EXCEL gurus gone wild

Do the IMPOSSIBLE with Microsoft Excel

Bill Jelen

EXCEL gurus gone wild

by

Bill Jelen

with 112K contributors

and 6,156,123 guests

HOLY MACRO! BOOKS

PO Box 82, Uniontown, OH 44685

Excel Gurus Gone Wild

Printed in USA by Malloy

First Printing: March 2009

Author: Bill Jelen

Technical Editor: Bob Umlas

Editor: Kitty Wilson

Interior Design: Fine Grains (India) Private Limited, New Delhi, India

Cover Design: Shannon Mattiza, 6Ft4 Productions

Published by: Holy Macro! Books, 13386 Judy Ave, PO Box 82, Uniontown OH 44685

Distributed by Independent Publishers Group

ISBN 978-1-932802-49-8

Library of Congress Control Number: 2008923727

ABOUT THE AUTHOR

Bill Jelen is the host of MrExcel.com and the author of 24 books. But this book was really written by thousands of people who contributed to the MrExcel.com message board or who have spoken up at one of my Power Excel seminars with a cool trick. In many cases, new ideas at the board happen through a collaborative process – someone asks a question, others answer, others build on that answer, Someone posts something simply amazing, more people build on the amazing concept, and then a whole bunch of really talented Excel gurus will offer kudos for a really slick solution. If you are one of the people who have participated in this process, this this book is also written by you.

About the MrExcel Message Board: While MrExcel.com debuted on November 21, 2008, the message board did not debut until 2009. Using a script from Matt's Script Archive, the original message board was born. Bill wrote, "you can still keep sending your questions to me via e-mail, or you can post them at the message board. And, if you happen to be at the message board and see an easy question, give that person an answer." He figured it would be sort of like the take-a-penny, leave-a-penny cup that you see by the cash register. In the early months, Ivan Moala, Dave and Cecilia became frequent regulars at the board. Today, over 100,000 people have signed up as members, there are many hundreds of regulars, and six million unique visitors pass through the site annually. The community at the MrExcel.com message board continues to provide answers to 30,000 questions a year. In fact, with 365,000 answers archived, it is likely that the answer to nearly any Excel question has already been posted.

ACKNOWLEDGMENTS

Thank you to the entire community at the MrExcel.com message board. This includes everyone who has posted a question, suggested an answer and those who lurk without posting.

MrExcel.com was started in 1998. Over the years, a number of folks have been on the payroll and helped grow the site: Anne Troy. Mala Singh. Juan Pablo Gonzalez. Tracy Syrstad. Suat Ozgur. Far more people volunteer and keep the community humming. Kristy Sharpe,

Greg Truby, Nate Oliver, Paddy Davies, Richard Schollar, Chris Smith, Suat Ozgur, Zack Barresse, Ivan Moala, Joe4, all volunteer as admins and moderators.

Thanks to the pioneers – the first regulars, back when it might take 4 hours before Ivan Moala, Dave from Oz or Cecilia from the Pacific Northwest would check in with an answer.

Mala Singh of XLSoft Consulting wrote the speedometer and macroeconomic supply curve add-ins and does all of our engineering projects (see examples at http://www.mrexcel.com/graphics.shtml). Mala pitched in to help write up some topics in this book. Thanks to Mala for helping to get the book to press.

Aladin Akyurek is the king of Ctrl+Shift+Enter formulas. Like everyone else, I usually have to run his formulas through the Auditor Toolbar's Evaluate Formula feature before I can figure out how it is working, but they always impress. Aladin's tireless formula examples have raised many to the level of Excel guru.

This list could fill an entire book, but thanks to the folks who answer a lot of questions: Norie, Aladin Akyurek, Andrew Poulsom, Smitty, Jindon, Erik.Van.Geit, Richardschollar, Vog, Jonmo1, Von Pookie, Paddyd, Joe4, Barry Houdini, Juan Pablo González, Mark W., Yogi Anand, HOTPEPPER, Peter_Sss, Just_Jon, Zack Barresse, Nateo, Fairwinds, Tusharm, Tom Urtis, Greg Truby, Sydneygeek, Brian From Maui, Joe Was, Lenze, Oaktree, Halface, Domenic, Brianb, Glennuk, Datsmart, Donkeyote, Nimrod, Tom Schreiner, Rorya, Mikerickson, Pgc01, NBVC, Acw, Onlyadrafter, Andrew Fergus, Daniels012, Tommygun, Tazguy37, Steveo59l, Ivan F Moala, Schielrn, Texasalynn, Fazza, Barry Katcher, Lewiy, Alexander Barnes, Phantom1975, Damon Ostrander, DRJ, Ralpha, Domski, Mark O'Brien, Tactps, Stanleydgromjr, Mudface, Richie(UK), Starl, Parry, Al_B_Cnu, Cbrine, Jack In The UK, Todd Bardoni, Jon Von Der Heyden, Gerald Higgins, Iridium, Jon Peltier, Dougstroud, Seti, Thenooch, Stormseed, Jaafar Tribak, Dk, PA HS Teacher, Xlgibbs, Xld, Anne Troy, Ravishankar, Dave3009, Hatman, Jimboy, Barrie Davidson, Venkat1926, Krishnakumar, Njimack, Jay Petrulis, Vicrauch, Iliace, P Sitaram, Gates Is Antichrist, Oorang, XL-Dennis, Shajueasow, Markandrews, RAM, Santeria, SIXTH SENSE, Giacomo, Chris Davison,

