

# Quick Tip: Crystal Reports' three-formula trick

Like any software package, [Crystal Reports](#) has its own set of pitfalls to avoid. I'll show you one of the handiest and most versatile tools you can have in your Crystal toolkit: the three-formula trick. If you've created a report but then had to filter at the group level, rather than at the record level, you may have noticed that your summary fields get all out of whack.

To show you what I mean, I've taken the top 20 highest-grossing movies of all time from movieweb.com, as shown in **Figure A**, and used them as the data source for my report.

Figure A

Movie	Takings (U \$Sm)
Titanic	\$601
Star Wars	\$461
E.T.	\$435
Star Wars: The Phantom Menace	\$431
Spider-Man	\$404
Jurassic Park	\$357
The Lord of the Rings: The Two Towers	\$340
Forrest Gump	\$330
The Lion King	\$329
Finding Nemo	\$328
Harry Potter and the Sorcerer's Stone	\$318
The Lord of the Rings: The Fellowship of the Ring	\$313
Star Wars: Attack of the Clones	\$311
Return of the Jedi	\$309
Independence Day	\$306
The Sixth Sense	\$294
The Empire Strikes Back	\$290
Home Alone	\$286
The Matrix Reloaded	\$278
Shrek	\$268
Total Takings:	\$6,989

If I filter out all movies that made less than \$300 million in the group selection, I get the results in **Figure B**.

Figure B

Movie	Takings (US\$m)
Titanic	\$601
Star Wars	\$461
E.T.	\$435
Star Wars: The Phantom Menace	\$431
Spider-Man	\$404
Jurassic Park	\$357
The Lord of the Rings: The Two Towers	\$340
Forrest Gump	\$330
The Lion King	\$329
Finding Nemo	\$328
Harry Potter and the Sorcerer's Stone	\$318
The Lord of the Rings: The Fellowship of the Ring	\$313
Star Wars: Attack of the Clones	\$311
Return of the Jedi	\$309
Independence Day	\$306
<b>Total Takings: \$6,989</b>	

Some of the movies are gone, but the Total Takings haven't changed. Crystal's automatic summary thinks the other movies are still there. You can fix this and many other summary problems by using three formulas.

Formula one: `takings_init`

The first of the formulas, shown in **Figure C**, defines the variable you'll use to add up the earnings.

Figure C

```
WhilePrintingRecords;
NumberVar takings:=0;
```

The first line, which should be default in all of your formulas, specifies that the formula will be run when the report is printing. This means it will be set after all filters and groupings have been applied to the data. The second line defines a new variable called `takings` and sets its value to zero.

Place this formula where you want the summary to reset. If you want your summary to be for each group, put `takings_init` in the group header. If you want it to be a summary for the entire report, put it in the report header. Remember to also suppress the field so the "0" does not appear in the final report.

Formula two: `takings_calc`

The second formula is the engine of the three. This sets the rules of the summary you want to

perform. In this case, I'm simply adding up the earnings as I move down through the report so the formula will look like **Figure D**.

Figure D

```
WhilePrintingRecords;  
NumberVar takings;  
takings:=takings+{movies_csv.Takings}
```

Again, I've used the WhilePrintingRecords statement. In fact, if you don't include this statement in all three formulas, you risk the summary not working at all. I've called the takings variable I set up in the first formula, and the third line specifies the rules of the summary. In this example, I'm simply adding all the takings, storing the value in my variable called takings.

This formula should be placed in the section where the field on which the summary is acting resides. In this and most cases, this will be the detail section. To see a running total, keep the formula unsuppressed; but if you are only interested in the final total, suppress it.

Other common rules that are used here are for a maximum, where the third line would be like this one:

```
takings:=maximum([takings,{movies_csv.Takings}])
```

This would be a minimum:

```
takings:=minimum([takings,{movies_csv.Takings}]) or a simple count of the number of movies:  
takings:=takings+1
```

Formula three: takings\_disp

The last formula displays the final result, as shown in **Figure E**.

Figure E

```
WhilePrintingRecords;  
NumberVar takings;
```

This is placed where you want the final result to display—in this case, the report footer. With the new formula placed to the right of the old one, you can see the difference in **Figure F**.

Figure F

Movie	Takings (U S\$m)	
Titanic	\$601	
Star Wars	\$461	
E.T.	\$435	
Star Wars: The Phantom Menace	\$431	
Spider-Man	\$404	
Jurassic Park	\$357	
The Lord of the Rings: The Two Towers	\$340	
Forrest Gump	\$330	
The Lion King	\$329	
Finding Nemo	\$328	
Harry Potter and the Sorcerer's Stone	\$318	
The Lord of the Rings: The Fellowship of the Ring	\$313	
Star Wars: Attack of the Clones	\$311	
Return of the Jedi	\$309	
Independence Day	\$306	
	<b>Total Takings:</b>	<b>\$6,989</b>
	<b>Total Takings:</b>	<b>\$5,573</b>

With these three formulas replacing the work of one automated summary field, the result is now accurate. Also, to add other summaries, you only need to add one extra display formula. The initialization and rule for the new summary can go in the existing formulas with a different variable used to store the value.

Crystal Reports is powerful but also has its eccentricities. If you find yourself filtering with the group selection but then discover that your summaries are going crazy, use formulas to restore the sanity. Even if you're filtering at the section level, by building the same filtering rules into your calculation formula, you can still keep your results consistent.

Another powerful way to use this trick is in subreports. By declaring the takings variable as a Shared NumberVar in all three formulas, you can have the initialization and display formulas in your main report while the calculation formula does the work in a subreport.