EXCEL

By Bill Jelen

Extracting Matching Records with an Array Formula

Let's say that you have a database of invoice records and would like to create a reporting tool that will extract all open invoices for a specific customer. Normally, you would use an Advanced Filter to copy the records to an extract range. If the criteria change, you have to go back and run the Advanced Filter again. What if you could have a set of formulas that would automatically extract the matching records based on values in input cells?

You can solve this problem with a complicated array formula. Array formulas are often a mystery in Excel. Up until now, they have been poorly documented. Excel MVP Mike Girvin recently authored a book called *Ctrl+Shift+Enter* that documents how to leverage array formulas in Excel to do seemingly impossible tasks, including extracting records that match criteria. The formula in this column is based on Chapter 15 of Girvin's book.

In Figure 1, the original invoice data appears in A1:H564. You enter the selected customer and date range in cells J2:J4. Normal COUNTIFS and SUMIFS formulas in cells J6 and J7 summarize the open invoices for that customer and date range. The goal is to return all of the matching records to the extract range in cells I12:L30.

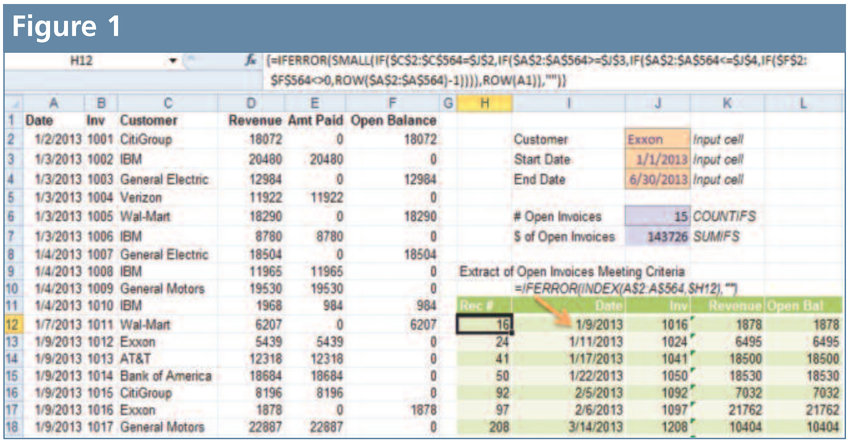
Girvin's array formula in column H returns the numbers of the records that match all of the criteria. We'll get back to that formula in a moment. First, let's review the INDEX function used to return the actual data in columns I:L after you've found the record numbers in column H.

Once you know a record number, you can use INDEX to return the value from that record. Most of the columns in the extract range will use the INDEX function to return the value from the original database. For example, the formula in I12 is essentially =INDEX(A$2:A$564,16). This returns the invoice date for the 16th record in the database. The 16 comes from the array formula in cell H12. Since there will be a different number of matching records based on the input cells, wrap the INDEX function in IFERROR to prevent #VALUE errors after the last matching record.

How the Array Formula Works

The magic is in the array formula to return the matching record numbers. Rather than tackling the complex formula in Figure 1, which has four criteria, the formula in Figure 2 returns the

