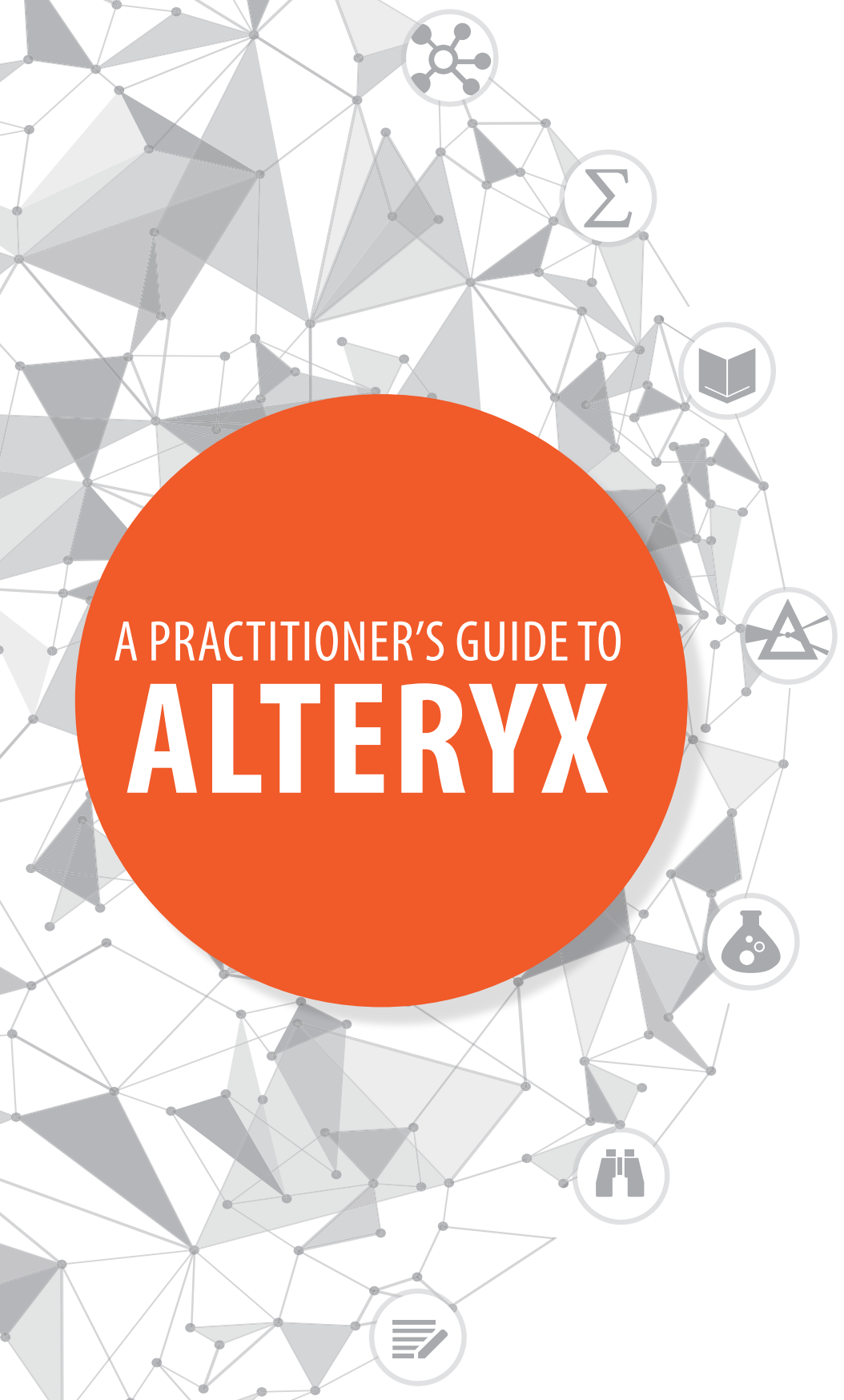


USEREADY

A PRACTITIONER'S GUIDE TO ALTERYX

Alteryx 10

A Practitioner's Guide to Alteryx®

Published by
USEReady
469 7th Ave Suite #745, New York, NY - 10018
www.useready.com

ISBN: 978-0-692-44795-6
Manufactured in the United States of America

About USEReady

USEReady is a leading provider of advanced business intelligence, data management, and network security solutions. Our mission is to help businesses succeed by fast-tracking their business performance - through our digital world ready solutions, industry and technological expertise, agile global delivery practices, and our customer fanaticism.

Alteryx Solutions

Being their long-term partner, we understand Alteryx like no one else. Staying ahead of the rapidly evolving data-scape, USEReady has developed a specific, unified and democratized approach to data management to ensure quality, accelerated decision-making and collaboration. From profiling to optimization, we leverage Alteryx's powerful capabilities to address the biggest challenge faced by analysts today – how to access the right data quickly.

Our Alteryx expertise includes:

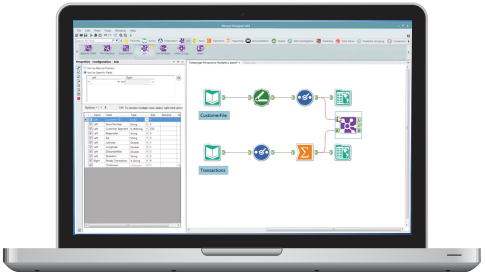
- Domain specific data preparation
- API-driven Extract-Transform-Load
- Reusable data preparation applications
- Analytics-oriented prepared answer sets
- Cloud and on-premise integration
- BI-ready modern data warehousing

Alteryx Training

We aim to give you the best chance of understanding the capabilities of Alteryx and make analytics interactive and explorable. You can choose from: virtual class-room or instructor-led, on-site format; as well as from Beginners, Intermediate or Advanced Training in Alteryx. We also offer customized Alteryx courses.

Our training will enable you to:

- Access, clean and blend data from multiple sources
- Join data sources of different types
- Create workflows, macros, and applications
- Prepare and deliver data sets for visualization
- Perform advanced predictive and spatial analytics



USEREADY

For more information, visit us at: **www.useready.com**,
or write to us at: **contact@useready.com**

tableau SERVICES PARTNER OF THE YEAR 2015 USEREADY

RED HERRING N. AMERICA 100 WINNER

AMERICA'S FASTEST-GROWING PRIVATE COMPANIES #113 Inc. 5000 2015

About Alteryx, Inc.

Alteryx, Inc. is the leader in data blending and advanced analytics software. Alteryx Analytics provides analysts with an intuitive workflow for data blending and advanced analytics that leads to deeper insights in hours, not weeks, which is typical of traditional approaches. Analysts love the Alteryx Analytics Platform because they can deliver deeper insights by seamlessly blending internal, third party and cloud data; and then analyze it using spatial and predictive drag-and-drop tools. This is all done in a single workflow, with no programming required. More than 1,000 customers and thousands of data analysts worldwide rely on Alteryx daily.

Visit www.alteryx.com or call 1-888-836-4274.

Acknowledgements

When we started writing this edition of the book, we decided to go with a team of authors instead of a single author. The team went through a rigorous iterative process of reading, updating and reviews to get the book where it is today.

I would like to express my gratitude to the team of authors and other colleagues who helped in making this book a reality. Without them the book truly would not be what it is today.

Uday Hegde: For being the driving force and a constant motivator.

Anand Vadul: For sharing the experiences and challenges faced in the last edition and being the technical guiding force.

Vijai Narasimha: For being one of main drivers of the project and putting in those long hours updating and reviewing the content.

Abhilash Ramanathan: For being a leader and helping the team with a very fine attention to detail.

Vijay Gowtham: For putting up with meetings at odd hours and being that much needed funny bone during our meetings.

Rahul Upadhye: For putting in endless hours of work in editing the book and always being committed.

Ashish Tergaonkar: For his enthusiasm in editing the book and his constructive suggestions during the meetings.

Rima Upadhye: For being involved despite a very busy schedule.

Prashant Singh: For creating marketing, a web presents, and an excellent cover design.

- Jagdeesh Narayanan

Letter from the CEO

First of all, we would like to thank our readers for making the first edition a grand success that we are super encouraged to attempt this edition. This edition is also a testament to growing popularity of Alteryx and appetite for self reliant solutions in the industry today.

We took a little bit of gamble with this edition. We invited a team of Alteryx practitioners within USEReady to contribute while enhancing the content with lot of experience and gotchas we learnt from the first edition. Jagdeesh Narayanan, Abhilash Ramanathan, Vijai Narasimha, Rahul Upadhye, Rima Upadhye, Anand Vadul, Vijay Gowtham, Ashish Tergaonkar and Prashant Singh have worked countless hours in shaping this book. The result has been phenomenal as you glean from the chapters.

Team has gone great heights to ensure the content is refined to the most recent version of Alteryx and exercises are useful to a fellow practitioner. With your valuable feedback we plan to augment the content to facilitate class-room like trainings.

We are grateful to our readers of the first edition and their constructive feedback has helped us improve this edition. We hope our efforts are well worth it and you are going to find this book useful.

Uday Hegde
Chief Executive Officer
USEReady

Foreword

As Vice President of product management at Alteryx, Inc. working closely with our product team, our customers and our partners, a training manual from USEReady is testament to the growing demand for easy-to-use data blending and advanced analytics solutions. USEReady knows the analytics industry and has created this book as both an independent guide and as a classroom aid to help its customers and others not only quickly learn Alteryx products, but more importantly grow in their ability to help gather deeper insights from their data.

The book has created a great launching point for beginners who want a manual, in addition to the real world use cases, so you can easily learn how best to use Alteryx Designer. By the time you finish working through this manual, someone who has never opened the Alteryx Designer before, will be able to create workflows, design reports, develop applications, and write macros to solve any of their data needs.

These are exciting times as the Alteryx community continues to grow globally, and we continue to witness an unprecedented demand for data analytics with actionable information. The Alteryx Designer and materials like this manual help the community of self-service data analysts make the most of their data.

We appreciate the partnership with companies like USEReady that know how to help these self-service analysts and reduce the time to insight with Alteryx.

Laura Sellers
Vice President, Product Management of Alteryx

Preface

Every day we are faced with options, questions, and choices. These decisions, as we all know, are much easier to make when when we are well informed. Let's say that we want to eat. We litterally have an entire world of possibilites, given the proper resources. However, practically, there are real limitations. Are we at home without transportation? Are we backpacking in the mountians? Are we in the middle of Times Square? Do we have food restrictions for health reasons? Do we have $5 or $5000? What are we in the mood for?

This task that we all solve day in and day out depends on a considerable amount of information that we know about our world, and often take for granted. This information is all based on data about our world.

What Is Data?

Data is stored information. It comes in various forms ranging from the number and types of items on our desk, to the total mass of the universe, to the contents of this book, to the information in digital files and systems, which will be our focus.

What Does Data Do?

Data does nothing. It simply exists. It is what we do with data that is important. When we look at data we interpret it to create meaningful information, which gives us the ability to make better informed decisions.

How Do We Consume Data?

Data can be consumed in many forms. We can look at all of the raw data and read every piece individually. We can use aggregation methods to create summary data so that we can

easily see high level trends. We can visualize the data because a picture truly is worth a thousand words. Since we often do not want to look directly at the original data source and read each individual piece of data, we need to perform data preparation.

What Is Data Preparation?

Data Preparation is the process by which raw data is converted into a clean, usable source for later consumption.

The three core components of data preparation are data retrieval, data manipulation, and data exportation. In more traditional analytic terms, data preparation refers to the extract-transform-load process referred to as ETL. However, in order to ease communication, we are going to avoid these technical terms and discuss the aspects of the processes as follows:

Data retrieval refers to the process of going to a data source, asking for data, and returning with the desired data.

Data manipulation refers to anything we decide to do to the data between the time we retrieve it and the time we export it.

Data exportation refers to what we do with data after we have extracted and manipulated it even if we haven't finished transforming it.

What is Data Manipulation?

Data retrieval and exportation are fairly straightforward; respectively, they can be likened to drawing water from a well and putting an ice cube in someone's drink. However, data manipulation is that tricky process of running the water through the pipes into our house, then filling the ice cube tray, then putting the tray in the freezer, and letting the water have enough time to freeze so that we have ice to consume. Going forward, we

will be using an allegory to a river to explain the entire process of the data preparation and specifically the data manipulation portion.

Data manipulation can come in many forms which typically fall into three buckets as follows:

Combination

One of the most common problems with data is that it comes from multiple sources. It is generally possible to perform the analysis separately, or through a significant amount of manual effort, but these methods often leave something to be desired or are too slow for effective use. In order to solve this problem, we will be designing data streams that come together.

If we think about data streams as actual rivers, original data locations can be thought of as glacial streams, smaller rivers, or lakes. Bringing data together is like the tributaries that bring these different water sources together to form a river. Along the course of this river, way we can perform calculations.

Calculation

If the data is to be used, it is generally advisable to have as much data pre-calculated as possible. One reason for this is that it allows an organization to create a standardized formula for everyone's use. Another is that when we can run calculations before data is provided to a front end user or system, the consumer will be experience a much faster process.

If the data is to be used in a report, then the calculations are often fundamental aspects of that report.

Returning to the river analogy, we can think of calculations as hydroelectric dams along a river, we are using the resources that already exist in order to generate something new.

We may change the landscape, because we are changing the flow of water, and we are also slowing down the river (introducing calculations will slow down the data preparation process).

We also have the ability to transform the data stream into a more useable format.

Transformation

It is often the case that data is not in the format that we need. We may have been given access to a database that has data stored in a very machine readable format, and we need to pivot the table to make it human readable, or we may have been working with an Excel file which has data extremely normalized that makes it hard to use in a front end system. Either way we need to transform the structure of the data so that it can be effectively consumed. In thinking about the river, we can imagine this as the process of cutting a channel into the riverbed so that the river is deep enough to move barges up and down. In doing this, we are fundamentally changing the structure of the river in order to make it more useable.

To the reader

In the following chapters, we are going to cover many topics, but the format of the chapters will all be the same.

You will assume the role of a new consultant at a company that works with Alteryx. We introduce a business scenario, discuss the tools that we will use to solve the problem(s), walk through the initial problem(s), and then provide you a self-guided exercise. We conclude this book with a capstone assignment in *NYC*.

The exercises will use data that can be download from http://www.useready.com/resources/a-practitioners-guide-to-alteryx by following the instructions on the website to unpackage the file.

Additional data will also be needed to install the *US 2010 Census SF1* and *USGS North America Map* packages from http://downloads.alteryx.com/data.html which we will start using in *Cultural Musings*.

Let us know what you think by emailing us at AlteryxBook@USEReady.com and we will try to incorporate reader requests going forward.

If interested in Alteryx training sessions or Alteryx consulting, visit http://www.useready.com/.

Best of Luck,

USEReady

Table of Contents

CHAPTER 1
An Introduction to Alteryx

1.1 What Is Alteryx Analytics Platform?

The Alteryx Analytics Platform from Alteryx Inc. consists of two products: Alteryx Designer and Alteryx Server.

Alteryx Designer allows us to build a data stream (or module) and run it locally. In addition to a base package, Alteryx offers datasets and tool packages to expand the product to enhance computing capabilities. These packages permit additional features like drive-time analysis when TomTom data is available.

Alteryx Server has two main functions. The first allows users who have Alteryx Designer to publish applications for consumption. The second allows scheduled workflows to run without external intervention.

The *Alteryx Analytics Platform* has solved two major issues affecting the data preparation process: time to run the process and high specialization of labor. Typically, the preparation process is highly time-intensive because it is largely manual. While automation solves the time required to run, it still requires personnel with skills in computer programming. This means we need additional resources allocated to every project, eventually raising the project costs and often preventing projects from being taken on. Alteryx, Inc. has elegantly addressed both of these problems. It has designed an easy-to-use drag-and-drop interface where all we need to do is give each tool the appropriate settings. This simple concept has allowed business users to develop workflows that meet their exact needs much faster than has ever been possible because it provides business people with the tools to design or modify a fully or semi-automated data preparation.

1.2 Alteryx Gallery

The *Alteryx Gallery* is a public version of *Alteryx Server* where the community can upload interesting data streams, share ideas, and get inspired. The *Alteryx Gallery* can be accessed at gallery.alteryx.com.

1.3 The Alteryx Interface

By the end of the chapter, the reader should be able to understand all aspects of the Alteryx Designer Interface and how to work with the product.

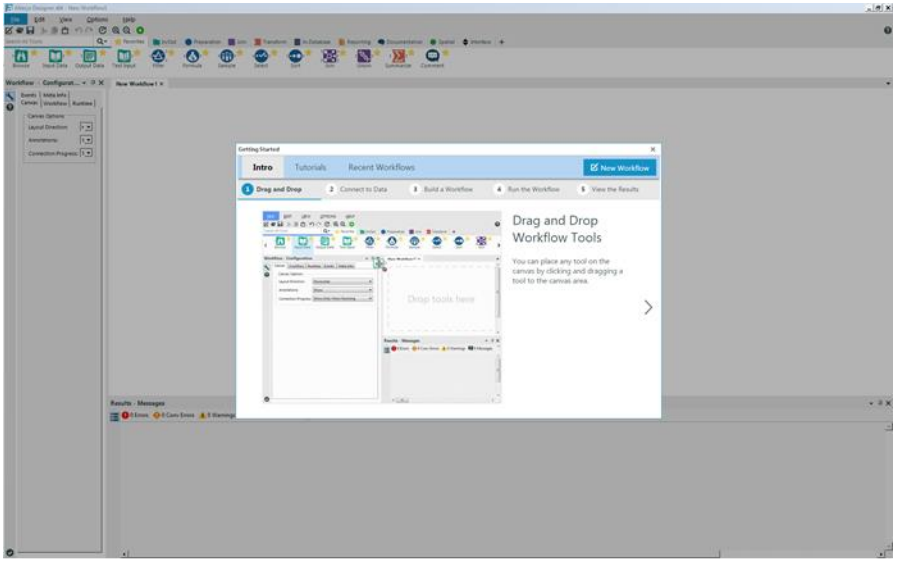


Figure 1-1 - Alteryx Intro

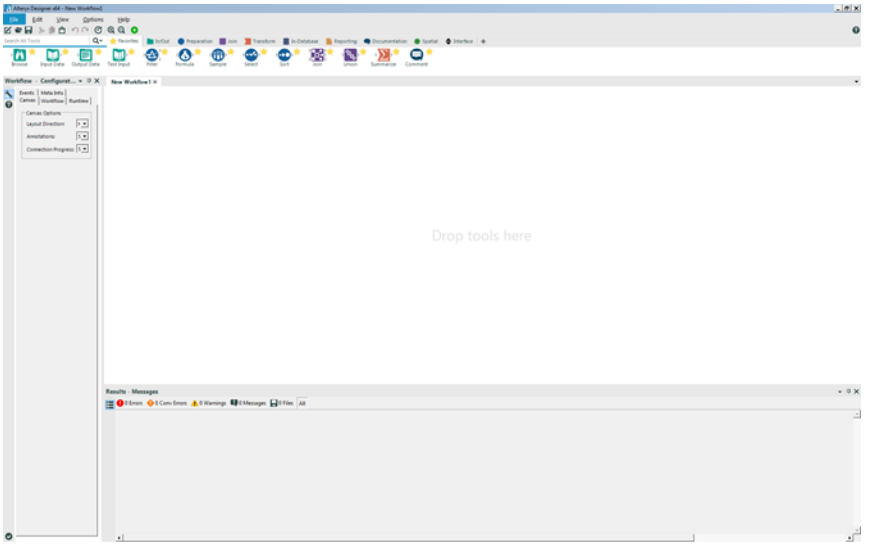


Figure 1-2 - Workflow Configuration

Above are the two things we may see when we open Alteryx Designer. The first image is the default for when the program is first opened. Alteryx Designer asks us what we want to do with the session. These options, fairly clearly, are to open up the basic tutorials, open an existing module, or open a new module. The second image is what comes up when we have previously selected the option *Don't show this again and always open a blank canvas*. When using this option, Alteryx Designer always opens a new module. (For those who see the first image, please select "New Workflow.")

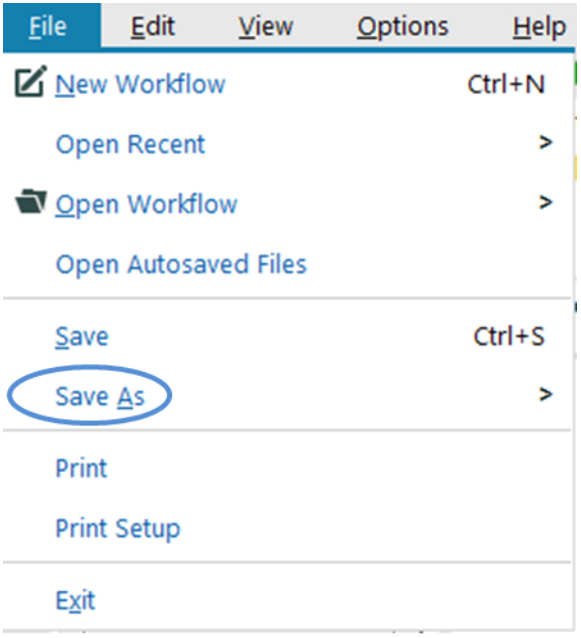


Figure 1-3 - File Dropdown

The first thing we will do is save our module. Open the file drop-down at the top left of the screen, and select the *Save As* option as indicated above.

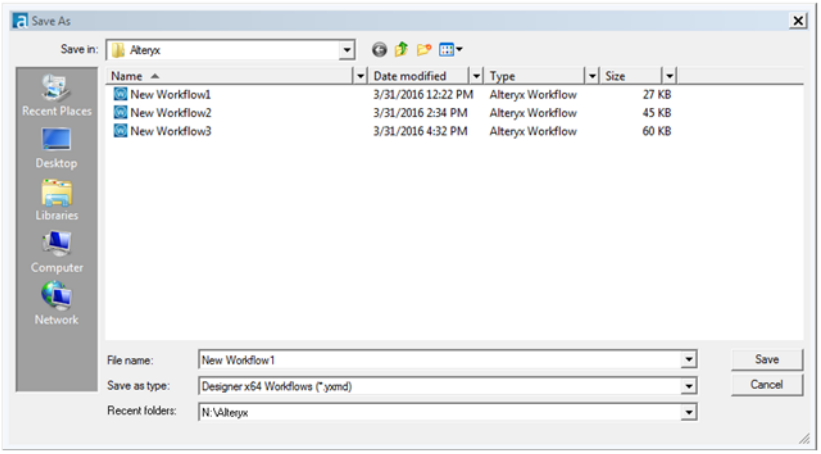


Figure 1-4 - Save As

We can save our data stream as one of three file types. At a high level, we have the ability to save our data streams as:

- .yxmd
  Files saved in this format are data streams in a standard *Alteryx Workflow.*

- .yxwz
  Files saved in this format are data streams that have been designed as *Analytic Applications*, which means they have user inputs and can be run on *Alteryx Server.*

- .yxmc
  Files saved in this format are data streams that have been designed as *Macros* that allow the reuse of the data stream.

Please save this workflow as *The Basics.yxmd*.

1.4 The Interface

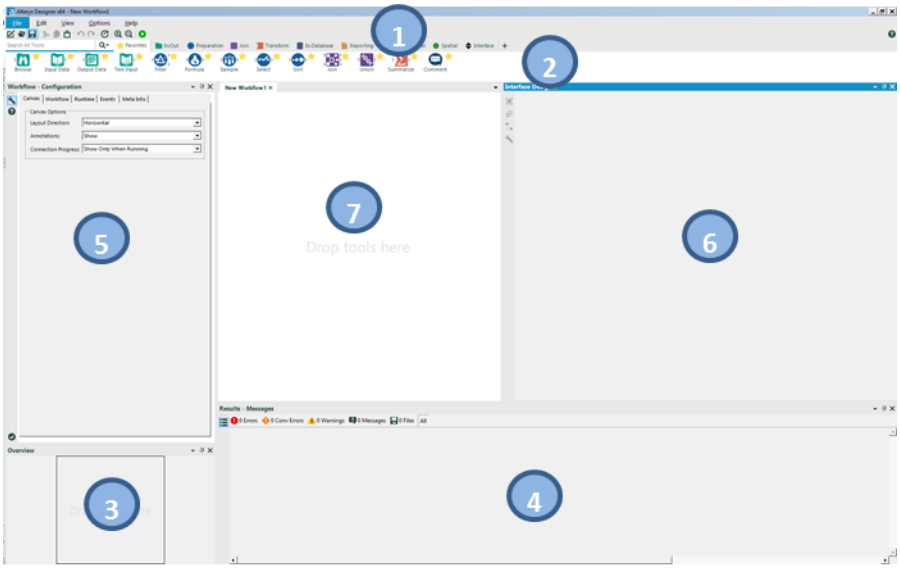


Figure 1-5 - Alteryx Designer Interface

This is one potential view of the *Alteryx Designer Interface*, if every *View* option is on. The numbers represent each of the seven primary aspects of the interface. One through six can be disabled and will be discussed under the corresponding sections in *View*. The seventh is the canvas, where we build workflows.

1.5 File Menu

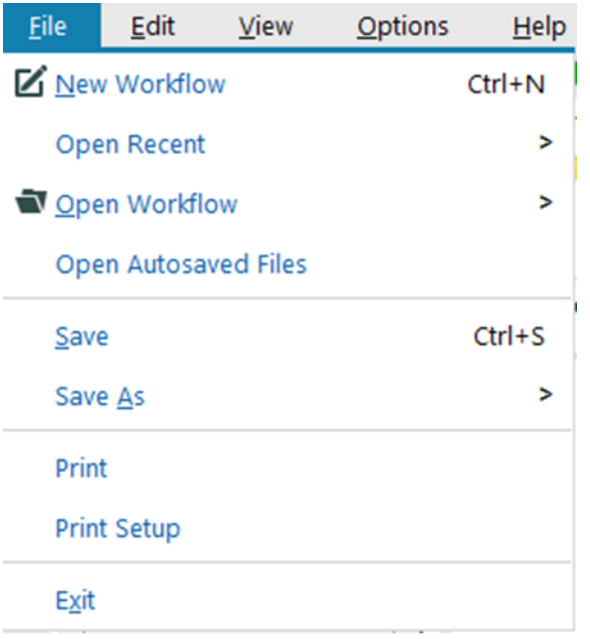


Figure 1-6 - File Menu

The *File Menu* functions in the same way as most programs, with a few specialized options:

New Workflow: Opens a blank canvas to work in. (*Ctrl+N* means that there is a *Hot Key* that allows us to use this action without opening the drop-down. Pressing the *Ctrl* and *N* buttons at the same time does the same thing as selecting *New Workflow*.) (Hot Key: Ctrl+N)

Open Recent: Opens a list of the most recently opened workflows.

Open Workflow: Opens a file browser that allows us to open an existing *Workflow*, *Analytic App*, or *Module* in a tab of the canvas. (Hot Key: Ctrl+O)

Save <*File Name*>: Saves the workflow that is in the active canvas. (We will see the name of the current workflow instead of <*File Name*>) (Hot Key: Ctrl+S)

Print: Prints a copy of the workflow.

Print Setup: Opens a window that allows us to configure how the workflow will print if we choose to print out a copy.

Exit: Closes Alteryx Designer entirely. This differs from *Close* because *Close* will close the active canvas but *Exit* will close every open data stream.

1.6 Edit Menu

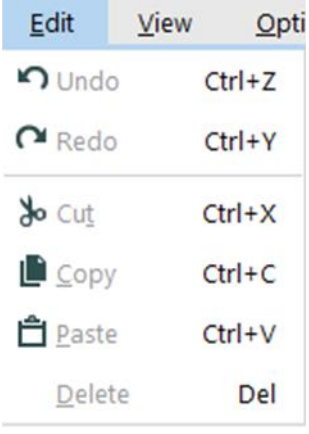


Figure 1-7 - Edit Menu

Undo: Allows us to undo the steps just performed, in reverse order. (Hot Key: Ctrl+Z)

Redo: Allows us to restore a step from what was undone, as long as we have not changed anything else. (Hot Key: Ctrl+Y)

Cut: Allows us to remove the selected items on the canvas while keeping a copy on the clipboard. (Hot Key: Ctrl+X)

Copy: Allows us to copy the selected items on the canvas to the clipboard. (Hot Key: Ctrl+C)

Paste: Allows us to take the most recent thing copied or cut into the clipboard and put it on the canvas. (Hot Key: Ctrl+V)

Delete: Removes the selected items on the canvas. (Hot Key: Del or Delete)

1.7 View Menu

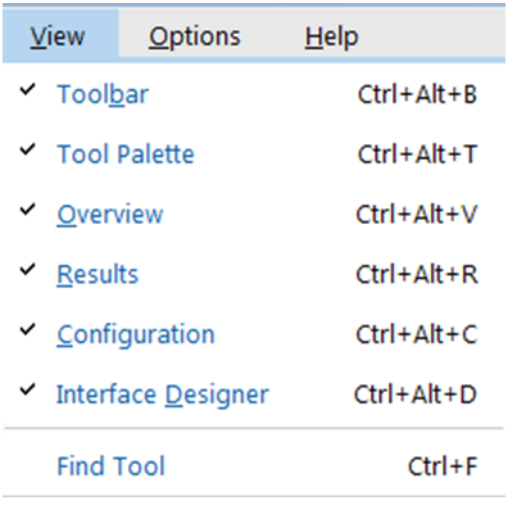


Figure 1-8 - View Menu

Toolbar: Turns the toolbar on and off. When it is checked, the buttons on the toolbar are visible (1 in Figure 1-5). (Hot Key: Ctrl+Alt+B)

Tool Palette: Turns the tool palette on and off. When checked, we will see the two rows of drag-and-drop tools (2 in Figure 1-5). (Hot Key: Ctrl+Alt+T)

Overview: Turns the overview window on and off. When checked, we can see a high-level overview of our canvas (3 in Figure 1-5). (Hot Key: Ctrl+Alt+V)

Interface Designer: Turns the interface designer window on and off. When it is checked, we will see a window that allows us to modify the user interface (6 in Figure 1-5). (Hot Key: Ctrl+Alt+D)

Find Tool: Opens a pop-up window that allows us to quickly find particular tools on the canvas. (Hot Key: Ctrl+F)

1.8 Options Menu

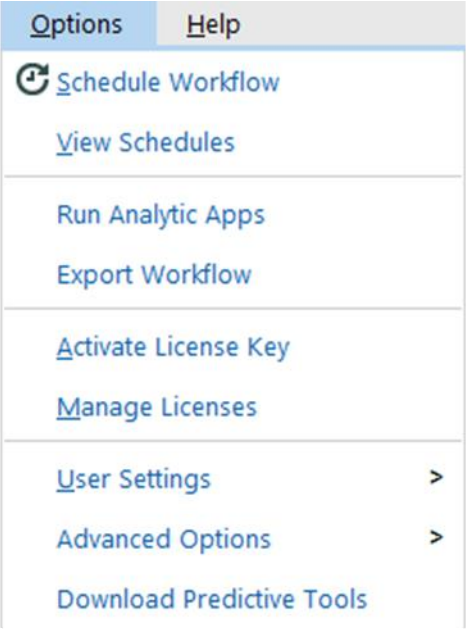


Figure 1-9 - Options Menu

Schedule Workflow: Allows us to set up a schedule to run our workflow as long as we have Alteryx Server.

View Schedules: Allows us to look at all of the scheduled jobs that we have access to on Alteryx Server.

Run Analytic Apps: Allows us to run Alteryx Analytical Apps that we have created.

Export Workflow: Allows us to export the workflow along with the packages that have been created.

Activate License Key: Opens a window that asks for an activation key so we can use Alteryx Desktop.

Manage Licenses: Opens a window that allows us to look at historical license keys and see what we currently have available.

User Settings: Opens a window that allows us to customize the way the canvas looks and the way some of the default settings behave.

Advacnced Options: Allows us to manage aliases, workflow dependencies, and encrypt workflows.

Download Predictive Tools: Allows us to download tools for the purpose of predictive analytics from the Alteryx repository.

1.9 Help Menu

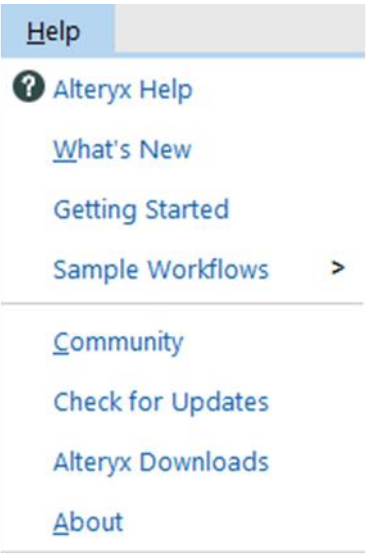


Figure 1-10 - Help Menu

Alteryx Help: Opens the Alteryx Help home page.

What's New: Opens the Alteryx Help release notes for the version of Alteryx that we are currently using.

Getting Started: Opens the Alteryx Getting Started page that is loaded in the beginning.

Sample Workflows: Allows us to access the Sample Workflows that are shipped with Alteryx.

Community: Opens the Alteryx Community web page.

Check for Updates: Checks our current version of Alteryx Designer and verifies if it is the most recent version.

Alteryx Downloads: Opens the Alteryx Download page where we can find additionally downloadable content. One example is the U.S. Census data.

About: Opens a window that tells us version, publisher, and serial number information about Alteryx Designer.

1.10 Samples

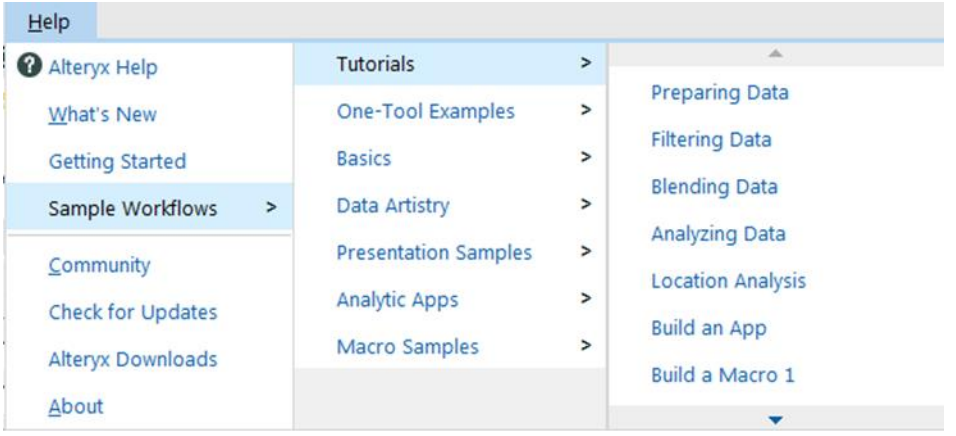


Figure 1-11 - Samples

Under the *Help Menu,* there is the option to open Sample Workflows. Depending on what we have downloaded from Alteryx, we will see different samples.

These samples are good starting points to get used to Alteryx and to learn how to use tools that we do not know.

Walking through the samples is beyond the scope of this book, as all of the samples are well documented and show how to build the workflow step by step. However, below is a picture of what the *Preparing Data* sample looks like before you begin to work with it.

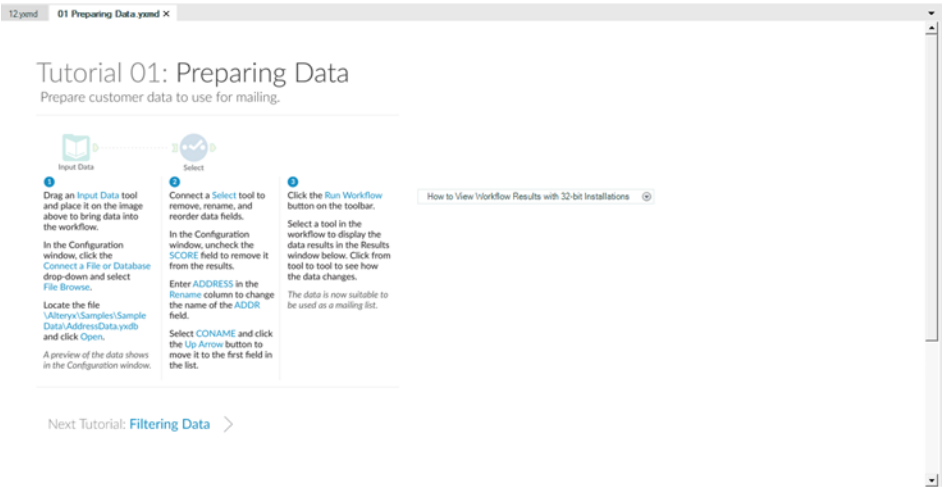


Figure 1-12 - Sample Tutorial

1.11 User Settings

Under the *Options menu,* there is an option called *User Settings*. This sub-menu has four options.

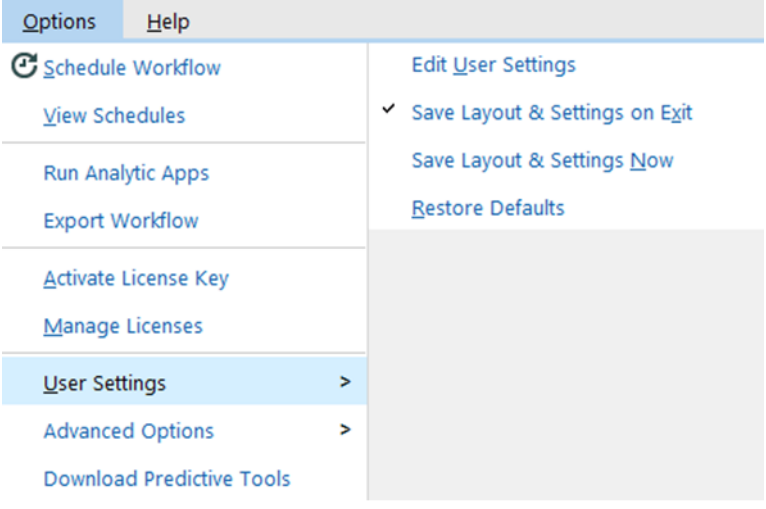


Figure 1-13 - User Settings

Edit User Settings: Opens a pop-up window for the user to edit the settings affecting the way Alteryx Designer looks and behaves.

Save Layout & Settings on Exit: Allows Alteryx to copy the settings we had when Alteryx was last closed as the default when we next open the program. Unchecking the option can turn off this functionality.

Save Layout & Settings Now: Allows us to tell Alteryx to use the settings we currently have as the default behavior for the program. This is most effectively used when we are not using *Save Layout & Settings on Exit*.

Restore Defaults: Allows us to revert Alteryx Designer to the default settings.

1.12 Toolbar

Figure 1-14 - Toolbar

The *Toolbar* has buttons we can use to do the following tasks. It is the same as if we were to navigate to the parallel options in the menus described above. The buttons simply make it easier to perform the most common actions.

 Allows us to create a new workflow. (Same as File > New Workflow) (Hot Key: Ctrl+N)

 Allows us to open a workflow. (Same as File > Open Workflow) (Hot Key: Ctrl+O)

 Allows us to save our current workflow. (Same as File > Save <File Name>) (Hot Key: Ctrl+S)

 Allows us to cut our current selection. It is greyed out and inaccessible when we have nothing selected and dark when we do. (Same as Edit > Cut) (Hot Key: Ctrl+X)

 Allows us to copy our current selection. It is greyed out and inaccessible when we have nothing selected and dark when we do. (Same as Edit > Copy) (Hot Key: Ctrl+C)

 Allows us to paste what we have on the clipboard. If we do not have something on the clipboard that can be pasted into Alteryx, we will get a pop-up that says *Nothing to paste.* (Same as Edit > Paste) (Hot Key: Ctrl+V)

 Allows us to undo what we just did. It is greyed out when we have nothing that we can undo. (Same as Edit > Undo) (Hot Key: Ctrl+Z)

 Allows us to redo what we just undid. It is greyed out when we have nothing that we can redo. (Same as Edit > Redo) (Hot Key: Ctrl+Y)

 Allows us to schedule the workflow on Alteryx Server. (Same as Tools > Schedule Workflow)

 Allows us to zoom in on our workflow. (Same as View > Zoom In)

 Allows us to zoom out on our workflow. (Same as View > Zoom Out)

 Allows us to run the workflow. (Same as Tools > Run Workflow) (Hot Key: Ctrl+R)

1.13 Tool Palette

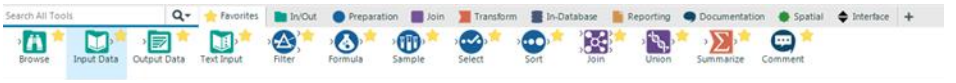


Figure 1-15 - Tool palette

The tool palette is one of the most important aspects of the Alteryx Designer interface. It is the easiest way to bring the tools onto our canvas. In the later section below called *Canvas,* we will discuss the alternative method.

The three main elements of the tool palette are:

- *Search*: Located in the top-left corner of the tool palette, this element allows us to type something into the bar, and every tool associated with that word is displayed in the Tool Selection.
- *Tool Categories*: Located to the right of *Search*, this element allows us to access all of the tools grouped into categories that Alteryx has defined. We have the ability to add to this list by developing macros, which we will discuss toward the end of the book.
- *Tool Selection*: Located at the bottom of the tool palette and shows all tools meeting either the criteria of the *Search* or *Tool Category* that we have selected.

There is one special category in the *Tool Categories* called *Favorites*. It is special because it does not have tools of its own. Instead, we have the ability to add and remove tools from this list by clicking the star at the top-right corner of each tool. When it is yellow, it means that it will show up in the *Favorites Category*. When we cannot see the star unless we point our mouse at the tool, and it shows up grey, that means that it is not in the *Favorites Category*.

Figure 1-16 - Favorites

The arrows on either side of the *Tool Categories* offer the ability to scroll across to see each of the categories. We see that, by default, *Favorites* doesn't follow this behavior. That is because it is pinned outside the scroll arrows. We can do this with any of the categories by right-clicking on them and selecting *Pin <Category Name>*. See the example below.

Right click on In/Out:

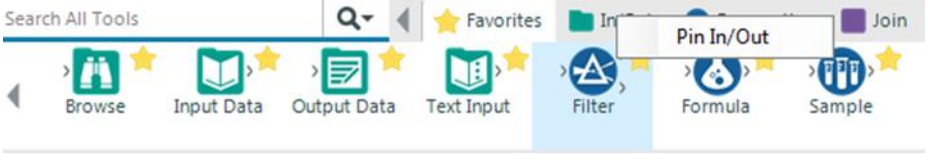


Figure 1-17 - Pin In/Out

Select Pin In/Out:

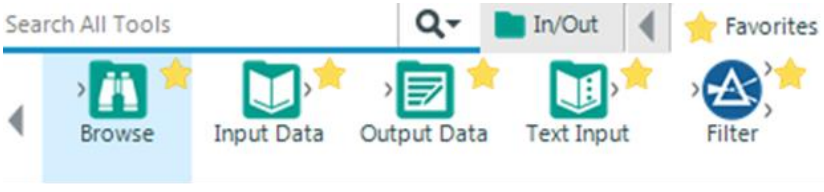


Figure 1-18 - Selecting Pin In/Out

In/Out is pinned:

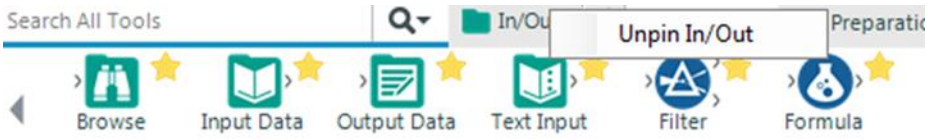


Figure 1-19 - Unpin In/Out

Select Unpin In/Out:

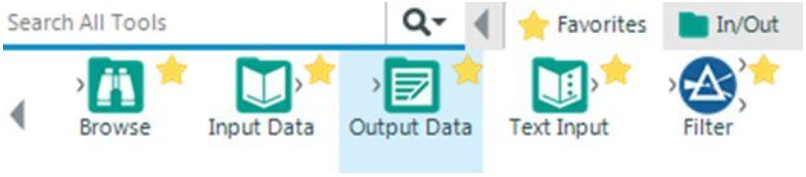


Figure 1-20 - Tool Categories

We can see that certain icons show up on the *Tool Category* names. That is because each of the categories has a *Default Tool*. If we were to click on the *Tool Category* and drag it to the canvas, the tool in the icon is what we bring onto the canvas.

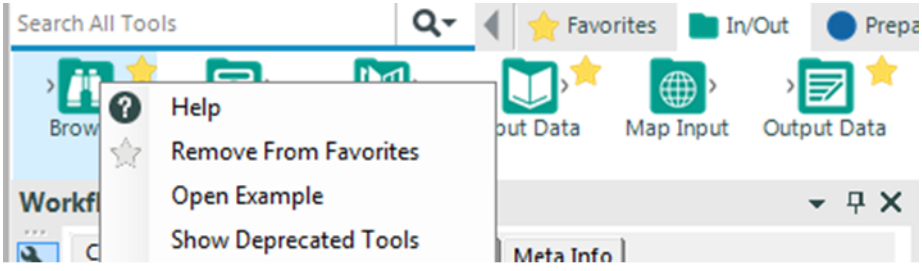


Figure 1-21 - Tool Category

We can see in Figure 1-21 an option called *Show Deprecated Tools*. Selecting this option shows tools that belong to outdated

versions of Alteryx. If we need a former functionality, we can access it here.

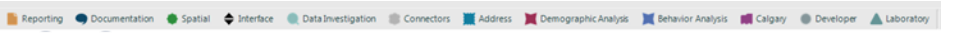


Figure 1-22 - Additional Tool Categories

All of these categories may not be available to all users because Alteryx has many additional packages that we can download in addition to the basic package. One example is if we had not chosen to download the Predictive package when we downloaded Alteryx Designer, we would not see the *Predictive Category*.

We will only cover the basic package in this book, plus some downloaded content.

1.14 Overview

Below is a picture of the canvas and overview window with a simple data stream that is too wide to see what is happening in the view of the canvas. The right side of the image is the *Overview* window, and it shows the view of the canvas in relation to the entire workflow.

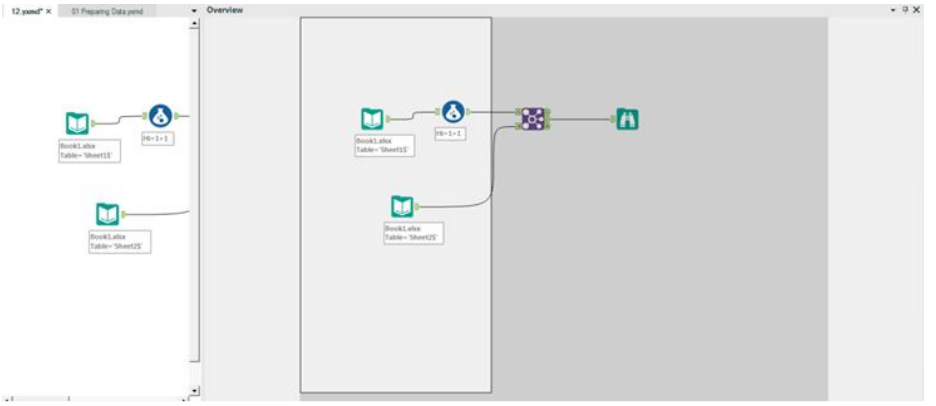


Figure 1-23 - Overview Window

In addition to showing where we are, the overview window makes it easy to find the part of our program that we are interested in, because we can move the view of the canvas by clicking and dragging the white box in the overview window.

1.15 Results Window

Every time we run a workflow, we will get a series of returns that lets us know things about the run. One example of this is below.

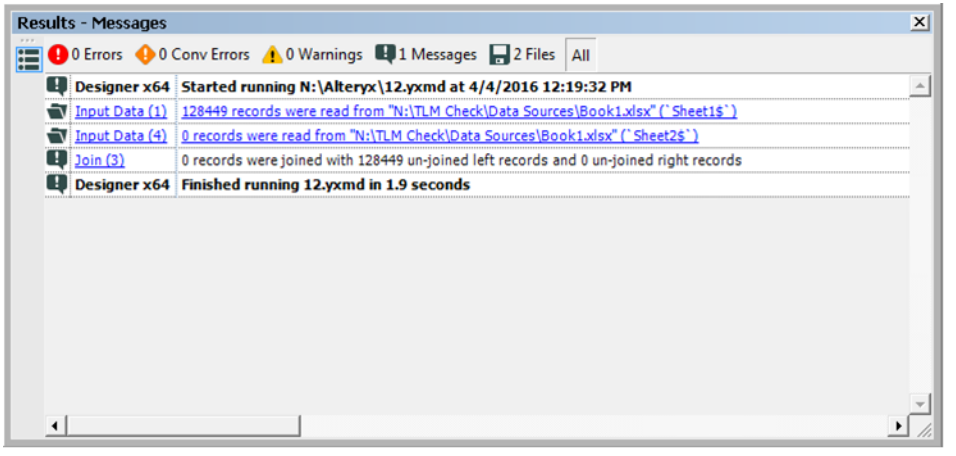


Figure 1-24 - Results Window

In this run, we can see that we had 0 Errors, 0 Conversion Errors, 0 Warnings, 1 Message, and 2 Files. The counts here reflect the eight notes we see about the run in the text portion of the Output window. We have eight notes instead of the six we would expect from the counts of the messages. This is because the first note is that the workflow started to run, and the last note is that the module had finished the run. Even if we have no issues or messages during the run, we will still have these two notes because they are always written. The last message changes depending on whether or not we have any errors during our execution. The output window is important in diagnosing issues with the data stream.

- *Errors*: These are critical failures that will terminate the data stream they are in.
- *Conversion (Conv) Errors*: These are potentially problematic issues that stem from a data mismatch when converted from one format to another.
- *Warnings*: These are things we should be aware of but could be expected, so the results are printed in the Output for us to know about.
- *Messages*: These are text in the program that are designed to print when certain stages of the data stream have been finished or certain conditions were met.
- *Files*: These are links that will allow us to open any file created during the running of the workflow

1.16 Properties Window

The properties window is where we will spend the vast majority of our time. Every tool has a different properties window, and we will discuss each of them as we discuss the individual tools. Here, we will discuss the properties window of the workflow and connections.

There is a pane on the left side of every properties window that has a universal set of symbols. We will discuss each of these as they first appear in this book. A list of all of these symbols is in Appendix E.

Workflow configuration

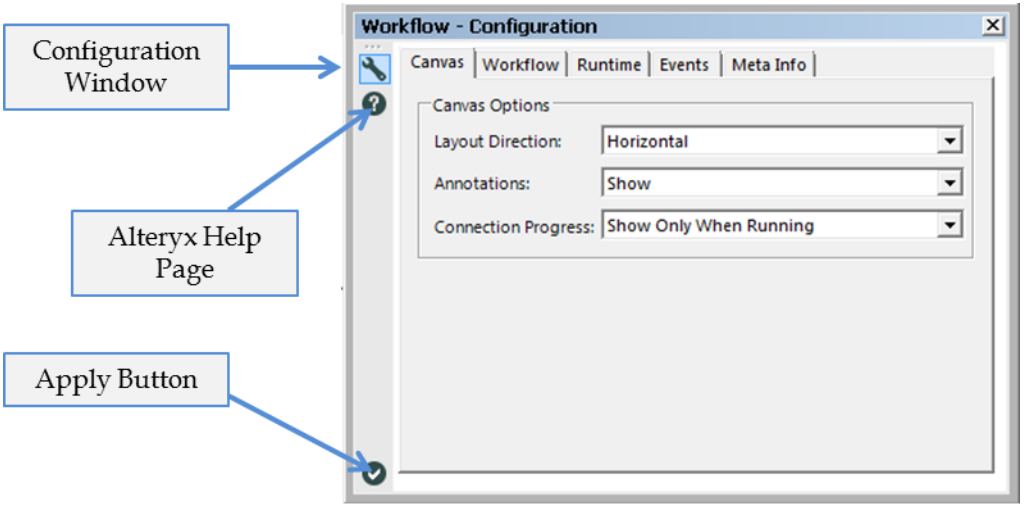


Figure 1-25 - Workflow Configuration – Canvas Tab

Notice the button that looks like a wrench at the top-left of each of these properties windows. Selecting the wrench shows the *configuration window* so that we can modify the settings of the tool. As we discuss other tools, the importance here will become apparent.

The "?" in the circle opens the Workflow Alteryx Help page.

The "✓" at the bottom-left corner is an apply button. We can click this to apply the changes we have made. We do not need to click this button; we can just click somewhere on the canvas.

The first tab of the *Workflow configuration* window is Canvas. We have the ability to change three things in this window.

- *Layout Direction* is a drop-down list that allows us to rotate our data stream from a horizontal layout, which is the default, to a vertical layout, which is the layout for older versions of Alteryx. We must set either *Horizontal* or *Vertical* and cannot use a combination of the two.
- *Annotations* is a drop-down list that allows us to change what the text boxes under Tools say. When this option is set to *Hide,* the text does not show. When this option is set to *Show*, the default text is shown for each tool. When this option is set to *Show w/ Tool Names,* the annotations tell us what the tool names are as well as the default text.
- *Connection Progress* is a drop-down list that allows us to change when we see the progress labels on each of the tools. By default, this is set to *Show Only When Running*, but we can also set it to *Hide* and *Show*, which will never or always show the last run's progress respectively.

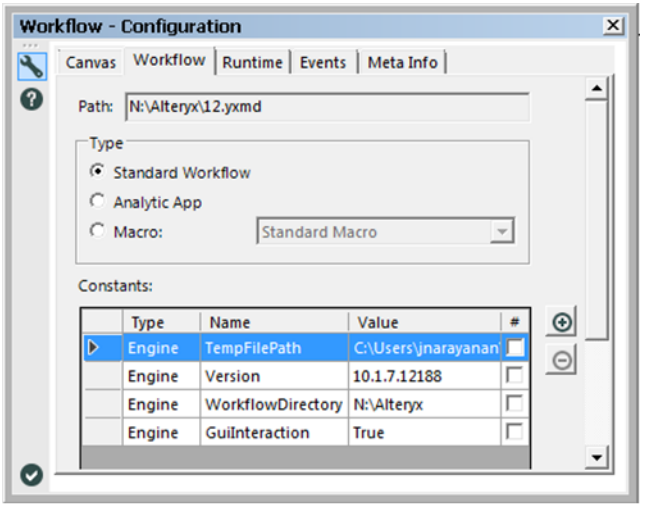


Figure 1-26 - Workflow Configuration – Workflow Tab

The next tab is *Workflow*.

- The *Path* (file location) shows up in this box when the file has been saved.
- The *Type* allows us to choose which format the file should be. This is also where we select the type of *Macro* we want to create. (This will be discussed in *The Mermaid Coffee Company* chapter.)
- The *Constants* section allows us to define variables that we can use throughout our Alteryx Module.

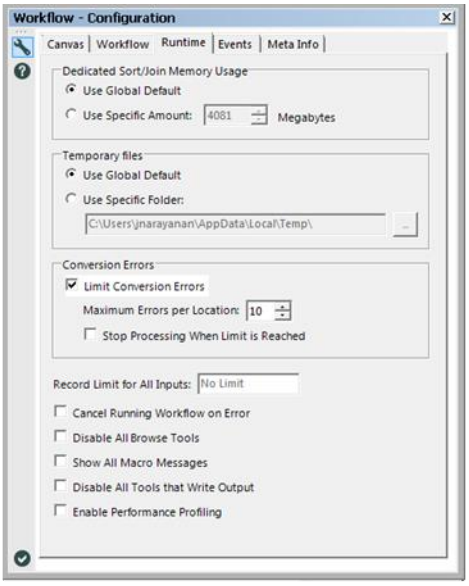


Figure 1-27 - Workflow Configuration – Runtime Tab

The *Runtime* tab allows us to configure settings that impact the way that Alteryx runs.

- The *Dedicated Sort/Join Memory Usage* allows us to set the maximum amount of memory that Alteryx is allowed to use for the *Sort* and *Join* functions, which can be highly memory-intensive.
- The *Temporary Files* setting allows us to define the folder that our temporary files are written to.
- The *Conversion Errors* setting allows us to change the way the workflow behaves if it experiences conversion errors, including limiting them from showing up for each tool and stopping the workflow if they occur.
- The *Record for All Inputs* setting allows us to set a maximum number of records read from any of our input files.

- The *Cancel Running Workflow on Error* is useful if we are trying to diagnose a specific issue in a workflow.
- The *Disable All Browse Tools* should be set after we finish testing and the process is in production. We want to do this because browse tools are slow to create and unnecessary unless testing. It is better to use this than to delete all of the tools because it will make modifying the workflow easier later.
- *Show All Macro Messages* allows us to see the messages we are getting in the macro tools so that when we are running a workflow, we can see everything that may be an issue.
- *Disable All Tools that Write Output* is useful for testing because we can turn off only the outputs so we are not rewriting the data every time we run the workflow.

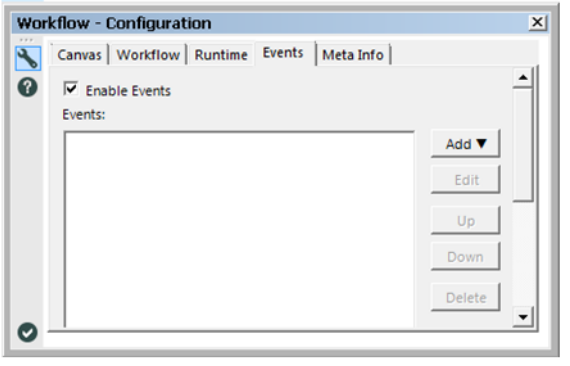


Figure 1-28 - Workflow Configuration – Events Tab

The *Events* tab allows us to define events that will allow secondary actions to take place.

When we add an event, we can choose to define a command line code or send an email when a condition is met. This is useful for security or maintenance because we can set up our workflows to let us know anytime the workflow ran or any time it ran with issues.

The *Meta info* tab allows us to define metainformation about our workflow. This is important when we are publishing a workflow to Alteryx Server or the public Alteryx Gallery.

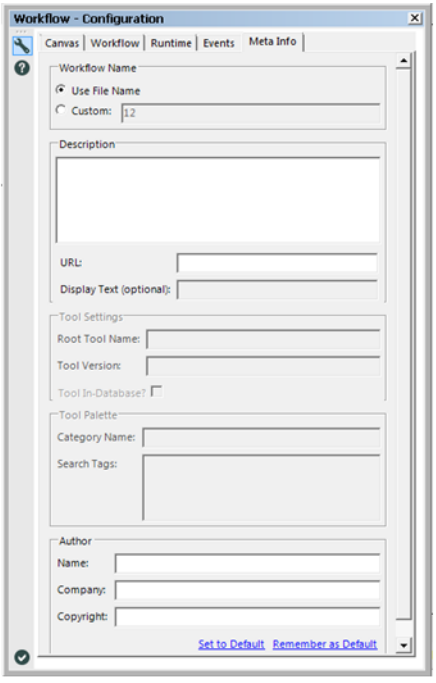


Figure 1-29 - Workflow Configuration – Meta Info Tab

- *Workflow Name* allows us to customize what we want the workflow to be called (independent of the file name) so that versioned workflows can have the same name, as far as the end user is concerned.
- The *Description* is a place for us to describe what the workflow does.
- The *URL and Display Test* fields allow us to identify where the module is published.
- The *Author* section allows us to publish information about us and our company along with the workbook.

Connection configuration

When using the default settings, connections are the curved black lines that connect one tool to the next.

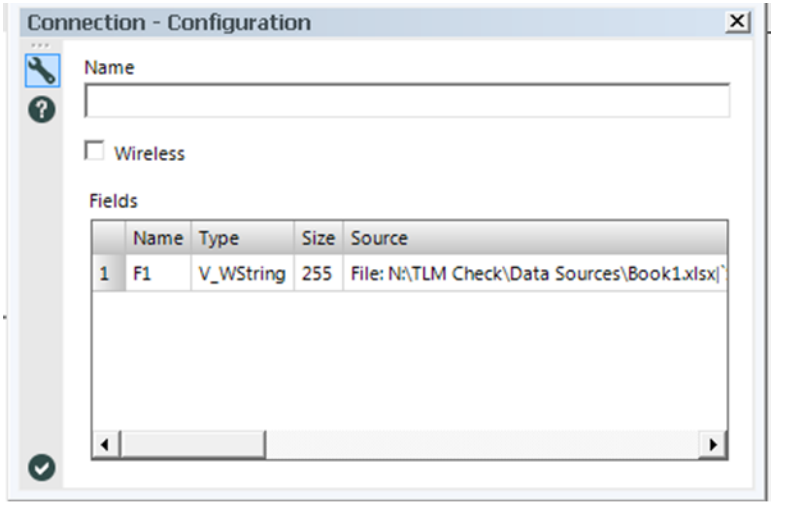


Figure 1-30 - Connection Configuration

The *Connection configuration window* has three components.

- *Name* allows us to rename our data connection. This is most useful when we have multiple data connections going into the same tool connector.
- The *Wireless* check box allows us to make the connection wireless (invisible) unless one of the tools it connects is selected.
- The *Fields* box allows us to see some of the metadata about each field in the data stream at this connection.

1.17 Interface Designer

The Interface Designer is a window used in developing apps and macros. This is where we get to design the user interface the end user will see.

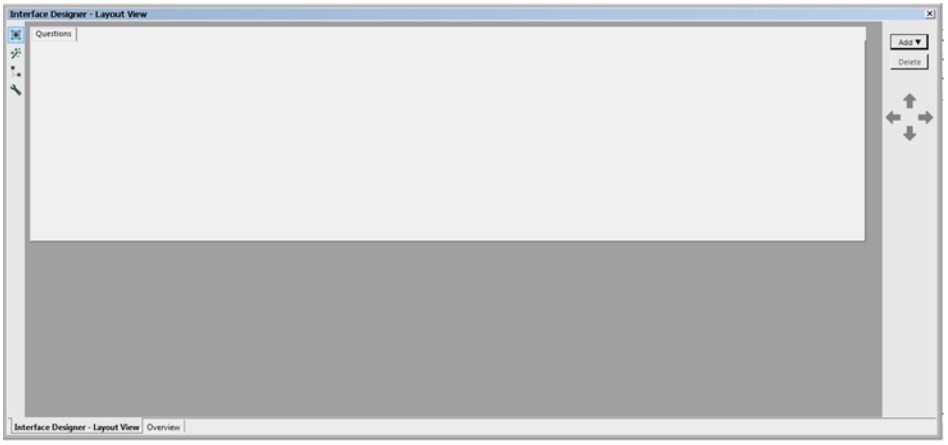


Figure 1-31 - Layout View

The default tab view of the Interface Designer window is of the Layout View, which allows us to design the user interface in a graphical format.

Figure 1-31a - Interface designer settings

The Interface Designer, like the Properties window, has a side bar that tells you what you are looking at. From top to bottom, the icons are:

- *Layout View*, where we design the interface graphically.
- *Test View*, where we see what the end user would see.
- *Tree View*, where we look at the structure of our questions in a tree structure
- *Properties*, where we set properties associated with the interface (this is independent of the Properties window itself).

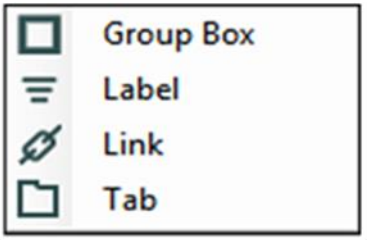


Figure 1-31b - Interface designer drop down

In the *Layout View,* we design the interface and thus have a significant amount of flexibility in what we can do.

The first and most intuitive thing to do is move questions up and down according to the order we want them asked to the end user. We can do this by clicking the up and down arrows on the right-hand side. We may notice that while we are moving questions, they sometimes move within a box that frames the question we were trying to pass. This is because we have the ability to make one question determine if the other is also asked. By leaving a question inside another, we make that question dependent.

The next thing we can do is select the Add drop-down menu and see that we can add any of four things. Those things are:

- *Group Box*: a text element that we can put questions in.
- *Label***:** a text element that we cannot put questions in.
- *Link*: a hyperlink in the interface.
- *Tab*: a tab in the interface.

When we are working with the *Interface Designer*, we will be opening up the Properties windows associated with whichever tool we have selected at the time. The following are the properties windows of the four objects described above.

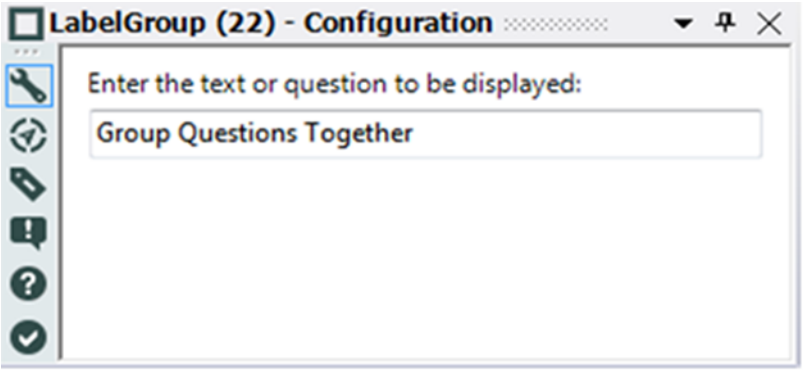


Figure 1-32 Label Group Configuration

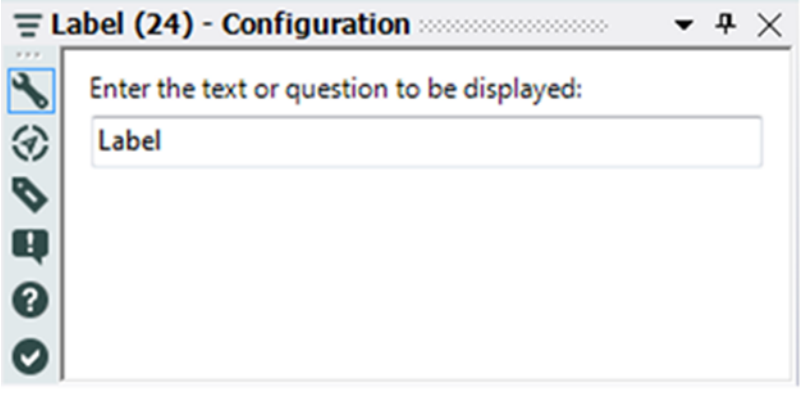


Figure 1-33 Label Configuration

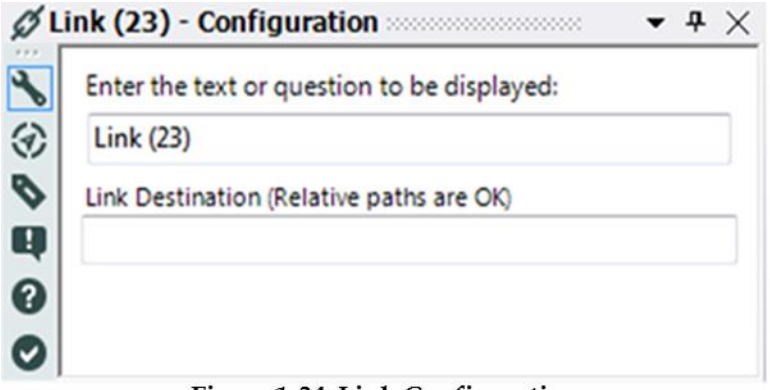


Figure 1-34 Link Configuration

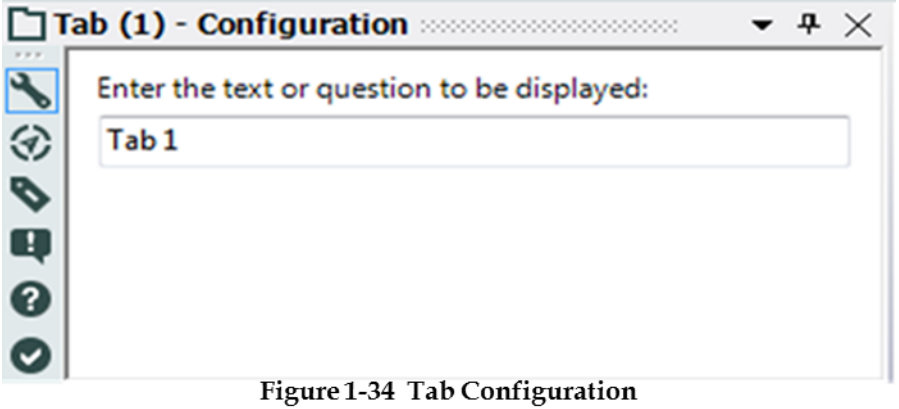


Figure 1-34 Tab Configuration

As we can see, the four windows are similar. All four ask what we want the element in the interface to be called, and *Link* also asks us what link address we want to add to the interface. In addition to these questions, we see for the first time that we have three new icons in the side bar of the Properties window.

Figure 1-34a Side Bar Icons

- *Navigation*: Allows us to move directly to the incoming or outgoing tools.
- *Annotation*: Allows us to change how this tool is named and how its annotation behaves.
- *Messages*: Allows us to look at the output messages specific to this tool.

Navigation, *annotation*, and *messages* are part of every tool properties window, and they always behave the same.

Below are examples of a *Check Box* Navigation, Annotation, and Messages window when connected to an *Action* tool.

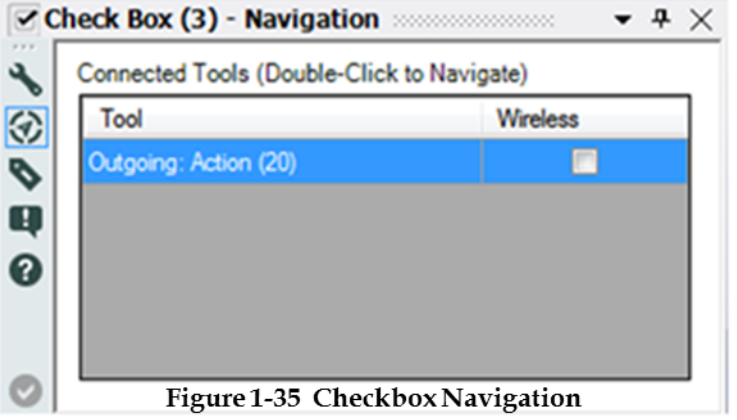


Figure 1-35 Checkbox Navigation

We can see that *Navigation* lists the tool that this *Check Box* is connected to, Action (20), and tells us that it is an outgoing connection. That means that the Action is downstream of the Check Box. If the Wireless box were checked, the wire would be invisible. We will discuss both of these tools, as well as wireless connections, in chapters *Applications Wanted* and *Meta-morphosis*.

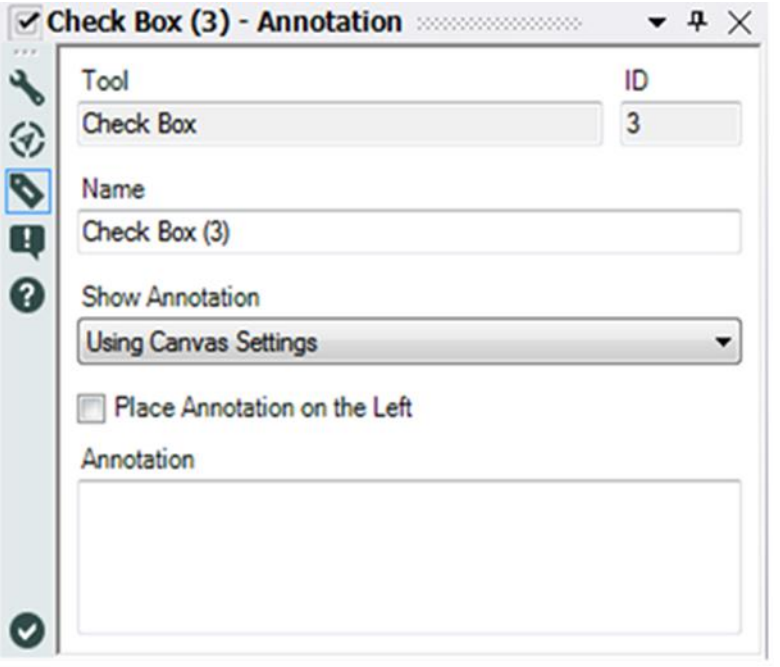


Figure 1-36 Checkbox Annotation

The Annotation window tells us that the tool is a Check Box and that it has the ID number 3 (which means it will try executing this tool third if there are not outside influences). We then have the ability to change the name of the tool. We can change the annotation setting specific to this tool as well as provide a custom annotation (this will show up under the tool in the data stream).

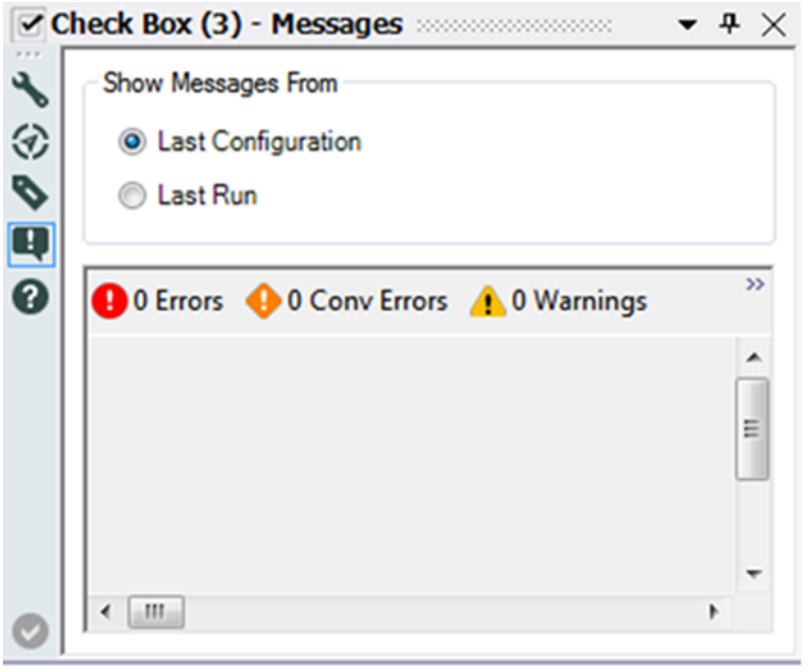


Figure 1-37 Checkbox Messages

The Messages window allows us to look at this tool's specific output messages generated during either the last configuration or last run, so we can investigate any issues while we are looking at the tool.

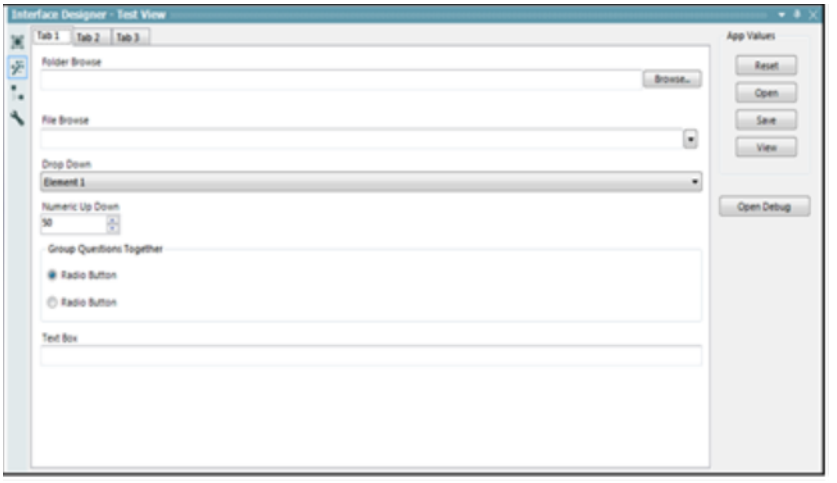


Figure 1-38 Test View

Looking back at the Interface Designer, *Test View* allows us to answer the questions in the same way that the end user would so we can test the app.

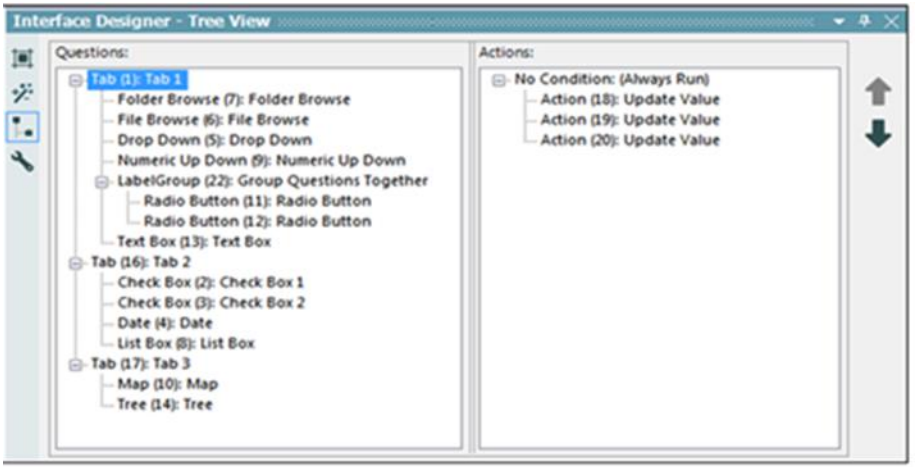


Figure 1-39 TreeView

TreeView allows us to look at our user interface as a tree structure so it's easier to make sure our logic is properly grouped.

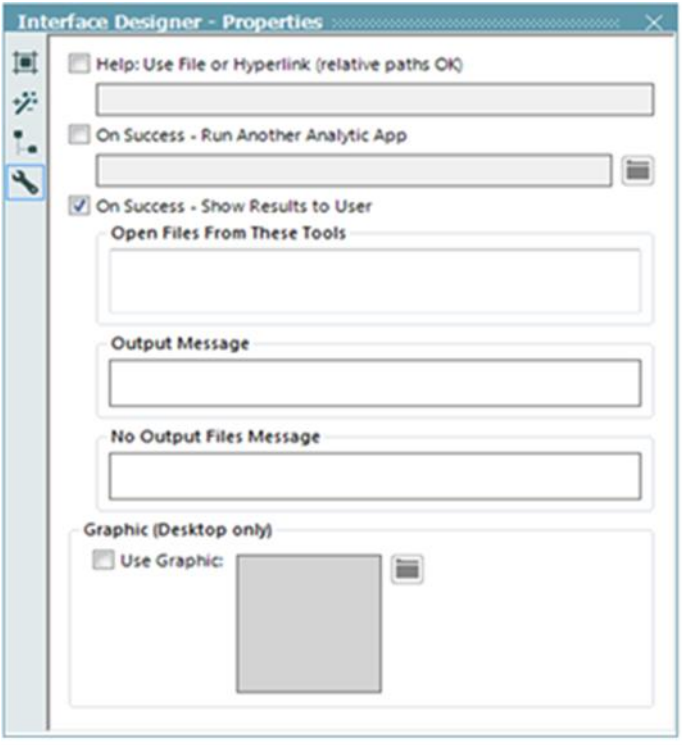


Figure 1-40 Interface Designer Properties

Interface Designer Properties allows us to customize many aspects of our app or macro.

1.18 Canvas

This is what the default canvas looks like before we have brought any tools onto it.

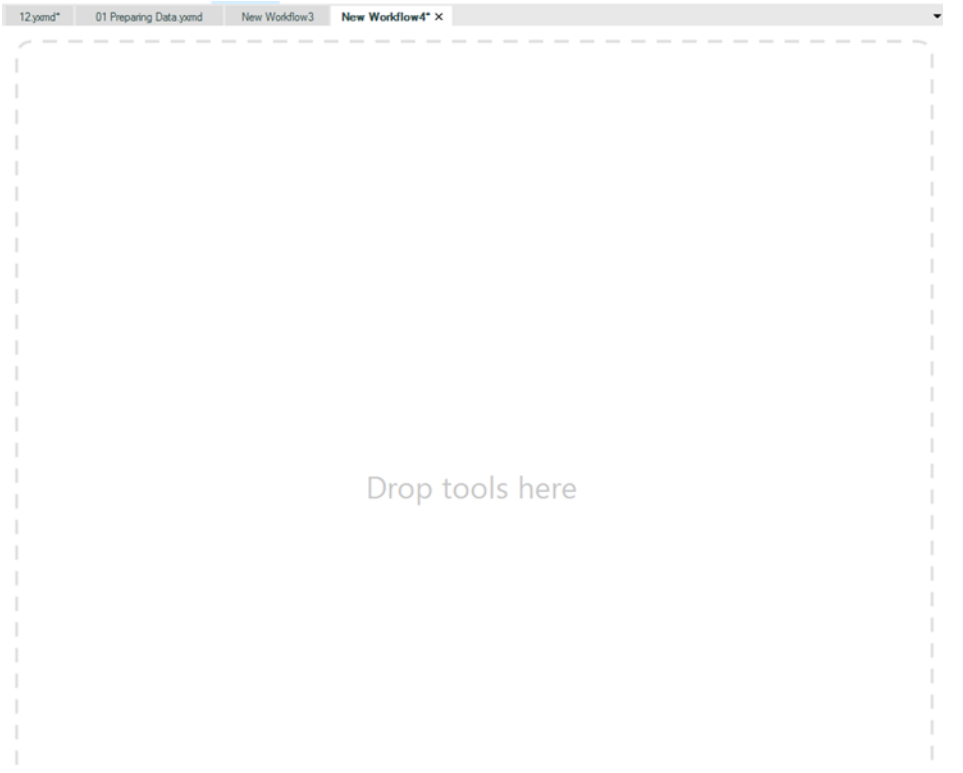


Figure 1-41 - Default Canvas

We can see three elements: the tab name, a drop-down list of all of the tabs, and a white area that reads “Drop tools here.”

Let's see what happens if we follow the following steps.

1. Open three new windows.
2. Click on the third tab.
3. Click on Window > New Horizontal Tab Group.
4. Click and drag the fourth tab (now third on the top) from the top group to the bottom group.
5. Click on Window > New Vertical Tab Group.

We should see that our canvas has split into three sections and looks similar to the image below.

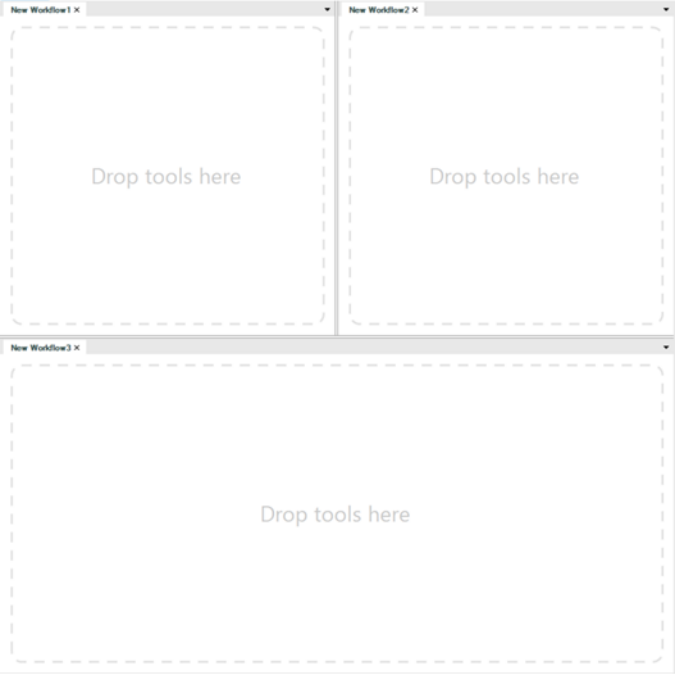


Figure 1-42 - Canvas with elements

The ability to display workflows next to one another makes it much easier to test parts of a program. We can copy and paste portions between canvases to test, compare potential changes, or just work with multiple workflows at the same time.

Moving the locations of window elements

The overview, output, interface designer, and properties windows allow us to move them around the screen by clicking and dragging the rough part of the title bar (or). When we do this, we have the ability to snap them into the window by dropping them on one of the arrows or let it float in front of the canvas or other windows by letting go of it while not over an arrow.

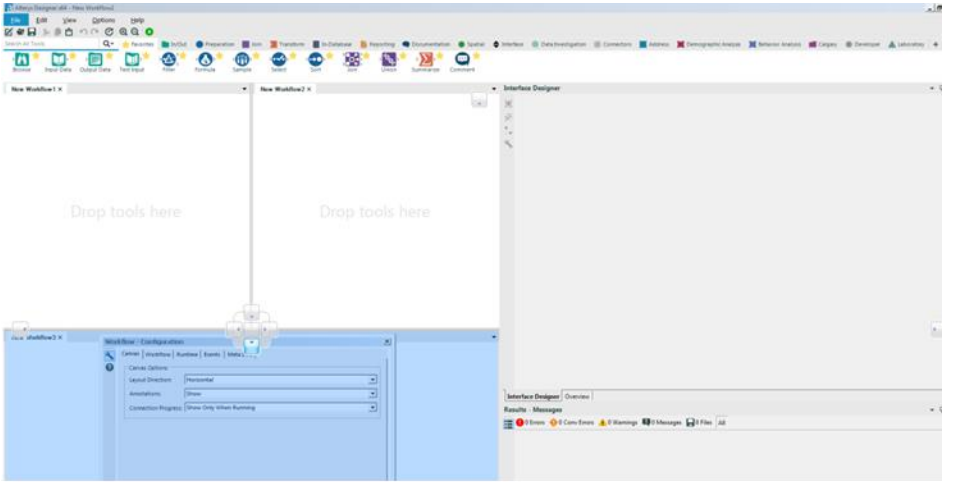


Figure 1-43 - Movement of Canvas Elements

1.19 Using Tools

Data streams start with some sort of data source, and in most cases, this will be the input tool. From there, we may do a series of operations before bringing the data back together to look at.

We will cover each of these methods in detail in the next chapter, but for now, we will look at the structure of a data stream.

This Alteryx Workflow takes input data from a file called Test.txt, splits the data stream (doubling the data), adds a new field to one side, and brings the two streams together to be viewed.

Notice that there are two types of arrows on the incoming (left) side of the tools. A green arrow indicates that a single input can be connected there, while a grey double arrow indicates that multiple connections can be made. This is important because it allows us to know how to connect items to each of these tools.

We can see that there is no parallel tool on the outgoing (right) side of the tools, despite the data stream having multiple outgoing connections. This is because every tool allows us to branch the data stream off into many directions. We also see lightning bolt and question anchors, which we will discuss when we first use them in *Applications Wanted*.

There are two ways to bring tools onto the canvas: We can either drag the tool from the Tool Palette or right-click on the canvas and navigate the menu on the next page.

1.20 Insert Tool Menu

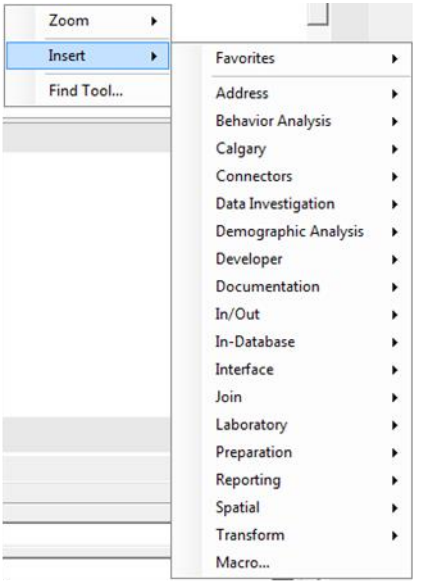


Figure 1-44 - Insert Tool Menu

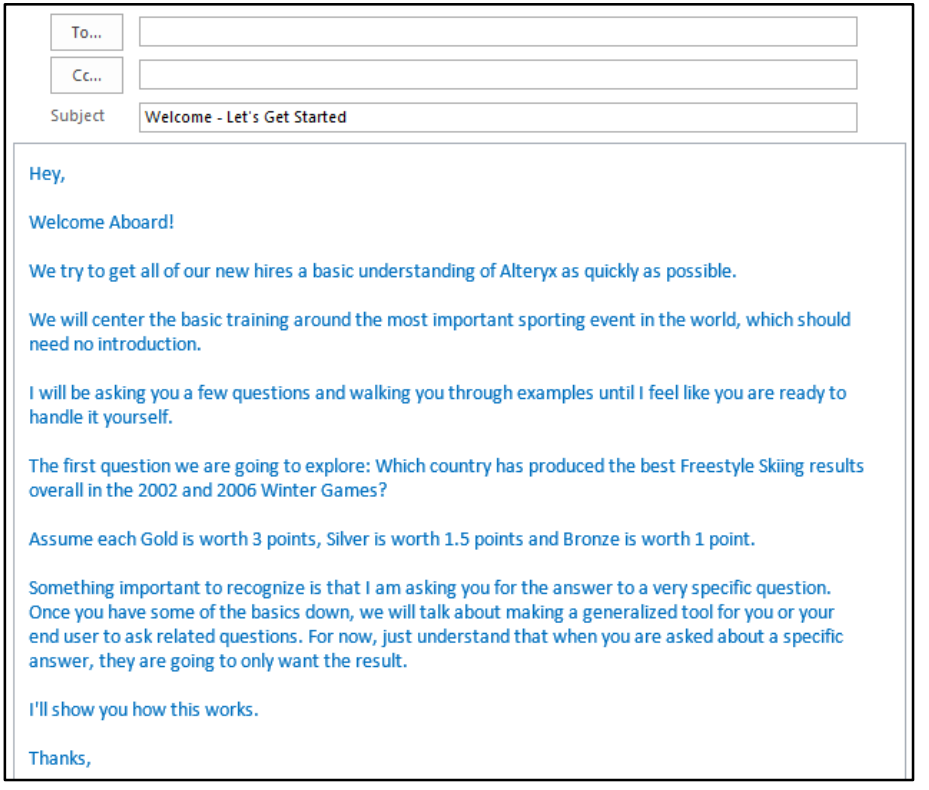
As an alternative to using the tool palette, you can right-click on the canvas to add tools using the Insert Tool Menu. Simply navigate this menu to find the tool you are looking for

CHAPTER 2

The Games

We all love playing make believe, don't we? As we discussed in the last chapter, assume that you, dear reader, are the new consultant at a company that works with the Alteryx Analytical Platform.

This book is your mentor and here is our first problem to solve!



2.1 Tools & Concepts

In this chapter, we will cover the improved features for Alteryx designer 10.x and the various tools and concepts as mentioned below:

Tools	**Concepts**
Browse	Importing Data
Comment	Viewing Data
Filter	Outputting Data
Formula	Identifying Desired Results
Input Data	Answers to Questions
Join	Tidy Datasets
Output Data	Normalized Datasets
Running Total	Creating Calculations
Sample	Combining Data
Select	Creating Data Subsets
Sort	Summarizing Data
Summarize	Organizing Data
Tool Container	Organizing Workflows
Transpose	Documenting Your Work
Union	

2.2 Improved Features

Several updates were implemented since Alteryx 10.0 was released. Most of the tool icons got a new look, and some tools got new features. New features for each tool will be covered in their respective sections. Here, we will cover the new UI and Alteryx Designer's updated features.

- Reading and writing Excel files (.xlsx) is done using a new Alteryx driver for Microsoft Excel.
- The Predictive Tools Installer has been separated from the Alteryx Designer install to allow for future updates to the Predictive Tools and packages without the need for an update to the Designer.
- After running a workflow, clicking on the Input or Output tools will automatically populate results (up to 1MB).
- Tool palette is customizable to show only those categories and tools you want to use.
- The File menu is reorganized to be simpler and more intuitive to use.
- The enhanced integration between the Designer and Gallery makes it easier to collaborate on workflows hosted in a Gallery. From within the Designer, one can add and maintain connections to a Gallery, open Gallery workflows and edit them, and then save a new version of the workflow back to the Gallery. Previous versions of the workflow are maintained and can be retrieved via a version history window in both the Gallery and the Designer. Additionally, any workflow version can be made the "published" version that users will see by default.
- In-Database support is added for these Data Platforms: Amazon Redshift, Impala, Teradata, and Spark.

- Changes to the Connect In-DB tool include: new streamlined UI, new option to allow password decryption, and new option for creating file-based In-Database connections in order to simplify obtaining connection information from IT Admin/DBA for server users.
- Changes to the Data Stream In tool include: new option to allow password decryption, users can write out to a permanent table when streaming data from an external data source.
- New Macro Input In-DB and Macro Output In-DB tools have been added for In-Database processing so users can build macros with In-Database.
- New Dynamic Input In-DB and Dynamic Output In-DB tools have been added for In-Database processing so users can retrieve the underlying SQL query and other metadata info being sent to the database or to the In-Database tools in a workflow.
- These connectors are now supported: Amazon Redshift bulk load (write), Netsuite (read/write), Qlik (read), PostgreSQL 9.4 (read/write), SAP Hana (read/write), Spark SQL (read).
- Users can now read and write to JSON from the Input and Output Data tools.
- Users can now browse to their Hadoop distributed file system and read/write to HDFS via the new connections in the Input Data tool. (HDFS tools released in 9.5 have been deprecated).
- Salesforce and Marketo connector tools built on the REST API provide enhanced functionality. The existing SOAP-based connectors are now deprecated.

- MongoDB Input and Output tools now support version 3.0. Mongo input will also read from replica set members including primary and secondary.
- Users now have the ability to retrieve a list of sheet names from an Excel (.xlsx) file and read the list via the Alteryx XLSX driver.
- Apache Avro support is no longer listed as being in beta. Support for SQLite has now been upgraded to 3.8.9.
- The Download tool now supports multithreading to increase the speed at which data is retrieved. The maximum number of connections that can be used in the same tool is 32.
- Authenticated proxy can now be enabled from User Settings. The Download Tool, Amazon Redshift Bulk loader, Amazon S3 Upload, and Amazon S3 Download tools will function through a proxy server.
- Alias Repository has been renamed to Alias Manager and supports both standard as well as In-Database connections.
- A new option that provides information about the performance of tools in a workflow is available. Select the "Enable Performance Profiling" checkbox on the Runtime tab of the workflow, and run the workflow to view the percentage of time spent processing each tool in the Results window. This option should only be used when debugging a specific workflow, as it may decrease the performance of the workflow slightly. Sample workflows that demonstrate different configuration options for one tool at a time are one example. Twenty new examples have been added, which can be accessed via Help > Sample Workflows > One Tool Examples. In/Out: Directory; Map

Input Preparation: Auto Field; Generate Rows; Imputation; RecordID; Tile; Unique Join: Append Fields; Find/Replace; Fuzzy Match; Make Group Parse: Date/Time; RegEx; Text To Columns; XML Parse Transform: Count Records; Cross Tab; Running Total; Transpose.

- The Block Until Done tool now has numbered outputs that output "in order."
- The Map tool now has a zoom/bounds option that makes the map zoom/pan to the reference file.
- In the Alteryx Designer version 10.x, the Configure Workflow and Results windows are split into two distinct views.

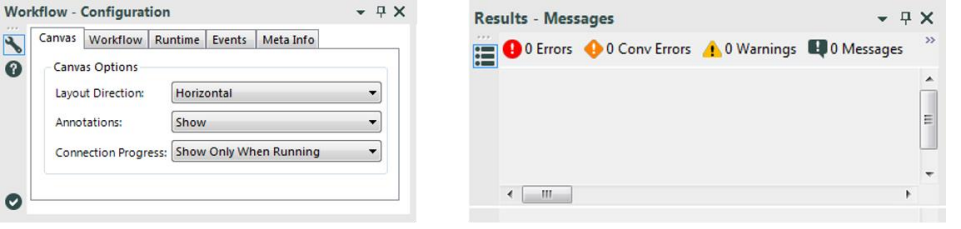


Figure 2-1 – Workflow and Results windows

- Configuration and Results windows can be dragged and displaced to different possible positions.

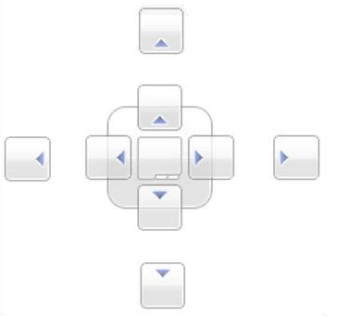


Figure 2-2 – Possible positions

- Every tool in Alteryx now has a Preview feature to show a snapshot of the data it contains.