Eliw, IML, Nbrcrunch, Cornflakegirl, Agihcam, Itr674, Dave Hawley, Mike Blackman, Sweater_Vests_Rock, Derek, Dominicb, Shades, Travis, Corticus, Brew, Plettieri, Dave Patton, Hawaiian Harry, Iggydarsa, Glaswegian, Bat17, Boller, Al Chara, Brian.Wethington, Colo, Ekim, Mortgageman, Jim May, Russell Hauf, Babycody, Yee388, Btadams, Mdmilner, Ktab, Howard, Mrkowz, Kskinne, Mickg, Gord, Earlyd, Pekkavee, Ian Mac, Kenneth Hobson, Verluc, Fergus, Willr, CT Witter, Davers, Ndendrinos, Arthurbr, Ajm, Sal Paradise, Thorpyuk, Staticbob, Pauljj, Dbmathis, Brettvba, Chitosunday, Pcc, ADAMC, PATSYS, Ponsy Nob., Ponsonby, Sykes, Martinee, Justinlabenne, Bubbis Thedog, Shippey121, Harvey, Excelr8r, Yard, Sssb2000

I appreciate the people from the MrExcel community whom I've had the opportunity to meet over the years. It is always cool to discuss Excel tricks over lunch or dinner. I'll miss a few, but Tracy, Juan Pablo, NateO, Greg Truby, Smitty, Richard, Jon, Bryony, Russ, Mel, Aaron, Brian from Maui, Matt aka Oaktree, Jay Petrulis, Tushar, Chip Pearson, Duane Aubin, Asaad Alli, Freddy Fuentes and a dozen more who I aren't popping into my head at this moment.

Thanks to the guys on the Excel team who I know on a first name basis. Dave, Chad, Charlie, Joe, - your dedication to the world's best spreadsheet is appreciated. I also appreciate that you always answer my e-mails when I have a bizarre Excel question or oddity.

Thanks to my friends; facebook friends, Facebook fans, Twitter followers, podcast viewers, friends from the old TechTV, readers who write in with ideas, anyone wearing an Excel Master pin, those who reaches for a MrExcel book when they have a problem, people who come to my seminars in places like Springfield Missouri, Columbus Indiana, and Madison Wisconsin (at least one cool tip in this book came from those cities and more).

At the office, thanks to Lora White for keeping things running and editing the podcasts and thanks to my sister Barb Jelen who packed and shipped this book to you.

Thanks to my family. Josh, Zeke, and Mary Ellen.

DEDICATION

Dedicated to every person who has ever answered a question at the MrExcel Message Board.

FOREWORD

This book was born in a British pub.

I was in England in 2007 to perform a couple more Power Excel Seminars. The night before the seminar in Southampton, a group of people from the MrExcel Message board got together for dinner. Russ Cockings, Bryony Stewart-Seume, Richard Schollar, Jon Von Der Hayden and Mel Smith were talking about some amazing tricks that they've seen at the board when someone, probably either Jon or Richard comments that their MrExcel favorites list was a veritable reference guide to Excel and VBA. These are very smart people who know a whole lot about Excel. You have to wonder what types of things would impress this group enough to cause them to add it to their favorites list.

You can check out that favorites list at http://www.mrexcel.com/favorites.html. You will find amazing ideas as you browse those topics. Yes, some are niche topics and many are arcane. However, if you use Excel all day, it is pretty wild to find someone who was able to coax the impossible out of our favorite spreadsheet.

To say that this book is a niche book is an understatement. I am not out to reach the masses with this book. Topics in this book are arcane. A person who uses Excel for 2000 hours per year might need to use any given topic once every 20,000 years. It is probably 1 tenth of one percent of the people using Excel will find any of this stuff fascinating. If you happen to be one of these people, then this book is for you. For the other 99.9%, take this book back to the bookstore and exchange it for Learn Excel 97-2010 from MrExcel or Pivot Table Data Crunching.

For the first two sections of this book, my general requirement for inclusion was that the topic had to be amazing to either me or favorited by a number of MrExcel MVP's. Some things came from the MrExcel Message Board, others came up during my Excel seminars. In the third section, I go through some basics to get you comfortable with Excel VBA and then launch into some amazing VBA utility macros. The appendix is an Excel function reference, with suggested uses for 120 of the 362 functions.

If you want to try out a technique, the files used in the production of the book are available for download at http://www.mrexcel.com/gurufiles.html.

Contents

SECTION 2 - TECHNIQUES

SECTION 3 - MACROS

PART 1

FORMULAS

Find the First Non-Blank Value in a Row

Challenge: You want to build a formula to return the first non-blank cell in a row. Perhaps columns B:K reflect data at various points in time. Due to the sampling methodology, certain items are checked infrequently.

Solution: In Figure 1, the formula in A4 is:

```
=INDEX(C4:K4,1,MATCH(1,INDEX(1-ISBLANK(C4:K4),1,0),0))
```

Although this formula deals with an array of cells, it ultimately returns a single value, so you do not need to use Ctrl+Shift+Enter when entering this formula.

