



Creating Diver Calculations



Oliver Gampe
Consultant
oliver@dimins.com



Topics covered in this session:

- A basic “Calc” in Diver
- Using functions
 - *Tearing strings apart*
 - *Elapsed time calculations*
- Show me everything, but only include...
- Nesting and complex “If” Functions
- Creating a goal or target column on the fly
- Cumulative columns



Comparing Excel and Diver Calculations

Calculations in Excel:

- Entered for an individual cell
- Begin with the equal sign
- Working with varied data is very flexible
- When using a different table in a different worksheet, any previously created columns have to be newly created or pasted into that worksheet again



Comparing Excel and Diver Calculations

Calculations in **Diver**:

- Entered for a column
- Do not begin with an equal sign
- Working with calculations is easier and faster when applying to a whole column
- Highly reusable and can (usually) be applied to any possible tabular that can be created from a Model
- Once a calculation is created and saved into a DivePlan, it will always be available when this DivePlan is used



Creating Calculations

1. Name
2. Definition
3. Model (Summary) Columns
4. Previously created Columns
5. Possible Columns from Info Fields
6. Functions
7. DimCounts
8. Named Groups

The screenshot shows the 'Add Column' dialog box with the following components:

- Definition** tab selected.
- Name & Definition** section: Name field contains '1.', Definition field contains '2.'. A calculator keypad is visible to the right.
- Other Columns** section: A list box contains 'Profit' with '4.' overlaid.
- Model Columns** section: Radio buttons for Total (selected), Min, Max, Average, % Total, and Std Dev. A list box contains 'Units', 'Cost', and 'Revenue' with '3.' overlaid.
- Info Columns** section: A checkbox for 'Display All' and a list box containing 'Count' and '% Count' with '5.' overlaid.
- Other Data** section: Three list boxes: 'Functions' (containing 'abs', 'and', 'arccos', 'arcsin', 'arctan' with '6.' overlaid), 'Dimension Counts' (containing 'Customer', 'Customer Zip', 'SIC Code', 'Sales Region', 'Salesperson' with '7.' overlaid), and 'Named Groups' (containing '8.' overlaid).
- Buttons: 'OK' and 'Cancel' at the bottom right.



Syntax

• Basic Column References

- <column type> [<column name>, *option*<limit 1>, ..., <limit x>]
- Examples: Total [Units], Calc [Forecast Units]

• Column Limit Filters

- <Dimension = "Dimension value">
- Example: Total [Units , Sales Region = "North"]

• Functions

- <function name> (argument 1, argument x)
- Examples: today (), abs (), substr ()



Tearing strings apart

`left()`, `right()` & `mid()` versus `substr()` & `scan()`

In **Excel**, string segments can be extracted with `left()`, `right()` and `mid()`, and often a `find()` is used to get the exact location of a separator character in the source string.

In **Diver**, `substr()` is used for simple, fixed character-length extractions, where the starting position (from the left) can be specified.

The `scan()` function allows a very flexible string segmentation based on separator characters in the string.



Tearing strings apart

Scan() - A Closer Look in **Diver**

scan(source string, segment number, segment separator)

The scan() function separates the source string into a number of segments based on the segment separator, and returns the segment defined by the segment number as counted from the left.

Example:

*scan("My favorite Product - #5662", 2, "#")
... returns 5662, the product ID*

The advantage of scan() is its sole regard for the separator and its disregard for the length of the segments.

In Excel: =RIGHT(A1 , LEN(A1) - FIND("#" , A1))



Tearing strings apart

Scan() - Some **Diver** Examples

Dimension	Dimension Value	New Column Definition	New Column Value
Patient Name	Smith, John	scan(dimension[Patient Name], 1, ",")	Smith
State	Braintree MA	scan(info[City], 2, " ")	MA
Item	3 Divers Merlot (017423)	scan(dimension[Item], 1, "(")	3 Divers Merlot



Calculations With Dates and Times

In **Excel**, given that the cells are properly formatted (or entered using the DATE function), dates and times can be simply added and subtracted.