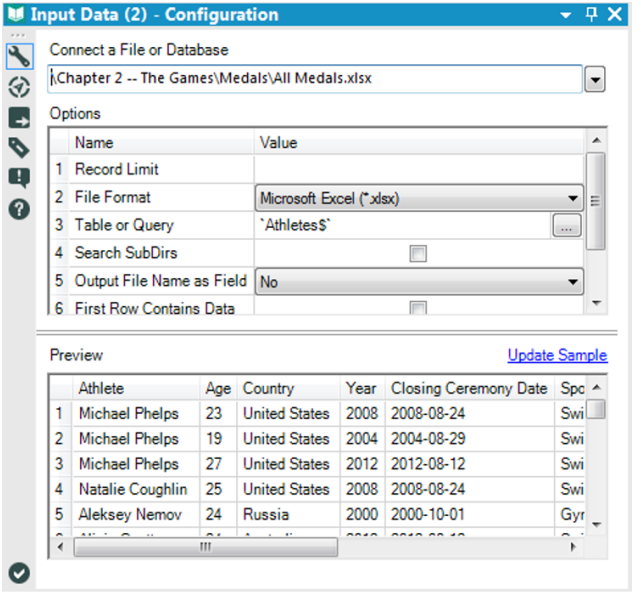


Figure 2-3 – Preview Data

2.3 Browse

Figure 2-4 - Browse

The *Browse* tool gives us a tabular view of the data in a data stream at the point it is connected.

Group	Input	Output
In/Out	Data stream	None

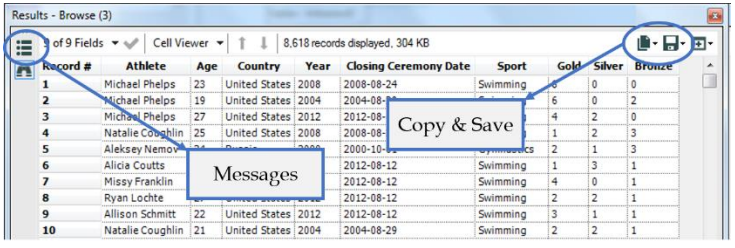
Note: It is very important for the development of workflows, applications, and macros but should be disabled when development is completed to improve speed.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Results Window:

The *Browse Results* Window allows us to view the data that was in the data stream during the last run.

The message option shows the number of records processed, processing time, and any errors during processing.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Gold	Silver	Bronze
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	8	0	0
2	Michael Phelps	19	United States	2004	2004-08-		6	0	2
3	Michael Phelps	27	United States	2012	2012-08-		4	2	0
4	Natalie Coughlin	25	United States	2008	2008-08-		1	2	3
5	Aleksey Nemov	24	Russia	2000	2000-10-	Gymnastics	2	1	3
6	Alicia Coutts	24	Australia	2012	2012-08-12	Swimming	1	3	1
7	Missy Franklin	17	United States	2012	2012-08-12	Swimming	4	0	1
8	Ryan Lochte	27	United States	2012	2012-08-12	Swimming	2	2	1
9	Allison Schmitt	22	United States	2012	2012-08-12	Swimming	3	1	1
10	Natalie Coughlin	21	United States	2004	2004-08-29	Swimming	2	2	1

Figure 2-5 – Browse, Results

The two icons on the right side (shown above) are *Copy* and *Save* functions. They allow us to directly copy the data out of the browse tool or save it to a file.

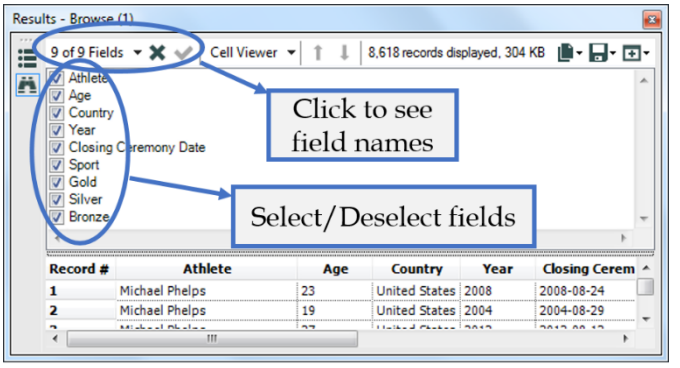


Figure 2-6 – Browse, Field Names

Clicking on the down arrow shown above gets us a list of all field names so we can select only the relevant fields to be displayed. The checkboxes ✓ in this list allow us to select or deselect every field.

The text shown after clicking on *Cell Viewer* depends on what is selected in the records.

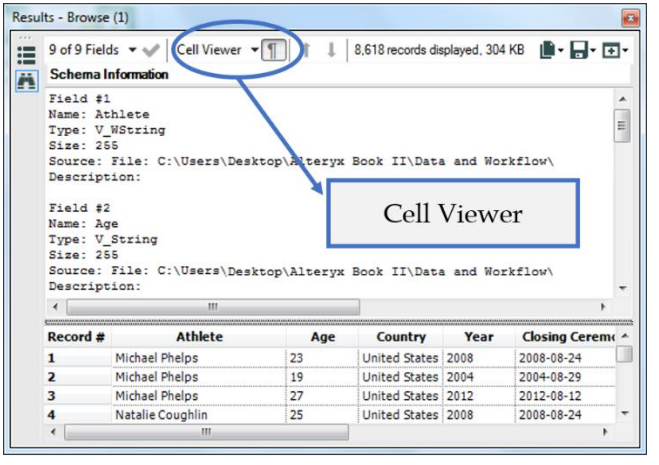


Figure 2-7 – Browse, Cell Viewer

Clicking *Record #* selects everything, allowing us to see the information about each column. If a single column is selected, we see metadata about that column. If a row is selected, we see how much data the record has in it. If a cell is selected, we can see the contents of that cell formatted with all line breaks.

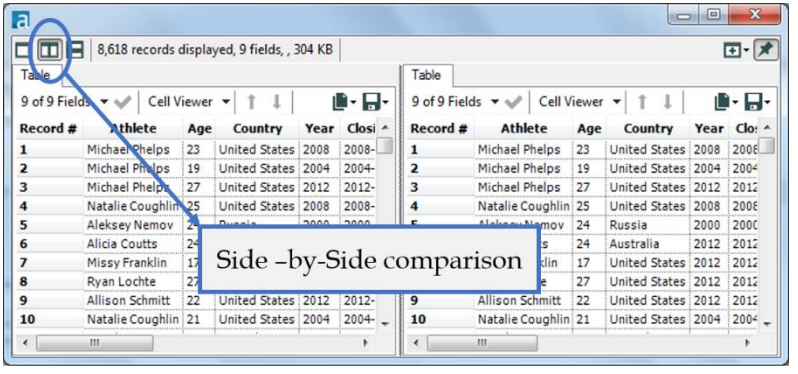
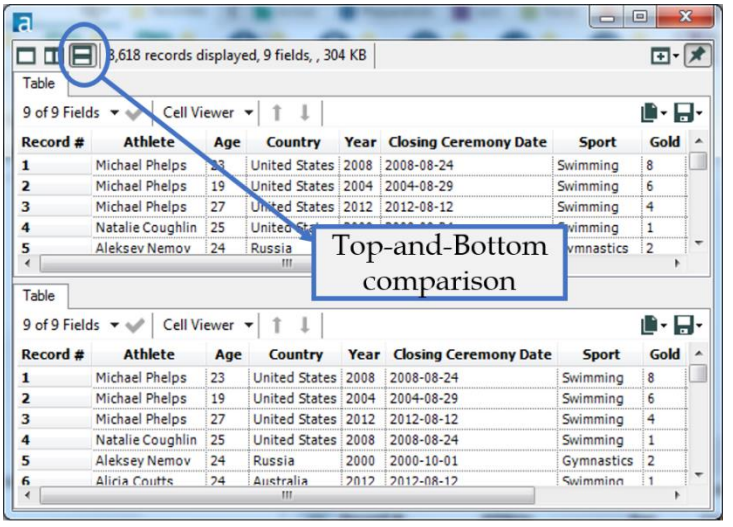


Figure 2-8 – Browse, Side by Side

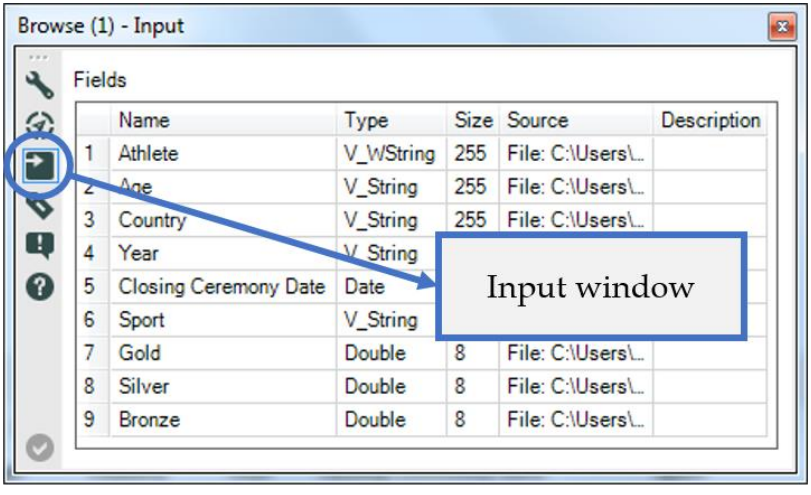
We have the ability to compare the dataset either Side-by-Side (vertical compare) or Top-and-Bottom (horizontal compare) by clicking on the new window button located at the top-right of the results window.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Gold
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	8
2	Michael Phelps	19	United States	2004	2004-08-29	Swimming	6
3	Michael Phelps	27	United States	2012	2012-08-12	Swimming	4
4	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	1
5	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	2
6	Alicia Coutts	24	Australia	2012	2012-08-12	Swimming	1

Figure 2-9 – Browse, Top and Bottom

You will also notice that we have an Input button on the Properties window side pane.



	Name	Type	Size	Source	Description
1	Athlete	V_WString	255	File: C:\Users\...	
2	Age	V_String	255	File: C:\Users\...	
3	Country	V_String	255	File: C:\Users\...	
4	Year	V_String			
5	Closing Ceremony Date	Date			
6	Sport	V_String			
7	Gold	Double	8	File: C:\Users\...	
8	Silver	Double	8	File: C:\Users\...	
9	Bronze	Double	8	File: C:\Users\...	

Figure 2-10 – Browse, Input

This is the *Input* window. It tells us information about each field that comes into the tool: the field name, field type, total size of each cell, the original source of the data, and a description of the field.

2.4 CrossTab

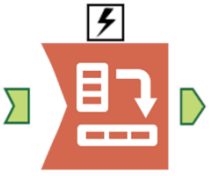


Figure 2-11 - Crosstab

The *Crosstab* tool creates a normalized (more human-readable) dataset by creating columns out of the rows of data.

Group	Input	Output
Transform	Any data stream	Data stream wider than input

Note: The *Crosstab* tool will convert all spaces and special characters to underscores in the column headers.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The Cross Tab Properties window has four components, as shown in the following figure.

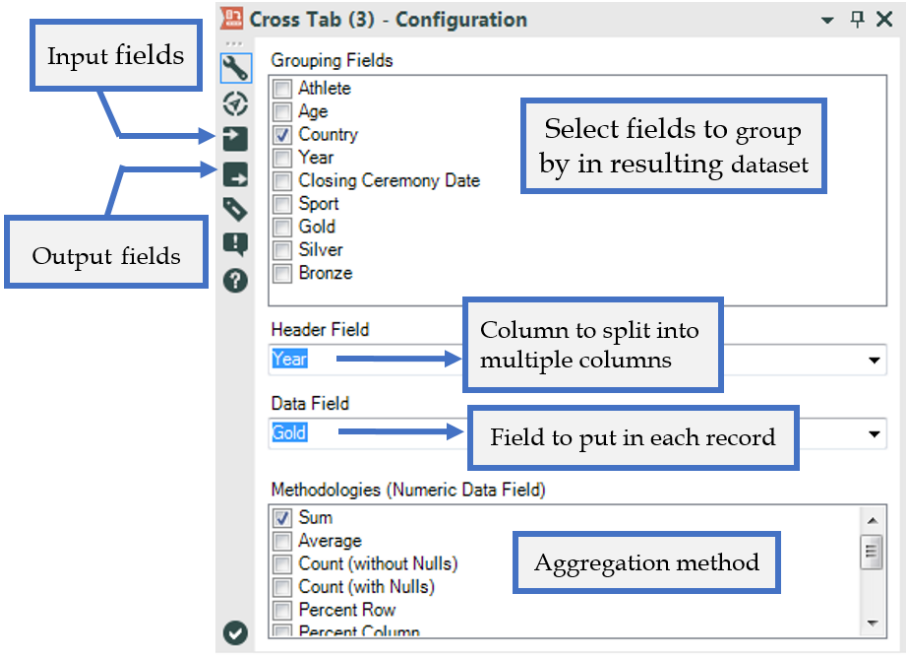


Figure 2-12 – Crosstab, configuration

We notice the grey boxes that have arrows at the top-left and bottom-right corners. These show us what the input and output fields are. These boxes are standard for tools that have input and output.

- *Grouping Fields* allows us to select which fields we want to group by in the resulting dataset. If nothing is selected, we will only get a single record as output.
- *Header Field* is the field we are splitting into multiple columns.
- *Data Field* is the field we want to put in each record for the columns created by the Header Field.

- *Methodologies* (Numeric Data Field) allows us to select the type of aggregation method used if we have multiple data entries that fit into the same cell of the resulting dataset.

2.5 Comment

Figure 2-13 - Comment

The *Comment* tool gives us the ability to write notes on our workflows to add additional information on the data stream.

Group	Input	Output
Documentation	None	None

Note: Comment is an annotative tool to help give meaning to developers using this workflow.

Properties Window:

The *Comment* Properties window allows us to customize the comment field that appears on the canvas.

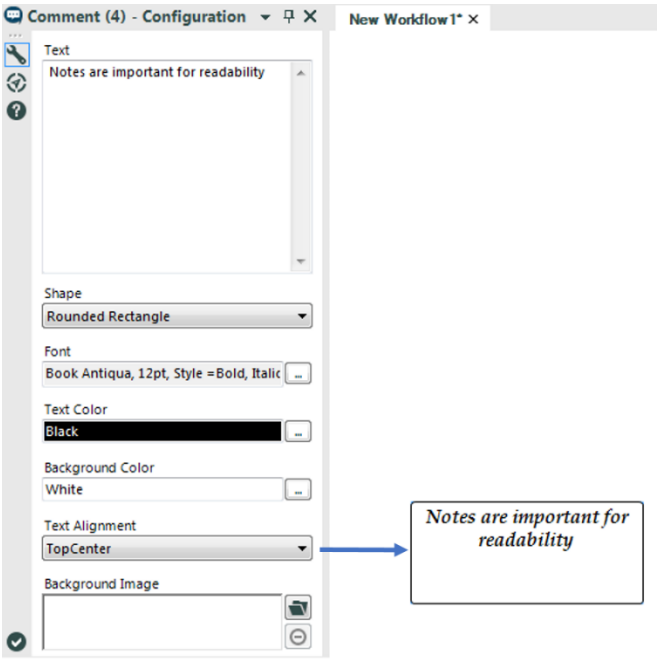


Figure 2-14 – Comment, configuration

The configuration settings in the figure above create the comment to the right of the window. We can edit text, shape, font and color of the comment background, adjust alignment of the text, or select an image to write over.

Using the settings in the *Comment* tool allows us to create easily recognizable, distinct comments throughout our data stream.

2.6 Filter

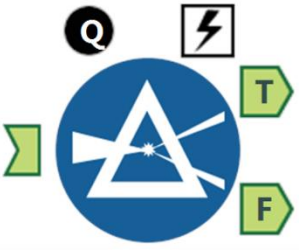


Figure 2-15 - Filter

The *Filter* tool gives us the ability to create a function that will split the data row by row into either the true or false outputs.

Group	Input	Output
Preparation	Any data stream	T & F section below

Note: The formula must evaluate to a True or False Boolean value (null evaluates to False unless the formula is looking for nulls). The formulas we create here can be arbitrarily complex, and thus can significantly slow our data stream. "//"is the comment character.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output T: This is the set of records where the formula evaluated to true.

Output F: This is the set of records where the formula evaluated to false.

A basic filter was added to the Filter tool. One can use this basic filter to quickly construct a simple query on a single field in the incoming data stream.

Properties Window:

We see that there are two different ways to create a filter: *Basic Filter* and *Custom Filter*. The regions for each are as shown below.

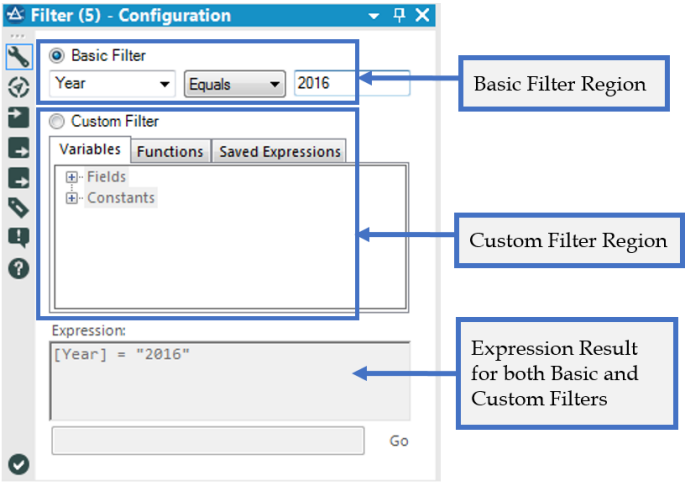


Figure 2-16 – Filter, basic filter

- *Basic Filter*: Allows us to pick a field and operator, and type in the value that the field should be compared to.

The options for the *Basic Filter* change depending on the field type. The *Basic Filter* option allows us to easily create simple filters as well as start building formulas before we know the syntax.

As we enter values into the *Basic Filter* options, it populates the Expression box at the bottom of the window. This helps us learn the associated syntax.

- *Custom Filter:* allows us to click on variables, function, and saved expressions to populate the expressions window or

type the formula directly. A sample custom filter is shown below.

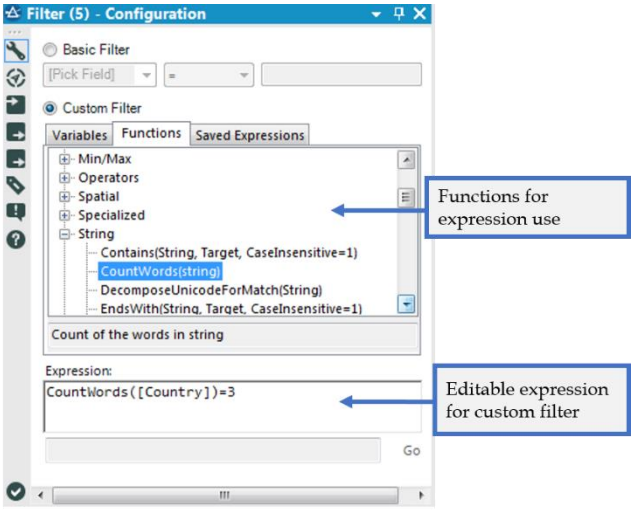


Figure 2-17 – Filter, custom filter

We will discuss creating formulas in the *Formula* tool and throughout the exercises in this book. On an error, the red Error symbol replaces the message symbol, as shown in the figure below.

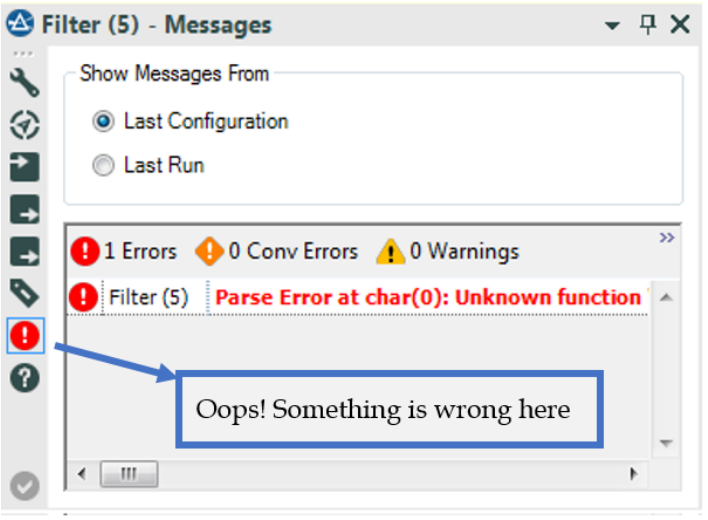


Figure 2-18 – Filter, error view

2.7 Formula

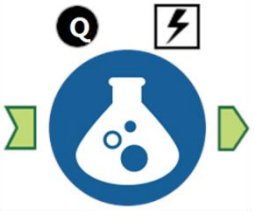


Figure 2-19 - Formula

The *Formula* tool gives us the ability to create a function that will be written to a new column in our data.

Group	Input	Output
Preparation	Any data stream	Augmented original data stream

Note: The formulas created here can be arbitrarily complex, and thus can significantly slow the data stream. Ensure the created output field has a file type that's compatible with the result being created. We can use the formulas created higher in the list in calculations lower in that list. "//" is the comment character.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with one additional field for each formula we create.

A new function is added to the Formula tool: Starts With, Ends With, and Contains.

Properties Window:

The formula window looks similar to the filter window; in fact, the expression building section here is identical to the *Custom Filter*.

The top of the *Formula Properties* window has an option to define multiple calculations, change the order the fields are created, and define the major metadata for the field that we are creating.

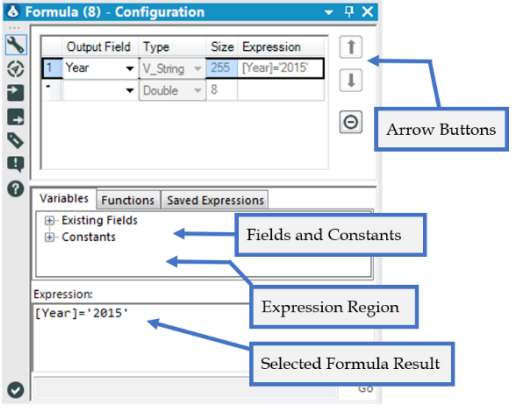


Figure 2-20 – Formula, options

The arrows on the side move the created formula up and down the list while the circle with the line in it removes the highlighted formula.

Each of these output fields will have an associated formula and will add a column to our data stream.

Under the *Variables Tab*, we can see *Fields* and *Constants*. The incoming data determines what list of Fields, and the *Environmental Variables* determine what the Constants are. (*Environmental Variables* can be defined in the workflow properties window; see Properties Window in Chapter 1 for more information.)

Under the *Functions tab*, we see a tree structure. This allows us to look for the functions needed by double-clicking on and moving them into the expressions window to work with.

The *Saved Expressions Tab* allows us to access recent and saved expressions, as well as save our current expression for later use.

2.8 Input Data

Figure 2-21 – Input Data

The *Input Data* tool gives us the ability to import data from specific databases.

Group	Input	Output
In/Out	None	Data stream in initial data format

Note: This is the most common start to a data stream. We can use full or relative file paths to files as well as database connections.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

A new option is available for relational database connections. When enabled, the data is stored in a yxdb file on disk so that data sources aren't hit repeatedly during workflow development.

Properties Window:

The *Input Data Property* window has three main components. The first is the field that shows our data connection. When connected to a data source, we can see the address of the file or database that we are connected to.

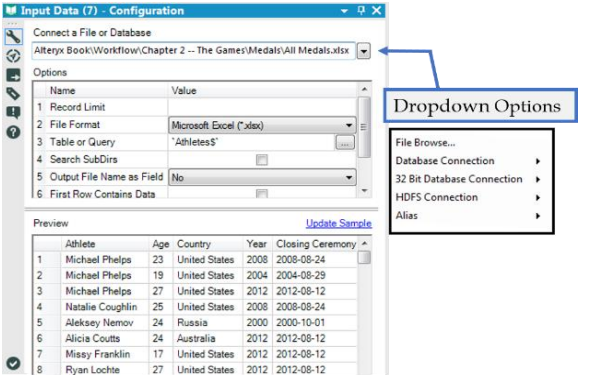


Figure 2-22 – Input Data, options

When we click on the drop-down arrow on the right-hand side of the field, we see this menu and can use this option to connect to data sources that are accessible. If present, a list of recent connections is shown, allowing us to save connections. This would appear under the Alias Link shown above.

The second main component of the window is the Options section where we can change the setting associated with the data connection to modify exactly what we are connecting to. This will allow us to modify many of the options that define the connection.

The third component is the Preview, which gives us a view of the first 100 records to help ensure that we are connected to the correct data source.

Table or Query: This option needs to be called out separately because it allows us to open a new connection window by clicking on the "…" button. That menu looks similar to this, depending on what we are connecting to.

The image below describes each of the tabs available when clicking on the "…" button.

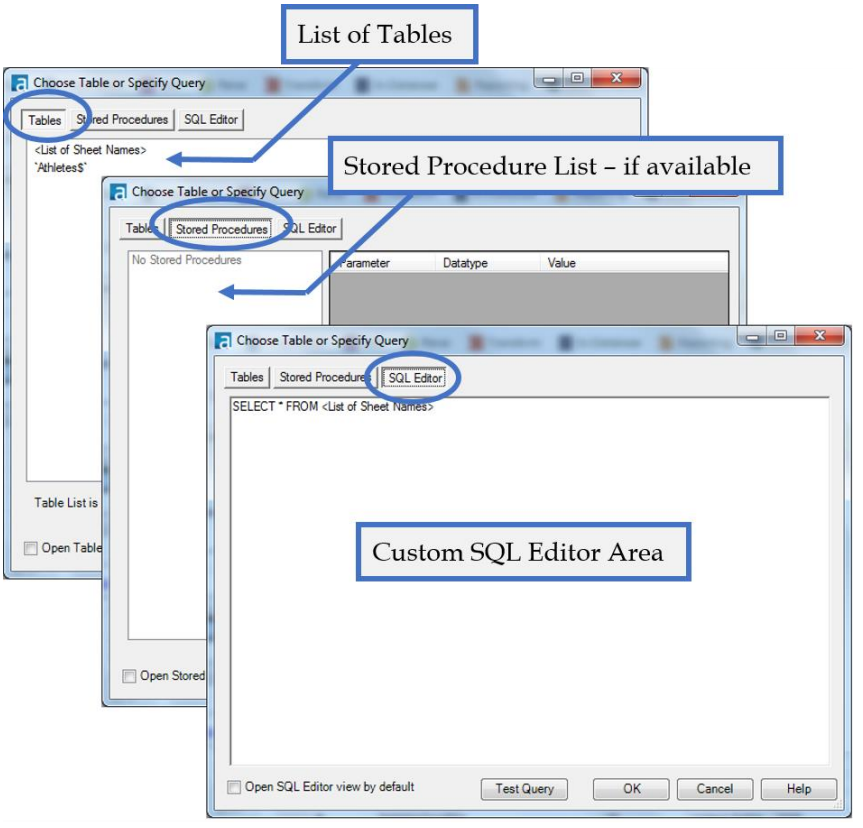


Figure 2-23 – Input Data, table and query options

2.9 Join

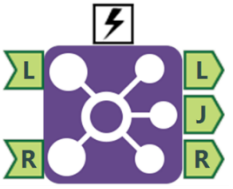


Figure 2-24 - Join

The *Join* tool gives us the ability to combine two data streams by lining up records based on matching fields.

Group	Input	Output
Join	See Input Left and Input Right	See Output Left, Output Join, and Output Right

Note: Join does not work like a join in SQL; the tool creates three groups in order to perform a left, right, or full outer join We need a *Union* tool after the join to identify which outputs should be brought together. (See example "Brains vs. Brawns.") If we have multiple fields that match each other, records are replicated from the original data stream.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input Left: A data stream with at least one common field to Input R (the fields do not need to share a name).

Input Right: A data stream with at least one common field to Input L (the fields do not need to share a name).

Output Left: Data stream containing records that did not match anything from the right input.

Output Join: Data stream containing records that match both left and right inputs. Records may be replicated as a result of this operation.

Output Right: Data stream containing records that did not match anything from the left input.

Properties Window:

The *Join Properties* window has two major components.

The top asks how we want to join the two data sets. We can join by position or by specific fields. Most of the time, we would be joining on specific fields because it allows greater control over the join.

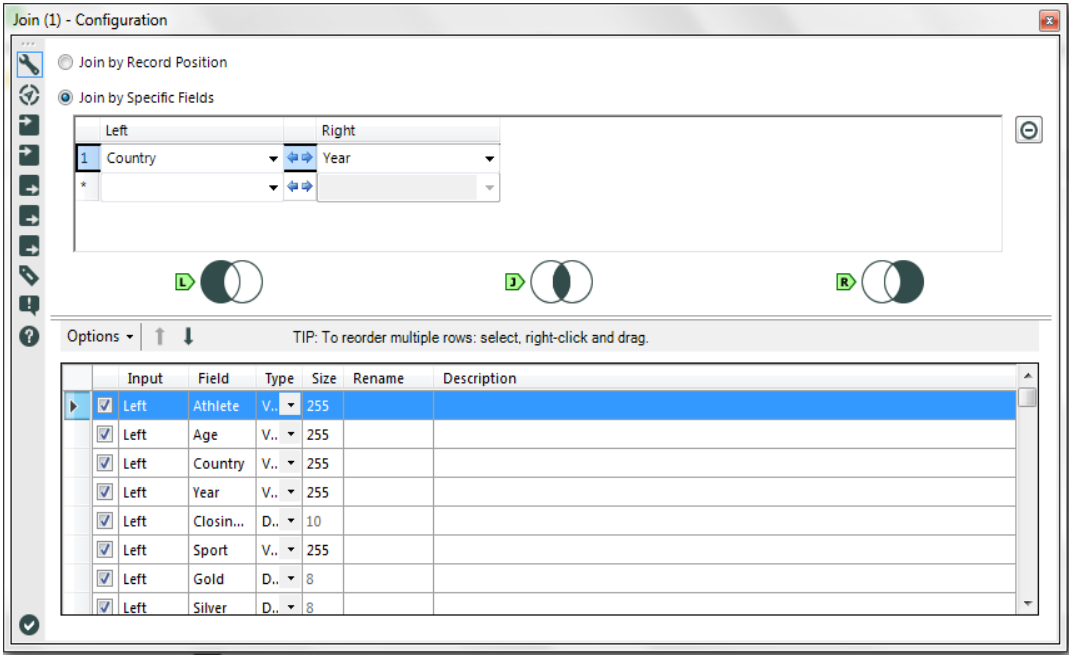


Figure 2-25 – Join Properties

The bottom allows us to define which fields will be in the output as well as define the metadata for those fields (more on this when we talk about the *Select* tool).

We can see that there are three images separating the top and bottom of the window. These Venn diagrams show us what will be in each of the three outputs. More succinctly, we can consider the image below. The two inputs are the pink and blue circles, and the three outputs are the pink, purple, and blue shaded regions.

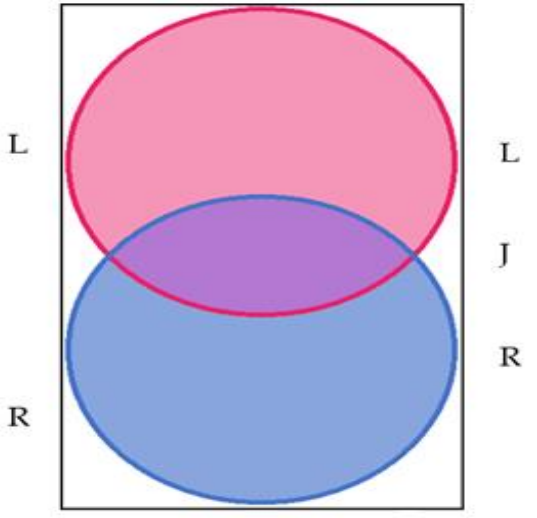


Figure 2-26 – Join Properties Venn Representation

2.10 Output Data

Figure 2-27– Output Data

The *Output Data* tool allows us to write the data stream out to a file or database

Group	Input	Output
In/Out	Any data stream	File Or Database

Note: The output window has the ability to write to files or to databases using SQL.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The top half of the *Output Data Property* window is very similar to the Input Data window. Both allow us to navigate to a file or database and set options related to the dataset.

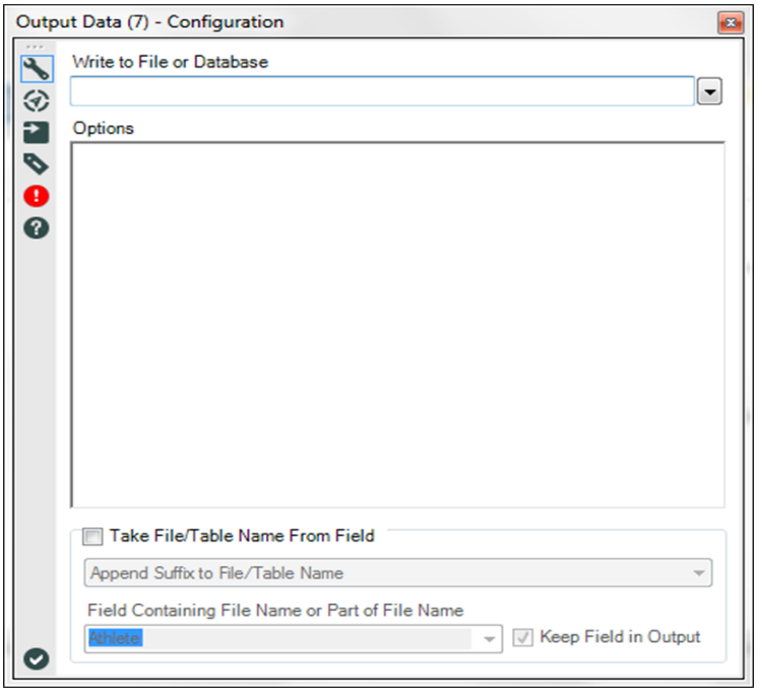


Figure 2-28 – Output Data Properties

The difference is that at the bottom, there are some special options that allow us to modify the way the metadata is written based on the incoming data stream.

We can see here that there is a file format called an "Alteryx database (*.yxdb)." Alteryx allows us to store data in files specifically designed to work well with Alteryx. These file types are native to Input and Output Data tools, so they do not need to perform any conversion to connect to these files. We can also use this file type to store both spatial and non-spatial data together.

2.11 Running Total

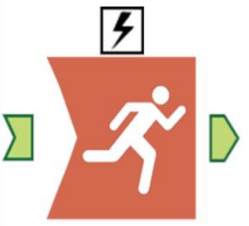


Figure 2-29 – Running Total

The *Running Total* tool allows us to create a running sum for a numeric field in the incoming data stream

Group	Input	Output
Transform	See details below	See details below

Note: Running Total produces the running sum of the data from the top of the column down, so it is important to make sure the data is properly ordered (See the Sort Tool).

Input: Any data stream with at least one numeric field.

Output: The original data stream with additional columns called *RunTot_<Original Field Name>* for each of the selected "Create Running Total" fields.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Running Total Properties* window has two components.

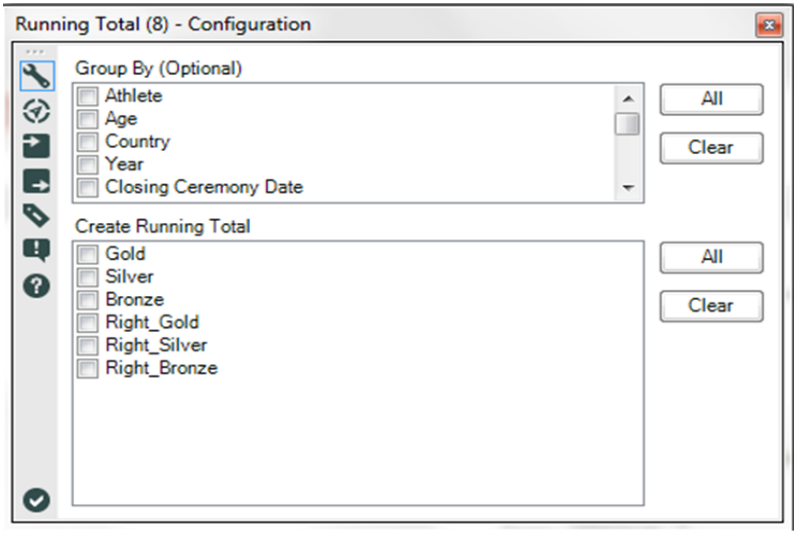


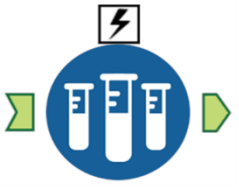
Figure 2-30 – Running Total Properties

The first is *Group By*, which allows us to define a field or set of fields that the running sum will be unique to the set of elements in our group by fields.

The second is the selection of which fields we want to create a running total on.

2.12 Sample

	The *Sample* tool allows us to work with a subset of data		
	Group	**Input**	**Output^**

 Figure 2-31 - Sample	Preparation	Any data stream	See details below

Note: This is useful for limiting the amount of data we've run through our data stream when we are testing, creating different samples of our dataset, and skipping header or footer information that may exist in our data.

Output: The original data stream with potentially modified metadata and truncated fields.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Sample Properties* window has three different settings.

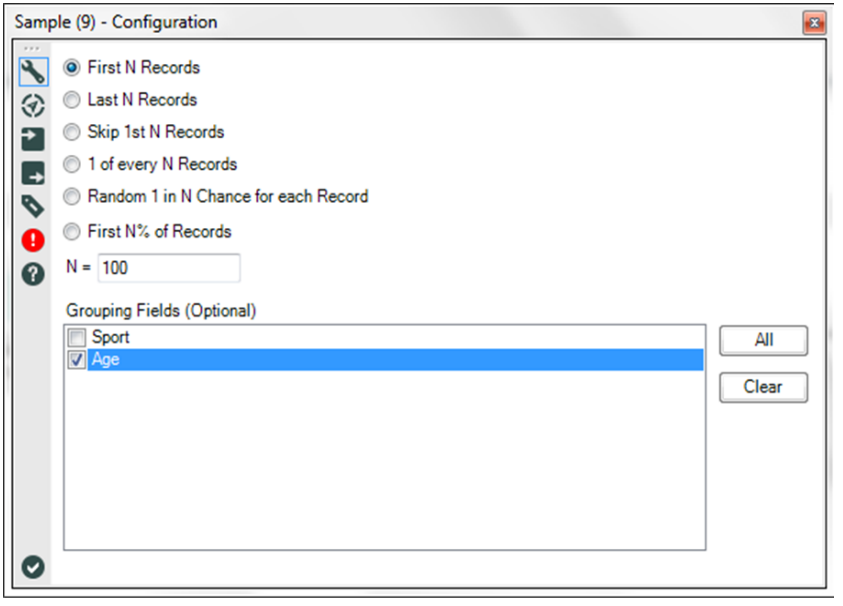


Figure 2-32 – Running Total Properties

The first of these settings is the type of sampling that is needed.

First N Records: the first N records in our data stream.

Last N Records: the last N records in our data stream.

Skip 1st N Records: all but the first N records in our data stream.

1 of every N Records: create groups of records based on the order, and take one of each of those records.

Random 1 in N Chance for each Record: Every record has a 1/N chance of being kept.

First N% of Records: the first N percent of Records in our data stream.

The second settings is 'N' to be used in the sampling.

The third is the ability to select the fields we want to group the sampling by. In the scenario pictured above, this filtering will keep the first 100 records for each date in the data.

2.13 Select

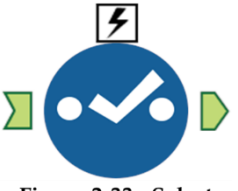


Figure 2-33 - Select

The *Select* tool allows us to modify metadata associated with the data stream, including the order of columns

Group	Input	Output
Preparation	Any data stream	See details below

Note: Select is used after every data connection and periodically throughout the data stream to ensure everything in the data stream is in the right format, named appropriately, and necessary. Use this tool to drop fields that are no longer needed to save space.

Output: The original data stream with potentially fewer fields with modified metadata.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Select Properties* window has multiple important tasks associated with the maintenance of data and metadata. The window shown below shows us a list of every field name coming in so we can reset the information, if needed.

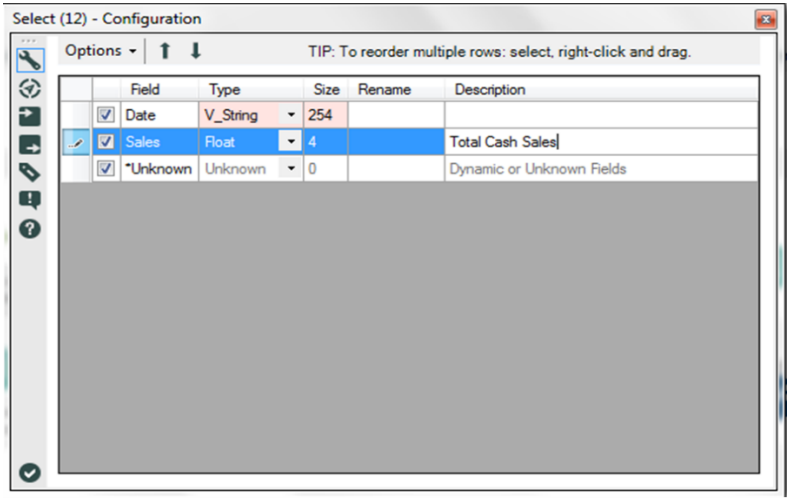


Figure 2-34 – Select Configuration

In this case, the *Date* field has been renamed to *Sales Date*, and the *Sales* field was converted to a float type and given a description saying that it is the *Total Cash Sales*. The red cells above indicate that the metadata in the cell has been modified.

There are two other things we can do with the fields without going into the options menu: The first is checking and unchecking the boxes at the left of each row in order to drop that field from the data stream. The second is reordering the fields so that the columns are in a different order downstream. This can be accomplished by selecting a field and clicking the up and down arrows or by right-clicking in the space to the left of the checkmark and dragging the field up or down in the list.

In addition to the two fields that are part of the incoming data stream, there is a special field called **Unknown* that acts like a placeholder for all new fields that come into this tool.

Select Options Menu:

The options menu allows us to systematically modify the fields.

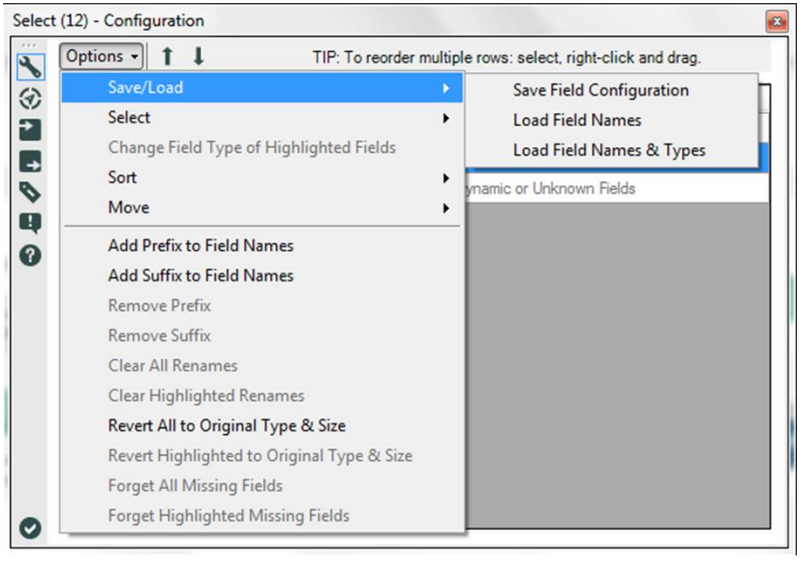


Figure 2-35 – Select Options 1

Save/Load allows us to create or load a Field Type File (.yxft). Field Type Files are a metadata file that can be used to appropriately define or redefine the columns in our data stream.

Save Field Configuration creates a new yxft file.

Load Field Names imports field names from an existing yxft file.

Load Field Names & Types imports field names and the type of field that should be allocated.

Other options are as shown in the figure below.

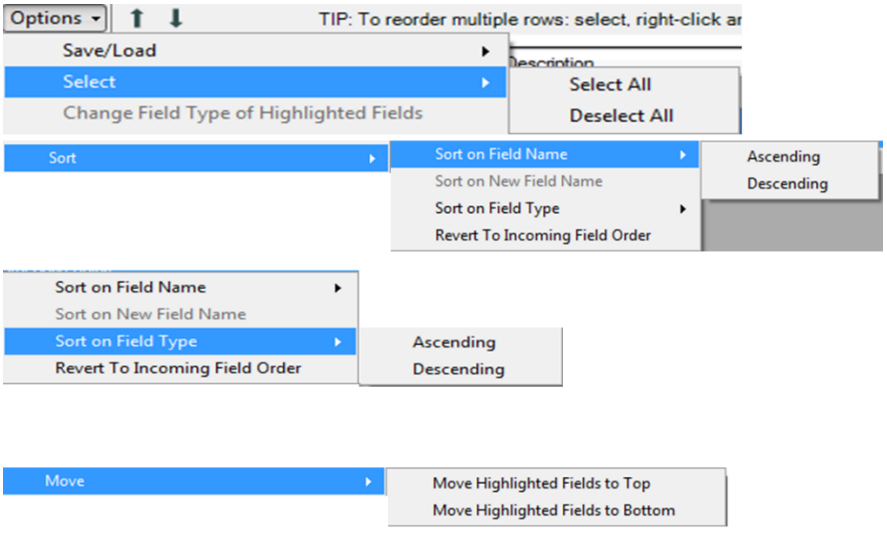


Figure 2-36 – Select Options 2

Select allows us to select or deselect all fields.

Sort has four primary methods of ordering fields.

Sort on Original Field Name will alphabetically sort our fields in either ascending or descending order based on the field names that came in from the data stream.

Sort on New Field Name will alphabetically sort our fields in either ascending or descending order based on the field names that leave the tool.

Sort on Field Type will group all fields that have the same data type together.

Revert To Incoming Field Order will clear the ordering of fields.

If we have selected fields, Move allows us to group them all at the top of the data field list.

Add Prefix to Field Names will allow us to add a prefix to all fields or all selected fields.

Add Suffix to Field Names will allow us to add a suffix to all fields or all selected fields.

Remove Prefix will allow us to remove a common prefix between selected fields.

Remove Suffix will allow us to remove a common suffix between selected fields.

Clear All Renames will remove all renaming that has been defined for this select.

Clear Highlighted Renames will remove all renaming in the highlighted (selected) fields.

Revert All To Original Type & Size will remove all changes to the field types or allocated data sizes.

Revert Highlighted To Original Type & Size will remove all changes to the field types or allocated data sizes for highlighted fields.

Forget all Missing Fields will remove this tool's metadata about fields that are no longer coming into the tool from the data stream.

Forget Highlighted Missing Fields will remove this tool's metadata about fields that are no longer coming into the tool from the data stream that are highlighted.

Type: Type is an important thing to know because each have different attributes and mean different things to other tools. Each of the types are described in Appendix D.

2.14 Sort

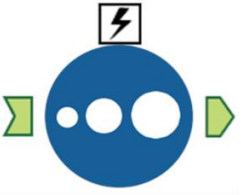


Figure 2-37 - Sort

The *Sort* tool allows us to reorder records.

Group	Input	Output
Preparation	Any data stream	See below

Note: Sorting is most important when we are working with calculations that consider multiple rows or ordering data for normalized consumption.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with records sorted in a different order.

Properties Window:

The *Sort Properties Configuration* window allows us to select one or more fields by name, and Ascending or Descending for each to determine an order to our records. We can change the order of these sorts by moving them up and down the list.

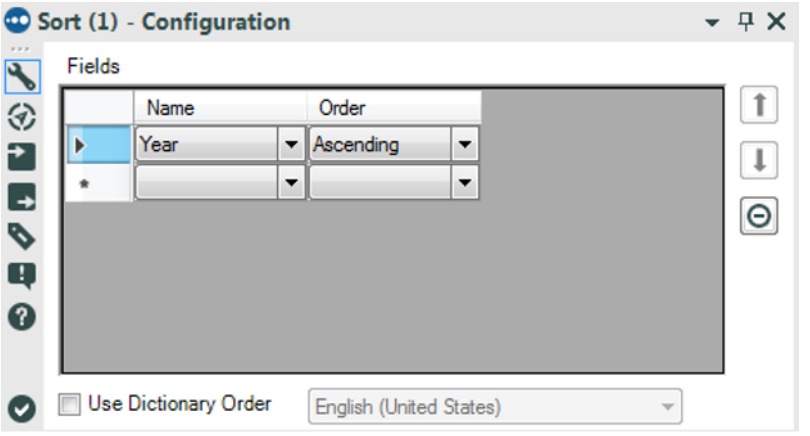


Figure 2-38 – Sort Configuration

By checking the *Use Dictionary Order* option, we can select the dictionary order that should be applied to sort the data when appropriate.

2.15 Summarize

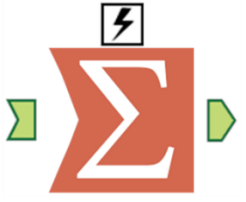


Figure 2-39 - Summarize

The *Summarize* tool allows us to summarize data in our data stream.

Group	Input	Output
Transform	See below	See below

Note: When summarizing data, it may be necessary to reexamine the underlying calculations because aggregating those calculations may not make sense. Summarizing a single field using Group By is a good way to get a unique list of the data. Running complex analysis like geocoding is often more efficient to do on a summarized list and then join back onto the full dataset.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with too granular a level of detail.

Output: A summarized data stream with a less granular level of detail.

Properties Window:

The *Summarize Configuration* window has two basic components.

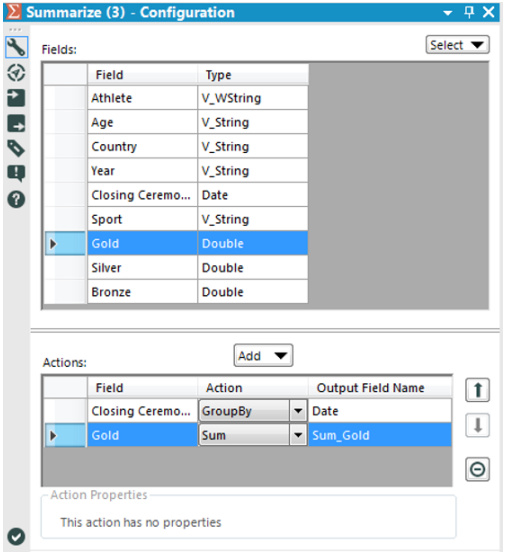


Figure 2-40 – Summarize Configuration

The *Fields* list shows each of the incoming fields, and the *Actions* list shows the fields created in this tool.

The select button at the top-right allows us to select tools in a systematic way so that if we want to take the sum of all of our Numeric fields, we can select them all and add them in one step.

When we have something in the fields list selected, we can click on the *Add* button, which shows the drop-down menu shown above. It lists every operation that can be used to aggregate the data using *Summarize*.

We will not be going through this menu in detail, as it is a list of aggregation methods; however, *Group By* needs to be given special attention.

When we use *Group By* on fields in the summary, we will end up with one line item for each combination of field elements

that we grouped by. If this is unclear, it should make more sense as we go through exercises.

2.16 Tool Container

Figure 2-41 – Tool Container

The *Tool Container* tool allows us to group tools together for clarity and allows the tools to be disabled when unnecessary.

Group	Input	Output
Documentation	None	None

Note: Click and drag tools onto the box to put them into the tool container. Tool containers make it much easier to navigate our data stream because they allow us to consider a series of tools as a single unit. If we click on the arrow at the top-right corner, it will collapse the box without disabling it.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Tool Container Configuration* window allows us to customize the text and formatting of the container as well as disable the tools inside it.

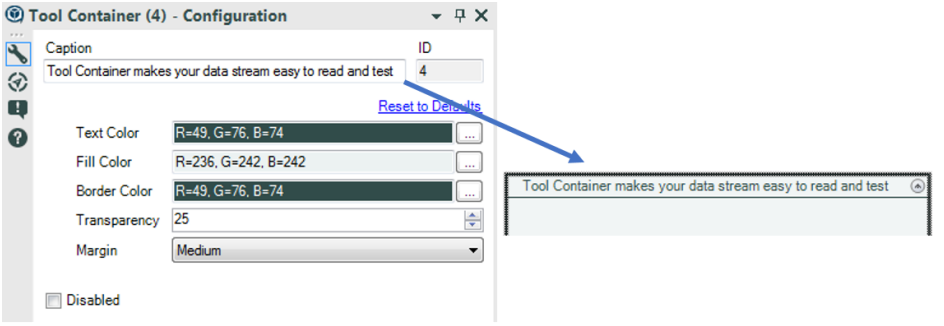


Figure 2-42 – Tool Container Configuration

The *Disabled* option allows us to turn off sections of our data stream. This is typically used in testing data and application building.

2.17 Transpose

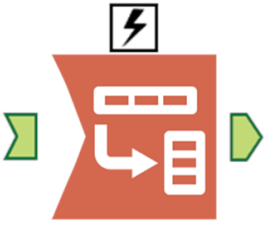


Figure 2-43– Transpose

The *Transpose* tool allows us to de-normalize data

Group	Input	Output
Transform	See below	See below

Note: This tool converts all spaces and special characters in the titles into underscores in each record.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with multiple fields that need to be combined into rows.

Output: A taller data stream because records have been duplicated in order to consolidate the columns. The columns consolidate into two columns; *Name*, which is a column with each of the former column names in it, and *Value*, which is a column with each of the data values.

Properties Window:

The *Transpose Configuration* window has three elements, as shown in the figure below.

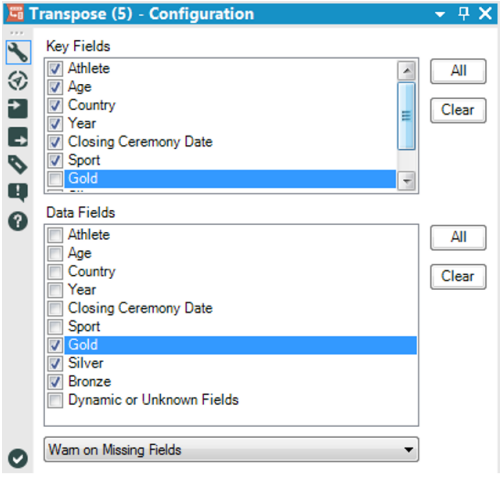


Figure 2-44 – Transpose Configuration

Key Fields allows us to select the fields that will be maintained after the transposition.

Data Fields allows us to select the fields that will be combined following the transposition.

The drop-down at the bottom allows us to change the message behavior if there are missing fields in the incoming data stream.

2.18 Union

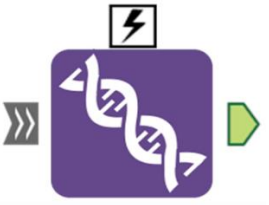


Figure 2-45 - Union

The *Union* tool allows us to append records together one after another from multiple data sources.

Group	Input	Output
Join	See below	See below

Note: The order that we connect data stream to this tool's input will determine the default order that they are combined. In this tool, naming the incoming connections is often helpful.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Multiple data streams that should be combined by adding the records from one set to the end of the others.

Output: A data stream that has the records from multiple data streams.

Properties Window:

The *Union Configuration* window has three core elements.

The first drop-down allows us to change the method that is used to align the columns from the different inputs.

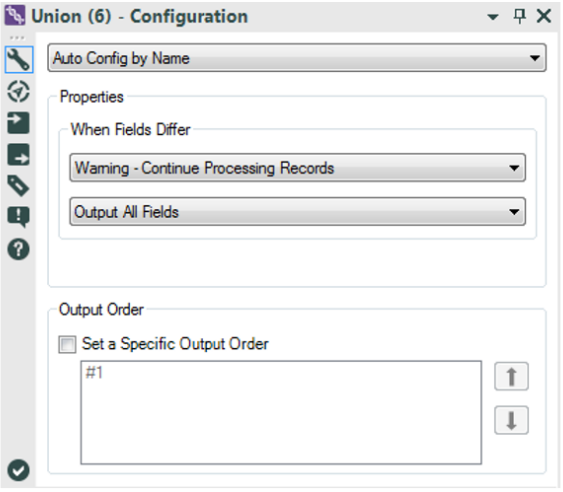


Figure 2-46 – Union Configuration by Name

Auto Config by Name makes the union align the fields that have the same name. This is best used if we know that our data will always be named the same way.

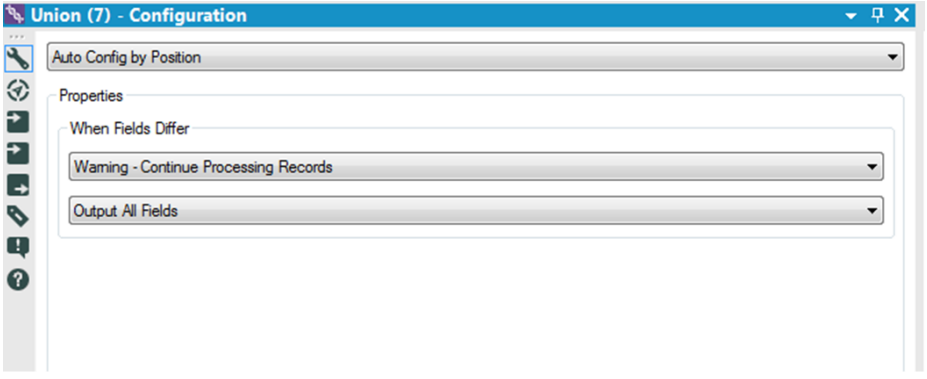


Figure 2-47 –Union Configuration by Position

Auto Config by Position makes the union align the fields by the column number. This is best used if we know our data will always be in the same order but may have different (or no) field names.

In both of these options, we see the same Properties section asking what should happen when the fields differ. The first drop-down allows us to change the behavior between an error, a warning, and nothing. The second drop-down allows us to decide if we want all fields or only the fields matched from all outputs to be in the output.

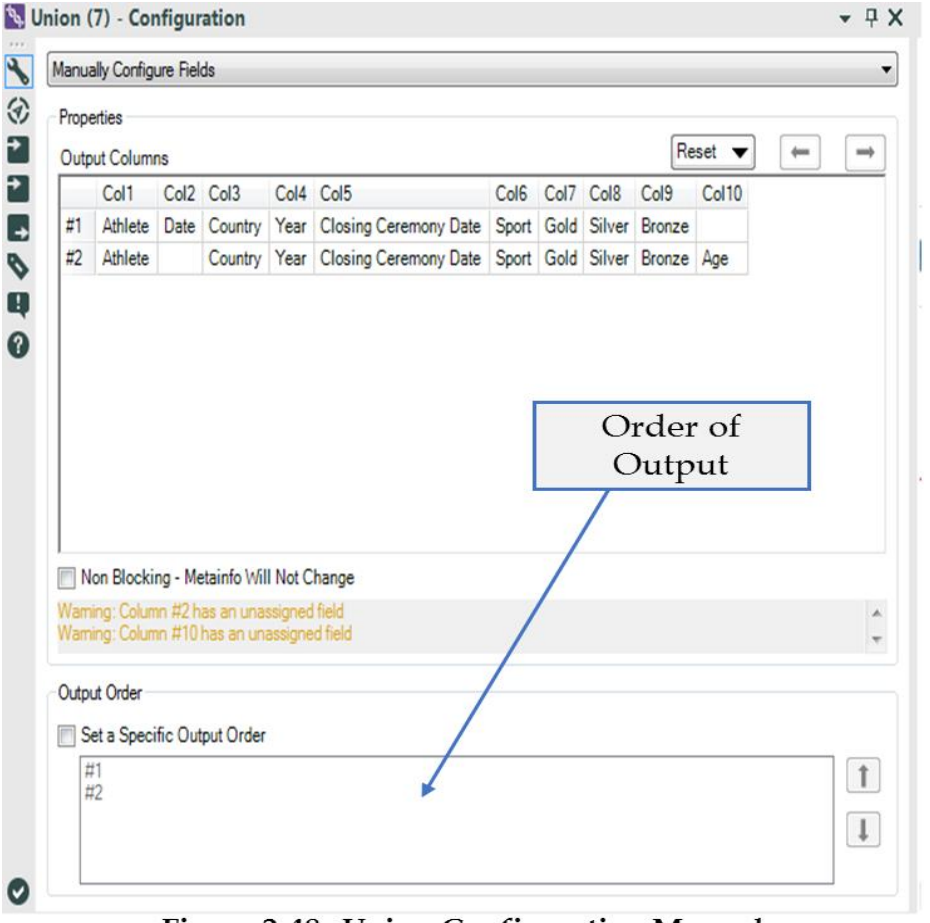


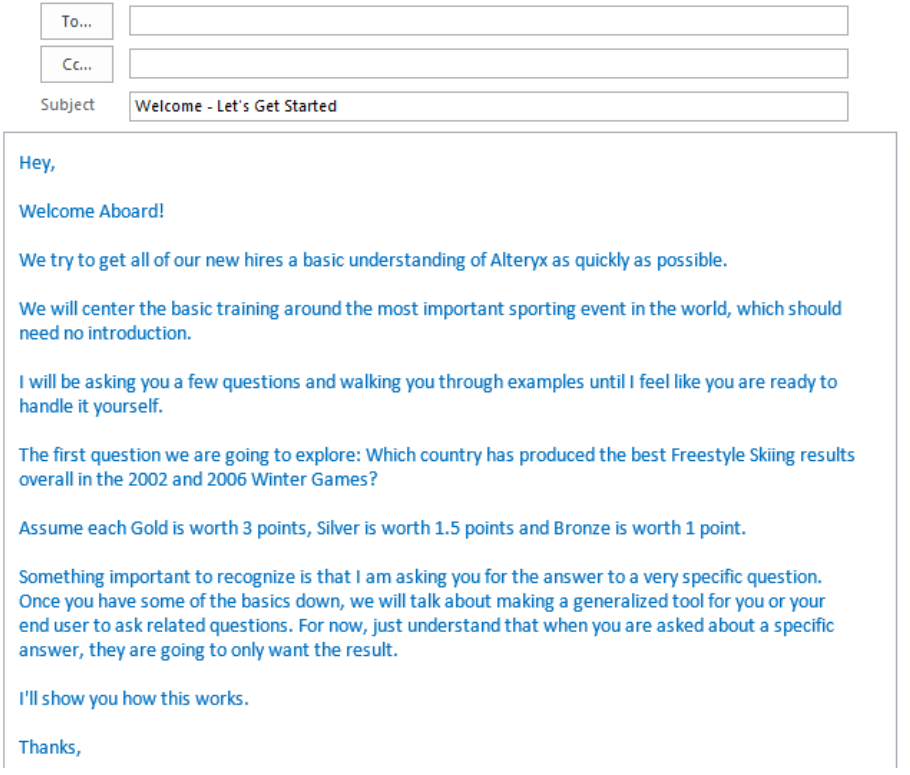
Figure 2-48 –Union Configuration Manual Configuration

Manually Configure Fields is the third option in the first drop-down. It allows us to select exactly which fields to be

brought together by manipulating the Output Columns portion of the configuration window. This is best used if we know that our fields may be named differently or may be in different order.

The final component is the *Output Order*. It allows us to set the order of the records by choosing the order that the data streams are combined.

2.19 Freestyle



Let's start building a workflow that will answer our question. We are going to start with a blank canvas and save it as Freestyle Skiing. Next, bring an Input Data tool so that we can connect to data.

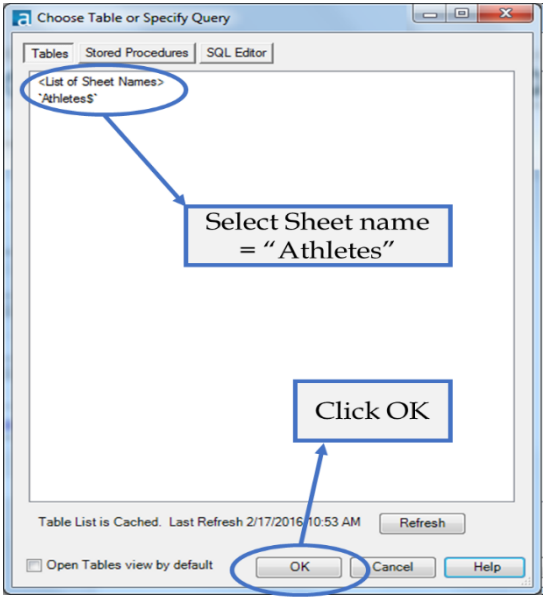


Figure 2-49 – Freestyle Skiing, Data Input

Now navigate to where data files are unpackaged, and connect to the file *All Medals.xlsx* in *Chapter 2 -- The Games*> *Medals*. For downloading the data associated with this book, please refer to the letter to the reader on this chapter's first page.

When connected, we see this window pop up. Click on *Athletes$* then OK to connect to the Athletes sheet in the All Medals Excel file. This is shown in the figure above.

Best practices are to put a select and a browse after every input.

- Browse helps us check the data at the time of import. This ensures that the data we are getting is correct.

- Select allows us to make sure that the fields are in the right format from the beginning.

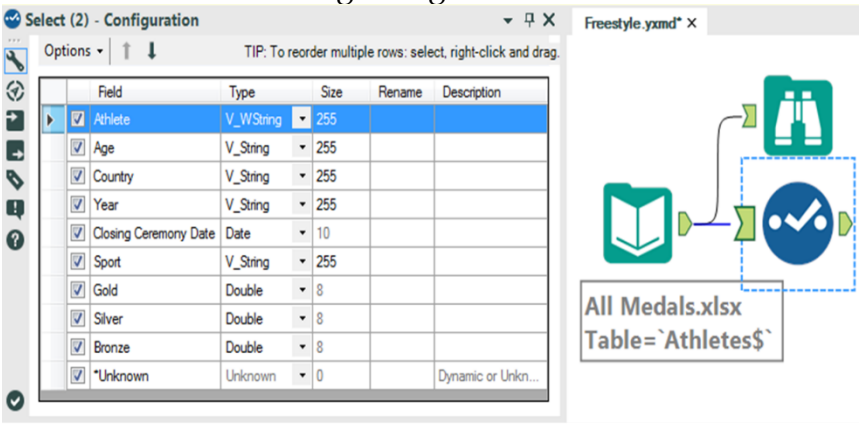


Figure 2-50 – Freestyle Skiing, Select configuration

If we click on Select, we should see that our fields are in different types than the above image. Change them to match what is shown.

Now that we have the data and the fields are the right type, the first thing we should do is filter the data. We always want to *limit the data as soon as possible*, since this will speed up our data stream and prevent memory errors by limiting the information.

Best practice is to remove data as soon as it is no longer needed.

It makes sense that the first step in filtering would be to bring in the Filter tool; however, if we are not familiar with the data set and we have not run it, we may not have enough information to filter properly. In this case, we want to run the module so that there's data in the Browse tool for us to work with.

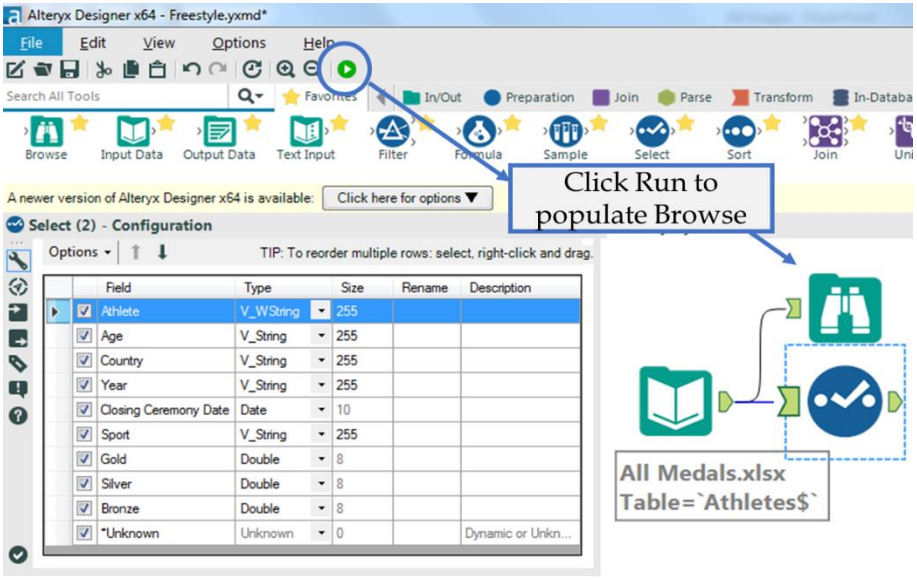


Figure 2-51 – Freestyle Skiing, populate Browse

We can see that when the module finishes running, we get a pop-up window that lets us know how long it took to run and if there were errors.

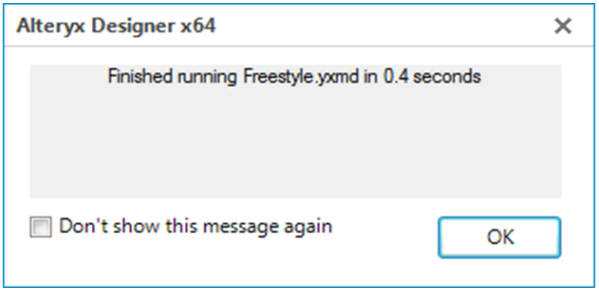
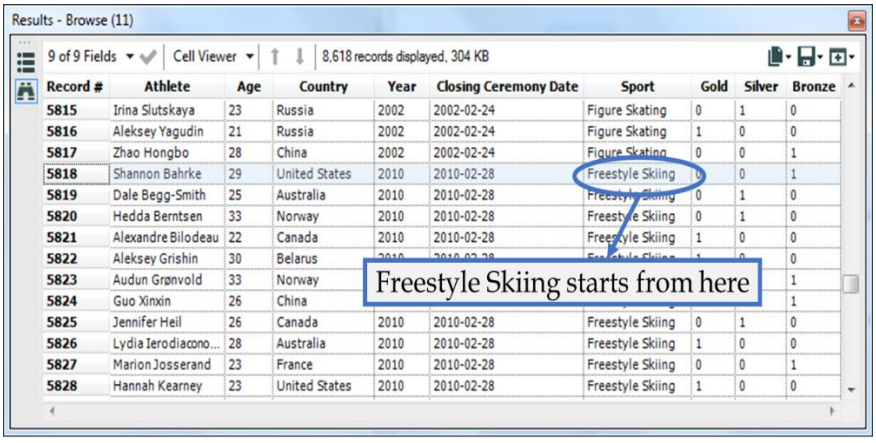


Figure 2-52 – Freestyle Skiing, message after Running

Feel free to click on the *Don't show this message again* check box before closing if the pop-up window is distracting.

Now, we can start thinking about the filter. We know that we are only interested in freestyle skiing results for the 2002 and

2006 games. So the first thing we are going to filter is the sport to "freestyle skiing." If we look at the browse tool, we first see freestyle skiing at row 5818, identified by the string *Freestyle Skiing*.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Gold	Silver	Bronze
5815	Irina Slutskaya	23	Russia	2002	2002-02-24	Figure Skating	0	1	0
5816	Aleksey Yagudin	21	Russia	2002	2002-02-24	Figure Skating	1	0	0
5817	Zhao Hongbo	28	China	2002	2002-02-24	Figure Skating	0	0	1
5818	Shannon Bahrke	29	United States	2010	2010-02-28	Freestyle Skiing	0	0	1
5819	Dale Begg-Smith	25	Australia	2010	2010-02-28	Freestyle Skiing	0	1	0
5820	Hedda Berntsen	33	Norway	2010	2010-02-28	Freestyle Skiing	0	1	0
5821	Alexandre Bilodeau	22	Canada	2010	2010-02-28	Freestyle Skiing	1	0	0
5822	Aleksey Grishin	30	Belarus	2010	2010-02-28	Freestyle Skiing	1	0	0
5823	Audun Grønvold	33	Norway	2010	2010-02-28	Freestyle Skiing	0	0	1
5824	Guo Xinxin	26	China	2010	2010-02-28	Freestyle Skiing	0	0	1
5825	Jennifer Heil	26	Canada	2010	2010-02-28	Freestyle Skiing	0	1	0
5826	Lydia Ierodiacono...	28	Australia	2010	2010-02-28	Freestyle Skiing	1	0	0
5827	Marion Josserand	23	France	2010	2010-02-28	Freestyle Skiing	0	0	1
5828	Hannah Kearney	23	United States	2010	2010-02-28	Freestyle Skiing	1	0	0

Figure 2-53 – Freestyle Skiing, Browse configurations

This is the crucial piece of information we didn't have before. Now that we know exactly what we need to look for in our data so we can create the filter.

Drag a Filter tool after the Select, and make sure there is a connection between the Select output and the Filter input. This time, we will use the Basic Filter builder. Set the field drop-down to *Sport*, and type *Freestyle Skiing* into the text box like we see below.

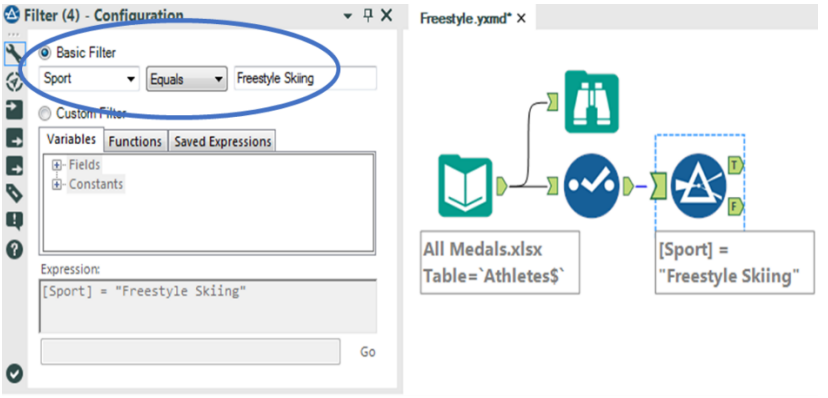


Figure 54 – Freestyle Skiing, Browse configurations

Notice that the Expression says **[Sport] = "Freestyle Skiing."** This is because field names are in brackets, and string values are in quotes. What is happening here is that for each record, we test to see if the value in Sport in exactly *Freestyle Skiing*. If it is, then True; if it isn't, then False.

The next thing we want to do is create a filter to keep only 2002 and 2006. If we look at the Select tool on the previous page, we will see that the *Year* field is a string. This is fine; we need to remember that when we are writing the filter formula. Drag a new Filter tool onto the canvas, and make sure that the first Filter's true (*T*) output is connected to the new Filter's input.

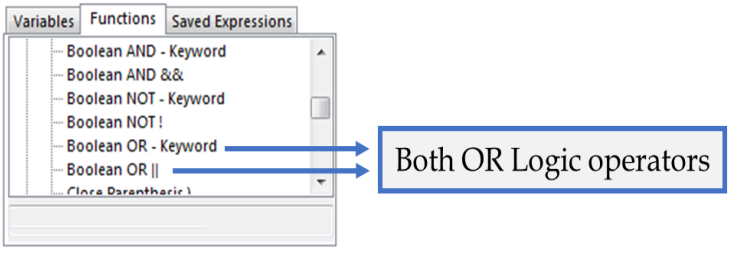


Figure 2-55 – Freestyle Skiing, Filter configurations - Functions

This time, we are going to create the filter logic on our own using the Custom Filter option. We know from the previous filter that **[<Field Name>] = "<Value>"** is the syntax for filtering a string field, so creating the first half of the filter is not hard – it is **[Year] = "2002"** – but what we need to do now is make sure 2006 is also kept.

One way we could do this is by using logical operators. *Logical operators* are terms that allow us to combine two or more Boolean (true or false) values to create a single Boolean from the two. The three Boolean operators that we will be discussing are *and*, *or*, and *not*.

AND: if **both** the value to the **left and** the value to the **right are true, then true**.

OR: if **either** the value to the **left or** the values to the **right or both** are **true, then true**.

NOT: **if** a value is **true** then **false**; **if** the value is **false,** then **true**.

We can see in the *Functions Tab* that we have the option to use the *Boolean OR - Keyword* or the *Boolean OR* ||. There is no computational difference between using the keyword or the double vertical bar symbol. Both options are available for our convenience. For those not used to programming, the keyword *OR* is much easier to remember and use, but for those who program, double vertical bars (||) is a common standard they may be used to.

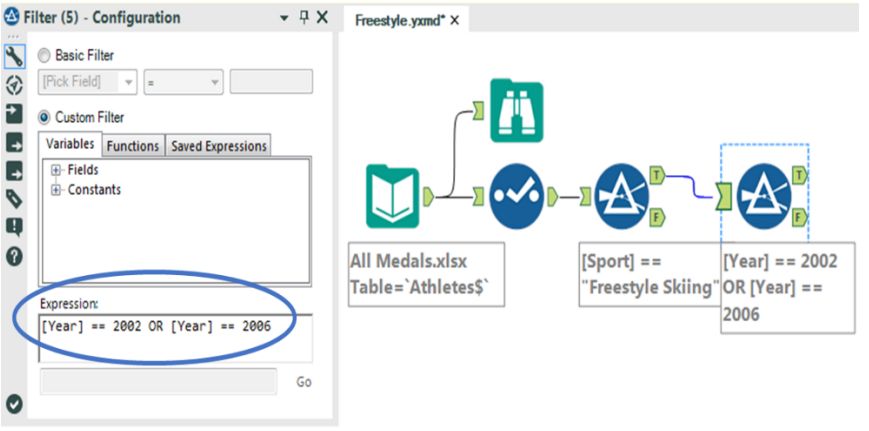


Figure 2-56 – Freestyle Skiing, Custom filter

See Appendix F for examples of Boolean logic.

Now that we know about logical operators, we can finally finish putting the filter together. We can use the formula **[Year] = "2002" OR [Year] = "2006"** in order to filter out this data.

Note: We could have combined both of these filters together by using the following: **[Sport] = "Freestyle Skiing" AND ([Year] = "2002" OR [Year] = "2006")**

The parentheses allow us to change the order of operations so that this formula reads "Freestyle Skiing in the years 2002 or 2006" instead of "Freestyle Skiing in 2002 or anything in 2006."

We now have removed all of the information we don't need in order to answer the question. But we have the problem that the data is too granular. We know who the athlete was and in which year they won their medal(s). We should bring in a Summarize Tool in order to bring the data up to the country level. If we add Country using Group By, and Gold, Silver, and Bronze using Sum, we will get a list of countries and their total medal

count for Freestyle skiing for 2002 and 2006. Place a Browse Tool, and run the module to see what we have so far.

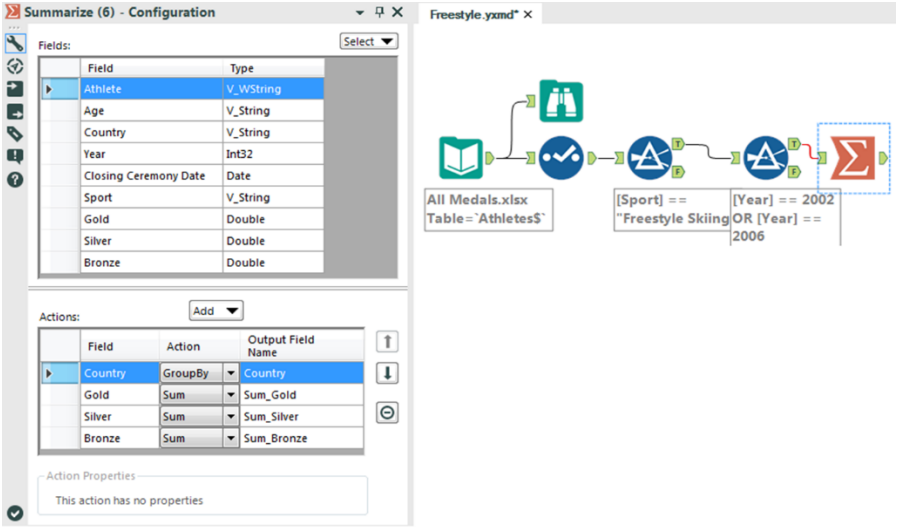


Figure 2-57 – Freestyle Skiing, Summary configuration

Best practice is to place a browse tool after every tool that transforms data into a significantly different shape. Summarize is one of the tools.

Results - Browse (12)

4 of 4 Fields | Cell Viewer

Record #	Country	Sum_Gold	Sum_Silver	Sum_Bronze
1	Australia	2	0	1
2	Belarus	0	1	1
3	Canada	1	1	1
4	China	1	1	0
5	Czech Republic	1	0	0
6	Finland	1	1	0
7	France	0	0	2
8	Japan	0	0	1
9	Norway	1	1	0
10	Russia	0	0	1
11	Switzerland	1	0	0
12	United States	0	3	1

Figure 2-58 – Freestyle Skiing, Browse configuration after summarize

We can see that we have four columns with the total counts of gold, silver, and bronze medals listed for each of the 12 countries that won freestyle skiing medals during 2002 and 2006.

Notice the fields are titled *Sum_* followed by the original field name. Alteryx is making sure we know the method used to summarize the data.

The next thing we need to do is determine which country was the *best*. If we look back at the email, we can see that *best* is defined as a function of the medals won; 3 points for gold, 1.5 points for silver and 1 point for bronze.

Bring a Formula Tool onto the canvas following the Summarize tool, and we are going to create a calculation called Score that has the Type Double, with the formula [Score] = 3*[Sum_Gold] + 1.5*[Sum-Silver] + [Sum_Bronze]

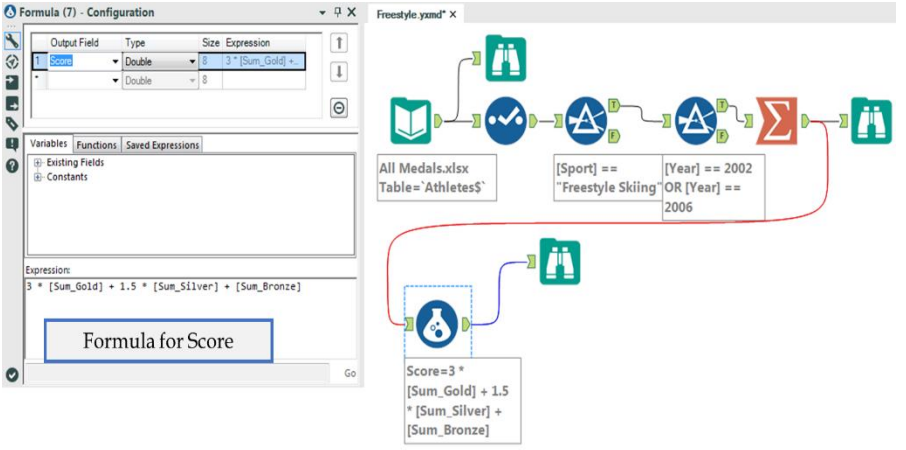


Figure 2-59 – Freestyle Skiing, Formula configuration

We can now add another Browse after the Formula Tool to see what the data looks like.

Results - Browse (13)

5 of 5 Fields | Cell Viewer

Record #	Country	Sum_Gold	Sum_Silver	Sum_Bronze	Score
1	Australia	2	0	1	7
2	Belarus	0	1	1	2.5
3	Canada	1	1	1	5.5
4	China	1	1	0	4.5
5	Czech Republic	1	0	0	3
6	Finland	1	1	0	4.5
7	France	0	0	2	2
8	Japan	0	0	1	1
9	Norway	1	1	0	4.5
10	Russia	0	0	1	1
11	Switzerland	1	0	0	3
12	United States	0	3	1	5.5

Figure 2-60 – Freestyle Skiing, Browse configuration after formula

We see there is a new field called Score that is an unordered dataset and with multiple unnecessary values. We can also see that Australia has the highest score and therefore is the answer to the original question. But for good practice, we are

going to continue to build this workflow so that no interpretation is needed.

This process is going to take four steps:

1. Reorder the data based on the score field.
2. Select only the top-scoring country.
3. Remove all data other than the name of the best country.
4. Browse that data.

Like we discussed, the Sort Tool is how we reorder the data and will be our first step. We will set up our data in a descending order based on Score, like we see here.

Figure 2-61 – Freestyle Skiing, Sort configuration

Next, we need only the first record, so we are going to use the Sample Tool to keep only the Top 1 Record coming out of the Sort.

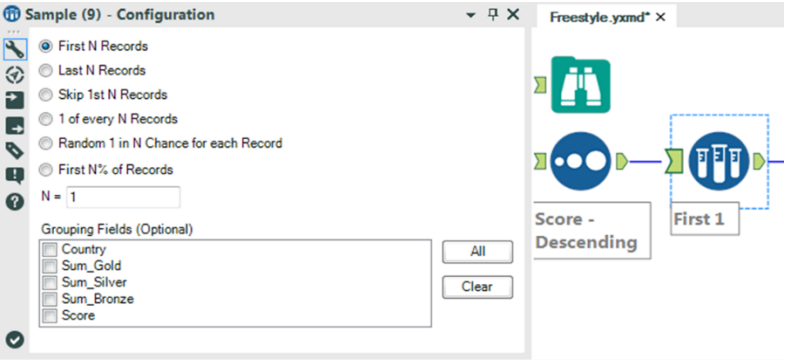


Figure 2-62 – Freestyle Skiing, Sample configuration

We know we have fields we no longer need, so we can use a Select Tool to eliminate everything that is not the country name.

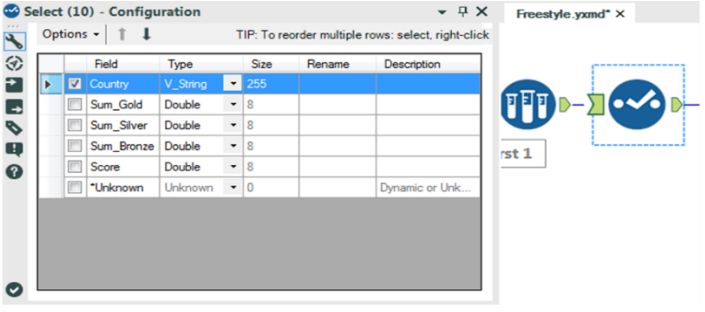


Figure 2-63 – Freestyle Skiing, Selection after Sample

Finally, we can put a browse tool at the end and run the workflow to see the results.

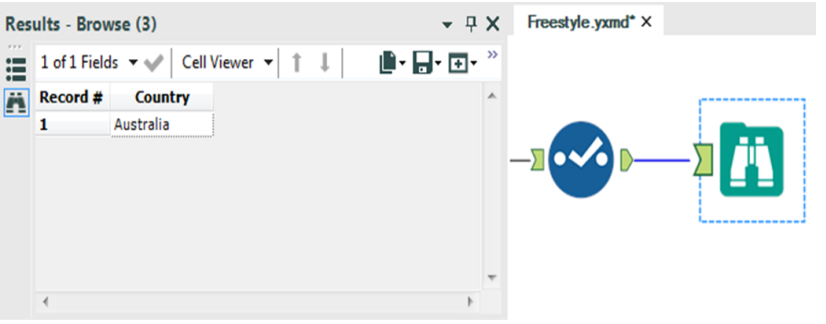


Figure 2-64 – Freestyle Skiing, Browse after Select

We could have stopped when we first saw Australia had the highest score in the previous browse tool. The reason we did not, is that when we are performing an analysis, we want our results to be perfectly repeatable. If we had interpreted the previous browse tool incorrectly then there would be no way of finding out why an error occurred. This is a problem because it makes the individual analyst entirely responsible for the answer, and anyone who checked the results could easily find the correct answer where we mistakenly picked the wrong one. Finishing the workflow in this way affords us two benefits:

- We would have a second verification that the answer was what we expected.
- Repeatability of the result so we can point to a single issue in the data preparation process that needs fixing instead of not being able to fix it at all.

Here is how the workflow would look like when complete:

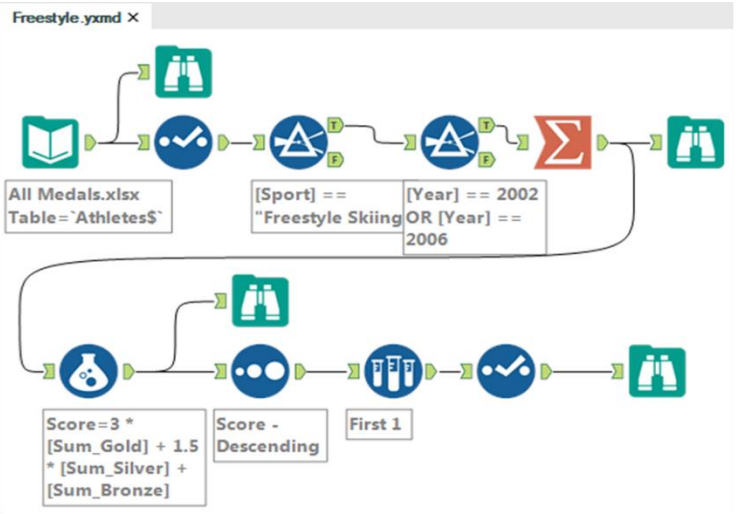
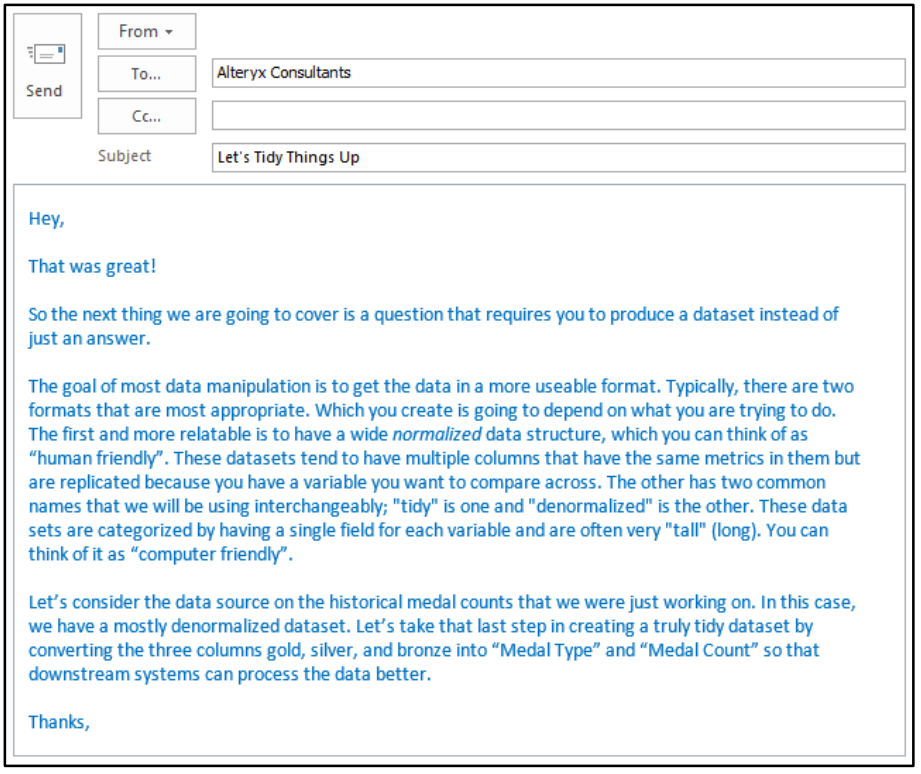


Figure 2-65 – Freestyle Skiing, Data stream after completion

2.20 Let's Tidy Things Up



This process must include at least four steps:

1) Import the dataset.
2) Transpose the dataset.
3) Make sure the fields are named correctly.
4) Export the dataset.

However, we are going to make the data cleaner and employ best practices. So our process is:

1) Import the data.
2) Browse the data.
3) Make sure the data has the right type.
4) Transpose the data.
5) Browse the restructured data.
6) Make sure the fields are named correctly.
7) Remove records that say there was no medals won.
8) Browse the data that will be exported.
9) Export the dataset.

Let's create a new workflow and save it as *Let's Tidy Things Up*.

We need to import the same data that we used in the last example. Bring an input tool onto the canvas, navigate to where we saved this book's data, and connect to the file *All Medals.xlsx* in *Chapter 2 -- The Games*> *Medals*.

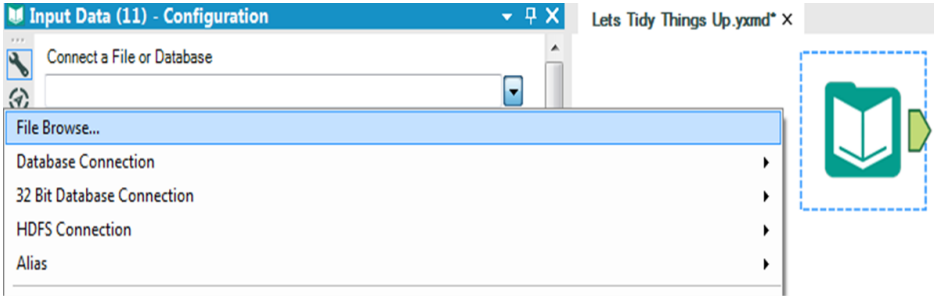


Figure 2-66 – Let's Tidy Things Up, Data Input

Now we will put a Browse and select statement following the Input Tool.

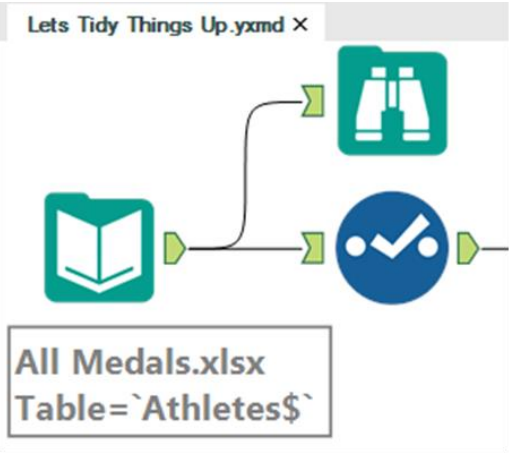
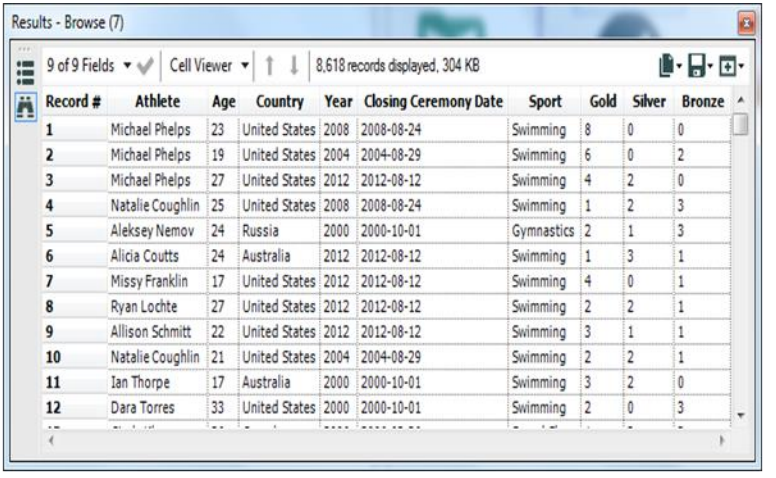


Figure 2-67 – Medals data

All of the fields are in appropriate types for what we are trying to do, so we can move directly to the transposition.

Let's run the Module to see how the data is structured.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Gold	Silver	Bronze
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	8	0	0
2	Michael Phelps	19	United States	2004	2004-08-29	Swimming	6	0	2
3	Michael Phelps	27	United States	2012	2012-08-12	Swimming	4	2	0
4	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	1	2	3
5	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	2	1	3
6	Alicia Coutts	24	Australia	2012	2012-08-12	Swimming	1	3	1
7	Missy Franklin	17	United States	2012	2012-08-12	Swimming	4	0	1
8	Ryan Lochte	27	United States	2012	2012-08-12	Swimming	2	2	1
9	Allison Schmitt	22	United States	2012	2012-08-12	Swimming	3	1	1
10	Natalie Coughlin	21	United States	2004	2004-08-29	Swimming	2	2	1
11	Ian Thorpe	17	Australia	2000	2000-10-01	Swimming	3	2	0
12	Dara Torres	33	United States	2000	2000-10-01	Swimming	2	0	3

Figure 2-68 – Let's Tidy Things Up, Browse configuration

The Transpose tool takes normalized data and de-normalizes it. If we take the data stream coming out of *Select* and pass it into a Transpose, we can make the data tidier.

