar. You spot the last line of data. You’ve wasted 15–20 seconds searching for that last cell.

To avoid that delay the next time around, you make a mental note of what row it appears in. Say that row happens to be 3874. Chances are, if it’s that far down, your spreadsheet is complex and you’re working under pressure.

The last thing you want to do is to tax your memory with information that has nothing to do with the underlying spreadsheet and interpretive analysis of its data. Also, when you put down your spreadsheet and reopen it next month to add or update data, do you really think you’re going to have row 3874 at the forefront of your thoughts? So there you go chasing after that row of data again, probably along with others as well.

I’m sure you’ve guessed that I have a simple solution in mind. I do. If it’s that simple, why go through all the coy disguises? Why not flat-out state it and be done with it? The answer is that the simple solution is not the moral of the story. The tendency and common practice among many people in business is to apply a “one-off” approach to everything. Nowhere is this more true than in how people use spreadsheets in the business environment.

THE SOLUTION

Here’s the simple solution: I previously said, the string of dates in column 1 was contiguous. Select the first cell containing data in column 1, press Ctrl+Down Arrow and you’re instantly brought down to row 1002 (or whatever row the last contiguous piece of data happens to reside in). Now press the right arrow key once. You’re now 1,000 rows directly below R2C2, which contains the length formula you want to replicate. Now press Shift+Ctrl and, keeping those keys down, also press the Up Arrow key. The addition of the Shift key in this action allows you to select all the intervening cells you just navigated. With all the intervening cells selected (the top cell of which having the formula you want to replicate), click Fill Down (or press Ctrl+D) and you’re done.

A REAL-WORLD COMPLICATION ... AND WHAT TO DO ABOUT IT

Let me add a little complication. Suppose that you want to do the same and select all the cells in the range, but the string of data in column 1 is not contiguous. You

might have two or three cells that are empty. Well, you could quickly “arrow over” to the last piece of data before hitting the empty cell. This could be useful to identify where the breaks are in an otherwise continuous chain of data. Suppose there are significantly more than two or three breaks? What if 10 percent of the many hundreds or thousands of lines have empty cells and for the most part are randomly placed? The technique to moving across the cells using the arrow key as presented wouldn’t be very useful. So, you need to improvise. After going down a screenful or two and seeing that this method is too slow, just select the cell one column over to the right (that is, press the right arrow key once to place yourself on to column 2, which is now essentially empty). Press Ctrl+Down Arrow. This action brings you down to row 65536, the bottom of the spreadsheet. Move one column to the left. Press Ctrl+Up Arrow to move to the very last piece of data in column 1. This is where you were trying to go. Move one column over to the right. Press Ctrl+Shift+Up Arrow and click Fill Down (or press Ctrl+D).

What advantages have you gained by this?

- ◆ You virtually eliminate any guesswork in finding, selecting, moving data for copying, pasting, moving, or otherwise manipulating large blocks of data.
- ◆ The overhead for managing large blocks of data remains constant regardless of how large your spreadsheet grows.
- ◆ By traversing the spreadsheet with lightning speed and removing the burden of thinking about cell locations and placing artificial visual markers, you are freer to focus on the important things in your spreadsheets.



The keystroke-based cell navigation technique will be useful while you are in the process of entering formulas that use cell references spanning many rows and columns. From the Excel Formula Bar, you can click in cells and navigate your spreadsheet using these keystroke techniques. As you move about, the Formula Bar will be populated with the corresponding cell ranges. They will be a definite time saver and reduce errors.

User-defined names within Excel spreadsheets

The feature of user-defined names is an important part of Excel. You likely already know how to use this feature, but I go over the bare-bone basics just in case.

There are two techniques for defining a batch of names.

USING KEYSTROKES TO DEFINE A SEQUENCE OF NAMES

Sometimes it is easier and quicker to define user names by navigating the user menu through keystroke sequences than by using the mouse. If your named ranges

30 Part I: Best Practice Essentials

are single cells, you might want to try the following steps. (I am assuming that you are using the default Excel Edit Options, which involves moving your selection down after pressing Enter. Appendix A has more on setting Excel Options).

