DateTime Functions

Introduction
In the world of analysis, “When” is an important and frequently asked question. Whether it’s estimating the time of a delivery, calculating a maturity date on an investment or even recording the time a workflow runs in Designer, you will need to execute calculations on DateTime data. The Formula tool’s function library includes a DateTime category for working with these values.

DateTime Data Types
DateTime functions can be applied to data classified as one of three types: Date, Time, and DateTime. The standard format for Date data is the four digit year – (dash) two digit month – two digit day. For Time, data should be formatted as a two digit hour : two digit minute : two digit seconds. As a combination of the two data types, DateTime combines both of these formats, separated by a space.

Dataset
A dataset contains information on the purchase date of a financial investment, the interest rate of that investment and the duration, in a varying number of months, until the investment reaches maturity. On which date will the investment reach maturity? And how many days must pass before that investment matures? Before writing expressions, ensure the values in the dataset are correctly classified so they can be used with DateTime functions.

Ensure the column [Purchase Date] is assigned an appropriate data type for DateTime calculations. In the Results Window, click "MetaData".

The column [Purchase Date] is classified as a DateTime datatype. DateTime calculations can be performed with these values.

Formula Tool on Canvas
After connecting a Formula tool to the Input Data tool, start to address the first question: on which date does each financial investment reach maturity?

Begin by creating a new column called "Maturity Date". Then, click Submit.

Since the output will be a date, select Date as the column's data type.

Click the Function Library. Expand the DateTime function category and select the function DateTime Add from the list.

The DateTime Add function requires three parameters: the DateTime column whose values serve as the initiation point, the time interval to add to the initiation point, and the unit of that time interval.

Drag and drop the values into the appropriate placeholder in the DateTimeAdd function.

The column [Purchase Date] contains the DateTime values used for the calculation.

Add a time interval by inserting a positive value into the placeholder, or subtract time with a negative value. Manually insert a value into the placeholder to add a constant time interval, or use values from another numeric column.

Units of years, months, days, hours, minutes or seconds can be added to a DateTime value. Specify the unit in quotes in the placeholder "u".

After running the workflow, a new column [Maturity Date] has been added to the data.

Question 2
As of today, how many more days are there until each investment reaches maturity? In other words, what is the difference between today's Date and the maturity date of each investment?

Open a new expression editor and create a column for today's date, entitled "Today", which has a "date" data type. To insert the current day, use the DateTimeNow function. This function has no parameters and is dynamic, meaning that it will return the current date any time the workflow is run.

Now, create another column, "Days to Maturity" which is numeric since it will output the number of days until the investment reaches maturity.

Expand the function library in the Expression Editor.

Click the DateTime function category. Then, select the function "DateTimeDiff" to insert it into the Expression Editor.

The DateTimeDiff function requires three parameters: two DateTime values and the units in which the difference in these dates should be returned.

In the placeholder for dt1, insert the variable [Maturation Date]. To find the difference between this date and today's date insert the column [Today] in the placeholder for dt2. Finally, enter the unit of time in which the diffence between the two dates should be calculated: days, in quotes.

The DateTimeNow() function is a powerful function that can introduce a dynamic element into DateTime calculations. This function returns the current date and time when the function is run to allow the values of "Today" and "Now" to be used in calculations. This function requires no parameters, and can be used to populate values in a column or as a DateTime value in an expression.

After running the workflow, the column [Days to Maturity] contains the number of days until each investment matures. Negative values indicate that the value in the placeholder dt2 is greater than the value specified in the placeholder dt1. In this case, the investment maturity date has already occurred. Positive values reflect maturity dates in the future; the value in the placeholder dt1 is greater than the value in the placeholder dt2.

DateTime Reporting

Sometimes, a DateTime value may not approriately communicate the information it contains. Perhaps the most meaningful information is the day of the week, or just the month of a returned value. Format DateTime values using a variety of specifiers to unlock additional information stored in each date and separators to customize outputs.

Designer supports the use of specifiers and separators to return string and numeric values from DateTime values. Specifiers are codes that combine a percent sign and case sensitive letter to output a specific information, such as the day of the week, the week number, or time on a 12 hour clock, to name a few. Separators include symbols like dashes, slashes and other punctuation to separate the values returned by the specifiers. Together, specifiers and separators can produce customized outputs that can be used for further analysis in a workflow, reporting and visualization.

In a new string column called [Reporting Output], use DateTimeFormat function to extract information from the column [Maturity Date] and match the desired format: Day, month, date, year, separated by commas and spaces.

Hover on the different specifiers and separators to explore the information they output. Click the values that can be used to create the desired format.

After wrapping the specifiers and separators in quotes and running the workflow, the column [Reporting Output] contains the user-friendly information as a string.