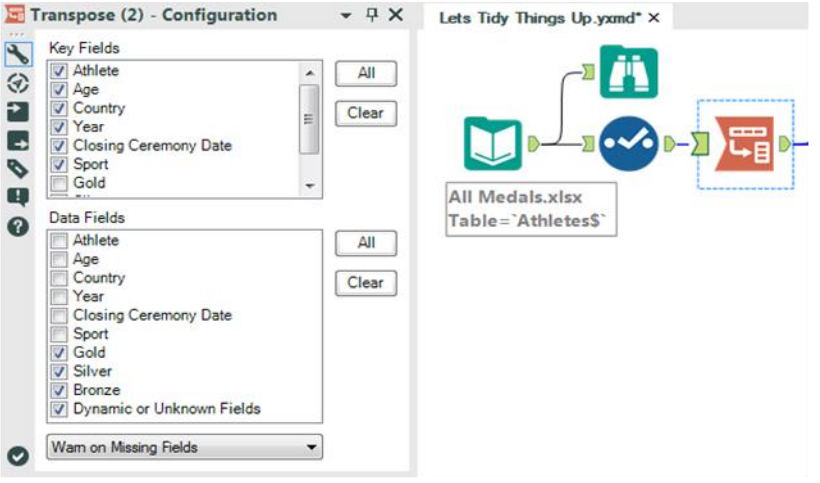
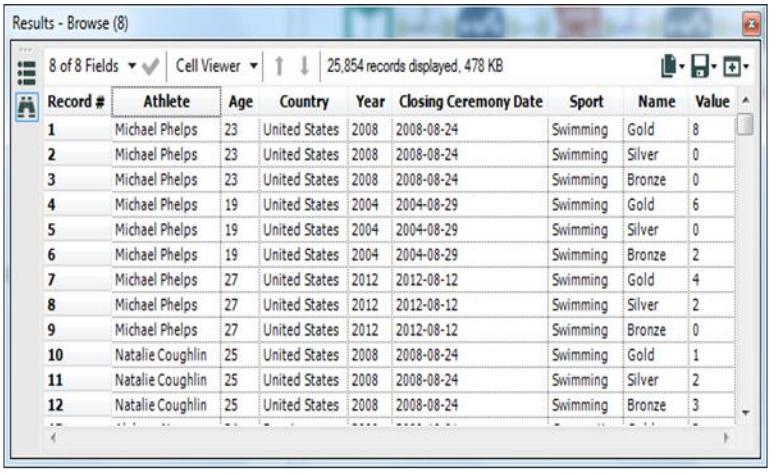


Figure 2-69 – Let's Tidy Things Up, Transpose configuration

We want to keep all of the fields as they are except for gold, silver and bronze. So we select all but those three fields under Key Fields and we select gold, silver and bronze under the Data Fields. If we had wanted to drop a field entirely – say, Closing Ceremony Date – we could leave it unchecked in both lists.

Best practice is to always include a Browse after a tool that modifies the structure of a data stream. Transpose is one of these tools. Let's add a Browse to the end of the data stream and run it to see what we have.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Name	Value
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Gold	8
2	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Silver	0
3	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Bronze	0
4	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Gold	6
5	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Silver	0
6	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Bronze	2
7	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Gold	4
8	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Silver	2
9	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Bronze	0
10	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Gold	1
11	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Silver	2
12	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Bronze	3

Figure 2-70 – Let's Tidy Things Up, Browse configuration

If we compare the top three records from the new Browse to the one that came out of the Input in figure 2-66, we see that we have two fields called *Name* and *Value* and no longer have the fields *Gold, Silver,* and *Bronze*. We also notice from *Athlete* to *Sport*, all fields are identical to the first three records in the original dataset. This is because we replicated them for each column we created.

This is one of the reasons that tidy data is not particularly human readable but is highly computer readable. Since all of the information is displayed in each record and there is only a single column to work on, interactive front-end software can work very fast with the data.

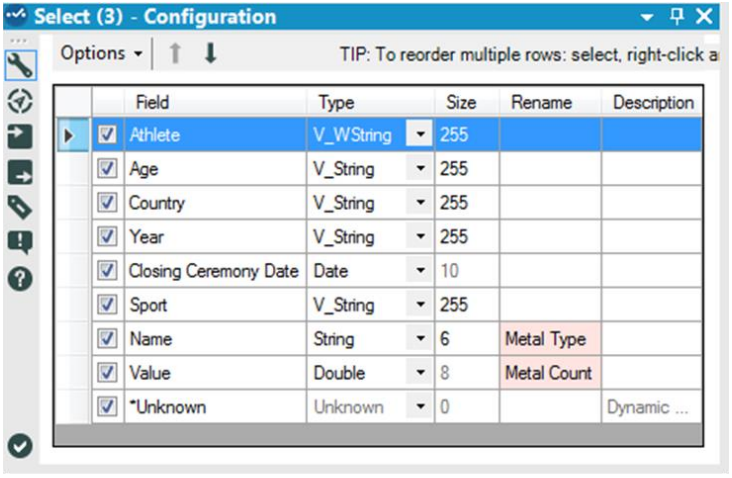


Figure 2-71 – Let's Tidy Things Up, Select configuration

Making this data truly tidy would mean we need to rename *Name* and *Value* to names that give better context to the field. Add a Select statement, and rename the Name and Value fields *Medal Type* and *Medal Count*, respectively.

We know we have rows that say zero medals were won by looking at the values in the last Browse tool we created. We are going to filter those data points out by adding a Filter tool after the Select.

Our goal is to filter out any records that have zero medals. We are filtering on a numeric field for the first time, which means we should use the Basic Filter to learn about the syntax. The configuration is as shown in the following figure.

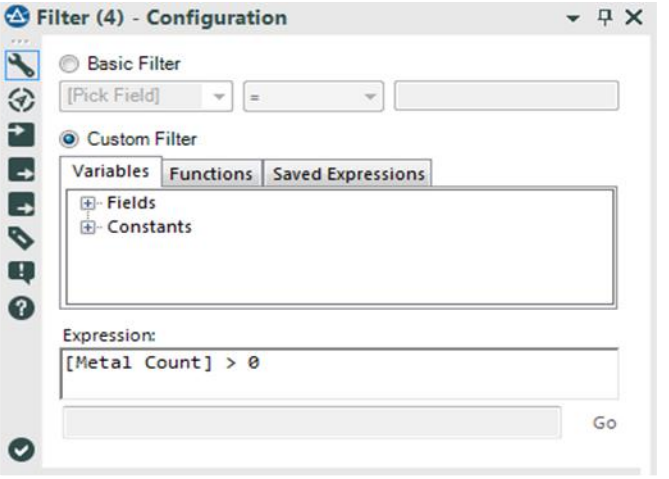


Figure 2-72 – Let's Tidy Things Up, Filter configuration

We can see how if we select Medal count, we have different options in the operator drop-down. This is because numeric fields allow different comparison methods than string fields.

We want to select greater than – ">" – and type "0" in the text box. When we look at the Expression below, we see that it says **[Medal Count] > 0**. This is because we do not put numeric values in quotes. Alteryx recognized that when we selected a numeric field in the basic filter drop-down, the "0" we typed in meant the number 0 and not the string 0, so it put the numeric value into the formula.

The last step involves two tools: the *Browse* tool and the *Output Data* tool.

Best practice dictates that we put a Browse before every data output so that we do not need to open the file to make sure we created it correctly.

We now add a Browse tool to the end of the data stream and also add an Output Data tool. We are going to write the file to the same folder we have saved the *Let's Tidy Things Up.yxmd*.

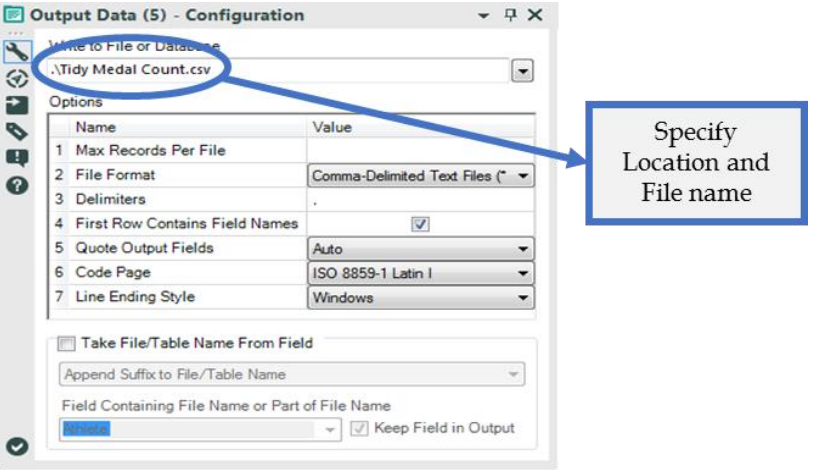


Figure 2-73 – Let's Tidy Things Up, Output Data configuration

To do this, we are going to type *.\Tidy Medal Data.csv* in the text box labeled *Write to File or Database*.

We just used a relative file path. Which allows us to reference files in relation to where we currently are. Some basics of relative paths are ".\", which means the current folder. "..\" means the parent folder (the folder that our current folder is in). ".\Folder Name\" will move our file into a folder below where we have the workflow.

We do not necessarily need to use relative paths, but if we are sharing Alteryx files, it is very beneficial to do so. We can use absolute paths (full file locations) by pasting them into this box or navigating to them in the File Browse option.

If we run the module, we can see what the transformed dataset looks like. This ensures that the information written into the .csv was correct.

Results - Browse (9)

8 of 8 Fields | Cell Viewer | 9,107 records displayed, 287 KB

Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Metal Type	Metal Count
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Gold	8
2	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Gold	6
3	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Bronze	2
4	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Gold	4
5	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Silver	2
6	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Gold	1
7	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Silver	2
8	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Bronze	3
9	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	Gold	2
10	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	Silver	1
11	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	Bronze	3
12	Alicia Coutts	24	Australia	2012	2012-08-12	Swimming	Gold	1

Figure 2-74 – Let's Tidy Things Up, Final Browse configuration

Here is how the *Let's Tidy Things Up* data stream looks on completion.

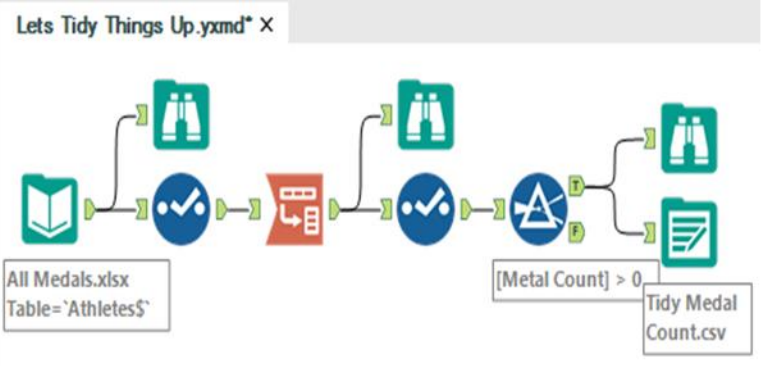
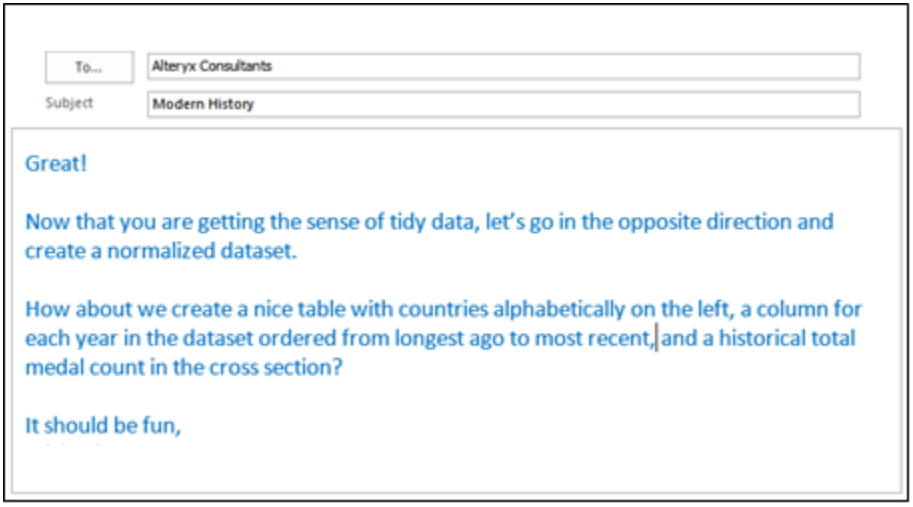


Figure 2-75 – Let's Tidy Things Up, Data stream after completion

2.21 Modern History



Notice that there is considerably less context built into this email. We often get very sparse information from people, and they will assume we know the context. In this case, it was assumed we were talking about the medal data that we have been working with during the training so far.

This is a much more complicated process than the last exercise, but that is only because the data was set up very well for what we were doing last time, and it isn't here.

We are going to be connecting to the same data source that we have been using, but we are going to use a shortcut in the connection process. Open a new workflow and save it as *Modern History.yxmd*, but make sure that *Let's Tidy Things Up.yxmd* is still open.

Click on the data input in *Let's Tidy Things Up* and copy it. Move over to the Modern History canvas, and paste what we have copied. We see that the input has been copied over and we do not need to recreate the connection.

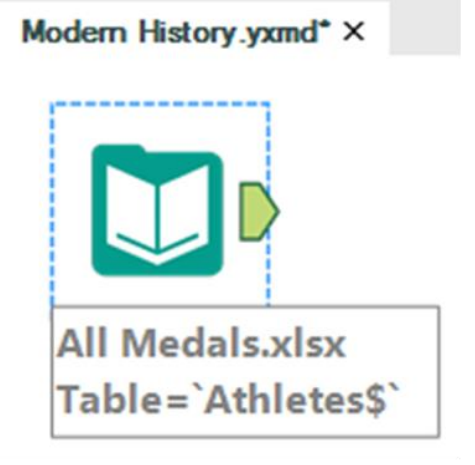


Figure 2-76 Modern history, Input Data

Best practice will once again bring in Browse and Select tools. But since we know from past experience what the data looks like and how it is read in, we will move directly into the next step.

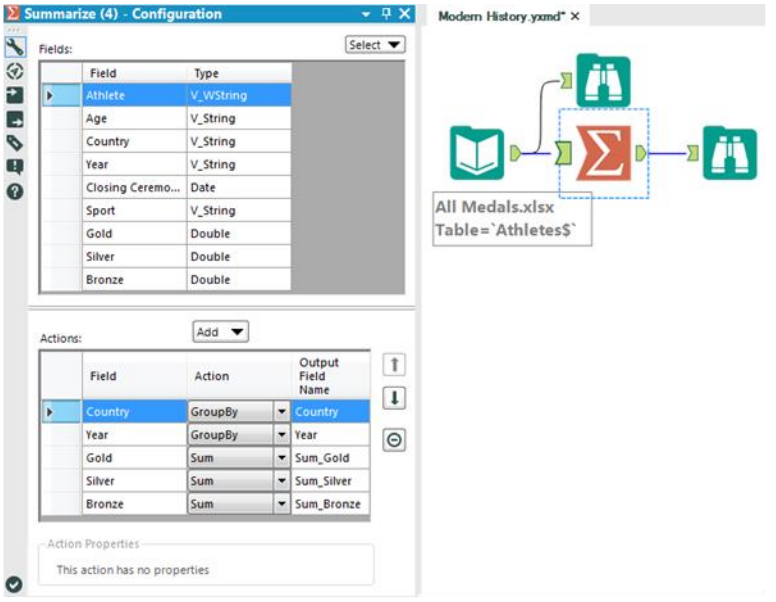


Figure 2-77 – Modern history, Summarize configuration

We know that this data is too granular for our desired result. So we can summarize it. Based on the email, we know the only information we will need in the end is the country, year, and something to do with the medals. So when using the Summary tool, we can group by the country and year fields and take the sum of each of the medal counts to take our first step down this path.

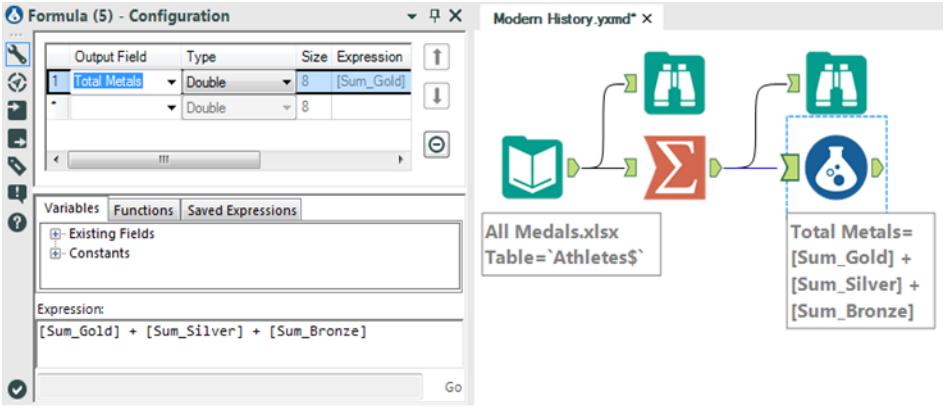


Figure 2-78– Modern history, Formula configuration

We now add a formula that creates a *Total Medal* count by adding the gold, silver and bronze fields for each record. (Remember that we used a Summarize tool so we should have a Browse tool.)

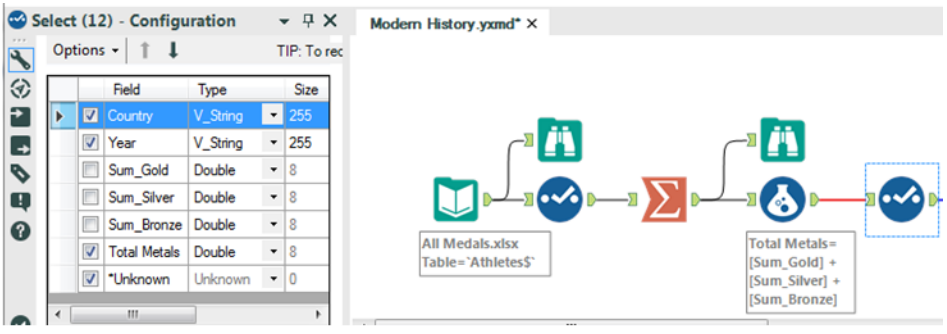


Figure 2-79 – Modern History Select Configuration

We can now add a select statement that will allow us to keep only the Country, Year, and Total Medals fields, which we will use to create the table.

We know we need a historical medal count, which means we are going to need to take the running total along with the

country and year. But because Running Total is a tool where order matters, we need to sort the data.

We can sort the Country and Year in ascending order to help us in two places: Initially, this will help because we are creating the order for the Running Total, but it will also help us with the order of records and columns when we normalize the data set.

Figure 2-80 – Modern History Sort Configuration

Now that we have ordered the data, we can create the Running Total for each country across years. To do this, we *Group By* Country and *Create Running Total* on Total Medals. What this will do is create the running sum of Total Medals down the data set (as time increases) and have that count restart every time a new country shows up.

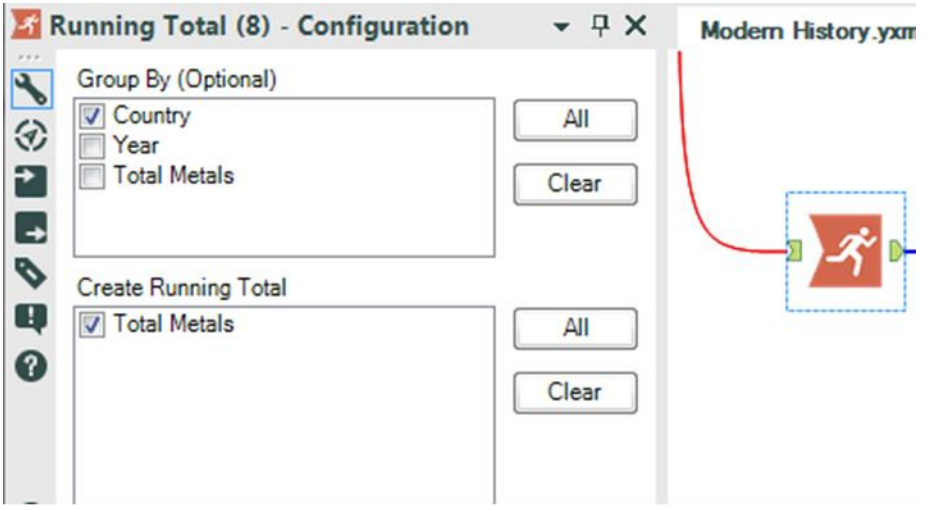


Figure 2-81 – Modern History Running Total Configuration

Let's take a look at what we have created so we can get a better sense of what the process so far has done.

Browse (9) - Configuration — Modern History

Results - Browse (9)

4 of 4 Fields | Cell Viewer | 401 records displayed, 7764 byt

Record #	Country	Year	Total Metals	RunTot_Total Metals
1	Afghanistan	2008	1	1
2	Afghanistan	2012	1	2
3	Algeria	2000	5	5
4	Algeria	2008	2	7
5	Algeria	2012	1	8
6	Argentina	2000	20	20
7	Argentina	2004	49	69
8	Argentina	2008	51	120
9	Argentina	2012	21	141
10	Armenia	2000	1	1
11	Armenia	2008	6	7

Figure 2-82 – Modern History Browse Configuration after Total

If we add a Browse tool and run the workflow, we can see that we have an alphabetical list of countries with a record for every year they won a medal. We can also see the year is increasing as we move down the list within a country. We then see the Total Medal count for that year and the running total for medals that the country has won going from one year to the next in a field called RunTot_Total Medals

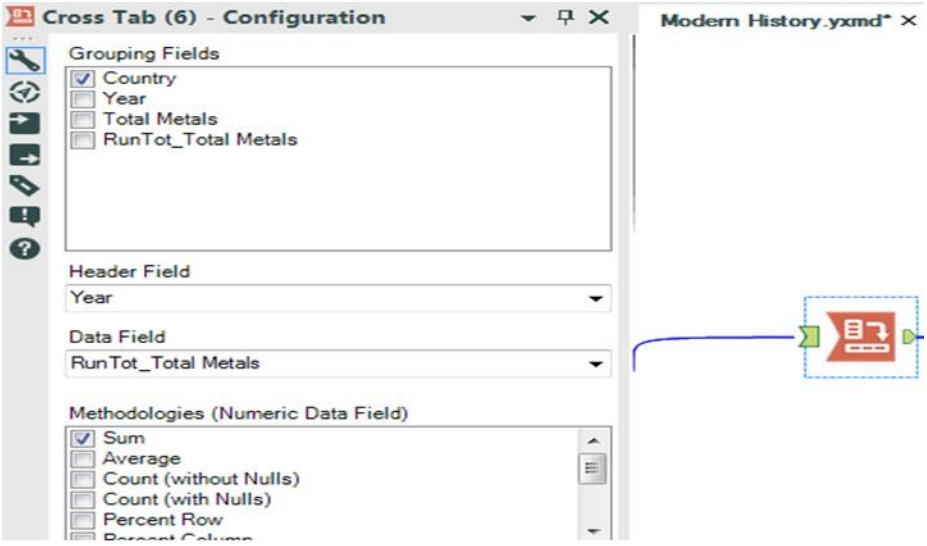


Figure 2-83 – Modern History Cross Tab Configuration

The next step in this process is to convert the data into a Cross Tab. If we add the Cross Tab tool to the end of the data stream and apply the settings in the above image, we will be close to our goal.

Let's add a browse tool and see what we have so far.

Browse (7) - Configuration | Modern History.yxmd*

Results - Browse (7)

8 of 8 Fields | Cell Viewer | 110 records displayed, 5181 bytes

Record #	Country	2000	2002	2004	2006
1	Afghanistan	[Null]	[Null]	[Null]	[Null]
2	Algeria	5	[Null]	[Null]	[Null]
3	Argentina	20	[Null]	69	[Null]
4	Armenia	1	[Null]	[Null]	[Null]
5	Australia	183	185	341	343
6	Austria	4	24	32	62
7	Azerbaijan	3	[Null]	8	[Null]
8	Bahamas	11	[Null]	13	[Null]
9	Bahrain	[Null]	[Null]	[Null]	[Null]
10	Barbados	1	[Null]	[Null]	[Null]

Figure 2-84 – Modern History Browse Configuration Cross Tab

The result in the image is close but not exactly what we wanted. We get the correct running totals in the years that each country won medals, however we get nulls in the years that they did not.

What we need to do now is create a series of formulas that replace the nulls with zero or the previous value as appropriate. Since we need to create formulas. we are going to need the formula tool; but this time, we are going to need to create seven similar calculations because we need to replace the values in seven different fields.

Let's think through these formulas. We want to change the cell only if it is null. If the column we are fixing is 2000, then it should be replaced with 0, and if the column is not 2000, it should be replaced with whatever is in the previous fixed column.

For those familiar with conditional statements, the syntax for an if-then statement is:

IF b1 THEN x ELSEIF b2 THEN y ELSE z ENDIF

For those unfamiliar with conditional statements, the concept is: Given a true or false (Boolean) expression, the calculation should do one of two things. The logic is if something is true, then do that; else, if the previous is false and something else is true, do the second option; else, do the default.

The other thing we need to know in writing these formulas is the test to see if something is null. The function used is:

IsNull(x)

Both of these syntax are under the functions tab in the Formula tool if we need to reference them.

The formulae that we need are:

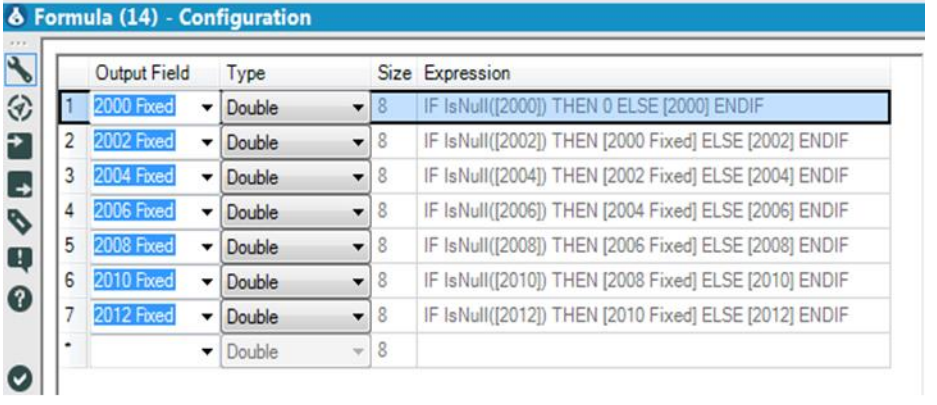
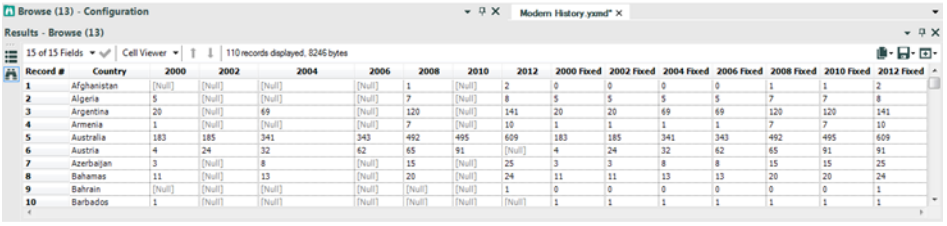


Figure 2-85 – Modern History Formula List

Add a Formula tool to the end of data stream, and add the seven formulas we see here with corresponding field names. We can also add a browse tool after that to see what we have created.

We can see that we have two sets of fields: those with the original sparse data, and those with the new dense data.



Record #	Country	2000	2002	2004	2006	2008	2010	2012	2000 Fixed	2002 Fixed	2004 Fixed	2006 Fixed	2008 Fixed	2010 Fixed	2012 Fixed
1	Afghanistan	[Null]	[Null]	[Null]	[Null]	1	[Null]	2	0	0	0	0	1	1	2
2	Algeria	5	[Null]	[Null]	7	[Null]	8	5	5	5	5	5	7	7	8
3	Argentina	20	[Null]	69	[Null]	120	[Null]	141	20	20	69	69	120	120	141
4	Armenia	1	[Null]	[Null]	[Null]	7	[Null]	10	1	1	1	1	7	7	10
5	Australia	183	185	341	343	492	495	609	183	185	341	343	492	495	609
6	Austria	4	24	32	62	65	91	[Null]	4	24	32	62	65	91	91
7	Azerbaijan	3	[Null]	8	[Null]	15	[Null]	25	3	3	8	8	15	15	25
8	Bahamas	11	[Null]	13	[Null]	20	[Null]	24	11	11	13	13	20	20	24
9	Bahrain	[Null]	[Null]	[Null]	[Null]	[Null]	[Null]	1	0	0	0	0	0	0	1
10	Barbados	1	[Null]	[Null]	[Null]	[Null]	[Null]	[Null]	1	1	1	1	1	1	1

Figure 2-86 – Modern History Formula List

The next thing that we need to do is remove and rename the columns that we have, so add a Select tool to the end of the data stream.

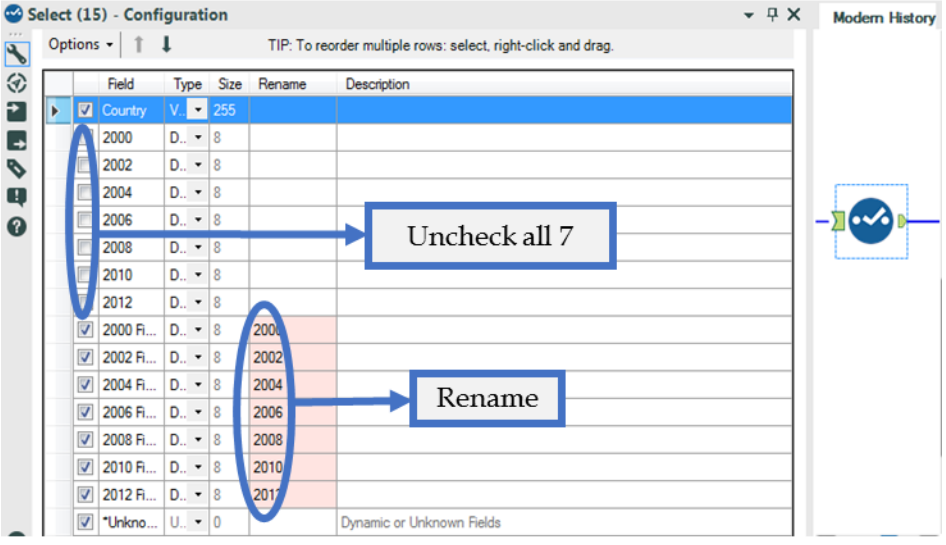


Figure 2-87 – Modern Select Configuration after Formula

Now the data has finished being prepped. We need to write it out, which we know because we were asked for a data set and not a specific answer. We should add a Browse tool and an

Output Data tool to end the data flow. Save the output as Historical Medal Count.csv.

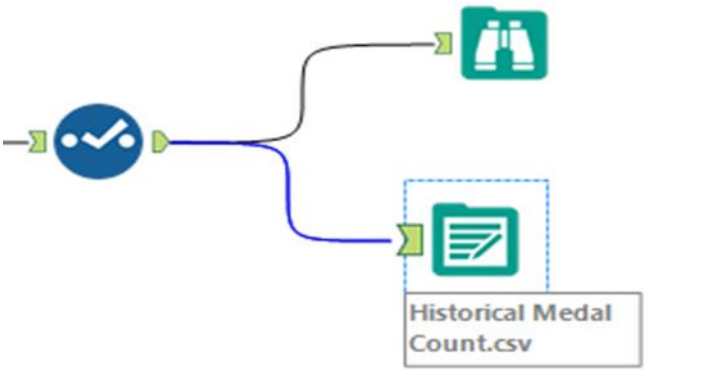


Figure 2-88 – Modern History Output

After doing these steps, the final workflow is as shown in the below figure.

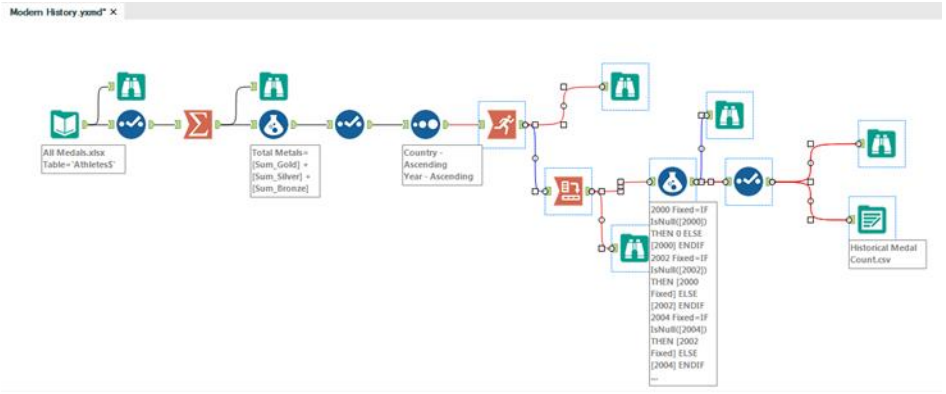
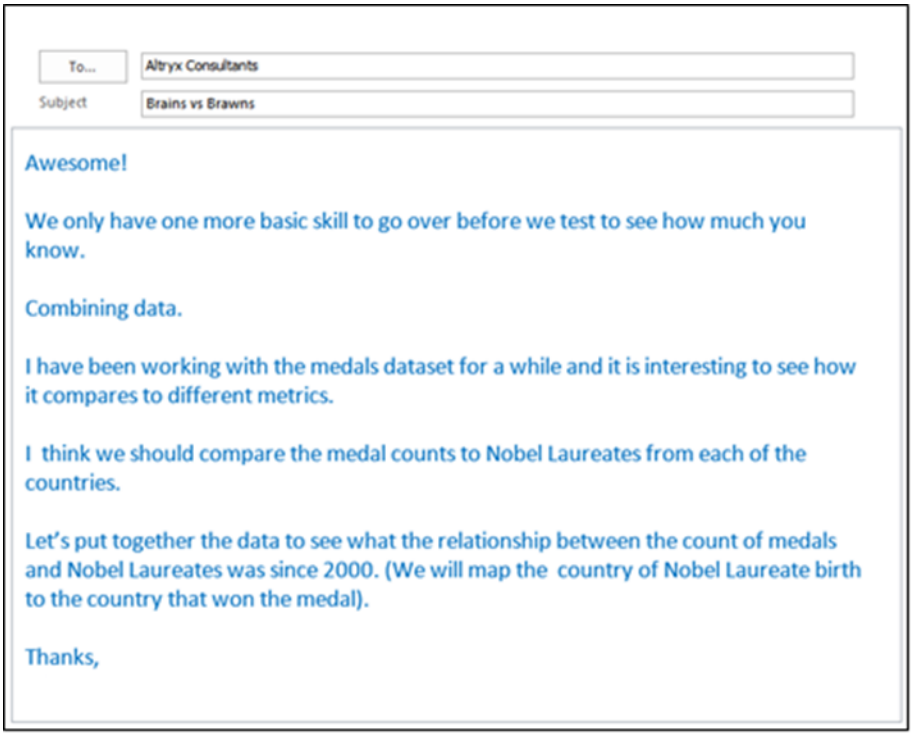


Figure 2-89 – Modern History Data Stream When Complete

2.22 Brains vs Brawns


To... Altryx Consultants

Subject Brains vs Brawns

Awesome!

We only have one more basic skill to go over before we test to see how much you know.

Combining data.

I have been working with the medals dataset for a while and it is interesting to see how it compares to different metrics.

I think we should compare the medal counts to Nobel Laureates from each of the countries.

Let's put together the data to see what the relationship between the count of medals and Nobel Laureates was since 2000. (We will map the country of Nobel Laureate birth to the country that won the medal).

Thanks,

Since we are combining data, let's revisit the analogy presented in the preface. When we look at a river, we see there are tributaries all along its length. Each of these tributaries may have gone through different terrain and could have started as very different sources. When they come together, they add whatever they carried along with them into the river they form.

To relate it to the task at hand, tributaries are branches of our data stream that come together, and when they come together, we have a richer data stream because we have the information that comes from everything contributing to it.

We are going to start by prepping the medals data and preparing them to be joined. We'll create a table with two columns called Country and Medal Count.

In order to do this, we are going to follow the following steps:

1. Import data
2. Transpose and rename the columns so that the data is tidy.
3. Filter out the 0 medal records
4. Summarize the data so that we only have one record per country and the total medal count.
5. Rename the medal count column *Total Medal Count*.

Since we have covered the tools and the concepts used in this exercise in previous exercises, overall flow should look familiar. Please rebuild the following workflow with the following configurations.

The properties windows for each of these tools as well as the data stream that is produced are shown in the following figures.

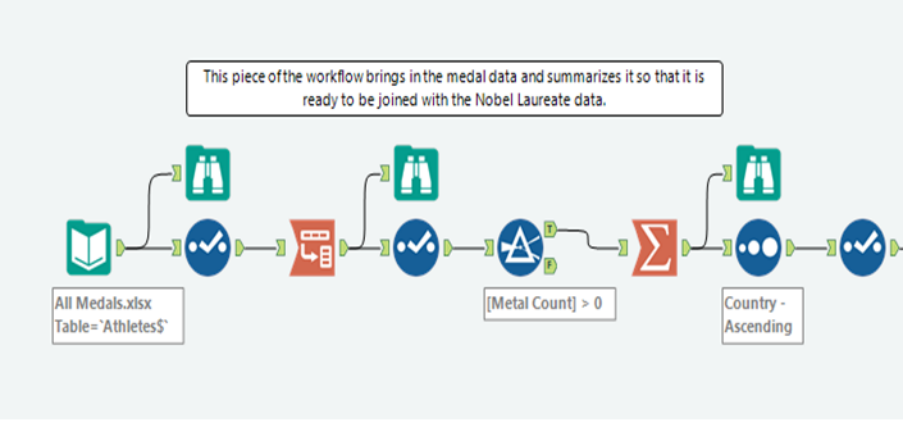
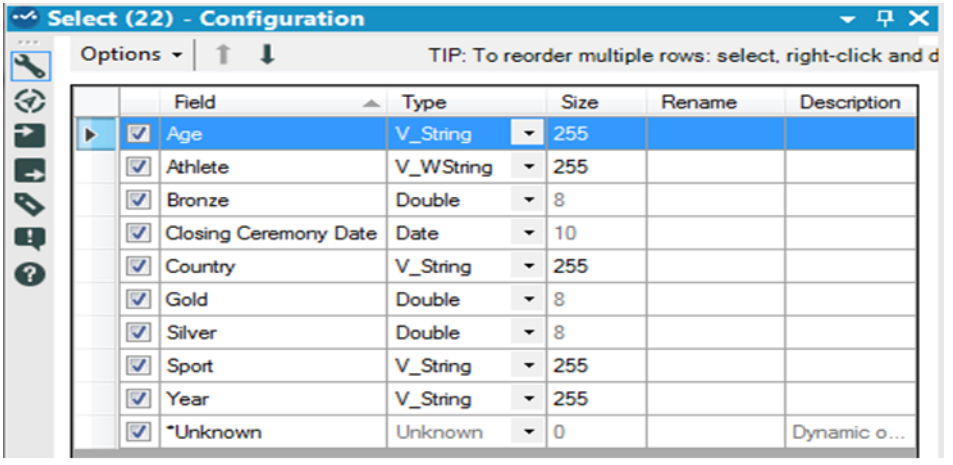


Figure 2-90 – Brains vs Brawns data stream



Field	Type	Size	Rename	Description
Age	V_String	255		
Athlete	V_WString	255		
Bronze	Double	8		
Closing Ceremony Date	Date	10		
Country	V_String	255		
Gold	Double	8		
Silver	Double	8		
Sport	V_String	255		
Year	V_String	255		
*Unknown	Unknown	0		Dynamic o...

Figure 2-91 – Initial Steps - Select

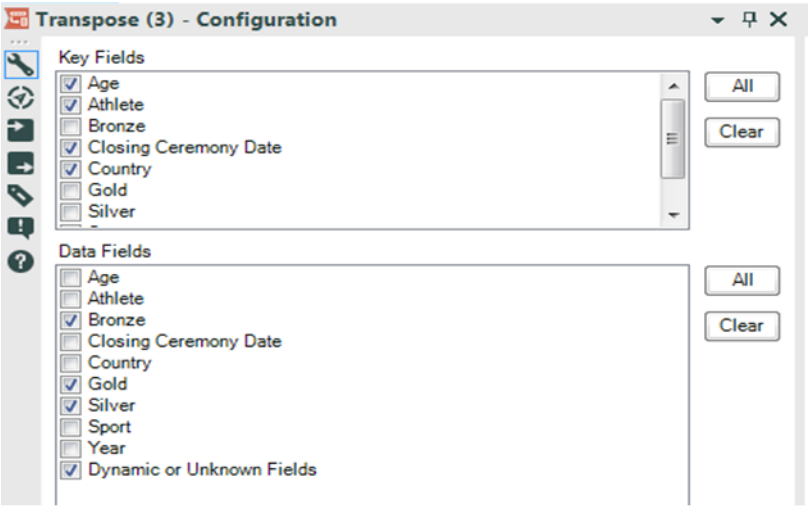


Figure 2-92 – Initial Steps - Transpose

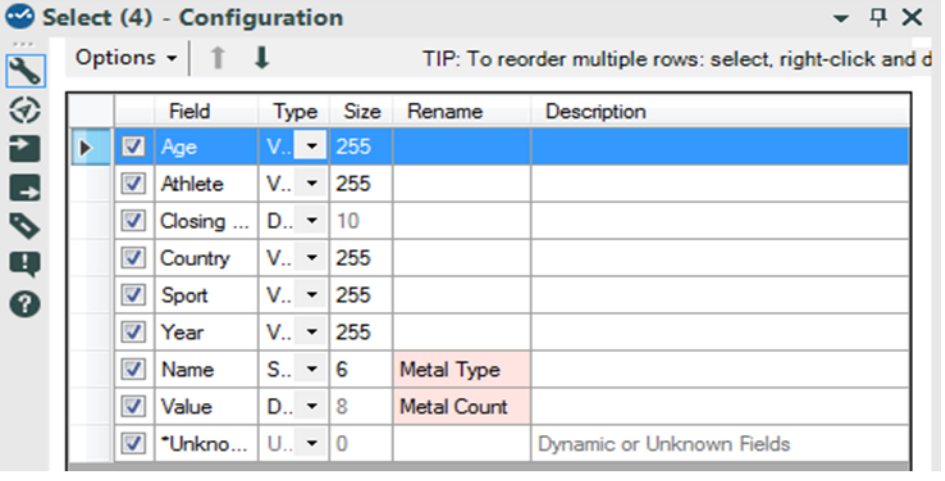


Figure 2-93 – Initial Steps - Select

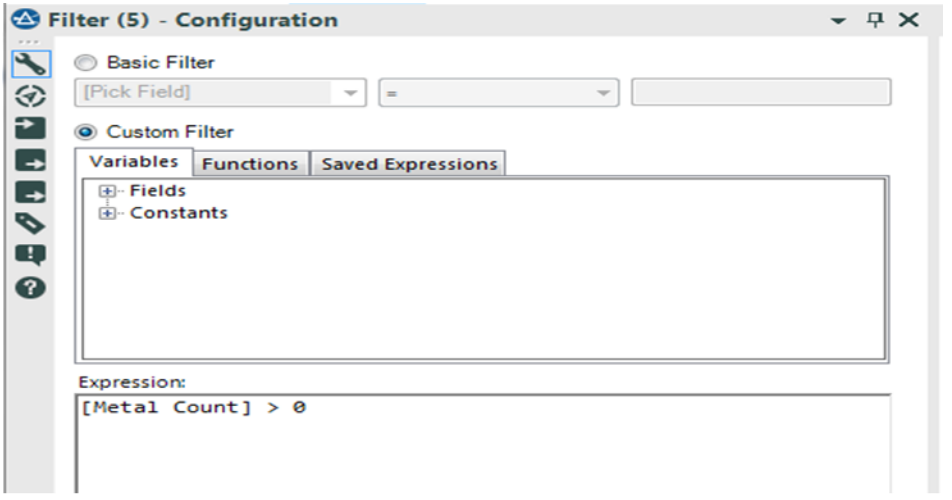


Figure 2-94 – Initial Steps - Filter

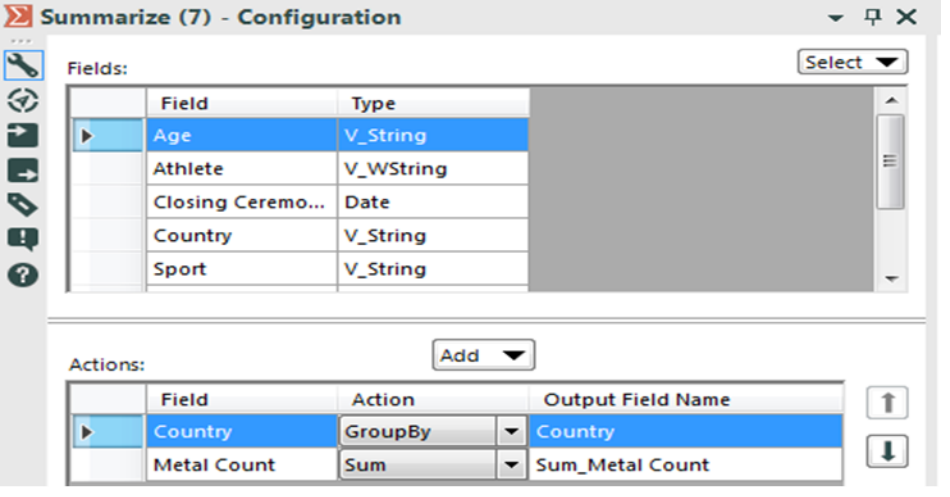


Figure 2-95 – Initial Steps - Summarize

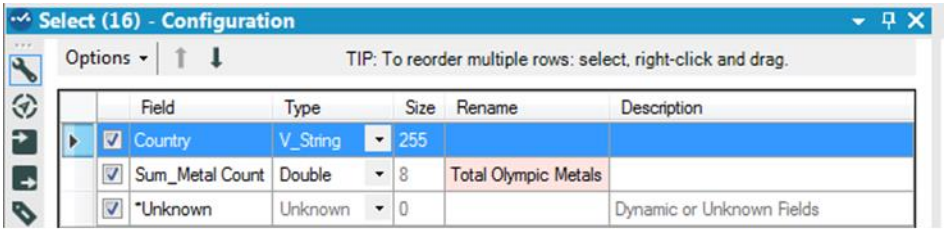


Figure 2-96 – Initial Steps - Sort

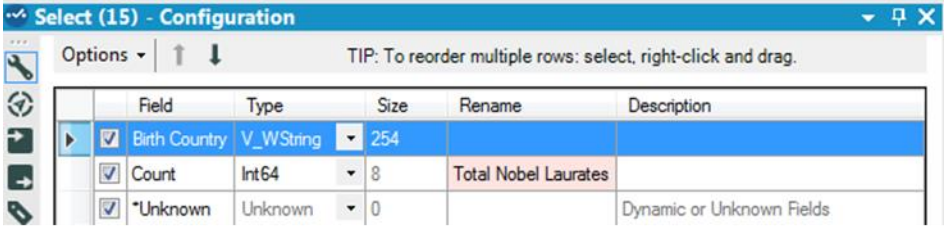


Figure 2-97 – Initial Steps - Select

Now that we have the data in the above stream prepared to be combined, we should prepare the other contributing data stream.

Let us open the file called *Nobel Laureates.csv* in the folder Chapter 2 -- The Games > Nobel Laureates. (Remember that we should always bring in a Browse and Select Tool with an input.)

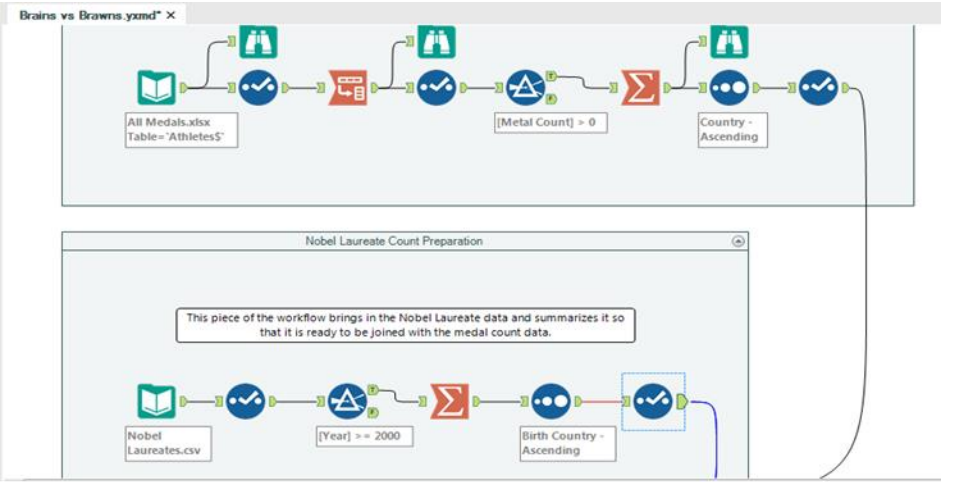
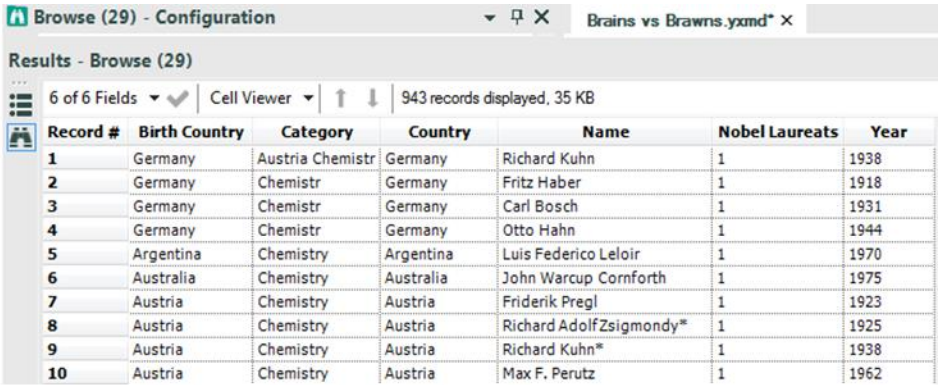


Figure 2-98 – Running unrelated analysis simultaneously

Notice that we now have two completely separate workflows. This ability is often a useful feature because we can run unrelated analyses at the same time, which aids in testing and in conditional application development.

If we run the workflow, we can look at the structure of the Nobel Laureates dataset. Here, we want to make sure that the field we plan on joining (Birth Country) is in the same type as County in the medal data stream.



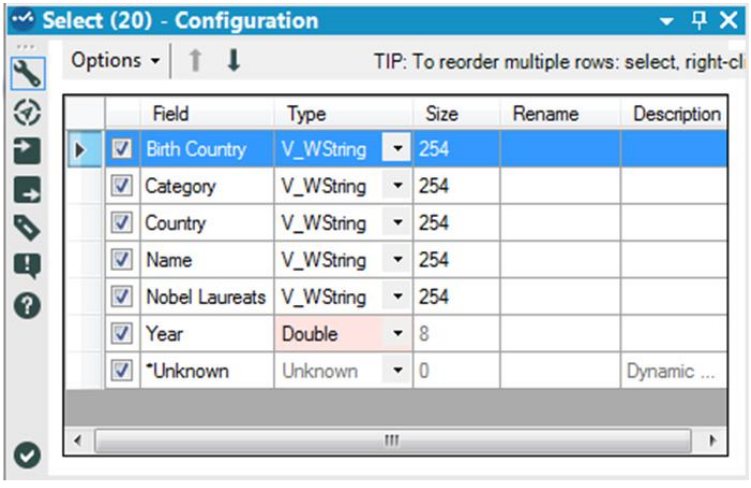
Browse (29) - Configuration | Brains vs Brawns.yxmd*

Results - Browse (29)

6 of 6 Fields | Cell Viewer | 943 records displayed, 35 KB

Record #	Birth Country	Category	Country	Name	Nobel Laureats	Year
1	Germany	Austria Chemistr	Germany	Richard Kuhn	1	1938
2	Germany	Chemistr	Germany	Fritz Haber	1	1918
3	Germany	Chemistr	Germany	Carl Bosch	1	1931
4	Germany	Chemistr	Germany	Otto Hahn	1	1944
5	Argentina	Chemistry	Argentina	Luis Federico Leloir	1	1970
6	Australia	Chemistry	Australia	John Warcup Cornforth	1	1975
7	Austria	Chemistry	Austria	Friderik Pregl	1	1923
8	Austria	Chemistry	Austria	Richard AdolfZsigmondy*	1	1925
9	Austria	Chemistry	Austria	Richard Kuhn*	1	1938
10	Austria	Chemistry	Austria	Max F. Perutz	1	1962

Figure 2-99 – Nobel Laureates – Browse Configuration



Select (20) - Configuration

Options | TIP: To reorder multiple rows: select, right-cl

	Field	Type	Size	Rename	Description
☑	Birth Country	V_WString	254		
☑	Category	V_WString	254		
☑	Country	V_WString	254		
☑	Name	V_WString	254		
☑	Nobel Laureats	V_WString	254		
☑	Year	Double	8		
☑	*Unknown	Unknown	0		Dynamic ...

Figure 2-100 – Nobel Laureates – Select Configuration

Now that we know what the data structure is and that it parallels the medals file, we can start our preparation for the join.

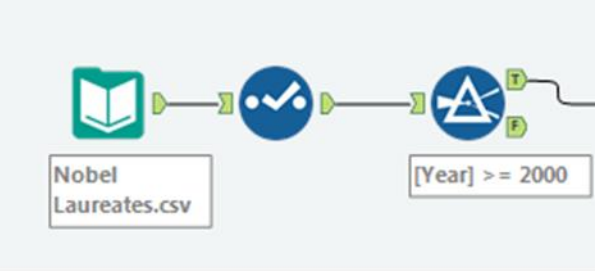
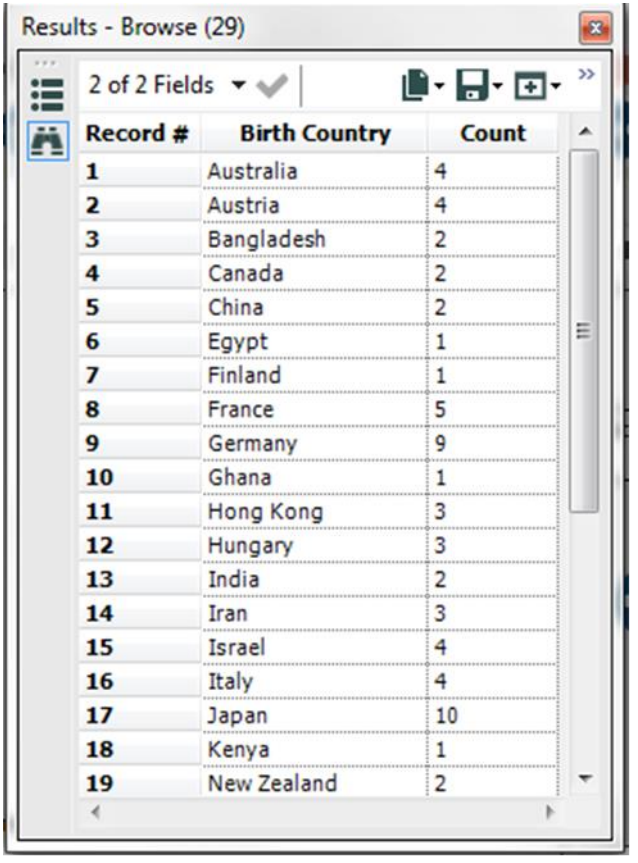


Figure 2-101 – Nobel Laureates – Preparing to join Medals

We know we want to limit this data to years starting in 2000. One way we can do this is to convert Year to a Double Type and set up a filter to be **[Year] >= 2000**.

Since we only need to know what the total number of Nobel Laureates there were for each country of birth, we can summarize the data.



Record #	Birth Country	Count
1	Australia	4
2	Austria	4
3	Bangladesh	2
4	Canada	2
5	China	2
6	Egypt	1
7	Finland	1
8	France	5
9	Germany	9
10	Ghana	1
11	Hong Kong	3
12	Hungary	3
13	India	2
14	Iran	3
15	Israel	4
16	Italy	4
17	Japan	10
18	Kenya	1
19	New Zealand	2

Figure 2-102 – Nobel Laureates – Browse after Summarize

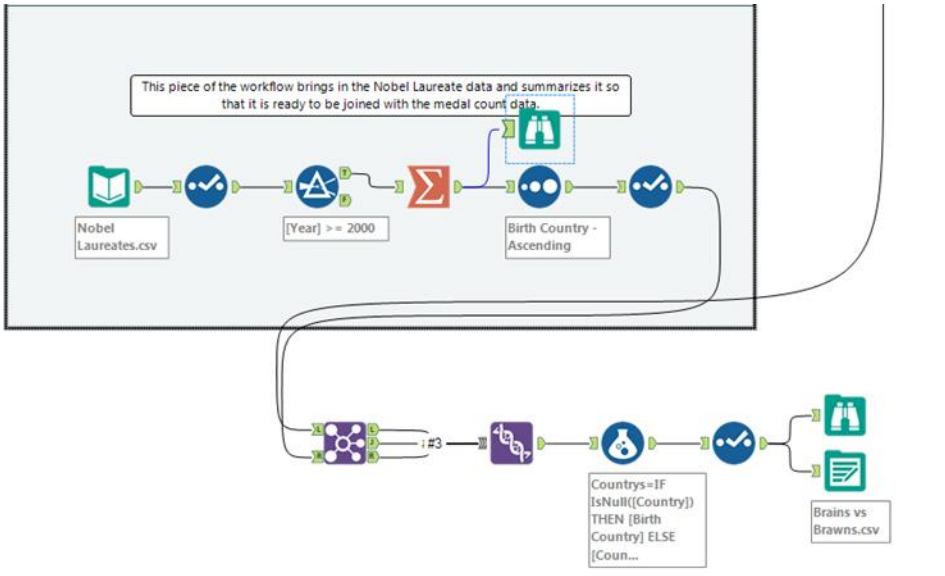


Figure 2-103 – Nobel Laureates – vis a vis All Medals

We see that we have a list of countries and a count of the number of Nobel Laureates. However, it is unclear what the number is because the field is called *Count*. We should rename it *Total Nobel Laureates*.

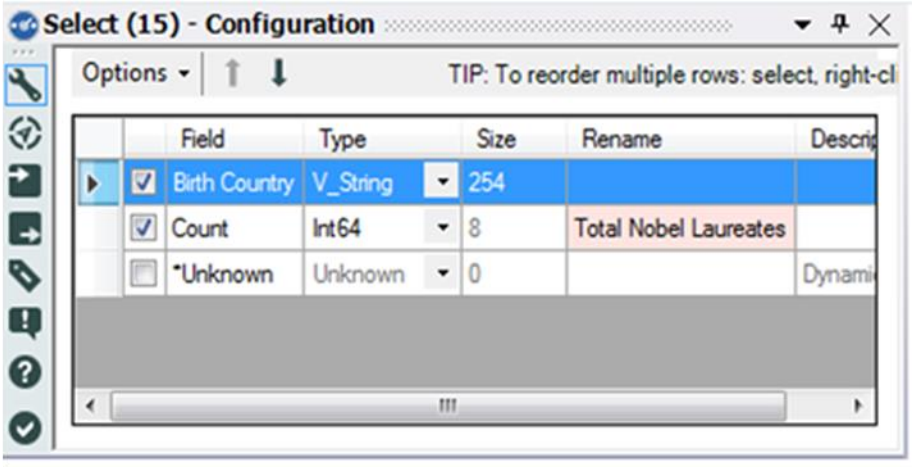


Figure 2-104 – Nobel Laureates – Select Configuration

We now have two data streams ready to be merged. We want to align the two datasets so that matching countries from each of the data steams share the same record, which means we want to join the data. Because we don't want to lose any data points if we have countries in one dataset but not the other, we will want to unite the three outputs from the join into a single data stream.

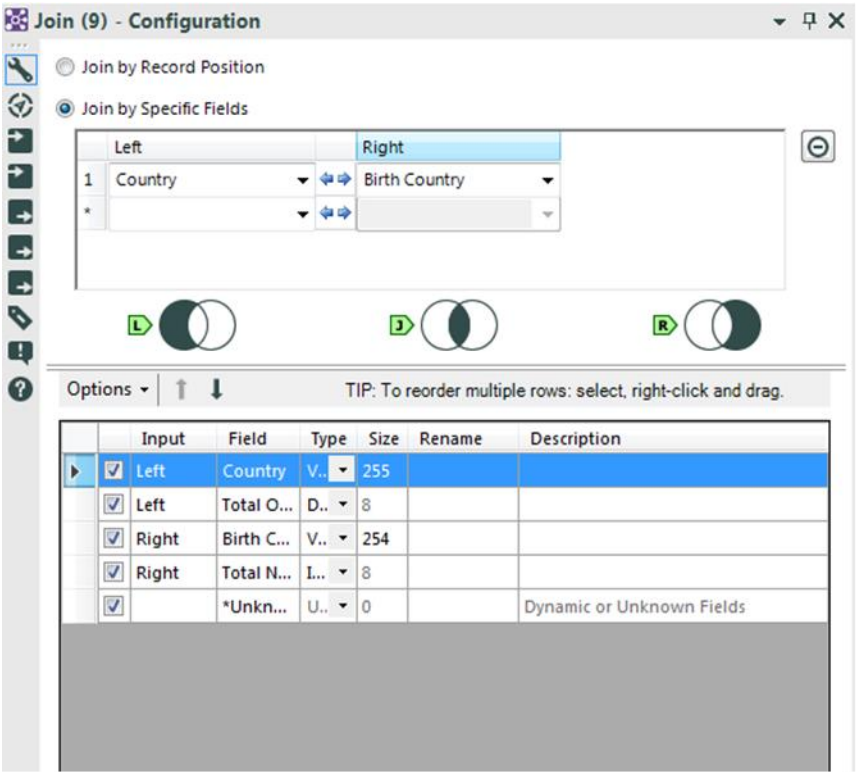


Figure 2-105 – Nobel Laureates – Join Configuration

We want to join on Country field from the Left (Input L) with Birth Country field from the Right (Input R).

It is important in this instance that we keep both joining fields because we intend to combine all three outputs in the next step. However, if this was not our intention, we could have removed the joining field from one of the two inputs.

Best practice is to give useful names to every connection that enters a multiple connection anchor.

Thus, we can see in the following image that we have relabeled the connections from *#1*, *#2*, and *#3* to *Left*, *Join*, and *Right*.

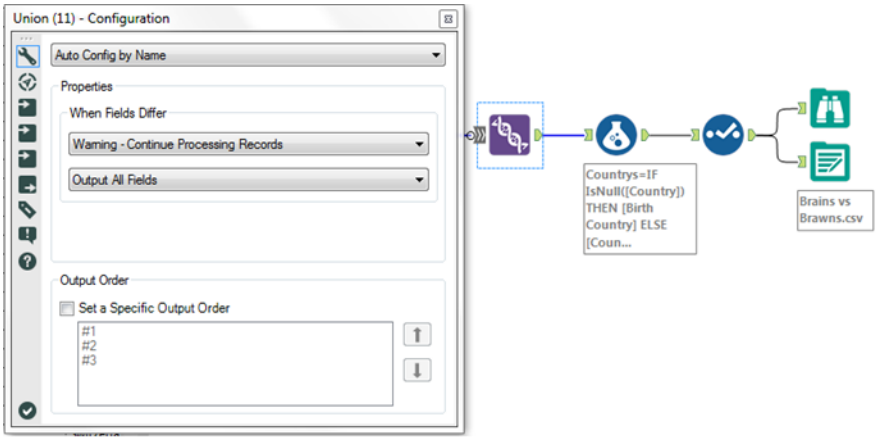


Figure 2-106 – Nobel Laureates – Union Configuration and Output Stream

Since we are doing a union of three output streams of a Join tool, we know that we will have matching column names. This allows us to use the *Auto Config by Name* setting for the Union tool and leave the rest of the defaults.

We need to add a browse tool again as we have just altered the structure of the data. This is to make sure the data looks the way we expect. Notice that we are doing this after the Union and not the Join. That is because when we are combining the three output streams of a Join tool using a Union, we are performing a single logical step called an outer join. Because this is a single step, we know that we should check both tools if an issue arises.

We are getting close to our goal; however; the data stream is also starting to get complex. So we should take a minute to annotate what we have so it will be easier to follow later. We are going to add *Tool Containers* and *Comments* to the two contributing data streams so we can easily identify different parts of this data stream. We can create the comments and containers like we see in the next image.

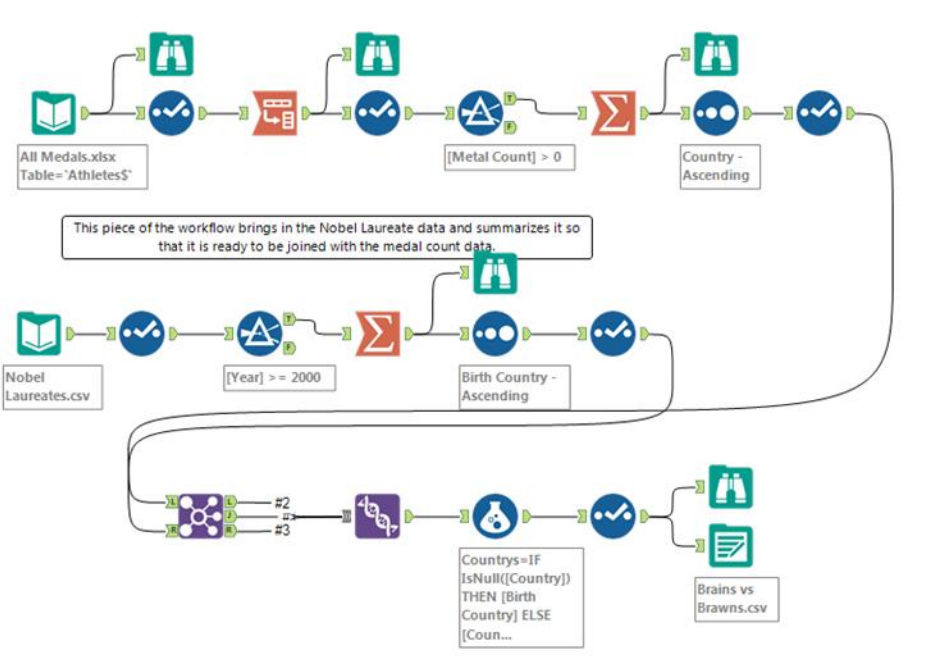


Figure 2-107 – Nobel Laureates – Comments and Containers

We can now drag the appropriate tools into the tool containers so the data stream is easier to understand.

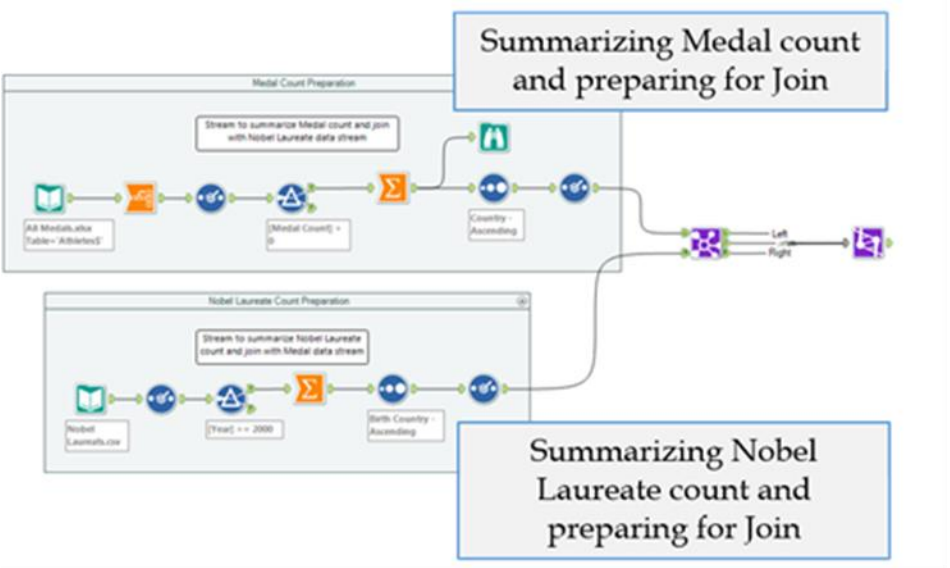


Figure 2-108 – Nobel Laureates –Output Stream with Comments

Looking at the data stream this way is helpful, but if we click on the arrows at the top-right corner, we can condense what we are looking at.

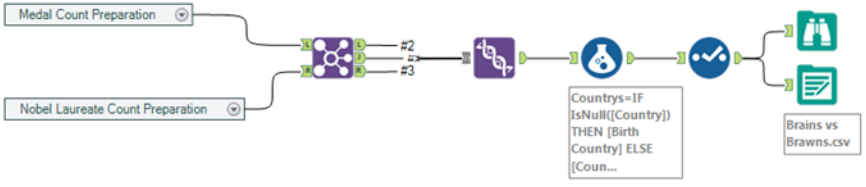
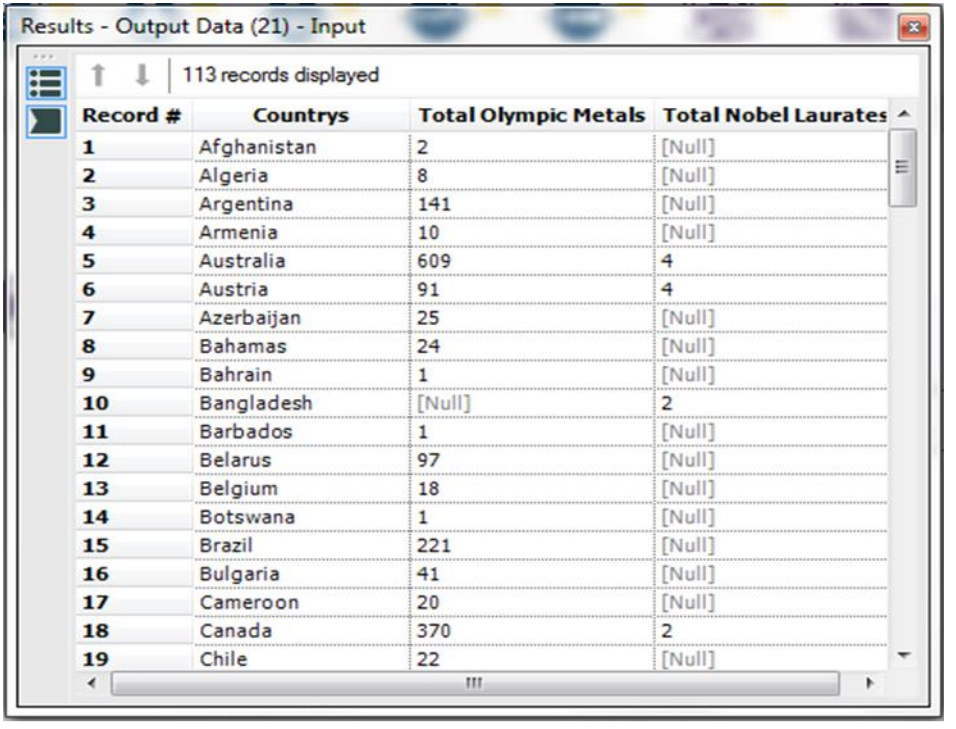


Figure 2-109 – Nobel Laureates –Simplified Output Stream

Now we can easily see the medal count preparation and the Nobel Laureate count preparation as two single processes instead of a series of tools. Now that we have made the data stream easier to understand, we should finish building the workflow.



Record #	Countrys	Total Olympic Metals	Total Nobel Laurates
1	Afghanistan	2	[Null]
2	Algeria	8	[Null]
3	Argentina	141	[Null]
4	Armenia	10	[Null]
5	Australia	609	4
6	Austria	91	4
7	Azerbaijan	25	[Null]
8	Bahamas	24	[Null]
9	Bahrain	1	[Null]
10	Bangladesh	[Null]	2
11	Barbados	1	[Null]
12	Belarus	97	[Null]
13	Belgium	18	[Null]
14	Botswana	1	[Null]
15	Brazil	221	[Null]
16	Bulgaria	41	[Null]
17	Cameroon	20	[Null]
18	Canada	370	2
19	Chile	22	[Null]

Figure 2-110 – Nobel Laureates –Browse Configuration

We can observe from the Browse that the country names matched the names in both the Country and Birth Country fields. Let's create a conditional formula that allows us to convert the two columns with nulls into a single column that always has a Country name.

Add a formula tool to the end of the data stream with a formula called **Countries** with the formula: **IF IsNull ([Country]) THEN [Birth Country] ELSE [Country] ENDIF**. This will take the Country value unless it is null and the Birth Country if it is.

Now we only need to clean up the data and export it to a .csv file. Add a Select tool with the following configuration, and export the file to *Brains vs Brawns.csv*.

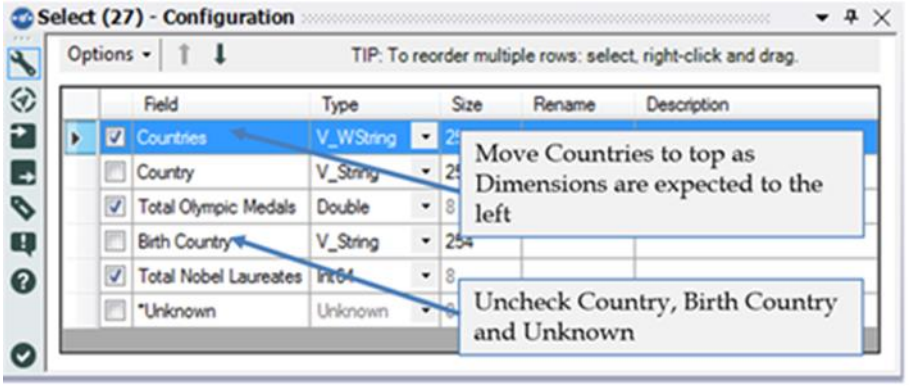


Figure 2-111 – Nobel Laureates –Select Configuration

The final workflow is as shown in the following figure.

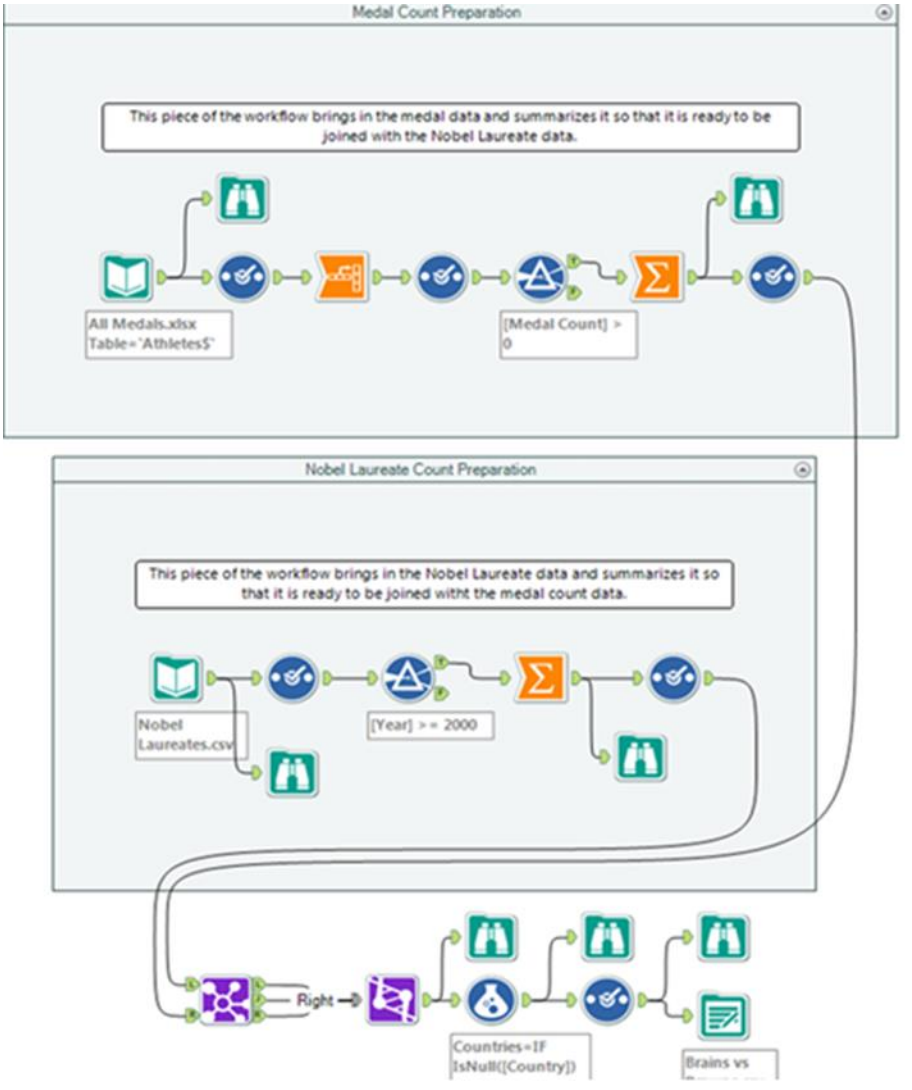
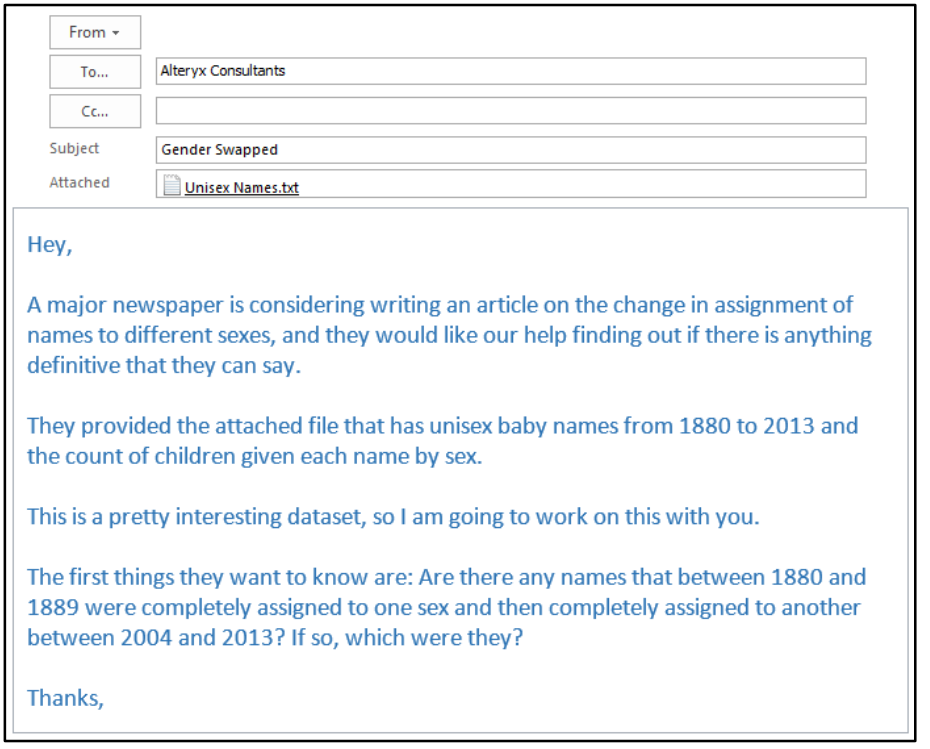


Figure 2-112 – Nobel Laureates –Complete Workflow

CHAPTER 3
Unisex Baby Names


From
To...
Alteryx Consultants
Cc...
Subject
Gender Swapped
Attached
Unisex Names.txt
Hey,
A major newspaper is considering writing an article on the change in assignment of names to different sexes, and they would like our help finding out if there is anything definitive that they can say.
They provided the attached file that has unisex baby names from 1880 to 2013 and the count of children given each name by sex.
This is a pretty interesting dataset, so I am going to work on this with you.
The first things they want to know are: Are there any names that between 1880 and 1889 were completely assigned to one sex and then completely assigned to another between 2004 and 2013? If so, which were they?
Thanks,

3.1 Tools & Concepts

Tools	**Concepts**
Imputation	Handling Nulls
Multi-Field Formula	Multiple Record Calculations
Multi-Row Formula	Repetitive Calculations
Text Input	Ad Hoc Data
	Problem Analysis

3.2 Imputation

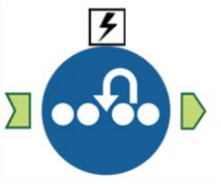


Figure 3-1 – Imputation

The *Imputation* tool replaces a specific value in numeric fields. It is most commonly used to handle Null values in formulas.

Group	Input	Output
Preparation	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with a numeric field.

Output: A data stream with at least one imputed field.

The tool will only modify an individual value and cannot be used to impute a range of values, such as all negatives.

Properties Window:

The *Imputation Configuration* window has five components.

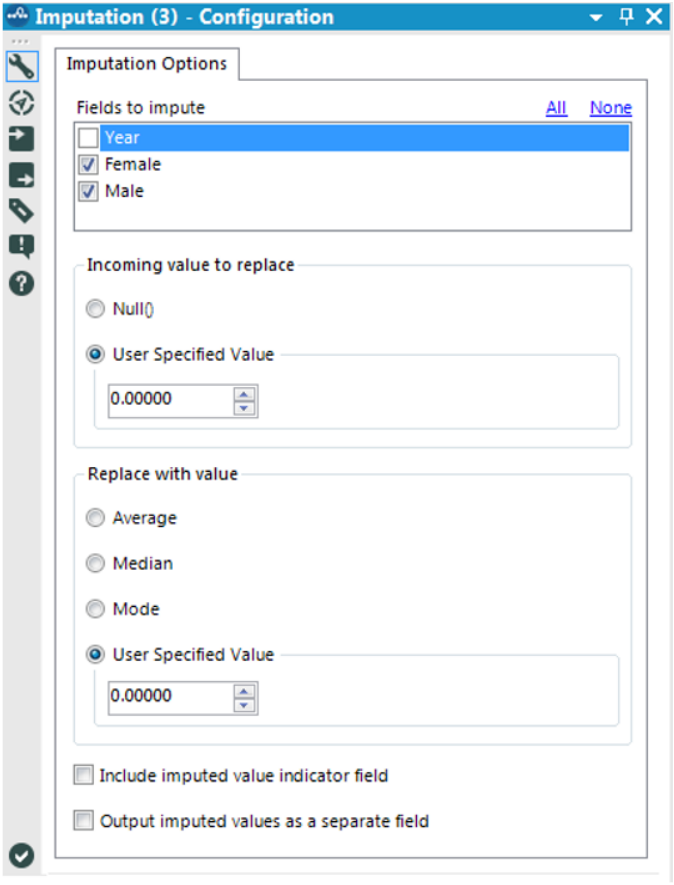


Figure 3-2 – Imputation configuration

- *Fields to Impute* is a list of numerical fields from the incoming data stream that one can choose to impute.

- *Incoming value to replace* allows us to choose to replace nulls or a specific numeric value that exists in the incoming dataset.
- *Replace with value* allows us to replace the value selected in *Incoming value to replace* with either the average (mean), median (middle), mode (most common), or a custom value that we select.
- *Include imputed value indicator field* creates a field for each of the *Fields to impute* with ones and zeros, where 1 indicates that the data was imputed and 0 that it was not.
- *Output imputed values as a separate field* creates a new field for each of the Fields to impute instead of replacing the values in the original field.

3.3 Multi-Field Formula

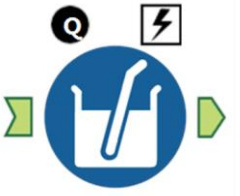


Figure 3-3 – Multi-field formula

The *Multi-Field Formula* tool provides the ability to create a formula that will be reused across multiple fields.

Group	Input	Output
Preparation	See below	See below

The formulas created here are arbitrarily complex, and thus can significantly slow down the data stream. Care must be taken to make sure the output field created has a field type compatible with the result created. // is the comment character.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with at least two fields of the same type.

Output: The original data stream with fields modified or original data stream with a new field for every selected field from the input.

Application questions can be connected to the *Black Question Anchor* for "Use Answers" in this tool.

Properties Window:

The *Multi-Field Formula Configuration* window has five main components.

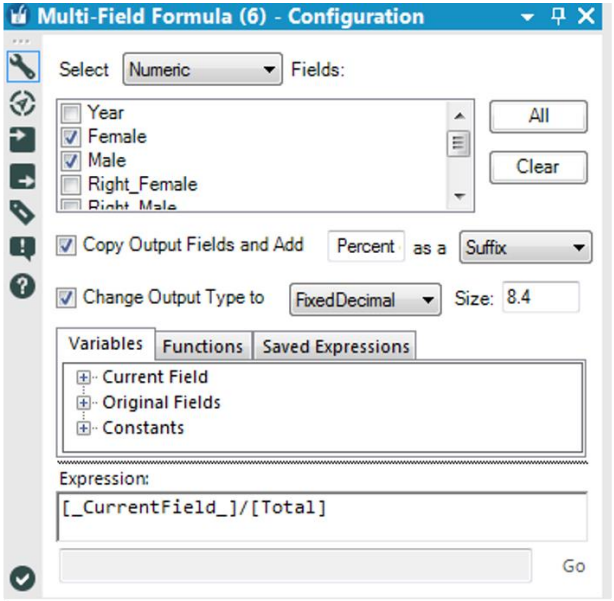


Figure 3-4 – Multi-field formula configuration

- *Select … Fields*: populates the list with the fields that match what we have selected in the drop-down. Calculation is performed on each of the fields selected from the generated list.
- *Copy Output and Add … as a …*: when selected, creates a new field as the output instead of replacing the ones that we are performing the calculation on. *Decimal*, in the example, will be appended to the field name as a Suffix or Prefix.
- *Change Output Type to … Size …*: when selected, we can change the metadata of the output that we create.

- The last two sections are similar to the formula window that we are familiar with, except that since we are working across multiple fields, we have the ability to use the *_CurrentField_* in our calculations.

3.4 Multi-Row Formula

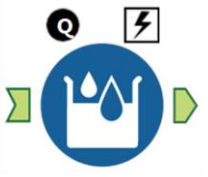


Figure 3-5 – Multi-row formula

The *Multi-Row Formula* tool provides the ability to create a formula that will reference other records.

Group	Input	Output
Preparation	Any data stream	See below

The values are updated by moving down the records in order to modify the current record based on the result of the formula in the previous record. The formulas we create here can be arbitrarily complex, and thus can significantly slow down the data stream. Care has to be taken to ensure the output field created has a file type that is compatible with the result being created. The // is the comment character.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the top *Black Question Anchor* for "Use those Answers" in this tool.

Output: The original data stream with a field modified or the original data stream with a new field.

Properties Window:

The *Multi-Row Formula Configuration* window has six core components.

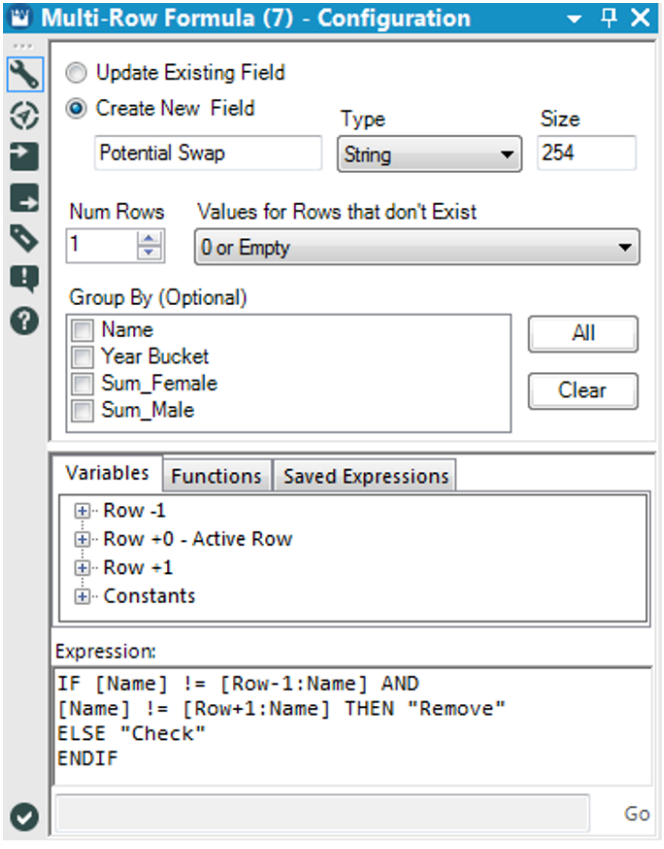


Figure 3-6 – Multi-row formula configuration

- *Update Existing Field* allows us to replace the entries in the field selected in the list below.
- *Create New Field* is the other option for the output, which allows us to create a new field instead of updating the current one. With this selection, we need to name the field and set the metadata of the field.
- *Num Rows* changes the number of rows before and after the current row we have access to in the formula.
- *Values for Rows that Don't Exist* allows us to define the behavior for records that do not actually exist. (In this case, for the first record, there is no *[Row-1: …]*. So the field is replaced by a 0 since the field is numeric.)
- *Group By* allows us to reset the calculation based on changes in values in the fields we select.
- The bottom two sections are similar to the formula boxes we are used to seeing except that we have the added ability to refer to field values in different rows.

3.5 Text Input

Figure 3-7 – Text input

The *Text Input* tool allows creation of datasets to be used by typing in the data.

Group	Input	Output
In/Out	None	Data stream

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Text Input Configuration* window allows us to create a data source that is internal to the module. The configuration has five options.

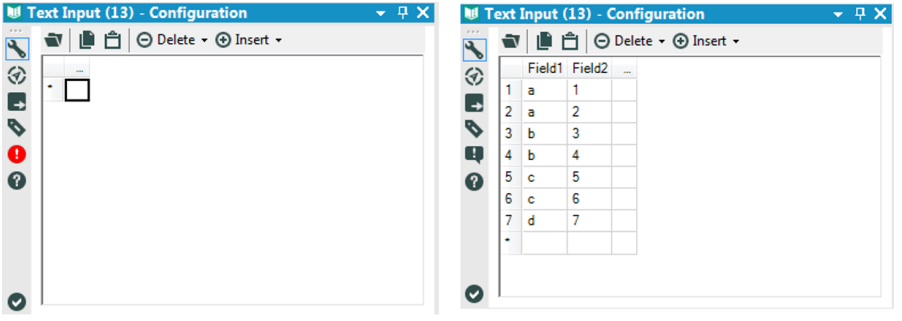


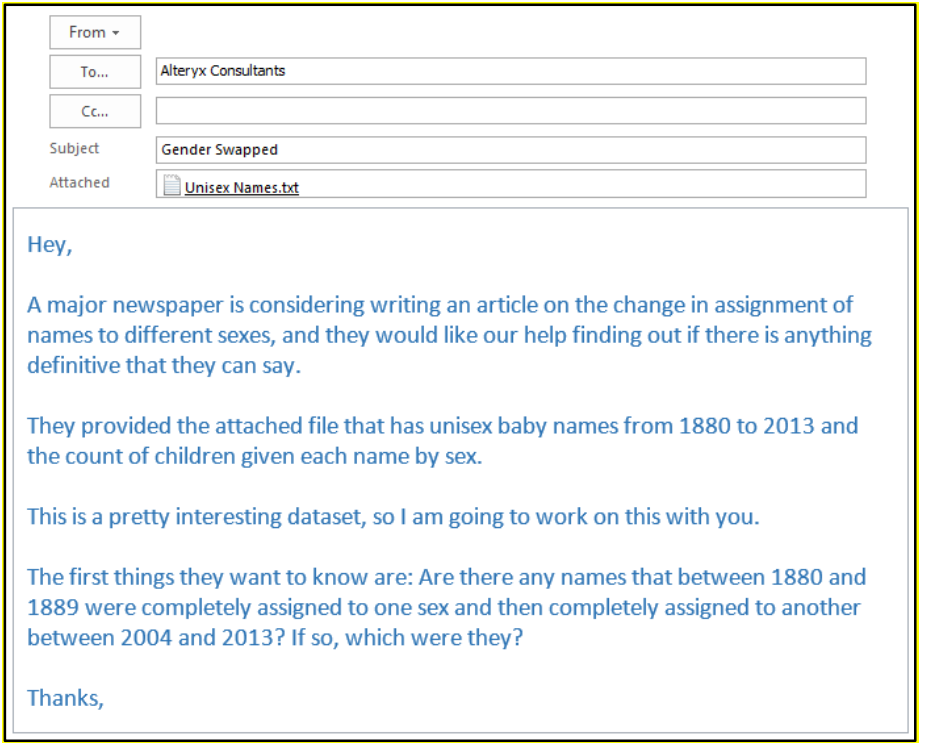
Figure 3-8 – Text input configuration

- *Import* allows us to import a dataset.

- *Copy* allows us to copy the cells we highlighted in the table.
- *Paste* allows us to paste values into the table.
- *Delete* allows us to delete everything, or the row or column we have selected.
- *Insert* allows us to create a row or column in the data set.

An example of what this would look like is provided in the previous image.

3.6 Gender Swapped



Let us refer to the *Unisex Names.txt* file in the folder *Chapter 3 – Unisex Names*. It is a tab-delineated file. We need to make sure the file pop-up window looks like the image represented here. Let us save the workflow as *Gender Swapped*.

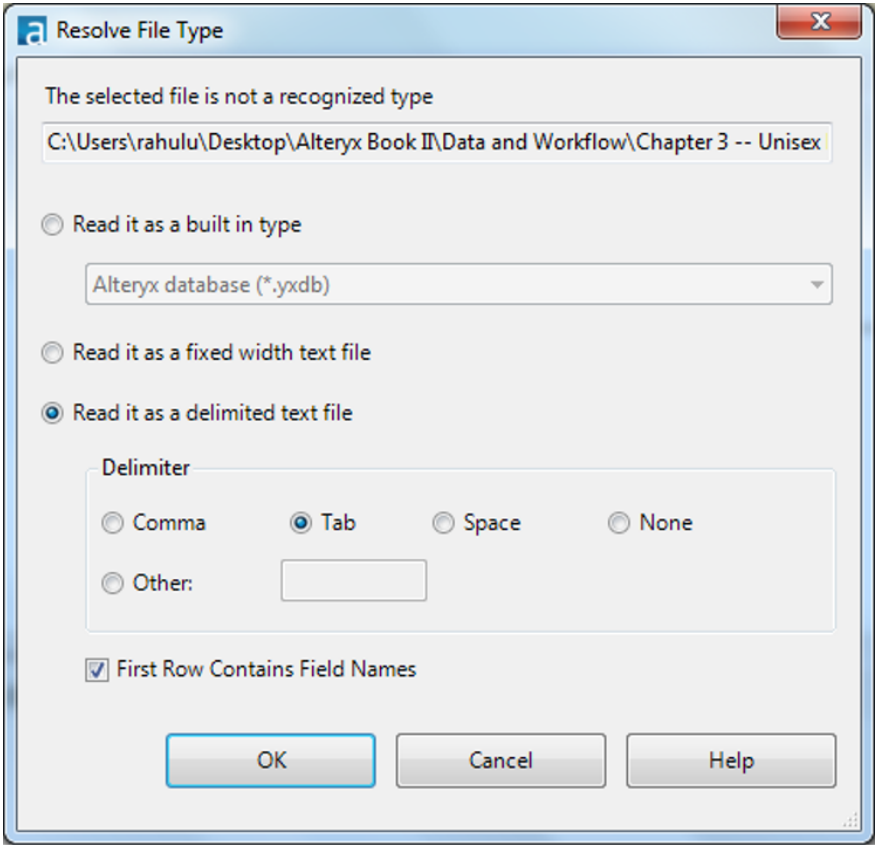
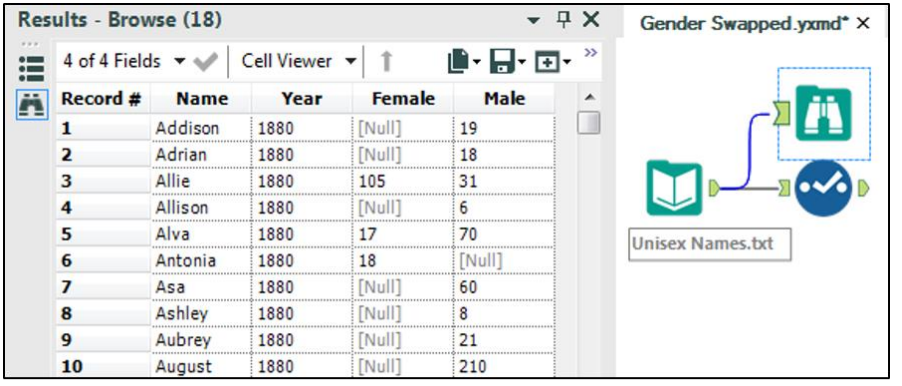


Figure 3-9 – Tab delineated pop-up window

Add a *Browse* tool and a *Select* tool to the input. After running the workflow, we should be able to see the records in the file.