1. Group the cells to be named together, generally in a column.
2. Pre-label the cells. The labels can be to the immediate left of the cells to be defined.
3. For each cell to be defined, press the following keystroke sequence:

Alt+i+n+ENTER+ENTER+ENTER

As you type in the keystrokes given in Step 3, the Excel menu will automatically pull down. Excel will “guess” and insert a suggested name in the edit line of the Define Name window (see Figure 1-17).

Note that in column A, a list of labels starts in row 2. These are the names you’ll be giving to the cells you define. Obviously, the choice of names is up to you. As you can see in Figure 1-17, the name being defined for the cell B5 is *TaxRate4*, which matches the label to its immediate left. This is no accident. Excel picked up the name from the label, sparing you the drudgery of entering it. For the nine or ten names you have pre-listed, press the repetitive keystrokes in Step 3 of the preceding list and you’ll quickly be done with defining user names.



This technique of rapid name definition does not apply to named ranges spanning multiple cells for a given name.

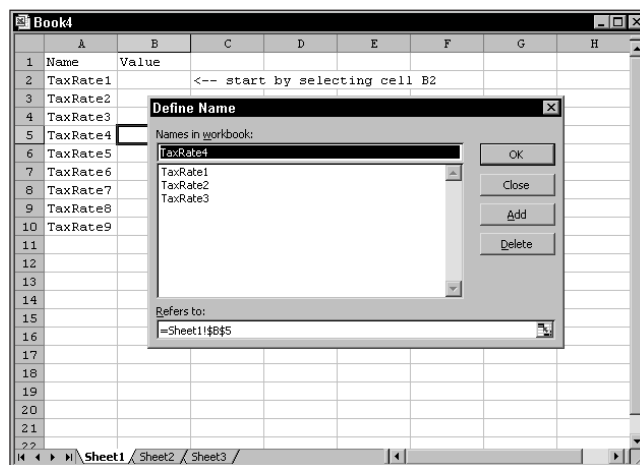


Figure 1-17: Using labels to generate user-defined names

Is there a limit to how many names you can define this way? There really isn't; however, if you'll be defining a great many names, such as hundreds, you may want to consider using an Excel Macro/VBA facility to automate this process. This lies beyond the scope of this book. Applying a visual marker to identify named cells is a good practice. You can apply cell shading, borders or even sticky notes.

USING CTRL+SHIFT+F3 TO DEFINE A BATCH OF NAMES

There's a more rapid and efficient way to define a group of names. You can select the labels and the group of cells to be given names and press Ctrl+Shift+F3. You will then be presented with a dialog box like the one shown in Figure 1-18.

This technique is much easier to perform than the one shown in the previous section, but it does not afford as much flexibility. In the previous method, as you create new names, you can pause momentarily to decide whether you want to tweak the name that Excel suggests.

Both of these methods are effective. Use whatever works for you.

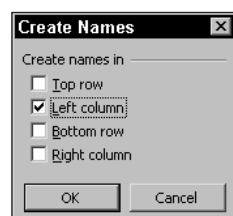


Figure 1-18: Batch creation of names

Excel Functions

Now that you're armed with a little bit about how to express formulas, I want to spend some time discussing Excel functions.

Excel has two very powerful features in it. It's got a terrifically efficient computing engine and is also a powerful tool for visually communicating numerical information. This combination makes Excel particularly potent and vital in business.

Obviously, there's a lot more to Excel than just these two facets. But without these, Excel would be just another faceless application that we would all tire of quickly.

When I speak about the computational engine, I am referring to both the raw number crunching capacity and the range of special-purpose and general functions it possesses.

Number crunching is not always about benchmarks such as how many multiplications can be performed each second. To be sure, Excel on a desktop computer is not going to outperform a supercomputer in sheer number-crunching capacity. One thing it does "excel" at, though, is its unwavering ability to take complex relationships (with multitudes of formulas, each of which can be dependent upon other computations) and combine them with large quantities of data to immediately

produce correct results. This ability to alter a formula in just one or two cells and at the turn of a dime, transforming the overall flow of numbers through spreadsheets, is what sets Excel apart from so many other software packages.

Programming without programming

Excel allows business people to do *programming without programming*. Programming is done visually on the screen, using formulas as your paintbrush and the grid of cells as the canvas. Excel represents programming and software development at its best. Those of you familiar with software development suites and modern programming languages would be hard pressed to come up with conventional programming tools that's as near-universally accepted, flexible, sophisticated, and easy to use, or one that makes the sharing and communication of ideas as easy.