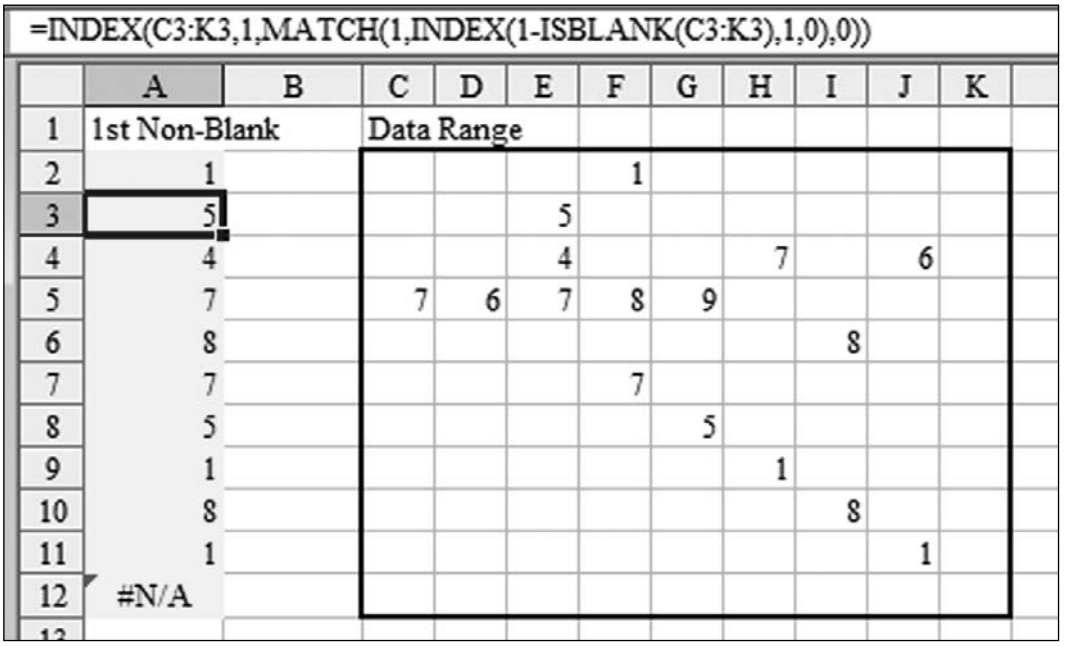


Figure 1. You find the first non-blank cell in each row of C2:K12 and return that value in column A.

Breaking It Down: Let's start from the inside. The `ISBLANK` function returns `TRUE` when a cell is blank and `FALSE` when a cell is non-blank. Look at the row of data in `C4:K4`. The `ISBLANK(C4:K4)` portion of the formula will return:

```
{TRUE,TRUE,FALSE,TRUE,TRUE,FALSE,TRUE,FALSE,TRUE}
```

Notice that this array is subtracted from 1. When you try to use `TRUE` and `FALSE` values in a mathematical formula, a `TRUE` value is treated as a 1, and a `FALSE` value is treated as a 0. By specifying `1-ISBLANK(C4:K4)`, you can convert the array of `TRUE/FALSE` values to 1s and 0s. Each `TRUE` value in the

ISBLANK function changes to a 0. Each FALSE value changes to a 1. Thus, the array becomes:

```
{0,0,1,0,0,1,0,1,0}
```

The formula fragment `1-ISBLANK(C4:K4)` specifies an array that is 1 row by 9 columns. However, you need Excel to expect an array, and it won't expect an array based on this formula fragment. Usually, the INDEX function returns a single value, but if you specify 0 for the column parameter, the INDEX function returns an array of values. The fragment `INDEX(1-ISBLANK(C4:K4),1,0)` asks for row 1 of the previous result to be returned as an array. Here's the result:


```
{0,0,1,0,0,1,0,1,0}
```

The MATCH function looks for a certain value in a one-dimensional array and returns the relative position of the first found value. `=MATCH(1,Array,0)` asks Excel to find the position number in the array that first contains a 1. The MATCH function is the piece of the formula that identifies which column contains the first non-blank cell. When you ask the MATCH function to find the first 1 in the array of 0s and 1s, it returns a 3 to indicate that the first non-blank cell in `C4:K4` occurs in the third cell, or `E4`:

Formula fragment: `MATCH(1,INDEX(1-ISBLANK(C4:K4),1,0),0)`

Sub-result: `MATCH(1, {0,0,1,0,0,1,0,1,0},0)`

Result: 3

At this point, you know that the third column of `C4:K4` contains the first non-blank value. From here, it is a simple matter of using an INDEX function to return the value in that non-blank cell. `=INDEX(Array,1,3)` returns the value from row 1, column 3 of an array:

Formula fragment: `=INDEX(C4:K4,1,MATCH(1,INDEX(1-ISBLANK(C4:K4),1,0),0))`

Sub-result: `=INDEX(C4:K4,1,3)`

Result: 4

Additional Details: If none of the cells are non-blank, the formula returns an `#N/A error`.

Alternate Strategy: Subtracting the ISBLANK result from 1 does a good job of converting TRUE/FALSE values to 0s and 1s. You could skip this step, but then you would have to look for FALSE as the first argument of the MATCH function:

```
=INDEX(C4:K4,1,MATCH(FALSE,INDEX(ISBLANK(C4:K4),1,0),0))
```

Summary: The formula to return the first non-blank cell in a row starts with a simple ISBLANK function. Using INDEX to coax the string of results into an array allows this portion of the formula to be used as the lookup array of the MATCH function.

Source: http://www.mrexcel.com/forum/showthread.php?t=53223

CALCULATE WORKDAYS FOR 5-, 6-, AND 7-DAY WORKWEEKS

Challenge: Calculate how many workdays fall between two dates. Excel's NETWORKDAYS function does this if you happen to work the five days between Monday and Friday inclusive. This topic will show you how to perform the calculation for a company that works 5, 6, or 7 days a week.

