**1-1– Using Multiple Files & Calgary**

Data preparation and analytics can take hours on end when you do not have the proper tools. Alteryx can provide an easy fix to this problem using its analytic process automation platform which lets you build processes efficiently and enables you to run the program on a recursive schedule. By leveling-up your alteryx skills, you can unlock more of its potential such as applying dynamic actions, using regex, and applying ML models to name a few.

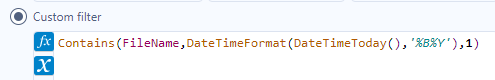
This course will help you understand and apply advanced toolsets in Alteryx. Each module will present slides to show descriptions and other critical information, in addition, it will also demonstrate how each tool works according to a specific scenario. Make sure that you already have the basics down pat before proceeding with this advanced alteryx course.

When working with files as input, its possible to have multiple records on different folders. We have discussed in the beginner course that alteryx can take in multiple files with the same schemas using the input tool. But what if you wanted to filter a bunch of files and take in only the most recent update? Or search the subdirectories for the file you need? In these cases, we can utilize the combination of directory and dynamic input tool. Let’s look at each tool in depth.

**The Directory Tool returns a list of all the files in the specified directory. The tool returns file names and other information about each file, for example, file size, creation date, last modified date, and more.**

In a new workflow, navigate to the In/Out tool tab and drag Directory to the canvas. There are 3 configurations for this tool. First is the directory that you wanted to check for file contents. You can either type the exact path into the textbox or use the file browse to navigate to the desired location. In this example, we are going to map the path to the parent folder 1-1. A shortcut for this is to simply drag the parent folder from your file explorer into the canvas which automatically create a directory tool.

Next, for file specification, we need to indicate the type of file or use wildcard characters for a specific filename. For example, we wanted to get all csv files with “SALES-“ on its filename, so under file specification we need to type “SALES-\*.csv” Asterisk is the wildcard character that signifies a whole string, aside from that you can also use “?” to signify a single character. Do take note that you can only use wildcard characters on the File Specification but not on the directory path. The last configuration is for enabling the subdirectory, meaning alteryx will look inside other folders with the parent directory path to search for the file. Running this workflow now without checking this option, will render us with no results, since the files we need are inside the year subdirectory. Tick the checkbox and run the workflow again to see the difference.

The output of the directory lists the full path, filename, date, size and attributes of the files that matches our file specification. We can narrow down the list by using a filter to set a condition for the file we need. From the preparation tool tab, drag a Filter tool and connect it to the output anchor of our directory. Set it to Custom Filter. We want to get the Sales file which contains the month and year during the date of runtime, which would be the “SALES-june2022.csv” within the 2022 folder. Use the expression “Contains(FileName,DateTimeFormat(DateTimeToday(),'%B%Y'),1)” This converts the current date into whole month name and year string, then proceeds to use this date to look at any filenames that contains it.   


**The next tool is the Dynamic Input Tool. It is used to read from an input database at runtime and dynamically choose what records are read in.**

On our workflow, navigate to the developer tool tab and find Dynamic Input. Drag the tool into the canvas and connect it to the T output anchor of the filter. The first configuration is for the Input Data Source Template which is a requirement as a standard schema of the files that will be read. Click edit to open a new window that lets you add a new data source or map the file for your template. The configuration on this new window is same to that of an input tool. For this example, we are going to set it to our sales template csv file inside the template folder. The files you are going to extract should have the same fields as that of your template. Once the template has been set, close the window to proceed with tweaking the tool. The next configuration lets you select how the input data sources will be updated during run time. You can either “Read a list of Data Source” or “Modify SQL Query”. **Read a list of Data Source is used if you already have the list of files ready in a column such as our current workflow, while Modify SQL Query is used if you wanted to minimize the records pulled from your data source using a SQL query.** Select “Read a List of Data Sources” since we already have the filenames upon using the directory tool. For the field, choose “FullPath” and set the action to “Change Entire File Path”. Since we are using CSV inputs, the field “FullPath” already contains the necessary data which is the complete file path and filename. But, if you are planning to use excel, make sure to concatenate the Sheet Name into the filename or full path.

Run the workflow to see the results. The output contains the complete 1000 records from our Sales-June2022.csv.

According to Alteryx Documentation, **Calgary is a list count data retrieval engine designed to perform analyses on large scale databases containing millions of records. In order to quickly retrieve records, Calgary uses indexing methodology which significantly improves the speed of data retrieval operations on a database table**. **Indexes** can be created using one or more columns of a database table, providing the basis for both rapid random look ups and efficient access of ordered records.

**In order to create a Calgary Database, you need to load the data into a Calgary Loader. It creates a Calgary database from any type of input file. A Calgary database is limited to 2 ^31 or approximately 2 billion number of records. Each field contained in the input file can be indexed to maximize the Calgary database performance.**

Let’s use a Calgary Loader to create a database from this excel superstore file. From the Calgary tool tab, drag Calgary Loader to the canvas and connect it to the output anchor of the Select tool. The first configuration we need to set is the Root File Name or the Calgary database name and its location. Click the file browse icon and navigate to your desired output folder. All Calgary databases has the file extension “.cybd”. In this workflow, we are going to save call this database as “superstore” and save it under our 1-1 module folder. Upon configuring the name and location, you now have the option to either “Load base CalgaryDB and/or 1 or More Standard Indexes” or “Load a Single Advanced Index”. Since we are starting from scratch, we wanted to create the base Calgary Database first so we are going with the first option for our first run.

The list box contains the fields you can add to the database. If you wish to exclude a certain fields from the database, you can uncheck the data checkbox next to its name. You can also deselect or select all using the options dropdown. Next to the Data checkbox is the Index. Check the index box of the fields that you need for queries. It is advisable to only Index the fields that you usually use for queries such as categorical columns like Segments and Countries because the more index you add the longer it takes for alteryx to create them. For this example, we are going index [Order Priority], [Region], [Customer Segment], [Customer Name], [Product Category] and [Product Sub-category]. You can manually set the Index Type to High Selectivity; for fields with very unique records like customer name, or Low Selectivity; for fields with repetitive data like [Order Priority]. The auto mode lets Calgary Loader tool automatically applies the proper index mode (High or Low Selectivity) based on the data contained in the Input field. All fields with more than 550 unique values are set to high selectivity. Run the workflow to create the Calgary DB.

If you check the output from file explorer, the workflow created 7 different files. 1 cydb for the database, 5 cyidx for each index and a single xml file.