Record #	Name	Year	Female	Male
1	Addison	1880	[Null]	19
2	Adrian	1880	[Null]	18
3	Allie	1880	105	31
4	Allison	1880	[Null]	6
5	Alva	1880	17	70
6	Antonia	1880	18	[Null]
7	Asa	1880	[Null]	60
8	Ashley	1880	[Null]	8
9	Aubrey	1880	[Null]	21
10	August	1880	[Null]	210

Figure 3-10 – Gender Swapped Browse

If we take a look at the data, we will see that some numeric fields have *nulls* in them. Whenever values are missing, Alteryx fills those cells with a "[Null]" placeholder. We could leave these values as nulls; however, it is easier to work with numeric fields when they do not have nulls in them. Instead, what we will do is impute the nulls to 0s.

Unfortunately, Alteryx did not recognize these fields as numeric, so we need to make modifications in the *Select* tool so that *Female* and *Male* are converted into *Double*.

Since we know we need to work with two 10-year periods, we should also convert *Year* to *Double* because it will make our formulas simpler and faster when we filter and flag the data.

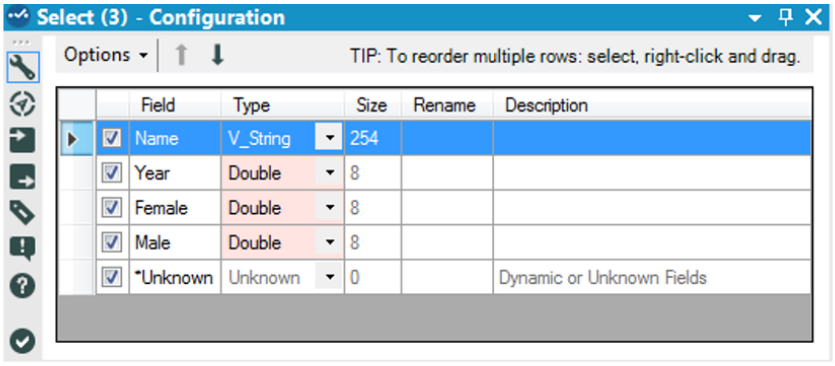


Figure 3-11 – Gender Swapped Select

If we add the *Imputation* tool with the settings as seen in the image, the tool will convert all of the null fields in *Female* and *Male* fields to zeros.

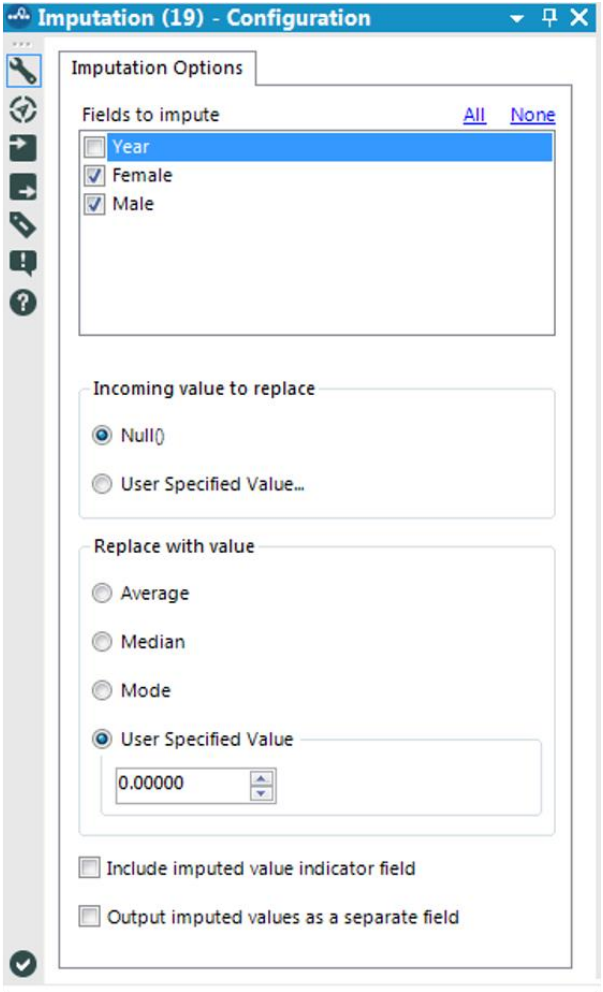
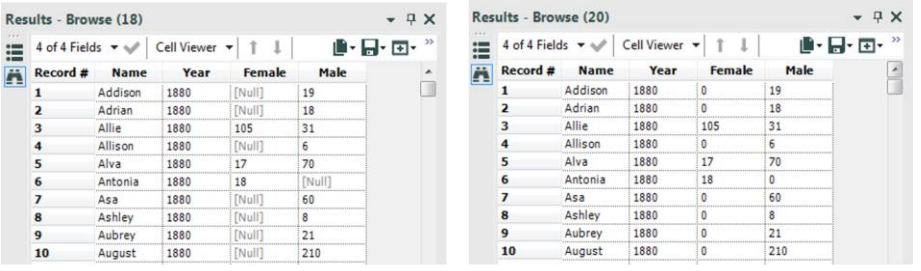


Figure 3-12 – Gender Swapped Imputation configuration

Figure 3-13 – Gender Swapped data stream

Now let's see how the data has changed.



Results - Browse (18)

4 of 4 Fields | Cell Viewer

Record #	Name	Year	Female	Male
1	Addison	1880	[Null]	19
2	Adrian	1880	[Null]	18
3	Allie	1880	105	31
4	Allison	1880	[Null]	6
5	Alva	1880	17	70
6	Antonia	1880	18	[Null]
7	Asa	1880	[Null]	60
8	Ashley	1880	[Null]	8
9	Aubrey	1880	[Null]	21
10	August	1880	[Null]	210

Results - Browse (20)

4 of 4 Fields | Cell Viewer

Record #	Name	Year	Female	Male
1	Addison	1880	0	19
2	Adrian	1880	0	18
3	Allie	1880	105	31
4	Allison	1880	0	6
5	Alva	1880	17	70
6	Antonia	1880	18	0
7	Asa	1880	0	60
8	Ashley	1880	0	8
9	Aubrey	1880	0	21
10	August	1880	0	210

Figure 3-14 – Gender Swapped Browse - Before and After

We can create a filter that limits the data set to *Years before 1890 or years after 2003* to isolate the two date ranges 1880 to 1889 and 2004 to 2013. We will add a filter to the end of the data stream. Use the expression box below to verify the Boolean formula.

Expression:

```
// [Year] < 1890 || [Year] > 2003
// or
[Year] < 1890 OR [Year] > 2003
```

Figure 3-15 – Gender Swapped expression box

Notice that in this expression, we have two lines starting with //. These two forward slashes mean that the line is a comment. It is a way to tell Alteryx to ignore that specific line from the calculation. We used it to show the two intended methods of filtering this data, but it can be used to maintain old versions of the formulas or leave descriptions of the formula for future benefit.

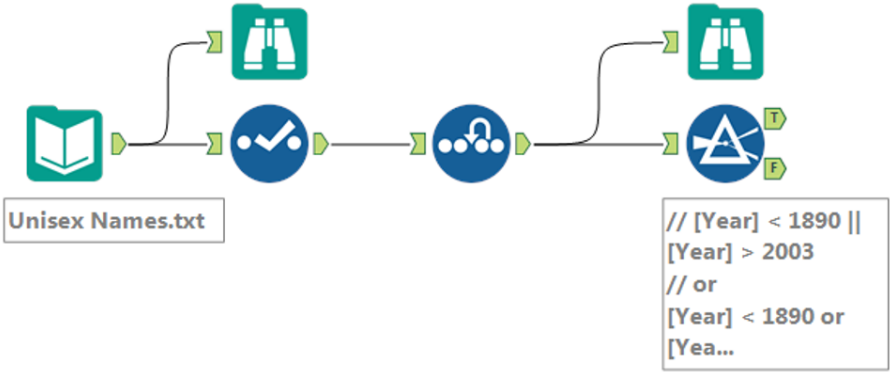


Figure 3-16 – Gender Swapped Data stream

The next thing we can do is create a *flag* for the first and last 10 years for the dataset. We want to do this to create a comparison between the two time frames. In order to do this, we will create a string field that will have *First 10 Years* or *Last 10 Years* in the field.

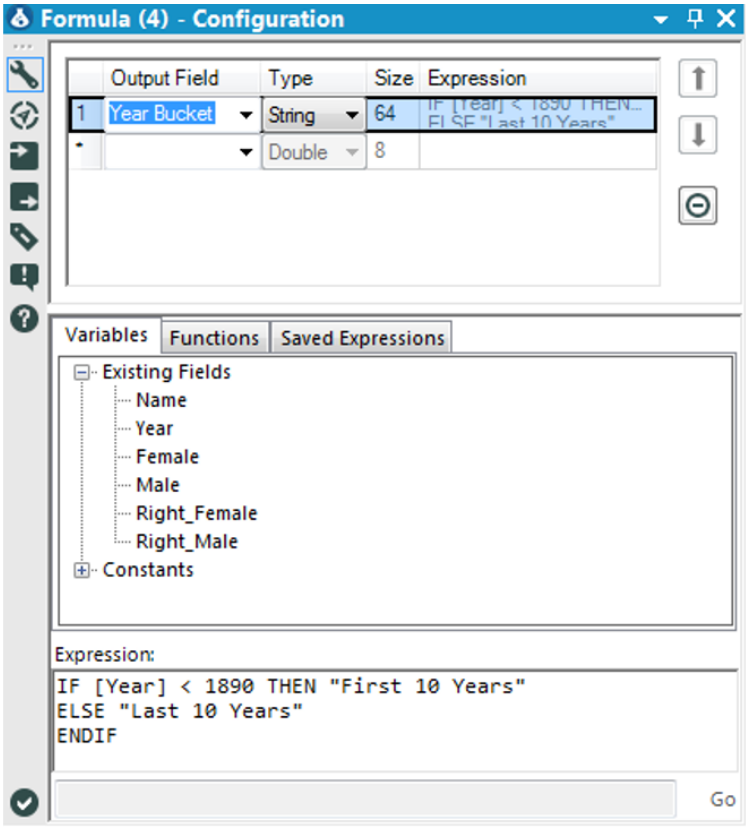


Figure 3-17 – Gender Swapped Formula configuration

We need to change the field type to string with a conditional formula to test if the record is in the first 10 years. Since the data is already filtered, we do not need to test the other values.

Now that we identified the data with the *Year Bucket* field, we can summarize the data to find out the total number of babies given each name during the 10 years we want to analyze.

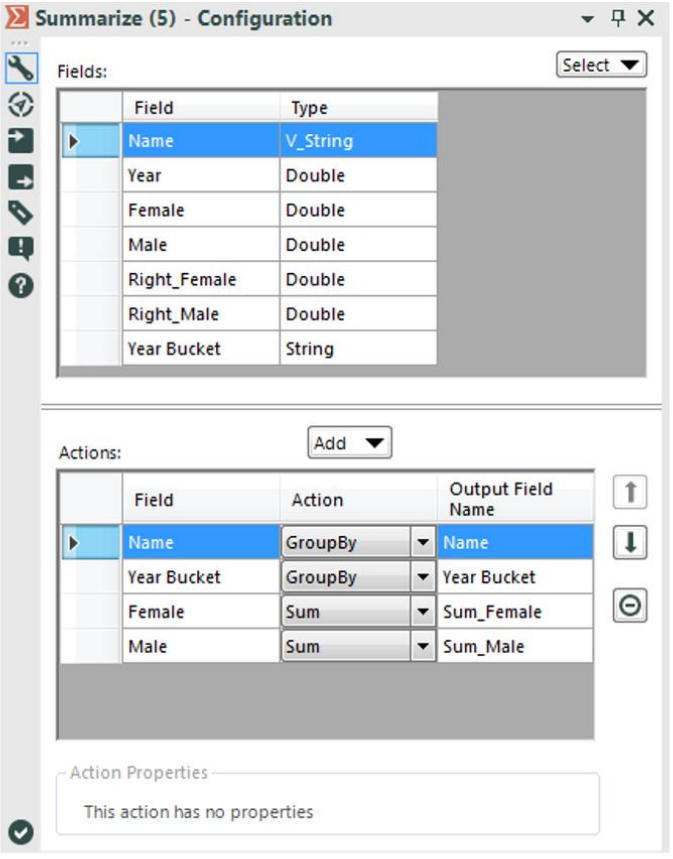


Figure 3-18 – Gender Swapped Summarize configuration

The data stream after *Summarize* looks as below:

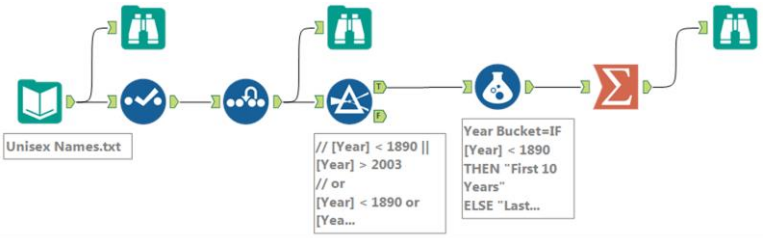


Figure 3-19 – Gender Swapped Data stream after summarize

We will next run the workflow and look at the *Browse* tool to see the data we have.

Results - Browse (6)

4 of 4 Fields | Cell Viewer

Record #	Name	Year Bucket	Sum_Female	Sum_Male
1	Addison	First 10 Years	0	170
2	Addison	Last 10 Years	81476	2803
3	Adrian	First 10 Years	0	148
4	Adrian	Last 10 Years	1603	71931
5	Adriel	Last 10 Years	258	4376
6	Aidan	Last 10 Years	1025	65368
7	Aiden	Last 10 Years	1179	131797
8	Alexis	First 10 Years	0	11
9	Alexis	Last 10 Years	90995	23277
10	Ali	Last 10 Years	2930	8056

Figure 3-20 – Gender Swapped Browse configuration

From the *Browse* tool, we can see that we have the data structured in a way we need. However, since we were asked to find the baby names that were previously only assigned to one sex and now are only assigned to the other, we need a way to check each of the names to see if they meet these criteria.

We will do this by using *Multiple Row Formula* to identify which records should be kept. Since we are using a Multi Row

Formula, we need to make sure the data is in the right order. So, we will first use a *Sort* tool.

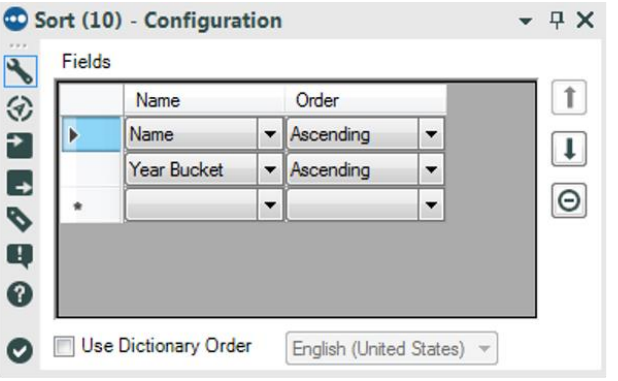


Figure 3-21 – Gender Swapped Sort configuration

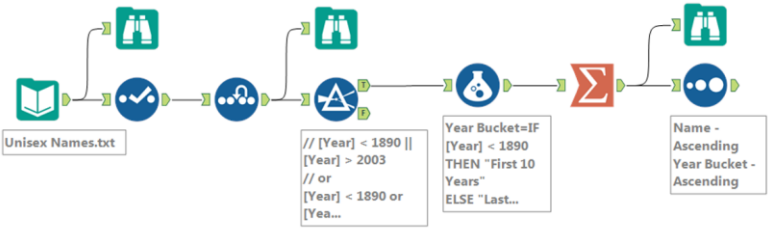


Figure 3-22 – Gender Swapped data stream after Sort

Now that we have made sure the data is sorted, we need to tackle the complex question. In order to do so, we will break it down into smaller, simpler ones:

- Which names occur in both the first and last 10-year lists?
- Which names have at least one count of zero?
- Which names have a female count alternate between something and zero?
- Which names have a male count alternate between something and zero?

By asking these four relatively simple questions, and filtering out the data that does not meet the criteria, we can answer the complex question we have been asked. We will use the same field as we move forward to filter the data, which we will call *Potential Swap*. We will create it in the first *Multi-Row Formula* tool and update it in all subsequent ones. Let us see how we can create this formula.

At this point, much of this formula should be familiar. We are looking at a conditional statement that tests two things, and if both are true, we write *Remove*; else, we write *Check*. What is different about this is that we have special operators in the field names *Row-1:* and *Row+1:*. These allow us to look at the row (record) above and below the current one so we can use the value there. In this case, we are checking to see if the name of the current row (*[Name]*) equals the previous or following rows.

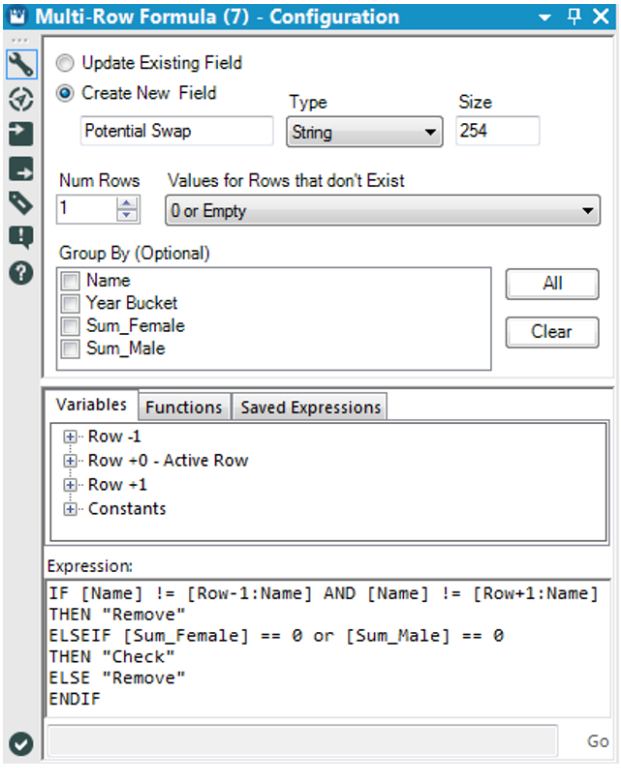


Figure 3-23 – Gender Swapped Multi-row formula configuration

Those familiar with Boolean logic may have understood what the formula here actually does. The formula checks to see if the *[Name]* does not match both the previous and following rows. This is a logically equivalent statement called the *contrapositive*, and we are bringing it up here to demonstrate two things: First, *!=* is the *logical operator* that means *does not equal*. The second is to show if we are having trouble with a logical statement. There are multiple ways one can approach it.

Between each *Multi-Row Formula*, we can add a *Filter* to improve speed by removing records we know are not needed for

analysis. Let us add a series of alternating *Filters* and *Multi-Row Formulas* to answer the remaining three questions.

All *Filter* configuration windows for the next three steps should look like the image represented here.

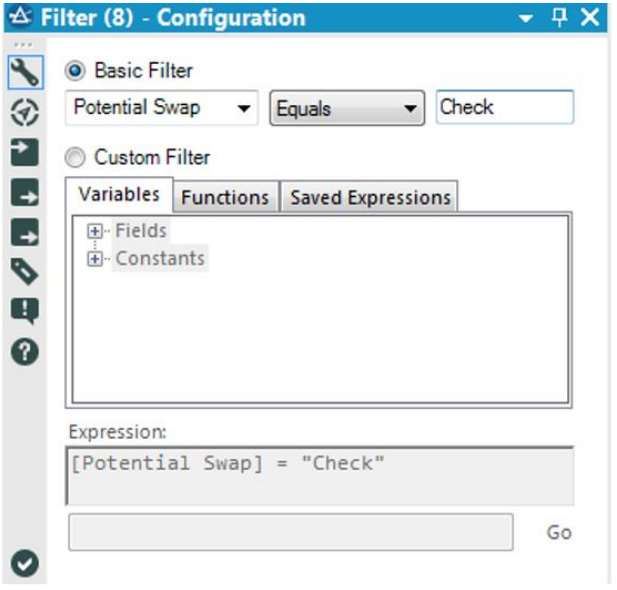


Figure 3-24 – Gender Swapped Filter configuration

The remaining *Multi-Row Formula* properties windows should look like the provided image, with the following table of equations in the expression box.

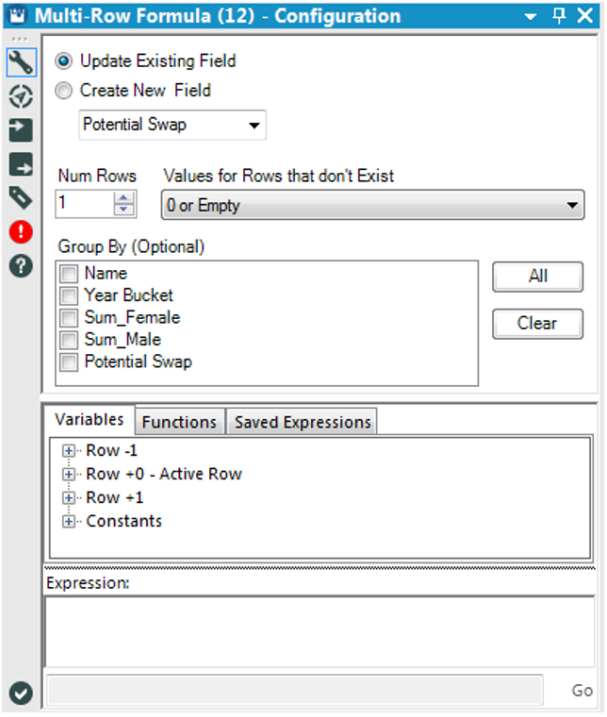


Figure 3-25 – Gender Swapped remaining Multi-row formula

2 IF [Sum_Female] == 0 or [Sum_Male] == 0

THEN "Check"

ELSEIF [Name] == [Row-1:Name] AND ([Row-1:Sum_Female] == 0 or [Row-1:Sum_Male] == 0) THEN "Check"

ELSEIF [Name] == [Row+1:Name] AND ([Row+1:Sum_Female] == 0 or [Row+1:Sum_Male] == 0) THEN "Check"

ELSE "Remove" ENDIF

3 IF [Name] = [Row-1:Name] AND [Sum_Female] != 0 AND [Row-1:Sum_Female] != 0

THEN "Remove"

ELSEIF [Name] = [Row+1:Name] AND [Sum_Female] != 0 AND [Row+1:Sum_Female] != 0

THEN "Remove"

ELSE "Check" ENDIF

4 IF [Name] = [Row-1:Name] AND [Sum_Male] != 0 AND [Row-1:Sum_Male] != 0

THEN "Remove"

ELSEIF [Name] = [Row+1:Name] AND [Sum_Male] != 0 AND [Row+1:Sum_Male] != 0

THEN "Remove"

ELSEIF [Name] = [Row+1:Name] AND [Sum_Male] !=0 AND [Row+1:Sum_Male] !=0

THEN “Remove”

ELSE "Check" ENDIF

Figure 3-26 - Gender Swapped Formula

At this point, the data stream is quite long, so we have moved the tools below one another to make it easier to see.

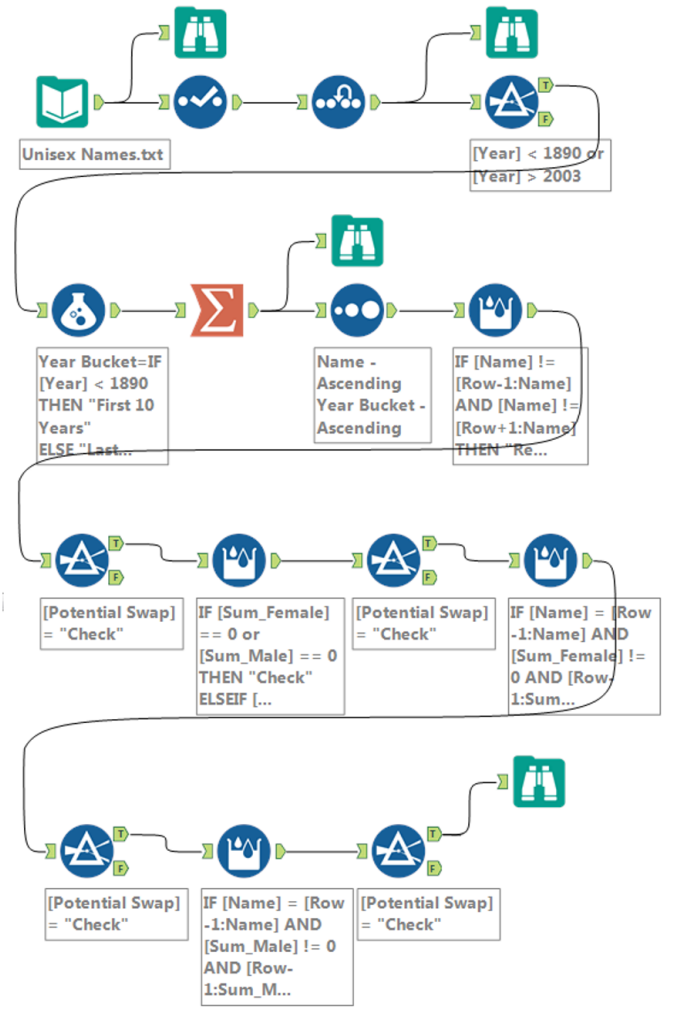


Figure 3-27 – Gender Swapped data stream

We will notice that a *Browse* tool has been added here. Typically, if the formulas are being tested, then a *Browse* tool should be used on both the *True* and *False* outputs of every *Filter*

tool. However, since these formulas were already tested, we are just going to take a look at the end result.

Results - Browse (15)

5 of 5 Fields | Cell Viewer

Record #	Name	Year Bucket	Sum_Female	Sum_Male	Potential Swap
1	Addison	First 10 Years	0	170	Check
2	Adrian	First 10 Years	0	148	Check
3	Alexis	First 10 Years	0	11	Check
4	Allison	First 10 Years	0	70	Check
5	Andrea	First 10 Years	51	0	Check
6	Angel	First 10 Years	0	8	Check
7	Antonia	First 10 Years	276	0	Check
8	Asa	First 10 Years	0	532	Check
9	Ashley	First 10 Years	0	80	Check
10	Ashton	First 10 Years	0	12	Check

Figure 3-28 – Gender Swapped Browse configuration

We can now see that Beverly and Hilary are the only names that meet our criteria.

However, just like in the *Freestyle Skiing* example, we should make Alteryx give us exactly the results we will give to the newspaper so we do not make a mistake. This time, all we need to do is *Summarize* by the *Name* field to have the results that we are looking for.

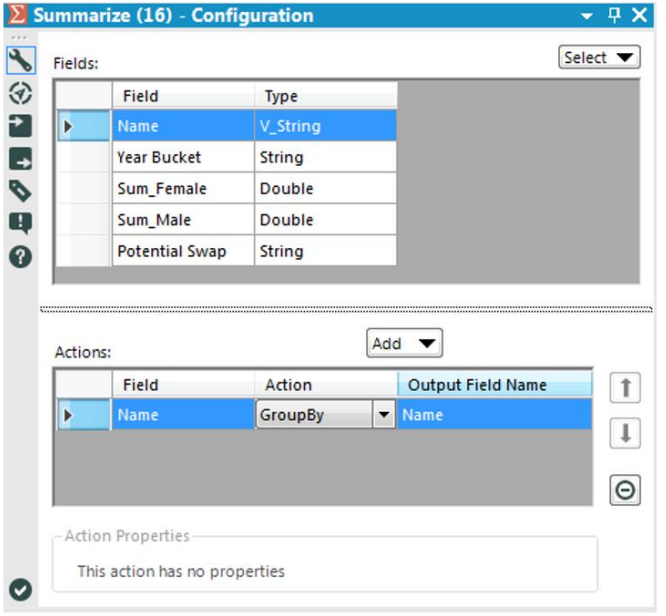


Figure 3-29 – Gender Swapped Summarize configuration

Running the *Browse* tool shows us the following:

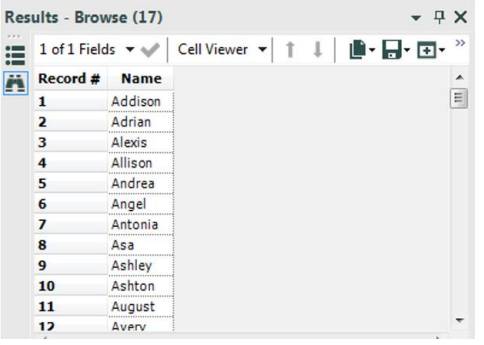


Figure 3-30 – Gender Swapped Browse after summarize

The *Gender Swapped* data stream should look like the image on the next page when it is complete.

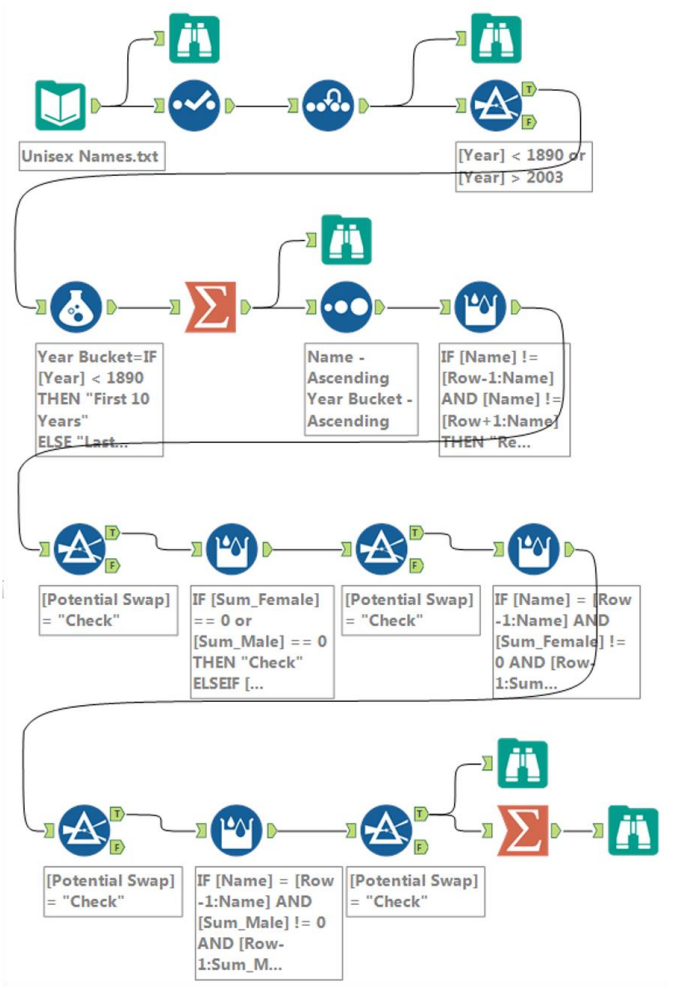
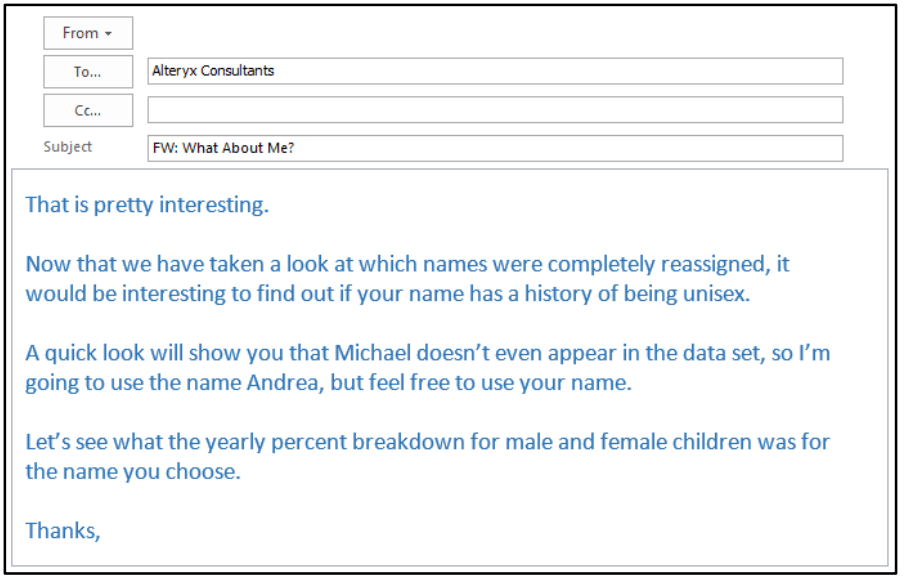


Figure 3-31 – Gender Swapped data stream when complete

3.7 What About Me?



We will add a few things to this list to adhere to best practices, but the necessary steps we need are:

3. Import the data.
4. Clean up the fields.
5. Input a name.
6. Limit the records by that name.
7. Create calculated fields that show us the percent breakdown.
8. Export the data

We already know what issues we have in the dataset, so we can copy the last data stream until the *Imputation* tool.

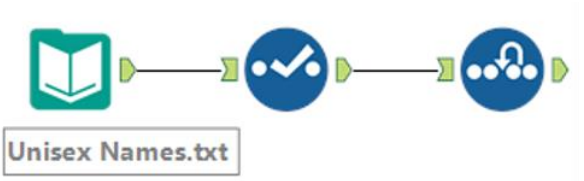


Figure 3-32 – Reuse Gender Swapped for What about me

Now that we have this, the next step is to add the name to the data stream.

If we add a *Text Input* with a column called *Name*, and name as the only record, below is what we will get.

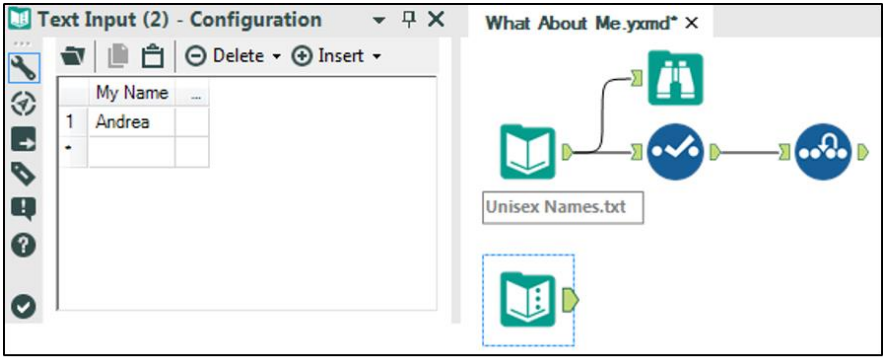


Figure 3-33 –What about me Text input

Since the *Text Input* will show us all of the data, we do not need to add a *Browse*. But it is a good practice to add the *Select* so that we can ensure that the data types from both data streams match.

Unisex Names Select ***Text Input Select***

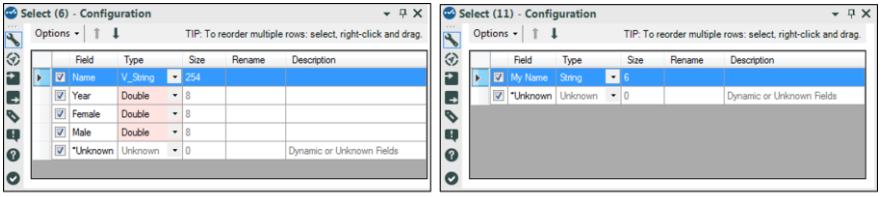


Figure 3-34 –What about me Select configuration

Though we have different types in our data field, we do not need to convert them because they are both strings. If they were not, it would cause an issue in joining the data. Due to the way we have approached this problem, we can use the *Join* tool to filter the data to the appropriate records. So we will take only the joined section and not keep the name field coming out of the right (*Text Input*).

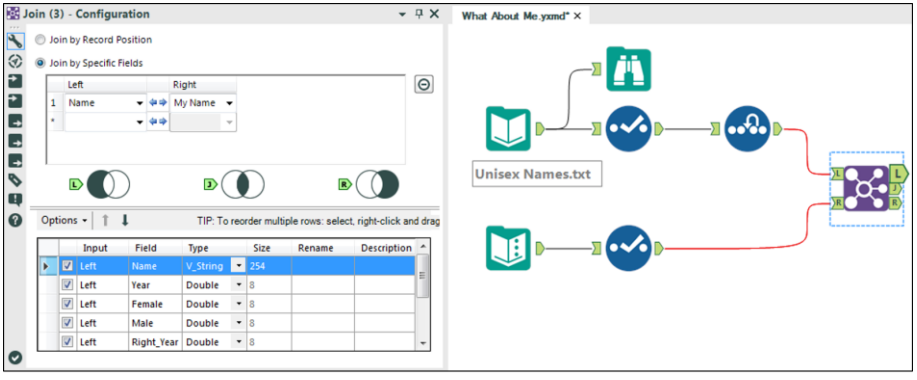
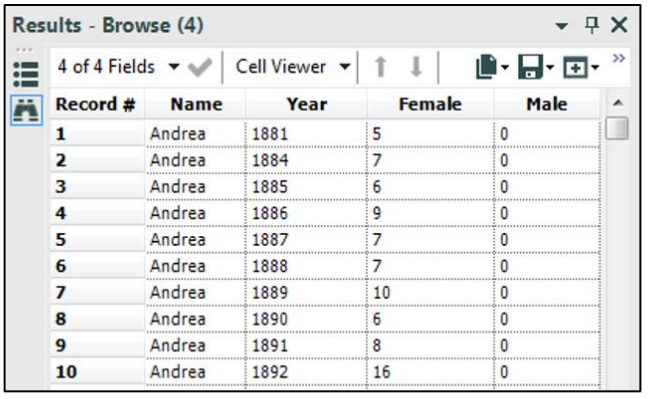


Figure 3-35 –What about me Join configuration

Running the *Browse* tool for the *J* output will show us the following:



Record #	Name	Year	Female	Male
1	Andrea	1881	5	0
2	Andrea	1884	7	0
3	Andrea	1885	6	0
4	Andrea	1886	9	0
5	Andrea	1887	7	0
6	Andrea	1888	7	0
7	Andrea	1889	10	0
8	Andrea	1890	6	0
9	Andrea	1891	8	0
10	Andrea	1892	16	0

Figure 3-36 –What about me Browse configuration

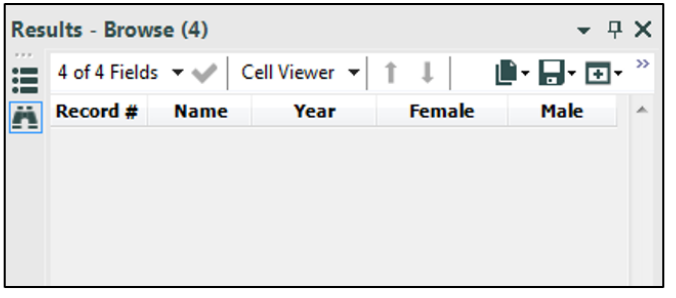


Figure 3-37 –What about me Browse configuration

If we see the image here, we see that no records have been returned. It is because the name entered in the *Text Input* does not match anything in the *Unisex Baby Names* dataset. Change the *Text Input* name to *Andrea* to follow along more easily.

At this point, we have covered steps 1 through 4. The next step is for us to create the percent breakdown for both male and female babies in each year. Since we intend to perform the same calculation on two different fields, we can use the *Multi-Field Formula* tool to accomplish this.

If we add the *Multi-Field Formula* tool after the *J* output from the *Join* with the following configuration, we will be able to create the percent of total for Male and Female babies each year.

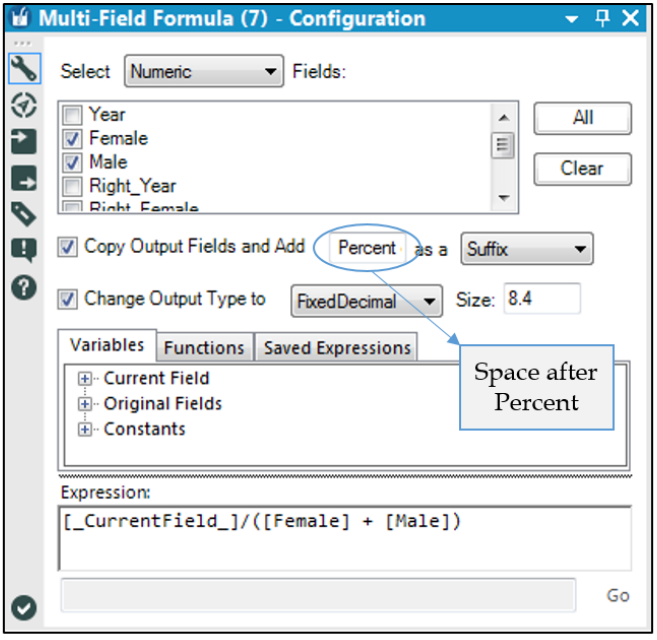


Figure 3-38 –What about me Multi-field formula configuration

Now, all that's left is to write the file out to *What About Me.csv*.

The *What About Me* data stream should look like the below image when it is complete.

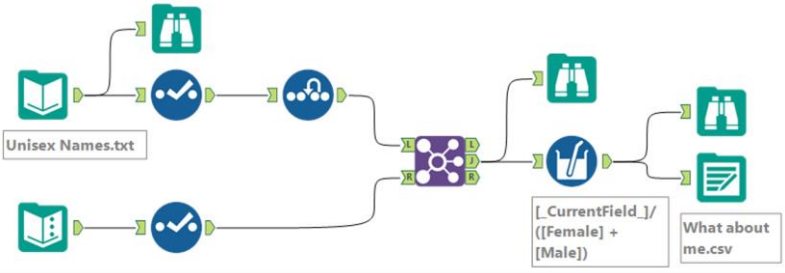
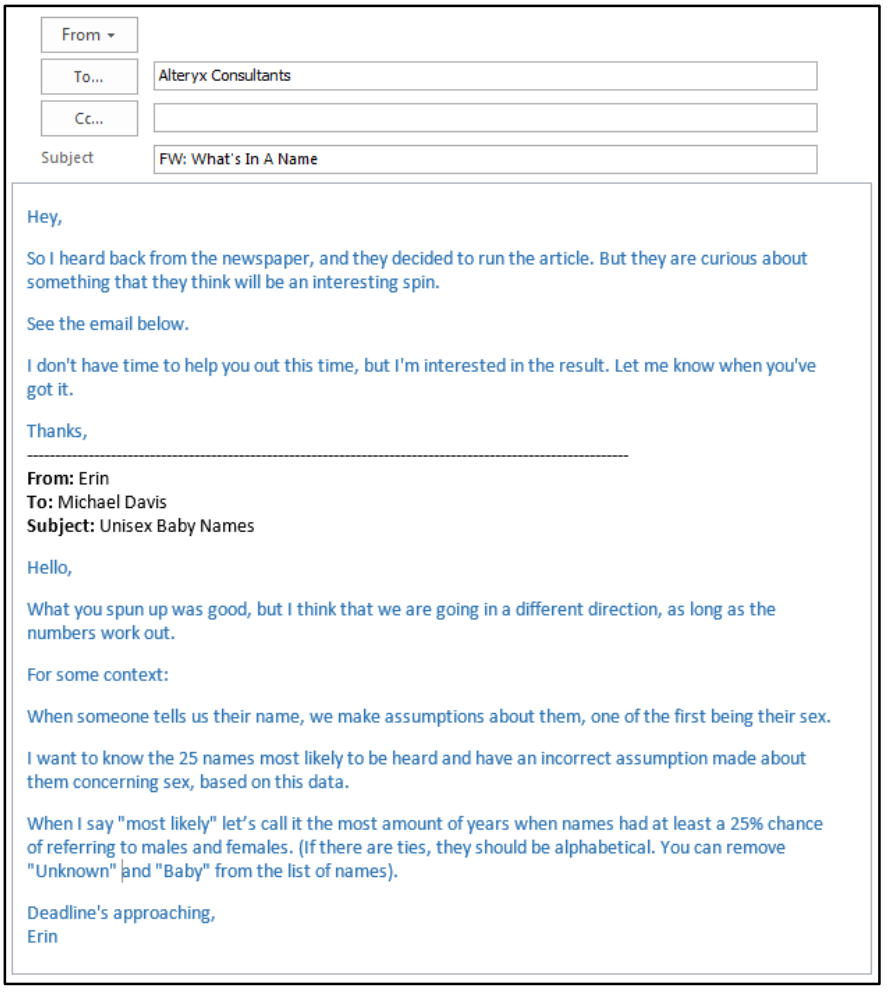


Figure 3-39 – What about me data stream when complete

3.8 What's In A Name?



From

To... Alteryx Consultants

Cc...

Subject FW: What's In A Name

Hey,

So I heard back from the newspaper, and they decided to run the article. But they are curious about something that they think will be an interesting spin.

See the email below.

I don't have time to help you out this time, but I'm interested in the result. Let me know when you've got it.

Thanks,

From: Erin
To: Michael Davis
Subject: Unisex Baby Names

Hello,

What you spun up was good, but I think that we are going in a different direction, as long as the numbers work out.

For some context:

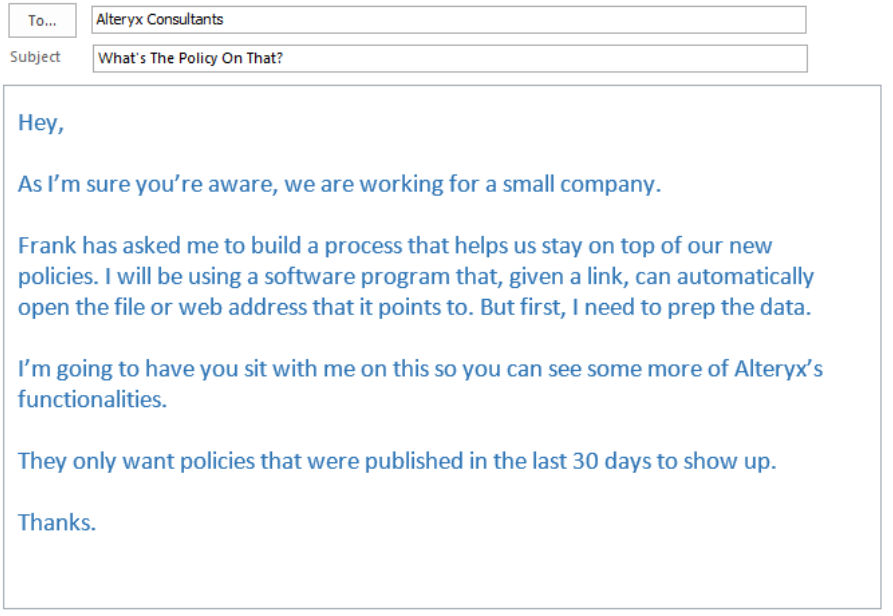
When someone tells us their name, we make assumptions about them, one of the first being their sex.

I want to know the 25 names most likely to be heard and have an incorrect assumption made about them concerning sex, based on this data.

When I say "most likely" let's call it the most amount of years when names had at least a 25% chance of referring to males and females. (If there are ties, they should be alphabetical. You can remove "Unknown" and "Baby" from the list of names).

Deadline's approaching,
Erin

CHAPTER 4
The Direct Approach



4.1 Tools

Tools	**Concepts**
Append Fields	Appending Data
Date Time Now	Cartesian Product
DateTime	Working With Dates
Directory	Working With Directories

4.2 Append Fields

Figure 4-1 – Append fields

The *Append* tool adds each record from *S (source)* to the end of each record in *T (target)*.

Group	Input	Output
Join	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input S: Any data stream.

Input T: Any data stream.

Output: A data stream of records that has the Cartesian product from inputs *T* and *S*, with the records of *S* appended to the end of *T* records.

This tool is very useful for appending a single data value onto a full dataset, such as appending the mean of a field to every row for calculation purposes. Since we are taking the Cartesian product of two data streams, we run the risk of creating a very large data stream, which could cause memory errors.

Properties Window:

The *Append Fields Configuration* window is identical to the *Select Configuration* window, with two exceptions.

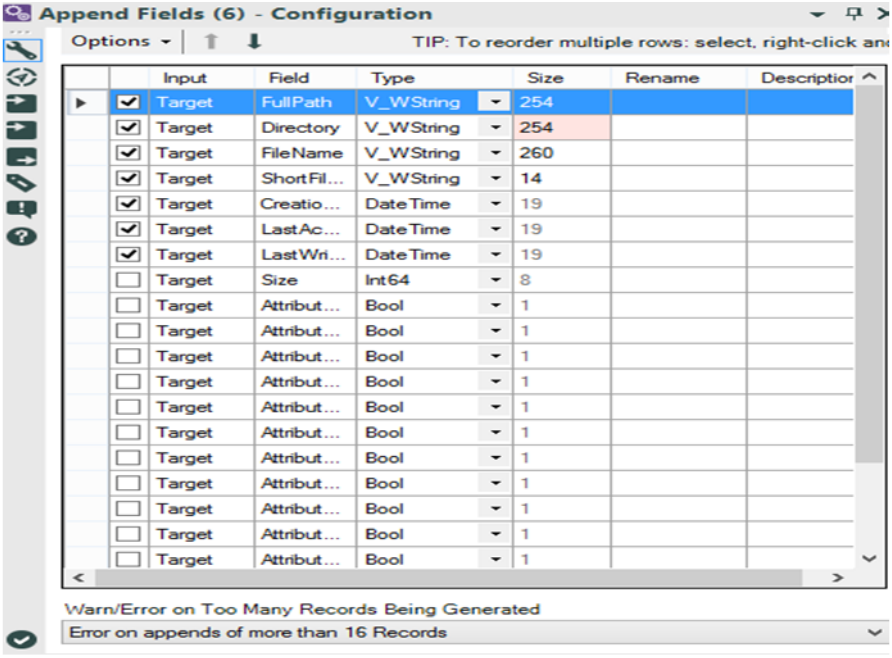


Figure 4-2 – Append fields configuration

- There is an additional element in the metadata section called *Input*. It identifies if the data is coming from the *Target (T)* or *Source (S)* inputs for the tool.
- There is a *Warn/Error on Too Many Records Being Generated*, which allows us to decide if and how we should be alerted to a high rate of replication of the *Target* field.

4.3 Date Time Now

Figure 4-3 – Date time now

The *Date Time Now* tool gets the system time when the module starts executing.

Group	Input	Output
In/Out	None	String

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A string representing the date or the date and time, depending on the settings.

The tool is useful if we want to limit our data stream based on the current time, or if we want to publish the time of a report or dataset when created.

Properties Window:

The *Date Time Now Configuration* window allows us to select the format of the string required in the output.

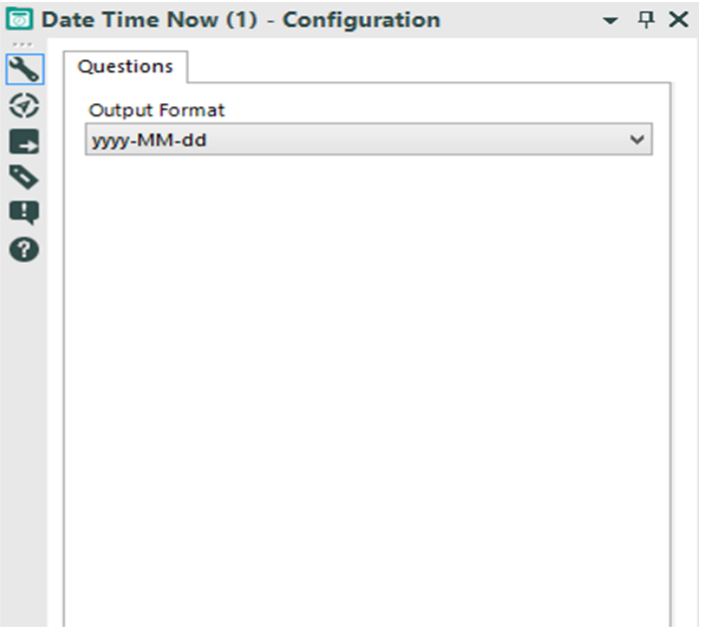


Figure 4-4 – Date time now configuration

The following image represents the format of the strings available. See *Appendix G* for a list of character descriptions.

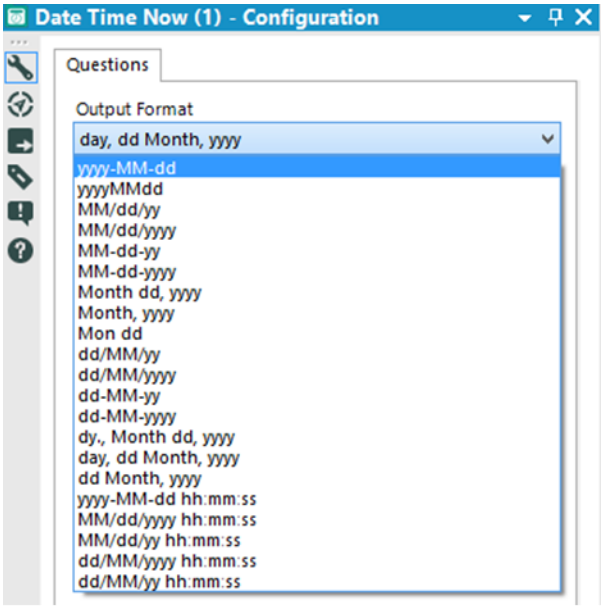


Figure 4-5 – Date time now formats

4.4 DateTime

Figure 4.6 – Date time

The *DateTime* tool converts between String and Date format fields

Group	Input	Output
Parse	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with a date field in a *string*, *date*, or *date/time* field.

Output: The original data stream with a new field. The new field cane be a *string* representing the date or the date/time depending on a *date* or *date/time* field that was represented as a string field.

The tool can only convert the formats in the menu. We need to write a *Formula* that does string conversions in addition to using this tool if we need a different type of date format.

Properties Window:

The *DateTime Configuration* window has four components.

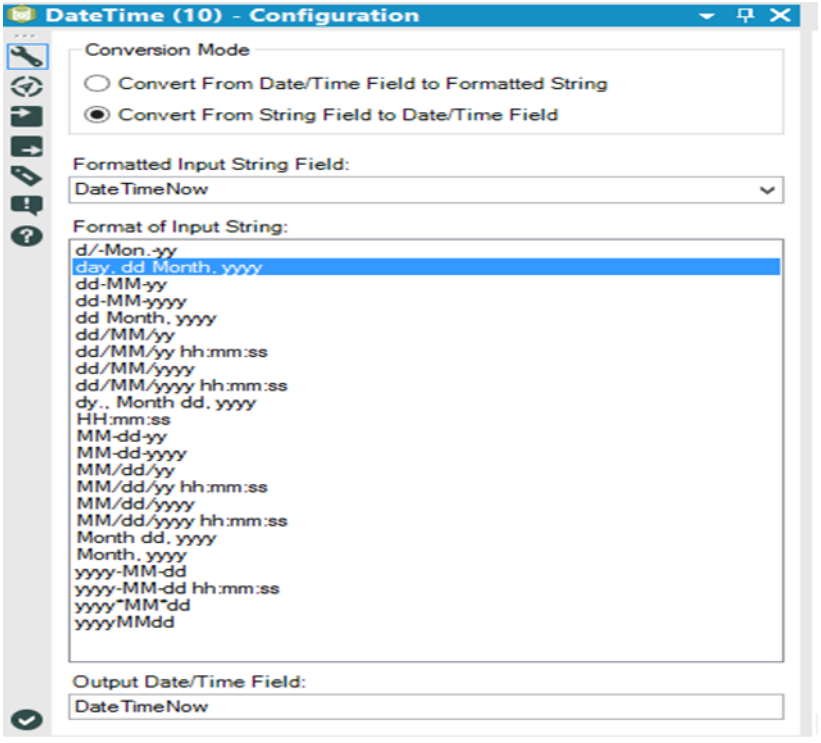


Figure 4-7 – Date time configuration

- *Conversion Mode* allows for converting from or to a string field.
- *Formatted Input String Field (Input Date/Time Field to be formatted)* is the field we want to convert.
- *Format of Input String (Desired Format of Output String)* is the format the input string is in.
- *Output Date/Time Field (Output Formatted String Field)* lets us name the field we are creating.

4.5 Directory

Figure 4-8 - Directory

The Directory tool creates a data stream that has the contents of a directory.

Group	Input	Output
In/Out	None	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream containing metadata about all of the files in a given directory.

We can use special characters to look for a file meeting a pattern. For example, *.* will return all files, and *a *.csv* will return all *.csv* files starting with the letter *a*. These file-matching commands are different from the regular expression syntax, which we will discuss in *Expensive Beauty Products.*

Properties Window:

The *Directory Configuration* window has three components.

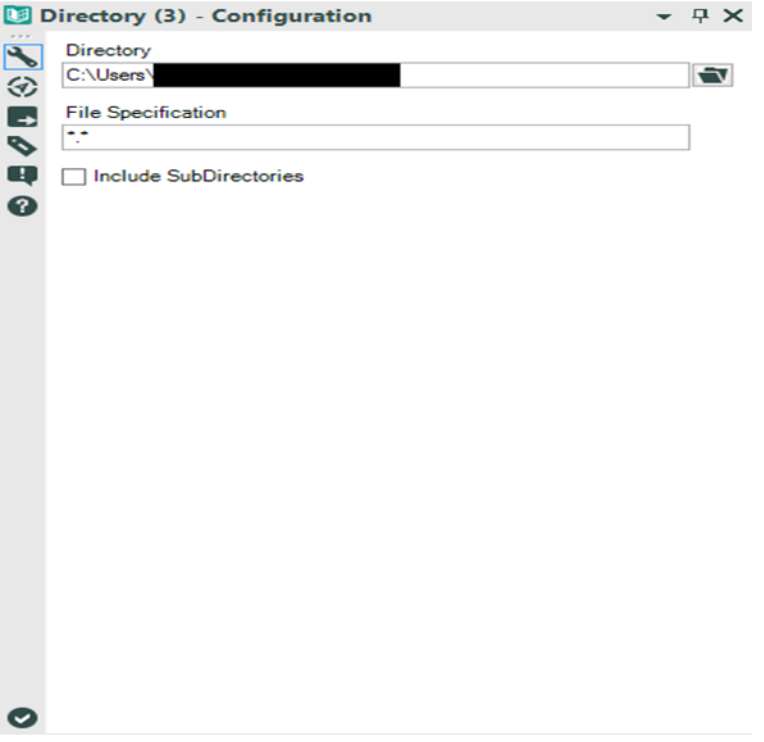
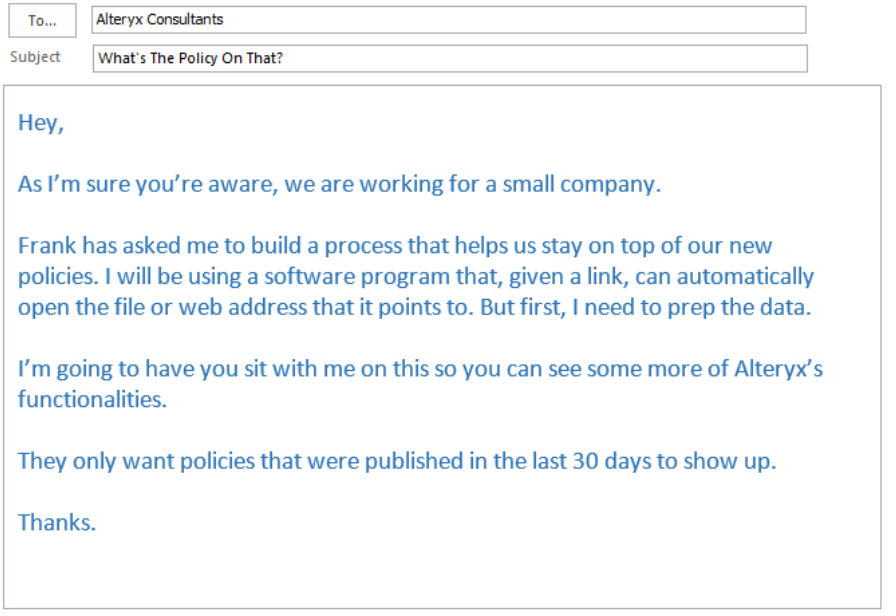


Figure 4-9 – Directory configuration

- *Directory* lets us navigate to a folder.
- *File Specification* allows us to type the generic format for a file name to be found in the *Directory*, where * denotes zero or more characters or spaces, while *?* denotes exactly one character or space.
- *Include SubDirectories*, when checked, includes each of the subfolders when it's looking for files. Otherwise, it will only look in the directory that we have navigated to.

4.6 What's The Policy On That?



Since we need to search for files, we are going to be using the directory tool. Our finalized policies are published as *pdf* files, so we can use that to limit our search. We have three divisions responsible for publishing policies right now, and they all publish them to subfolders in *Chapter 4 – The Direct Approach* > *What's The Policy On That*.

The plan is to:

- Bring in a list of all of the *PDF* files in the directory.
- Get the current date.
- Remove files created more than 30 days ago.
- Export the data to *What's The Policy On That.csv*.

When we bring the *Directory* tool onto the canvas and navigate to the *Policy* folder in the *Directory* file browse, we can use the expression **.pdf* to find all *PDF* files in the directory. But since we need to search all of the sub-folders (subdirectories), we need to check the box at the bottom of the configuration window.

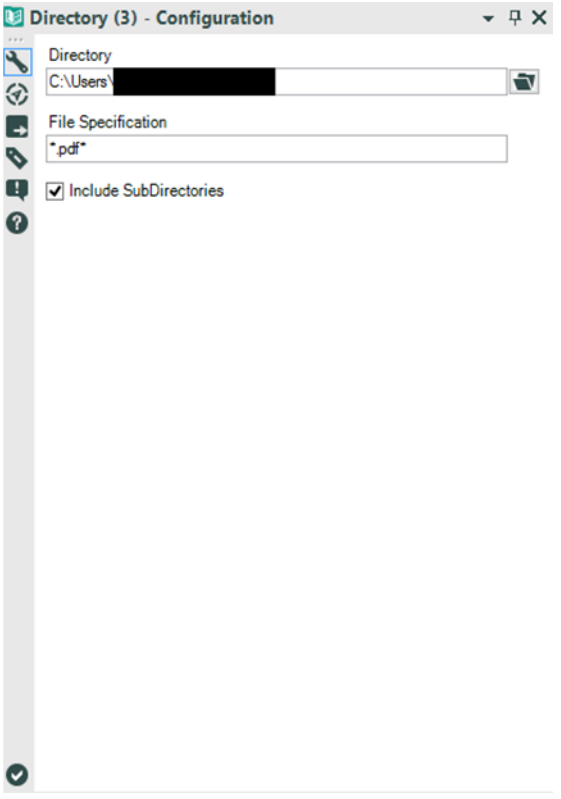


Figure 4-10 – Policy directory configuration

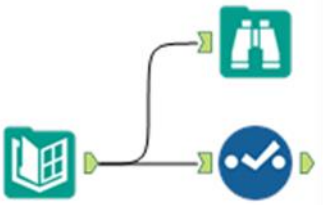


Figure 4-11 Policy Data stream

As usual, we will add a *Browse* and S*elect* tool following the data connection. Let's look at the *Select* tool to see what's in the data stream.

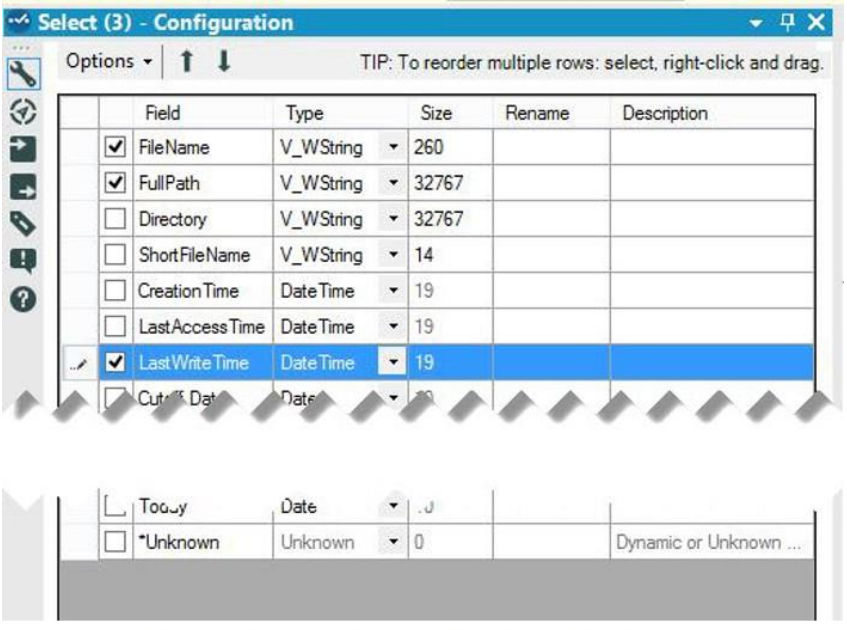


Figure 4.12 – Policy Select Configuration

We can see the data stream has 19 fields. Whenever we use the *Directory* tool, it will return these 19 metadata fields about each of the files that were found. See *Appendix K* for details on each.

In our case here, we will only be working with the *FullPath*, *FileName*, and *LastWriteTime* fields. Now that we have the appropriate information from the list of files, the next thing we need is to add the *current date* into the workflow. This is where the *Date Time Now* tool comes in. Please note here that the annotations are hidden to save space.

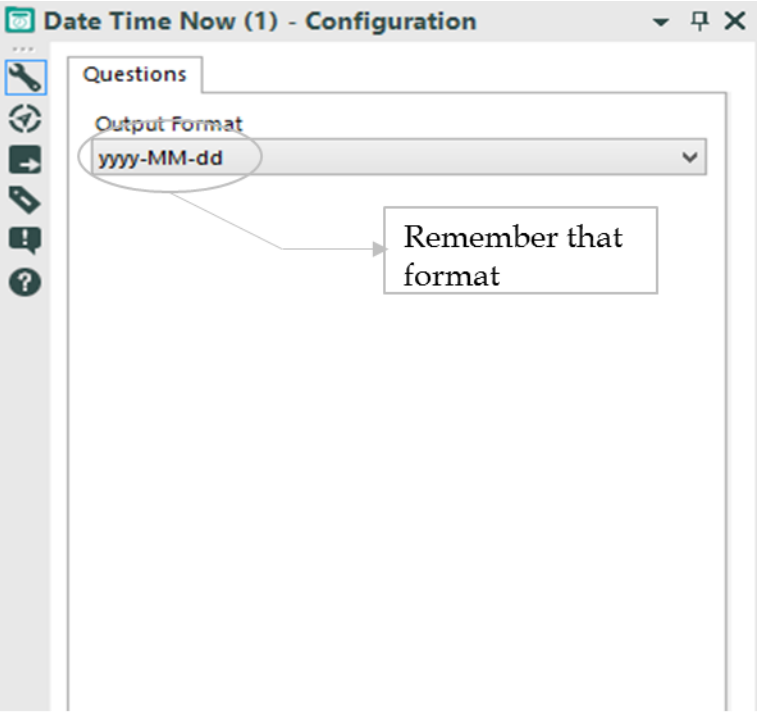


Figure 4-13 – Policy Date time now configuration

We will set the output format of the *Date Time Now* input to *yyyy-MM-dd* so it's easy for us to know which format it's in when we are converting it to the cutoff date 30 days ago. It's important to note that the *Date Time Now* tool creates a single field with a single record, which is a string representing the date when the module runs.

Running the *Browse* and *Select* tools show us the following information.

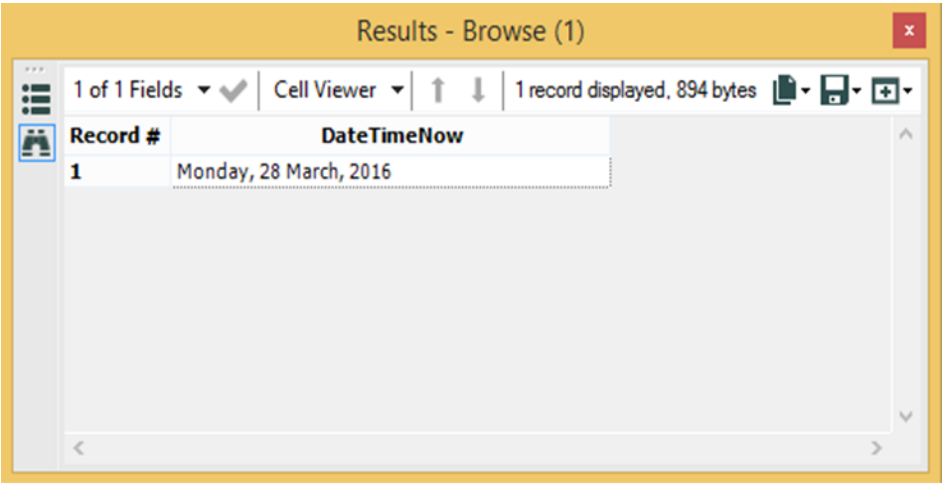


Figure 4-14 – Policy browse configuration for Date time now

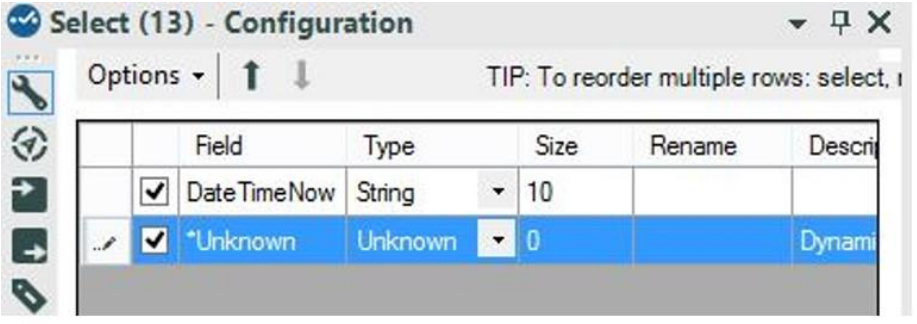


Figure 4.15 – Policy Select Configuration for Date time now

In order to convert this string into a *DateTime* field, we need to use the *DateTime* tool. So if we add it right after the *Select* tool like the following image, we can create a new field called *Today* that has today's date as a *Date* field.

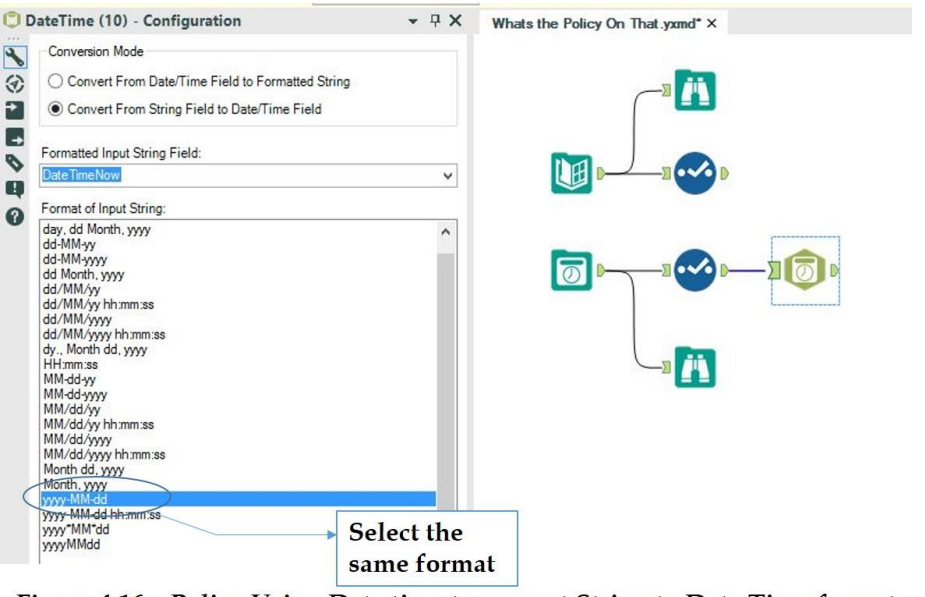


Figure 4.16 – Policy Using Date time to convert String to Date Time format

Now we need to find out what the date 30 days ago was. In order to do this, there's a simple *Formula* function we can use. If we look under the *Functions* tab of the *Formula Configuration window*, we see that one of the branches is called *DateTime* and it has the formulas we can use to manipulate *Date*, *Time*, or *DateTime* fields. We want to find out what the date 30 days before today was. We need to remember that we must add a negative interval of days when using the *DateTimeAdd* function to obtain a previous date.

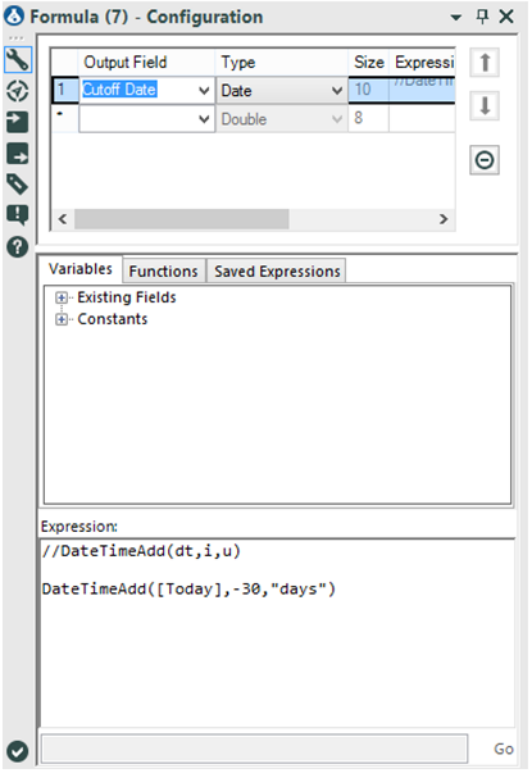


Figure 4-17 – Policy using Formula to find Date 30 days ago

A description and example of how the tool works is displayed when we click on the function. A double-click on it shows us to add the *DateTimeAdd(dt,i,u)* expression to the *Expression* box.

- *dt* should be replaced with a *Date*, *Time*, or *DateTime* field.
- *i* should be replaced with the interval we want to add. (-30 for 30 days ago).
- *u* should be replaced with a unit keyword from the list in Appendix H. In this case, it is "days."

Now that we have these two data streams ready to be combined, the question becomes: What method do we use?

A *Union* would allow us to combine the data. Unfortunately, since we will be performing a calculation to see if the dates were in the last 30 days, we need the *Cutoff Date* in every record not added to the end of the dataset. A union will not work.

A *Join* would allow us to do this if we had a field we could match, which means we could use a function on both data streams that just writes 1 to the field, and then join on that. But that's difficult to explain and maintain.

Fortunately, Alteryx has a tool called *Append Fields* that does exactly what we want to do with the *Formulas* and a *Join*, in an optimized way. We shall connect the list of *PDFs* to the *Target (T)* input and the date to the *Source (S)* input. Since we are changing the structure of the data by combining two data streams, we should put a *Browse* tool following the *Append Fields* tool to follow the output at this stage.

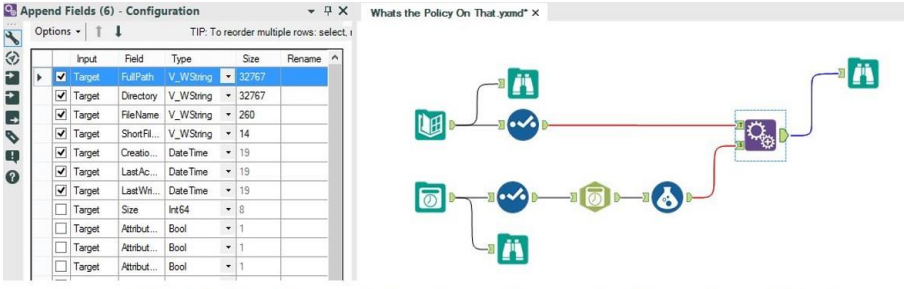


Figure 4.18– Policy Append Configuration to Use Formulas with Join

As we can see, *DateTimeNow* and *Today* fields can be excluded when we were joining the data. This is because they are no longer needed, as we only need to compare the *LastWriteTime* to the *Cutoff Date* to determine if the file should be included in the list.

Now we can compare the *LastWriteTime* to the *Cutoff Date*. If it is before the cutoff date, we don't need those records. The way we're going to approach this is a little different from what we've done so far. Let's write a formula in the *Filter* tool to satisfy the condition.

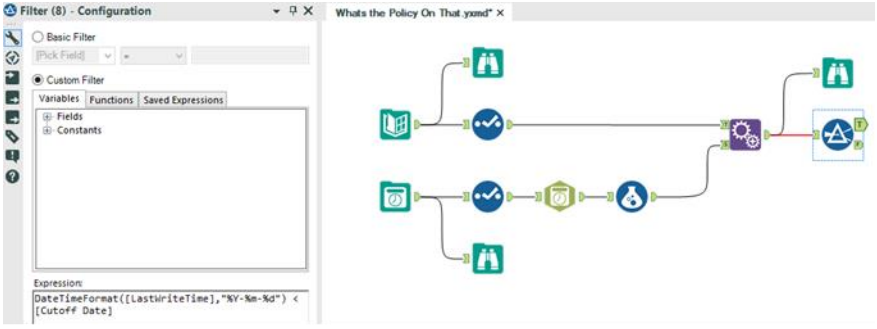


Figure 4-19 – Policy Comparing Last Write Time to Cutoff Date

This looks normal. However, notice that the formula is true any time *LastWriteTime* is less than (before) the *Cutoff Date*. This is because we can easily use the *False (F)* output to finish the data stream similar to the way we can use the *True (T)*.

Now, let's output the data to a file called *What's The Policy On That.csv*.

The *What's The Policy On That* data stream should look like the below when it's complete.

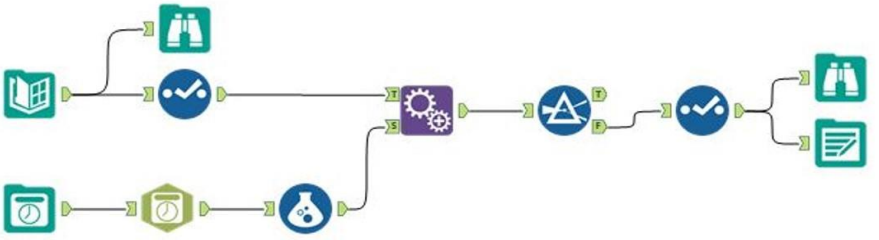


Figure 4.20– Policy data stream when complete

4.7 Where, Oh Where Have The Three Files Gone?

To... Alteryx Consultants

Subject Where, Oh Where Have The Three Files Gone?

Hey,

I know this is a little unorthodox, but it will be a good way to score points with me and my boss.

Usually, I would take care of things like this, but Frank needs me in a meeting.

Frank has lost some important files, and they don't remember what the files are called.

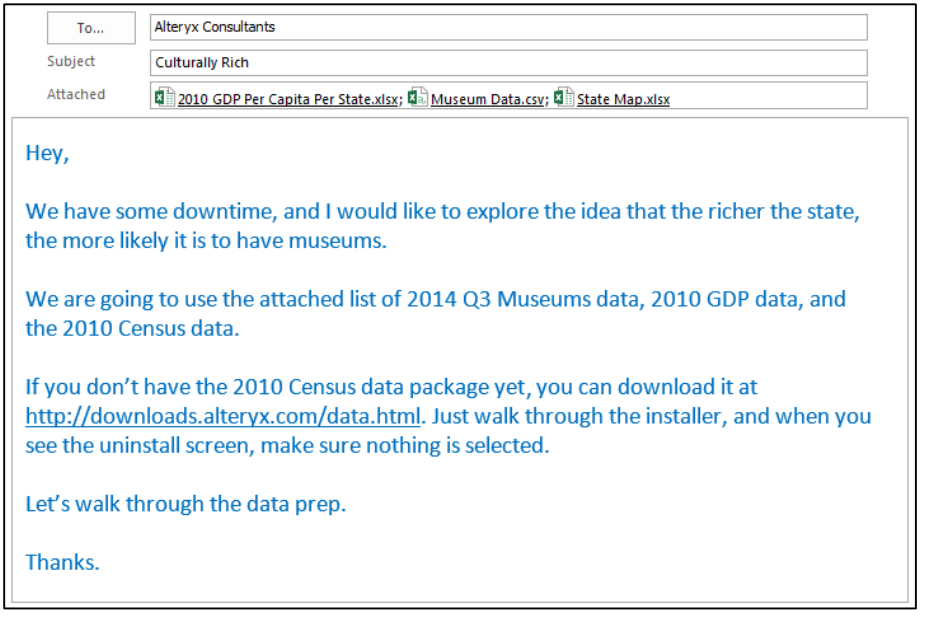
Frank knows they saved the files somewhere on their network drive under a particular folder each, and that you will recognize the names of the files as soon as you find them, but searching the network drive is something Frank doesn't have time to do.

I need you to build a directory and sub directory search for an Excel file (.xlsx) in the "Excel Files" folder, a PDF file (.pdf) in the "PDF files" folder, and an image file (.png) in the "Images" folder in "Chapter 4 – The Direct Approach" > "Looking for Files." Look at the results of each of the three independent file lists, and identify the ones we need.

After that, bring the data together into an output file with the two columns "File Name" and "Full Path", so that by the time the meeting is over, we can direct Frank right to the files.

I really appreciate this.

CHAPTER 5
Cultural Musings


To... Alteryx Consultants
Subject Culturally Rich
Attached 2010 GDP Per Capita Per State.xlsx; Museum Data.csv; State Map.xlsx

Hey,

We have some downtime, and I would like to explore the idea that the richer the state, the more likely it is to have museums.

We are going to use the attached list of 2014 Q3 Museums data, 2010 GDP data, and the 2010 Census data.

If you don't have the 2010 Census data package yet, you can download it at http://downloads.alteryx.com/data.html. Just walk through the installer, and when you see the uninstall screen, make sure nothing is selected.

Let's walk through the data prep.

Thanks.

5.1 Tools & Concepts

Tools	Concepts
Allocate Input	Using Census Data
Find Replace	Data Mapping
Join Multiple	Complex Joins
Text To Columns	Splitting field members at delimiters

5.2 Allocate Input

Figure 5-1 – Allocate Input

The *Allocate Input* tool allows us to get input from demographic data using specific packages.

Group	Input	Output
Demographic Analysis	None	See below

Note: Specific packages have to be downloaded to use this tool. The *Census Data* is a free package to use here.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream that contains the selected fields from an Allocate Dataset.

Properties Window:

The *Allocate Input Configuration* window has three core components and two sub-components for selection and customization.

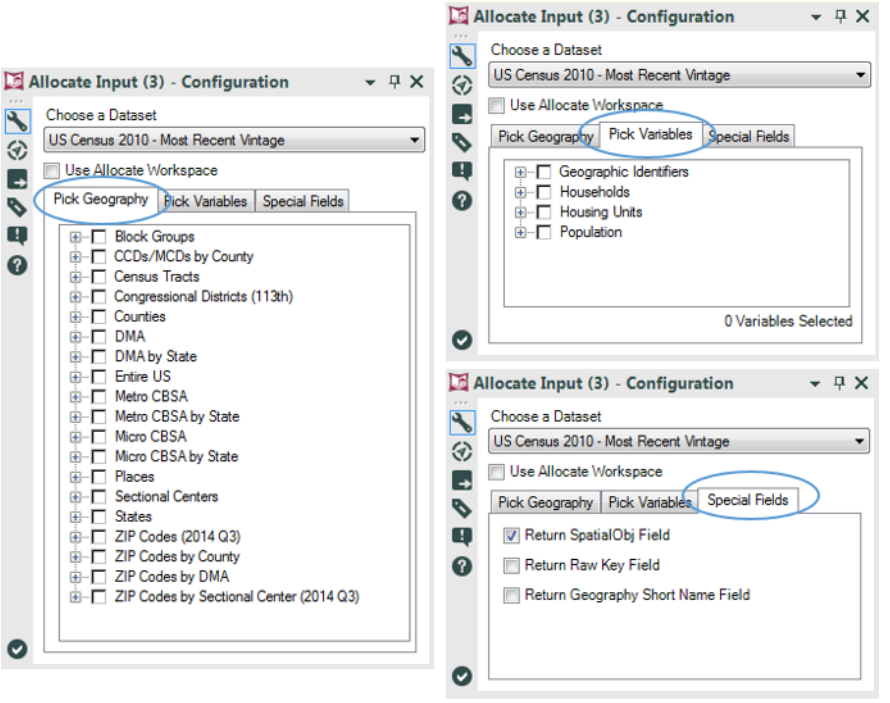


Figure 5-2 – Allocate Input Configuration

Choose a Dataset drop-down allows us to select the dataset associated with one of the packages we have downloaded.

Use Allocate Workspace check box helps us define the fields and geographies we want to use.

The selection window is broken down into tabs. We see different tabs depending on whether or not the *Use Allocate*

Workspace is checked. When unchecked, we see the *Pick Geography*, *Pick Variables* and *Special Fields* tabs.

- *Pick Geography* tab allows us to select the geographic level of the data required.
- *Pick Variables* tab allows us to select the demographic variables needed in the output.
- *Special Fields* tab allows us to select the required special fields relevant to our selections in the output.

When *Use Allocate Workspace* is checked, we see the *Workspace* and *Special Fields* tabs.

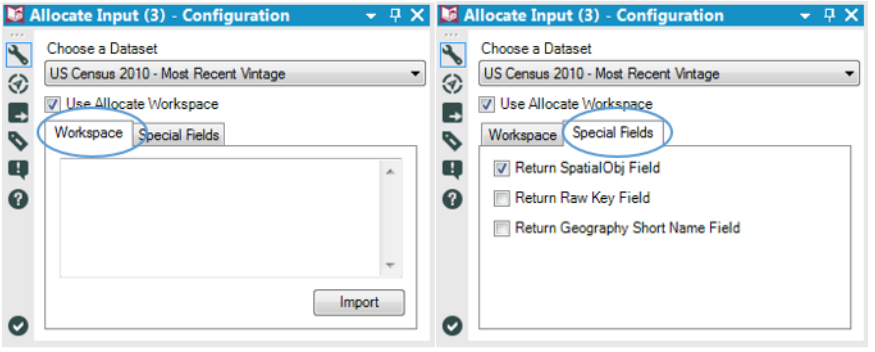


Figure 5-3 – Allocate Input Configuration (contd.)

- *Workspace* can import and modify the *Allocate Workspace* file (.aws), which allows us to save tree-based selections.
- *Special Fields* allows us to select the special fields needed in the output.

5.3 Find Replace

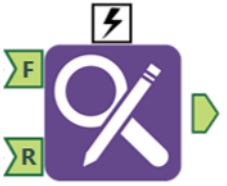


Figure 5-4 – Find Replace

The *Find Replace* tool allows us to replace information in a data stream by entering the *F* (find) (target) input with information that matches in the *R* (replace) (source) input.

Group	Input	Output
Join	See below	See below

Note: The *Find Replace* tool is useful in renaming field entries or for replacing specific information in a data stream. In simple words, it works like a lookup function.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input F: A data stream with at least one string field

Input R: A data stream with at least two fields: one mapping to the string field being replaced in the *F* input, and the other field that's required in the output by replacing the original.

Output: The *F* input data stream with an updated field or the *F* input data stream with additional fields from the *R* input.

Properties Window:

The *Find Replace Configuration* window has two main components: *Find* and *Replace*.

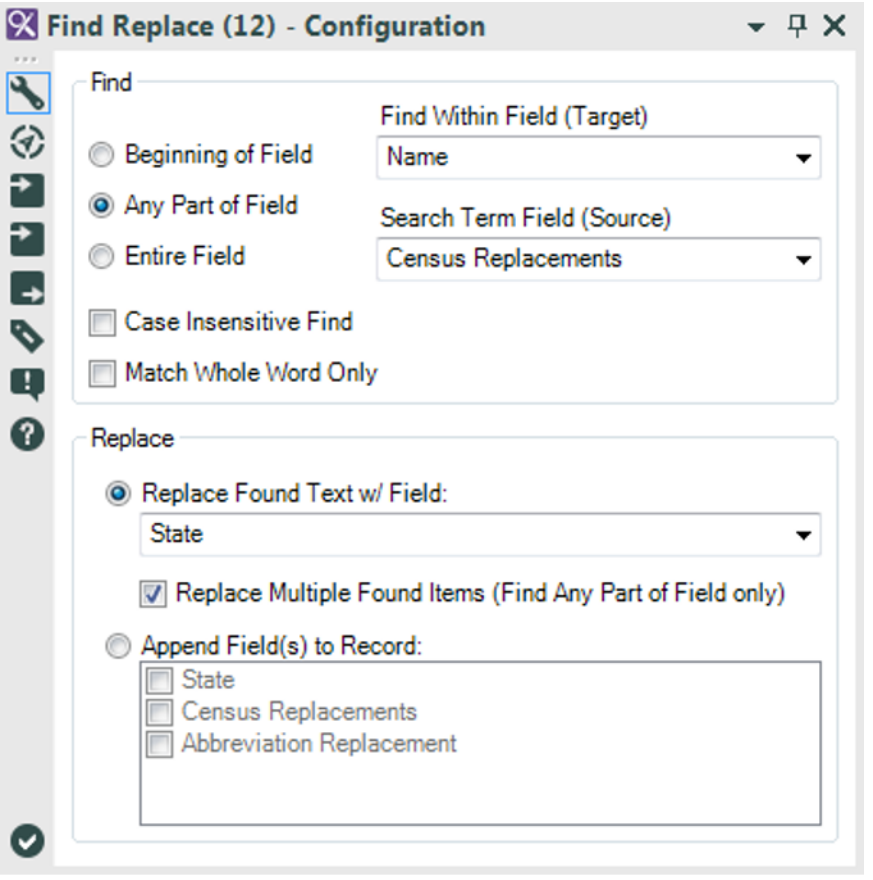


Figure 5-5 – Find Replace Configuration

- *Find* allows us to define the matching criteria.
  - *Find Within Field (Target)* is the field we want to match in the data stream connected to the *F* input.

- *Search Term Field (Source)* is the field that we want to match in the data stream connected to the *R* input.
  We can select one of the three ways to match the two fields:
  - *Beginning of Field* will match only if the *Search Term Field (Source)* matches the beginning of the *Find Within Field (Target)*.
  - *Any Part of Field* will match if the Search Term Field record is found anywhere in the *Find Within Field* record.
  - *Entire Field* will match if the entire *Search Term Field* record matches with the *Find Within Field* record.
- *Case Insensitive Find* allows us to match a field irrespective of uppercase or lowercase in both the sources.
- *Match Whole Word Only* allows us to match only if the *Search Term Field* matches to a word or set of words in the *Find Within Field*. For example, if this option were checked, *apple* would not match *pineapple*, but if it were unchecked, it would match.

- *Replace* allows us to define what the replacement field is
  - *Replace Found Text With Field* allows us to select a field from the R input with which to replace the values.
  - *Replace Multiple Found Items (Find Any Part of Field only)* allows us to choose if we want multiple occurrences of a string to be replaced in a single record.

- *Append Field to Record* allows us to add specific fields to the data instead of just the replaced value.

5.4 Join Multiple

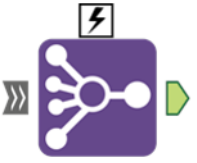


Figure 5-6 – Join Multiple

The *Join Multiple* tool will perform a full outer join or an inner join between data streams sharing a set of key fields.

Group	Input	Output
Join	See below	See below

Note: If there are multiple records that match each other (based on the joining field(s)), records from the original data streams be replicated.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

The *Input* must have at least two data streams with the same mapping fields.

The *Output* will be a data stream with all records joined by the key fields. It can either include all records from each data stream or only those that match across all data sources.

Properties Window:

The *Join Multiple Configuration* window looks and functions similarly to the *Join Properties* window with only the following three differences.

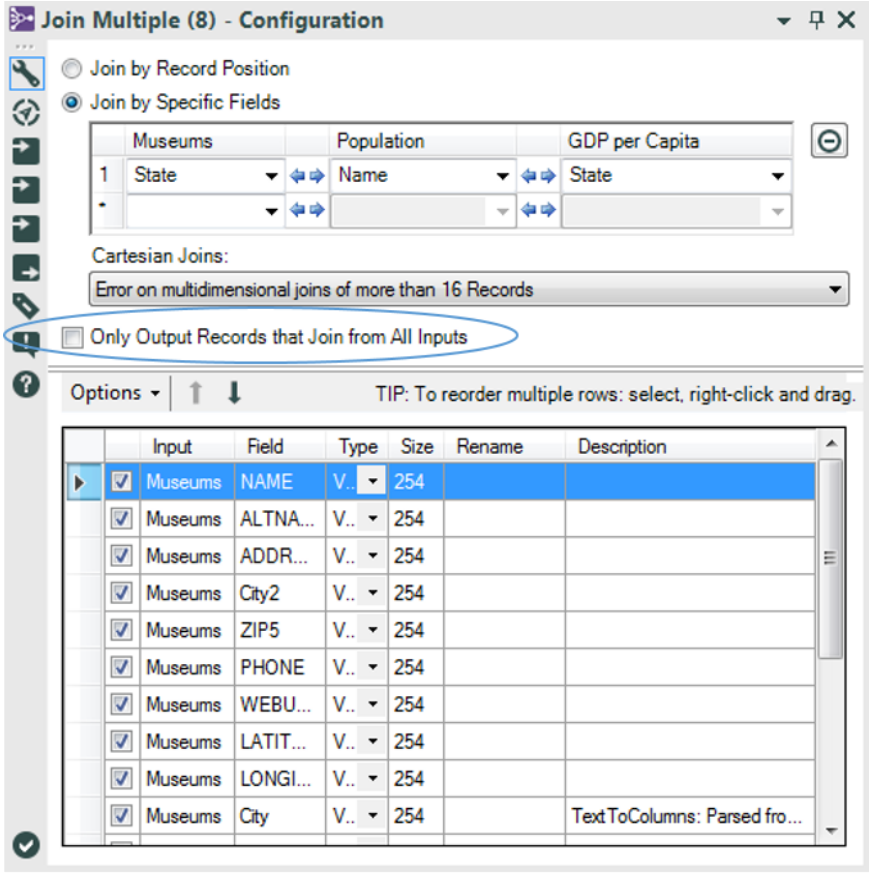


Figure 5-7 – Join Multiple Configuration

- It will have as many columns to join by, as there are input data streams.
- It has the ability to alert in case of too many replications.

- Instead of displaying Venn diagrams for the potential outputs, there is a checkbox (*Only Output Records that Join from All Inputs*) for deciding if we want the intersection of all of the datasets (the *J* output in the *Join* tool – Inner Join) or the full outer join (when unchecked).