Background: The NETWORKDAYS function calculates the number of workdays between two dates, inclusive of the beginning and ending dates. You specify the earlier date as the first argument, the later date as the second argument, and optionally an array of holidays as the third argument. In Figure 2, cell C3 calculates only 5 workdays because February 16, 2009, is a holiday. This is a cool function, but if you happen to work Monday through Saturday, it will not calculate correctly for you.

C3 fx =NETWORKDAYS(A3,B3,$E$2:$E$11)

	A	B	C	D	E	F
1	Project Start	Project Due	Workdays		Holidays	
2	Tue, Jan 20, 09	Tue, Jan 27, 09	6		1/1/2009	
3	Tue, Feb 10, 09	Tue, Feb 17, 09	5		1/19/2009	
4	Fri, Jul 10, 09	Mon, Jul 13, 09	2		2/16/2009	
5	Fri, Jan 2, 09	Thu, Dec 31, 09	251		5/25/2009	
6					7/3/2009	
7					9/7/2009	
8					10/12/2009	
9					11/11/2009	
10					11/26/2009	
11					12/25/2009	

Figure 2. Traditionally, NETWORKDAYS assumes a Monday–through-Friday workweek.

Setup: Define a range named Holidays to refer to the range of holidays.

Solution: The formula in C3 is:

```
=SUMPRODUCT(--(COUNTIF(Holidays,ROW(INDIRECT(A3&":"&B3)))
=0),--(WEEKDAY(ROW(INDIRECT(A3&":"&B3)),3)<6))
```

Although this formula deals with an array of cells, it ultimately returns a single value, so you do not need to use Ctrl+Shift+Enter when entering this formula.

Breaking It Down: The formula seeks to check two things. First, it checks whether any of the days within the date range are in the holiday list. Second, it checks to see which of the dates in the date range are Monday-through-Saturday dates.

You need a quick way to compare every date from A3 to B3 to the holiday list. In the current example, this encompasses only 8 days, but down in row 5, you have more than 300 days.

The formula makes use of the fact that an Excel date is stored as a serial number. Although cell A3 displays February 10, 2009, Excel actually stores the date as 39854. (To prove this to yourself, press Ctrl+` to enter Show Formulas mode. Press Ctrl+` to return to Normal mode.)

It is convenient that Excel dates in the modern era are in the 39,000–41,000 range, well within the 65,536 rows available in Excel 97-2003. The date corresponding to 65,536 is June 5, 2079, so this formula will easily continue to work for the next 70 years. (And if you haven't upgraded to Excel 2007 by 2079, well, you have a tenacious IT department.)

Excel starts evaluating this formula with the first `INDIRECT` function. The arguments inside `INDIRECT` build an address that concatenates the serial number for the date in A3 with the serial number for the date in B3. As you can see in the sub-result, you end up with a range that points to rows 39854:39861:

Formula fragment: `INDIRECT(A3&":"&B3)`

Sub-result: `INDIRECT("39854:39861")`

Normally, you would see something like `"A2:IU2"` as the argument for `INDIRECT`. However, if you have ever used the `POINT` method of entering a formula and gone from column A to the last column, you will recognize that `=SUM(2:2)` is equivalent to `=SUM(A2:IV2)` in Excel 2003 and `=SUM(A2:XFD2)` in Excel 2007.

The first step of the formula is to build a reference that is one row tall for each date between the start and end dates.

Next, the formula returns the `ROW` function for each row in that range. In the case of the dates in A3 and A4, the formula returns an array of eight row numbers (in this case, `{39854;39855;39856;…;39861}`). This is a clever way of returning the numbers from the first date to the last date. In row 5, the `ROW` function returns an array of 364 numbers:

Formula fragment: `ROW(INDIRECT(A3&":"&B3))`

Sub-result: `{39854;39855;39856;39857;39858;39859;39860;39861}`

Now you can compare the holiday list to the range of dates. `=COUNTIF(Holidays,sub-result)` counts how many times each holiday is in the range of dates. In this case, you expect the function to return a 1 if a holiday is found in the range of dates and a 0 if the holiday is not found. Because you want to count only the non-holiday dates, the formula compares the `COUNTIF` result to find the dates where the holiday `COUNTIF` is 0:

Formula fragment: `--COUNTIF(Holidays,ROW(INDIRECT(A3&":"&B3)))=0`

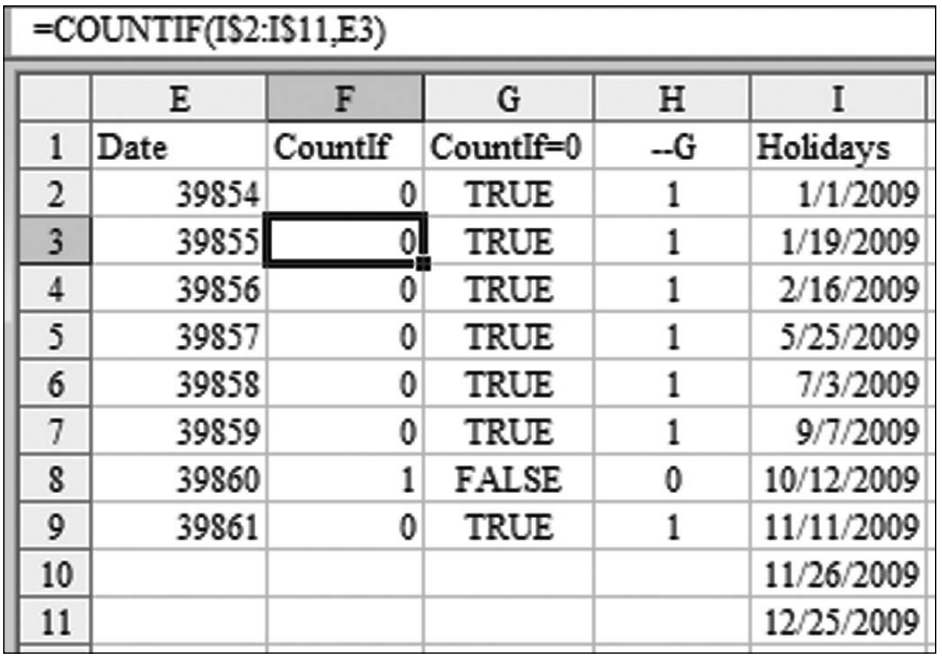
Result: `{1;1;1;1;1;1;0;1;1}`

For every date in the date range, the `COUNTIF` formula asks, “Are any of the company holidays equal to this particular date?” Figure 3 illustrates what is happening in the first half of the formula. Column E represents the values returned by the `ROW` function. Column F uses `COUNTIF` to see if any of the company holidays are equal to the value in column E. For example, in E3, none of the holidays are equal to 39855, so `COUNTIF` returns 0. However, in F8, the formula finds that one company holiday is equivalent to 39860, so `COUNTIF` returns 1.

In column G, you test whether the result of the `COUNTIF` is 1. If it is, the `TRUE` says to count this day.

In column H, the minus-minus formula converts each `TRUE` value in column G to 1 and each FALSE value in column G to 0.

In Figure 3, cells H2:H9 represent the virtual results of the first half of the formula, which finds the dates that are not holidays.



=COUNTIF(I$2:I$11,E3)

	E	F	G	H	I
1	Date	CountIf	CountIf=0	--G	Holidays
2	39854	0	TRUE	1	1/1/2009
3	39855	0	TRUE	1	1/19/2009
4	39856	0	TRUE	1	2/16/2009
5	39857	0	TRUE	1	5/25/2009
6	39858	0	TRUE	1	7/3/2009
7	39859	0	TRUE	1	9/7/2009
8	39860	1	FALSE	0	10/12/2009
9	39861	0	TRUE	1	11/11/2009
10					11/26/2009
11					12/25/2009

Figure 3. The first half of the formula counts days that are not holidays.

The second half of the formula uses the WEEKDAY function to find which dates are not Sundays. The WEEKDAY function can return three different sets of results, depending on the value passed as the Return_Type argument. Figure 4 show the values returned for various Return_Type arguments. In order to isolate Monday through Saturday, you could check to see if the WEEKDAY function with a Return_Type of 1 is greater than 1. You could check to see if the WEEKDAY function with a Return_Type of 2 is less than 7. You could check to see if the WEEKDAY function with a Return_Type of 3 is less than 6. All these methods are equivalent.

=WEEKDAY($A3,B$2)

	A	B	C	D	E
1		Return_Type ------->			
2		1	2	3	
3	Sun, Feb 8, 09	1	7	6	
4	Mon, Feb 9, 09	2	1	0	
5	Tue, Feb 10, 09	3	2	1	
6	Wed, Feb 11, 09	4	3	2	
7	Thu, Feb 12, 09	5	4	3	
8	Fri, Feb 13, 09	6	5	4	
9	Sat, Feb 14, 09	7	6	5	
10					
11	1	Sunday = 1, Saturday = 7			
12	2	Monday=1, Sunday = 7			
13	3	Monday=0, Sunday=6			
14					

Figure 4. The WEEKDAY function can return 1, 7, or 6 for Sundays.

The second half of the formula uses many of the tricks from the first half. The INDIRECT function returns a range of rows. The ROW function converts those rows to row numbers that happen to correspond to the range of dates. The WEEKDAY(,3) function then converts those dates to values from 0 to 6, where 6 is equivalent to Sunday. The virtual result of the WEEKDAY function is shown in column L of Figure 5. The formula compares the WEEKDAY result to see if it is less than 6. This virtual result is shown in column M of Figure 5. Finally, a double minus converts the TRUE/FALSE values to 0s and 1s, as shown in column N. Basically, this says that we are working every day in the range, except for N7, which is a Sunday.

Formula fragment: `--(WEEKDAY(ROW(INDIRECT(A3&":"&B3)),3)<6)`

Result: `{1;1;1;1;1;1;0;1}`

L3 =WEEKDAY(K3,3)

	K	L	M	N	O	P
1	Date	Weekday(3)	L<6	Not Sunday		H*N
2	39854	1	TRUE	1		1
3	39855	2	TRUE	1		1
4	39856	3	TRUE	1		1
5	39857	4	TRUE	1		1
6	39858	5	TRUE	1		1
7	39859	6	FALSE	0		0
8	39860	0	TRUE	1		0
9	39861	1	TRUE	1		1
10				Total		6

Figure 5. The 1s in column N mean the date is not a Sunday.

Finally, SUMPRODUCT multiplies the `Not Holiday` array by the `Not Sunday` array. When both arrays contain a 1, we have a workday. When either the `Not Holiday` array has a 0 (as in row 8) or the `Not Sunday` array has a 0 (as in row 7), the result is a 0. The final result is shown in the SUM function in P10: There are 6 workdays between the two dates.

As with most array solutions, this one formula manages to do a large number of sub-calculations to achieve a single result.