In **Diver**, the same is true for dates and times, but they must be recognized as a number. (Dates can typically be type identified during Build.)

For handling date and time *strings*, Diver gives us two functions:

`format_date()` and `date_value()`

`format_time()` and `time_value()`



Elapsed Time Calculations

A Note about Timestamps

Given a string of this format (or similar):

02/13/2010 10:34:37

In **Excel**, it will be seen as a *timestamp*, that is an exact period of time of day, in this case, AM.

In **Diver**, this concept does not exist. Times are seen as an elapsed period of time from 00:00:00. The string above, identified as a date during Build, will become a date stripped of the time portion.

If the timestamp is in the Model as a string, then the various functions discussed here can be used to segment, calculate time, and format the string as needed.



Elapsed Time Calculations

A Note About Periods

Periods, most commonly months, are considered strings until identified in the DivePlan. These are not the subject of this discussion today.

The “Today” function

The function, `today ()`, uses the *sysdate* of the DiveLine host computer to return the numerical value of today's date.

Using Dimension Values

Frequently the date, or dates, of interest is one of your Dimensions. This date can be used in calculations if it is one of the Dimensions in the Dive Window (e.g. MultiTab) and the syntax, `dimension[date]` is used.



Elapsed Time Calculations

`format_date()` and `date_value ()`

`format_date(number, "date_format")`

This function converts a number representing days since 12/31/1899 into a date in the date format specified, e.g. "YYYY/MM/DD".

`date_value("date", "date_format")`

This function converts a date string in its existing date format into the number of days.



Elapsed Time Calculations

`format_time()` and `time_value ()`

`format_time(seconds, "time_format")`

This function converts the number of seconds into a given time format, e.g. "hh:mm:ss". This can be used to segment the time, i.e. just hours, just minutes or seconds, or any combination.

`time_value("time", "time_format")`

The inverse of above, this function converts a time string into the number of seconds.



Elapsed Time Calculations

Example 1

Init Date	Out Date	Elapsed Days	
Totals	Totals		
2010/01/01	2010/06/01	151.00	
2010/03/01	2010/06/01	92.00	
2010/06/01	2010/06/01	0.00	
2010/06/01	2010/06/02	1.00	
2010/06/02	2010/06/03	1.00	

Elapsed Days from now:

today () - dimension [Out Date]

Elapsed Days:

dimension [Out Date] - dimension [Init Date]



Elapsed Time Calculations

Example 2

Init Date	Init Time	Out Date	Out Time	Elapsed Time	Elapsed Days
Totals	Totals	Totals	Totals		
2010/01/01	01:44:34	2010/06/01	23:21:23	3645:36:49	151.88
2010/03/01	01:44:34	2010/06/01	23:21:23	2229:36:49	92.88
2010/06/01	01:44:34	2010/06/01	23:21:23	21:36:49	0.88
2010/06/01	03:44:34	2010/06/02	08:21:23	28:36:49	1.17
2010/06/01	23:20:34	2010/06/01	23:21:23	00:00:49	0.00
2010/06/02	03:44:34	2010/06/03	08:21:23	28:36:49	1.17

Elapsed Time:

```
format_time((dimension[Out Date]*86400 +  
time_value(dimension[Out Time], "HH:MM:SS")) -  
(dimension[Init Date]*86400 +  
time_value(dimension[Init Time], "HH:MM:SS")),  
"HH:MM:SS")
```

Elapsed Days:

```
value(scan(calc[Elapsed Time], 1, ":"))/24
```



Show me everything, but only include...

Sumif() versus Filtered Calcs

In **Excel**, you can create a cell containing the sum of only specific rows from a dataset using the sumif() function.

Example:

```
=SUMIF(F:F, "=Ice Cream", G:G)
```

This will return the sum of all cells in column G where the value in column F is "Ice Cream".

In **Diver**, Filtered or Limit Calcs can be used to get the same result.