5.5 Text To Columns

Figure 5-8 – Text to columns

The *Text To Columns* tool will break string fields into multiple string fields based on a delimiter.

Group	Input	Output
Parse	See below	See below

Note: This tool is useful in breaking down strings like email addresses or transaction elements so as to be able to look at the component pieces. It uses a single character to split the fields.

For example, if we type @| into the *Delimiters* text box, *Text To Columns* will split fields at @ and again at |, and not when it encounters @| together.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

The *Input* data stream should have at least one string field.

The *Output* will have the original data with either additional columns or rows generated by splitting the original field.

Properties Window:

The *Text To Columns Configuration* window has the following five core components.

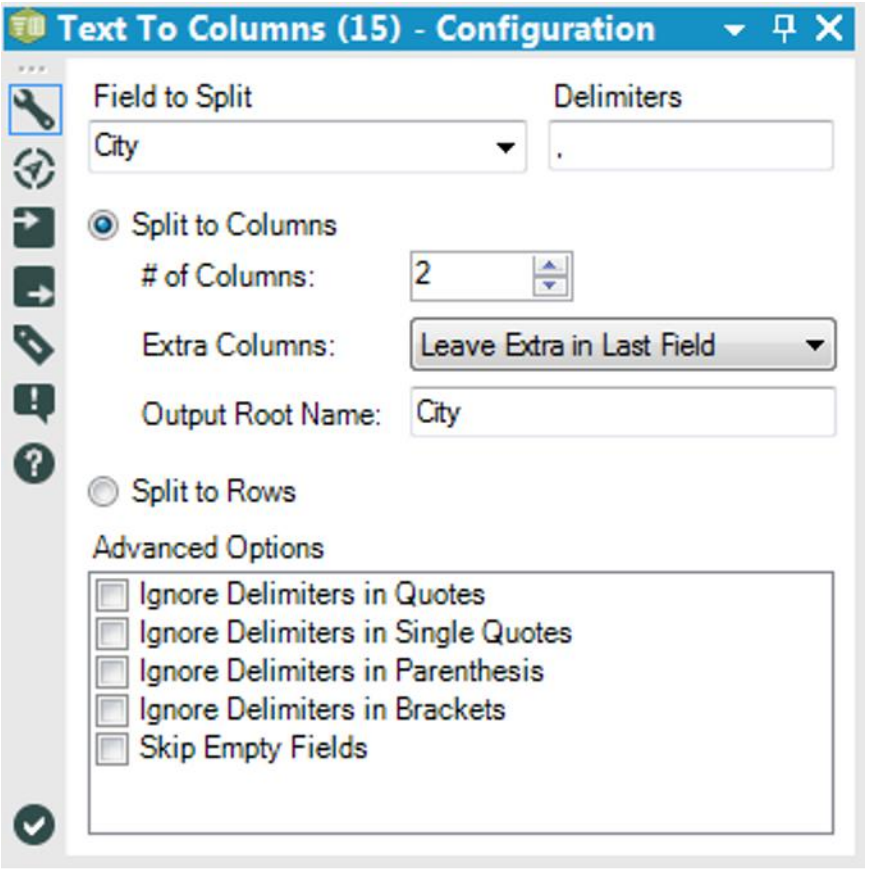
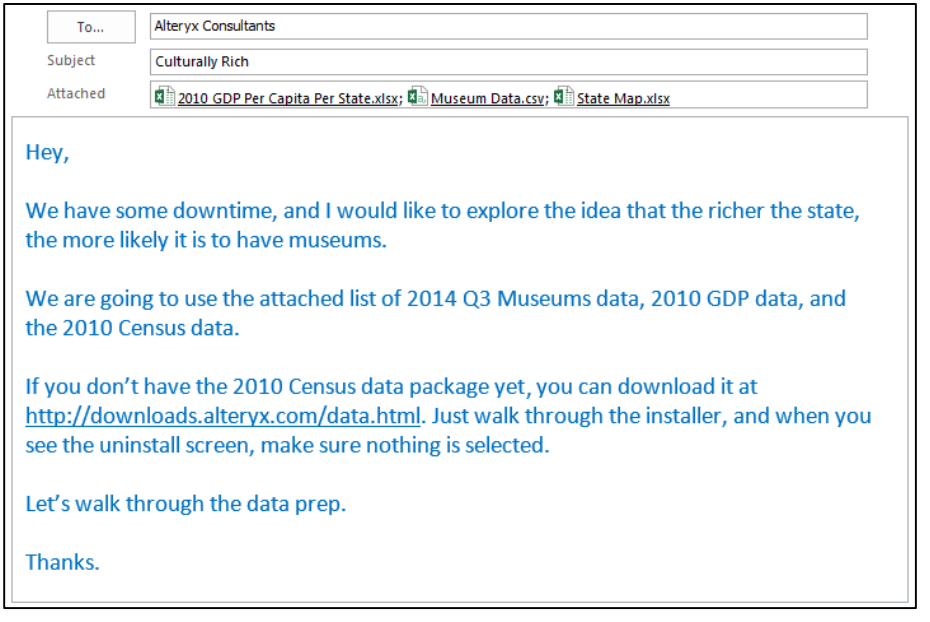


Figure 5-9 – Text to columns configuration

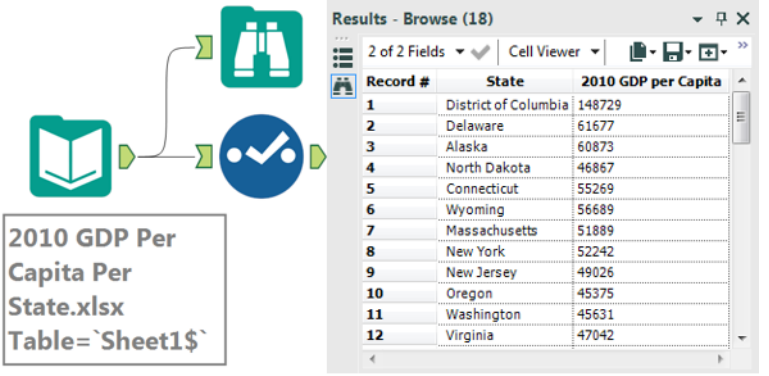
- *Field to Split* allows us to select the field to be split from the incoming data stream.
- *Delimiters* are the list of special characters or separators used to split the field by.

- *Split to Columns* allows creation of new columns for each split and has the following three secondary questions:
    - # *of Columns* allows us to select the number of new columns to be created.
    - *Extra Columns* is a drop-down menu that allows us to decide what happens to that data that would be split into more than the number of columns selected.
    - *Output Root Name* is the beginning of the name of the columns created by the split.
- *Split to Rows* changes the behavior such that new rows are created instead of additional columns for every split that is found.
- *Advanced Options* allow us to decide if there are more granular ways to control the splits.

5.6 Culturally Rich



The files required for this chapter have been placed in clearly marked sub-folders in *Chapter 5 – Cultural Musings* > *Culturally Rich*. We will start by bringing in each of the four data sources to see what we have. Let us open all three files and then connect to the *Census* data.



Record #	State	2010 GDP per Capita
1	District of Columbia	148729
2	Delaware	61677
3	Alaska	60873
4	North Dakota	46867
5	Connecticut	55269
6	Wyoming	56689
7	Massachusetts	51889
8	New York	52242
9	New Jersey	49026
10	Oregon	45375
11	Washington	45631
12	Virginia	47042

Figure 5-10 – Culturally Rich – GDP data

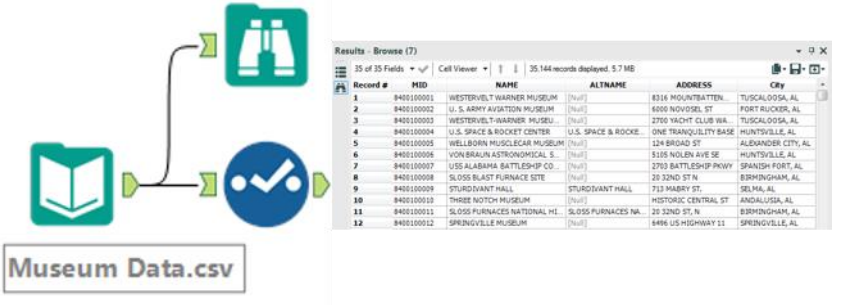


Figure 5-11 – Culturally Rich – Museum Data

As can be seen in these two files, the names of the states appear in different formats, with the *GDP per Capita* file having the entire state name spelled out, and the *Museum* data having only the two-letter state abbreviation within the city field.

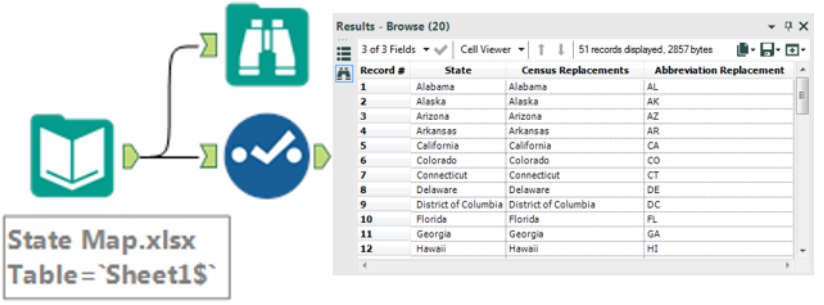


Figure 5-12 – Culturally Rich – State Map

However, the third file has three separate columns with either the complete state name or its abbreviation. This is because *State Map.xlsx* is a file created especially for the purpose of field mapping by linking the data sources from the *GDP*, *Museum* and *Census* data sets, which all have state identifiers in different formats.

The following are the observations from the data in the three files we connected to.

- The *GDP per Capita* data does not need any preparation before the join.
- The *State* associated with each museum needs to be parsed out of *City*, and then mapped to the *GDP per Capita* name.
- The *State Map.xlsx* file can be used for mapping all data sources together.

Let us now bring in the *Census* data in order to plan what needs to be done with that data stream.

We shall use an *Allocate Input* tool on the canvas, then choose the *US Census 2010 – Most Recent Vintage* dataset. Let us

select the *States* option under *Pick Geography,* and click the checkbox to select all states.

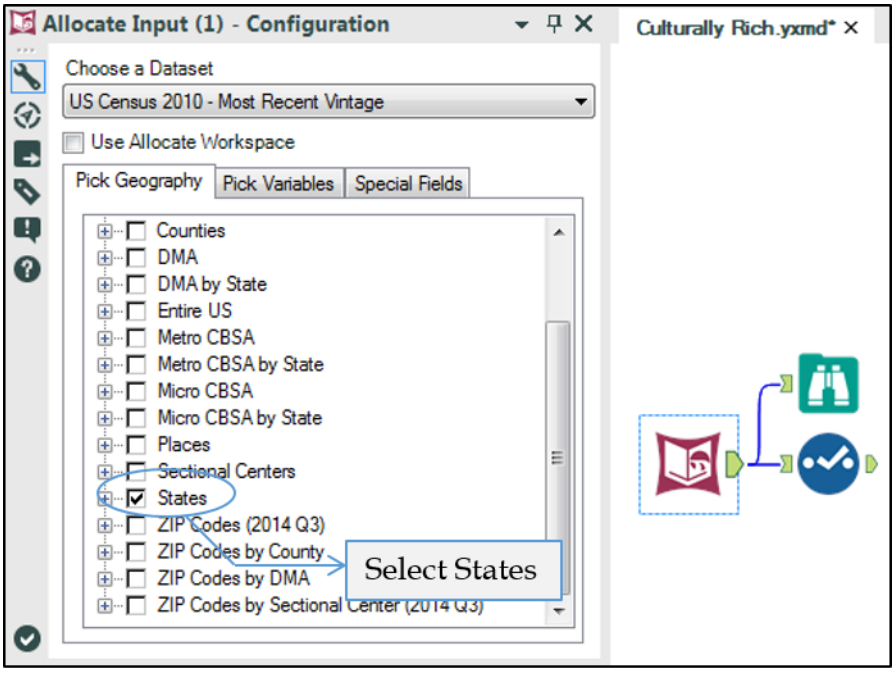


Figure 5-13 – Culturally Rich
Allocate Input configuration – Pick Geography

The purpose of bringing in the *Census* data is to compare the *GDP* to the number of museums in each state instead of the *GDP per Capita.* We can use the two data sources to generate the GDP value.

Now that the goal is identified, figuring out the variable needed becomes easy.

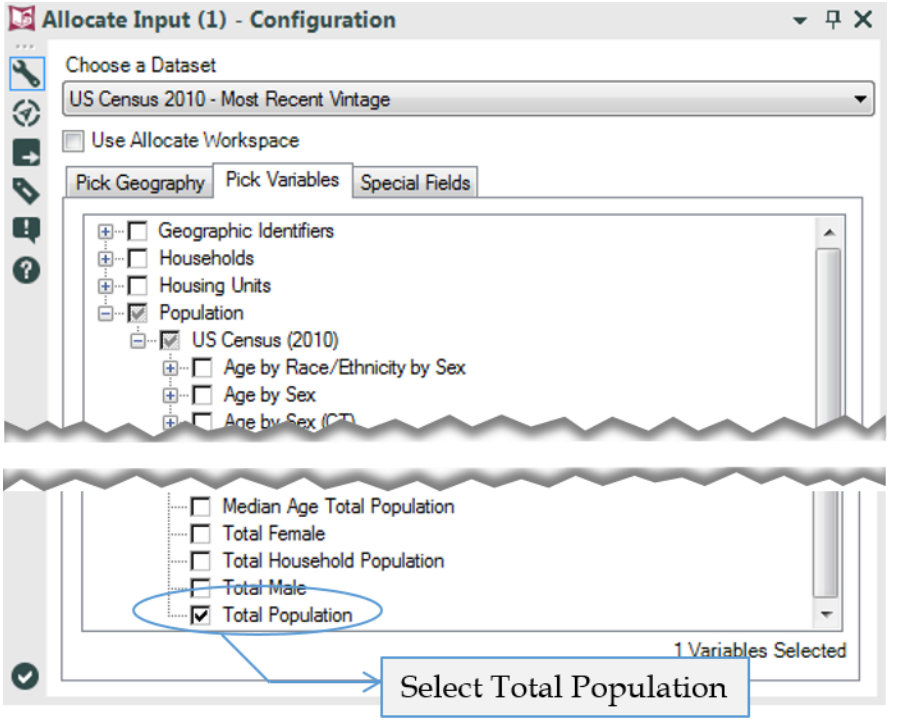


Figure 5-14 – Culturally Rich
Allocate Input configuration – Pick Variables

Under *Population* and *US Census (2010)*, the last option is *Total Population*. This signifies the total number of people that have lived in each chosen State (Geography).

Since we need only this information, click on the *Special Fields* (as shown in image below) and uncheck all options.

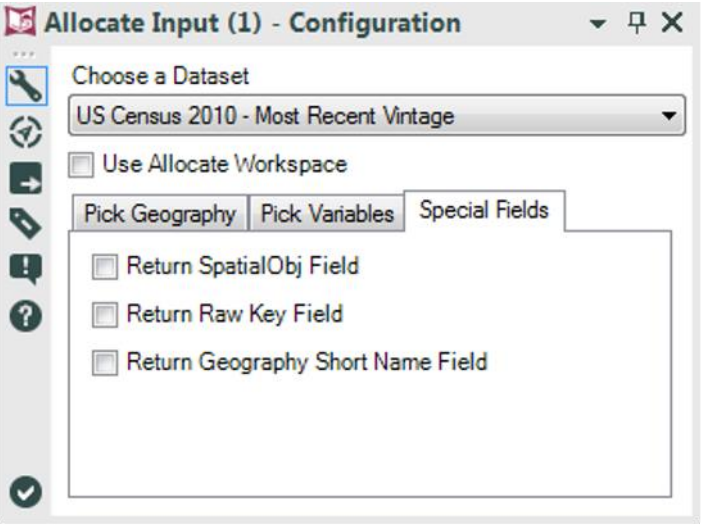
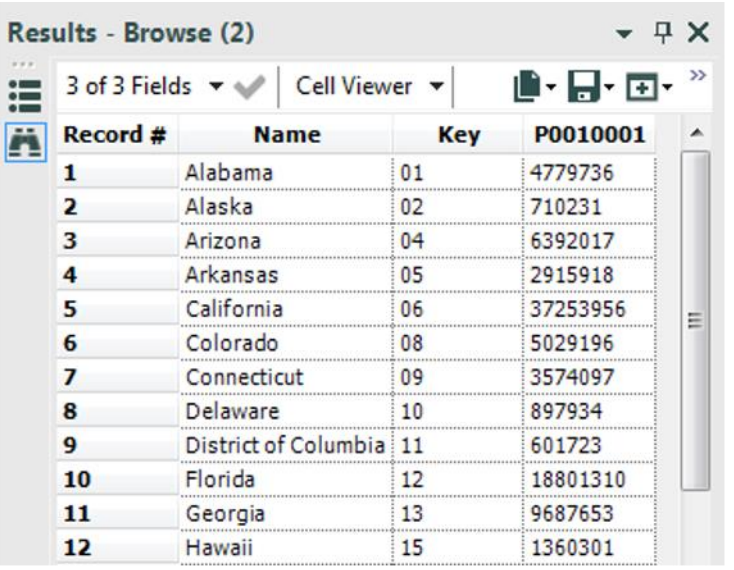


Figure 5-15 – Culturally Rich
Allocate Input configuration – Special Fields

Now that we have started with the final data stream, let us take a look at the data that comes out of the *Census* data connection.



Record #	Name	Key	P0010001
1	Alabama	01	4779736
2	Alaska	02	710231
3	Arizona	04	6392017
4	Arkansas	05	2915918
5	California	06	37253956
6	Colorado	08	5029196
7	Connecticut	09	3574097
8	Delaware	10	897934
9	District of Columbia	11	601723
10	Florida	12	18801310
11	Georgia	13	9687653
12	Hawaii	15	1360301

Figure 5-16 – Culturally Rich
Allocate Input Browse configuration

The following two points about this data stand out:

- Even though we have only selected one geography and one variable, we still have three fields. This is because the *Key* field uniquely identifies all geographies so that even if we have regions with the same name, they have unique identifiers. In this case, because they are States, we do not need the *Key*.
- There is a field called *P0010001*. This is because the data is stored with keyed column headers. We will simply rename the field to *2010 Population* in the *Select* tool, as shown in the following image.

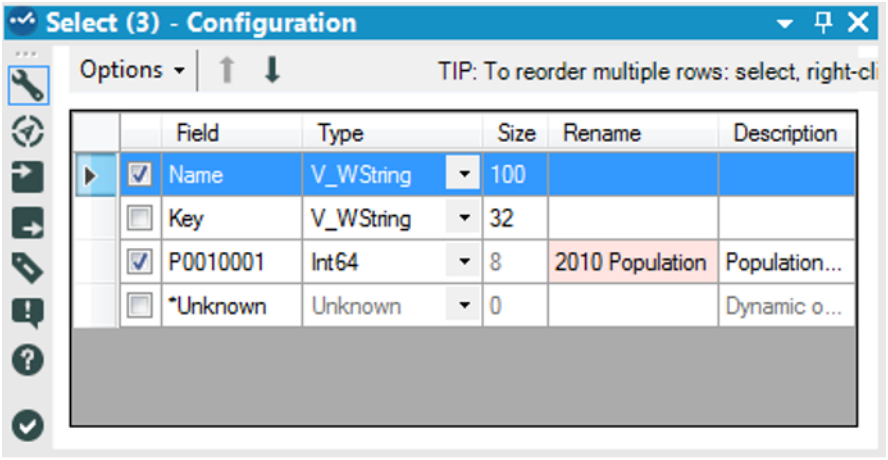


Figure 5-17 – Culturally Rich Census Population data Select configuration

Now, the remaining preparation is to map all of the state names from the different data sources back to the *GDP per Capita State* names before joining these data streams.

Let us start with mapping the *Census* data since it requires only one step before the join. In order to do this, we will use the *Find Replace* tool. Connect *Census* data stream to the *Find (F)* input and the mapping data stream to the *Replace (R)* input as shown in the following image.

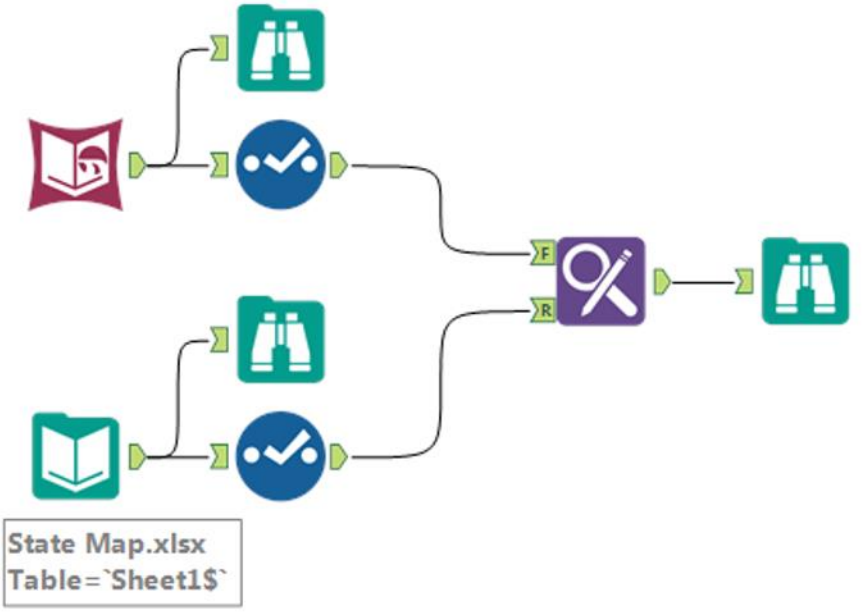


Figure 5-18 – Culturally Rich
Connecting census data to Find and State Map data to Replace

Now, let us take a look at the settings of *Find Replace* tool.

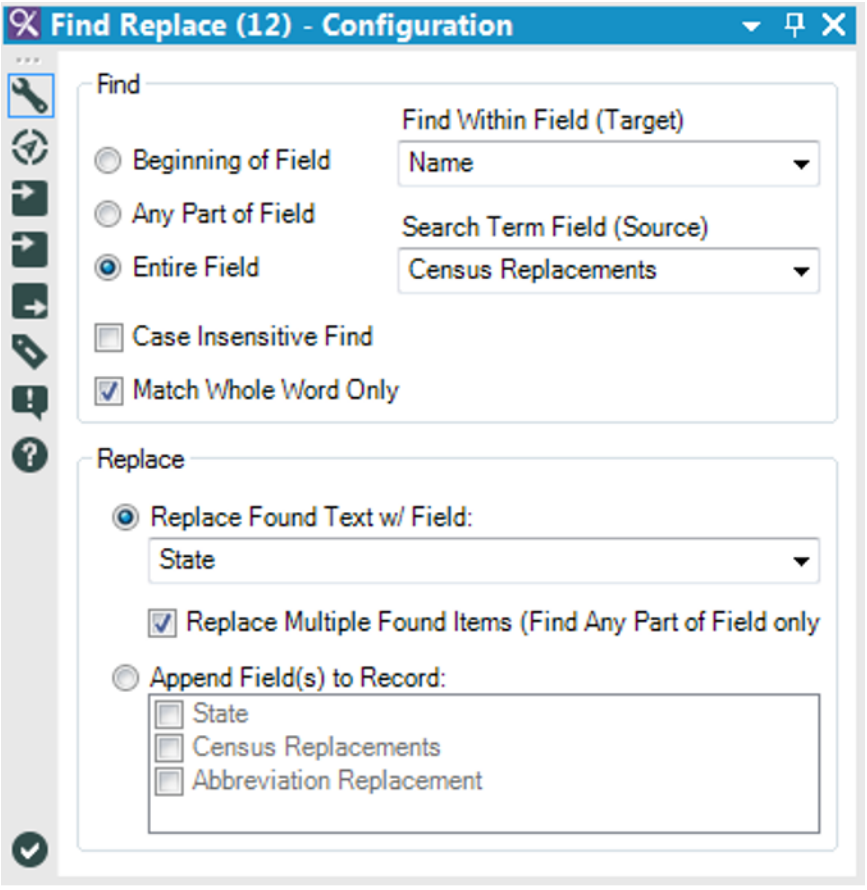


Figure 5-19 – Culturally Rich first Find Replace configuration

Since the map has been structured in such a way that the field *Census Replacements* has all of the *Census State* names in it, we will be looking for the *Entire Field* matches of the *Census Replacements* in the *Census Name* field. We also want to make sure that we replace the fields found with the *State* field (which has the *State* names from the *GDP* data). Since the *Census* data stream and the *GDP* data stream share a mapping field, let us get the *Museums* data ready.

We will first remove all of the fields except *Name* and *City* in the *Select* tool following the *Input Data* tool. Next, we will split the data apart. To do this, use the *Text to Columns* tool and add it to the end of the *Museums* data stream with the following settings.

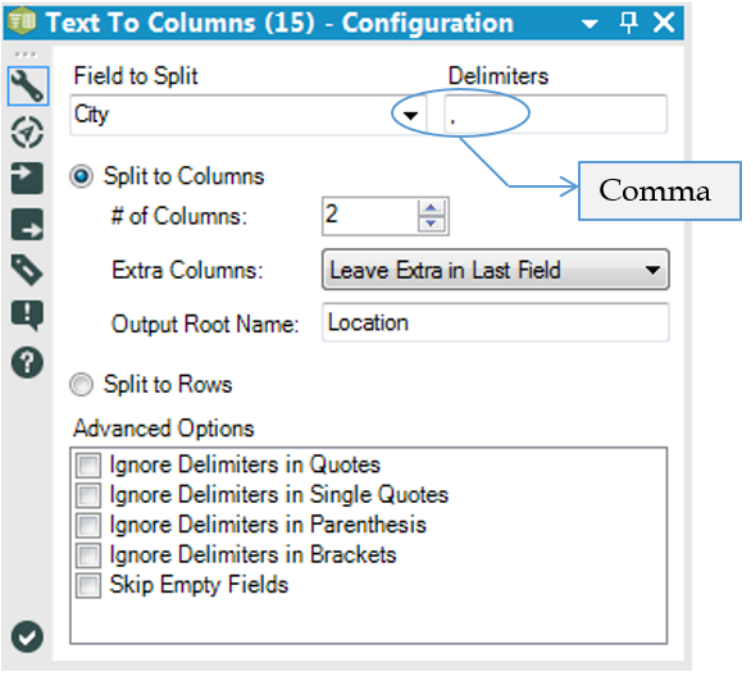
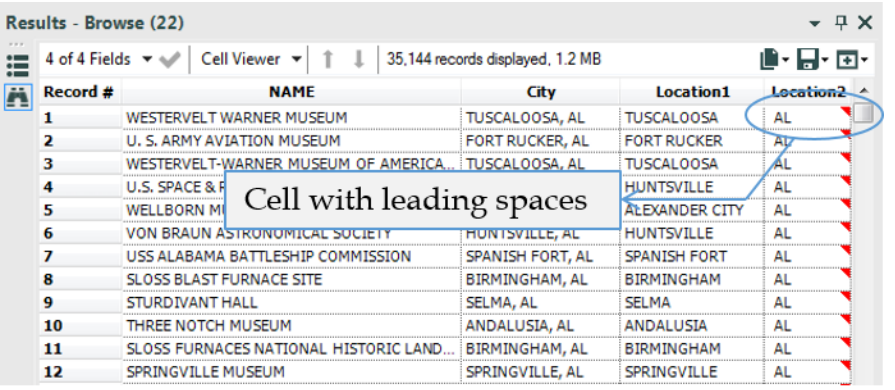


Figure 5-20 – Culturally Rich
Text to columns configuration to split at a separator

Then, let us look at the updated data after it has been modified with the *Text to Columns* field.



Record #	NAME	City	Location1	Location2
1	WESTERVELT WARNER MUSEUM	TUSCALOOSA, AL	TUSCALOOSA	AL
2	U. S. ARMY AVIATION MUSEUM	FORT RUCKER, AL	FORT RUCKER	AL
3	WESTERVELT-WARNER MUSEUM OF AMERICA...	TUSCALOOSA, AL	TUSCALOOSA	AL
4	U.S. SPACE & R		HUNTSVILLE	AL
5	WELLBORN MU		ALEXANDER CITY	AL
6	VON BRAUN ASTRONOMICAL SOCIETY	HUNTSVILLE, AL	HUNTSVILLE	AL
7	USS ALABAMA BATTLESHIP COMMISSION	SPANISH FORT, AL	SPANISH FORT	AL
8	SLOSS BLAST FURNACE SITE	BIRMINGHAM, AL	BIRMINGHAM	AL
9	STURDIVANT HALL	SELMA, AL	SELMA	AL
10	THREE NOTCH MUSEUM	ANDALUSIA, AL	ANDALUSIA	AL
11	SLOSS FURNACES NATIONAL HISTORIC LAND...	BIRMINGHAM, AL	BIRMINGHAM	AL
12	SPRINGVILLE MUSEUM	SPRINGVILLE, AL	SPRINGVILLE	AL

Figure 5-21 – Culturally Rich Modified Museum data

When we look at the *Browse* tool, we can see that *Location2* has a *red triangle* at the top-right corner of the cell. It is an alert indicating that Alteryx recognizes a potential issue with the data. If we hover the mouse over the cell, we will see a little text-box pop-up that reads "This cell has leading spaces". This means the first character of the cell is a space (attributed to the split and a space after "," delimiter) and that we should address it. We will do this by using the *Formula* tool to create a new field, we will call *State*.

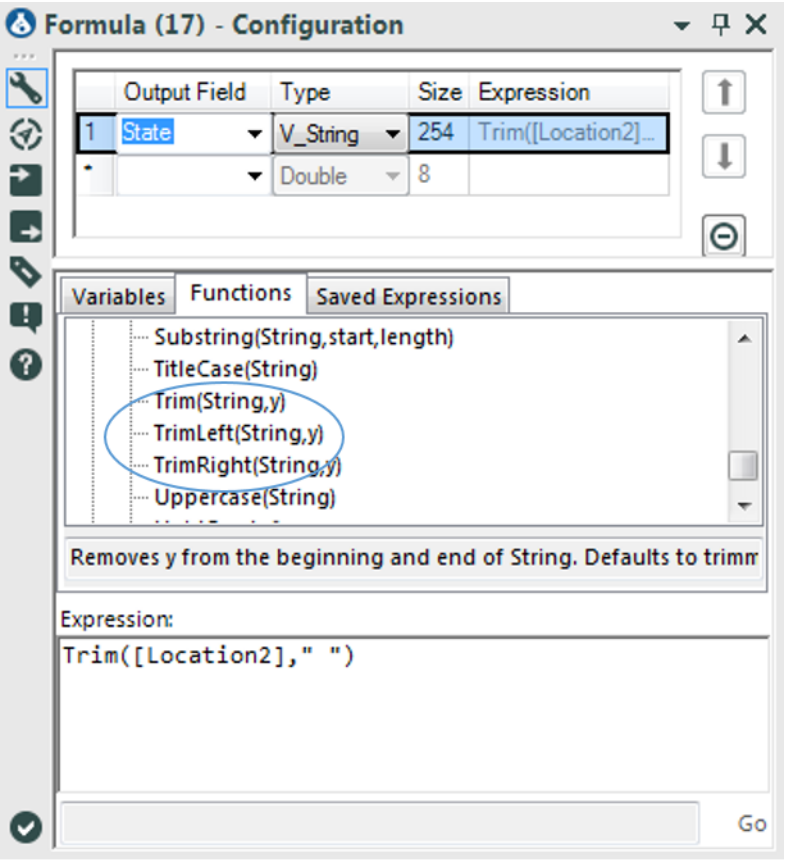


Figure 5-22 – Culturally Rich
Trimming Museum data using Formula

This time, we will be creating a string. If we look under the *String* functions, we can find three functions designed to remove one string from another. We will use the *Trim* function because we do not know if there are cells having trailing spaces which would cause issues with mapping.

Since we are removing spaces from the field, we could simply write the formula *Trim([Location2])* because the function removes white spaces by default. However, for the purpose of

clarity and readability, the best practice is to explicitly define all the variables. In this case, it would mean writing the formula as *Trim([Location2],” “)*.

Now that we are left with clean two-character *State* abbreviations, we can map them to the *GDP State* names. Once again, we will bring a *Find Replace* tool onto the canvas and connect the mapping data to the *Replace* input.

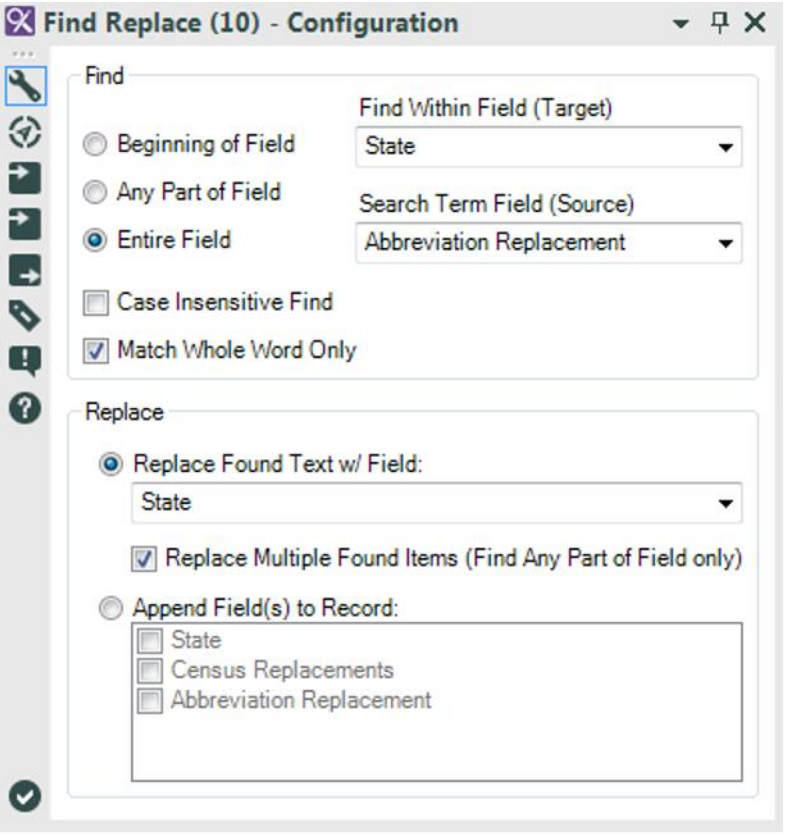


Figure 5-23 – Culturally Rich second Find Replace configuration

Now that we have prepared all the contributing data streams, this is what the data stream should look like.

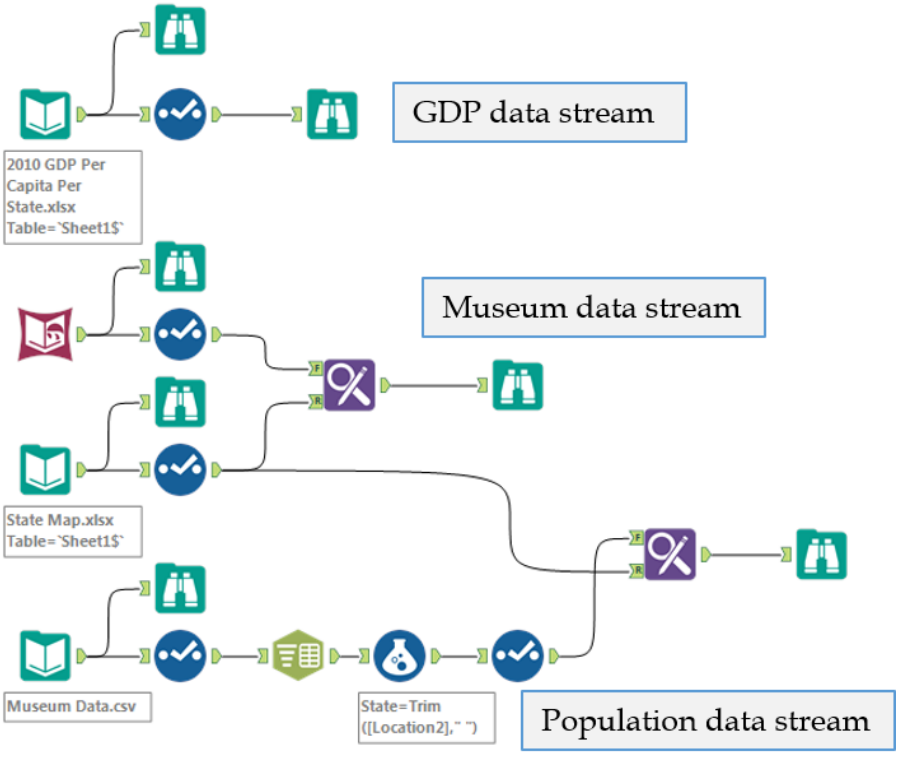


Figure 5-24 – Culturally Rich Consolidated Modified data streams

Notice that we have crossed the data streams. From a technical perspective, it does not matter if they are crossed. However, practically, the workflow becomes harder to read when the data streams are crossed frequently.

Best practice is to create workflows with as few crossovers as possible. In this case, we leave it as-is because there will invariably be at least one cross over in this module.

We could use a couple of *Join* tools now, but since all data streams share the same key (*State* name), we can use a *Join Multiple* tool to join all three data streams at once. We will connect the *GDP Per Capita*, *Museum,* and *Population* data streams to a *Join Multiple* input in the same order. We can then rename the connectors similarly and configure it as shown in the image.

One thing to note is that Alteryx renames conflicting fields with the data stream name and an "_". For example, since there are two fields that are named *State*, the second occurrence is renamed with the data stream as an identifier: *Museum_State*.

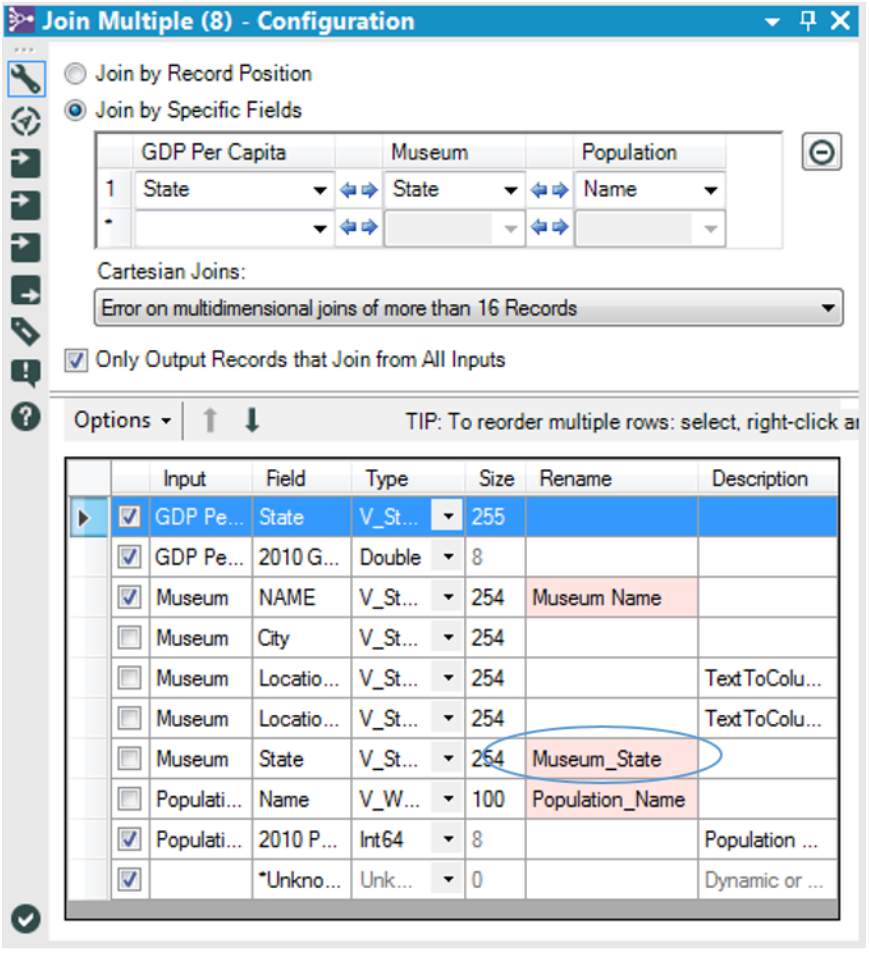


Figure 5-25 – Culturally Rich Join Multiple configuration

This is what the data stream would look like.

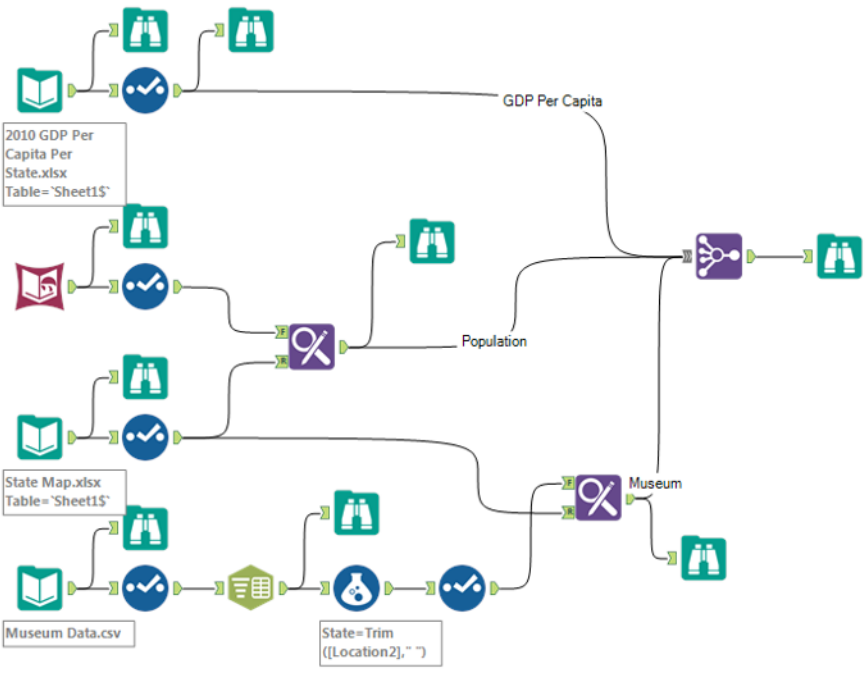


Figure 5-26 – Culturally Rich Unified data stream

Once we perform the following steps, our data preparation will be complete.

- Add a field called *Total GDP* (product of GDP Per Capita and Population).

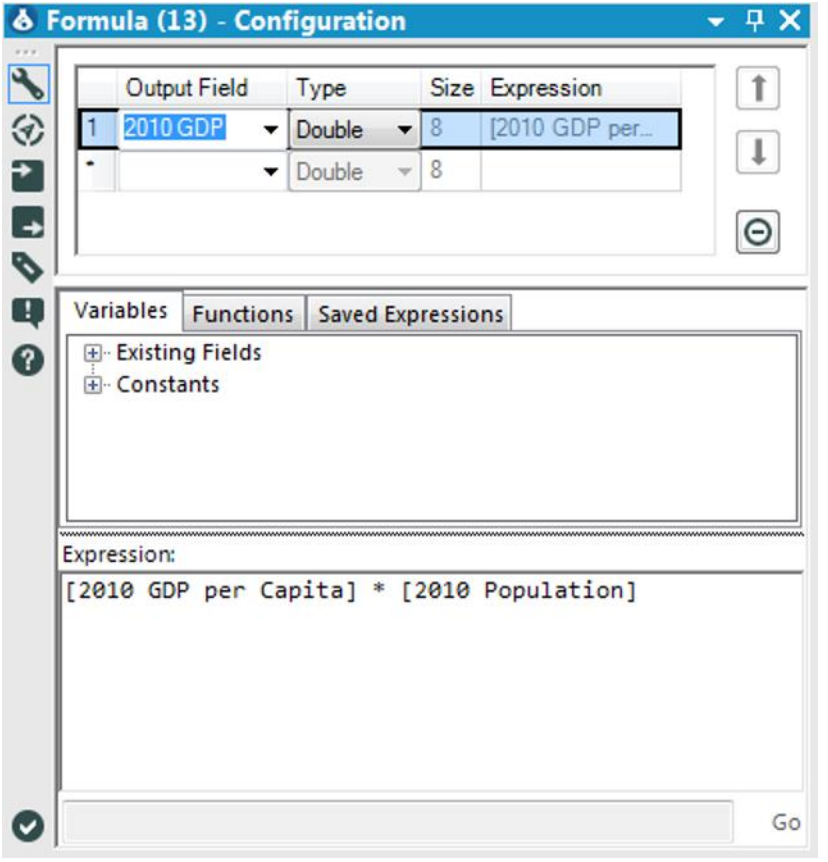


Figure 5-27 – Culturally Rich
Total GDP calculation

- Export the data to a file called *Culturally Rich*.

The *Culturally Rich* data stream should look like this when it is complete.

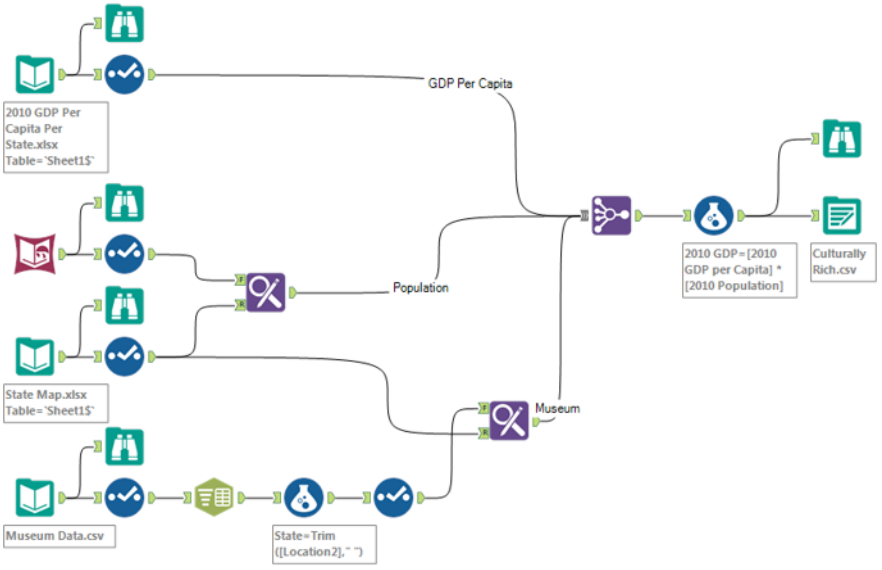
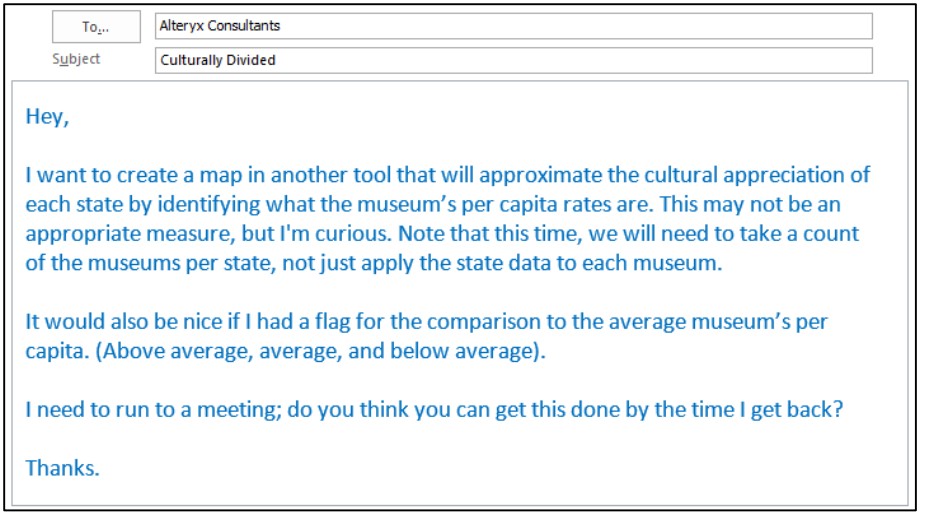


Figure 5-28 – Culturally Rich Complete data stream

5.7 Culturally Divided


To... Alteryx Consultants
Subject Culturally Divided

Hey,

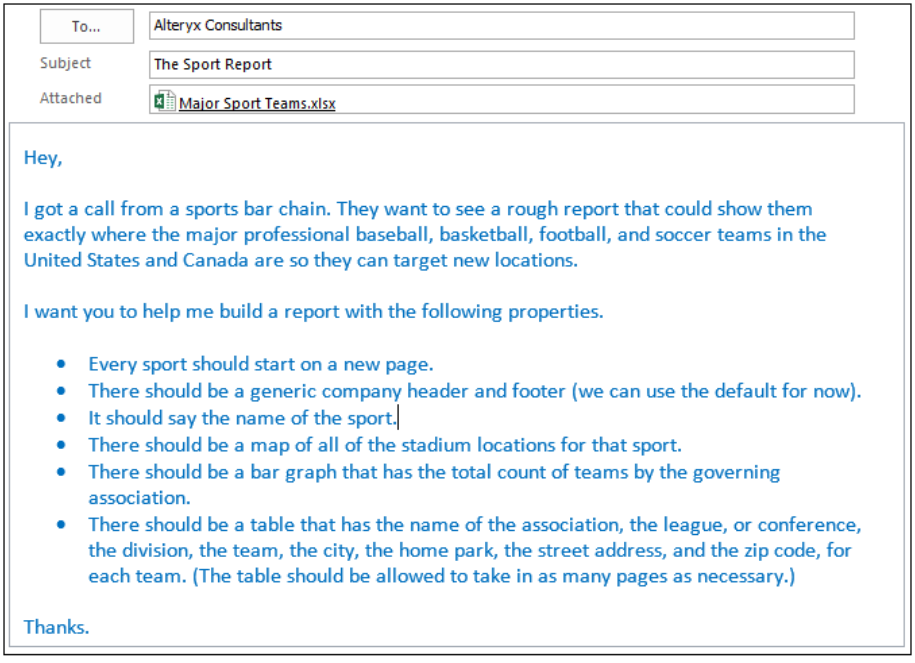
I want to create a map in another tool that will approximate the cultural appreciation of each state by identifying what the museum's per capita rates are. This may not be an appropriate measure, but I'm curious. Note that this time, we will need to take a count of the museums per state, not just apply the state data to each museum.

It would also be nice if I had a flag for the comparison to the average museum's per capita. (Above average, average, and below average).

I need to run to a meeting; do you think you can get this done by the time I get back?

Thanks.

Chapter 6
The Sport Report


To...
Alteryx Consultants
Subject
The Sport Report
Attached
Major Sport Teams.xlsx
Hey,
I got a call from a sports bar chain. They want to see a rough report that could show them exactly where the major professional baseball, basketball, football, and soccer teams in the United States and Canada are so they can target new locations.
I want you to help me build a report with the following properties.
Every sport should start on a new page.
There should be a generic company header and footer (we can use the default for now).
It should say the name of the sport.
There should be a map of all of the stadium locations for that sport.
There should be a bar graph that has the total count of teams by the governing association.
There should be a table that has the name of the association, the league, or conference, the division, the team, the city, the home park, the street address, and the zip code, for each team. (The table should be allowed to take in as many pages as necessary.)
Thanks.

6.1 Tools & Concepts

In this chapter, we will be covering the following Tools and Concepts:

Tools	Concepts
Charting	Report Building
Create Points	Spatial Objects
Layout	Maps
Render	
Report Footer	
Report Map	
Report Header	
Table	

To the reader,

We will be covering the basics of report building in Alteryx. However, since building reports in Alteryx is an advanced technique, we will not be exploring the Report tools as completely as the tools in other sections.

USEReady

6.2 Charting

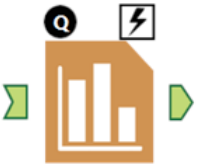


Figure 6-1 - Charting

The *Charting* tool creates a chart object for reporting.

Group	Input	Output
Reporting	See below	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Input: A data stream with at least one numeric field.

Output: A data stream with a record for every element in fields grouped by, a column for each grouping field, and a field called *Chart* with a *Chart object* in each record.

This tool creates a reporting object.

Properties Window:

The *Charting Configuration* window has four components: *Chart Type*, *Select Field(s) To Chart*, *Preview*, and *Setup Chart Appearance*.

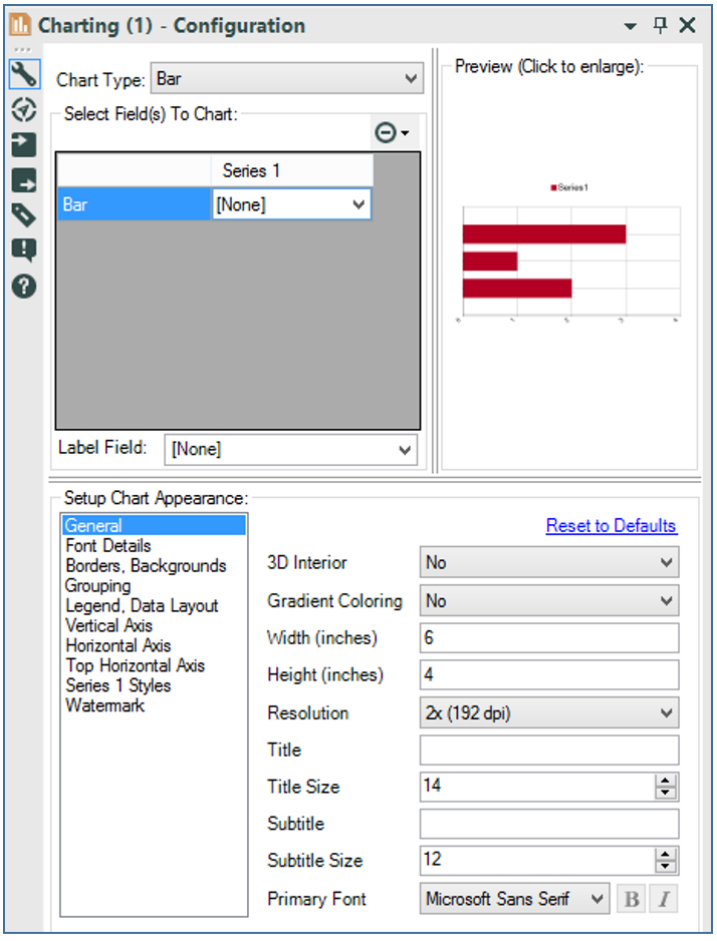


Figure 6-2 – Charting Configuration

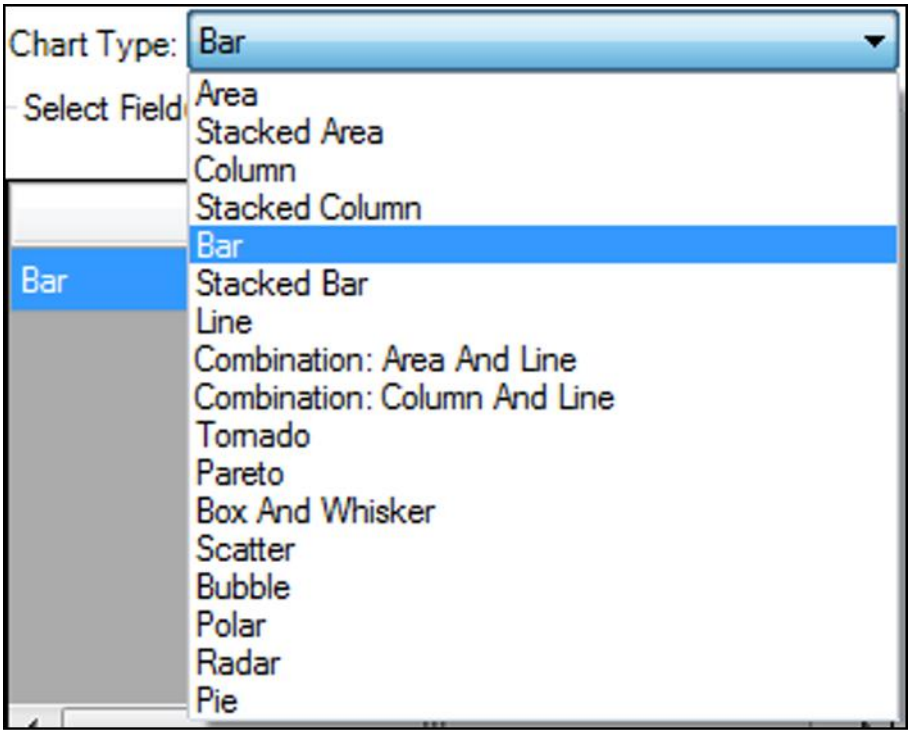


Figure 6-3 - Chart Types

Chart Type allows us to select from a list of charts.

For each of these charts, the *Setup Chart Appearance* will reflect the appropriate settings. We will be walking through the basic settings when we use a bar chart.

Select Fields to Chart is where we can select each of the individual items that represent a series in the data. When working with a *Bar* chart, this means we can take additional columns to create additional bars for each field. At the bottom of this window, there is an ability to select the *Label Field*, which allows us to change the label on the bars.

Preview shows us a sampled version of what our resulting chart will look like. It only shows the formatting and does not refer to the real data stream. If we click on the image, it opens in a pop-up window so we can get a clearer look at what we are designing.

The *Setup Chart Appearance* has two main components: the list on the left side, which allows us to select which aspect of the chart we want to modify, and the settings on the right side that we can modify. Most of the settings we will see in the following options are fairly self-explanatory, and some may have secondary questions that appear when we select them. We will be covering all of these options at a conceptual level.

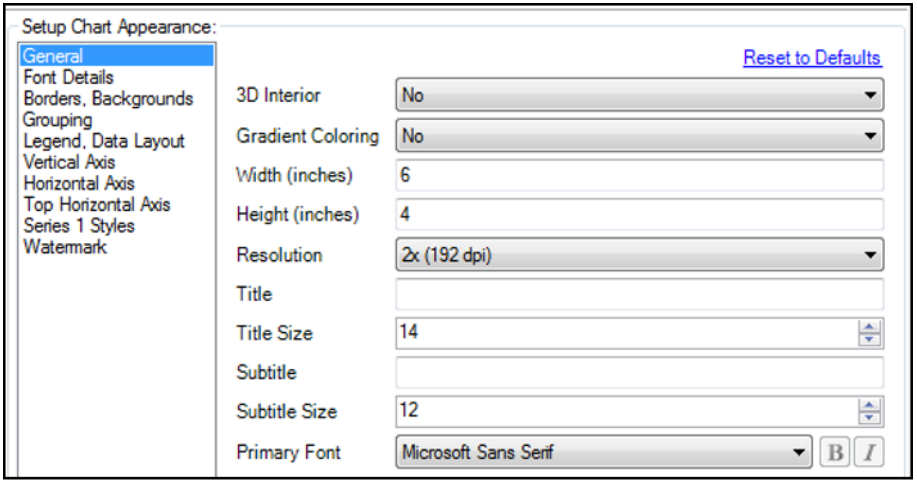


Figure 6-4 - Chart Appearance - General

Under *General* setting, the first two settings change the formatting of the bars. *3D interior* makes the bar appear 3D while *Gradient Coloring* makes the bars look rounded. With *Width* and *Height*, we can set the dimensions of the graph image. *Resolution* allows us to set the quality of the image. *Title* and *Subtitle* allow

us to type a title onto the graph, while *Title Size* sets the size of the font. *Primary Font* allows us to set the default font used for the graph.

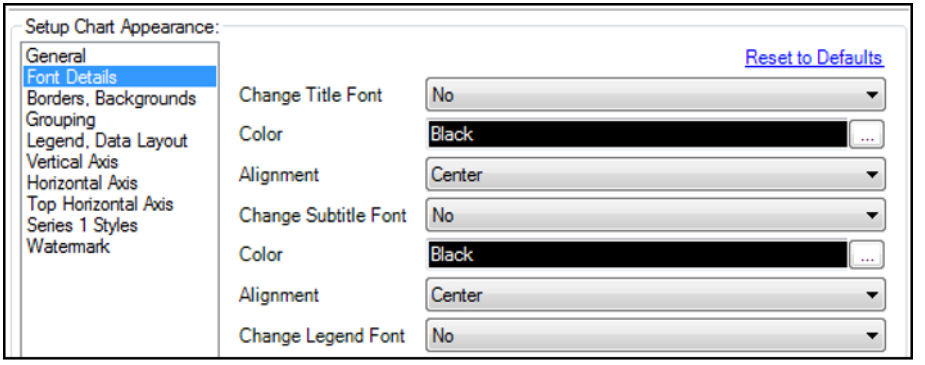


Figure 6-5 - Chart Appearance - Font Details

Font Details allow us to change the font settings for the Title, Subtitle, and Legend.

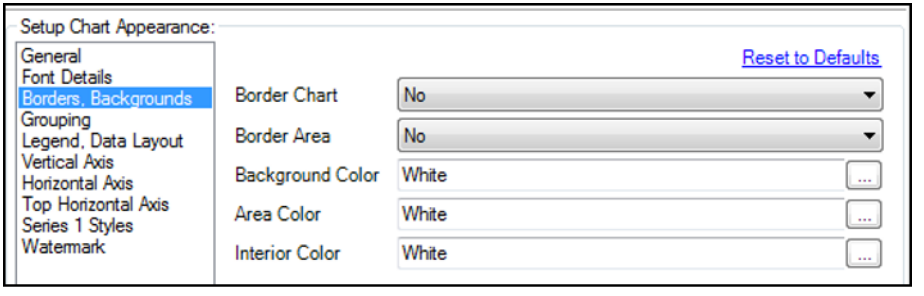


Figure 6-6 - Chart Appearance - Borders, Backgrounds

Borders, Backgrounds allow us to change the color and size of borders throughout the graph (excluding the bars).

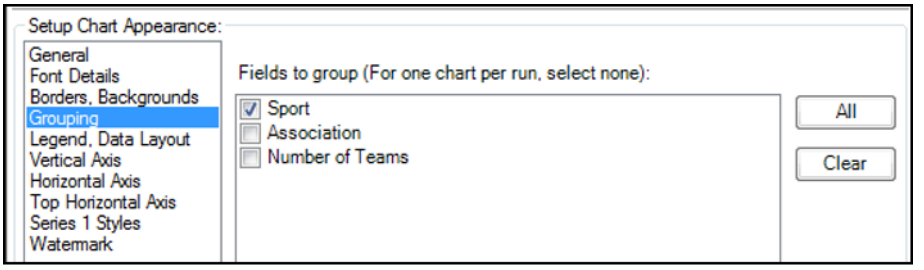


Figure 6-7 - Chart Appearance - Grouping

With *Grouping*, for each unique element in the field(s) selected, a different chart will be created (and a different record will be created in the output).

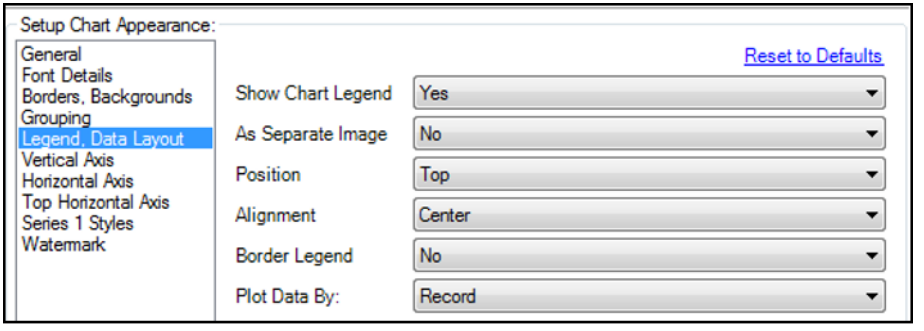


Figure 6-8 - Chart Appearance - Legend, Data Layout

In *Legend, Data Layout*, the first five tools allow us to customize the legend. *Plot Data By* changes the way the data is colored within the graph. The default is to color by series; h. However, if we change this setting from *Record* to *Field*, we will be able to change the color of the data by a specific field.

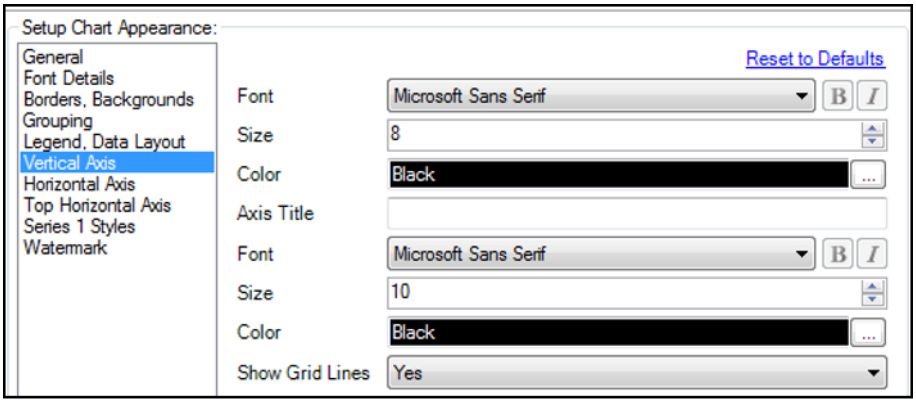


Figure 6-9 - Chart Appearance - Vertical Axis

Vertical Axis allows us to customize the font of the *Axis Title* and labels, as well as turn the gridlines on or off.

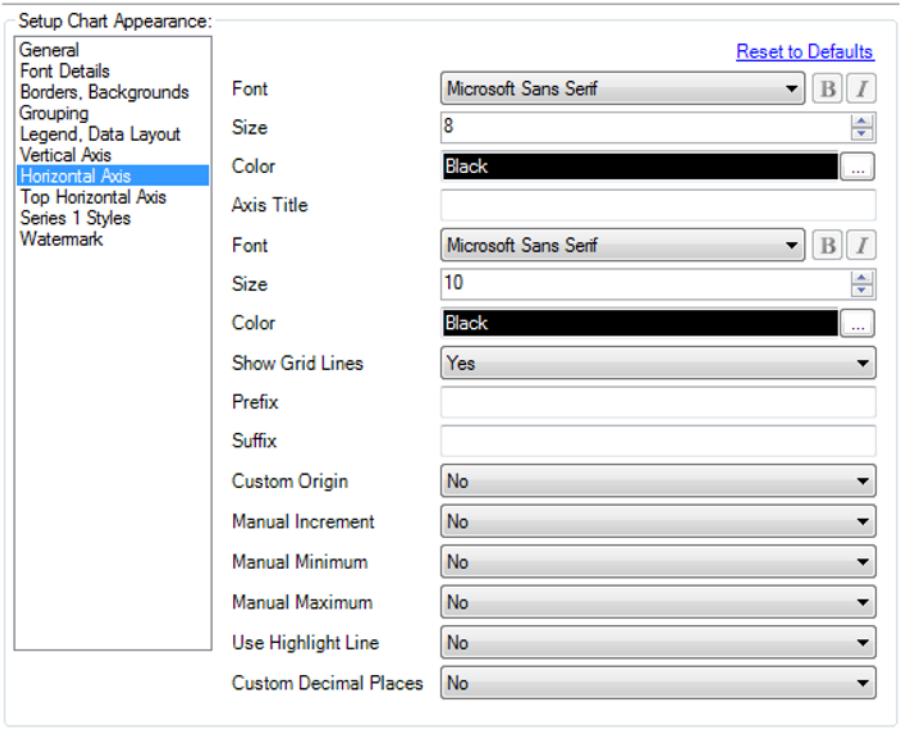


Figure 6-10 - Chart Appearance - Horizontal Axis

Horizontal Axis allows us to customize the horizontal axis. The top half is the same as the *Vertical Axis* settings. However, this being a numeric axis, it gives us the ability to customize the range, set a reference (highlight) line, and change the number format.

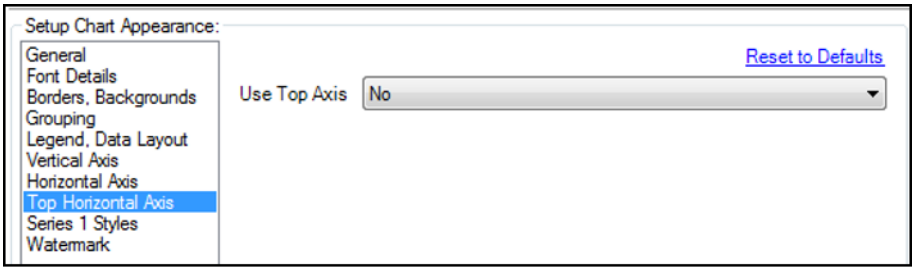


Figure 6-11 - Chart Appearance - Top Horizontal Axis

The *Top Horizontal Axis* allows us to set, define, and customize a secondary axis for the chart.

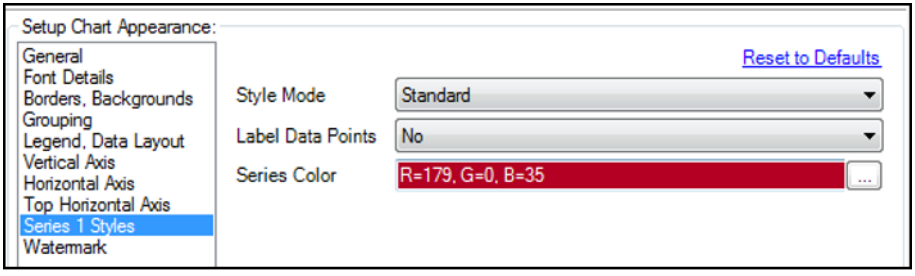


Figure 6-12 - Chart Appearance - Series X Styles

The *Series X Styles* allows us to set the style for the series X so we can differentiate them.

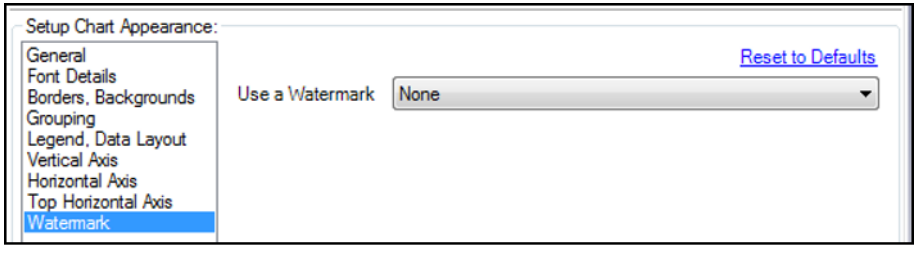


Figure 6-13 - Chart Appearance - Watermark

Watermark allows us to set a watermark on the graph.

6.3 Create Points

Figure 6-14 – Create Points

The *Create Points* tool converts decimal latitudes and longitude columns into a new field called *Centroid* with a point object for each record.

Group	Input	Output
Spatial	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with latitude and longitude columns.

Output: The incoming data stream with an additional field containing point objects.

This tool creates a spatial object field.

Properties Window:

The *Create Points Configuration* window has two core components.

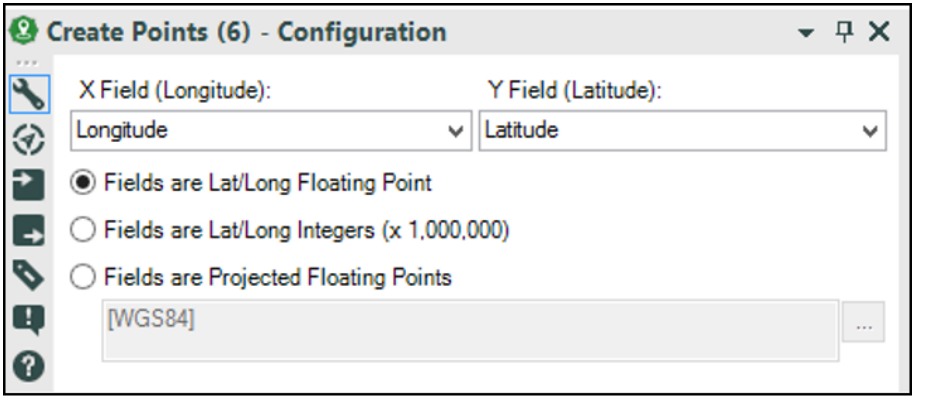


Figure 6-15 – Create Points Configuration

The *X Field (Longitude)* and *Y Field (Latitude)* are drop-downs that allow us to select the longitude and latitude fields from the input data stream.

The single value list following the field selection allows us to define how the latitude and longitude are defined.

- *Fields are Lat/Long Floating Point* means the fields are in a standard decimal format.
- *Fields are Lat/Long Integers (x 1,000,000)* means the fields have been standardized to an integer format.
- *Fields are Projected Floating Points* allows us to identify a specific formatting method to pinpoint spatial fields.

6.4 Layout

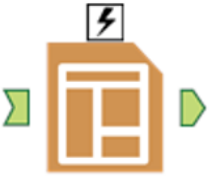


Figure 6-16 - Layout

The *Layout* tool creates a data stream with reporting objects and allows us to structure their layout.

Group	Input	Output
Reporting	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with at least one report object field.

Output: The original data stream with a single *Layout* object that combines other report objects. Depending on the settings, the original report object fields may be removed.

This tool creates a reporting object.

Properties Window:

The *Layout Configuration* window has the following five core components.

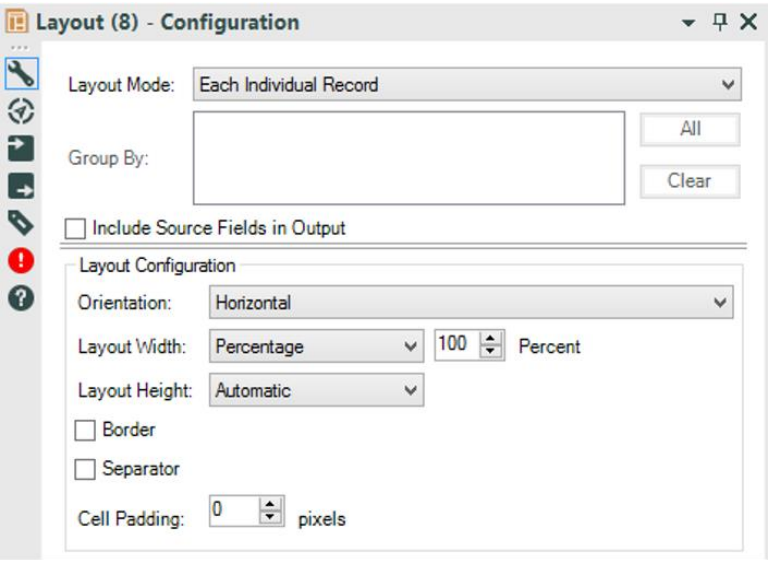


Figure 6-17 – Layout configuration

- *Layout Mode* allows us to define the method used to combine the different report objects.
- *Group By* (when using the *Each Group of Records* option from the *Layout Mode* drop-down) allows us to define what is to be combined by the group of records that share a field(s).
- *Include Source Fields in Output* allows us to decide whether or not the original report fields that have been combined should be in the outgoing data stream (assuming we have chosen the *Each Individual Record* from the *Layout Mode* drop-down).
- *Layout Configuration* allows us to set some parameters for the look and feel of the resulting *Layout* object(s).

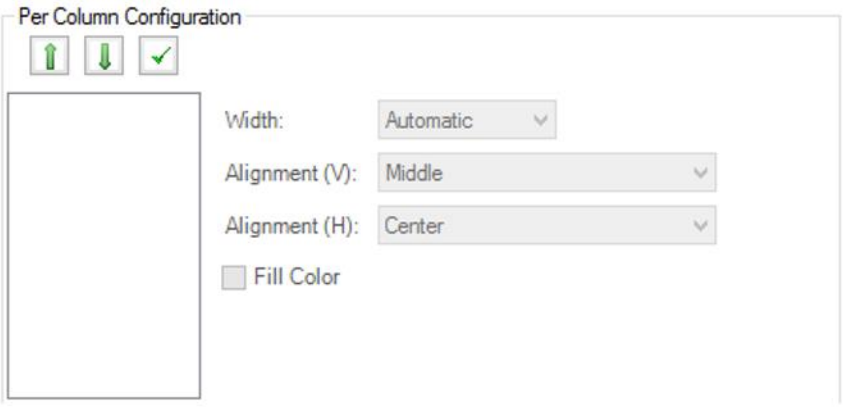


Figure 6-18 – Per Column Configuration

Per Column Configuration has two setups. *Each Individual Record* mode gives us the ability to select the reporting objects that we want to combine across a record, while the other two modes allow us to set the field(s) that we want to combine when creating a single record per group.

6.5 Render

Figure 6-19 - Render

The *Render* tool creates a file that allows us to see the structured reporting that we have designed.

Group	Input	Output
Reporting	See below	File

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Input: Any data stream with at least one report object.

Output is a file that has the structured report we've designed.

If we require more than one report object, we need to combine them with the *Layout* tool and use the *Layout* object.

Properties Window:

The *Render Configuration* window has three core components.

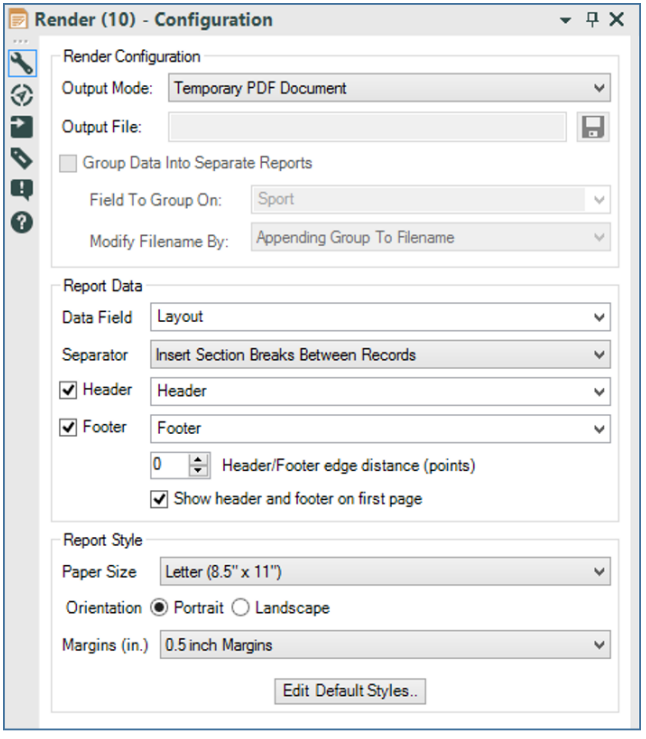


Figure 6-20 – Render Configuration

Render Configuration allows us to set the type and location of the report output.

Output Mode allows us to define a temporary file in one of a few different formats, as well as define a permanent file to write to.

Output file lets us define a path and filename for the file.

Group Data Into Separate Report, when selected, allows us to create multiple output reports based on a grouping field selected in *Field To Group On*.

Modify Filename By allows us to define how each of the files should be named in the output.

Under *Report Data*, *Data Field* allows us to select the field in the body of the report.

Separator allows us to set what is between each record.

Header and *Footer* allow us to define the header and footer fields respectively.

Header/Footer edge distance (points) is the number of points everything is from the edge of the paper.

Show header and footer on first page allows us to turn those aspects on or off for the first (often title) page of the report.

Report Style allows us to set the paper and layout properties of the final report.

Paper Size allows us to define page dimensions.

Orientation defines the way the report is laid out.

Margins (in.) define the margin width in inches.

Edit Default Styles allows us to change the default fonts and coloring.

6.6 Report Footer

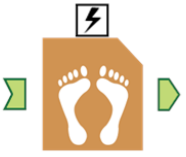


Figure 6-21 – Report Footer

The *Report Footer* tool allows us to create a footer to be used in reports.

Group	Input	Output
Reporting	Data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream, with the additional field *Footer*, which has a *Layout* object in each record.

This tool creates a reporting object and should be used after report elements have been joined together.

Properties Window:

The *Report Footer Configuration* window allows us to define the footer.

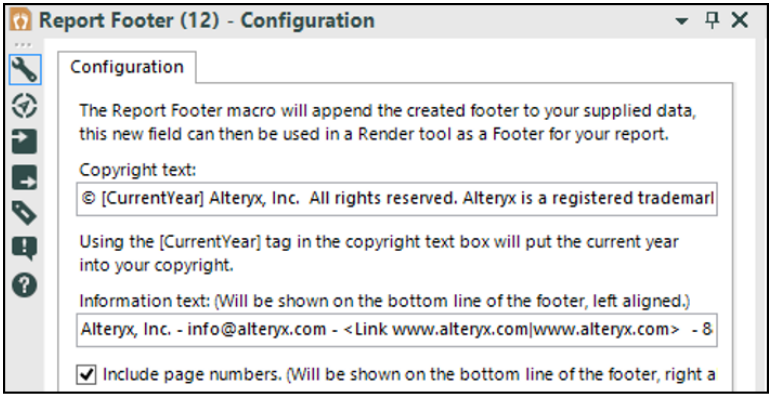


Figure 6-22 – Report footer configuration

Copyright text is intended for a copyright line on each page.

Information Text is a left-aligned text field that can include hyperlinks in the format <Link text to display | URL>.

Include page numbers allows to set the visibility of the page number at the bottom-right corner of the report.

6.7 Report Map

Figure 6-23 – Report Map

The *Report Map* tool creates a data stream with one or more maps.

Group	Input	Output
Reporting	See below	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Input: Any data stream with at least one spatial object field.

Output: A data stream with a record for every element in fields grouped by, a column for each grouping field, and a field called *Map* with a *Map object* in each record.

This tool creates a report object.

Properties Window:

The *Report Map Configuration* window has two main components. The *Preview* window gives us a look at the map to be generated.

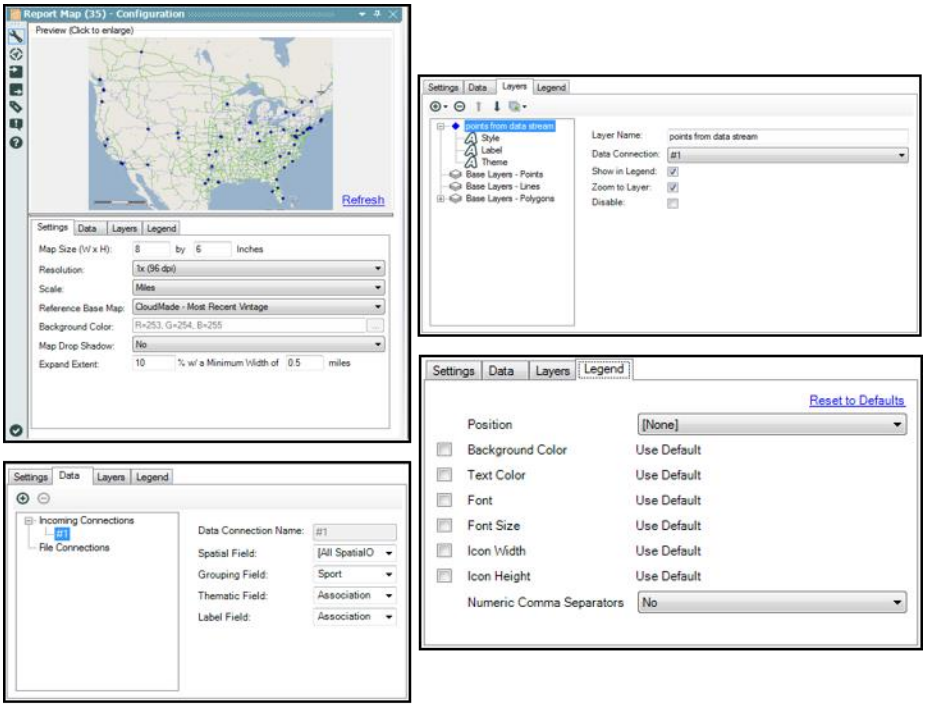


Figure 6-24 - Report Map Configuration

The bottom of the window is broken into four tabs:

- The *Settings* tab allows us to set high-level options including dimensions, resolution, scale units, the background map to be used, background color (if not using a map), whether or not the objects should have a drop shadow, and some zoom settings.
- The *Data* tab allows us to select data streams and add them as data layers. When an incoming connection or file connection is selected, we can define how the layers that use that connection will behave. *Data connection name* is how it will be identified. *Spatial Field* allows us to select one or all of the spatial object fields in the dataset to define.

Grouping Field will create a new map for each unique element in the selected field. *Thematic Field* allows us to color the elements based on a particular field. If a string field is to be used, we need to set the theme to *unique* and type in the values. *Label Field* allows us to select the field we will display the text from on the map.

- The *Layers* tab allows us to modify specific aspects of the map. We can add and remove layers to the map using the plus and minus symbols, as well as change the floating order of the objects. By selecting different objects in the tree, we can modify various settings. Some of these settings will be covered in the following exercise.
- The *Legend* tab allows us to set a number of options related to including a legend with the map. By default, the legend is off and accepts the default values.

6.8 Report Header

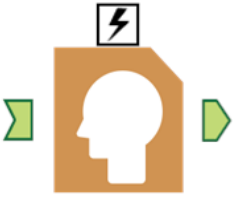


Figure 6-25 - Header

The *Report Header* tool allows us to create a header to be used in reports.

Group	Input	Output
Reporting	Any data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with the additional field *Header*, which has a *Layout* object in each record.

This tool creates a reporting object and should be used after report elements have been joined together.

Properties Window:

The *Report Header Configuration* window allows us to design a header for the report.

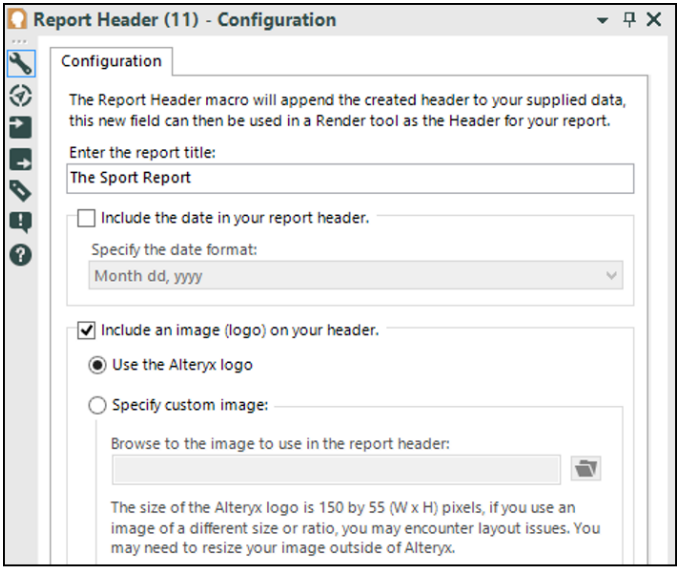


Figure 6-26 – Report header configuration

We can type in a title, decide if we want a date and set its format, and also decide if we want an image logo in the header.

6.9 Table

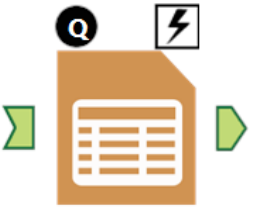


Figure 6-27 - Table

The *Table* tool creates a data stream with one or more tables.

Group	Input	Output
Reporting	Any data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: A data stream with a record for every element in the fields grouped by, a column for each grouping field, and a field called *Table* with a *Table object* in each record.

This tool creates a report object, which can be used with other report objects in the table tool.

Properties Window:

The *Table Configuration* window allows us to create a report table.

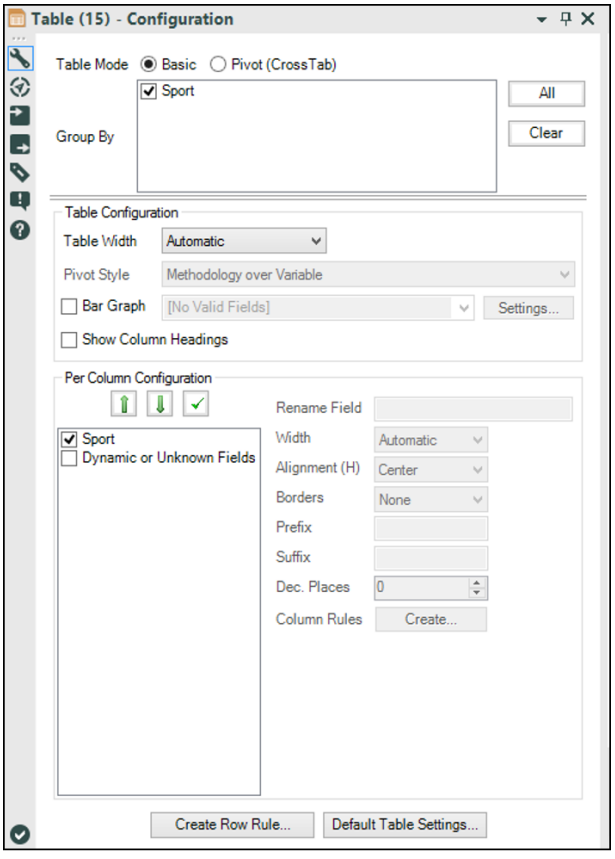


Figure 6-28 – Table configuration

Table Mode has the following two options:

- *Basic* will create a table from the rows and columns we have.
- *Pivot (CrossTab)* allows us to use fields we have created using a *CrossTab* tool in order to create a series of more flexible tables.

Group By will create a different table for each of the unique elements in the grouping fields.

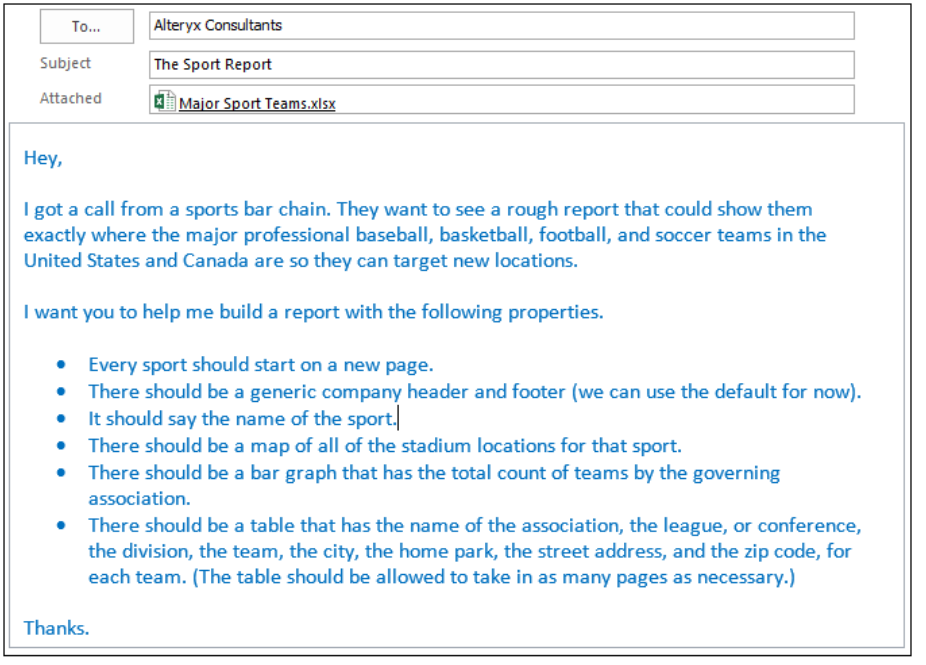
Table Configuration allows us to modify the appearance of the table. *Table Width* allows us to set a standard width or let it vary with the available space. *Pivot Style* allows us to display elements of the pivoted data from a *Crosstab* tool in different ways. *Bar Graph* allows us to create a spark bar for each element in the table. *Show Column Headings* adds a label to the top of each of the columns.

Per Column Configuration allows us to select columns to be display in the table and to set the format for each column. *Column Rules* allows us to create a conditional formatting for the column.

Create Row Rule allows us to create a record level formula.

Default Table Settings allows us to change the default formatting of the table.

6.10 Major Sports Teams



When building reports in Alteryx, we need to remember the following key points:

- Most report objects should be created in individual data streams and then brought together. The exceptions are *Layout*, *Header*, and *Footer* objects.
- Report objects should be in the same record for each grouping.

Let us start by connecting to the data (*Chapter 6 – Sport Report*) > M*ajor Sports Teams.xlsx*.

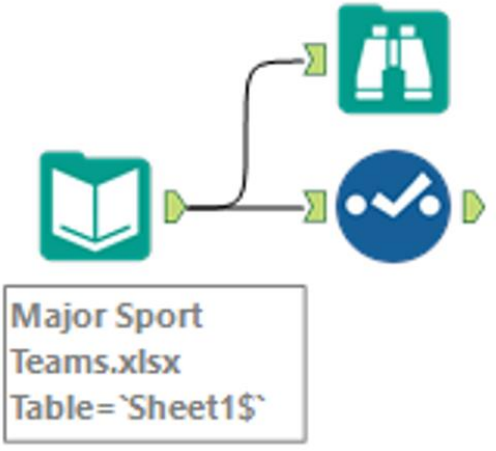


Figure 6-29 – Opening Sport Report

As in any other workflow, we will bring in the data and ensure that it is in the correct format.

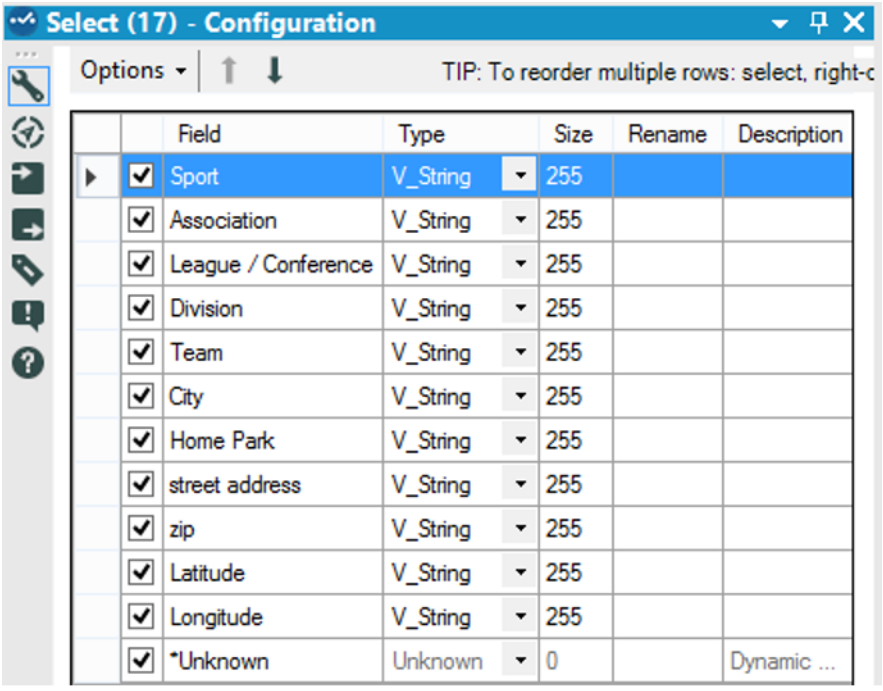


Figure 6-30 – Sports Report- Data formatting

Since we have *Latitude* and *Longitude* fields, we can easily start tackling the mapping aspect of our report. We start by adding a *Create Points* tool to the end of the workflow and using the accompanying configuration.

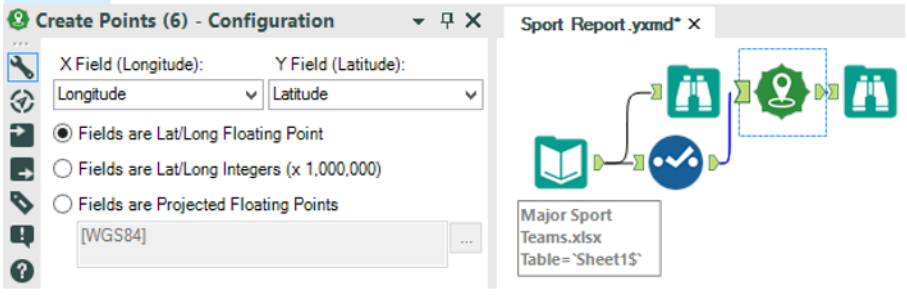


Figure 6-31 Sport Report – Mapping and creating points

Then, when we add a *Browse* tool and run the workflow, we will see that we have a field called *Centroid* in the last column with a green point object in each record. This is a special spatial field, and we will use this field in order to plot the locations of each team's home field.