Meanwhile, every spreadsheet is a full-fledged program in its own right. The programming can be as simple as adding a few numbers in a column or as complex as analyzing the monthly sales and financial trends of stores in hundreds of shopping malls throughout the country collected over a five-year period and broken out by NAICS/SIC codes, demographic regions, square footage of leased properties, and so forth.

The essential point is that both extremes, the simple and complicated; are handled by the exact same software tool (Excel). You just don't find many software programs that competently handle both ends of the spectrum.

This is the mental picture I want you to use when thinking about spreadsheets.

I cover only a few Excel functions in this chapter, but introduce others in the next several chapters, in the context in which those functions are used.

Types of Excel formulas you will encounter

My purpose here is twofold: to make you aware of a variety of spreadsheet functions and categories of functions, and to present enough information on some of these functions to keep from interrupting later chapters to explain them there. If you need an encyclopedic presentation of spreadsheets, you'll need to look elsewhere, be it through the Excel Help menu or one of the many books on Excel formulas, such as *Excel 2003 Formulas*, by John Walkenbach (published by Wiley Publishing, Inc.). You may also find some useful information and links on my Web site, at <http://www.evolvingtech.com/excel>.

Excel functions come in a variety of sizes and shapes, which can be viewed within the context of three broad groups. The following list includes these groups and some representative functions for each.

◆ Functions requiring no inputs

```
=RAND()  
=TODAY()  
=TRUE()  
=ROW()
```

◆ Functions that operate on a single value

```
=TRIM(" Remove extra spaces at start and end of string ")  
=INT(99.44)  
=DAY("1/28/2004")  
=ROW(Some_TaxRate_I_Defined)
```

◆ Those that can accept more than one input value or argument

```
=SUMPRODUCT(A1:A10,D1:D10)  
=MID("pick four characters starting the 6th character,6,4)  
=OFFSET(SA$1,3,4)
```

Some Excel functions are “overloaded” and can overlap these categories.

Some important functions and how they’re used

Somewhere along the way in this chapter and other portions of this book, you’ll pick up some useful information on a variety of Excel functions. Don’t concern yourself too much with a formal knowledge of these functions. Instead, focus on the hands-on approach, paying attention to the “gotchas” and subtleties. Think about when and where you would be using these functions. Make no mistake – the functions described here are all put to use somewhere in this book. Chances are good that many of them will become directly relevant to how you use spreadsheets in your line of work.

NO-INPUT FUNCTIONS

No-input functions are just what the name says. Being independently minded creatures, they don’t need to be fed any special parameters to work properly. The most independently minded functions of this type are `TRUE()`, `FALSE()`, and `PI()`.

These functions stubbornly return the same values no matter where or when they perform a computation. You can imagine what `PI()` returns. The function `TRUE()` returns the value `TRUE`; `FALSE()` returns `FALSE`. Why do you need a `TRUE()` or `FALSE()` function in Excel? It turns out there are good uses, but they are not as you might expect. Rather than use one of these functions in a “finished formula,” you might use it as a placeholder in a complex formula in preparation, thereby aiding in testing and validation. Consider the formula:

```
=IF(SomethingUsuallySmall>100000,ExceptionalSituation,UsualSituation)
```

`SomethingUsuallySmall` is a placeholder for a number that is, well, usually small. So how often is it going to be over 100,000? Not very often. As a consequence, the `ExceptionalSituation` is almost never run. In building your spreadsheet, you may have a whole bunch of computations that behave differently when the `ExceptionalSituation` is returned. You may want to temporarily force things

34 Part I: Best Practice Essentials

so that the `ExceptionalSituation` prevails. You can do this by replacing `SomethingUsuallySmall>100000` with the value `TRUE` or the function that generates `TRUE`. Your formula might look like:

```
=IF(TRUE(),ExceptionalSituation,UsualSituation)
```

Of course, you'll change `TRUE()` back to `SomethingUsuallySmall>100000` later. Previously, I allude to more than one kind of use for `TRUE` and `FALSE`. Near the end of this chapter you will see that `TRUE/FALSE` are sometimes needed as special parameters for certain Excel functions.

EXCEL FUNCTIONS HAVING A SPATIAL NATURE

Other no-input functions are not so independently minded. For some functions, it may not matter *when* you call them, but it does matter *where* they are called from. In particular, consider the functions: `=ROW()` and `=COLUMN()`.