Additional Details: What if you work 7 days a week but want to exclude company holidays? The formula is simpler:

```
=SUMPRODUCT(--(COUNTIF(Holidays,ROW(INDIRECT(A2&":"&B2)))=0))
```

The problem becomes trickier if days in the middle of the week are the days off. Say that you have a part-time employee who works Monday, Wednesday, and Friday. The `Not Sunday` portion of the formula now needs to check for 3 specific weekdays. Note that the `Return_Type 2` version of the WEEKDAY function never returns a 0. Because this version of the WEEKDAY function returns digits 1 through 7, you can use it as the first argument in the CHOOSE function to specify which days are workdays. Using `=CHOOSE(WEEKDAY(Some Date,2),1,0,1,0,1,0,0)` would be a way of assigning 1s to Monday, Wednesday, and Friday.

Because CHOOSE does not usually return an array, you have to enter the following formula, using Ctrl+Shift+Enter:

```
=SUMPRODUCT(--(COUNTIF(Holidays,ROW(INDIRECT(A3&":"&B3)))=0),-
-(CHOOSE(WEEKDAY(ROW(INDIRECT(A3&":"&B3)),2),1,0,1,0,1,0,0)))
```

Summary: This topic introduces the concept of creating a huge array from two simple values. For example, `=ROW(INDIRECT("1:10000"))` generates a 10,000-cell array filled with the numbers from 1 to 10,000. You can use this concept to test many dates while only specifying a starting and ending point, thus solving the `NETWORKDAYS` problem for any type of workweek.

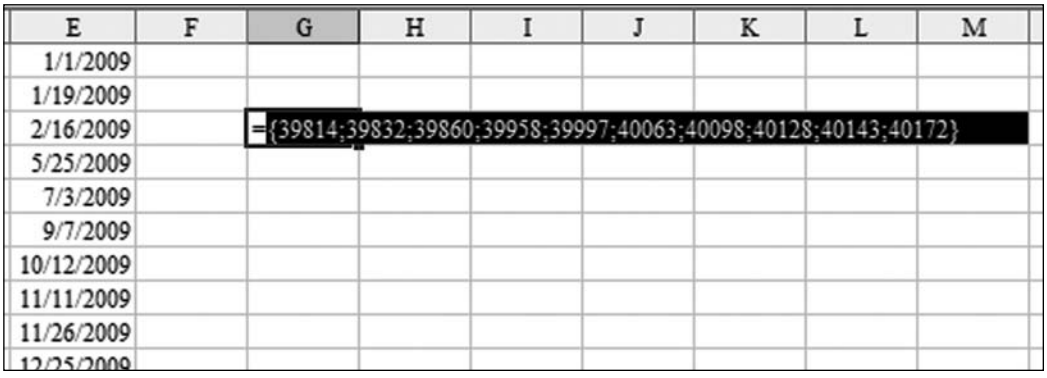
Source: http://www.mrexcel.com/forum/showthread.php?t=69761

STORE HOLIDAYS IN A NAMED RANGE

Challenge: The `NETWORKDAYS` and `WORKDAY` functions can take a list of company holidays as the third argument. If you store the list of holidays in AZ1:AZ10, there is a chance that someone will inadvertently delete a row, so you want to move the range of company holidays to a named range.

Solution: There is an easy way to convert the range of holidays to a named range. Follow these steps:

1. Type your company holidays as a column of dates in E1:E10.
2. In a blank cell, type `=E1:E10`. Do not press Enter. Instead, press the F9 key. Excel calculates the formula and returns an array of date serial numbers, as shown in Figure 6. Notice that everything after the equals sign is already selected.
3. Press Ctrl+C to copy the array to the Clipboard.
4. Press Esc to exit Formula Edit mode. The formula disappears.
5. Visit the Name dialog box. (In Excel 97-2003, select Insert, Name, Define. In Excel 2007, select Formulas, Define Name.)
6. Type `Holidays` as the name.
7. In the Refers To box, clear the current text. Type an equals sign. Press Ctrl+V to paste the array of dates to the box. Click OK.



E	F	G	H	I	J	K	L	M
1/1/2009								
1/19/2009								
2/16/2009		={39814;39832;39860;39958;39997;40063;40098;40128;40143;40172}						
5/25/2009								
7/3/2009								
9/7/2009								
10/12/2009								
11/11/2009								
11/26/2009								
12/25/2009								

Figure 6. Press F9, and Excel converts the range reference to an array of serial numbers.

Now you can use the named range `Holidays` as the third argument of the `WORKDAY` and `NETWORKDAYS` functions.

Gotcha: While these names work fine with `WORKDAY` and `NETWORKDAYS`, they fail in complex array formulas.

Summary: You can convert a range of dates to a named array to simplify the use of the `WORKDAY` and `NETWORKDAYS` functions.

SUM EVERY OTHER ROW OR EVERY THIRD ROW

Challenge: In Figure 7, someone set up a worksheet with dollars in rows 2, 4, 6, 8, and so on and percentages in rows 3, 5, 7, 9, and so on. You want to sum only the dollars, which are stored in the even rows. While you're at it, you'd like to know how to sum the odd rows or every third row.

Solution: There are a lot of possible approaches to this problem, some of which require you to figure out which rows to sum.