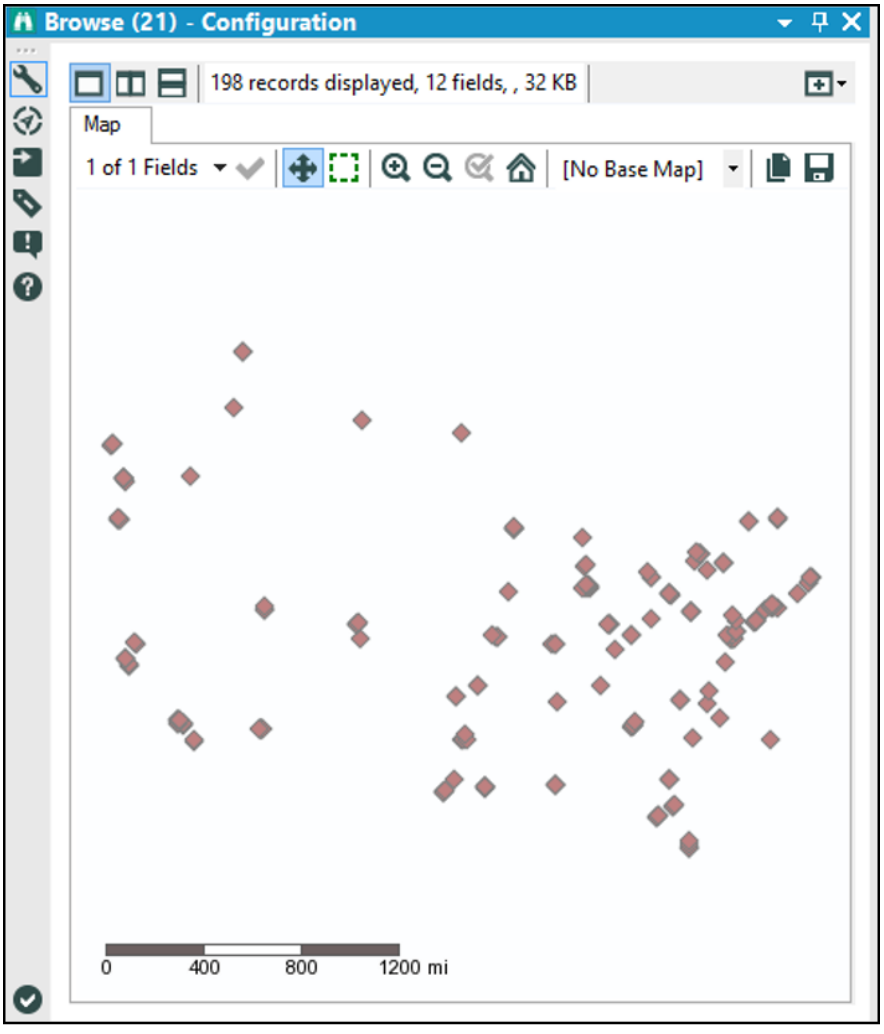


Figure 6-32 – Sport Report - run module using Browse

The resulting data stream will look like the below.

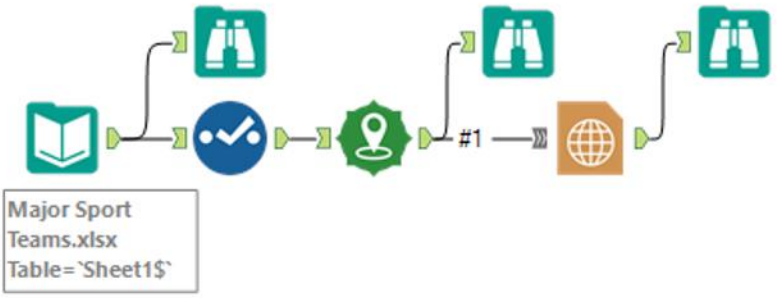


Figure 6-33 – Sports Report Data Stream

Since we are only going to use the one input for this *Report Map,* we can leave the connection labeled *#1*. We need to think about the way the report is going to look, so we should consider the overall layout. It will likely flow better if the map and chart are placed next to each other. This way, we know that image will not take more than half of the page.

In the following pages, the map configuration will allow us to create the desired maps.

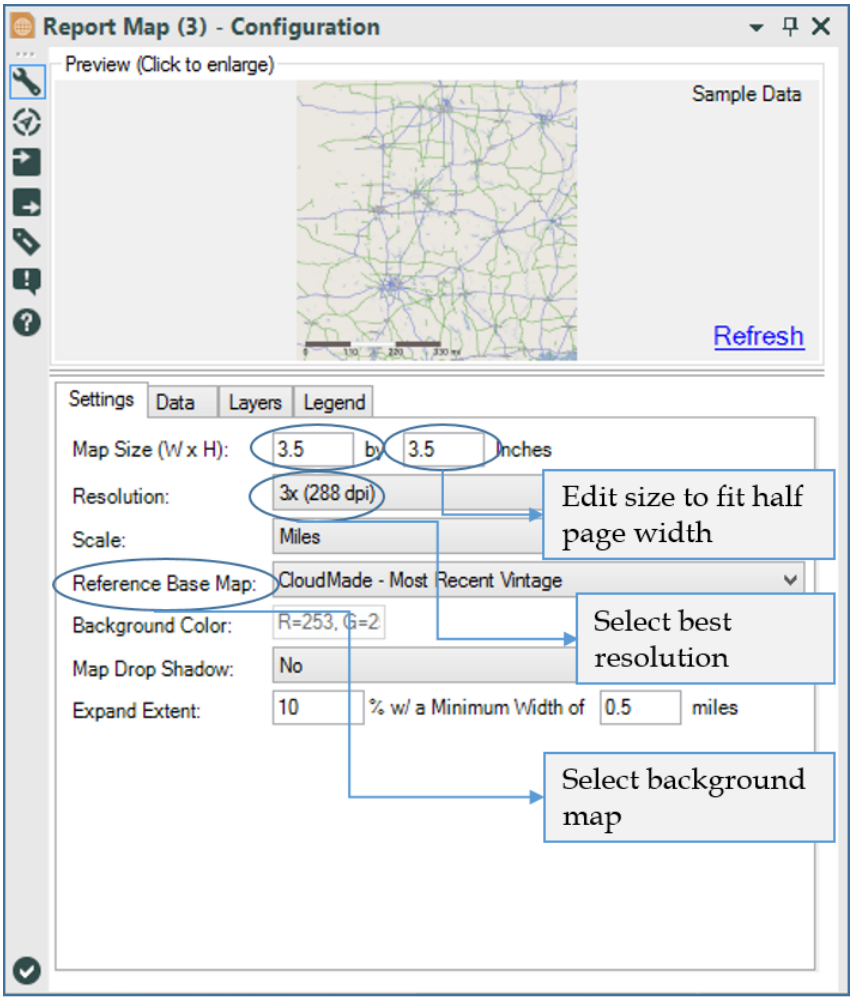


Figure 6-34 – Report map setting configuration

The following are the Report Map Data and Layers settings.

Note that we want to group by sport to make one map for each.

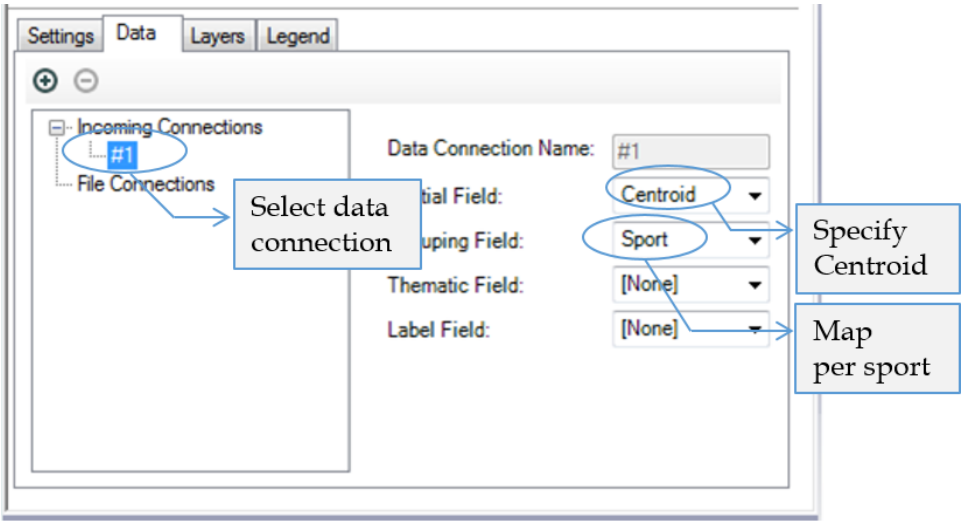


Figure 6-35 - Sport Report - Report Map Data configuration

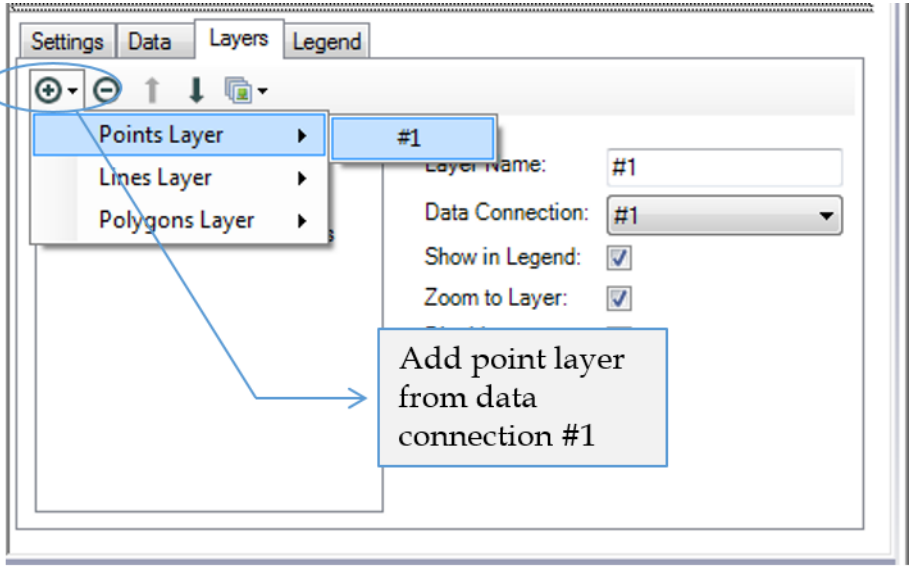


Figure 6-36 - Sport Report - Report Map Layers configuration

Normally, we would not put a *Browse* tool after *Report* tools, but it will help us understand the output of the *Report* tools.

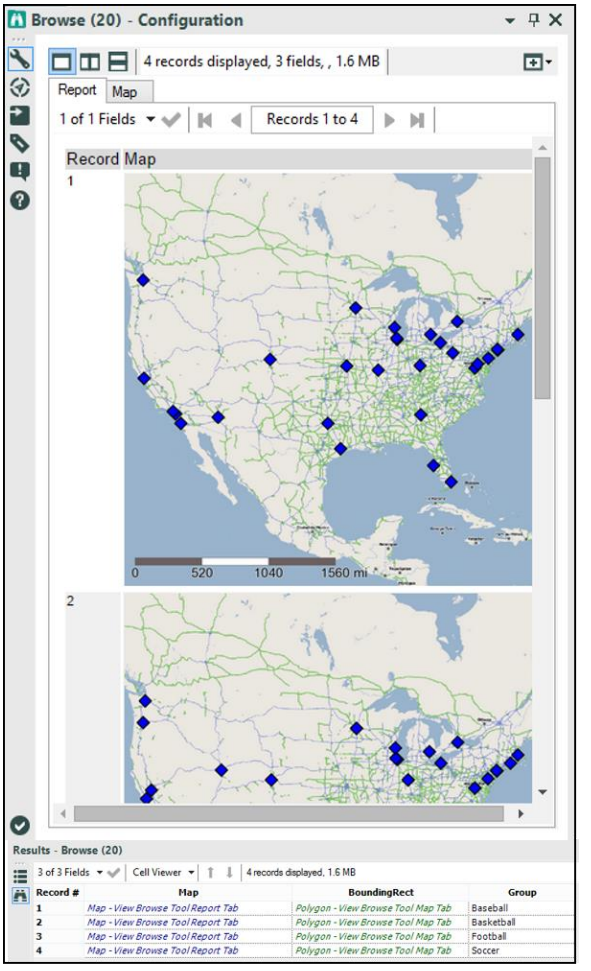


Figure 6-37 – Sport Report - Browse

If we run the module, we should see the output in the *Browse* tool at the end of the data stream.

We see that we have four records, one for each of the four sports we grouped by, identified by the *Group* column at the right. A *Map* field contains the *Report* object map for each sport. The *BoundingRect* spatial field can be used to create new maps

that are zoomed to the same area as these. Since we will not be creating any more maps, we can ignore this field.

Also, we should note that we can preview the map and report it if we want by selecting the tabs along the top.

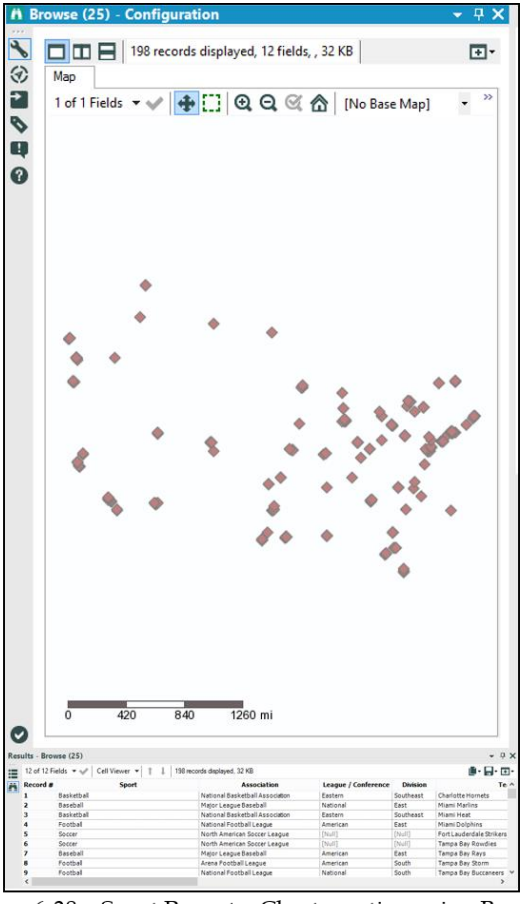


Figure 6-38 – Sport Report – Chart creation using Browse

Now that we have our maps, let us create our charts. We will be making a bar graph for each sport, showing the number of teams in each association. This means that the first thing we

should do is summarize the data by sport and association, then take the count of the team names.

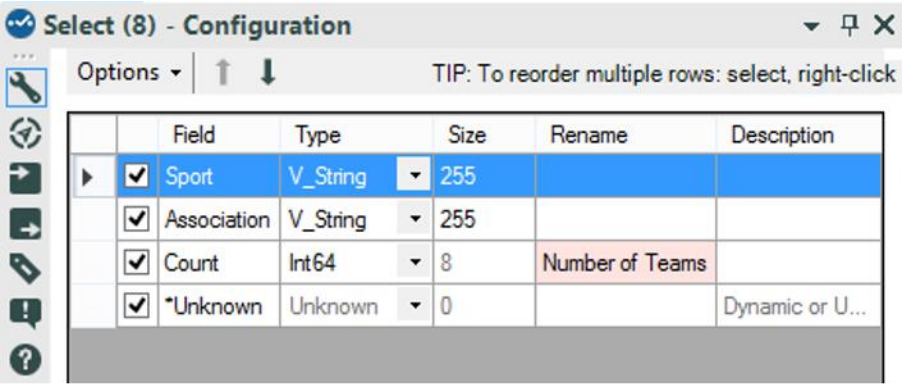


Figure 6-39 – Sports Report – Chart creating using Select

Since we want to have clean field names while reporting, we will change the name of the *Count* column to *Number of Teams*.

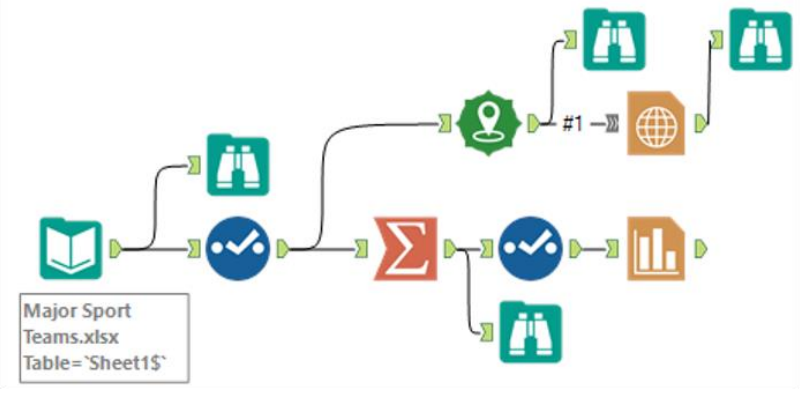


Figure 6-40 – Sport Report - Adding Charting Tool to end of data stream

Now that we have the data prepared we can add a *Charting* tool to the end of this data stream and apply the settings as shown in the image.

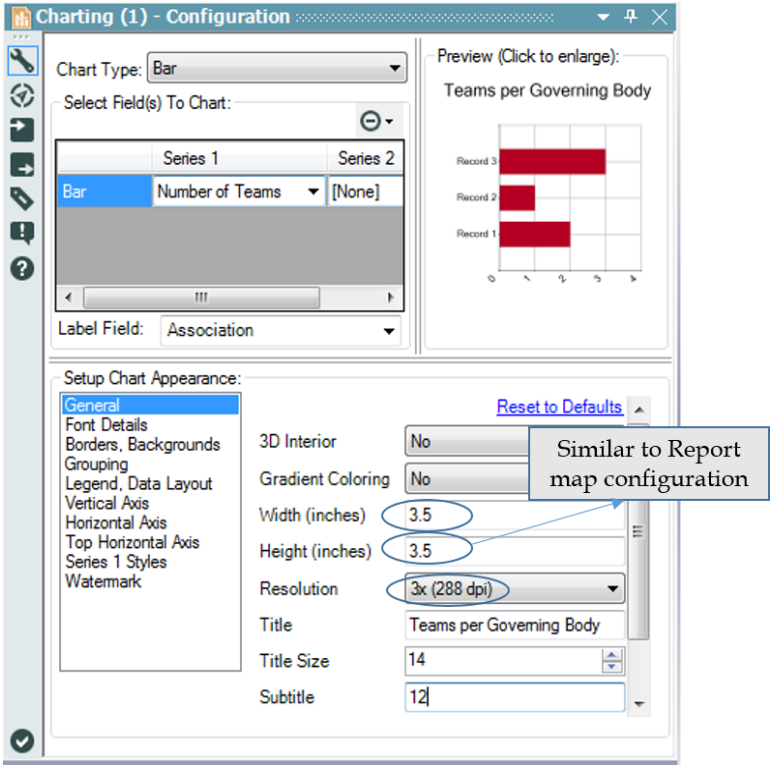


Figure 6-41 - Sport Report – Charting Tool

The images below show the grouping, legend, and data layout settings made to the *Charting* tool.

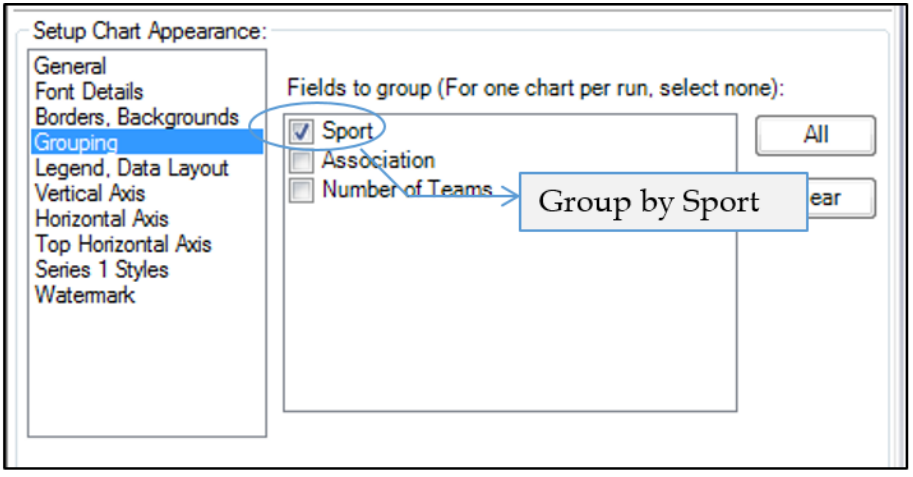


Figure 6-42 - Sport Report – Setup Chart Appearance, Grouping set to Sport

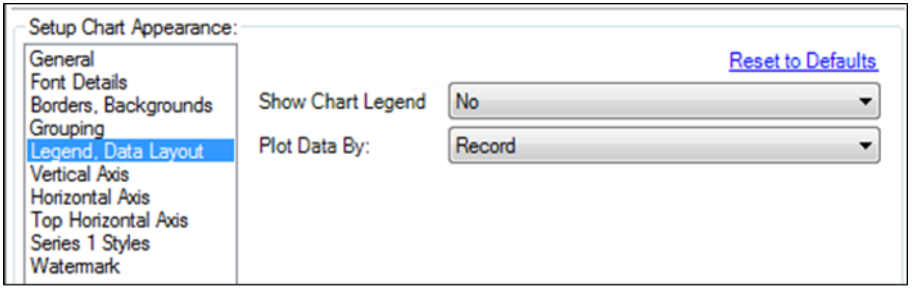


Figure 6-43 - Sport Report –
Setup Chart Appearance – Setting Legend, Data Layout

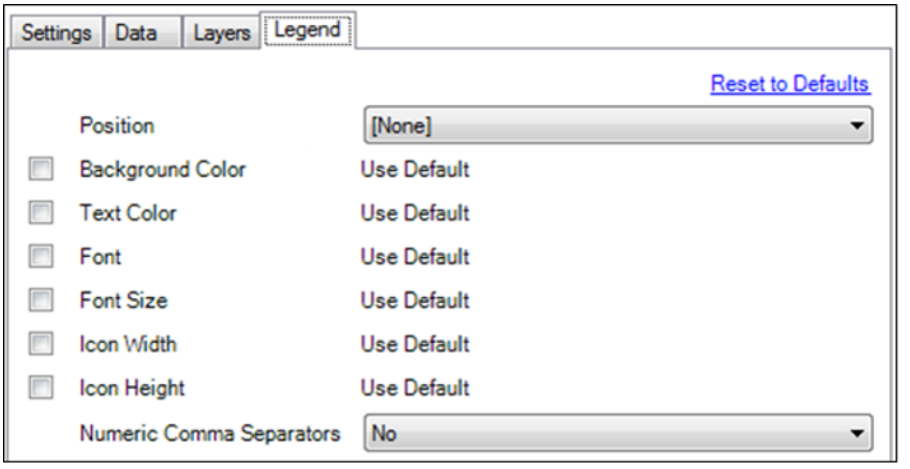


Figure 6-44 – Sport Report – Setting Legend settings (refer Report Map tool)

Let us understand the settings we have changed. The high-level settings are easy to understand. Since we wanted to create a bar chart, we selected *Bar*. Since we wanted to compare the number of teams, we selected that field for our series, and as we wanted to know the governing body related to each bar, we chose *Association*.

Under the *General* settings, we set the width and height to the same dimensions as the maps because we want them to fit side by side. We increased the resolution to have a cleaner image. We provided a descriptive title for the chart with a reduced text size so it fits on one line and we have more room for the chart. We grouped by *Sport* to make one chart for each. We removed the chart legend because we do not need it in this case.

Now that we have covered our maps and charts, we should set a title for each section. We have made a map and chart for each of the sports, but we haven't mentioned anywhere what the sport actually is. We need an object that we can put at the top of the page, with the sport in it.

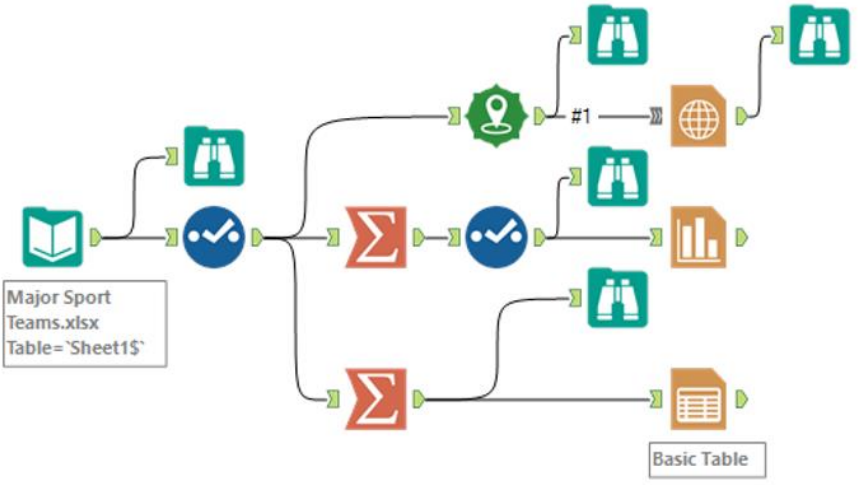


Figure 6-45 – Sports Report – Data Stream Branching

In order to create this object, we will branch off into another data stream, as shown in the previous image. We will use the *Summarize* tool to group by *Sport* so that we have a single record for each. We are then going to add a *Table* tool to create our *Report* objects.

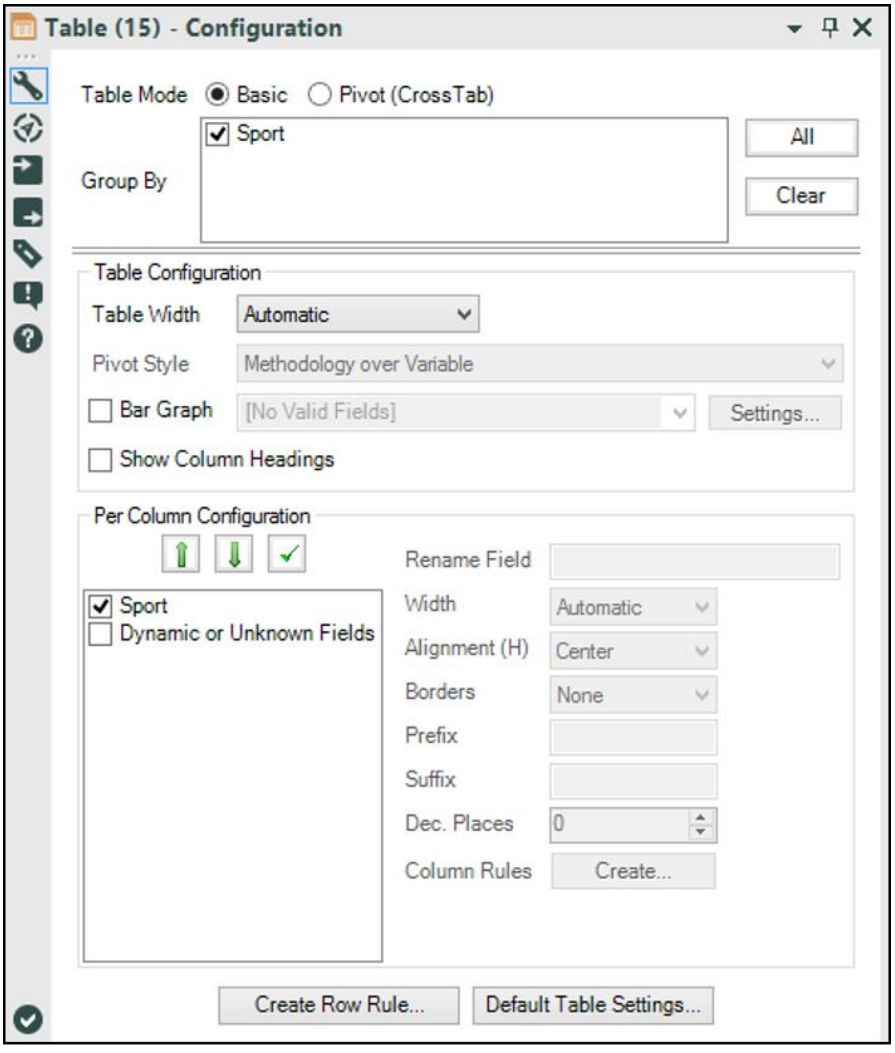


Figure 6-46 – Sport Report – grouping by sport using Table

Since we want a *Table* object for each sport, we will group by Sport. Since we are using *Sport* as a section title, we will be able to identify the sections we are interested in. Additionally, since we want to ensure that only *Sport* makes it into this table, we can deselect the *Dynamic or Unknown Fields* option.

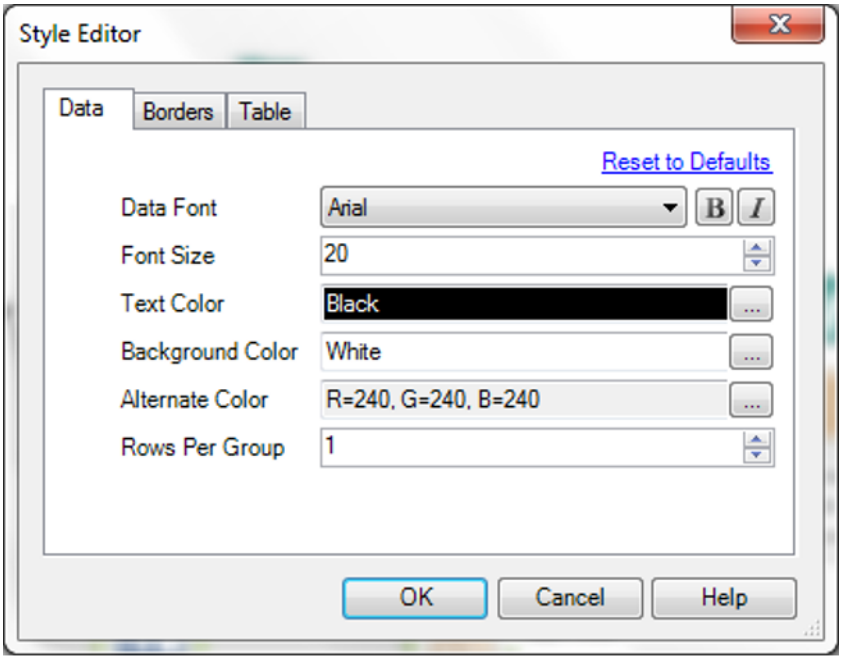


Figure 6-47 - Sport Report – setting title style using Style Editor

Since this will be the title in each section, we should increase its font size.

At this point, the following image is what our data stream will look like.

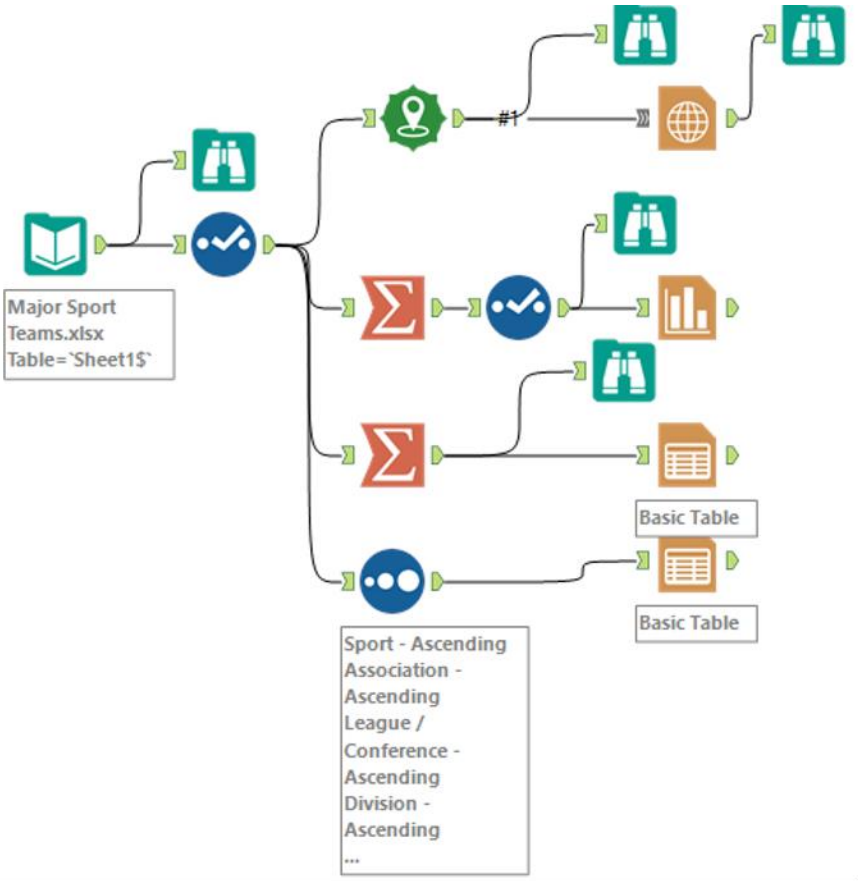


Figure 6-48 – Sport Report - data stream group by each sport

We now need to create a table for the demographic information about the teams. Since this table will be in the report, it should be sorted so the consumer can easily see what's going on.

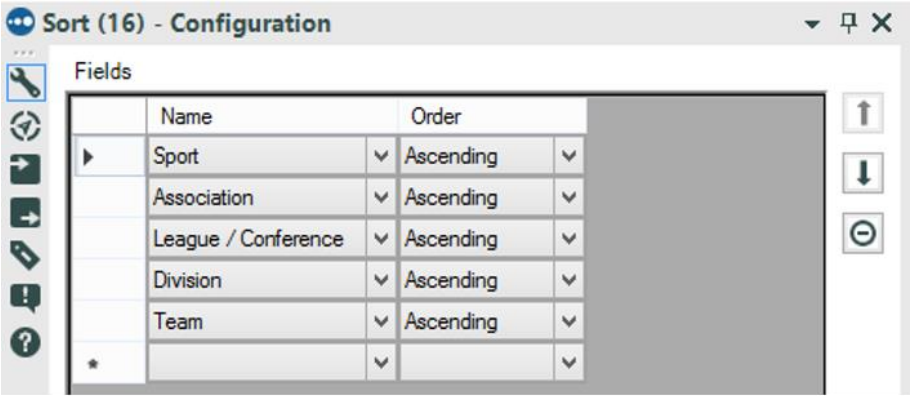


Figure 6-49 – Sport Report – Applying sort to data

Let us sort the data like we see in the *Sort Configuration* window. This will allow the report consumer to easily scan the list for the team they are interested in.

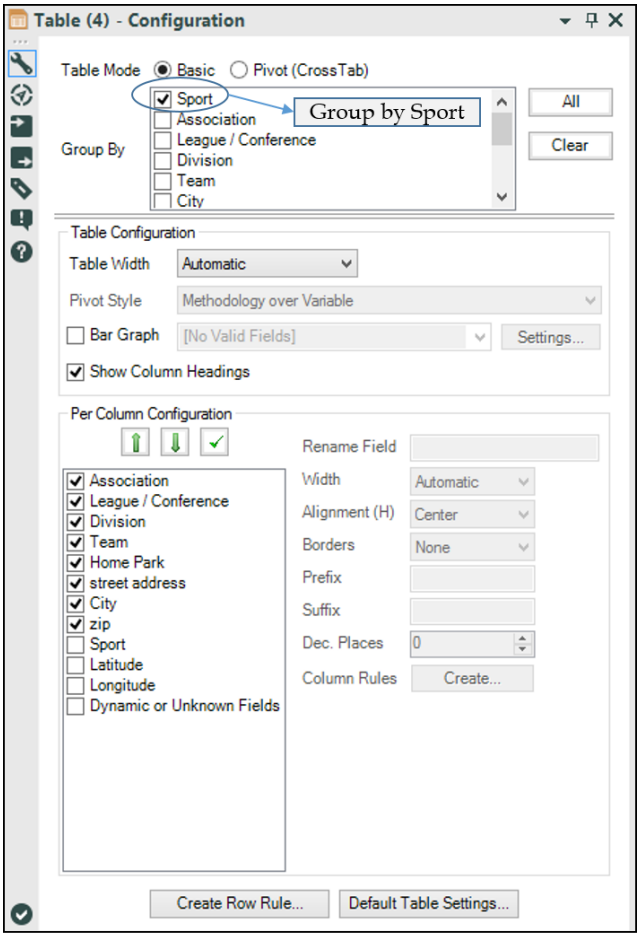


Figure 6-50 – Sports Report – team information using Table

Using these settings, we can create a table with the desired useful information about each team.

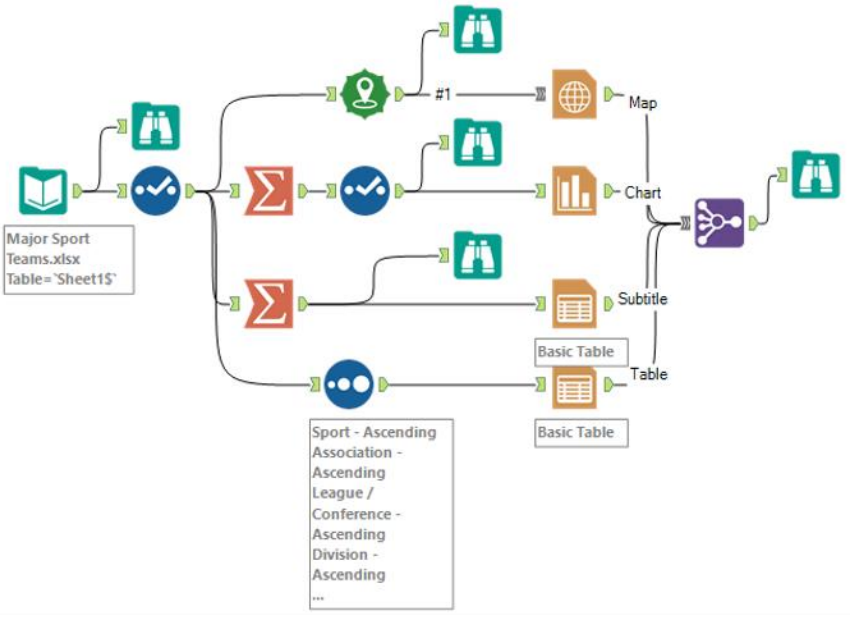


Figure 6-51 – Sport Report – with separate data streams

Now that we have created each of the objects for the body of our report, we need to bring them into a single data stream so we can format the layout, header, and footer.

In the *Browse* tool following the map we created, we saw a single record for each sport because that's the field we grouped by. Adding a *Browse* tool after each of the report objects would also result in the same, except that instead of the field being called *Group*, it would retain the title *Sport*. Since we know that the sport name makes a unique key in all of our data streams, we can use the *Multiple Join* tool to bring them together.

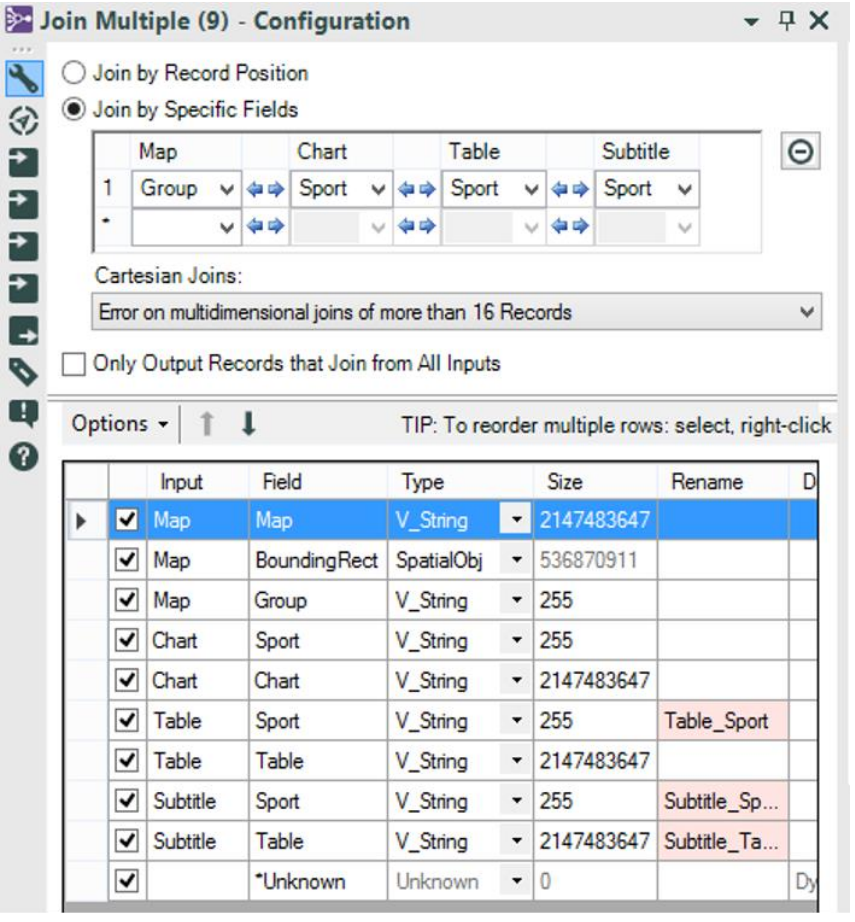


Figure 6-52 – Sport Report – Multiple Joins using Sport name

If we had named the incoming connections the way we see them on the previous image, we would see something similar to the above *Join Multiple Properties* window. Let us use the same setup as above to keep the information that will help us downstream.


Results - Browse (19)

4 of 4 Fields | Cell Viewer | 4 records displayed, 100 KB

Record #	Group	Chart	Table	Subtitle_Table
1	Baseball	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab
2	Basketball	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab
3	Football	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab
4	Soccer	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab

Figure 6-53 – Sport Report – using Browse to get data after Join

At this point, we should have five fields: The *Sport* field for our understanding, and the four report object fields use in the workflow.

Now that we have the objects for the body of our report, we need to incorporate the following into the layout:

- The sport should be at the top of each section.
- We designed the map and chart object to fit side by side, but there is no rule for where they should go.
- We have a table that can vary in length because it has one row for each team. In such a situation, the table is typically put at the bottom of the report body.

    - Based on what we see here, we need two *Layout* tools. The first is a horizontal *layout* tool that will align the map and chart objects. The second *Layout* tool is to align the section title, map-chart layout, and team information table.

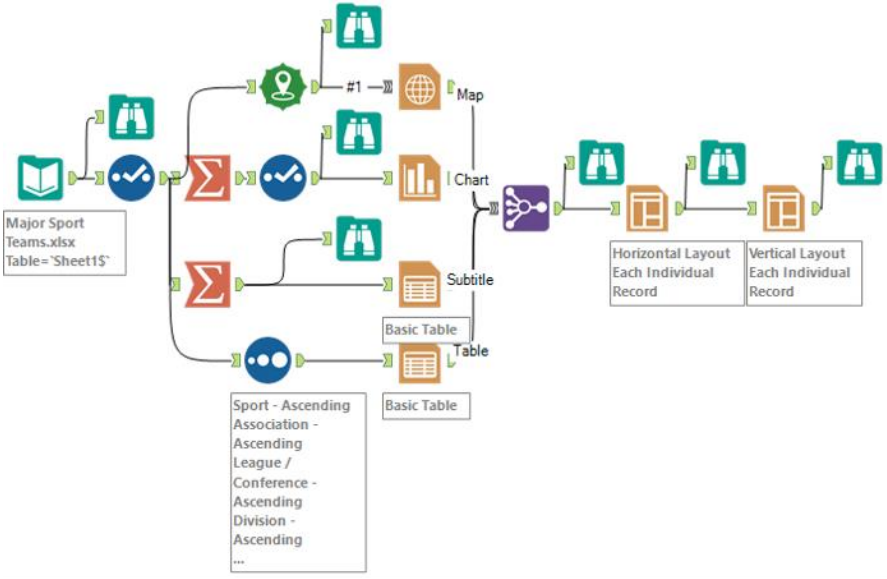


Figure 6-54 – Sport Report – Updated Data Stream

Here are the settings for the first *Layout* tool.

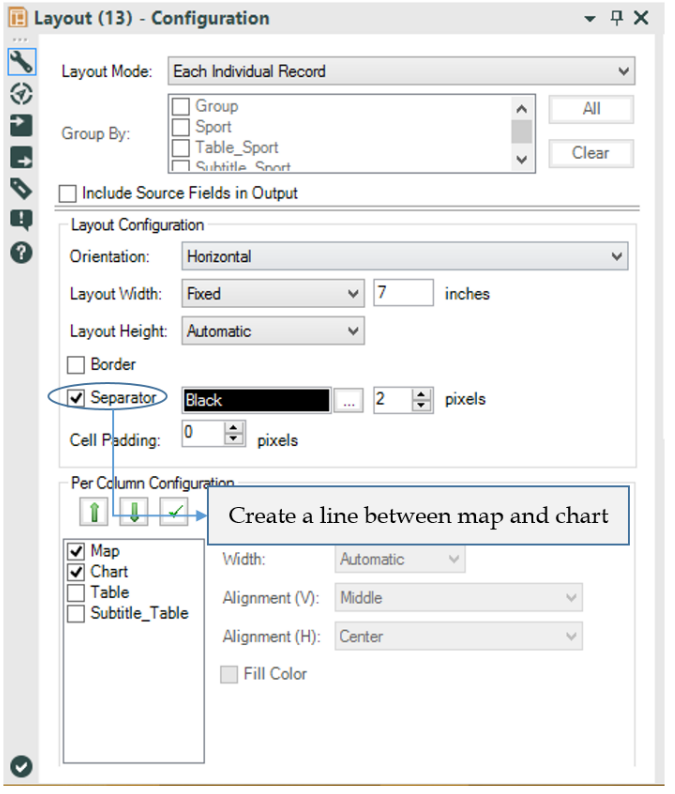


Figure 6-55 – Sport Report – using Layout to align maps, team info table etc.

A *Browse* will show us the four columns for the first *Layout* tool.

Record #	Sport	Table	Subtitle_Table	Layout
1	Baseball	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Layout - View Browse Tool Report Tab
2	Basketball	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Layout - View Browse Tool Report Tab
3	Football	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Layout - View Browse Tool Report Tab
4	Soccer	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Layout - View Browse Tool Report Tab

Figure 6-56 – Sport Report – Data after applying Layouts

The image below shows the settings for the second *Layout* tool.

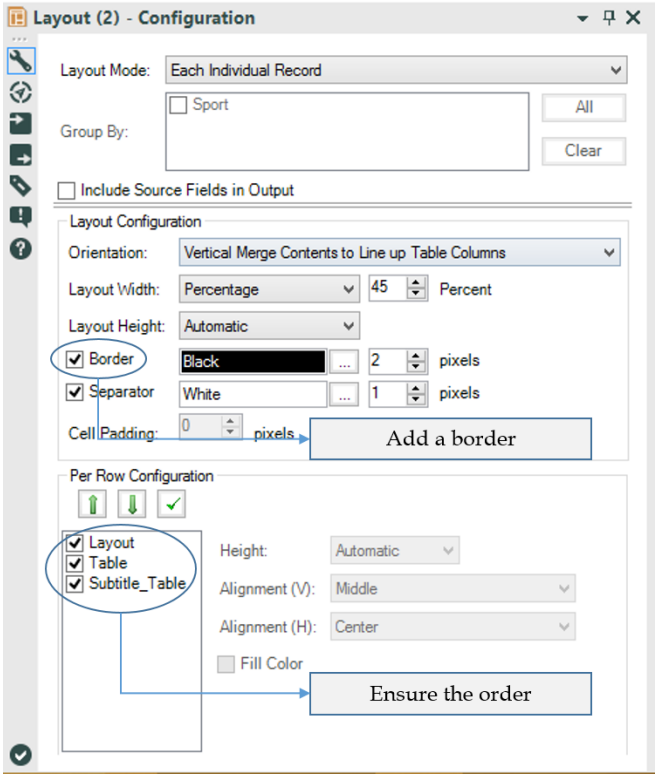


Figure 6-57 – Sport Report – applying Layout configuration

Rerunning the flow now would show the two columns in the second *Layout* tool.

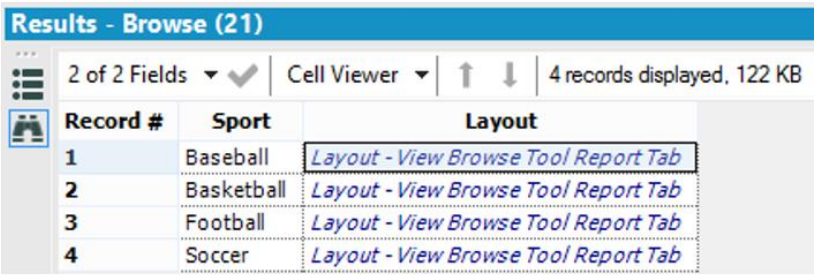


Figure 6-58 – Sport Report – selected data after applying Layout

The data stream after having the Layouts in place would look like the below.

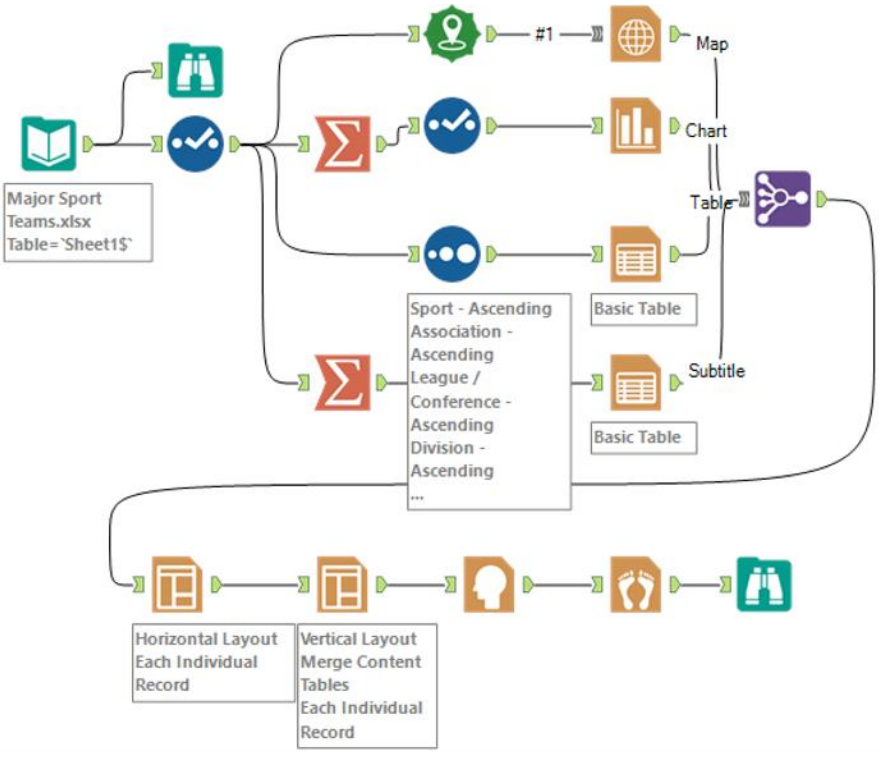


Figure 6-59 – Sport Report – data stream with Layouts in place

Now that we have combined all of the elements of the body of the report into a single report object, we can add a *Report Header* and *Report Footer* to the report. We will use the default settings for both because the client wants to see what would typically be there. We are ready to render the report, but before we do that, let us see what the data we are feeding into the *Render* tool looks like.

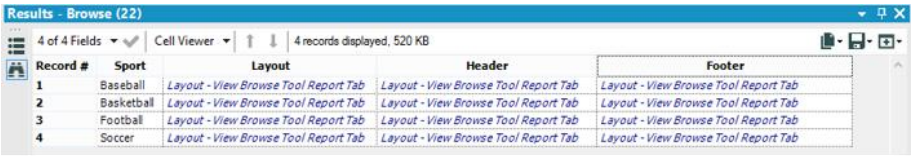


Figure 6-60 – Sport Report – Data selected after adding Report Header and Report Footer

As we can see, there is no special format for the *Header* and *Footer* fields. This means we could put any object into the *Header* and *Footer* options when we render the report. However, the *Report Header* and *Report Footer* tools create appropriately sized objects that are designed for standard report information.

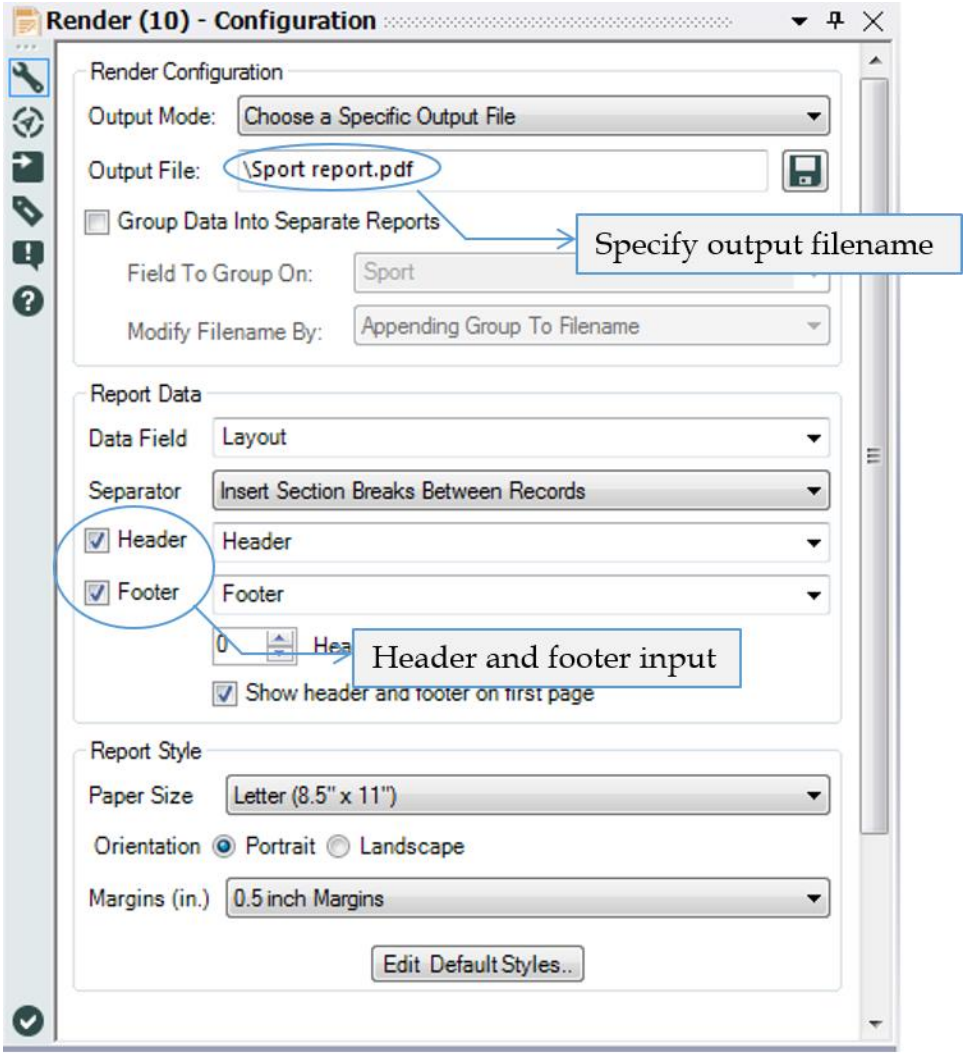


Figure 6-61 - Sport Report – applying Render to specify path

In the *Render Properties* window, we will specify that we are creating an output file, and specify a path for it. Since we want to start each sport on a new page, we will use the section break option. Finally, we will turn the Header and Footer fields on and define them.

The Major Sports Teams data stream should look like the following when its complete.

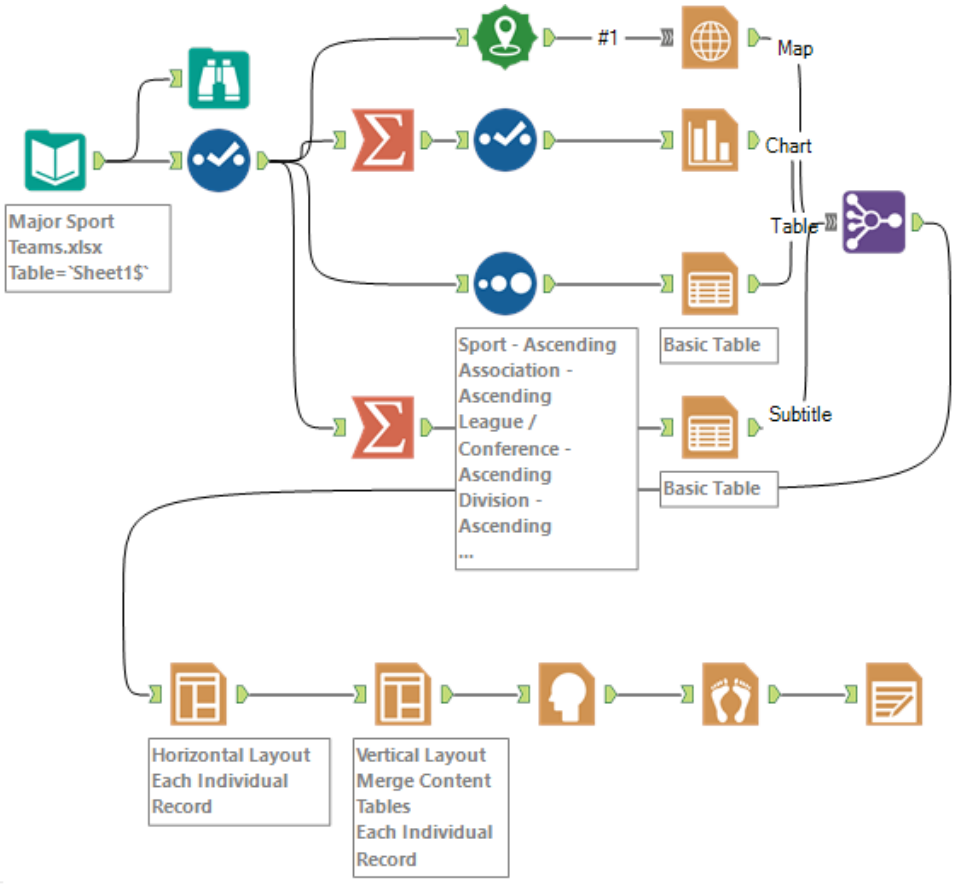
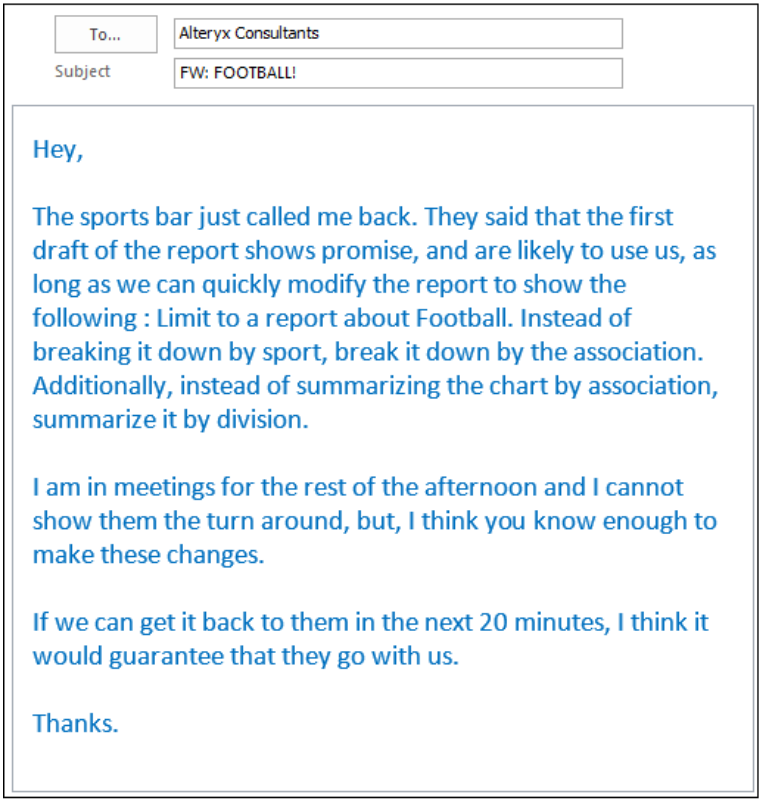


Figure 6-62 – Sport Report data stream on completion

6.11 Football!


To... Alteryx Consultants

Subject FW: FOOTBALL!

Hey,

The sports bar just called me back. They said that the first draft of the report shows promise, and are likely to use us, as long as we can quickly modify the report to show the following : Limit to a report about Football. Instead of breaking it down by sport, break it down by the association. Additionally, instead of summarizing the chart by association, summarize it by division.

I am in meetings for the rest of the afternoon and I cannot show them the turn around, but, I think you know enough to make these changes.

If we can get it back to them in the next 20 minutes, I think it would guarantee that they go with us.

Thanks.

CHAPTER 7
Expensive Beauty Products

To...	Alteryx Consultants
Subject	Expensive Beauty Products
Attached	Body Care Co.csv

Hey,

See the email below from Jan at Body Care Co.

Body Care Co. is a company that I work with pretty regularly. They have Alteryx and tend to ask us to work on projects when they don't have the bandwidth.

I think working together on this would be a good way to introduce you to a few new topics.

Thanks.

I've got something fairly straightforward for you this time.

I am using data that looks like the attached data set in order to run a lot of analyses. I'm swamped, so if you could build me an Alteryx flow that I can use going forward, that would be great!

I am using a dashboarding tool that needs to be highly responsive. Unfortunately, it is running too slow because I don't have an easy way to identify the selected data.

Below is the logic to create a flag for a number of different things I need to look at. Would you be able to create a process so the data can be modified to have these flags?

1) Lotions ("Lotion" is in the description)
2) Moisturizing Products ("Moisturizing" is in the description)
3) Baby Products ("Baby" is in the description)
4) Acne Products ("Anti-Acne" or "Anti-Oil" is in the description)
5) Create groups for all products that are named 90% identically by word (we are looking into similarly named products to see how their sales correlate).

Thanks in advance,
Jan

7.1 Tools & Concepts

Tools	Concepts
Fuzzy Match	Fuzzy Logic
Record ID	Regular Expressions
RegEx	Pre-Calculation
Unique	Data Flags
Tool Container	Removing Duplicates
	Indexing
	Group Tools
	Organize Module

7.2 Fuzzy Match

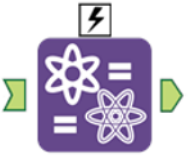


Figure 7-1 – Fuzzy Match

The *Fuzzy Match* tool allows us to apply fuzzy logic to a data stream to match similar items.

Group	Input	Output
Join	See below	See below

Note: The *Fuzzy Match* tool has multiple different methods of determining what matches. If the reader is not familiar with fuzzy logic, then it is suggested to research these techniques in order to understand how each works.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with a text field

Output: A data stream has the key field and selected *Fuzzy Match* information fields.

Properties Window:

The *Fuzzy Match Configuration* window has the following five core components.

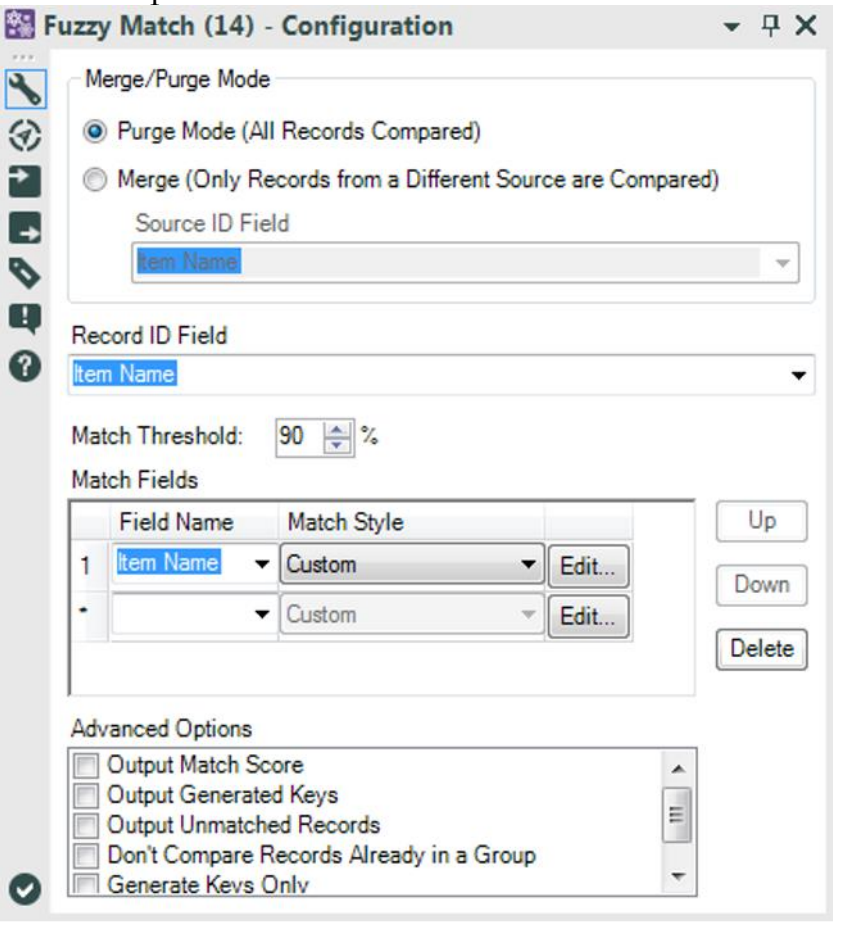


Figure 7-2 – Fuzzy Match Configuration

- *Merge/Purge Mode* allows us to decide if we need to compare all of the records in the list, or to compare records that have different sources. This means if we have data that is broken into logical buckets, we can compare across those buckets without comparing within them.
- *Record ID Field* is the field we will match on.

- *Match Threshold* is the minimum match percentage to have a positive result.
- *Match Fields* allows us to select fields and methodologies to run the analysis for matches.
- *Advanced Options* allows us to modify the output of the *Fuzzy Match*.

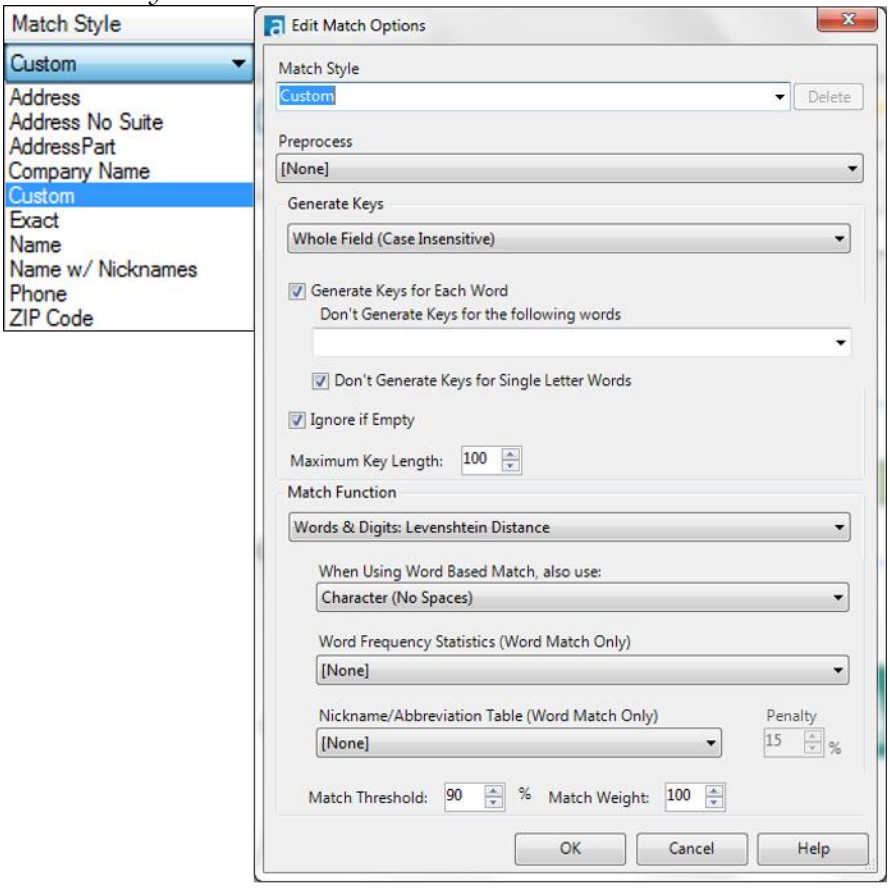


Figure 7-3 – Fuzzy Match Configuration
Edit Match options

Under *Match style,* there are some predefined matching methods. These match styles are Alteryx's suggested fuzzy logic methodologies for the type of things they name.

Custom allows us to define a methodology, which can be saved as a permanent setting if we are an administrator or super user.

The *Edit* option under the *Match Fields* section opens the *Match Options* window and offers more customization. There are many options here that allow us to define a matching algorithm. Many of the options are self-explanatory. However, if we are not familiar with fuzzy logic, we should research Levenshtein Distance and Jaro Distance before designing our own match methodology.

7.3 Record ID

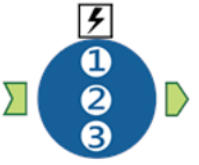


Figure 7-4 – Record ID

The *Record ID* tool creates an index column in our data stream.

Group	Input	Output
Preparation	Any data stream	See below

Note: This is useful for creating an index

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with a new field, which numerically identifies each record.

Properties Window:

The *Record ID Configuration* window has four basic components:

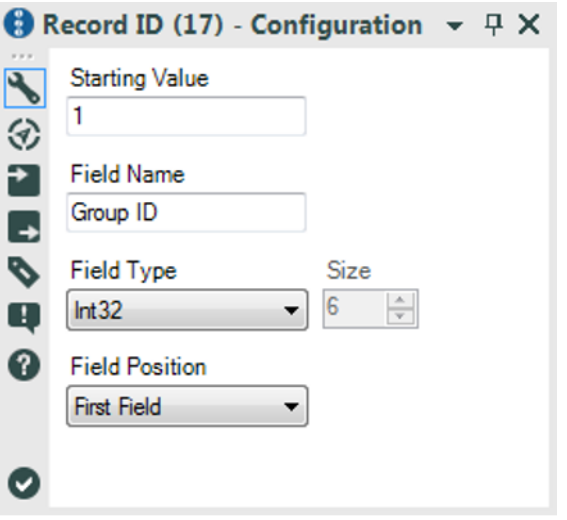


Figure 7-5 – Record ID Configuration

- *Starting Value* is the number we want to assign to the first record (which will be incremented following that record).
- *Field Name* is the resulting field name for our index.
- *Field Type* allows us to select the type and size of the resulting field.
- *Field Position* allows us to set the new field to the beginning or end of the dataset.

7.4 RegEx

Figure 7-6 – RegEx

The *RegEx* tool allows us to run regular expressions on a field in our data stream.

Group	Input	Output
Parse	See below	See below

Note: Alteryx uses the Perl regular expression syntax. See Appendix I.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with a string field.

Output: The original data stream and additional columns or rows depending on the selections made in the tool.

Properties Window:

The *RegEx Configuration* window has five basic components:

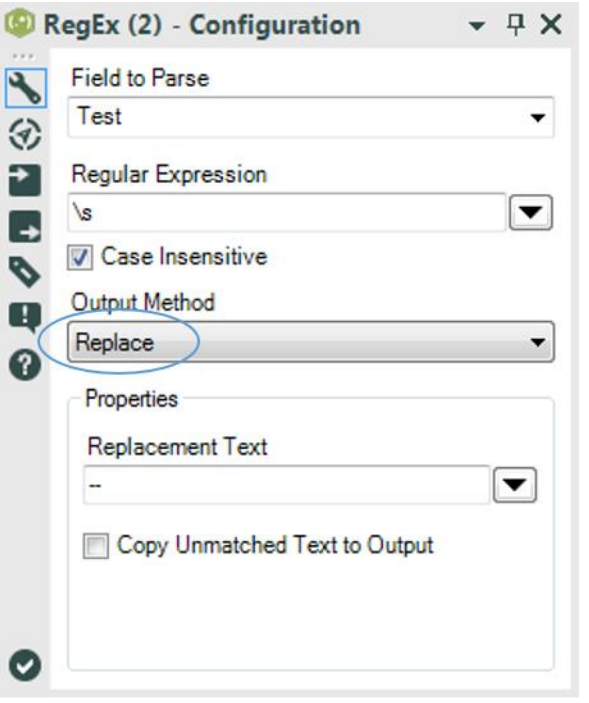


Figure 7-7 – RegEx Configuration
Replace method

- *Field to Parse* is the text field we want to manipulate.
- *Regular Expression* is the expression we use in order to identify sections of the string.
- *Case Insensitive* allows us to specify if case is important to our search.
- *Output Method* allows us to select what the regular expression will result in.
- *Properties* define specific options depending on which *Output Method* is selected.

Replace allows us to replace the text every time the *Regular Expression* is found in the *Replacement Text* string.

Copy Unmatched Text to Output will maintain any text before, between, or after the occurrences of the *Regular Expression*. If this option is not checked, the output will only have the *Replacement Text* as many times as the mentioned *Regular Expression* is found. In the above case, every time a space (\s) is in the field, it is replaced by the string characters "- -" (two hyphens).

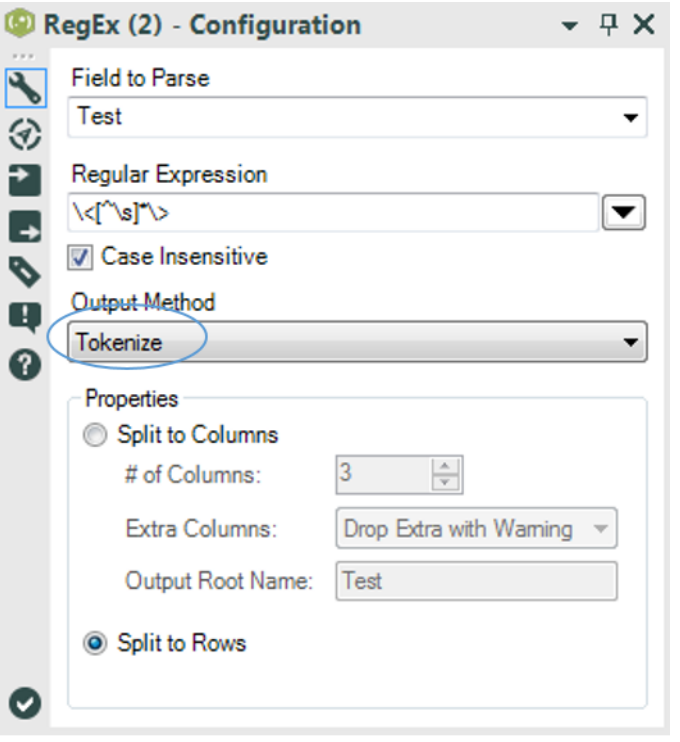


Figure 7-8 – RegEx Configuration
Tokenize method

Tokenize allows us to split the field we have, into an additional column or row every time we find the *Regular Expression* in the field.

In this case we are creating a new record for every word in a field. Since the regular expression will look for the longest occurrence of the format we need to specify that we do not want any spaces in the unknown characters between the beginning and ending of a word.

We can see that \<[^\s]*\> looks for *words* and creates a new record with all other data replicated for each of the resulting words. This is because:

\< identifies the beginning of a word

[^\s] means any character that is not a space

* means repeat the previous thing 0 or more times

\> identifies the end of a word

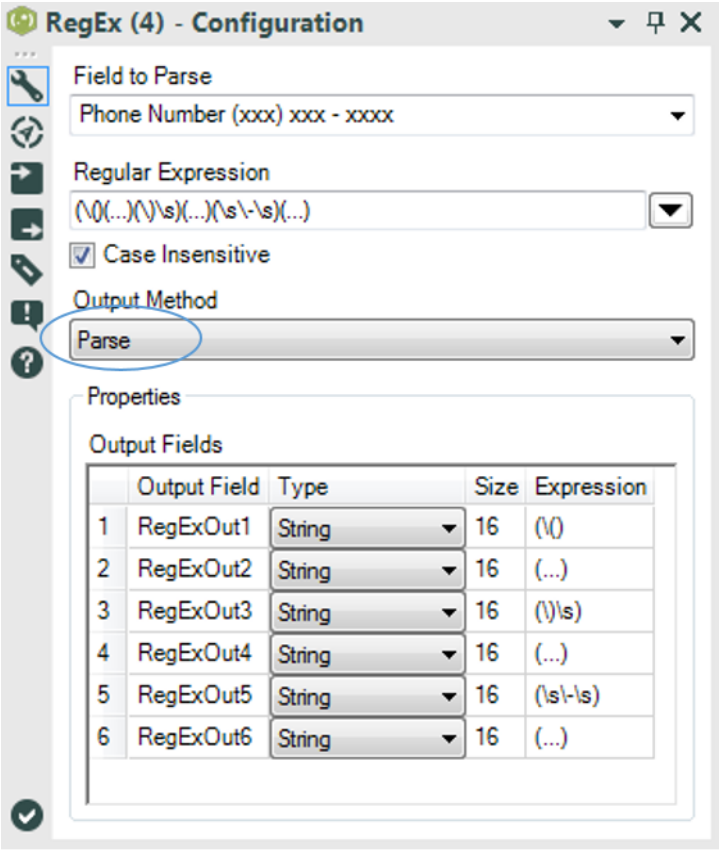


Figure 7-9 – RegEx Configuration
Parse method

Parse allows us to define *Marked Groups* in order to break down fields with specific syntax into new columns, which can be named in the *Output* fields section.

In this example image, the tool isolates the numbers from a list of phone numbers formatted as *(XXX) XXX – XXXX*.

Marked groups are identified by parentheses. The marked group (X) would select X and create a column.

The expression is made of six marked groups:

(\() Since "\(" means the "(" character, the first group is an opening parenthesis.

(…) Since "." means any single character the second group is the three characters following the previous group (which would be the first three digits of the phone number).

(\)\s) Since "\)" means the ")" character, the third group is the closing parenthesis and space before the second set of numbers.

(…) As seen earlier, it represents the next three characters. So we have the second three digits of the phone number in the fourth group.

(\s-\s) Since "-" is literally the hyphen character, the fifth group is the "space hyphen space" between the second and third set of digits.

(….) gets us the last four digits of the phone number.

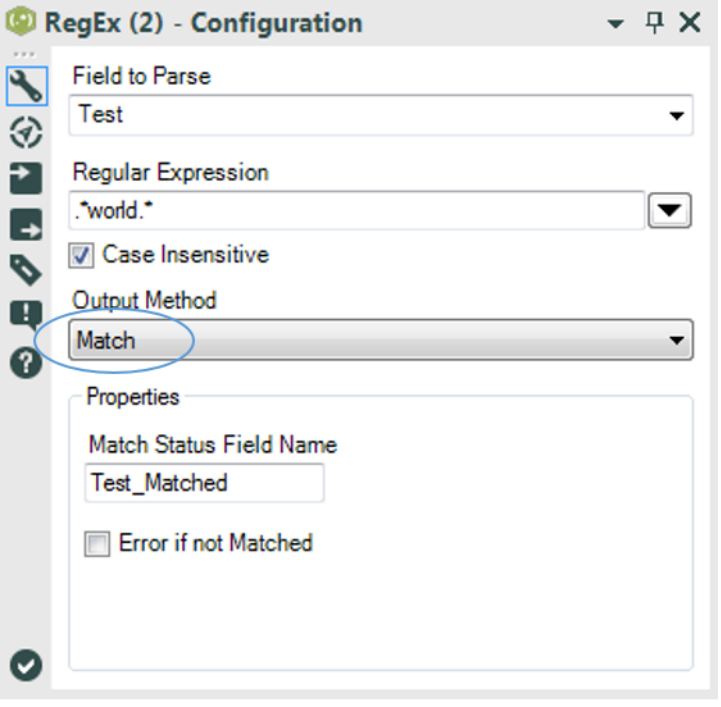


Figure 7-10 – RegEx Configuration Match method

Match creates a Boolean field such that anytime the *Regular Expression* is in the field, the result is true else it is false.

In this case we have the regular expression, .*world.*. This means that the word *world* with any characters before or after it.

The new field created is called *Test_Matched*, which can be used downstream.

7.5 Unique

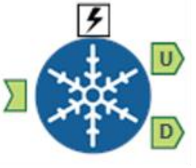


Figure 7-11 – Unique

The *Unique* tool allows us to separate unique or duplicate records based on the field(s) we select.

Group	Input	Output
Preparation	Any data stream	See below

Note: Unique records are kept if the record is the first to pass through the data stream. Unique differs from grouping by the same fields in a *Summarize* tool. It keeps all of the data associated with the record instead of just the used fields.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output U: a data stream where each record is unique based on the fields that were selected.

Output D: a data stream where each record is a duplicate of the one in the U (unique) output based on the fields that were selected.

Properties Window:

The *Unique Configuration* window is a list of each of the fields in the dataset.

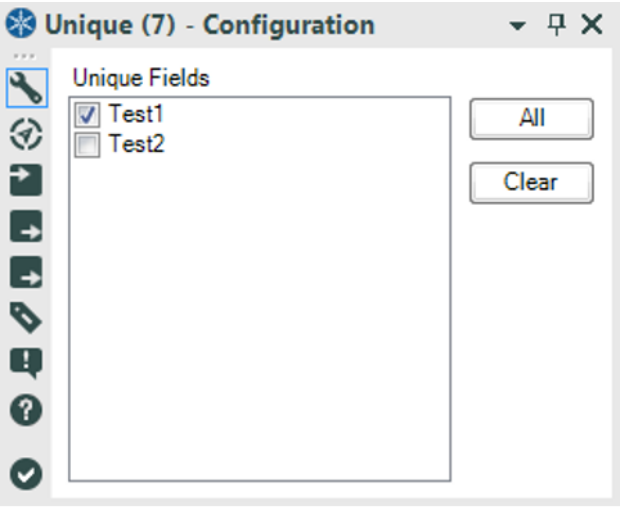


Figure 7-12 – Unique Configuration

We need to select all those we want to use to identify unique records.

7.6 Tool Container

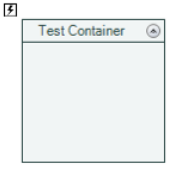


Figure 7-13 – Tool Container

The *Tool Container* allows us to group tools or segments of a data stream for easy identification of functionality within the workflow. It helps organize the workflow into modules.

Group	Input	Output
Documentation	None	None

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

It can either be enabled or disabled when the flow is executed

Properties Window:

The *Tool Container Configuration* window contains all the settings and format options for the container

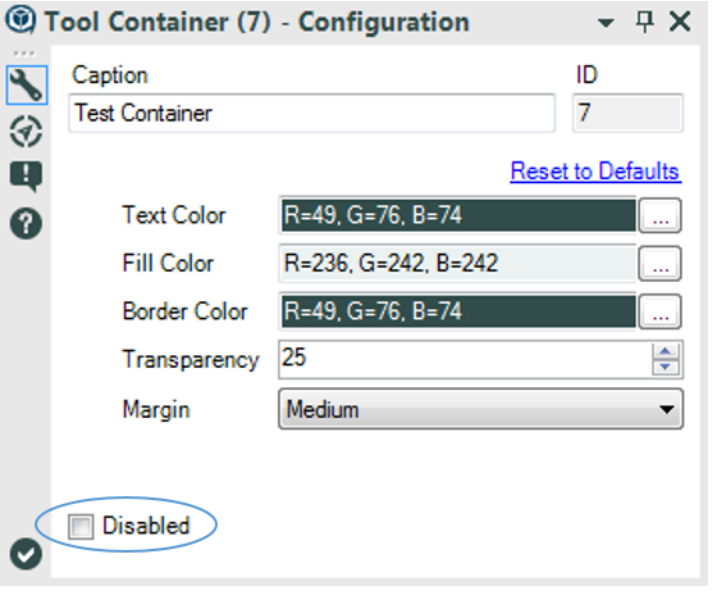


Figure 7-14 – Tool Container Configuration

Caption provides a title for the container

Color formats can be specified for the container, text within the container, and border.

The *Margin* option allows the user to set the size of the box.

The *Disabled* checkbox allows the user to control the execution of the flow or functions within the container. If the checkbox is selected, the data stream does not execute any tools in the container. The same can be controlled using an *Action* tool.

7.7 Expensive Beauty Products

To... Alteryx Consultants

Subject Expensive Beauty Products

Attached Body Care Co.csv

Hey,

See the email below from Jan at Body Care Co.

Body Care Co. is a company that I work with pretty regularly. They have Alteryx and tend to ask us to work on projects when they don't have the bandwidth.

I think working together on this would be a good way to introduce you to a few new topics.

Thanks.

I've got something fairly straightforward for you this time.

I am using data that looks like the attached data set in order to run a lot of analyses. I'm swamped, so if you could build me an Alteryx flow that I can use going forward, that would be great!

I am using a dashboarding tool that needs to be highly responsive. Unfortunately, it is running too slow because I don't have an easy way to identify the selected data.

Below is the logic to create a flag for a number of different things I need to look at. Would you be able to create a process so the data can be modified to have these flags?

1) Lotions ("Lotion" is in the description)
2) Moisturizing Products ("Moisturizing" is in the description)
3) Baby Products ("Baby" is in the description)
4) Acne Products ("Anti-Acne" or "Anti-Oil" is in the description)
5) Create groups for all products that are named 90% identically by word (we are looking into similarly named products to see how their sales correlate).

Thanks in advance,
Jan

The problem that Jan is describing is important to be aware of when we create data for dashboard and reporting tools. These tools are expected to create visuals in real time based on dynamic user inputs. Ideally, the performance is fast enough that the dashboard consumers do not observe a lag in the data load.

In order to achieve this goal, dashboard creators need well-designed and usually tidy data. This means that as much should be built into the data preparation process as possible. In Jan's case, they are doing some complex string calculations in order to filter the data appropriately.

Best practice is that any calculation the dashboard creator needs to run at a record level should be done in the data preparation stage.

To improve performance, pre-calculation steps are performed and it is a fundamental part of good data preparation.

Even when we are not preparing the data for analytics tools, it is important to pre-calculate fields the end user may need. This is because building and vetting a calculation once in a workflow means there will be a significant reduction in risk caused by human error downstream.

Let us imagine that we work for a large retailer that wants to know the profit ratios for each transaction. We may have 20 different analysts working on this project and reproducing the results every month. If these analysts are each rewriting the same formula *(Profit/Sales)* every month, it means we need to make sure they do not accidently type *(Sales/Profit)* in 240 different formulas per year. Instead, if we know they need transaction level profit ratio, we can create a standard formula in the data

preparation process that writes this value into the source data that they all use.

Besides the pre-calculation, we can see from this email that Jan is only interested in a module. This is the case because they work with Alteryx and will do some modifications, like replacing the data connection and outputs. We will often come across things like this when we are supporting other Alteryx users because they either do not know how or do not have time to build a portion of a module.

Considering the information in the email, it seems Jan was talking about flagging item names that fall into each of those buckets. We will use two techniques to achieve this. For requirements one through four, we will write regular expressions that create flags for matches and mismatches. And for the fifth, we will use fuzzy logic to define grouping for each of the tools.

Regular expressions are pattern-matching formulae that allow us to define a pattern to look for in a string. Alteryx uses the Perl programing language syntax for its regular expressions, so any documentation related to Perl regular expressions will help us get up to speed. A cheat sheet is provided at the end. (See Appendix I.)

Fuzzy logic is essentially a logical process that allows computers to say, "Eh, close enough!" As far as Alteryx is concerned, this is limited to string parsing; however, more generally this extends to programming a definition of *tall*, *heavy*, and *blue*. There are various methodologies of matching strings. Alteryx uses the *Levenshtein Distance* and *Jaro Distance* to achieve this. The nuances of the differences are well documented and it

is suggested that the reader learn more about them if they plan to use Alteryx's *Fuzzy Match* tool.

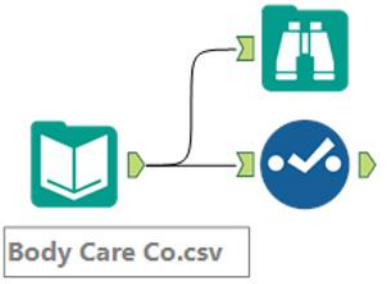


Figure 7-15 – Expensive Beauty Products
Body Care Company data

If we connect to the file in *Chapter 7 -- Expensive Beauty Products* > *Body Care Co.csv*, we see that Alteryx is reading all of the fields in this file as strings.

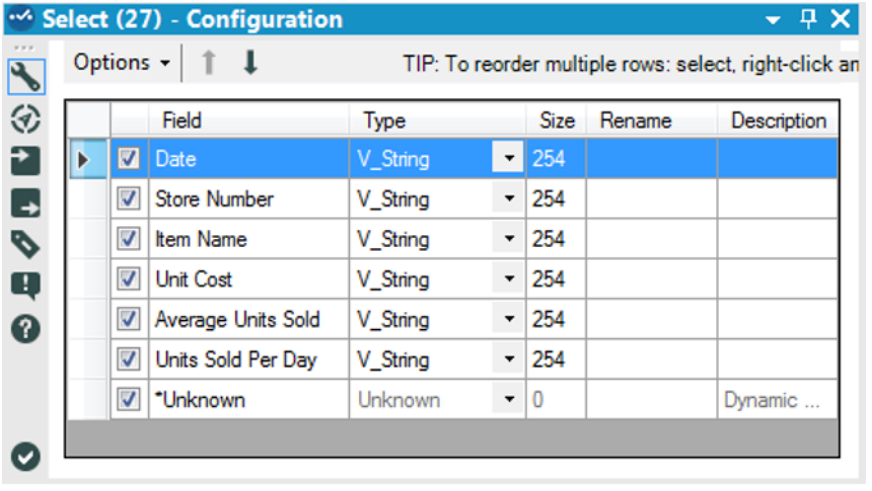


Figure 7-16 – Expensive Beauty Products
Select Configuration

Ideally, we should change the last three fields to numeric. But since we are only going to use *Item Name* and we don't know

what the rest of Jan's process looks like, we should leave the fields the way they are.

Since we need to make sure Jan knows what each piece of this module is doing, we have to make sure that it is well documented. The first thing we will do is put our initial connection into a tool container that indicates that the entire set of tools should be replaced when the connection is changed.

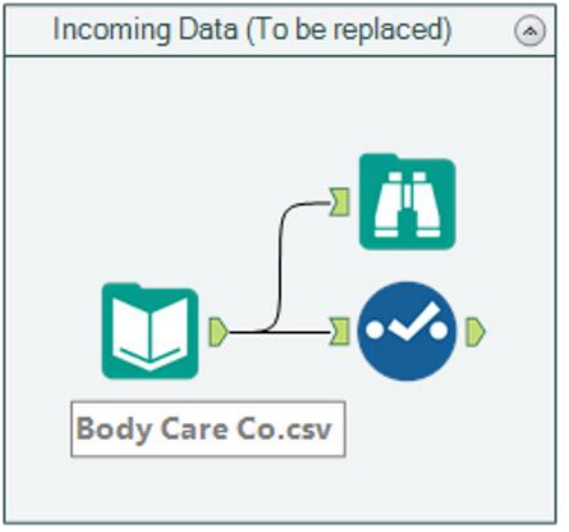


Figure 7-17 – Expensive Beauty Products Input data – to be replaced in a container

Let us take a look at the contents of this file.

Results - Browse (29)

6 of 6 Fields | Cell Viewer | 188,868 records displayed, 2.8 MB

Record #	Date	Store Number	Item Name	Unit Cost	Average Units Sold	Units Sold Per Day
1	1/1/2014	1	Vitamin Restoring Anti-Aging Cream 1 Ounce	1.99	0	3
2	1/1/2014	1	Anti-Itch Cream 1 Ounce	1.99	0	3
3	1/1/2014	1	Anti-Itdh Lotion 5 Ounce	1.99	1	2
4	1/1/2014	1	Natural Protection Sunscreen SPF 50 3 Ounce	10.99	0	3
5	1/1/2014	1	Baby Calming Bubble Bath 8 Ounce	3.99	0	3
6	1/1/2014	1	Baby Lotion 8 Ounce	2.99	0	2
7	1/1/2014	1	Baby Lotion 20 Ounce	5.99	0	1
8	1/1/2014	1	Baby Soothing Anti-Itch Cream 3 Ounce	3.99	0	3
9	1/1/2014	1	Baby Wash And Shampoo 18 Ounce	2.99	0	2
10	1/1/2014	1	Lavender Body Wash 14 Ounce	2.99	0	2
11	1/1/2014	1	Stress Relief Body Wash 10 Ounce	2.99	0	1
12	1/1/2014	1	Anti-Acne Medicated Body Cream 2 Ounce	11.99	0	2

Figure 7-18 – Expensive Beauty Products
Input data – Browse configuration

Next, let us create flags using regular expressions. We will create a new *Tool Container* called Regular Expressions with a *RegEx* tool in it.

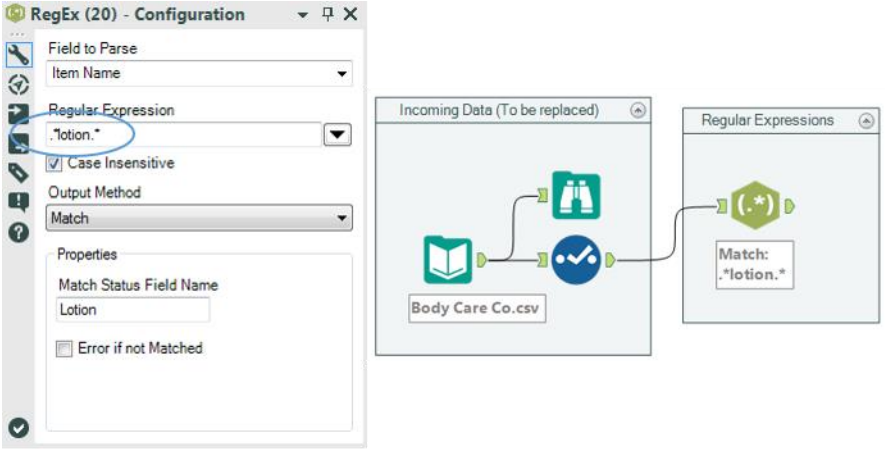


Figure 7-19 – Expensive Beauty Products
RegEx – "lotion" flag – Match method

The first requirement we want is a flag to identify if *lotion* is in the *Item Name* field. We can use the expression *.*lotion.** to identify any field that has the string "lotion" in it.

This regular expression checks to see if the literal pattern "0 or more characters followed by 'lotion' followed by 0 or more characters" is in each record and returning true or false.

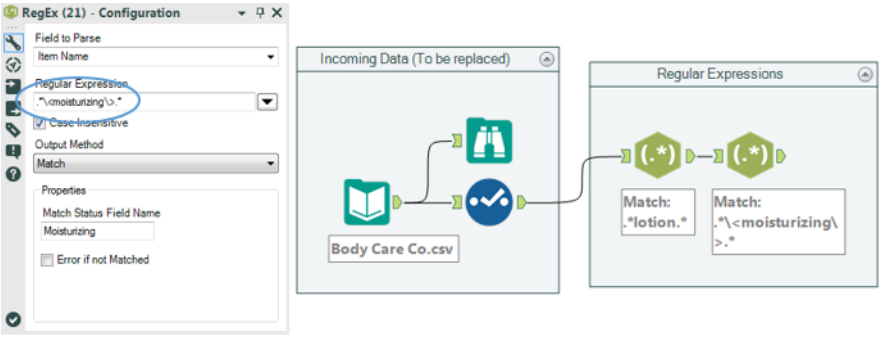


Figure 7-20 – Expensive Beauty Products
RegEx – "moisturizing" flag – Match method

A familiarity with Body Care Co's data lets us know their item names always have spaces between words. This means we can actually make our expression more specific as shown in the image, with .*\<*moisturizing*\>.*

This regular expression reads "0 or more characters followed by the word 'moisturizing' followed by 0 or more characters."