These functions report the row and column number of the cell that happens to be making the call. At first glance, the actual row or column number of a cell may not seem that important. Harnessing the positional arrangement can be important.

Consider setting up a table that has a user-defined name called `StartPosition` situated in the top-left corner. In the cells below it, you may be computing values that are dependent on how far below they are in relation to the `StartPosition`. You might use a formula like this:

```
=(ROW()-ROW(StartPosition))*SomeFinancialPenaltyUsefulForYourSpreadsheet
```

Pay attention to the fact that when an input is provided to the `ROW` or `COLUMN` functions, it uses the cell position of that referenced name or cell, rather than its own. `ROW` is one of those “overloaded” functions referred to earlier in this chapter. When no input is provided, the `ROW` function uses its own cell position.

You can probably write an equivalent formula without involving rows. Your table could have been flanked on the left with a vertical column of numbers instead. Because of space and presentation constraints, you may not be free to place a separate vertical column of numbers. The strategy used in computing with rows is to create an “in-lined” numbering system. Instead of relying on an explicit set of numbers that increment by 1 with each row, the `ROW()-ROW(StartPosition)` term accomplishes the same inside the formula. There is an advantage with this technique; as you insert or delete intervening rows and columns, the count on these positional arrangements remains correct. You will see this technique being used in the Data Viewer tool of Chapter 8, “Analyzing Data.” Most typically, this technique of in-lined counting is used with the Excel `OFFSET` function.

EXCEL FUNCTIONS HAVING A TEMPORAL NATURE

Previously I spoke about no-input functions that depend on where they're being called. What about *when* they are called? Spreadsheets can include temporal aspects in their formulas. I would like to call your attention to two Excel functions

that serve that purpose: `TODAY` and `NOW`. To understand the difference between the two, think of `TODAY` as returning `INT(NOW())`. The function `NOW` returns a numeric representation of the current date and time. The hours, minutes, seconds, and fraction of a second form the decimal portion of this numeric quantity. When the function `INT` is applied to `NOW`, it lops off the fractional portion, leaving whole numbers that represent the exact calendar date that would be computed using the `TODAY` function.

For the moment, pay attention to the fact that spreadsheets can be armed with the sense of time. You can, for instance, have a spreadsheet that computes how many days you have left until a note or bond matures. This is not a hypothetical date computation, but rather a *real-time* computation.

EXCEL FUNCTIONS THAT DEPEND ON WHEN AND WHERE THEY ARE CALLED

I suppose by now you must be wondering about no-input functions that change *when* and *where* they are called. On that possibility, let me tell you about the `RAND` function.

The `RAND` function is about as fickle as you can get. It tries not to repeat itself or show any discernable pattern other than providing a decimal number between zero and one. You should be aware that the `RAND` function is really a pseudo random function that only approximates a true random function.

If you need more mathematical horsepower in a spreadsheet environment, you may want to consider using a tool such as Mathematica Link for Excel coupled with Mathematica (products of Wolfram Research).



The Excel `RAND` function is not a strong enough random number generator to be useful for cryptographic routines.

Random numbers can serve many useful purposes. You might, for instance, want to spot-check products by their SKU number or warehouse location and need to generate a list. The `RAND` function obviously generates random numbers, but it's constantly changing. Besides, the numbers are between zero and one. This may not be the range you're looking for.

On both these counts, easy fixes exist. To fix the range problem, "magnify" the random number; that is, perform an arithmetic manipulation to multiply the random numbers by a magnification factor. This magnification factor is the numerical difference between the `UpperLimit` and `LowerLimit`.

```
=LowerLimit+(UpperLimit-LowerLimit)*RAND()
```

The spreadsheet file `ch01-06random.xls` (Figure 1-19) shows this specific formula being used to generate the test values.