- To sum the odd rows: `=SUMPRODUCT(MOD(ROW(3:100),2),(C3:C100))`
- To sum the even rows: `=SUMPRODUCT(--(MOD(ROW(2:99),2)=0),(C2:C99))` or `=SUMPRODUCT(MOD(ROW(1:98),2),(C2:C99))`
- To sum every third row (2, 5, 8, etc.): `=SUMPRODUCT(--(MOD(ROW(2:148),3)=2),(C2:C148))`. See Figure 8

C100 fx =SUMPRODUCT(--(MOD(ROW(A2:A99),2)=0),(C2:C99))

	A	B	C	D	E	F
1	Customer	Info	Q1	Q2	Q3	Q4
93	Wonderful Briefcase Supply	GP %	49%	51%	46%	48%
94	Wonderful Glass Corporation	Dollars	14266	15842	10897	19081
95	Wonderful Glass Corporation	GP %	49%	53%	45%	47%
96	Wonderful Jewelry Company	Dollars	13274	14183	12042	11075
97	Wonderful Jewelry Company	GP %	48%	54%	50%	52%
98	Wonderful Scooter Corporation	Dollars	17934	19855	11155	15019
99	Wonderful Scooter Corporation	GP %	48%	46%	52%	40%
100			713681	767432	720601	733809

Figure 7. You want to sum the even rows.

Summing the Odd Rows

Think back to when you were just learning division. If you had the problem 38 divided by 5, you would write that the answer is 7 with a remainder of 3. Excel

provides the MOD function to return the remainder in a division problem. –For example, =MOD(7,2) calculates 7 divided by 2 and returns 1 as the remainder. The remainder of an odd number divided by 2 is 1. It is 0 for all even numbers. You can therefore use MOD to assign a 1 to each odd-numbered row and a 0 to each even-numbered row.

The problem is simple if you want only the odd rows. You can use an array of 1s and 0s in SUMPRODUCT. Multiplying the range C3:C100 by the result of the MOD function (an array of alternating 1s and 0s) results in every other number being added up.

Summing the Even Rows

The MOD(ROW(),2) function returns 1 for an odd row, and it returns 0 for an even-numbered row. Therefore, if the result of the MOD function is 0, you know you're working with an even-numbered row. Using MOD(ROW(),2)=0 will return an array of TRUE and FALSE values. You can then use the double minus sign to convert the TRUE/FALSE values to 1/0 values.

A simpler but less intuitive solution is to adjust the MOD argument so that it is one row behind the sum range. If you hope to grab the even rows from C2:C99, you can specify a range for the ROW function that starts one row above the real range. Use MOD(ROW(1:98),2) to ensure that the first value MOD returns is 1, followed by 0, 1, 0, 1, and so on.

Summing Every Third Row

Figure 8 shows a situation in which cost rows have been added. In this case, you would like to sum every third row—rows 2, 5, 8, etc. If you use =MOD(Row,3), you get 1 for rows 1, 4, and 7. You get 2 for rows 2, 5, and 8. You get 0 for rows 3, 6, and 9. To sum only the sales rows, you need to test if the result of the MOD function is a 2. Since this test will return True/False values, use the double minus to convert the True/False values to 1/0 values. So the formula becomes:

```
=SUMPRODUCT(--(MOD(ROW(2:148),3)=2),(C2:C148))
```

C149 fx =SUMPRODUCT(--(MOD(ROW(2:148),3)=2),(C2:C148))

	A	B	C	D	E	F
1	Customer	Info	Q1	Q2	Q3	Q4
142	Wonderful Glass Corporation	GP %	49%	53%	45%	47%
143	Wonderful Jewelry Company	Dollars	13,274	14,183	12,042	11,075
144	Wonderful Jewelry Company	Cost	6,902	6,524	6,021	5,316
145	Wonderful Jewelry Company	GP %	48%	54%	50%	52%
146	Wonderful Scooter Corporation	Dollars	17,934	19,855	11,155	15,019
147	Wonderful Scooter Corporation	Cost	9,326	10,722	5,354	9,011
148	Wonderful Scooter Corporation	GP %	48%	46%	52%	40%
149			713,681	767,432	720,601	733,809

Figure 8. You want to sum every third row.

Alternate Strategy: While all the solutions presented so far are going to amaze your co-workers, they are all inherently dangerous. If someone inserts a new row in the worksheet, the MOD functions won't work as you want them to.

It was not stated in the original problem, but if the worksheet really has a column B that identifies Dollars and GP%, then it would be safer to use a SUMIF function to sum the dollar amounts:

```
=SUMIF($B2:$B99,"Dollars",C2:C99)
```

This formula instructs Excel to look through B2:B99. If the value in that row says "Dollars", Excel adds up the corresponding value from column C. With this solution, there is no worry that dollars on even rows will accidentally shift to odd rows.

Summary: While you can guru-out with SUMPRODUCT solutions galore, the simplest solution might be to use SUMIF.

Source: http://www.mrexcel.com/forum/showthread.php?t=232025

WHY THE MINUS MINUS? COERCE NUMBERS FROM TRUE/FALSE

Challenge: While IF and other functions that expect logical tests can easily convert TRUE and FALSE values to 1s and 0s, the SUMPRODUCT function cannot do this. Why do you sometimes use a minus minus in SUMPRODUCT?

In Figure 9, for example, the SUMPRODUCT formula to calculate a 2% bonus for sales above $20,000 and with GP% above 50% fails:

```
=SUMPRODUCT((C4:C14>20000),(D4:D14>0.5),C4:C14)*0.02
```

If you simply build a SUMPRODUCT formula with your criteria and the numeric field, you end up with calculations such as TRUE * TRUE * 21000, which SUMPRODUCT incorrectly evaluates to 0.