The special characters \< and \> in regular expressions take into account specific characters preceding or ending a word to identify if the entire word is matched.

Following the same pattern, we can create a field called *Baby Product*.

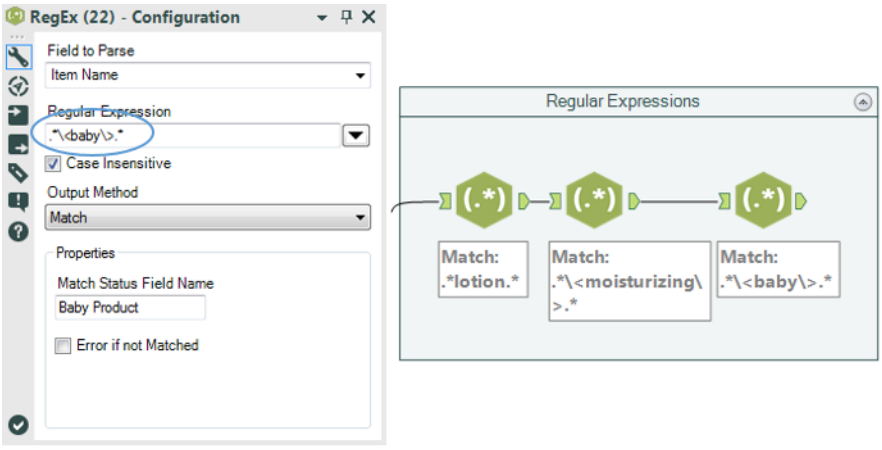


Figure 7-21 – Expensive Beauty Products
RegEx – "baby" flag – Match method

We then make another for acne products.

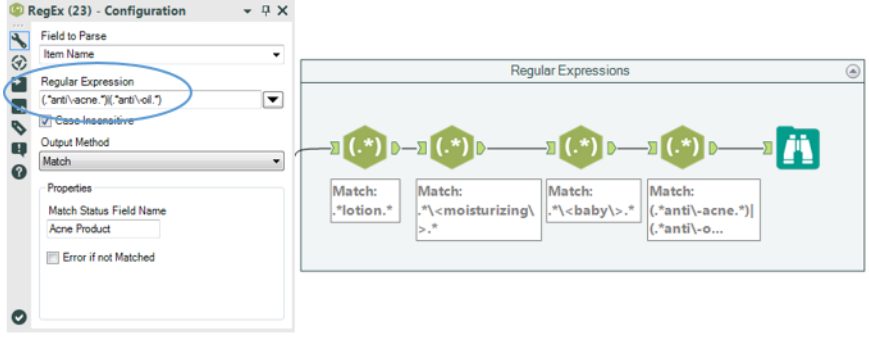


Figure 7-22 – Expensive Beauty Products
RegEx – a complex flag – Match method

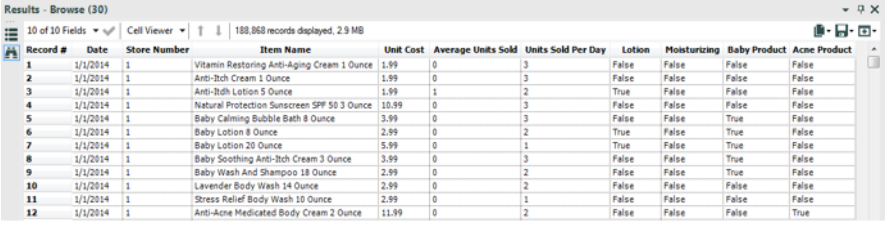
Our last flag is more complicated, because we have two different things that could define acne products.

We will combine what we know about Boolean statements and regular expressions to create the expression

(.*anti\-acne.*)|(.*anti\-oil.*)

This regular expression is checking to see if the pattern "0 or more characters followed by 'anti-acne' followed by 0 or more characters, or 0 or more characters followed by 'anti-oil' followed by 0 or more characters' exists.

As we can see, regular expressions can get complex, but there are always multiple ways to create them. In this case, we could have also used *.*anti\-(acne|oil).**, where it would have looked for the pattern "0 or more characters followed by 'anti-' followed by 'acne' or 'oil' followed by 0 or more characters".



Results - Browse (30)

10 of 10 Fields | Cell Viewer | 188,868 records displayed, 2.9 MB

Record #	Date	Store Number	Item Name	Unit Cost	Average Units Sold	Units Sold Per Day	Lotion	Moisturizing	Baby Product	Acne Product
1	1/1/2014	1	Vitamin Restoring Anti-Aging Cream 1 Ounce	1.99	0	3	False	False	False	False
2	1/1/2014	1	Anti-Itch Cream 1 Ounce	1.99	0	3	False	False	False	False
3	1/1/2014	1	Anti-Itdh Lotion 5 Ounce	1.99	1	2	True	False	False	False
4	1/1/2014	1	Natural Protection Sunscreen SPF 50 3 Ounce	10.99	0	3	False	False	False	False
5	1/1/2014	1	Baby Calming Bubble Bath 8 Ounce	3.99	0	3	False	False	True	False
6	1/1/2014	1	Baby Lotion 8 Ounce	2.99	0	2	True	False	True	False
7	1/1/2014	1	Baby Lotion 20 Ounce	5.99	0	1	True	False	True	False
8	1/1/2014	1	Baby Soothing Anti-Itch Cream 3 Ounce	3.99	0	3	False	False	True	False
9	1/1/2014	1	Baby Wash And Shampoo 18 Ounce	2.99	0	2	False	False	True	False
10	1/1/2014	1	Lavender Body Wash 14 Ounce	2.99	0	2	False	False	False	False
11	1/1/2014	1	Stress Relief Body Wash 10 Ounce	2.99	0	1	False	False	False	False
12	1/1/2014	1	Anti-Acne Medicated Body Cream 2 Ounce	11.99	0	2	False	False	False	True

Figure 7-23 – Expensive Beauty Products Boolean filed (flags) for RegEx Match

We can see that we have the four Boolean fields to identify each of the four sets of products we have defined.

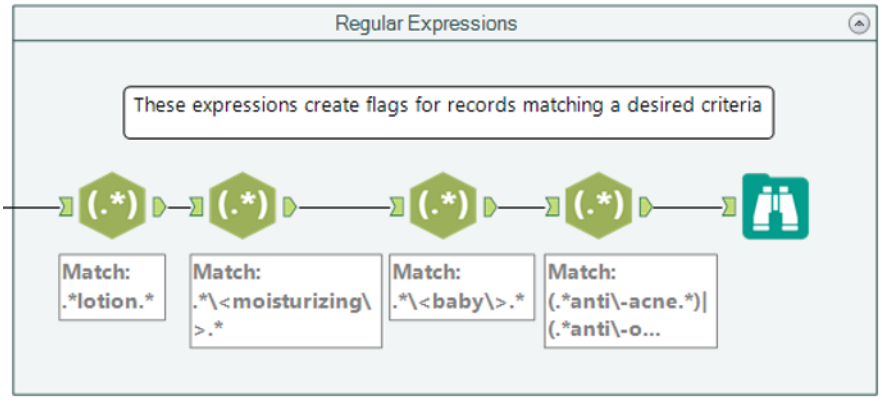


Figure 7-24 – Expensive Beauty Products
Adding comments to indicate contents in Tool Container

We can now add a *Comment* tool that describes what is happening in the *Tool Container* and move onto the fuzzy logic.

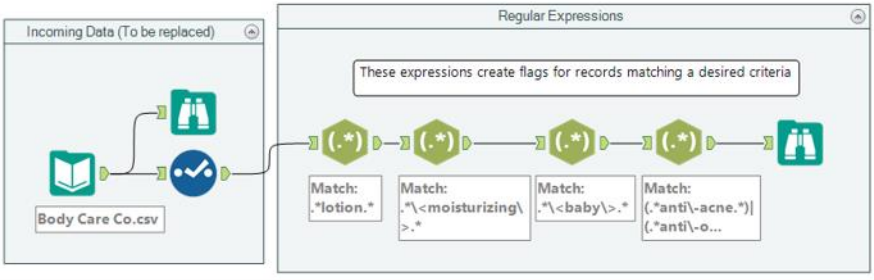


Figure 7-25 – Expensive Beauty Products
Updated data stream

We need to define the groups of products with similar names. The first thing we should do is to create a unique list of product names. Let us introduce a *Summarize* tool within a container and connect it to data input stream.

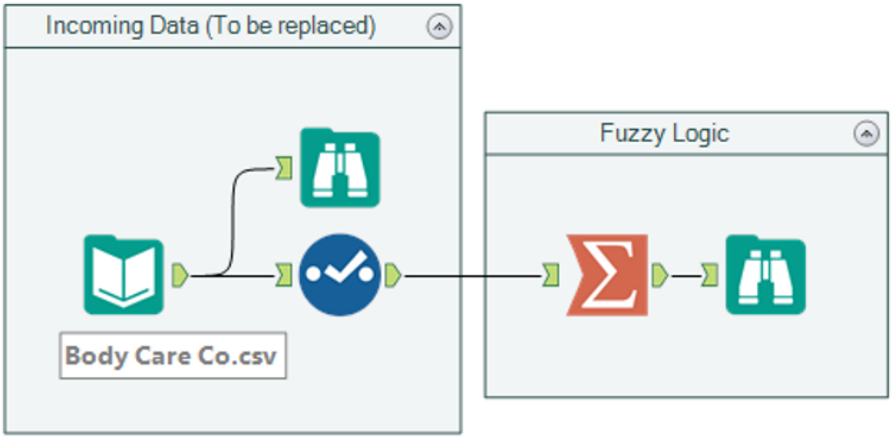


Figure 7-26 – Expensive Beauty Products second data stream for fuzzy match

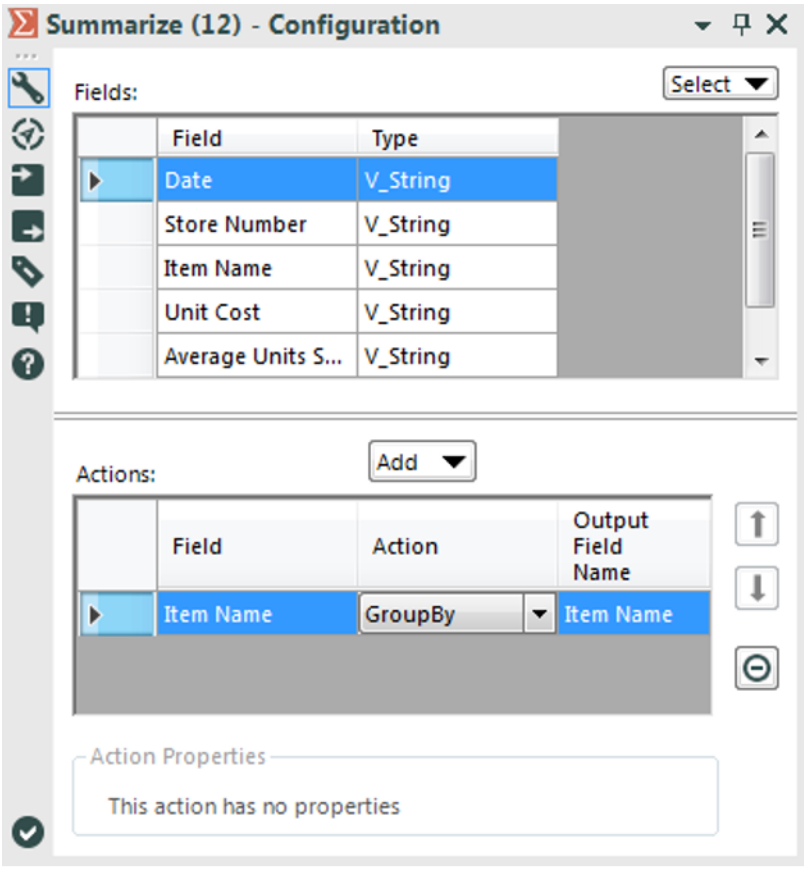
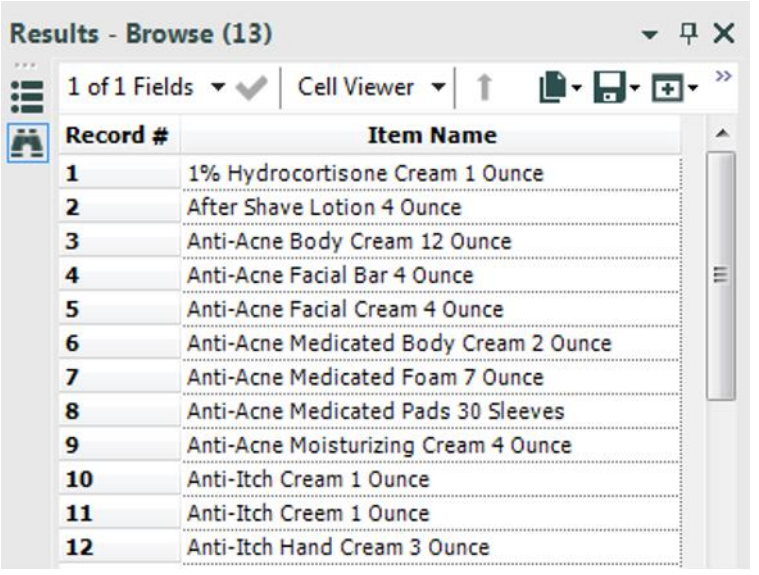


Figure 7-27 – Expensive Beauty Products Summarize configuration

Let us run the workflow and take a look at the unique list.



Record #	Item Name
1	1% Hydrocortisone Cream 1 Ounce
2	After Shave Lotion 4 Ounce
3	Anti-Acne Body Cream 12 Ounce
4	Anti-Acne Facial Bar 4 Ounce
5	Anti-Acne Facial Cream 4 Ounce
6	Anti-Acne Medicated Body Cream 2 Ounce
7	Anti-Acne Medicated Foam 7 Ounce
8	Anti-Acne Medicated Pads 30 Sleeves
9	Anti-Acne Moisturizing Cream 4 Ounce
10	Anti-Itch Cream 1 Ounce
11	Anti-Itch Creem 1 Ounce
12	Anti-Itch Hand Cream 3 Ounce

Figure 7-28 – Expensive Beauty Products Browse configuration after Summarize

Now that we have a unique list of products, we can determine which of those products have similar names.

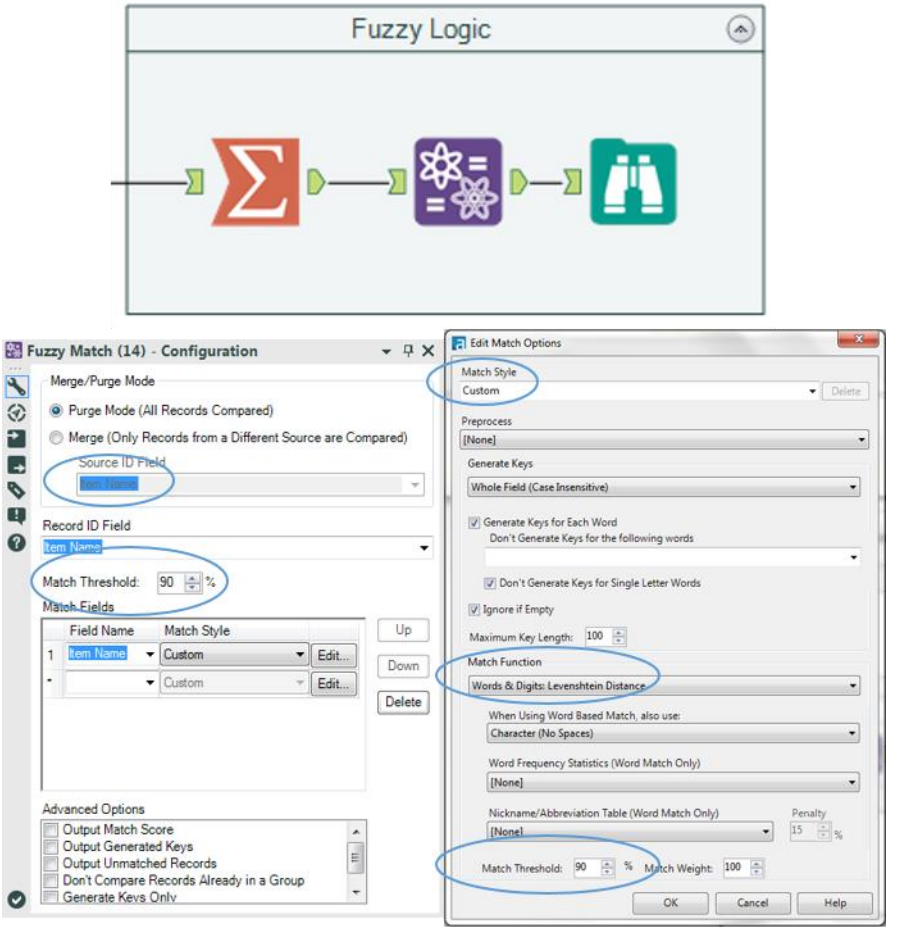


Figure 7-29 – Expensive Beauty Products Fuzzy Logic configuration

If we add a Fuzzy Match tool with the above settings, we know we are matching *Item Name* at a minimum 90% threshold. The settings here define a matching algorithm using a word and digit-based *Levenshtein distance method*. This algorithm is looking at the whole field by word (except for single-character word; up to 100 words per field) and keeping anything that it finds with at least a 90% match.

Because of the nature of this methodology, we will introduce an issue in the output data stream. That will result in duplicate records for the fields that match based on multiple keys. In order to correct this, we will isolate the unique records to be kept.

If we add a *Unique* tool after the *Fuzzy Match,* we can see a new field has been created called *Item Name2*. The new field allows us to see which entries match.

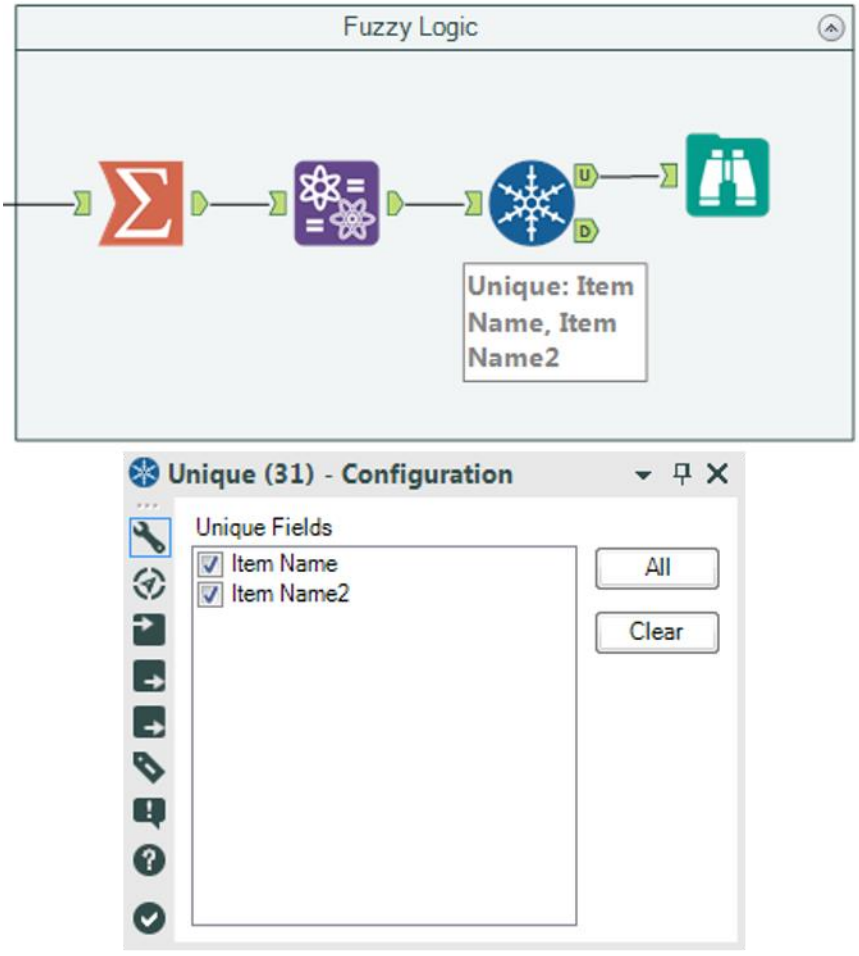
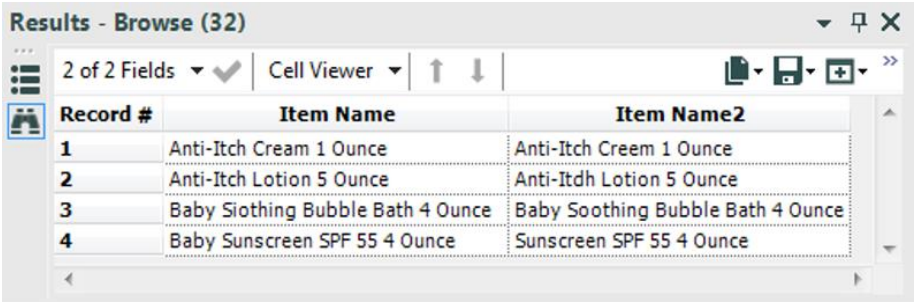


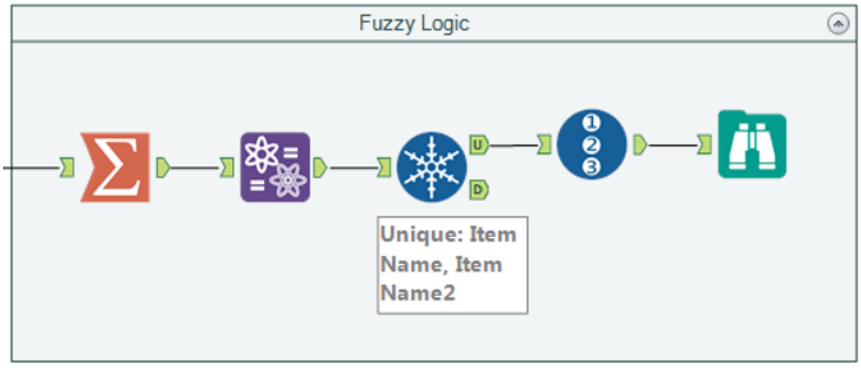
Figure 7-30 – Expensive Beauty Products Unique configuration

A quick run using the *Browse* tool shows us the matched items.



Record #	Item Name	Item Name2
1	Anti-Itch Cream 1 Ounce	Anti-Itch Creem 1 Ounce
2	Anti-Itch Lotion 5 Ounce	Anti-Itdh Lotion 5 Ounce
3	Baby Siothing Bubble Bath 4 Ounce	Baby Soothing Bubble Bath 4 Ounce
4	Baby Sunscreen SPF 55 4 Ounce	Sunscreen SPF 55 4 Ounce

Figure 7-31 – Expensive Beauty Products
Viewing matching items



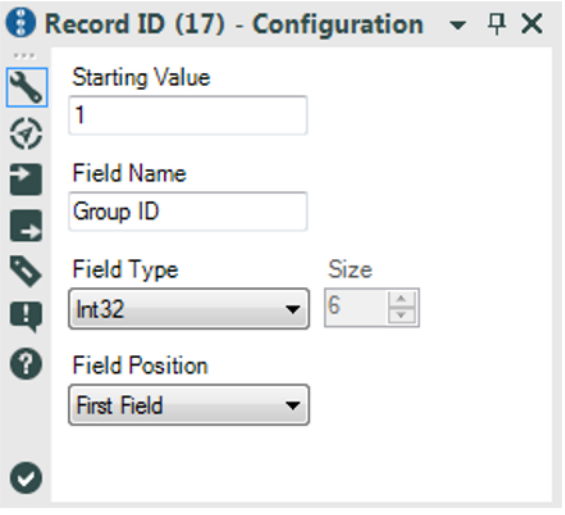


Figure 7-32 – Expensive Beauty Products Record ID for matched items

We can see by looking at the resulting data stream that we have four groups of items, and three of those are a result of misspellings in the data.

What we can do is provide a numeric grouping for each of these matches, which will solve Jan's problem of data comparison. Since each of the groups is unique, we can use the *Record ID* field to identify them.

Now we can add a comment that will allow Jan to understand what we did, and then we will be ready to combine the data streams.

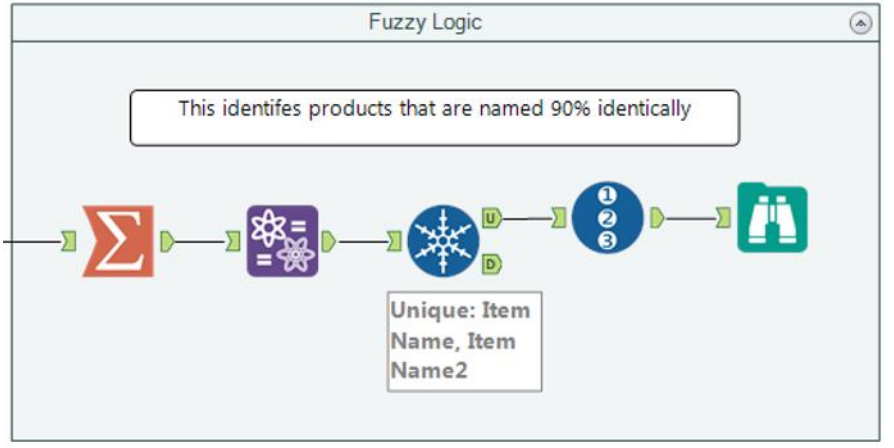


Figure 7-33 – Expensive Beauty Products
Adding comment to indicate the process

In order to combine the data streams, we need to join the data twice so that each part of the group can be flagged with the *Group ID*.

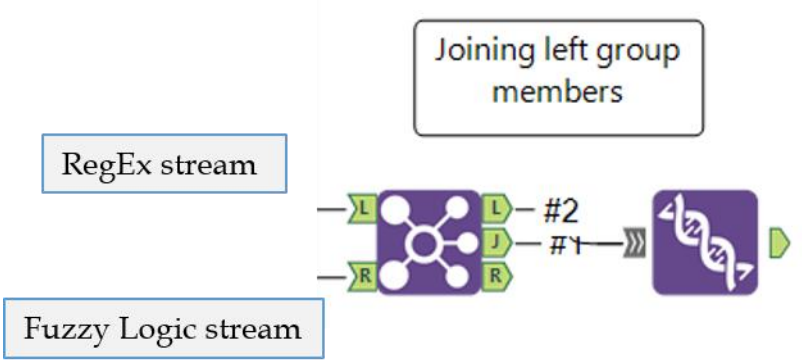


Figure 7-34 – Expensive Beauty Products
First Join for left group members

The *Join* tools settings are shown below.

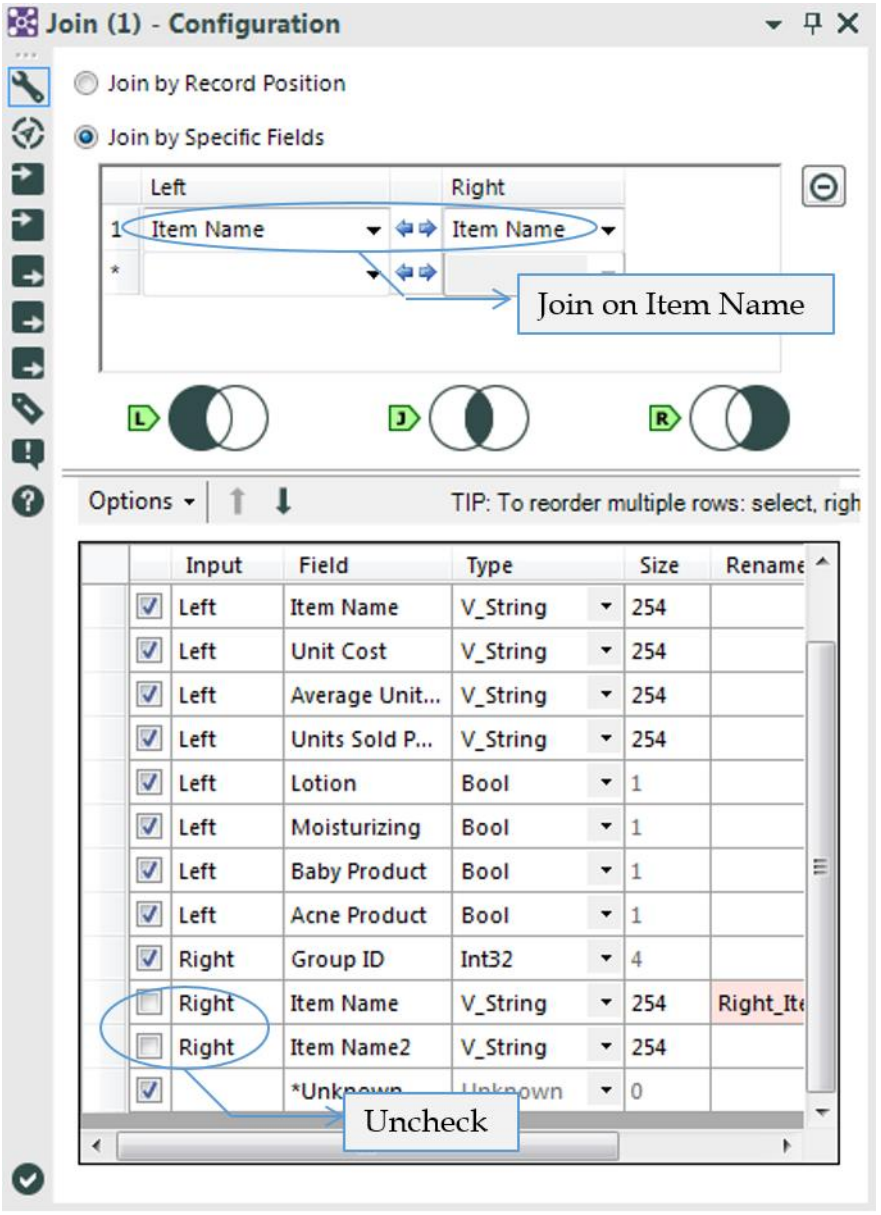


Figure 7-35 – Expensive Beauty Products
Join configuration for "Item Name"

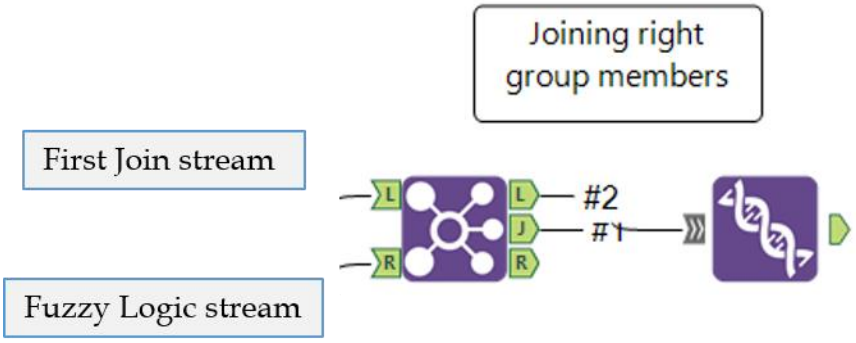


Figure 7-36 – Expensive Beauty Products
Second Join for right group members

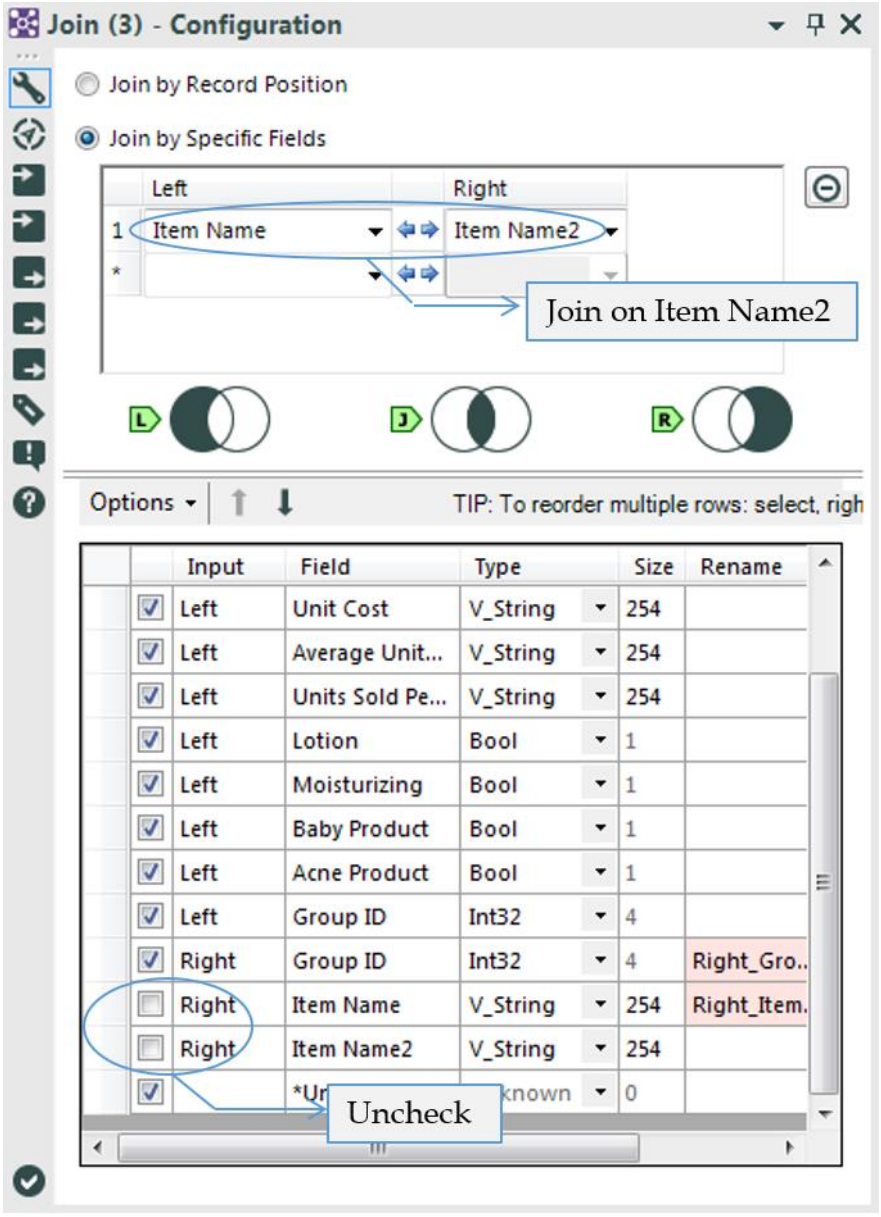
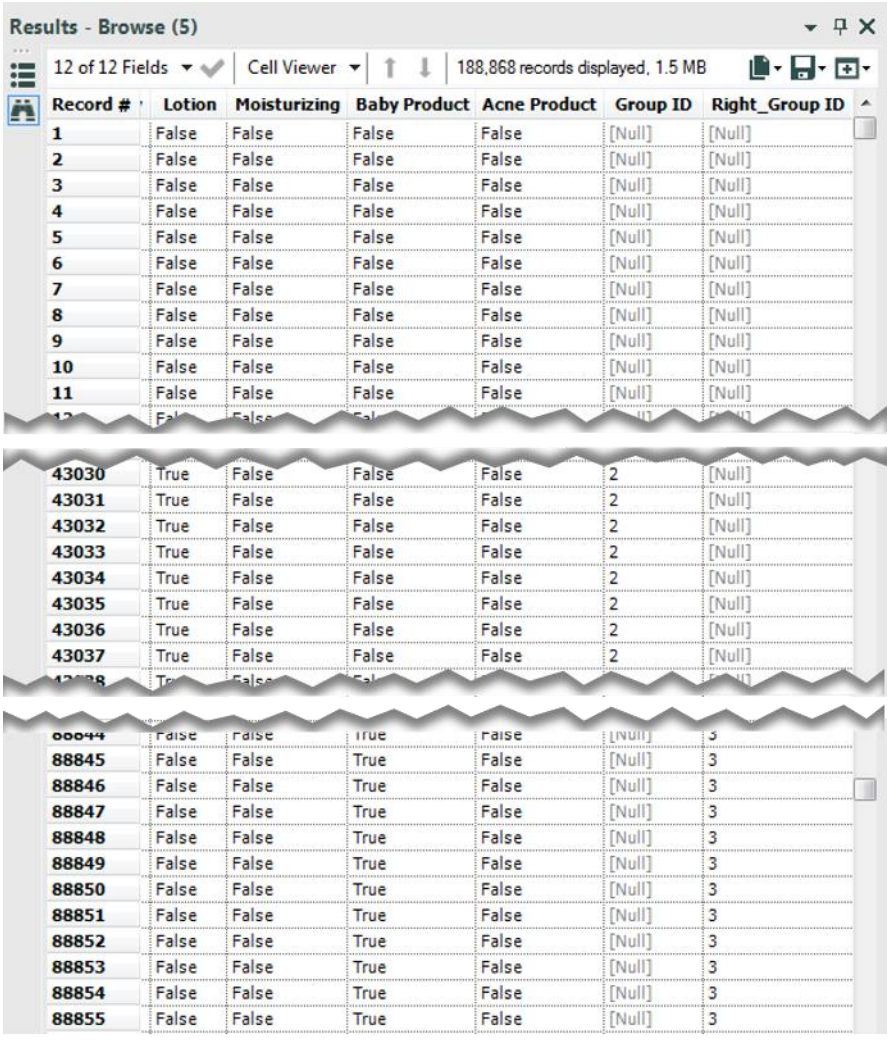


Figure 7-37 – Expensive Beauty Products
Join configuration for "Item Name2"



Record #	Lotion	Moisturizing	Baby Product	Acne Product	Group ID	Right_Group ID
1	False	False	False	False	[Null]	[Null]
2	False	False	False	False	[Null]	[Null]
3	False	False	False	False	[Null]	[Null]
4	False	False	False	False	[Null]	[Null]
5	False	False	False	False	[Null]	[Null]
6	False	False	False	False	[Null]	[Null]
7	False	False	False	False	[Null]	[Null]
8	False	False	False	False	[Null]	[Null]
9	False	False	False	False	[Null]	[Null]
10	False	False	False	False	[Null]	[Null]
11	False	False	False	False	[Null]	[Null]
43030	True	False	False	False	2	[Null]
43031	True	False	False	False	2	[Null]
43032	True	False	False	False	2	[Null]
43033	True	False	False	False	2	[Null]
43034	True	False	False	False	2	[Null]
43035	True	False	False	False	2	[Null]
43036	True	False	False	False	2	[Null]
43037	True	False	False	False	2	[Null]
88845	False	False	True	False	[Null]	3
88846	False	False	True	False	[Null]	3
88847	False	False	True	False	[Null]	3
88848	False	False	True	False	[Null]	3
88849	False	False	True	False	[Null]	3
88850	False	False	True	False	[Null]	3
88851	False	False	True	False	[Null]	3
88852	False	False	True	False	[Null]	3
88853	False	False	True	False	[Null]	3
88854	False	False	True	False	[Null]	3
88855	False	False	True	False	[Null]	3

Figure 7-38 – Expensive Beauty Products Browse configuration after Joins

If we applied the previous settings along with the default *Union* settings, we should see that we have two new fields at the end of the data stream. We need to combine these fields in order to give Jan a simple dataset to work from.

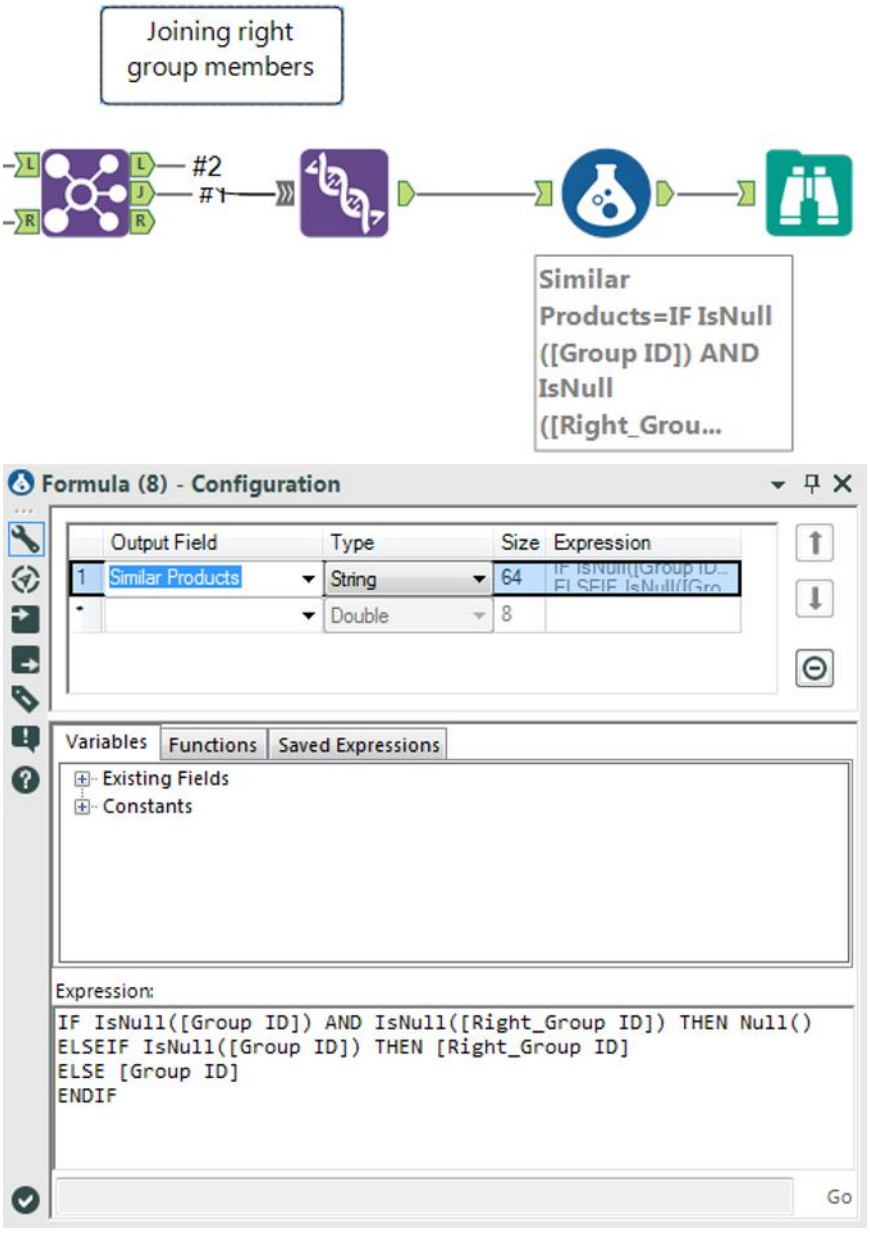


Figure 7-39 – Expensive Beauty Products Formula to collate fields

We can create a conditional formula called *Similar Products* that will bring the fields together:

IF IsNull([Group ID]) AND IsNull([Right_Group ID]) THEN Null()

ELSEIF IsNull([Group ID]) THEN [Right_Group ID]

ELSE [Group ID]

ENDIF

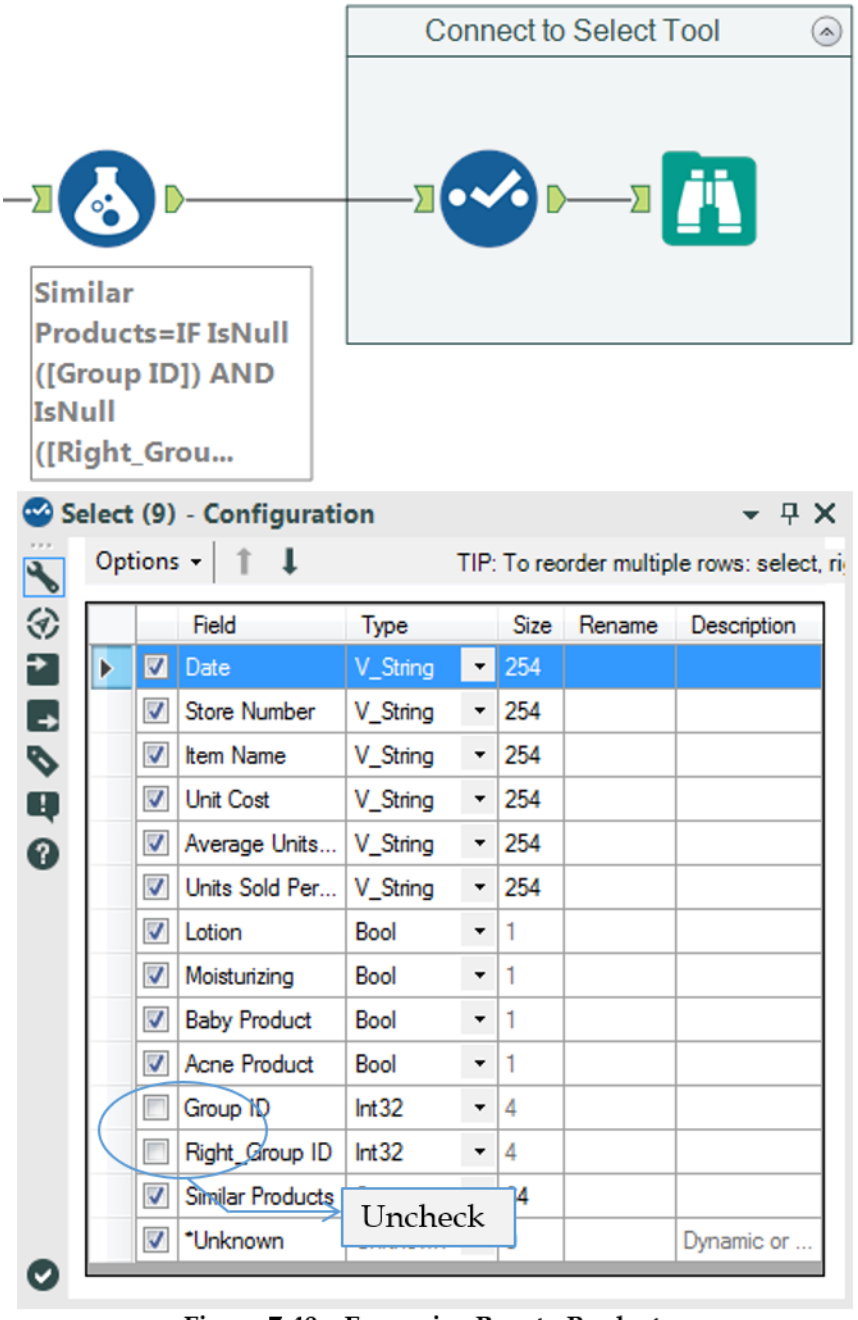


Figure 7-40 – Expensive Beauty Products
Select fields for final output

In order to finish this data stream, we can add a select statement that has the two *Group ID* fields removed. If we then use a *Tool* container, we can very clearly show where any downstream tools should be connected.

The *Expensive Beauty Products* data stream should look like the following image when it is complete.

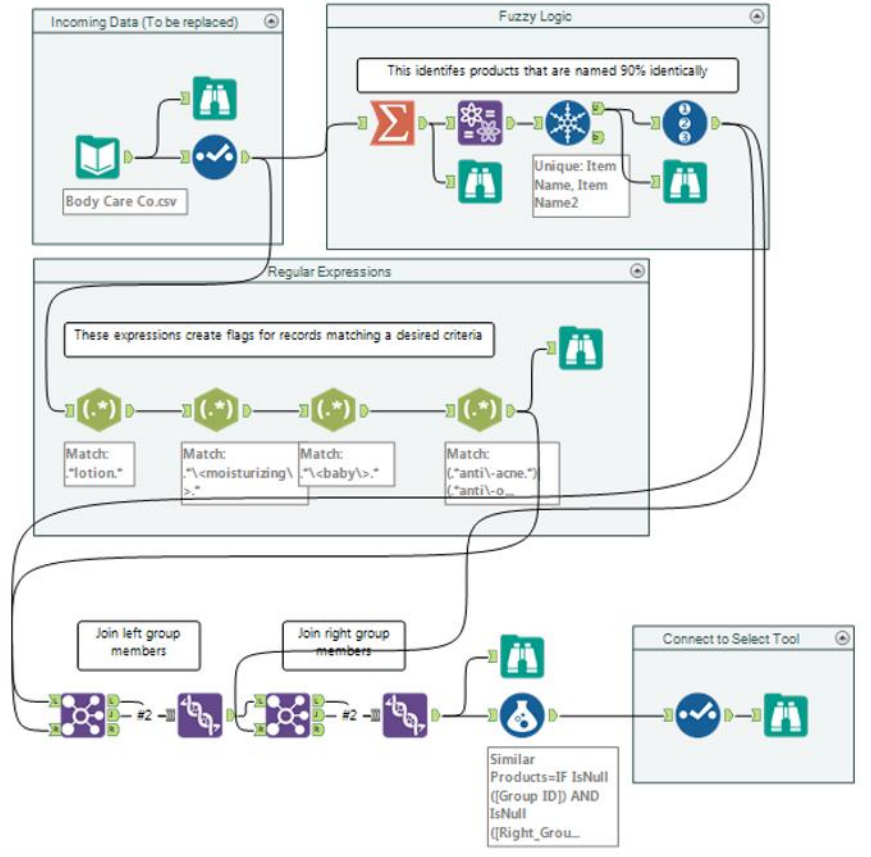
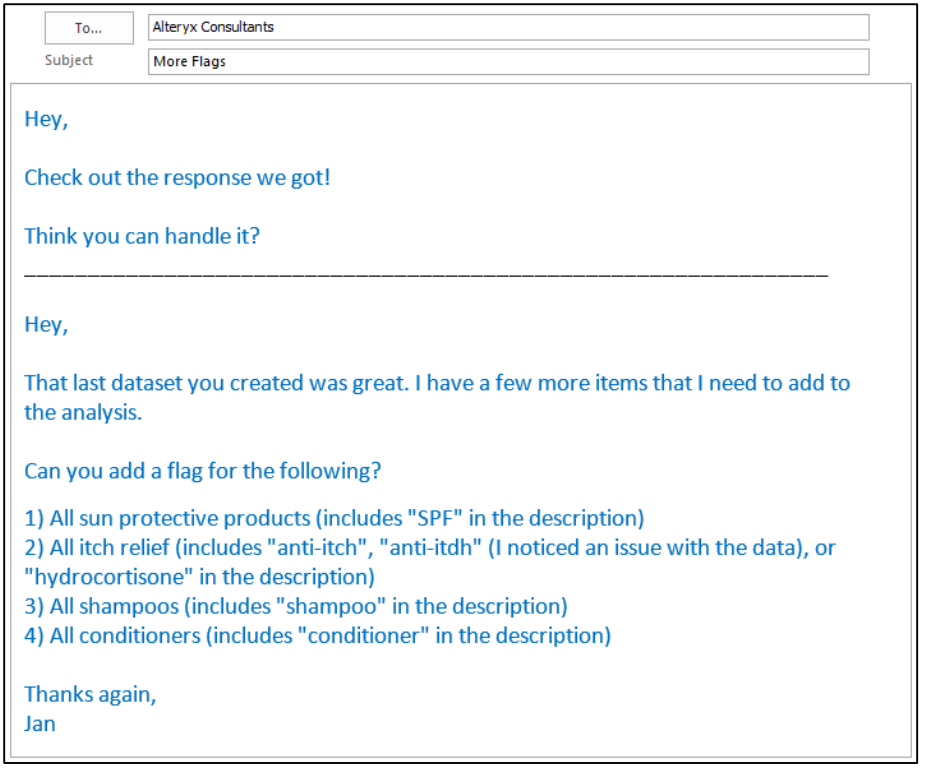
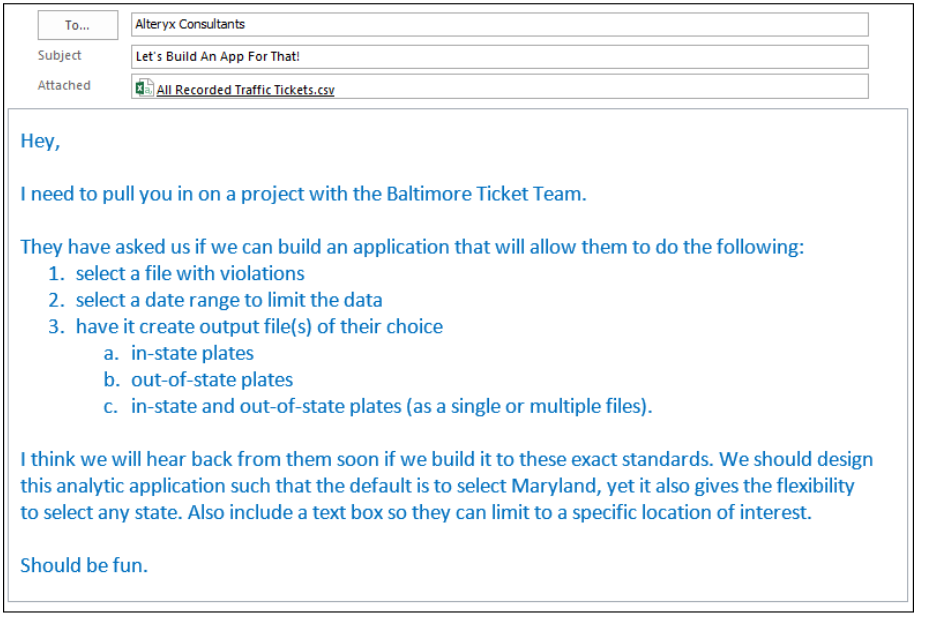


Figure 7-41 – Expensive Beauty Products Complete data stream

7.8 More Flags



Chapter 8
Applications Wanted


To...
Alteryx Consultants
Subject
Let's Build An App For That!
Attached
All Recorded Traffic Tickets.csv
Hey,
I need to pull you in on a project with the Baltimore Ticket Team.
They have asked us if we can build an application that will allow them to do the following:
1. select a file with violations
2. select a date range to limit the data
3. have it create output file(s) of their choice
a. in-state plates
b. out-of-state plates
c. in-state and out-of-state plates (as a single or multiple files).
I think we will hear back from them soon if we build it to these exact standards. We should design this analytic application such that the default is to select Maryland, yet it also gives the flexibility to select any state. Also include a text box so they can limit to a specific location of interest.
Should be fun.

8.1 Tools & Concepts

In this chapter, we will be covering the following Tools and Concepts:

Tools	Concepts
Action	Building Applications
Condition	
Date	
Date Filter	
Drop Down	
File Browse	
Radio Button	
Text Box	

8.2 Action

Figure 8-1 – Action

The *Action* tool modifies the values in other tools.

Group	Input	Output
Interface	None	None

Application questions can be connected to the top *Black Question Anchor*.

A condition tool can be connected to the *Diamond Anchor*.

The *Lightning Bolt Anchor* can be connected to a tool to update that tool based on the *Action* tool's settings.

Action tools are special tools that update settings for the connected tools in the workflow.

Properties Window:

The *Action Configuration* window has only one component.

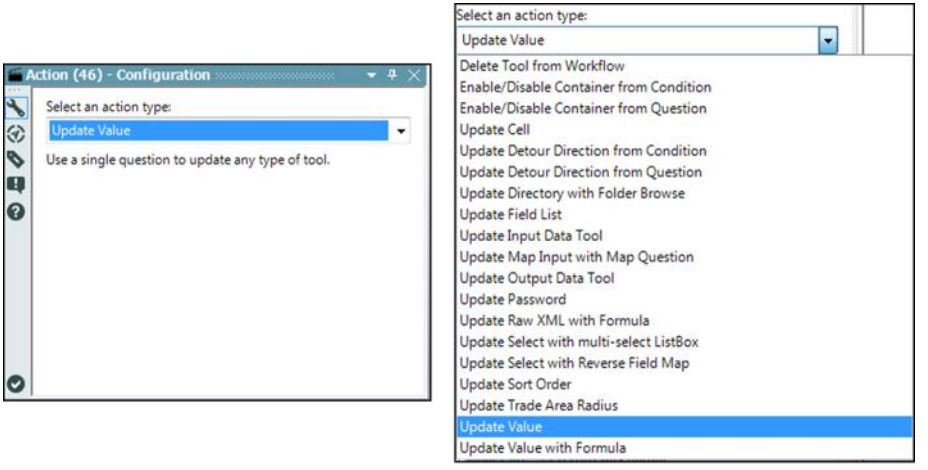


Figure 8-2 – Action configuration with list of Action types

Selecting an action type allows us to configure the changes in the tool connected to the action.

The list of options appear depending on the tool that is connected, and we can see a list of tools that behave in the same way as seen in Appendix J.

The image is a comprehensive list of action types. The methods are self-explanatory, and we will almost always use the default action associated with each tool.

8.3 Condition

Figure 8-3 - Condition

The *Condition* tool allows us to trigger different results depending on the condition in the tool.

Group	Input	Output
Interface	None	None

At least one *Action* tool should always follow this tool.

Multiple Application questions can be connected to the top *Black Question Anchor*.

Diamond Anchor T connects to an Action tool and provides True result if the condition is true.

Diamond Anchor F connects to an Action tool and provides False result if the condition is false.

Properties Window:

The *Condition Configuration* window allows us to create a Boolean expression using the connected questions.

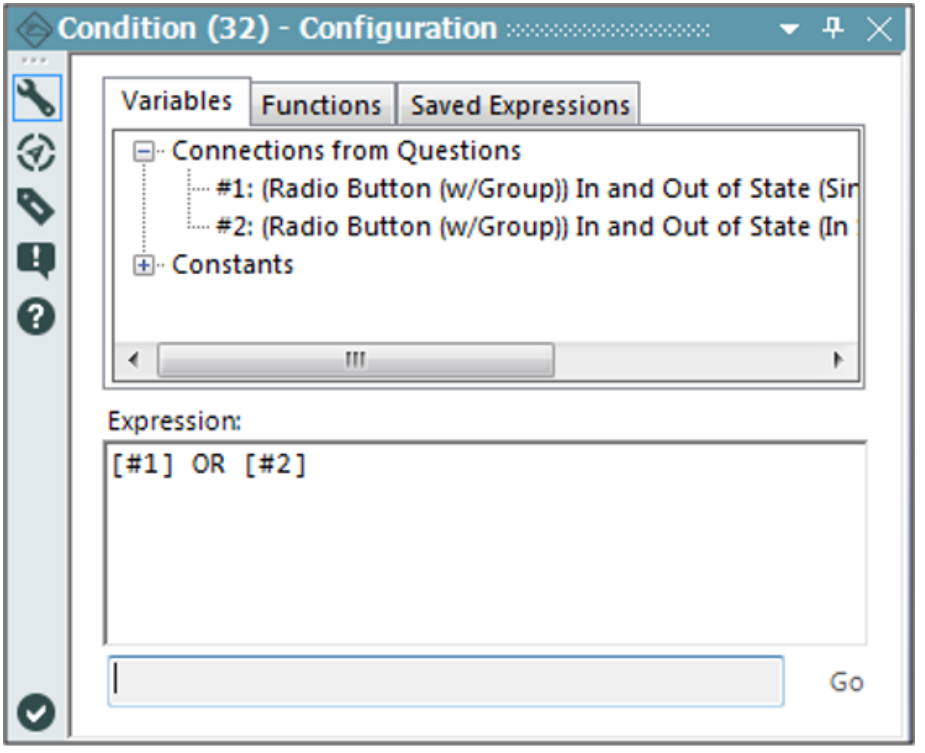


Figure 8-4 - Condition Configuration

In this case, since connections 1 and 2 are both *Radio Button* tools, their values are either true or false.

The expression is true if either of these options were selected.

- If the expression is true, then the true (T) path has a true value passed down it, and the false (F) path has a false value passed down it.

- If the Expression is false, then the true (T) path has a false value passed down it, and the false (F) path has a true value passed down it.

8.4 Date

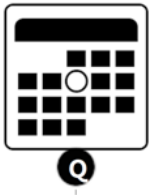


Figure 8-5 - Date

The *Date* tool allows for date selection from a calendar menu.

Group	Input	Output
Interface	None	None

The *Black Question Anchor* connects the results to other tools, which have incoming anchors.

The *Date* tool creates a string of the selected date in the format YYYY-MM-DD.

Properties Window:

The *Date Configuration* window has only one component.

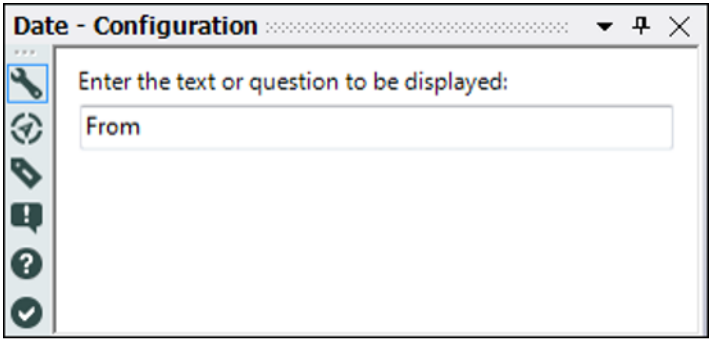


Figure 8-6 – Date Configuration

Enter the text or question to be displayed allows us to write the prompt that the end user sees.

8.5 Date Filter

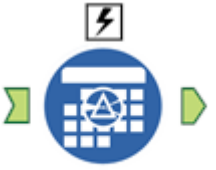


Figure 8-7 – Date Filter

The *Date Filter* tool provides the ability to filter based on dates.

Group	Input	Output
Preparation	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with at least one date field.

Output: The input data stream limited to appropriate date range.

Properties Window:

The *Date Filter Configuration* window has two components.

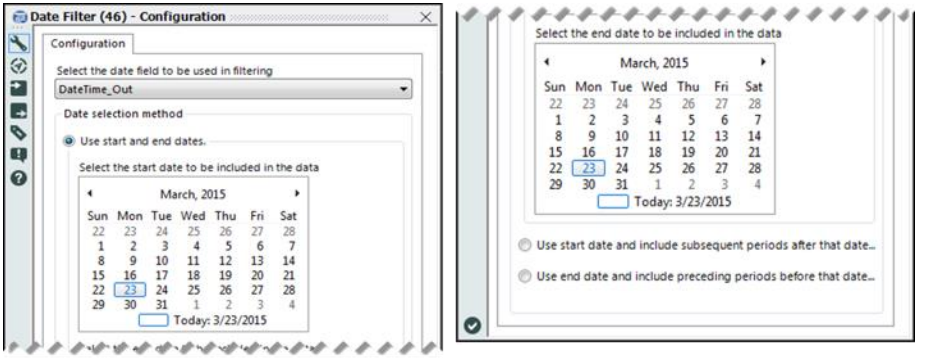


Figure 8-8 - Date Filter Configuration

- The *Select the date field to be used in filtering* allows in selecting the date field in the incoming data source.
- *Date selection method* allows us to define which method the date filter will use to limit the data.
- *Select the start and end dates* allows us to select two dates on a calendar and only keep the dates restricted between them.

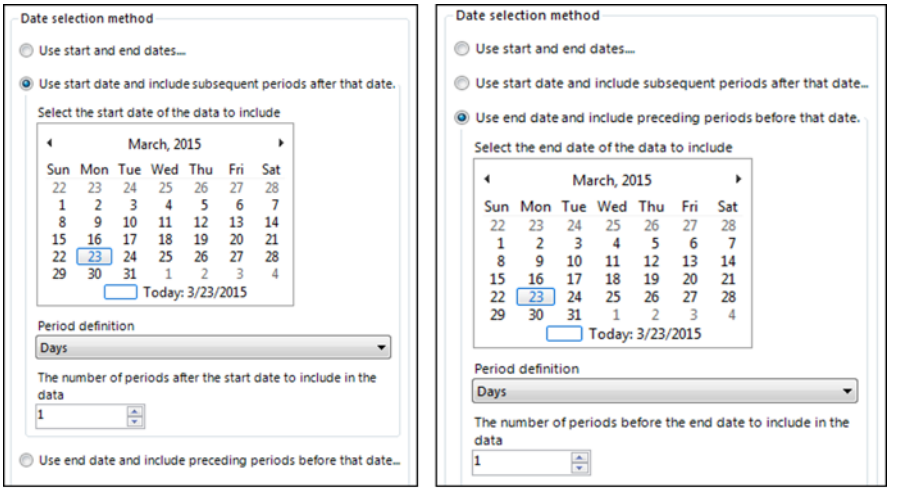


Figure 8-9 - Date Filter Configuration - Using Start and End Dates

- *Use start date and include subsequent periods after that date* allows us to select a date to start from and then define how long after that date we want to include.
- *Use end date and include preceding periods before that date* allows us to select a date to end the data and then define how long before that date we want to include.

8.6 Drop Down

Figure 8-10 – Drop Down

The *Drop Down* tool allows for the selection of a single value from a provided list.

Group	Input	Output
Interface	None	Data stream

The *White Question Anchor* receives information from other tools to populate the list.

The *Black Question Anchor* connects the results to other tools that have incoming anchors.

The result of this question is the string of the selected option.

Properties Window:

The *Drop Down Configuration* window has three basic components.

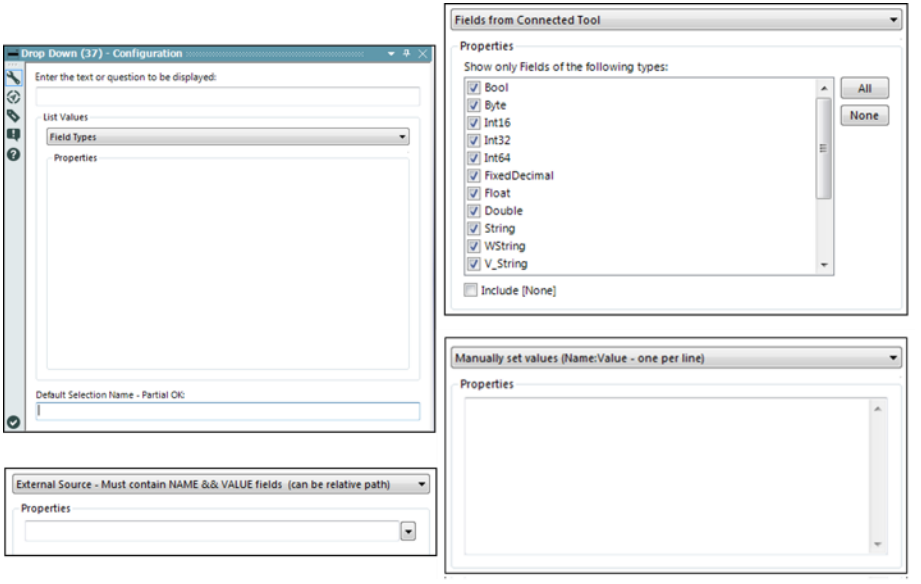


Figure 8-11 - Drop Down Configuration

- *Enter the text or question to be displayed* allows us to write the prompt that the end user sees.
- *List Values* allows us to select a method of populating a list and any related properties.
- *Default Selection Name – Partial OK* allows us to set a default value by typing in a unique part of the option.
- *Field Types* are the list of field types.
  - *External Source – Must contain NAME & VALUE fields (can be relative path)* uses a reference file that has both a Name and a Value field.

- *Field from Connected Tool* allows us to connect to a tool and select one of the fields from that data connection, as long as they match the file type in the multi-select list.
- *Manually set Values (Name: Value – one per line)* allows us to create a list of options in the properties box. The format for this is: *Text to display: Resulting value in data*.

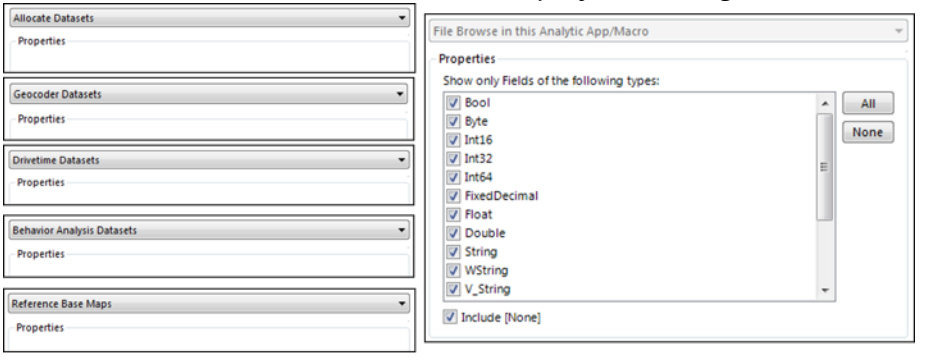


Figure 8-12 - Dataset types

- *Allocate Datasets* allows us to select from the Allocate dataset list we have available.
- *Geocoder Datasets* allows us to select from the Geocoder dataset list we have available.
- *Drivetime Datasets* allows us to select from the Drivetime dataset list we have available.
- *Behavior Analysis Datasets* allows us to select from the behavior dataset list we have available.
- *Reference Base Maps* allows us to select from the base maps we have available.
- *File Browse in this Analytic App/Macro* connects to a file browse tool and allows for selection of one of the fields from that data connection, as long as they match the file

type in multi-select list. *Include [None]* means we are providing an option that will result in no selection.

8.7 File Browse

Figure 8-13 – File Browse

The *File Browse* tool allows to connect to a file of choice instead of a predesignated file.

Group	Input	Output
Interface	None	None

The *Black Question Anchor* connects the results of questions to anchors on other tools.

The tool can be used to update any tool that used a file connection. When using this tool, the end user should know what information should be included in the data so the module runs correctly.

Properties Window:

The *File Browse Configuration* window has three components.

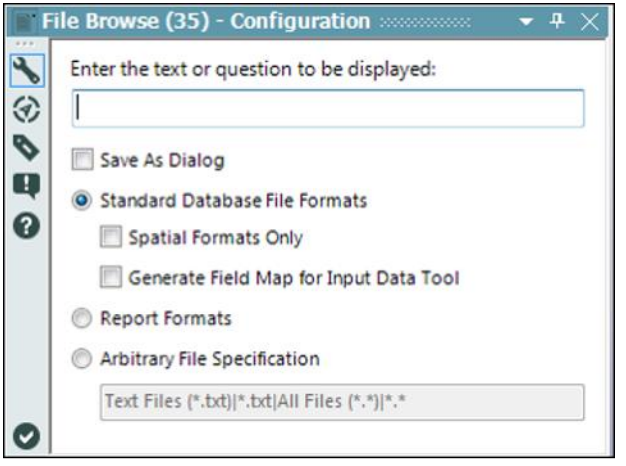


Figure 8-14 - File Browse Configuration

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.
- *Save As Dialog* allows us to output the data as a dialog box instead of a file.
- *Standard Database File Formats* are a typical file browse.
    - *Spatial Formats Only* will allow only file types that can have spatial objects.
    - *Generate Field Map for Input Data* forces the end user to align the fields in the selected data source with the fields in the development data source.
- *Report Formats* option allows us to select only report formats that are compatible with the *Render* tool.
- *Arbitrary File Specification* allows us to define specific file formats that we want to be accessed or preferred.

8.8 Radio Button

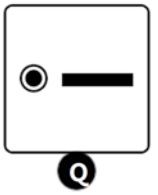


Figure 8-15 - Radio Button

The *Radio Button* tool creates a single select option for use in the *Interface designer*.

Group	Input	Output
Interface	None	None

The *Black Question Anchor* connects the results of questions to anchors on other tools.

The question creates a Boolean. It will be a single select option along with all other single select options in the group (all, if no groups are defined).

Properties Window:

The *Radio Button Configuration* window has three components.

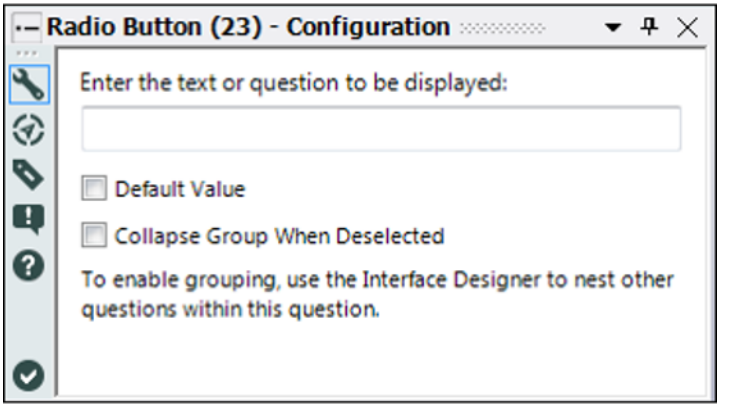


Figure 8-16 - Radio Button Configuration

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.
- *Default Value*, when selected, will use this question as the default selected.
  Best practice is to make sure this question is first in the list of values.
- *Collapse Group When Deselected* allows us to create a group of questions that only display when the question is selected.

8.9 Text Box

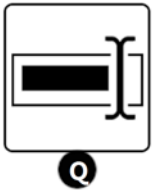


Figure 8-17 - Text Box

The *Text Box* tool allows the user to enter a string of choice.

Group	Input	Output
Interface	None	None

The *Black Question Anchor* connects the results of questions to anchors on other tools.

The questions produce a string output.

Properties Window:

The *Text Box Configuration* window has five components.

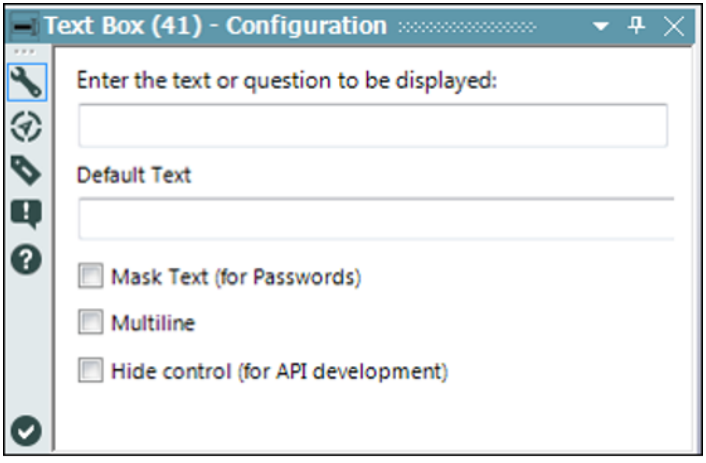
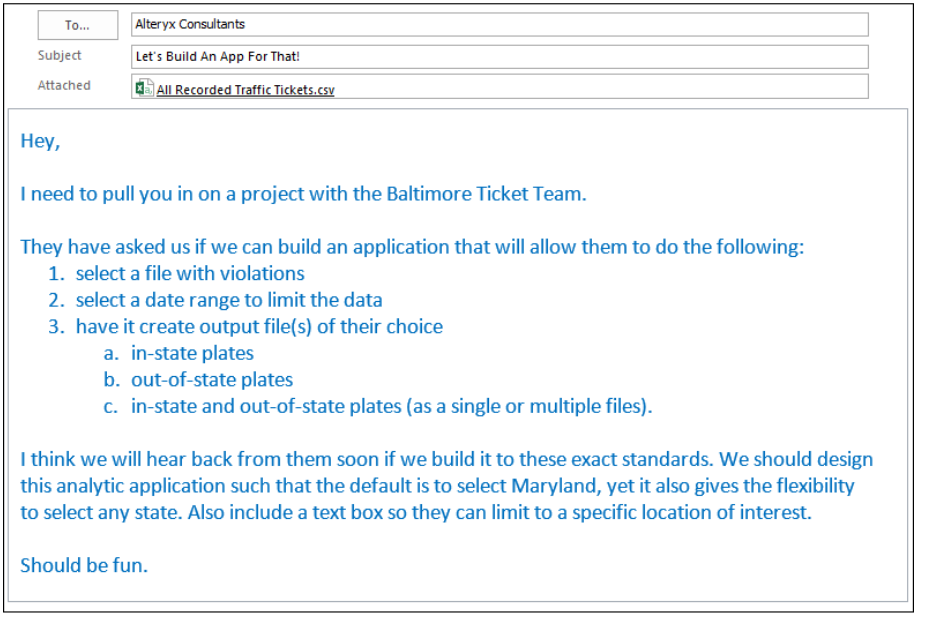


Figure 8-18 - Text Box Configuration

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.
- *Default Text* is a string that will populate the text box.
- *Mark Text (for Passwords)*, when checked, replaces all visible text characters in the response with "*".
- *Multiline*, when checked, allows the end user to create a multiple line input and makes the text entry larger.
- *Hide control (for API development)* gives the ability to hide the entry field from the interface designer but allows the API to continue to edit the entry.

8.10 Let's Build an App For That!


To... Alteryx Consultants
Subject Let's Build An App For That!
Attached All Recorded Traffic Tickets.csv

Hey,

I need to pull you in on a project with the Baltimore Ticket Team.

They have asked us if we can build an application that will allow them to do the following:

1. select a file with violations
2. select a date range to limit the data
3. have it create output file(s) of their choice
    a. in-state plates
    b. out-of-state plates
    c. in-state and out-of-state plates (as a single or multiple files).

I think we will hear back from them soon if we build it to these exact standards. We should design this analytic application such that the default is to select Maryland, yet it also gives the flexibility to select any state. Also include a text box so they can limit to a specific location of interest.

Should be fun.

Alteryx allows us to develop Analytic Applications that make it so the workflow can be modified by user inputs. Apps have a couple of special properties that allow us to:

- Use an interface to run the workflow instead of opening the file in Alteryx Designer.
- Publish the workflow to an Alteryx Gallery, giving users without Alteryx Designer the ability to run workflows.

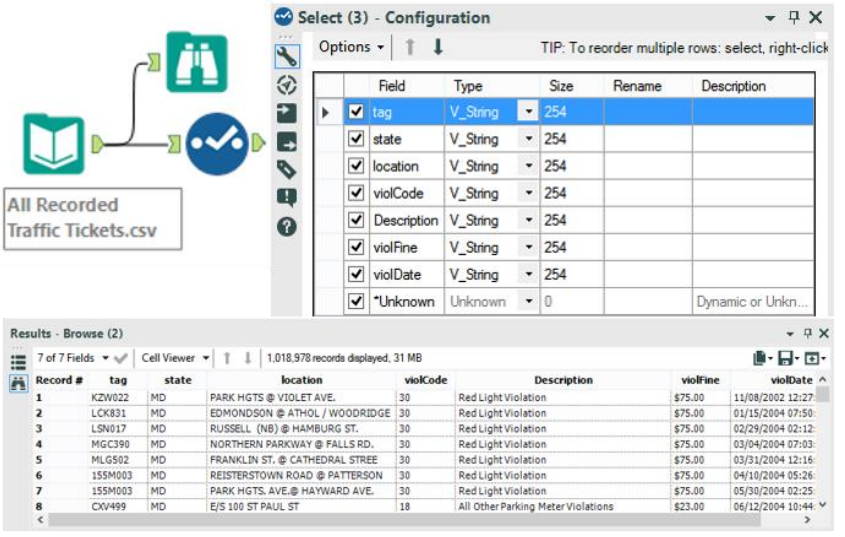


Figure 8-19 – Traffic Tickets – Opening, selecting and viewing data

We are going to approach this by laying out the workflow and then adding the interface tools that allow for the modification. So, let us get started by connecting to the file *All Recorded Traffic Tickets.csv*. Since they haven't asked us to do anything with the Fine amount field, we can leave it in a string format.

However, since we know we need to limit the data by dates, we are going to want the *violDate* in a date format. If we take a look at the format in *violDate*, we see it is a string-formatted date with an *AM/PM* flag. If we look at the *DateTime*, tool we can see that we do not have a matching date format, so we need to use a formula tool to manipulate the string first.

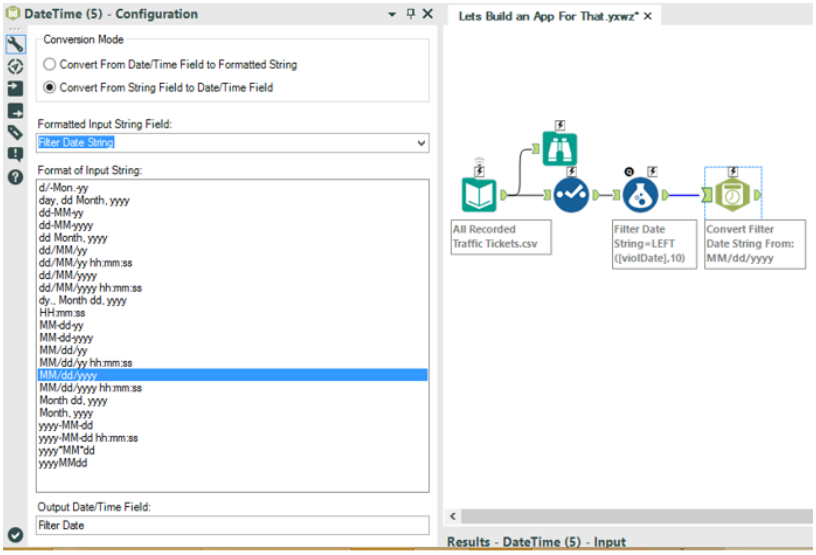


Figure 8-20 – Traffic Tickets – Conversion from string to date

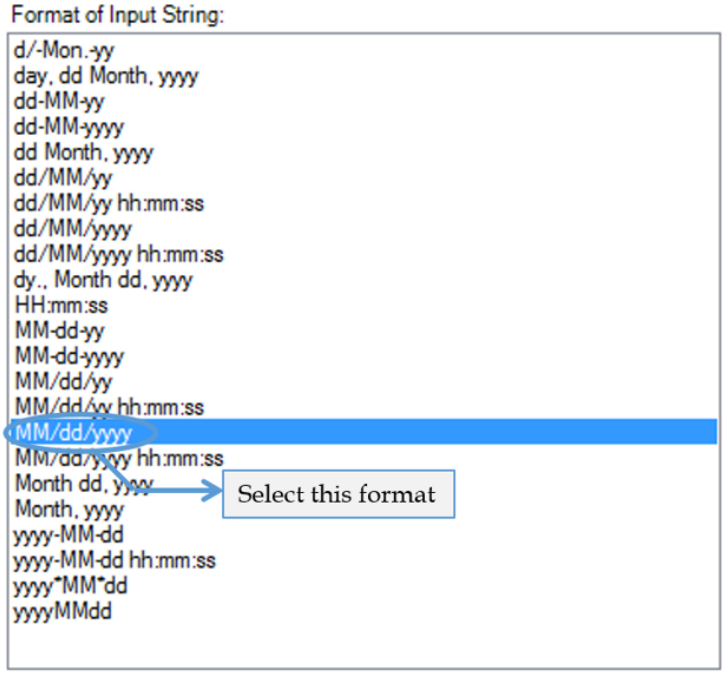


Figure 8-21 - Traffic Tickets – Input String Format

Since we are only going to allow the end user to select dates, we can drop the time portion of the field. This means that we only need the first 10 characters. The formula is: *LEFT([violDate],10)*

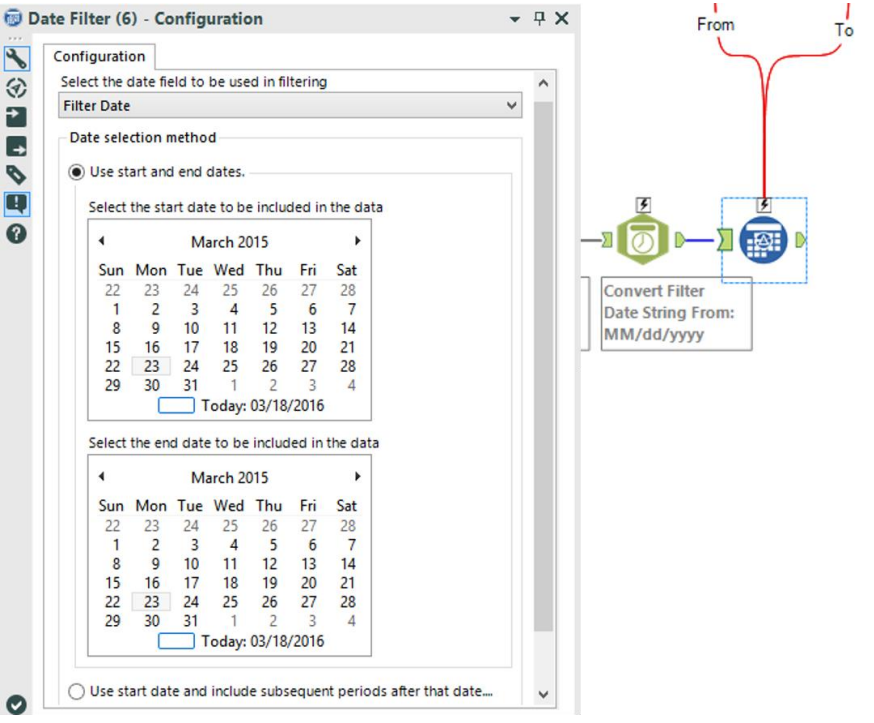


Figure 8-22 – Traffic Tickets – Using Date filter tool

Now that we have a field in date format, we can create the *Date Filter* tool. Because we only have one date type field, *Filter Date* populates in the field selection. We are going to allow the end user the ability to select the beginning and end date, so we can select that option and leave all other settings alone. It does not matter that both dates are referring to today because we are going to replace these values.

Note: While we are testing, it may make it hard to see what is going on if we do not select a date range that is reflected in the data.

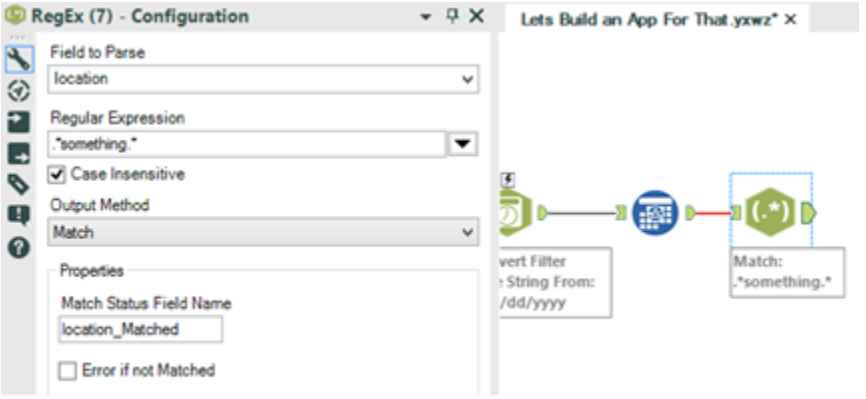


Figure 8-23 – Traffic Tickets – Using RegEx to flag string

Now we can add the feature they didn't ask for. We can write a regular expression that allows us to capture partial or full location names to limit the data. We want to look at the location field and then create a regular expression that will tell us if the field matches the user input. We can use the following expression along with a *text box interface* tool to do this.

.*something.*

Using the above expression, we will update *something* with the user-specified value so we can flag the appropriate records. (If the user wants everything, they can leave the field blank resulting in the *expression* .*.*, which will mark all fields as true.)

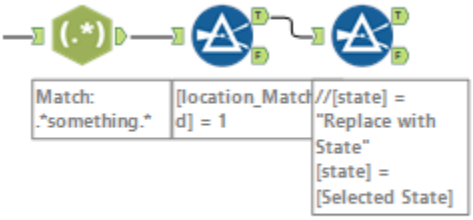


Figure 8-24 – Traffic Tickets – match with flagged string

Now that we have flagged which records match the desired location, we can use the *location_Matched* field that was generated in the *RegEx* tool to filter the data. If we use the basic filter for the *location_Matched,* we can see the field is true and the function becomes the expression *location_Matched* = 1.

Now that we have limited the data to the appropriate locations, we can flag which state they may want to analyze in detail. For now, we can use a custom filter expression in order to finish building the core data stream.

[state] = "Replace with State"

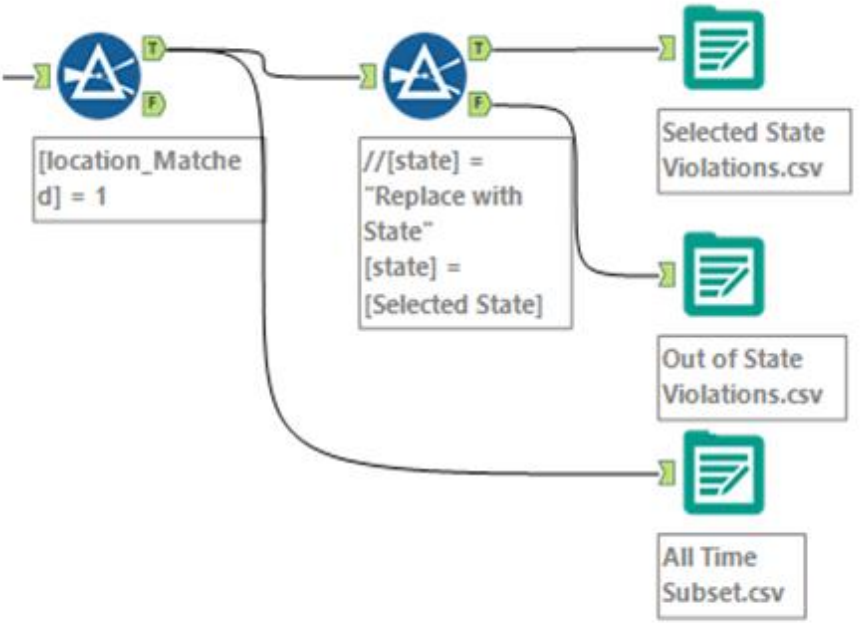


Figure 8-25 – Traffic Tickets – getting three possible outputs

The last step in the workflow is to create the three possible outputs. We can add them to the end of the data stream where appropriate.

- Connect the *Selected State Violations.csv* to the True output on the state *filter*.
- Connect the *Non-Selected State Violations.csv* to the False output on the state *filter*.
- Connect the *All Matching Violations.csv* to the True output on the Location Match *filter*.

We can put the three outputs in *Tool Containers*. This is not for organizational purposes like we saw earlier. When we are building applications, we can enable and disable *Tool Containers* based on the user selection. In this case, we are going to disable

all three tool containers so the only time the output is created is when we have selected that option.

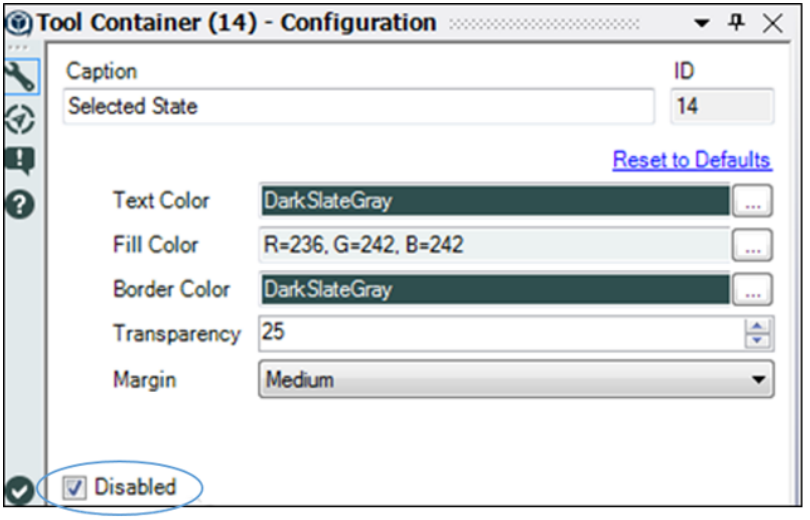


Figure 8-26 - Traffic Tickets – using Tool Container to disable it except when appropriate option is selected

We can disable the containers using the *Disabled* option in the bottom-left corner of the properties windows. This option only shows the container and not the contents and will turn off all the three outputs.

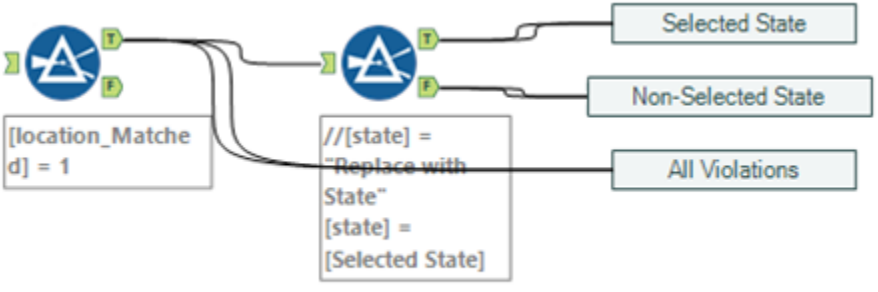


Figure 8-27 – Traffic Tickets – selected options

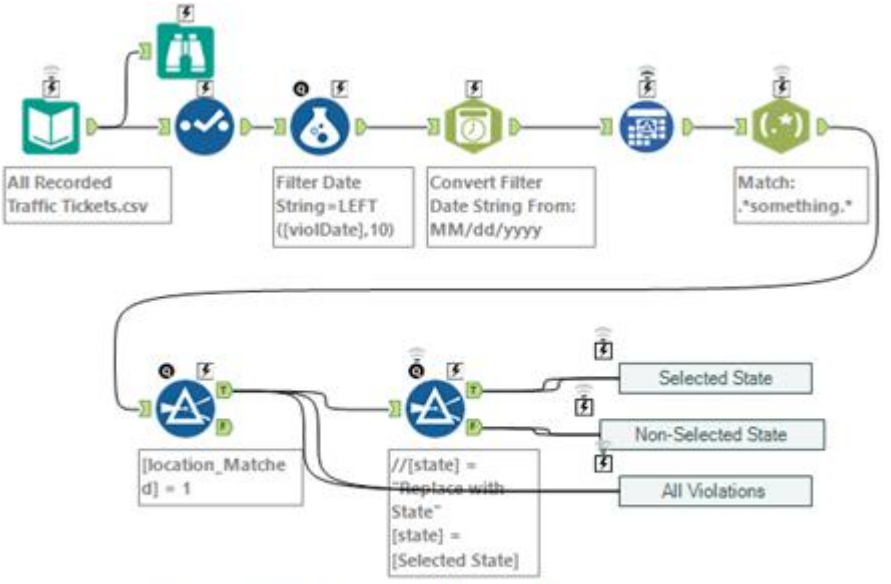


Figure 8-28 – Traffic Tickets – Basic data stream

We now have the basic data stream designed for the analytic app. We will have to modify some of the tools, but almost everything is ready for the transition. Let us look at the list of interactions we are providing the Baltimore Ticket Team.

1. Select files.
2. Choose the date range.
3. Name a location (optional).
4. Select a State (Maryland, by default).
5. Select which combination of the three output files is to be returned.

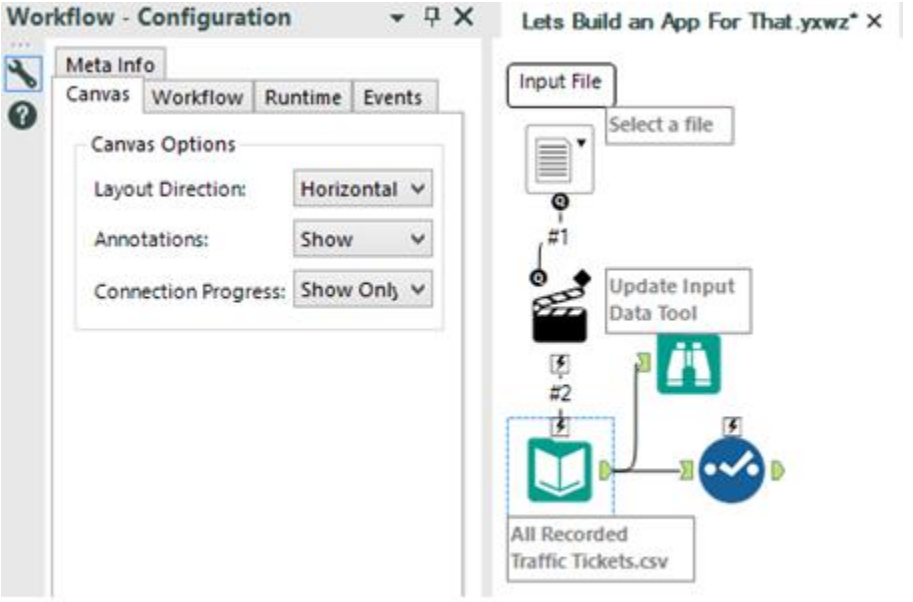


Figure 8-29 – Traffic Tickets - File Browse

The first question we can address is which file should be selected. If we drop a *File Browse* tool onto the canvas, we can see that the tools show their *lightning bolt* and *question mark* anchors indicating we are now working with an *Analytic Application or Macro*. We can then change the text so it reads *Select a file* and use the remaining default settings.