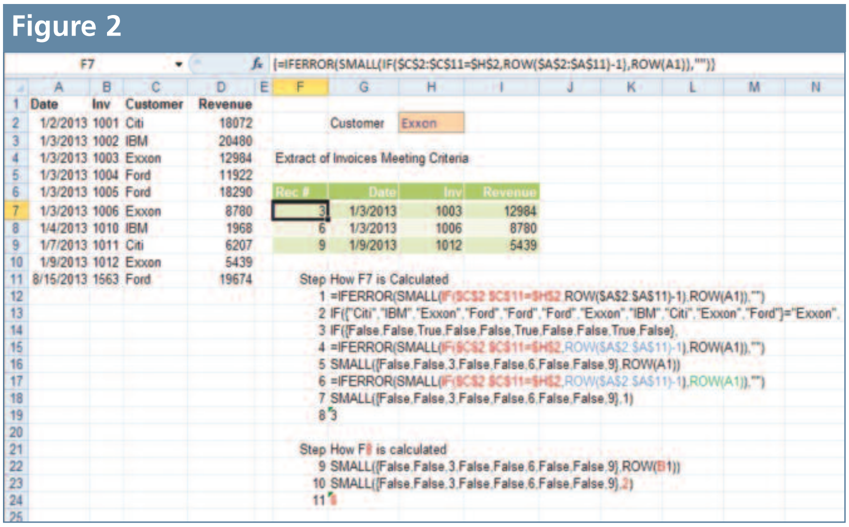
Figure 1



H12 {=IFERROR(SMALL(IF($C$2:$C$564=$J$2,IF($A$2:$A$564>=$J$3,IF($A$2:$A$564<=$J$4,IF($F$2:$F$564<>0,ROW($A$2:$A$564)-1)))),ROW(A1)),"")}

	A	B	C	D	E	F	G	H	I	J	K	L
1	Date	Inv	Customer	Revenue	Amt Paid	Open Balance						
2	1/2/2013	1001	CitiGroup	18072	0	18072			Customer	Exxon	*Input cell*	
3	1/3/2013	1002	IBM	20480	20480	0			Start Date	1/1/2013	*Input cell*	
4	1/3/2013	1003	General Electric	12984	0	12984			End Date	6/30/2013	*Input cell*	
5	1/3/2013	1004	Verizon	11922	11922	0						
6	1/3/2013	1005	Wal-Mart	18290	0	18290			# Open Invoices	15	*COUNTIFS*	
7	1/3/2013	1006	IBM	8780	8780	0			$ of Open Invoices	143726	*SUMIFS*	
8	1/4/2013	1007	General Electric	18504	0	18504						
9	1/4/2013	1008	IBM	11965	11965	0		Extract of Open Invoices Meeting Criteria				
10	1/4/2013	1009	General Motors	19530	19530	0			*=IFERROR(INDEX(A$2:A$564,$H12),"")*			
11	1/4/2013	1010	IBM	1968	984	984		Rec #	Date	Inv	Revenue	Open Bal
12	1/7/2013	1011	Wal-Mart	6207	0	6207		16	1/9/2013	1016	1878	1878
13	1/9/2013	1012	Exxon	5439	5439	0		24	1/11/2013	1024	6495	6495
14	1/9/2013	1013	AT&T	12318	12318	0		41	1/17/2013	1041	18500	18500
15	1/9/2013	1014	Bank of America	18684	18684	0		50	1/22/2013	1050	18530	18530
16	1/9/2013	1015	CitiGroup	8196	8196	0		92	2/5/2013	1092	7032	7032
17	1/9/2013	1016	Exxon	1878	0	1878		97	2/6/2013	1097	21762	21762
18	1/9/2013	1017	General Motors	22887	22887	0		208	3/14/2013	1208	10404	10404

Figure 2



F7 {=IFERROR(SMALL(IF($C$2:$C$11=$H$2,ROW($A$2:$A$11)-1),ROW(A1)),"")}

Date	Inv	Customer	Revenue
1/2/2013	1001	Citi	18072
1/3/2013	1002	IBM	20480
1/3/2013	1003	Exxon	12984
1/3/2013	1004	Ford	11922
1/3/2013	1005	Ford	18290
1/3/2013	1006	Exxon	8780
1/4/2013	1010	IBM	1968
1/7/2013	1011	Citi	6207
1/9/2013	1012	Exxon	5439
8/15/2013	1563	Ford	19674