=SUMPRODUCT((C4:C14>20000),(D4:D14>0.5),C4:C14)*0.02

	A	B	C	D	E	F
1	Pay a Bonus if Revenue is > 20000 and Profit % is > 50%.					
2						
3	Invoice	Rep	Revenue	GP%		Bonus
4	1010	Fred	24011	49.1%		0
5	1011	Bob	20489	48.8%		0
6	1012	Joey	19040	48.4%		0
7	1013	Joey	21000	55.0%		1
8	1014	Bob	19898	49.4%		0
9	1015	Bob	21818	50.5%		1
10	1016	Bob	19324	51.0%		0
11	1017	Bob	22207	50.7%		1
12	1018	Bob	18642	48.1%		0
13	1019	Fred	21602	48.3%		0
14	1020	Fred	22902	51.6%		1
15						
16		Bonus:	0			
17						

Figure 9. You would think Excel's Boolean logic rules could handle this.

In Figure 10, the first term of SUMPRODUCT has been evaluated. You see the array TRUE;TRUE,….

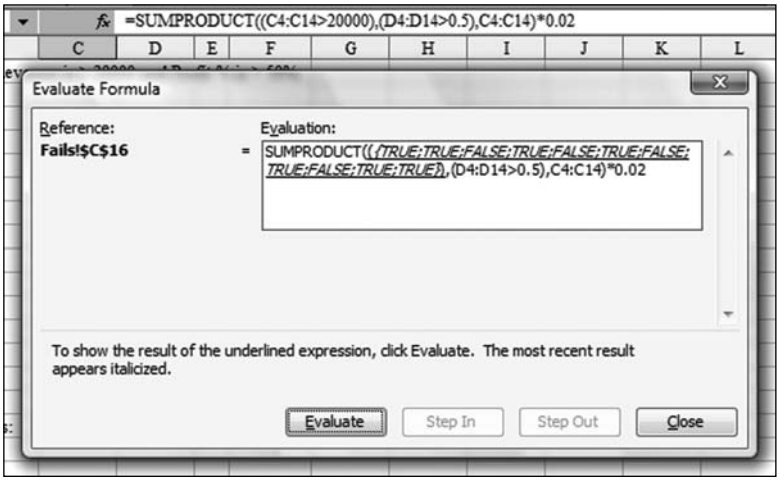


Figure 10. The SUMPRODUCT function does not deal well with TRUE * TRUE * a number.

Solution: You need a way to convert the TRUE/FALSE values to 1/0 values. Excel gurus use the minus minus in order to coerce Excel to change an array of TRUE/FALSE values to 1s and 0s:

```
--(C4:C14>20000)
```

As shown in Figure 11, this formula does the trick:

```
=SUMPRODUCT(--(C4:C14>20000),--(D4:D14>0.5),C4:C14)*0.02
```

=SUMPRODUCT(--(C4:C14>20000),--(D4:D14>0.5),C4:C14)*0.02

	A	B	C	D	E	F
1	Pay a Bonus if Revenue is > 20000 and Profit % is > 50%.					
2						
3	Invoice	Rep	Revenue	GP%		
4	1010	Fred	24011	49.1%		
5	1011	Bob	20489	48.8%		
6	1012	Joey	19040	48.4%		
7	1013	Joey	21000	55.0%		
8	1014	Bob	19898	49.4%		
9	1015	Bob	21818	50.5%		
10	1016	Bob	19324	51.0%		
11	1017	Bob	22207	50.7%		
12	1018	Bob	18642	48.1%		
13	1019	Fred	21602	48.3%		
14	1020	Fred	22902	51.6%		
15						
16		Bonus:	1758.54			

Figure 11. By using minus minus, you convert the TRUE/FALSE to 1/0, and the formula works.

Part I

Alternate Strategy: In fact, all the following operations also convert an array of TRUE/FALSE to an array of 1/0:

```
N(C4:C14>20000)
1*(C4:C14>20000)
(C4:C14>20000)+0
(C4:C14>20000)^0
```

You could multiply the criteria terms together, replace the comma with an asterisk, and let Excel perform all the logical tests. The formula to calculate the bonus would be:

```
=SUMPRODUCT((C4:C14>20000)*(D4:D14>0.5),C4:C14)*0.02
```

This syntax allows you to combine AND and OR logic. Say that you want to pay the bonus if both conditions are met or if the rep is Joey. You would add some parentheses and indicate that the bonus is also paid when the rep is Joey:

```
=SUMPRODUCT(((C4:C14>20000)*(D4:D14>0.5))+(B4:
B14="Joey"),C4:C14)*0.02
```

Figure 12 shows a formula that conditionally sums based on two AND and one OR criteria.

=SUMPRODUCT(((C4:C14>20000)*(D4:D14>0.5))+(B4:B14="Joey"),C4:C14)*0.02

	A	B	C	D	E	F	G	H
1	Pay a Bonus if Revenue is > 20000 and Profit % is > 50%, or if Rep="Joey"							
2								
3	Invoice	Rep	Revenue	GP%				
4	1010	Fred	24011	49.1%				
5	1011	Bob	20489	48.8%				
6	1012	Joey	19040	48.4%				
7	1013	Joey	21000	55.0%				
8	1014	Bob	19898	49.4%				
9	1015	Bob	21818	50.5%				
10	1016	Bob	19324	51.0%				
11	1017	Bob	22207	50.7%				
12	1018	Bob	18642	48.1%				
13	1019	Fred	21602	48.3%				
14	1020	Fred	22902	51.6%				
15								
16		Bonus:	2559.34					
17								

Figure 12. You can build the Boolean logic as one term of the SUMPRODUCT function.

Summary: To use logical tests in SUMPRODUCT, you can convert the TRUE/FALSE values to 1/0 values by using minus minus or other methods described in this topic.

Source: http://www.mrexcel.com/forum/showthread.php?t=221125 and http://www.mrexcel.com/forum/showthread.php?t=128907