Show me everything, but only include...

Filtered Calcs

Filtered Calcs allow you to show columns that are sums of different groupings next to each other.

The example from the previous slide translates to the following Filtered Calc in Diver:

```
Total[Revenue, Product Family="Ice Cream"]
```

A Calc can have multiple filters, so you could use something like this to get more detailed numbers:

```
Total[Revenue, Product Family="Ice Cream",  
Sales Region="West"]
```



Nesting and Complex "If" Functions

IFs, ANDs, and ORs - no BUTs!

When it comes to nested IFs and multiple conditions, Excel and Diver have many similarities.

The syntax in **Excel** to use the value from column x if conditions a and b are met, and the value from column y otherwise would look like this:

```
=IF(AND(V2="A", X2="A"), Z2, Y2)
```

This could mean that if, for example, both the Product Class and the Customer Class are "A", the Special Discount in Column Z is used, otherwise the Standard Discount in Column Y.



Nesting and Complex “If” Functions

IFs, ANDs, and ORs – no BUTs!

In **Diver**, the same functionality would look like this:

```
IF(AND(dimension[Product Class]="A",  
dimension[Customer Class]="A"), Special  
Discount, Standard Discount)
```

This means that if you are already familiar with Excel syntax for the IF, AND, and OR functions, you would create the calculations in Diver *the same way*.

It is worth noting that, as in Excel, both AND() and OR() are *functions* and their multiple arguments are a comma separated list within parenthesis.



Nesting and Complex "If" Functions

Nesting IFs

The IF function in Diver can be nested multiple times. Two examples:

1.

```
IF(condition1, IF(condition2, "True2", "False2"),  
"False1")
```

If the condition1 is met, condition2 is evaluated, otherwise the result will be "False1".

2.

```
IF(condition1, "True1", IF(condition2, "True2", "False2"))
```

If the condition1 is not met, condition2 is evaluated, if not true the result will be "False2".



Creating a target column on the fly

When it comes to adding columns with new data, Excel has the flexibility...

In **Excel**, there is almost no limit* to the new columns and the values, therein contained, that can be created.

Adding a target or goal column that is, for example, last year's revenue times a multiplier, can be done in a number of ways, depending on where the data is sourced.

*In Excel 2007 the limit is 16,384 columns



Creating a target column on the fly

...but Diver has the brains...

In **Diver**, “cells” are not individually accessible, but User Defined Dimensions (UDDs) are an easy way to add new layers. Creating a UDD adds a new Dynamic Dimension that contains values that can be edited directly in Diver.

Sales Region [demo_drs.mdl-Dive A]				
Sales Region	Sales Area	Units	Cost	Revenue
Totals	Totals	698,400	14,157,350.40	15,051,698.07
Boston	Area 1	201,507	4,114,284.34	4,388,322.27
North	Area 1	92,320	1,849,864.62	1,989,536.88
Rhode Island	Area 2	128,843	2,523,042.75	2,712,350.74
South Shore	Area 1	113,306	2,353,137.81	2,502,888.66
Southwest	Area 2	87,171	1,773,891.27	1,868,662.48
West	Other	75,253	1,543,129.61	1,589,937.04



Working with User Defined Dimensions

The UDD is stored in the DivePlan and can be used like any other Dynamic Dimension while diving. To access the value in a UDD, the UDD can be referenced through "*dimension[UDD Name]*" when it is a string column in the current window.

If the value should be numeric, it can be included in calculations by using:

value(dimension[UDD Name]).



Creating a target column on the fly

Using a UDD to create a Target Revenue column based on the previous year's revenue

1. Dive into the Dimension that will decide the target value, eg. Sales Region
2. Create a UDD that contains the multiplier that should be applied to the previous year's revenue
3. Close all windows, dive into the Dimension that you want to see the target for and MultiTab the UDD in.
4. Add the calc:

```
Total[Revenue,YearMo="Previous  
Year"]*value(dimension[Multiplier])
```



Cumulative Columns

Cumulative Columns

In **Excel** you can create a cumulative column with a dragged simple calculation based on a source column.