	A	B	C
1	EXCEL BEST PRACTICES		
2	Random Number Generator		
3	using the RAND() Worksheet function		
4			
5	UpperLimit	50	
6	LowerLimit	25	
7			
8	Test Value 1	46.61084	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
9	Test Value 2	39.94134	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
10	Test Value 3	26.97563	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
11	Test Value 4	32.51038	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
12	Test Value 5	42.58617	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
13	Test Value 6	33.45058	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
14	Test Value 7	39.64517	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
15	Test Value 8	27.87429	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
16	Test Value 9	26.16128	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
17	Test Value 10	30.22252	=LowerLimit+(UpperLimit-LowerLimit)*RAND()
18			
19			
20			
21			
22			

Figure 1-19: Generating random numbers between an upper and lower limit

Note that the preceding formula contains user-defined names rather than cell references. This method makes the formulas much easier to read. Also note that the identifying information appears at the top of the spreadsheet (see more about this in Chapter 2, “Mastering Spreadsheet Construction Techniques”).

Okay, you have a pretty spreadsheet, but it’s not quite what you want. Suppose the lower and upper limits represent monetary amounts, using U.S. dollars as the currency. You want the generated values to be in terms of dollars and cents. You don’t need all the extra digits. Your first inclination might be to just “format away” the problem; that is, change the formatting of the generated numbers to display information to only two decimal places. This method is not a solution, though, because after you copy and paste values, you still have to contend with the extra digits.

EXCEL FUNCTIONS INVOLVING PRECISION AND FUZZINESS

You can use the `ROUND` worksheet function to round the generated random numbers to two decimal places as is done on the second tab of your spreadsheet workbook, called `TRUNCATED` (see Figure 1-20).

The formula used is

```
=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
```

In Figure 1-20, the Test Value 6 shows as 42.7 rather than 42.70. Nothing is wrong with this number. Both 42.7 and 42.70 represent the same value. For some people, however, showing numbers this way, especially for published reports, can create an eyesore. These people want everything to line up or add up to exactly 100 percent. The usual tendency is for people to adjust these formatting details at the outset so that they don’t have to think about it later. Okay, so you can adjust the

formatting for the generated list to two decimal places, and whisk away the problem. For the generated random numbers, this doesn't present a problem. But what about percentages that must add up to exactly 100%?

	B	C
1	EXCEL BEST PRACTICES	
2	Random Number Generator	
3	rounding to two decimal places	
4		
5	UpperLimit	50=UpperLimit
6	LowerLimit	25=LowerLimit
7		
8	Test Value 1	30.78=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
9	Test Value 2	26.96=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
10	Test Value 3	33.62=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
11	Test Value 4	45.01=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
12	Test Value 5	42.43=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
13	Test Value 6	42.7=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
14	Test Value 7	35.34=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
15	Test Value 8	35.46=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
16	Test Value 9	40.19=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
17	Test Value 10	31.55=ROUND(LowerLimit+(UpperLimit-LowerLimit)*RAND(),2)
18		
19		
20		
21		
22		

Figure 1-20: Rounding to two decimal places

Believe it or not, people make the mistake all the time of displaying numbers using one level of precision and storing it internally in another. I discuss this issue at greater length in Chapter 3, “Your Handy Reference for Manipulating Data.”



Separate your raw data and analysis from your presentation portion of your spreadsheet. Retain the needed numerical precision throughout your computations until you get to the presentation layer, where you can make needed formatting adjustments.

Let me say a word or two for now about rounding. `ROUND` has the following syntax:

`ROUND(number, num_digits)`

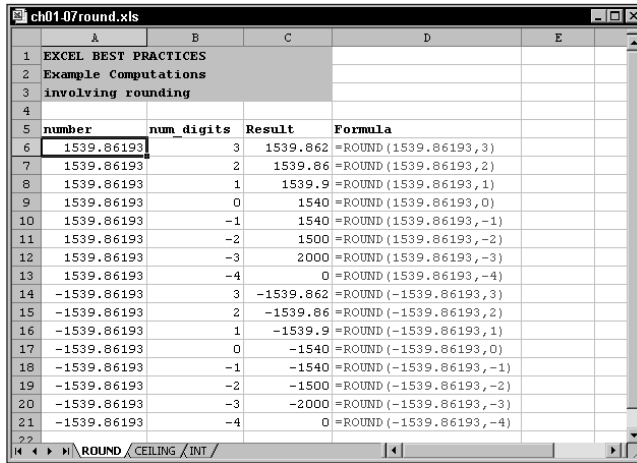
As you've seen, rounding numbers to two decimal places is achieved by using the value 2 for `num_digits`. To zero decimal places or whole numbers, you would use 0 for `num_digits`. Suppose you want to round numbers to the nearest hundred or thousand? You'll notice that rounding works in exponents of 10 though in an inverted pattern. To round to the nearest 1,000, you would use -3 for `num_digits`. Figure 1-21 shows this in action.