Customer: Exxon

Extract of Invoices Meeting Criteria

Rec #	Date	Inv	Revenue
3	1/3/2013	1003	12984
6	1/3/2013	1006	8780
9	1/9/2013	1012	5439

```
Step How F7 is Calculated
  1 =IFERROR(SMALL(IF($C$2:$C$11=$H$2,ROW($A$2:$A$11)-1),ROW(A1)),"")
  2 IF({"Citi","IBM","Exxon","Ford","Ford","Ford","Exxon","IBM","Citi","Exxon","Ford"}="Exxon",
  3 IF({False,False,True,False,False,True,False,False,True,False},
  4 =IFERROR(SMALL(IF($C$2:$C$11=$H$2,ROW($A$2:$A$11)-1),ROW(A1)),"")
  5 SMALL({False,False,3,False,False,6,False,False,9},ROW(A1))
  6 =IFERROR(SMALL(IF($C$2:$C$11=$H$2,ROW($A$2:$A$11)-1),ROW(A1)),"")
  7 SMALL({False,False,3,False,False,6,False,False,9},1)
  8 3

Step How F8 is calculated
  9 SMALL({False,False,3,False,False,6,False,False,9},ROW(B1))
 10 SMALL({False,False,3,False,False,6,False,False,9},2)
 11 6
```

record numbers from a 10-row data set that match the Customer input cell (in this example, it's Exxon). The formula in cell F7 of Figure 2 is =IFERROR(SMALL (IF($C$2:$C$11=$H$2,ROW($A$2: $A$11)-1),ROW(A1)),""). Note that the formula bar in Figure 2 shows curly braces. You don't type the curly braces. Instead, type the formula and then hold down CTRL+SHIFT while pressing ENTER. This marks the formula as an array, and Excel automatically displays curly braces in the formula bar.

To evaluate the formula, Excel starts with the IF statement. It will check each customer in C2:C11 to see if it's equal to "Exxon" or not. Steps 1-3 in Figure 2 show how the IF statement returns a series of 10 values of True or False.

Each IF function typically contains arguments for value_if_true and value_if_false. If you leave out the value_if_false argument, the IF function will return a value of False. This works to your advantage later in the formula. In Step 4, the blue characters are the value_if_true, which is equal to the row number of each record. Remember to subtract the row number of the heading row. The -1 at the end of ROW($A$2:$A$11)-1 is because the headings are in row 1.

As Excel finishes evaluating the IF function, it will have a series of answers that are either False or the record number, as shown in Step 5 of Figure 2. When the record matches the criterion, you get the record number. When the record doesn't match the criterion, you get a False.

You've probably used MIN and MAX in Excel to return the smallest or largest value. SMALL and LARGE will also return the smallest or largest value, but they can also return the second smallest, third smallest, and so on: =SMALL(D2:D11,1) returns the smallest item, =SMALL(D2: D11,2) returns the second smallest item, and so on. Here's the trick that makes this formula work: SMALL ignores True/False values! When {False,False,3,False,False,6, False,False,9,False} is passed to the SMALL function, it only returns numeric values, such as 3, 6, or 9. In cell F7, you want to return the smallest numeric record number, so you would use 1 as the second argument. A common trick is to replace that argument with ROW(A1). When you copy the formula down a row, this argument becomes ROW(A2), which is 2. Steps 9-11 show how cell F8 returns the record number 6 as the second matching row.

Handling Multiple Criteria

In Figure 1, there are four criteria. Not all of the 400 functions in Excel will work inside an array formula. Girvin's *Ctrl+Shift+Enter* is the first book to document which functions work with array formulas and which don't. IF or SMALL will, but AND won't. While it would have been easier to use an IF and an AND function in the formula, the AND function simply won't handle arrays. That's why the formula in Figure 1 had to use four nested IF functions.

Array functions are incredibly powerful but hard to learn. Pick up a copy of Girvin's book to learn how these complex functions can make previously impossible tasks possible. **SF**

Bill Jelen is the author of Learn Excel 2007-2010 from MrExcel *and a frequent speaker at IMA® chapter professional development days. Send questions for future articles to IMA@MrExcel.com.*