In **Diver**, to create cumulative columns that can be used in calculations, a different approach is necessary.

There is an option to display any column as a cumulative column. It is important to know this is for display only and all calculations are performed on the underlying original data. This option is not for calculations.



Cumulative Columns

Method to Create Cumulative Columns in Diver

1. Create Named Groups for all periods that the cumulative columns should cover. Each Named Group contains all periods up to that point.

The screenshot shows the 'Edit Named Groups' dialog box. It has a title bar with a close button. The 'List Group' dropdown is set to 'Apr. c'. There are 'Edit Group' and 'Remove' buttons. The 'Group Dimension' is set to 'Month' and 'Define Group By' is set to 'Values'. Below these are two list boxes: 'Values' (containing January through September) and 'Selected' (containing January through April). There are '>>' and '<<' buttons between the lists. An 'Exclude Values' checkbox is at the bottom right. At the bottom are 'Edit group and close' and 'Close' buttons.



Cumulative Columns

Cumulative Columns in Diver

- Next create Filtered Columns for all the periods:
e.g. *Revenue Feb, c* is the accumulated total of Jan & Feb sales

Name & Definition

Definition Name

Sales Region [demo_drs.mdl-Dive A]						
Sales Region	Revenue Jan, c	Revenue Feb, c	Revenue Mar, c	Revenue Apr, c	Revenue May, c	Revenue Jun, c
Totals	1,530,557.41	3,180,101.32	4,606,550.53	6,351,703.28	7,914,968.74	9,562,800.38
Boston	467,429.69	924,038.28	1,331,522.04	1,863,724.44	2,283,370.33	2,712,007.46
North	163,728.53	381,557.85	537,791.31	674,130.19	915,688.36	1,114,955.67
Rhode Island	311,543.01	575,613.30	873,486.17	1,216,903.61	1,513,382.69	1,872,090.15
South Shore	235,432.28	504,279.53	726,454.71	1,037,617.08	1,309,705.61	1,592,734.43
Southwest	222,731.53	452,571.17	601,977.68	806,726.96	1,005,091.79	1,239,794.35
West	129,692.37	342,041.19	535,318.62	752,601.00	887,729.95	1,031,218.33



Cumulative Columns

Cumulative Columns in Diver

Step 3. These columns can now be used as the basis for calculations, e.g. the Rolling Average Revenue per Unit based on the cumulative values for each month.

Whenever cumulated values are needed in calculations, you create the individual columns for each period and use those in the calculations. One year would require 12 columns.



Cumulative Columns

Sales Region [demo_drs.mdl-Dive A]						
Sales Region	Revenue Jan, c	Revenue Feb, c	Revenue Mar, c	Revenue Apr, c	Revenue May, c	Revenue Jun, c
Totals	1,530,557.41	3,180,101.32	4,606,550.53	6,351,703.28	7,914,968.74	9,562,800.38
Boston	467,429.69	924,038.28	1,331,522.04	1,863,724.44	2,283,370.33	2,712,007.46
North	163,728.53	381,557.85	537,791.31	674,130.19	915,688.36	1,114,955.67
Rhode Island	311,543.01	575,613.30	873,486.17	1,216,903.61	1,513,382.69	1,872,090.15
South Shore	235,432.28	504,279.53	726,454.71	1,037,617.08	1,309,705.61	1,592,734.43
Southwest	222,731.53	452,571.17	601,977.68	806,726.96	1,005,091.79	1,239,794.35
West	129,692.37	342,041.19	535,318.62	752,601.00	887,729.95	1,031,218.33



Topics covered in this session:

- A basic “Calc” in Diver
- Using functions
 - *Tearing strings apart*
 - *Elapsed time calculations*
- Show me everything, but only include...
- Nesting and complex “If” Functions
- Creating a target column on the fly
- Cumulative columns