Note how the rounding goes up to 1,540 and then back down again to 1,500.

Another function, `CEILING` is similar to `ROUND` in some regards. Rather than specify the number of digits, you specify the actual amount you want to round to. The rounding in `CEILING` *always* rounds up. For example, `CEILING(67,15)` returns

38 Part I: Best Practice Essentials

75. If you want to explore more on the `CEILING` function, look at the second tab of the file `ch01-07round.xls` on your CD-ROM. You may also want to explore the `FLOOR` worksheet function (information on this can be searched from the Excel Help menu).



number	num_digits	Result	Formula
1539.86193	3	1539.862	=ROUND(1539.86193,3)
1539.86193	2	1539.86	=ROUND(1539.86193,2)
1539.86193	1	1539.9	=ROUND(1539.86193,1)
1539.86193	0	1540	=ROUND(1539.86193,0)
1539.86193	-1	1540	=ROUND(1539.86193,-1)
1539.86193	-2	1500	=ROUND(1539.86193,-2)
1539.86193	-3	2000	=ROUND(1539.86193,-3)
1539.86193	-4	0	=ROUND(1539.86193,-4)
-1539.86193	3	-1539.862	=ROUND(-1539.86193,3)
-1539.86193	2	-1539.86	=ROUND(-1539.86193,2)
-1539.86193	1	-1539.9	=ROUND(-1539.86193,1)
-1539.86193	0	-1540	=ROUND(-1539.86193,0)
-1539.86193	-1	-1540	=ROUND(-1539.86193,-1)
-1539.86193	-2	-1500	=ROUND(-1539.86193,-2)
-1539.86193	-3	-2000	=ROUND(-1539.86193,-3)
-1539.86193	-4	0	=ROUND(-1539.86193,-4)

Figure 1-21: Example computations involving rounding

There is another function you need to know about. It's called `INT`. As would be expected for positive numbers, it literally lops off the decimal digits. `INT(2.9)` becomes 2. Negative numbers are treated somewhat differently. They are chopped down to the next lower number. The third tab of the file `ch01-07.xls` shows some examples of this.



Reversing the signs of computation (that is, flipping positives to negatives and vice versa) may not behave as expected, as the following example shows:

`INT(2.9)` returns 2, whereas `INT(-2.9)` returns -3

This is an important subtlety.

EXCEL ADD-IN FUNCTIONS

The random list generator that brought on this whole discussion may still not be quite satisfactory for your needs. You might want to pick whole numbers and dispense decimal points, rounding, and `RAND` altogether. There's an Excel Add-In function called `RANDBETWEEN`. `RANDBETWEEN` takes two inputs, a lower limit and an upper limit. Both these numbers must be integers. `RANDBETWEEN` returns an integer randomly chosen within this range.

I use the term *Add-In* as opposed to worksheet because to use the function, you must tell Excel to load a supplemental module called the Analysis ToolPak (see Figure 1-22). You can load supplemental modules from the Excel Tools: Add-Ins menu. See Figure 1-23 for an example of using `RANDBETWEEN`.

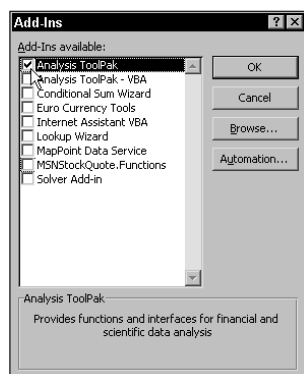
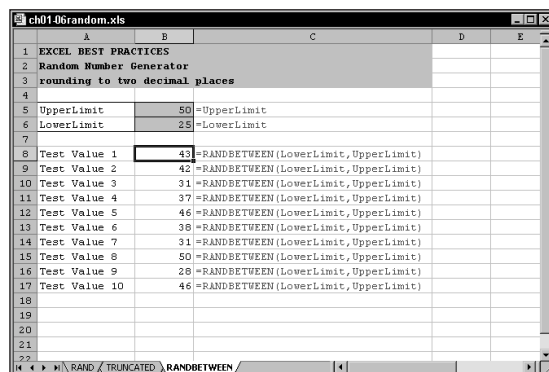


Figure 1-22: Selecting Excel add-ins

Figure 1-23: Usage of `RANDBETWEEN`

When you tell Excel to load an Add-In, it does so every time you fire up Excel from that point on, with one exception. Sometimes your Excel may be loaded off of a network rather than from your local machine. The settings for your copy of Excel may be preset and unchangeable, and the changes you make to your settings may not be retained. If this is the case, you've got a bigger set of issues than keeping `RANDBETWEEN` in the sights of Excel. If your Excel configuration is pre-set, changes you make to your settings will be forgotten the moment you quit Excel.



If you are sharing your spreadsheet with a colleague or third party and the spreadsheet uses `RANDBETWEEN` or some other Add-In function, make sure that user has the appropriate Add-In modules loaded.

Assuming that you have no problems getting your copy of Excel to automatically work with `RANDBETWEEN`, you still may have a problem down the road. Suppose that you want to share your spreadsheet with a colleague and your spreadsheet happens to use `RANDBETWEEN` or some other function in one of the other modules. Your colleague may not have that module loaded on his or her machine. Think carefully about your usage of Add-In functions if you plan on sharing work with others.

There are still two issues looming on the random-number list generator. They're actually quite easy to solve. The first issue is that anytime something changes in the spreadsheet, the random numbers are recalculated. Because the formulas are constantly changing the values returned, you can't preserve the numbers within the formulas. Aha! There's no need to preserve the numbers within the formulas. Just copy and paste the values to an adjacent column or wherever is appropriate. That was an easy fix. The next issue is a little more involved and context based.

REAL WORLD USE OF `RANDBETWEEN`

Selecting random in a range is straightforward. Not everything in the world is going to be listed as a continuous range of numbers like the kind you might generate from the previously shown worksheets. If you are looking for "items" in a list, then you can use one of the lookup functions that Excel provides and combine that function with your random-number generator.

Imagine, for example, that you are auditing credit card transactions for a business for the month of December 2003. There are 800 transactions for this period. You have the responsibility of spot-checking or auditing the amounts of 20 randomly selected items from the list of recorded transactions. As you can see in Figure 1-24, there are two principal regions in the spreadsheet. On the right side is a table containing all 800 transactions, including date and dollar amount. If you open the file `ch01-08sampling` from the CD with this book, you can see that the transaction data has a name, `Table01`, which extends from G8 through I808 (in R1C1 notation the cell coordinates are `R7C7:R809C9`). The left side of the spreadsheet contains the data being used for testing. The transaction IDs were randomly selected between 1 and 800 using this formula:

```
=RANDBETWEEN(LowerLimit,UpperLimit)
```

MORE EXCEL SPATIAL FUNCTIONS

The smaller list of transaction IDs that will be used for audit testing were given a user-defined name, `TransID` (which extends from A7 through A26). After the values were generated, the cells were copied and pasted (values only) so that the original

formulas were obliterated. This ensures that the values won't continue to change. On the next two columns, table lookups for specific transaction data are performed. The function used for this purpose is `VLOOKUP`. As it is used in `ch01-08sampling.xls`, `VLOOKUP` takes the following form:

`VLOOKUP(value_to_lookup, table_or_range_of_data, column_number)`

The formula for looking up the transaction amounts is

`=VLOOKUP(TransID,Table01,3)`

The number 3 appears in this formula because the transaction amount is found on the third column of `Table01`.

You should be aware that there is also an `HLOOKUP` function that is like `VLOOKUP` turned sideways. Both of these functions are overloaded.

Sampled Data for Testing					Population Data		
TransID	Date	Sales per Books	Audited Value	Difference	transID	Date	Amount
672	12/27/2003	56.71			1	12/1/2003	23.56
505	12/20/2003	60.22			2	12/1/2003	32.74
158	12/7/2003	56.0			3	12/1/2003	23.49
308	12/13/2003	65.31			4	12/1/2003	68.08
101	12/5/2003	7.73			5	12/1/2003	36.06
396	12/16/2003	124.5			6	12/1/2003	46.87
6	12/1/2003	46.87			7	12/1/2003	40.6
112	12/5/2003	79.97			8	12/1/2003	19.27
217	12/10/2003	97.37			9	12/1/2003	69.66
542	12/22/2003	144.36			10	12/1/2003	88.17
243	12/11/2003	64.36			11	12/1/2003	17.79
643	12/26/2003	28.05			12	12/1/2003	44.07
91	12/4/2003	47.89			13	12/1/2003	33.68
647	12/26/2003	59.12			14	12/1/2003	20.31

Figure 1-24: Worksheet for comparing audited values to book values

I want to turn your attention to some potential trouble spots with `VLOOKUP`. If you're the list of items you are searching is "out of sequence," the value returned by `VLOOKUP` may not be what you are looking for.

In this example, the lookup values in `Table01` starts from an ID of 1 and ends with 800. It follows a nice, structured order. What if lines 6 and 8 were switched (including the ID, date, and transaction amount)? The data is still correct; it's just presented in a different sequence. (See Figure 1-25, in which the relevant information is highlighted.) While looking up the number 6 in `Table01`, `VLOOKUP` finds 5, and then immediately stubs its toe when it reaches the number 8. This out-of-sequence number causes `VLOOKUP` to stumble and fall out of the computation

42 Part I: Best Practice Essentials

prematurely, holding the data for the value 5. The transaction amount having the ID of 5 is 36.06. This is not, of course, the number that you're looking for.

Sampled Data for Testing					Population Data		
TransID	Date	Sales per Books	Audited Value	Difference	transID	Date	Amount
672	12/27/2003	56.71			1	12/1/2003	23.56
505	12/20/2003	60.22			2	12/1/2003	32.74
158	12/7/2003	56.8			3	12/1/2003	23.49
308	12/13/2003	65.31			4	12/1/2003	68.08
101	12/5/2003	7.73			5	12/1/2003	36.06
396	12/16/2003	124.5			8	12/1/2003	19.27
6	12/1/2003	36.06			7	12/1/2003	40.6
112	12/5/2003	79.97			6	12/1/2003	46.87
217	12/10/2003	97.37			9	12/1/2003	69.66
942	12/22/2003	144.36			10	12/1/2003	88.17
243	12/11/2003	64.36			11	12/1/2003	17.79
643	12/26/2003	28.05			12	12/1/2003	44.07
91	12/4/2003	47.89			13	12/1/2003	33.68
647	12/26/2003	59.12			14	12/1/2003	20.31

Figure 1-25: VLOOKUP is easily confused if the data being looked up is out of sequence.

This problem can be easily fixed, fortunately. Add an additional parameter called FALSE to your formula. The revised lookup formula is

```
=VLOOKUP(TransID,Table01,3,FALSE)
```

This formula returns the result of 46.87 and is the result you would expect.



If you are using VLOOKUP or HLOOKUP and the table you are searching is not in strict numeric order, you may need to place the parameter FALSE in your formula.

Closing Thoughts

I hope that this chapter got you on the road to understanding differences among absolute, relative and hybrid cell references, along with ways to put them to use. Also illustrated in this chapter were different ways to write your formulas using A1 notation, R1C1 notation and user-defined names. To assist you in this process, you now have at your disposal a “SwitchTool” spreadsheet, and keystroke techniques for defining names in rapid succession and navigating/managing larger spreadsheets.

You have been introduced to a view that Excel functions can be used for more than just plain-vanilla arithmetic calculations. From a hands-on perspective, you have seen a variety of spreadsheet functions in action, including TRUE, FALSE, ROW, COLUMN, TODAY, NOW, RAND, RANDBETWEEN, ROUND, CEILING, INT, and VLOOKUP.

Chapter 1: A Foundation for Developing Best Practices

43

What should start to emerge is that powerful things you can do with spreadsheets begin to happen when you combine things. When `RANDBETWEEN` and `VLOOKUP` were combined, it became possible to randomly select items that did not match the format of random integers.

There are subtleties in functions that you should now be aware of, such as when you may need to insert a `FALSE` parameter in the lookup functions. This kind of knowledge should help you steer clear of sand traps that frequently snag people. When people do get snared, they consume and waste precious time coming up with a fix. More often than not, the fix is a quick fix. Pretty soon, people fall into the habit of doing things in a “one-off” style. This habit is a slippery slope.

The tips, cautionary notes and discussions presented in this chapter and throughout the book should help you to avoid this one-off style; whether in the writing of spreadsheet formulas or the construction of spreadsheets. Chapter 2 continues and expands on best practices for spreadsheet construction.