We know we want to connect the file selection question to the *Input Data* tool; however, the *lightning bolt anchor* only connects to *Action* tools. We could bring in the tool, but Alteryx has a clever way of handling it: When we connect a question directly to a *lightning bolt anchor,* it brings an *Action* tool onto the canvas with all of the default settings in place. In this case, it will do exactly what we want, and we can move onto the next question. (Click the Action tool On and Off to confirm the settings.)

Figure 8-30 – Traffic Tickets – using Action Tool for Date

The next question is slightly more complicated. We need to set the beginning and end dates for the selected timeframe. We can drag in two *Date* tools and label one *From* and the other *To* so we know which dates refer to the beginning and end of the range. This time, when we connect the questions to the *Date Filter* tool, there is no indication what value should be updated. What we want to do is update two values in the *Date Filter*. However, since there are multiple inputs, we need to select which value each action should update.

- From's *Action* tool should update the variable *start-end.start*
- To's *Action* tool should update the variable *start-end.end*

Now that we have this filter getting its values updated, we can move onto the next question.

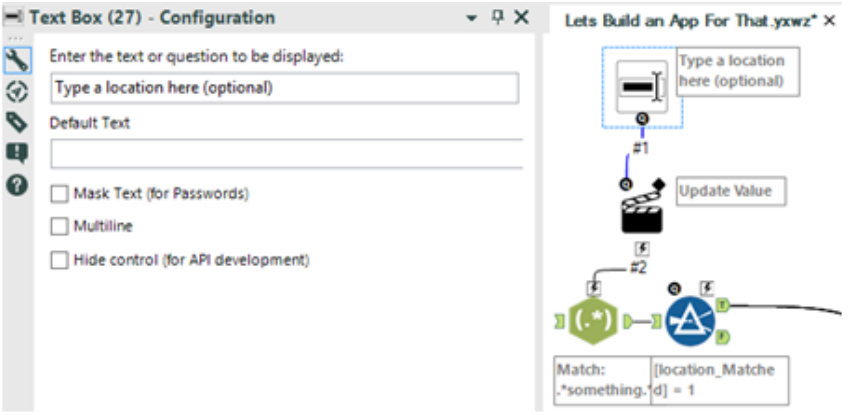


Figure 8-31 – Traffic Tickets – using Text Box to specify location

This time, we are going to add a *Text Box* tool with the prompt *Type a location here (optional)*. This will allow the end user to type anything they want to limit the locations required. We want this tool to modify the regular expression in the *RegEx* tool.

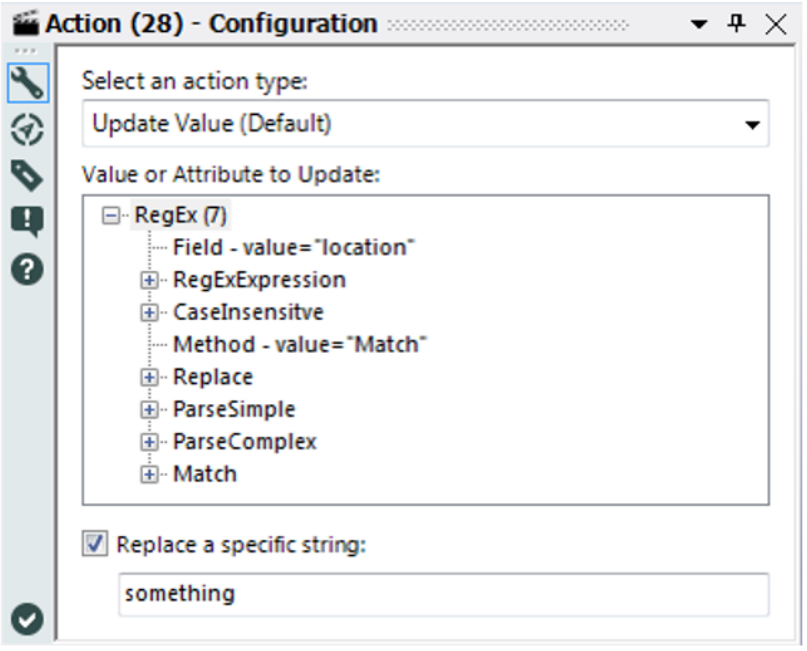


Figure 8-32 - Traffic Tickets – using Action with RegEx to specify string

We can see the *Action* tool is giving us an error, and if we take a look, we need to specify what should be updated. In this case, we know that in the *RegEx* tool, there is the expression string *.*something.** We want to replace *something* with the value the end user types in. Thus, we can use the *Replace a specific string* option to replace the string *something* within the regular expression. Now let's see how we can modify the state filter.

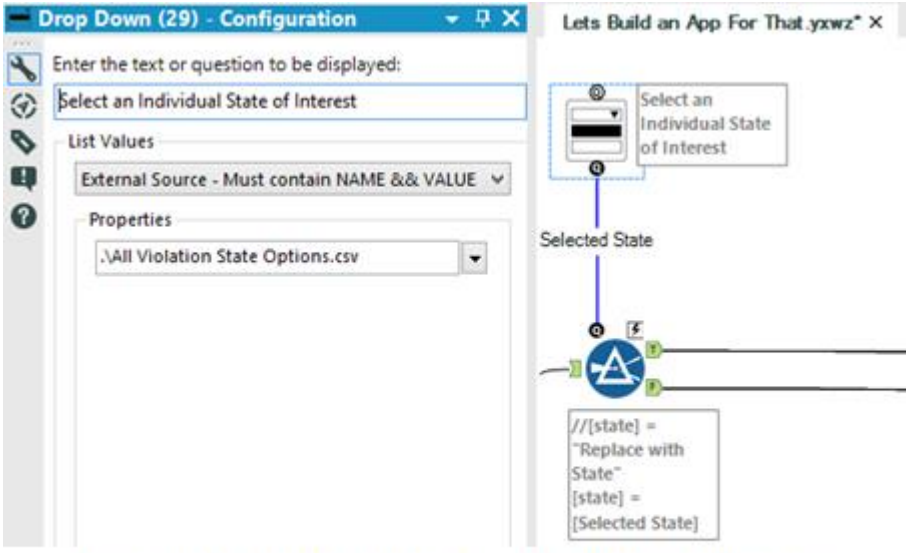


Figure 8-33 – Traffic Tickets – using Drop Down for selecting an individual State of Interest

This time, we are going to feed the answer of a question directly into a tool for use in the expression. We are going to create a *Drop Down* question with the prompt *Select an Individual State of Interest*. We are going to use the second file we received from the Baltimore Ticket Team to import the list of possible state codes from a file called *All Violation State Options.csv*. Finally, we are going to set the default value to *MD* so the team can quickly run the in-state analysis.

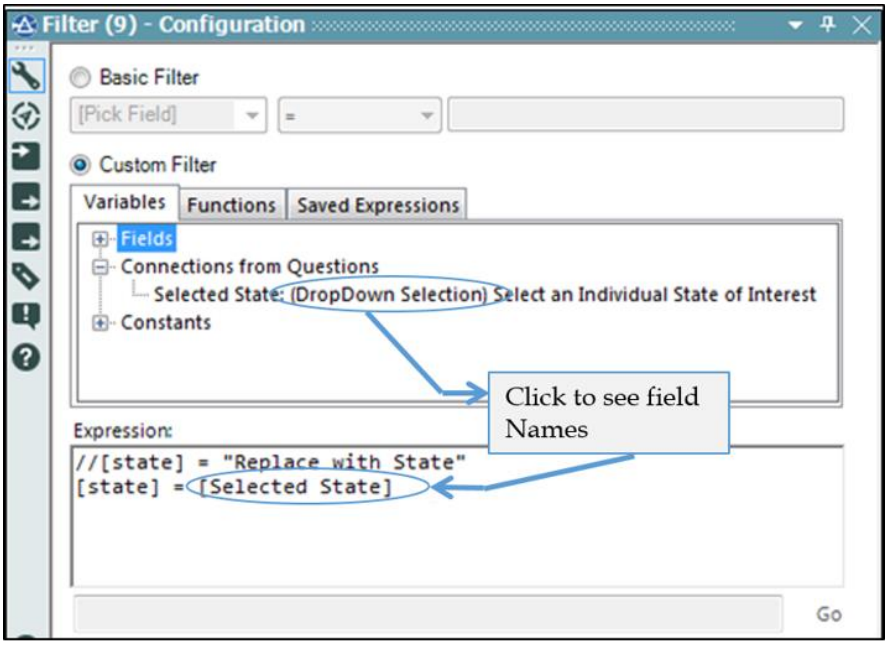


Figure 8-34 - Traffic Tickets – using Formula to filter the data

We are almost done with this question, except that we need to update the formula used to filter the data. We can replace the original formula with the one in the expression mentioned. (If we do not rename the connection, it should say [#1] instead of [Selected State].)

There is only one more question, but this one is the most complicated. This last question involves selecting which output(s) get created. Since the end user may want to output any combination of three different files (we will force them to select at least one), there are five different options we need to provide.

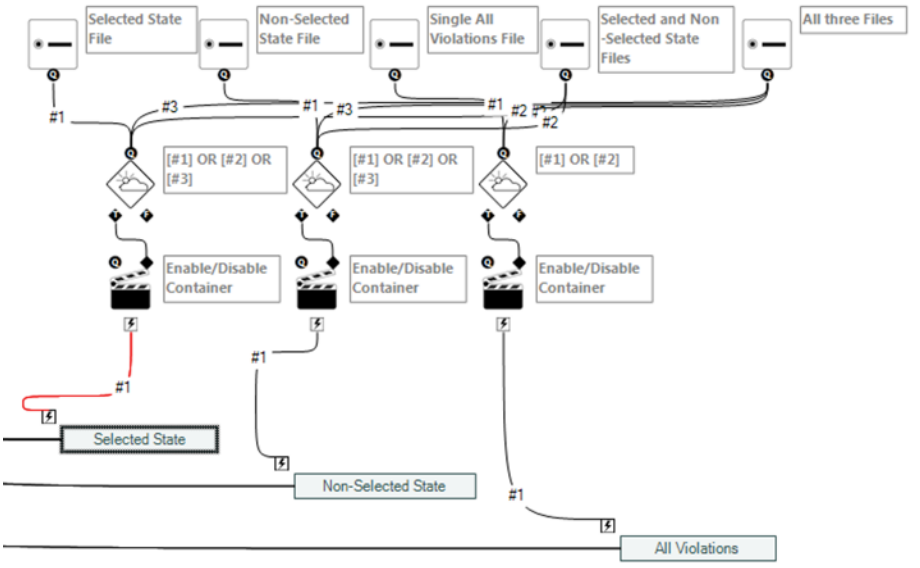


Figure 8-35 - Traffic Tickets – choosing from 5 different option

Create five Radio Button tools with the default settings other than the labels:

1. Selected State File
2. Non-Selected State File
3. All Violations File
4. Selected State and Non-Selected Files
5. All three Files

The goal here is to develop a structure where anytime we want a particular file created, we enable the *Tool Container(s)*. In order to activate the tool containers, we need *Action* tools connected to them that enable the container if the incoming connection is true.

Since multiple values can result in each file getting produced, we have complex conditions that need to be met. In

order to evaluate these conditions, we need to add a *Condition* tool before each *Action* tool.

Each of these Condition tools should be connected in the following way:

1. Condition connected to the Selected State Output
   - Selected State File (connection #1)
   - Selected and Non-Selected State Files (connection #2)
   - All three files (connection #3)
2. Condition connected to the Non-Selected State Output
   - Selected State File (connection #1)
   - Selected and Non-Selected State Files (connection #2)
   - All three files (connection #3)
3. Condition connected to the All Violations Output
   - All Violations File (connection #1)
   - All three Files (connection #2)

The expressions in each of the three conditions should be:

1. [#1] OR [#2] OR [#3]
2. [#1] OR [#2] OR [#3]
3. [#1] OR [#2]

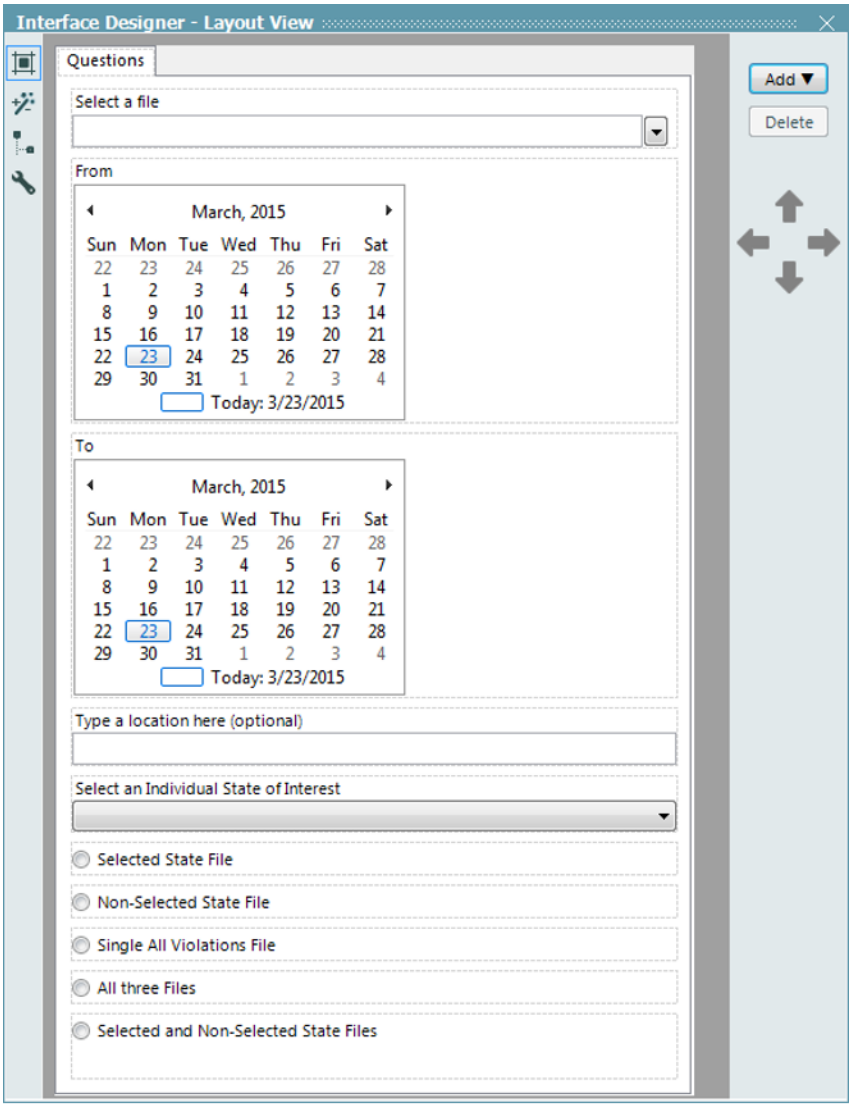


Figure 8-36 - Traffic Tickets – Interface designer

Now that we have the workflow developed, we can see the *Interface Design*. The interface is how the end users are going to interact with the workflow. As we can see from the previous page, the *Interface Designer* shows us the list of questions we created in the workflow. We could leave the questions like this, yet it provides a nicer user experience if we organize the tools.

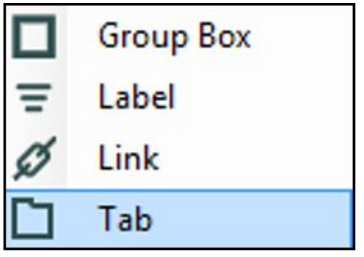


Figure 8-37 - Traffic Tickets –Choosing Tab from the drop down

We can create a new tab by clicking *add drop down* and selecting *Tab*. We can then click on questions we want to move in order to have related questions together.

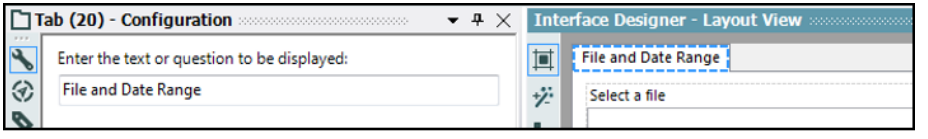


Figure 8-38 - Traffic Tickets – Entering text to be displayed

We can rename this tab by clicking on the tab at the top of the list and changing the text in the *Tab configuration window*.

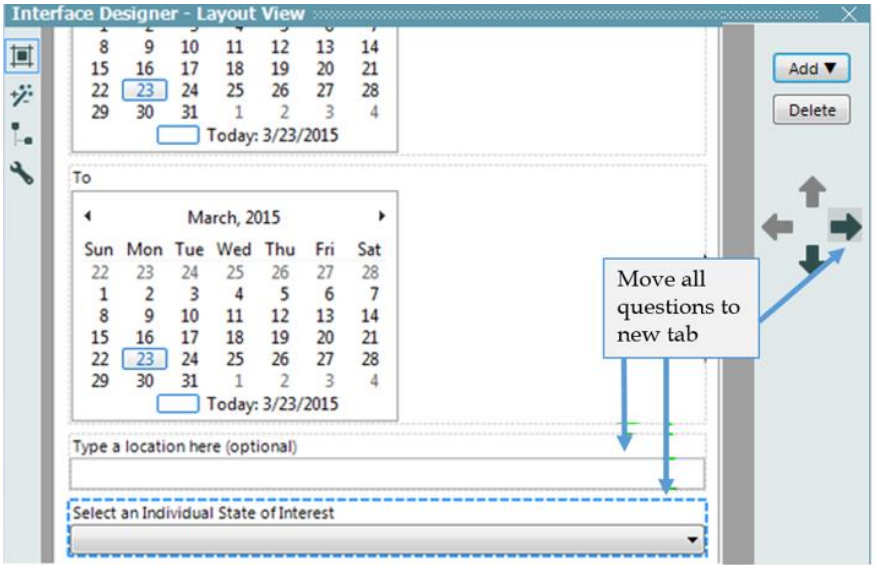


Figure 8-39 - Traffic Tickets – Location information

We can then add a new tab and rename it *Location Information*. Once it is created, we can move the questions in the previous image to that tab that is about Location (of the driver's home state or violation).

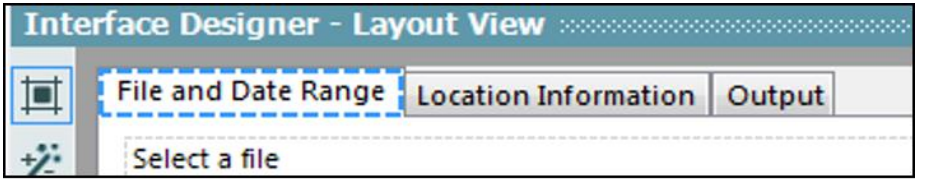


Figure 8-40 - Traffic Tickets – Adding a tab called Output

Now we can create one more tab called *Output*. But before we move questions, we need to add a *Group Box* from the *Add* menu.

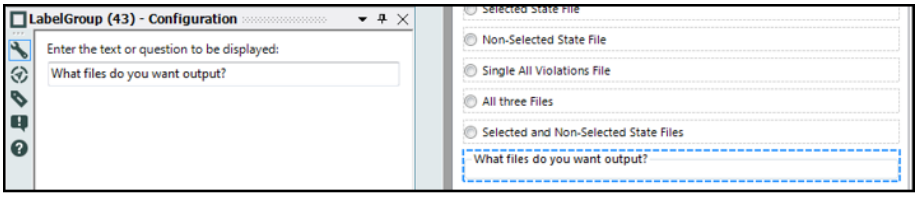


Figure 8-41 - Traffic Tickets – Moving all questions into a group using LabelGroup

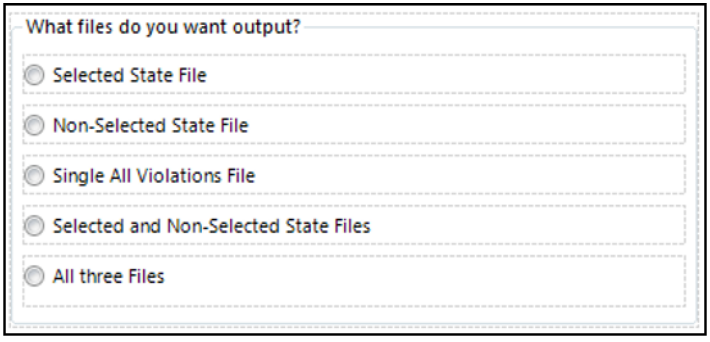


Figure 8-42 - Traffic Tickets – Grouped questions moved to Output tab

Now we will move all the required questions down into the *Group Box* by selecting the question and clicking down until our list looks like the image.

Once we have all of our output questions grouped together, it is easy to move them across into the Output tab. Then we can click on the *Group Box* title and move that.

After completion, we should have the three tabs for our questions.

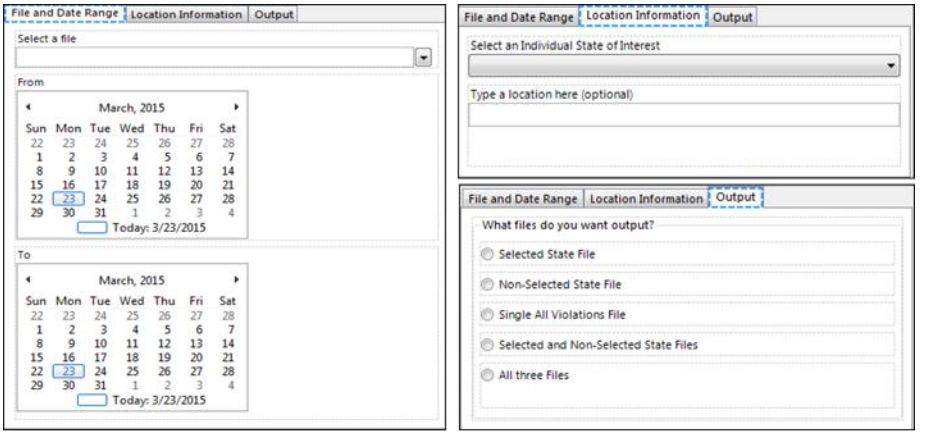


Figure 8-43 - Traffic Tickets – all three tabs

For better representation purposes, some of the connections are made wireless. The final workflow after adding the Interface tools looks as shown in the image.

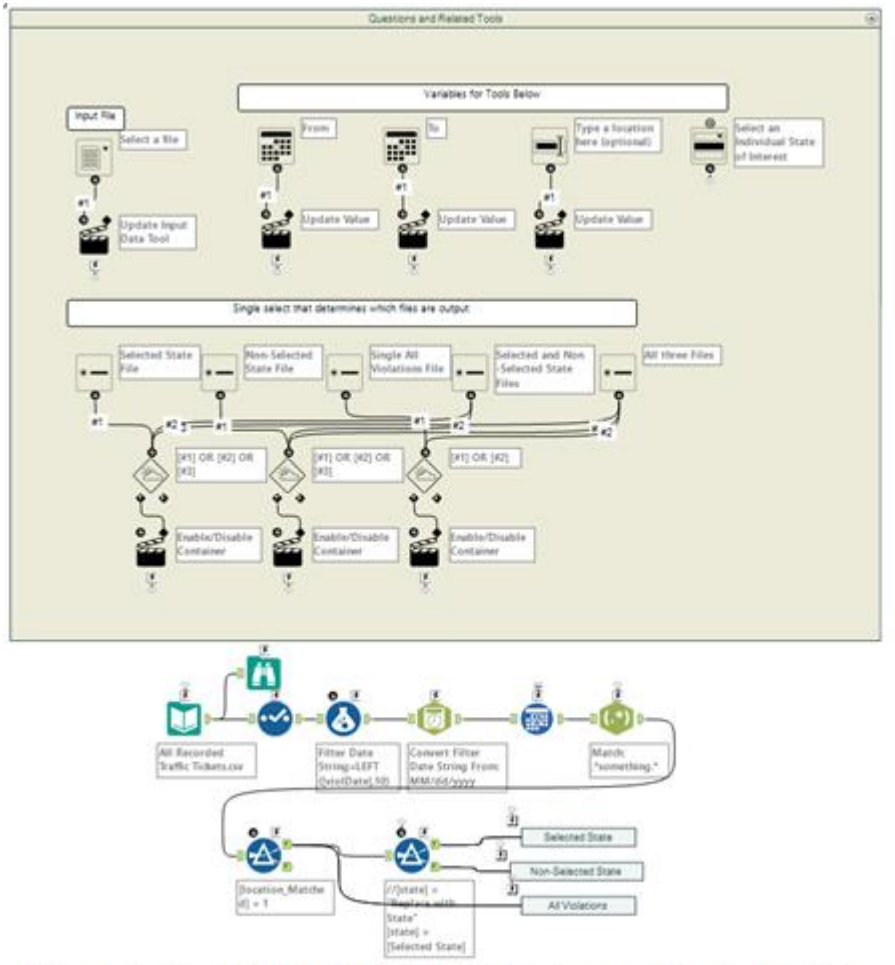
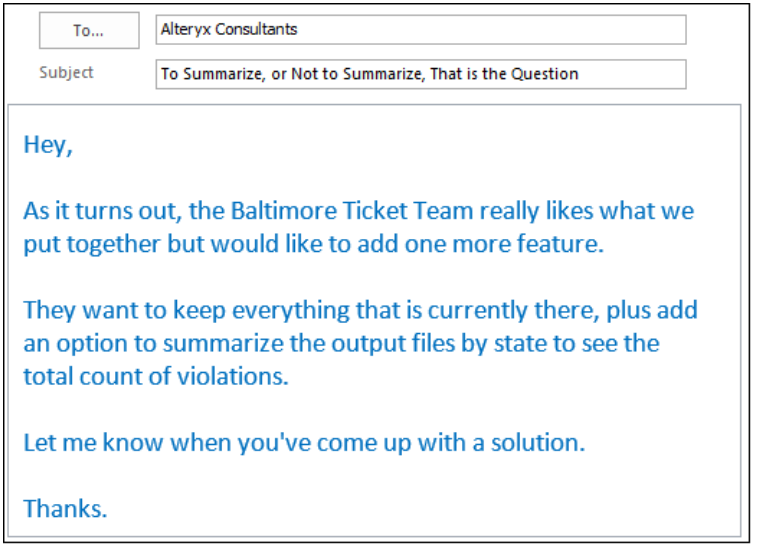
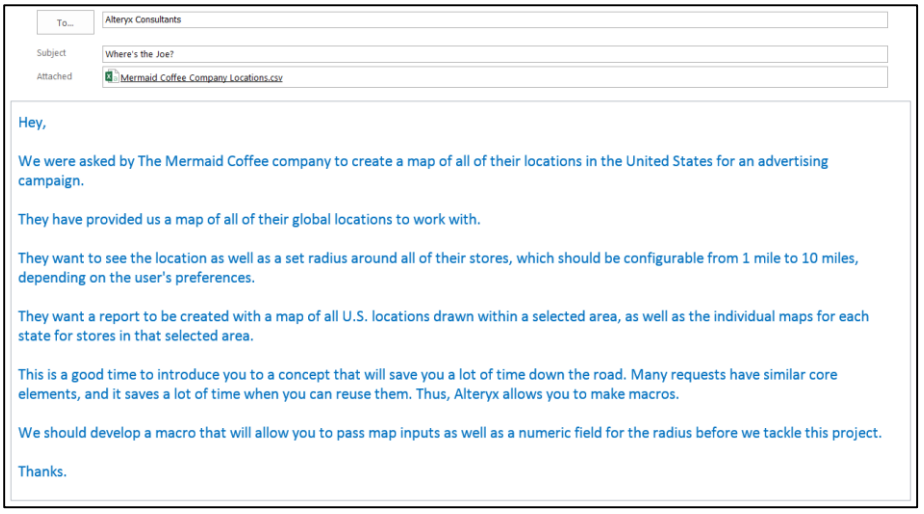


Figure 8-44 – Traffic Tickets – data stream when complete

8.11 To Summarize or Not to Summarize: That is the Question



Chapter 9
Where's the Joe?



To... Alteryx Consultants

Subject Where's the Joe?

Attached Mermaid Coffee Company Locations.csv

Hey,

We were asked by The Mermaid Coffee company to create a map of all of their locations in the United States for an advertising campaign.

They have provided us a map of all of their global locations to work with.

They want to see the location as well as a set radius around all of their stores, which should be configurable from 1 mile to 10 miles, depending on the user's preferences.

They want a report to be created with a map of all U.S. locations drawn within a selected area, as well as the individual maps for each state for stores in that selected area.

This is a good time to introduce you to a concept that will save you a lot of time down the road. Many requests have similar core elements, and it saves a lot of time when you can reuse them. Thus, Alteryx allows you to make macros.

We should develop a macro that will allow you to pass map inputs as well as a numeric field for the radius before we tackle this project.

Thanks.

9.1 Tools & Concepts

In this chapter, we will be covering the following Tools and Concepts:

Tools	**Concepts**
Macro Input	Building Macros
Macro Output	Using Macros
Map	Mapping
Map Input	
Numeric Up Down	
Spatial Match	
Trade Area	

9.2 Macro Input

Figure 9-1 – Macro Input

The *Macro Input* tool allows for creation of an incoming data connection for a macro.

Group	Input	Output
Interface	See below	Any data stream

Note: The *Macro Input* tool receives data from the calling workflow.

The bottom *Black Question Anchor* can be connected to a question anchor on top of a tool to pass the result to the subsequent tool.

Input: The sample data connection or the calling module's data stream.

Properties Window:

The *Macro Input Configuration* window has four basic components.

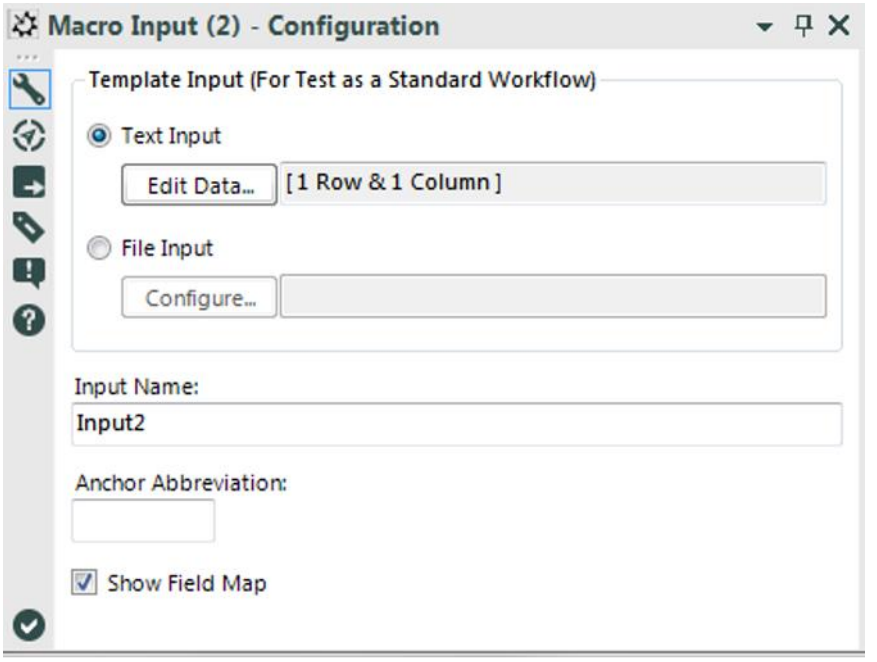


Figure 9-2 – Macro Input Configuration

- *Template Input (For Test as a Standard Workflow)* is the data we will use to build the macro's data stream. The fields here need to match the incoming connections. We can use either a text input or a file input for the source.
- *Input name* is the name of the data connection.
- *Anchor Abbreviation* allows us to type a single character to be displayed on the macro's input connection. Typically, this is used to identify which data stream should be connected to which connector, if we have multiple inputs.

- *Show Field Map* allows us to manually match the fields to be used in the data stream if the names do not match the original data source.

9.3 Macro Output

Figure 9-3 – Macro Output

The *Macro Output* allows us to create an outgoing data connector for a macro.

Group	Input	Output
Interface	Any data stream	See below

Note: The *Macro Output* tool returns data back to the calling workflow.

Output: The incoming data stream into the workflow that called this macro.

Properties Window:

The *Macro Output Configuration* window has two components as shown in the following figure.

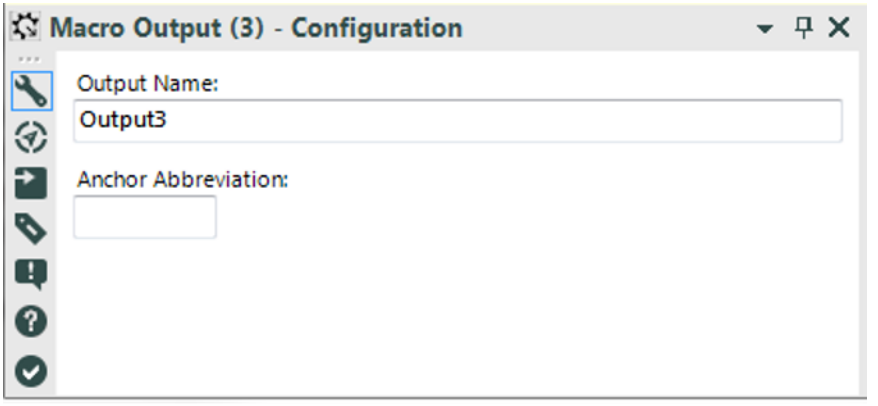


Figure 9-4 – Macro Output Configuration

- *Output Name* allows us to rename the output data connector.
- *Anchor Abbreviation* allows us to type a single character to be displayed on the macro's output connection. Typically this is used to identify which data will be coming out of the connector if we have multiple outputs.

9.4 Map

Figure 9-5 - Map

The *Map* tool allows us to create spatial object(s) on a map for use in the data stream.

Group	Input	Output
In/Out	None	None

Note: The *Map* tool creates spatial objects to be used with a Map Input tool.

Application questions can be connected to the top *White Question Anchor* to update other questions.

The bottom *Black Question Anchor* can be connected to a question anchor on top of a tool to pass the result to the subsequent tool.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to change the spatial objects created.

Properties Window:

The *Map Configuration* window has six basic components.

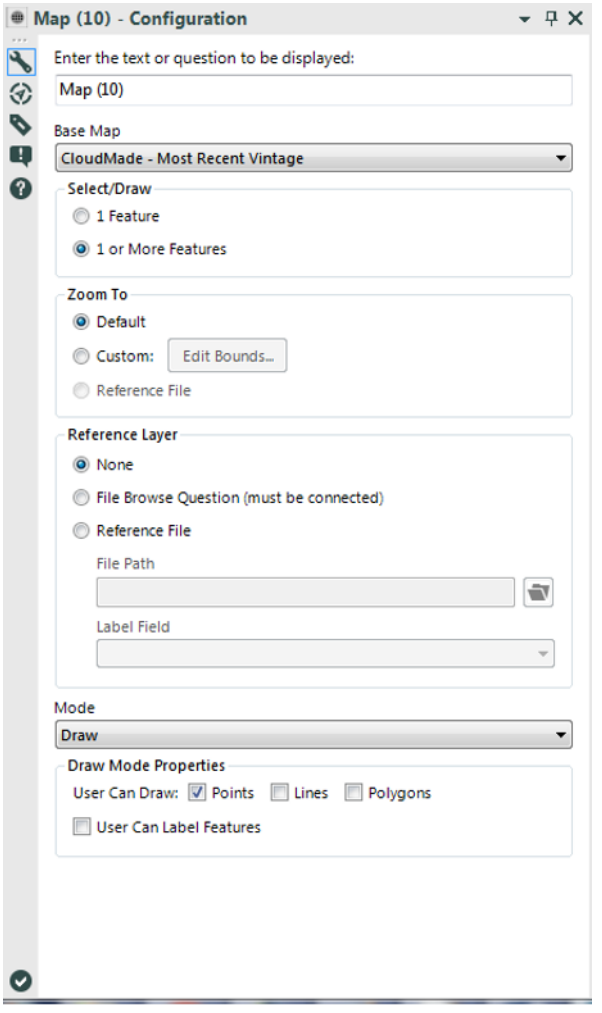


Figure 9-6 – Map Configuration

Enter the text or question to be displayed allows us to write the prompt the end user sees.

- *Base Map* allows us to select the map the end user will be able to use in order to understand where they are clicking.
- *Select/Draw* allows us to decide if the end user is allowed to create single or multiple spatial objects.
- *Zoom To* allows us to decide what area of the map should be shown by default.
- *Reference Layer* allows us to use a spatial file in order to feed spatial information into the map. We would want to do this so end users can select created objects or use those objects for reference points when they create their own. If we choose to add a reference layer, we can feed it in from a *File Browse* question or set a specific file to always use. When we are using either of these modes a Label Field must be selected (if we use the *File Browse* option setting the label field is done in the interface).
- *Mode* allows one of two input methods. *Draw* allows the end user to create objects on their own, while *Select* allows them to choose objects in the reference layer. When we are in *draw* mode, we have the ability to allow the end user to create points, lines or polygons and decide if the user can create custom labels for the objects.

9.5 Map Input

Figure 9-7 – Map Input

The *Map Input* allows us to use map as a source to create spatial objects.

Group	Input	Output
In/Out	None	See below

Note: This tool creates spatial objects.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream with spatial objects and a label field.

Properties Window:

The *Map Input Configuration* window has four basic components.

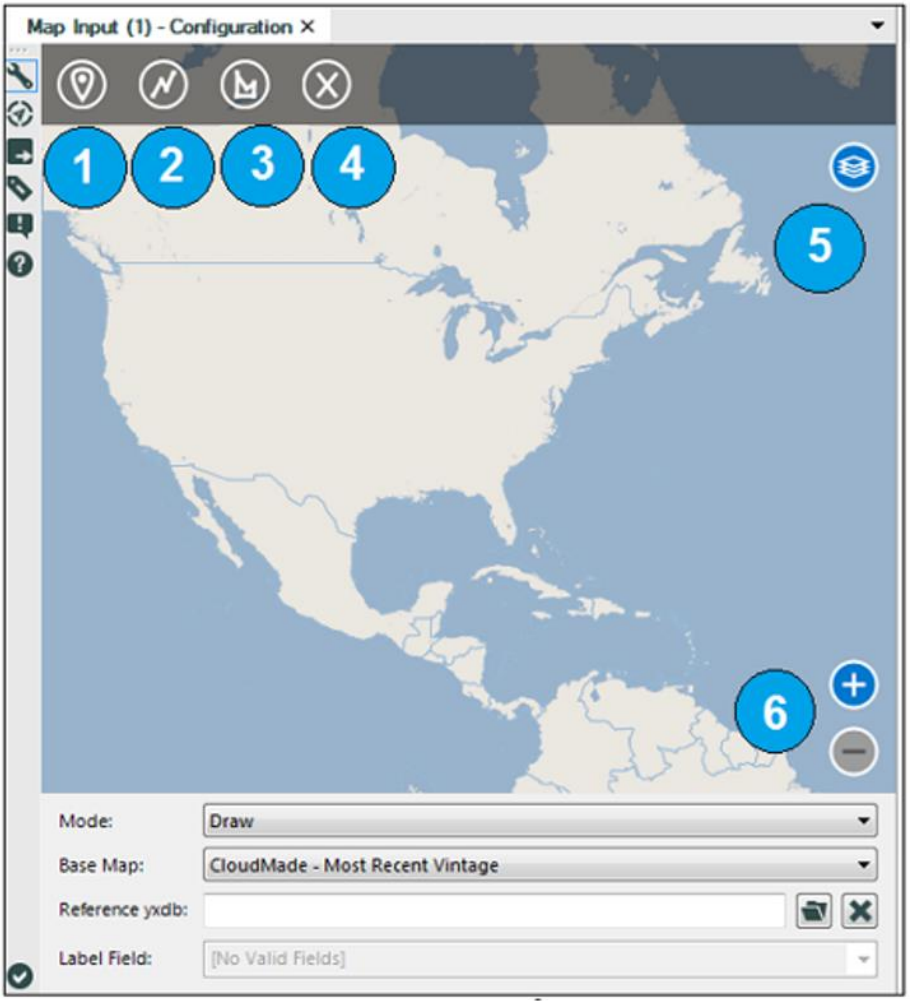


Figure 9-8 – Map Input Configuration

- *Mode* allows us to select between *Draw*, where we create the objects, and *Select*, where we can use objects from a file.

- *Base Map* is the background map we are using to give reference.
- *Reference yxdb* allows us to select an Alteryx database file with spatial objects defined so we can reuse previously created map objects.
- *Label Field* allows us to define which field in the Alteryx database file has the spatial object label names.

The map allows us to create spatial objects. The icons labeled 1 – 6 are:

1. create point objects
2. create line objects
3. create polygon objects
4. clear all created objects
5. show the base map or objects
6. zoom controls

9.6 Numeric Up Down

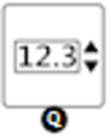


Figure 9-9 – Numeric Up Down

The *Numeric Up Down* question tool allows us to input a numeric value to be used in the data stream.

Group	Input	Output
Interface	None	None

The bottom *Black Question Anchor* can be connected to a question anchor on top of another tool to pass the result to the subsequent tool.

Properties Window:

The *Numeric Up Down Configuration* window has six components.

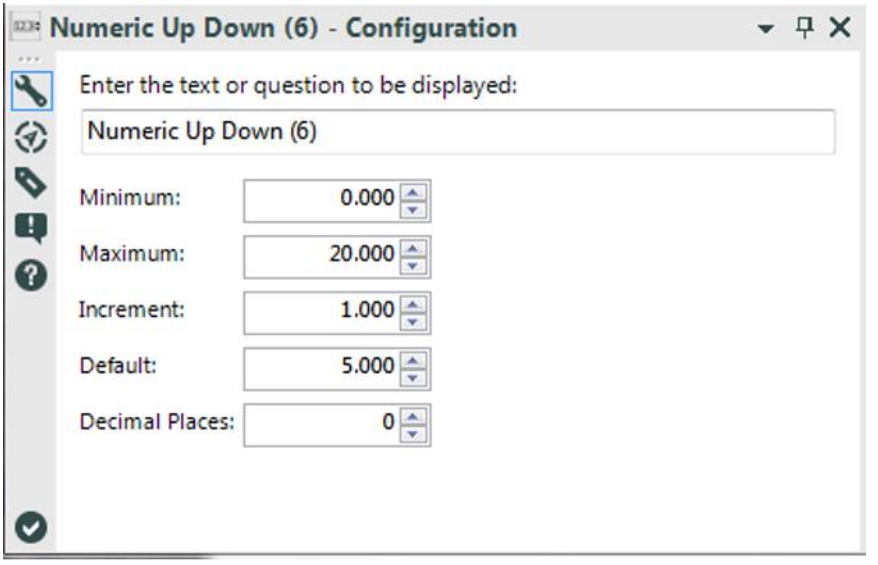


Figure 9-10 – Numeric Up Down Configuration

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.
- *Minimum* allows us to set a lower threshold for the input.
- *Maximum* allows us to set an upper threshold for the input.
- *Increment* allows us to define the most granular level of change.
- *Default* is the value that will auto-populate.
- *Decimal Places* is the number of decimal places that are displayed.

9.7 Spatial Match

Figure 9-11 – Spatial Match

The *Spatial Match* allows us to use one spatial field to filter the data from another.

Group	Input	Output
Interface	Input T & Input F below	Output M & Output U below

Note: One of the two inputs needs to use a polygon. When we have multiple target spatial objects that match multiple universe spatial objects, the *Spatial Match* tool will replicate data in the output.

Input T: Any data stream with a spatial object field.

Input U: Any data stream with a spatial object field (optional depending on settings).

Output M: The target (T) input data stream limited by the records that matched in universe (U) input data stream.

Output U: The target (T) input data stream limited by the records that did not match (unmatched records) in universe (U) input data stream.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Spatial Match Configuration* window has five components.

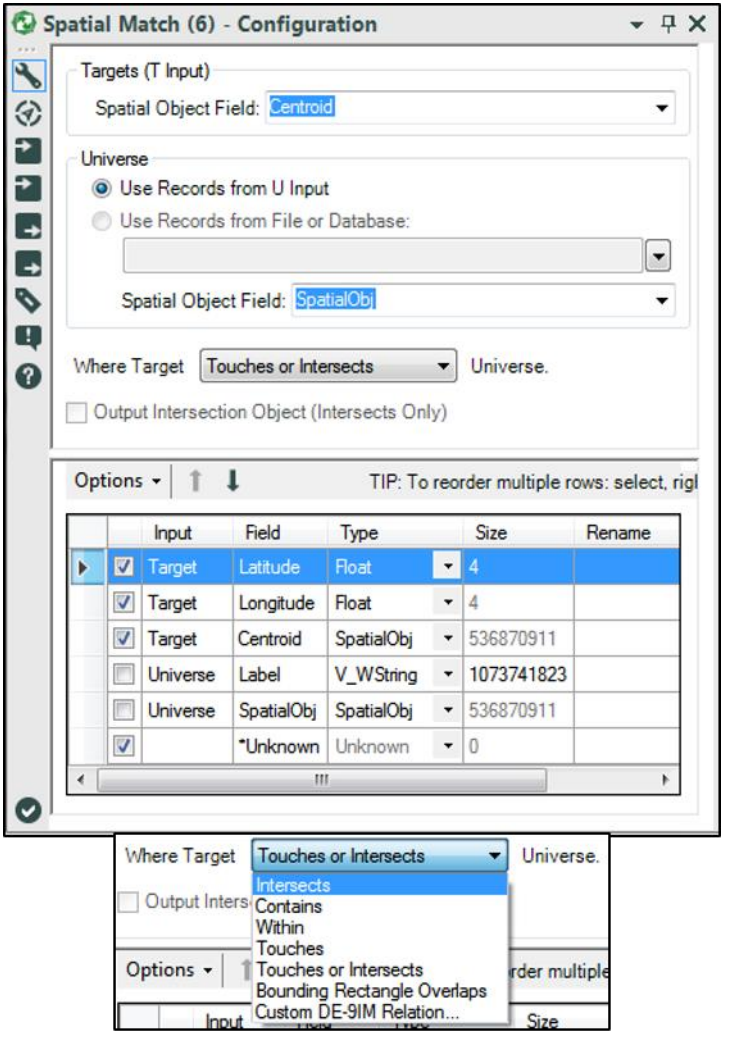


Figure 9-12 – Spatial Match Configuration

- *Targets (T Input)* allows us to select a spatial object field from the target input.

- *Universe* allows us to define if we are going to use a field in the universe input or connect to a spatial data file and then select the field.
- *Where Target* _____ *Universe* allows us to define the relationship that defines a match between the target and universe spatial fields, while the check box will include the intersection as an additional spatial object field.

The types of interaction are:

- *Intersects* means they share the same space in at least a partial manner.
- *Contains* means the entire universe spatial object is in the target spatial object.
- *Within* means the entire target spatial object is in the universe spatial object.
- *Touches* means the two spatial objects share a border but do not intersect.
- *Touches or Intersects* means the two spatial objects share a boarder or intersect.
- *Bounding Rectangle Overlaps* means the fields are matched if the smallest rectangle to contain the entire target spatial object intersects the smallest rectangle to contain the entire universe spatial object.
- *Custom DE-91M Relation* allows us to define customized relationships between the objects.
- The bottom of the tool has an embedded *Select* tool.

9.8 Trade Area

Figure 9-13 – Trade Area

The *Trade Area* tool draws a polygon centered on a point object that is provided.

Group	Input	Output
Spatial	Any data stream with a point spatial field	None

Note: In order to use the drive radius data functionality we need to have the *TomTom* data package.

Lightning Bolt Anchor: An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros

Properties Window:

The *Trade Area Configuration* window has four basic components.

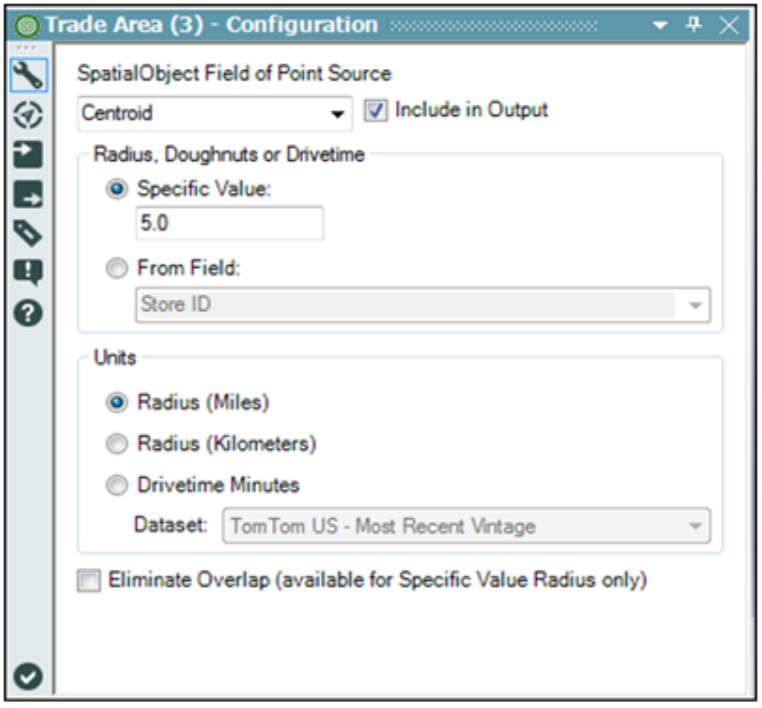
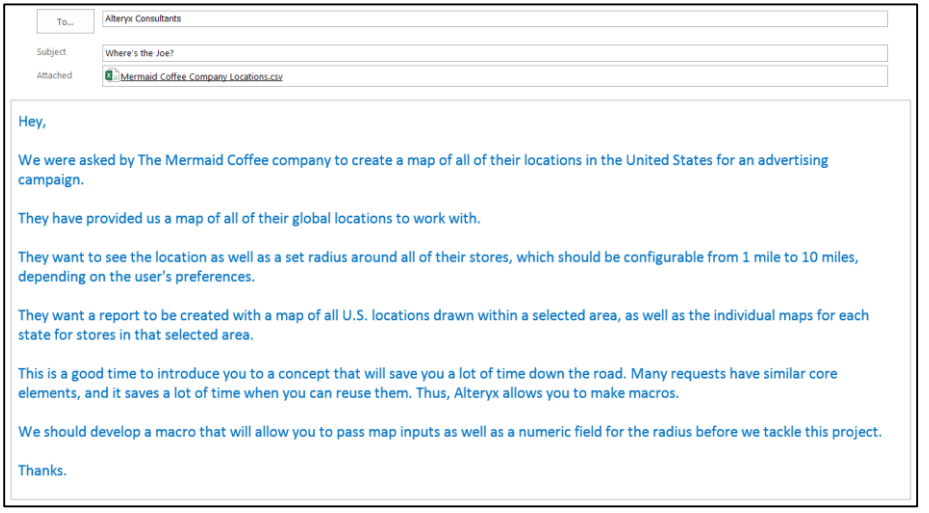


Figure 9-14 - Trade Area Configuration

- *Spatial Object Field of Point Source* shows us the list of spatial object fields in the incoming data and allows us to select the one we are interested in. Checking the *Include in Output* box maintains the point field in the output.
- *Radius, Doughnuts or Drivetime* allows us to set a numeric value for the distance between the point and the edge of the polygon.
- *Units* allows us to define what the unit of measurement Specific Value represents, and if it is Drivetime Minutes, which dataset to use to create the distance.

- *Eliminate Overlap* (available for Specific Value Radius only) allows us to create non-overlapping polygons.

9.9 Where's the Joe?


To... Alteryx Consultants

Subject Where's the Joe?

Attached Mermaid Coffee Company Locations.csv

Hey,

We were asked by The Mermaid Coffee company to create a map of all of their locations in the United States for an advertising campaign.

They have provided us a map of all of their global locations to work with.

They want to see the location as well as a set radius around all of their stores, which should be configurable from 1 mile to 10 miles, depending on the user's preferences.

They want a report to be created with a map of all U.S. locations drawn within a selected area, as well as the individual maps for each state for stores in that selected area.

This is a good time to introduce you to a concept that will save you a lot of time down the road. Many requests have similar core elements, and it saves a lot of time when you can reuse them. Thus, Alteryx allows you to make macros.

We should develop a macro that will allow you to pass map inputs as well as a numeric field for the radius before we tackle this project.

Thanks.

Macros are tools we develop so processes we need to reuse in the same workflow or others don't need to be replicated. This is useful for multiple reasons: The first is it will save us time in creating workflows, and the second is it makes maintenance much easier by allowing us to make a change to a single macro instead of each instance in all files that use the macro.

Let's start by building a macro that is given a data stream with latitude and longitude field. *Macros* are a special type of Analytic Application, so we will be taking the same approach in designing it as we do with apps.

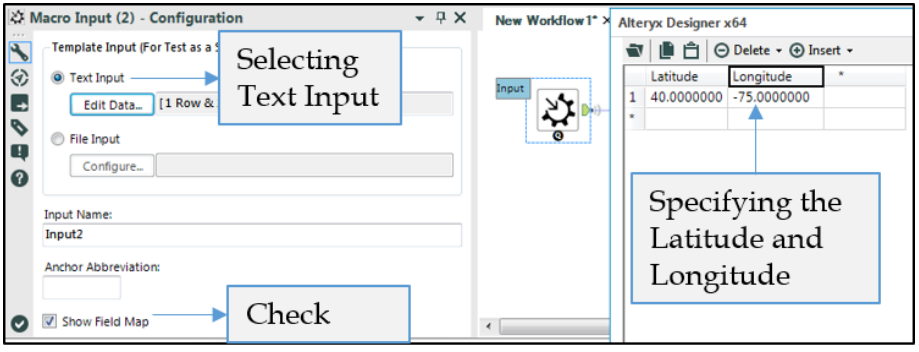


Figure 9-15 – Mermaid Coffee Macro Input for Location

We are going to build this macro by starting with a *Macro Input* tool. We are going to create a sample data using the *Text Input* option and make sure that *Show Field Map* is checked. We are setting up the data this way because we need to make sure we have latitude and longitude fields in any data that uses the tool. In the event the incoming data doesn't have fields named *Latitude* and *Longitude* we are giving the person using this macro the ability to map the fields accordingly.

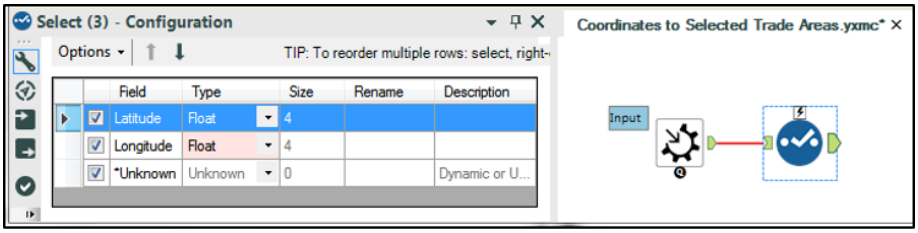


Figure 9-16 – Mermaid Coffee using Select for getting data

Now for the first time we pay attention to the **Unknown* option in the select tool. Because we are building a macro, we don't know what data will be coming into this tool; this option allows us to pass through any unknown fields in the data stream without modifying them.

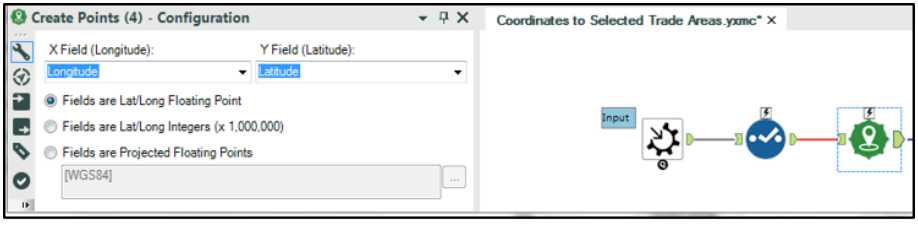


Figure 9-17 – Mermaid Coffee Creating points using default settings

Since we have named the latitude and longitude fields in a way that the *Create Points* tool recognizes, it will auto populate with those fields in the appropriate places in the settings. We are going to use the default settings to create our points.

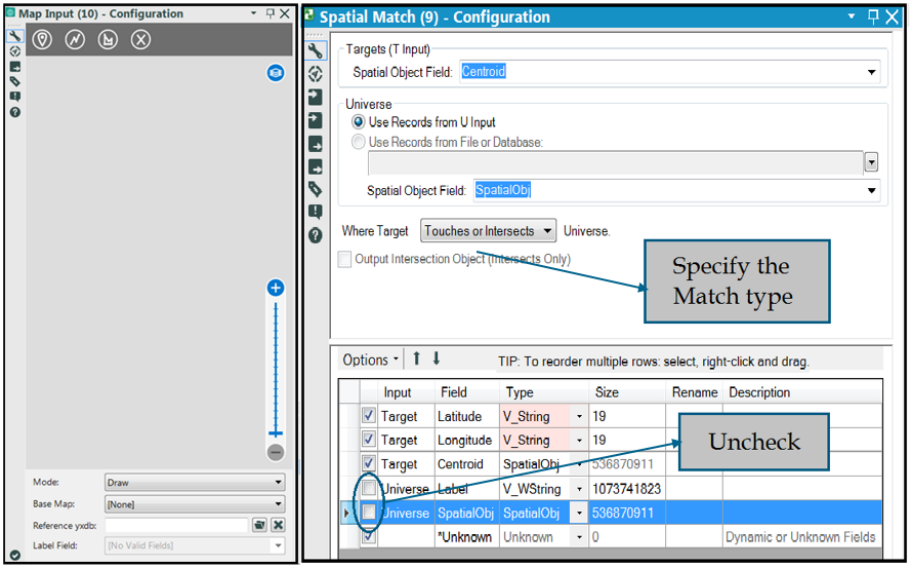


Figure 9-18 - Mermaid Coffee using Map Input and Spatial Match to filter incoming data

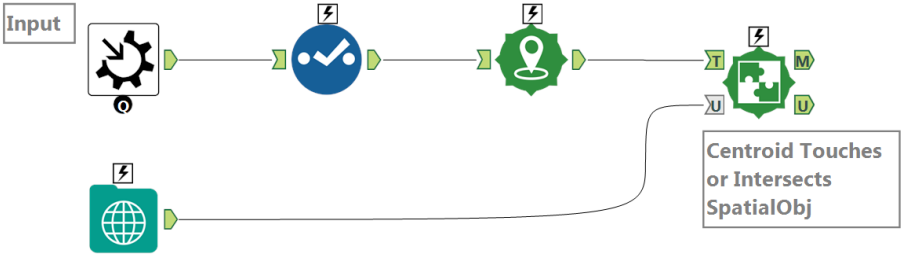


Figure 9-19 - Mermaid Coffee incoming data stream

In order to limit the incoming data, we are going to use a *Map Input* tool to create a spatial object that will be passed to the *Spatial Match* tool. The *Spatial Match* tool has the ability to compare different spatial objects to determine if they share space.

We are not going to configure the *Map Input* tool because we are going to be giving it spatial objects from an interface tool called *Map* once we have finished the basic workflow.

The *Spatial Match* tool will use the configuration shown in the image to include only locations that are in/or touch the user defined Map Input region.

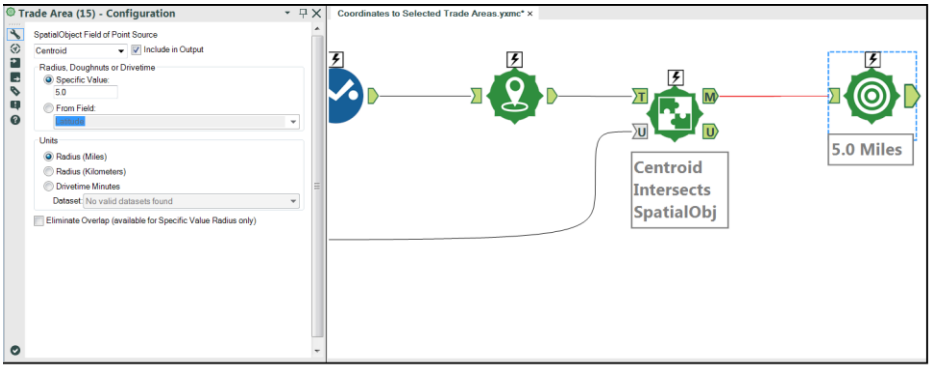


Figure 9-20 - Mermaid Coffee with points and limits

Now that we have limited the points to only those in the area of interest, we can create the *Trade Area* polygons for those points. We are going to make sure that the units are miles, but it doesn't matter that the *Specific Value* is 5 because we are going to use a *Numeric Up Down* to set the distance.

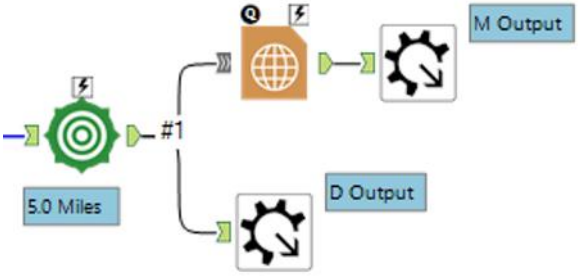


Figure 9-21 – Mermaid Coffee Trade Area Polygons

Now that we have created our trade areas, we need to create outputs.

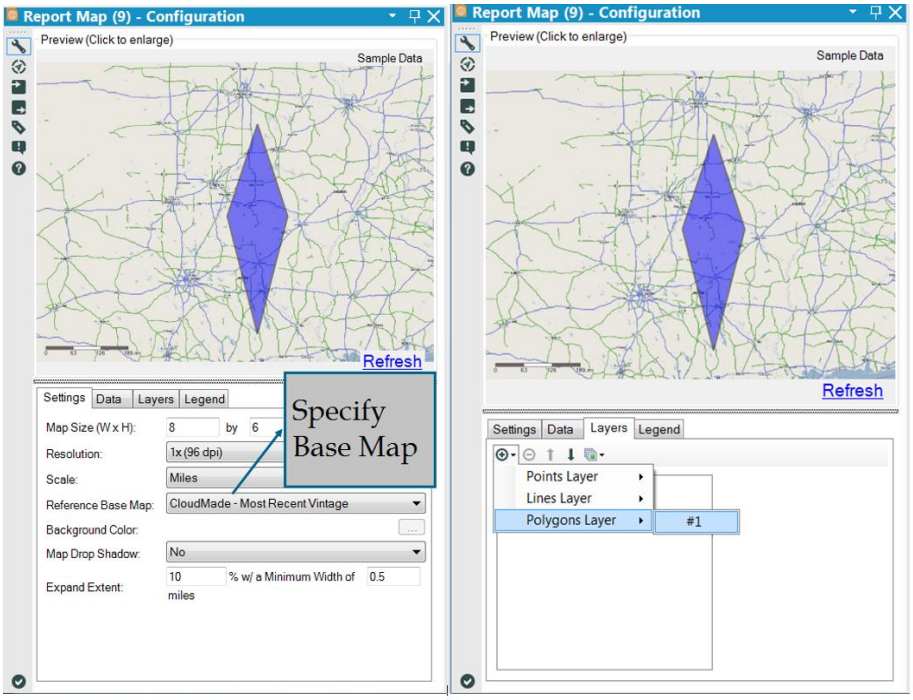


Figure 9-22 - Mermaid Coffee - Using Report Map to create output

For the map output we need to create a map report object. Use any setting of choice; just make sure to include a base layer and the polygon layer from the data stream.

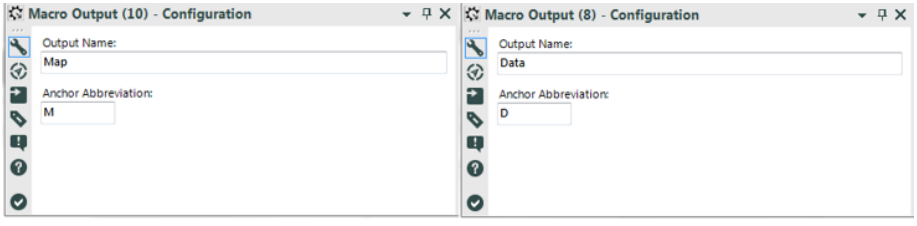


Figure 9-23 – Mermaid Coffee Macro Output

Now all that is left for the outputs is to name them and provide them with an identifying anchor character.

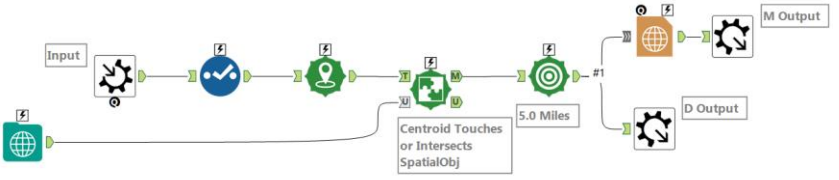


Figure 9-24 - Mermaid Coffee data stream after Macro Output

This macro now only needs the user settings so that the controls can be defined for the two tools we want to be able to change.

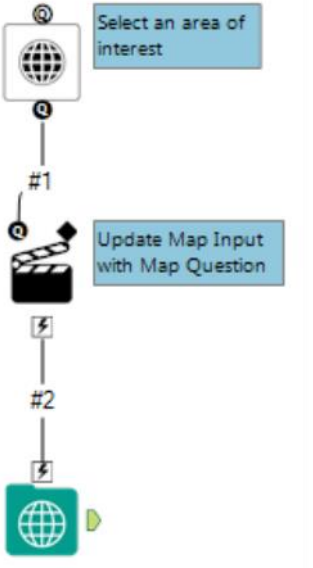


Figure 9-25 – Mermaid Coffee using Map Tool

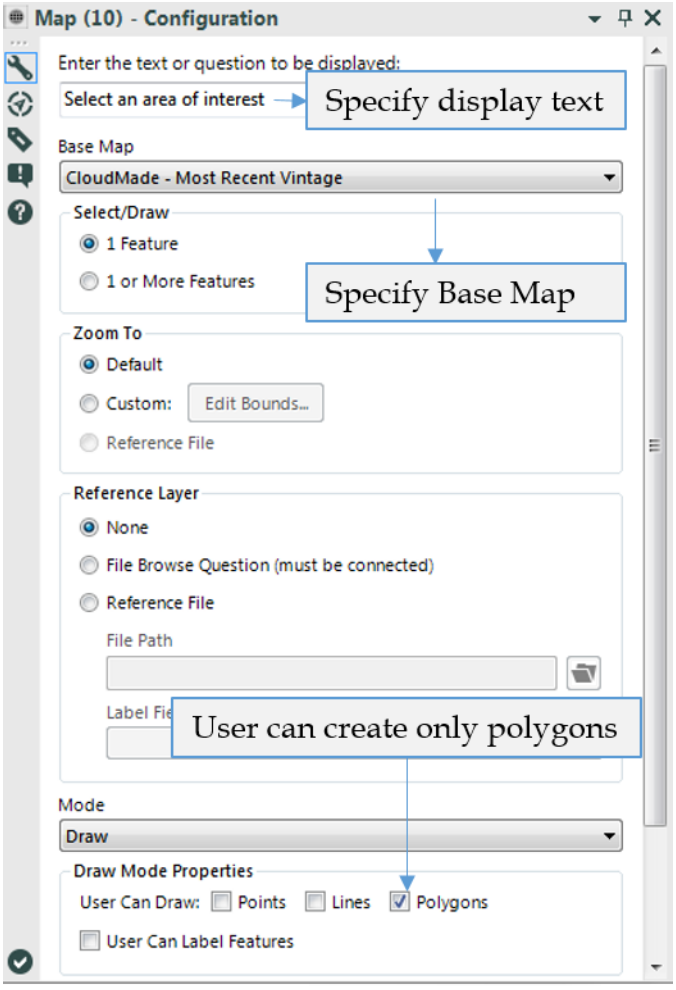


Figure 9-26 – Mermaid Coffee configuration using Map Tool

We can use the *Map* tool to create a map in the interface, which allows the end user to select a specific region.

If we connect the *Map* tool directly to the *Map Input* tool, we see the *Action* tool will auto-populate with the appropriate settings.

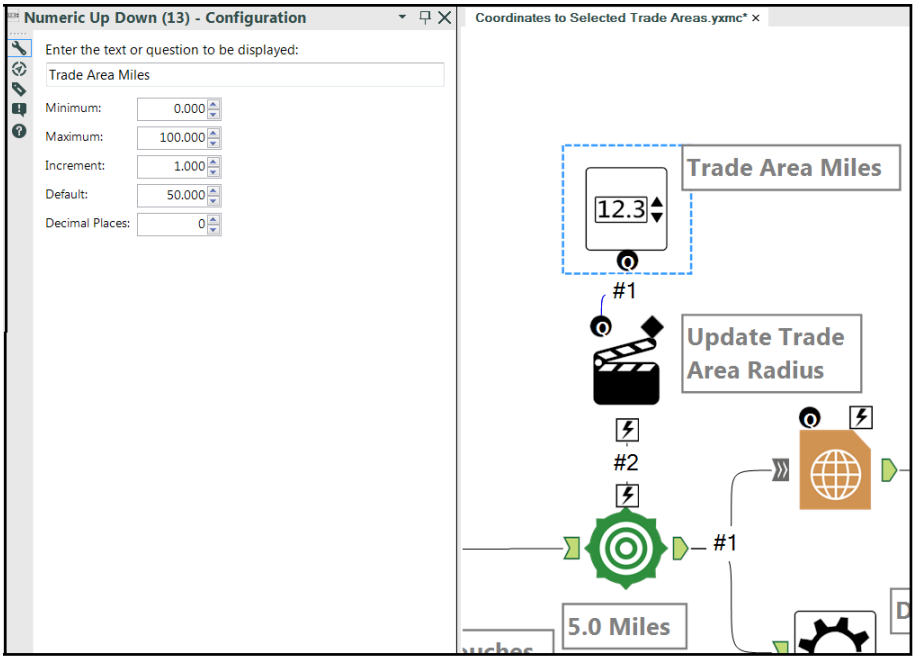


Figure 9-27 - Mermaid Coffee using Numeric Up Down to set Number of miles a trade should cover

If we add a *Numeric Up Down* tool, we will have a numeric input so the user can set the number of miles a *Trade Area* should cover. If we connect this directly to the *Trade Area* tool, we can see the *Action* tool uses the special *Trade Area Radius* setting (so we don't need to configure the *Action* tool). We are giving a much wider set of selected radius here because we don't know how we may want this tool to be used in the future. Since we will be embedding this in another tool we can further limit the options for our current project.

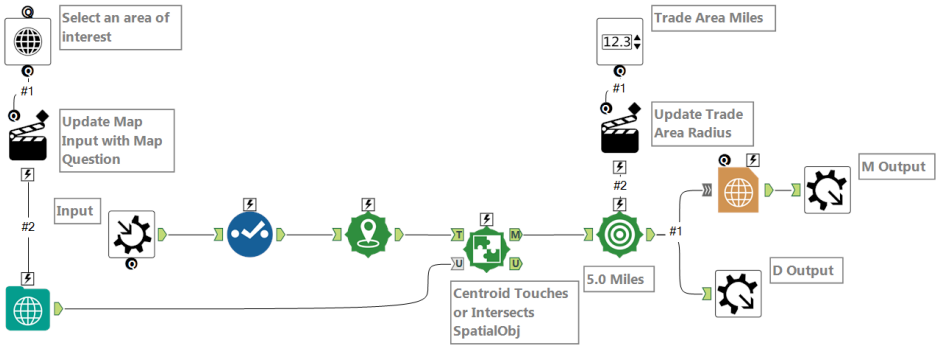


Figure 9-28 - Mermaid Coffee data stream

Now that we have this macro, we can save it to use in other data streams. We are going to create a macro folder in *My Documents* called *Alteryx Macros* and create a sub-folder called *Training Macros*. Let us save this macro there, and we should modify the user settings so our macro is available in the *Tool Palette*.

Navigate to the Edit User Setting window in Tools > User Settings > Edit User Settings.

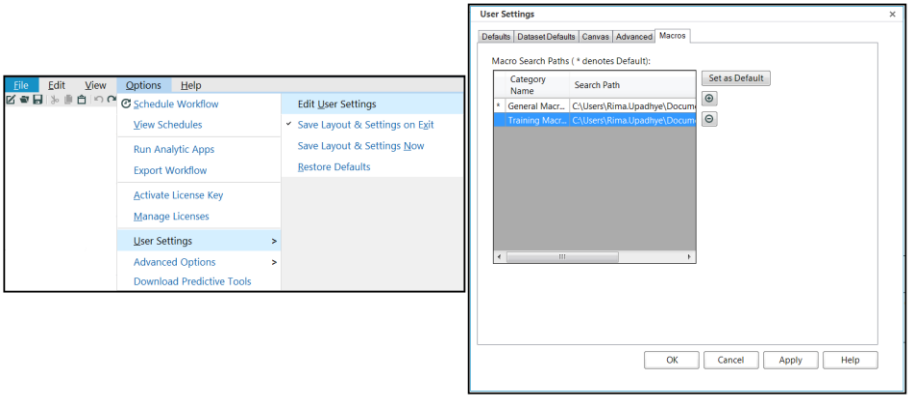


Figure 9-29 - Editing User Settings of Macro

If we go to the Macros Tab, we see the categories we defined and the folder location related to those categories. We have a *General Macros* category and a *Training Macros* category. If we click on the add symbol to the right of the window, it navigates to the *Training Macros*. Let's add it to this list with the title *Training Macros*.

Figure 9-30 – Tool Palette

If we take a look at the *Tool Palette,* we see at the far-right side a Category called *Training Macros*. The image shown here as well as description information can be edited if we modify the properties in the workflow *Configuration window* and *Interface designer.*

Figure 9-31 – Newly saved as Macro

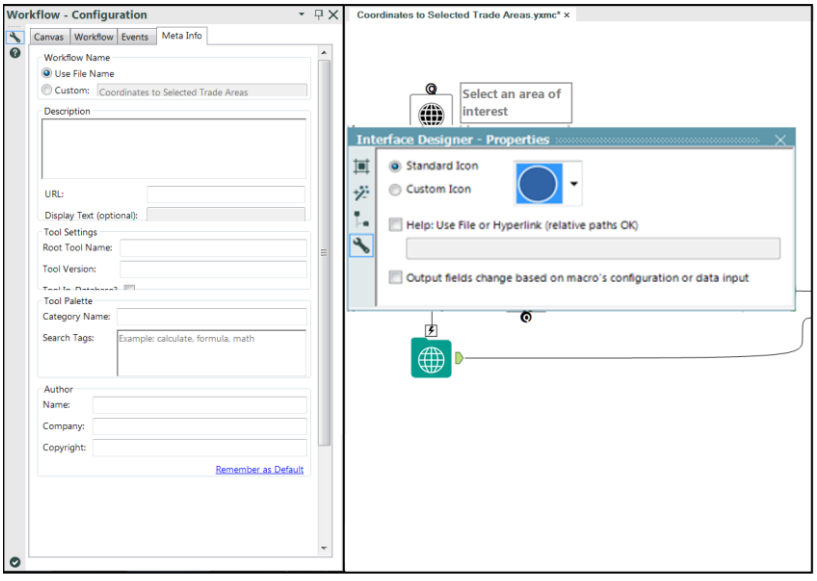


Figure 9-32 - Training Macro Workflow Configuration

Now that we have created the macro, we can build the app specific to the Mermaid Coffee Company. Open a new workflow and connect to the Mermaid Coffee Company Locations file. We can see that all fields are V_String. But we need Latitude and Longitude in a decimal format, which means we need to convert those fields to double so we will have no issue using our macro.

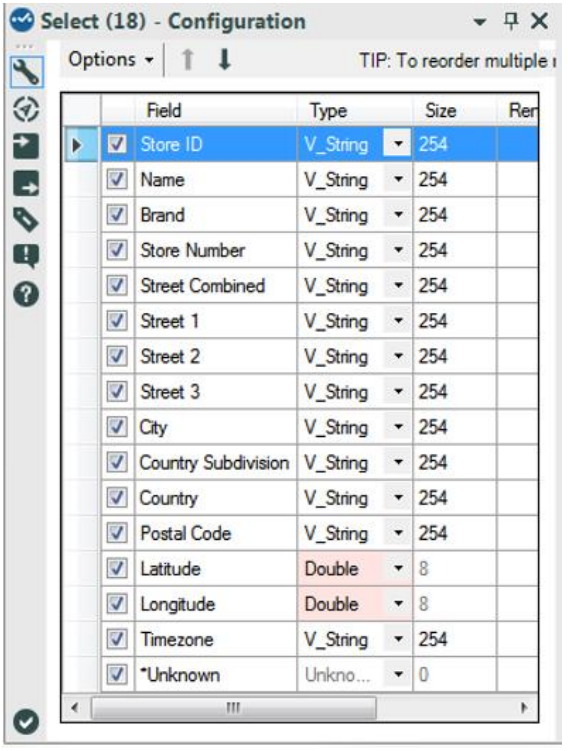
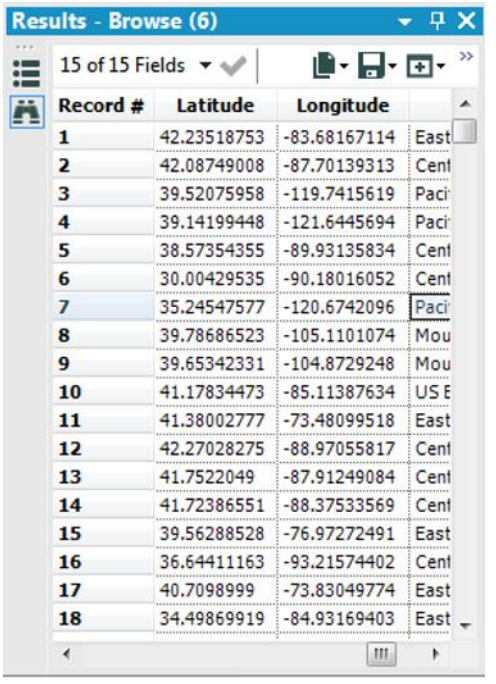


Figure 9-33 – Mermaid Coffee Select



Record #	Latitude	Longitude	
1	42.23518753	-83.68167114	East
2	42.08749008	-87.70139313	Cent
3	39.52075958	-119.7415619	Paci
4	39.14199448	-121.6445694	Paci
5	38.57354355	-89.93135834	Cent
6	30.00429535	-90.18016052	Cent
7	35.24547577	-120.6742096	Paci
8	39.78686523	-105.1101074	Mou
9	39.65342331	-104.8729248	Mou
10	41.17834473	-85.11387634	US E
11	41.38002777	-73.48099518	East
12	42.27028275	-88.97055817	Cent
13	41.7522049	-87.91249084	Cent
14	41.72386551	-88.37533569	Cent
15	39.56288528	-76.97272491	East
16	36.64411163	-93.21574402	Cent
17	40.7098999	-73.83049774	East
18	34.49869919	-84.93169403	East

Figure 9-34 – Mermaid Coffee Browse

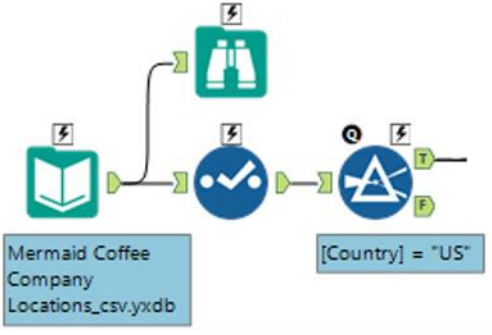


Figure 9-35 – Mermaid Coffee US Location

We can see from the email that they are only interested in US locations, so we can remove all other countries. This means the data is ready for use in our macro.

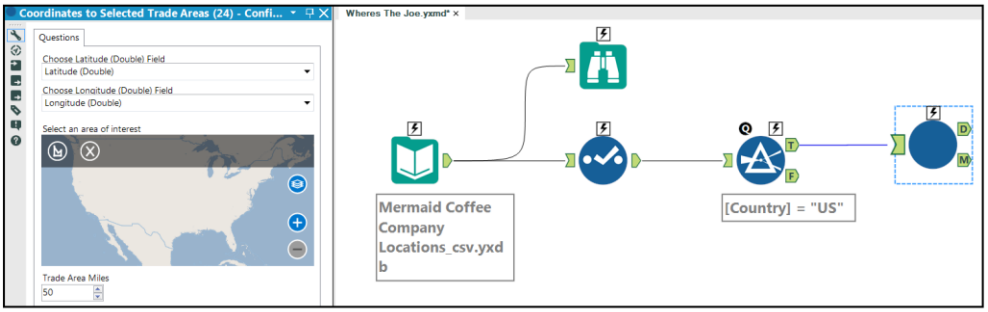


Figure 9-36 - Mermaid Coffee Trade Area configuration to restrict to US

We must make sure the Latitude and Longitude fields match correctly but leave the other *Map* and *Trade Area Miles* as defaults for now. We will connect a *Map* tool and *Numeric Up Down* here to allow the end user to update these values.

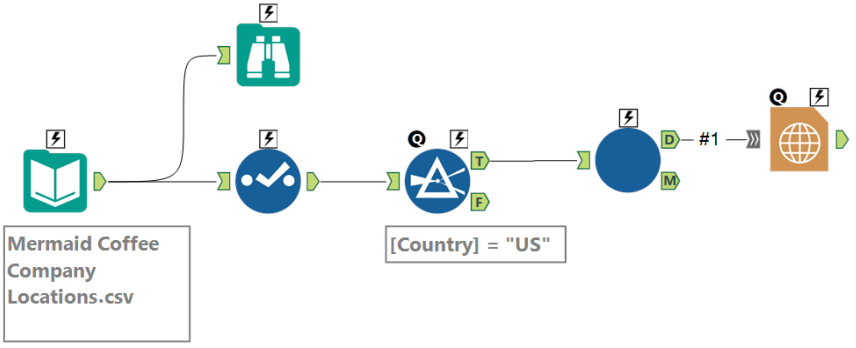


Figure 9-37 - Mermaid Coffee US locations after setting Trade Area

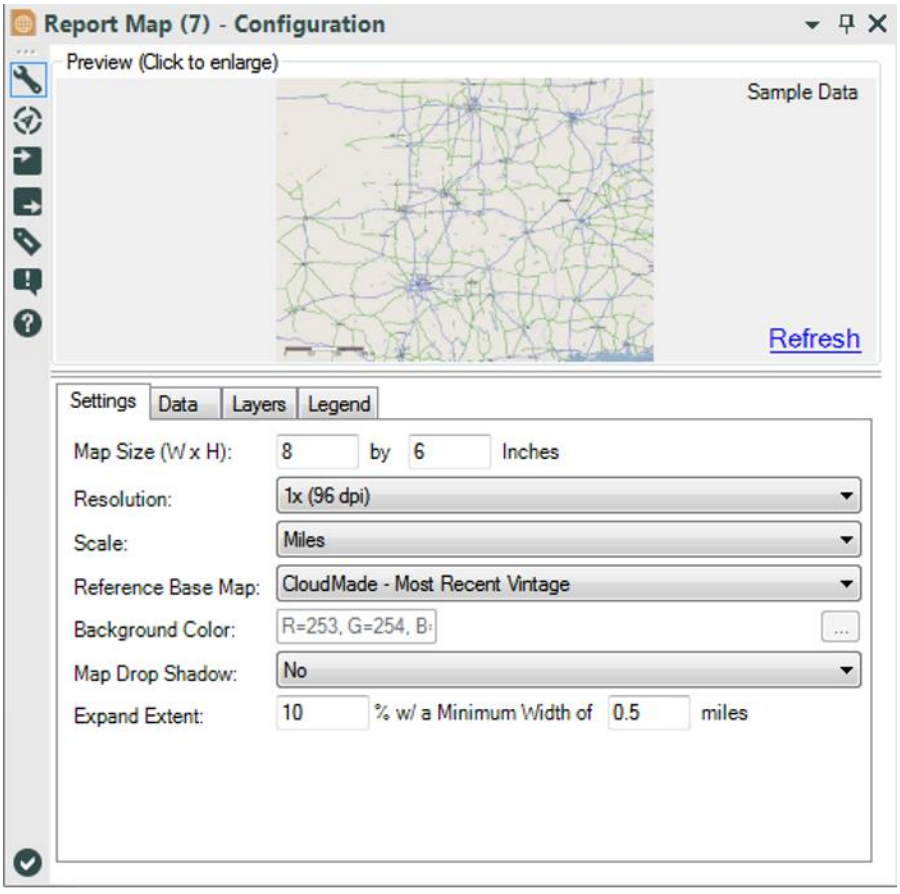


Figure 9-38 – Mermaid Coffee Report Map Setting

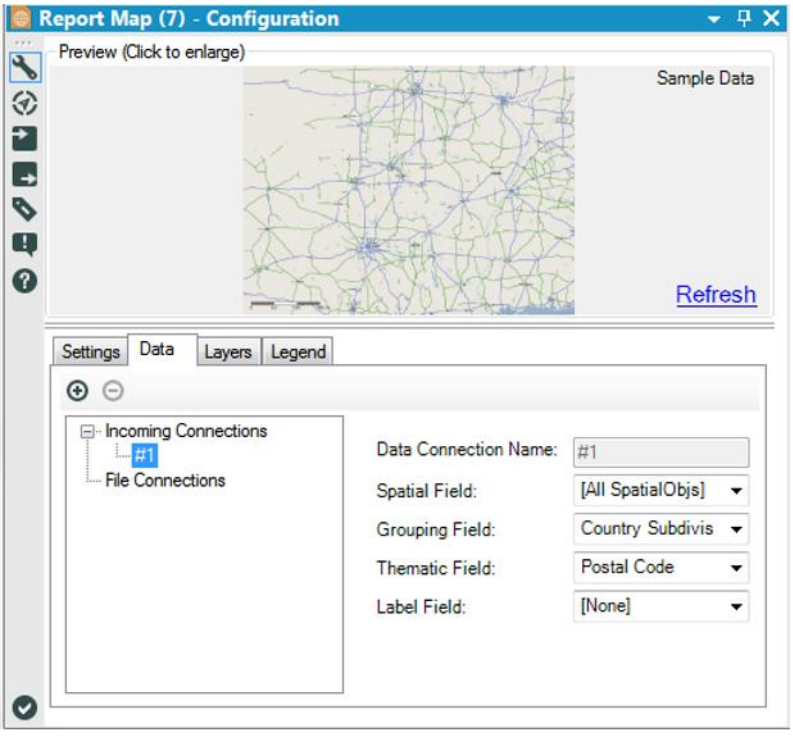


Figure 9-39 – Mermaid Coffee Report Map Data

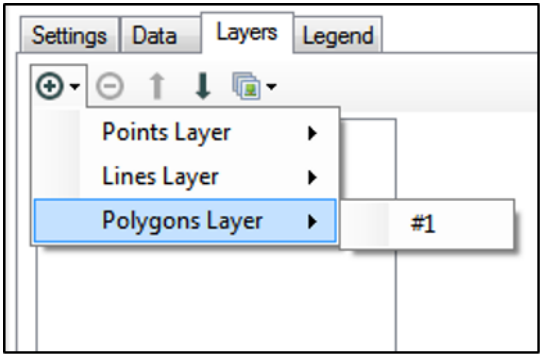


Figure 9-40 – Mermaid Coffee Report Map Layers

We know our macro created a universal map, so we can use that directly. However, we also need individual maps for each state for which we use another *Report Map* tool with the

previously detailed settings to create each of the individual state maps.

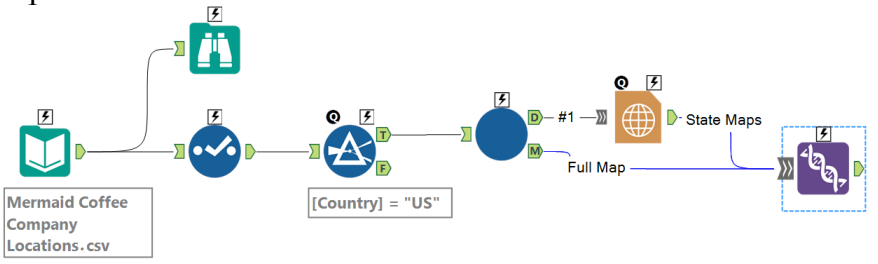


Figure 9-41 - Mermaid Coffee data stream with Maps

Now that we have both sets of maps, we can combine them. If we want the overall map to precede the others, we need to adjust the order in the *Union* tool.

We have all of our maps in a single field. We can use the *Render* tool to create our report.

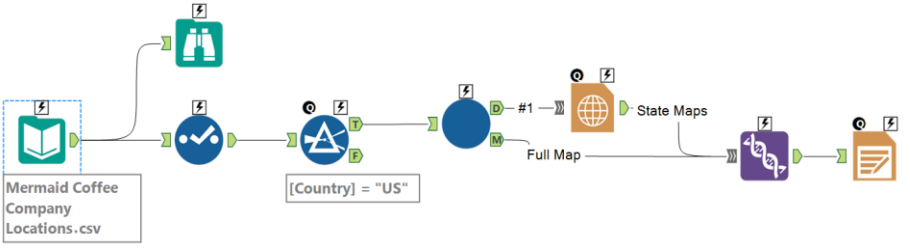


Figure 9-42 - Mermaid Coffee using Union Tool to get overall map

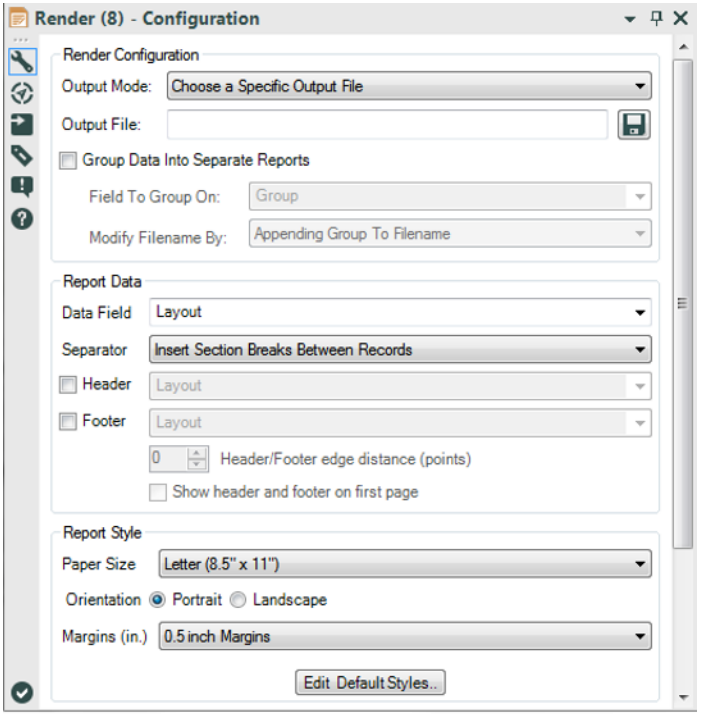


Figure 9-43 – Mermaid Coffee creating Report from Maps

Note that we are inserting section breaks between each record to make sure each map is on a page by itself.

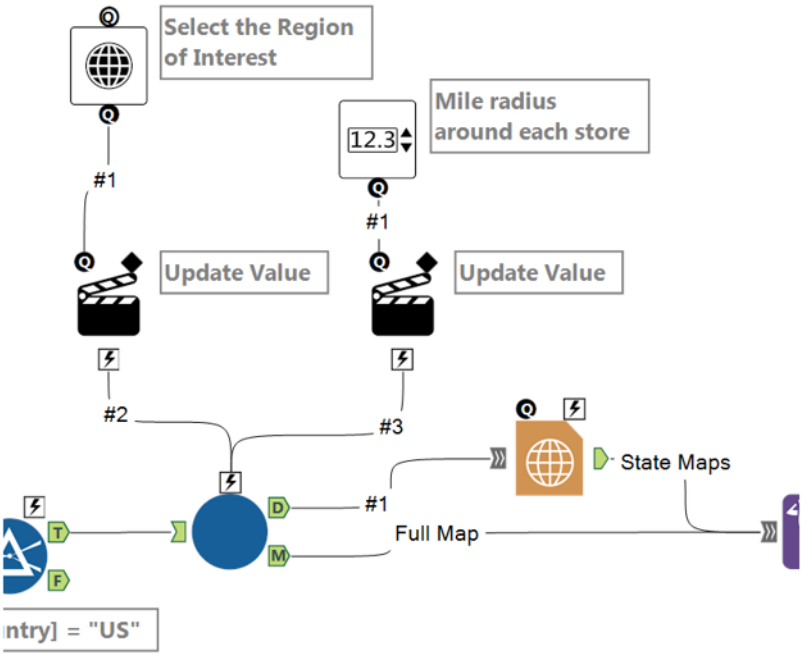


Figure 9-44 – Mermaid Coffee selection Region of interest

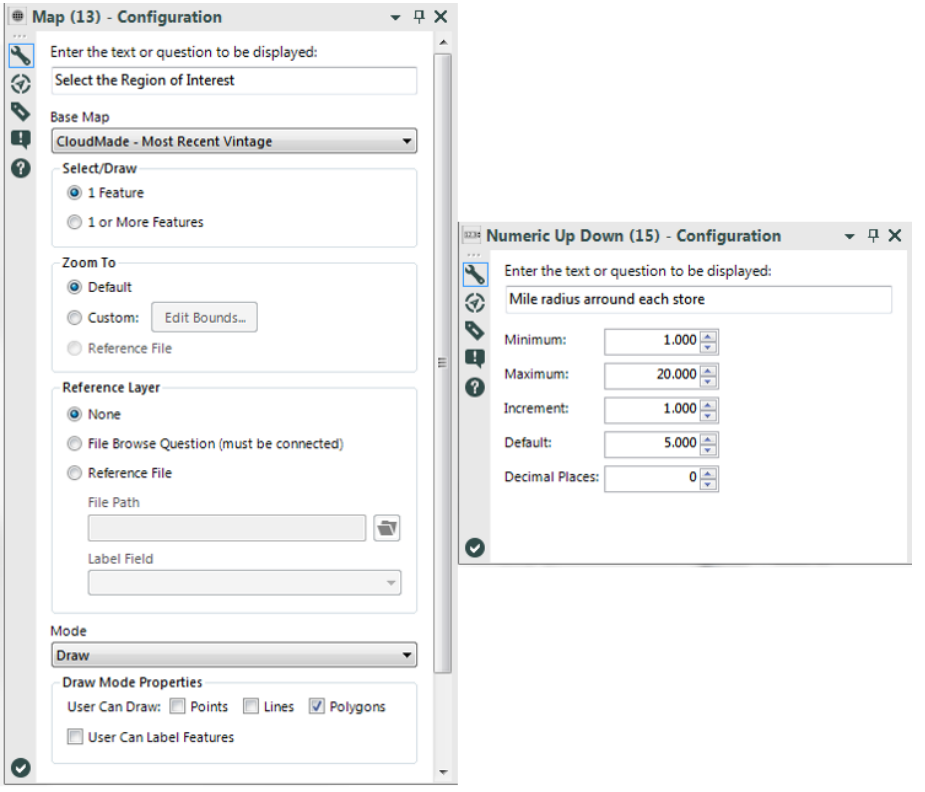


Figure 9-45 – Mermaid Coffee using Map and Numeric Up Down to get desired view

Now that we have the workflow developed, we can finish off this application by adding the two desired user inputs. We will define the *Map* input the same way we did in the macro. However, the *Numeric Up Down* will have a different range of possible values and default value so it reflects what the Mermaid Coffee Company would like to see.

The Where's The Joe? data stream should look like this when it's complete.

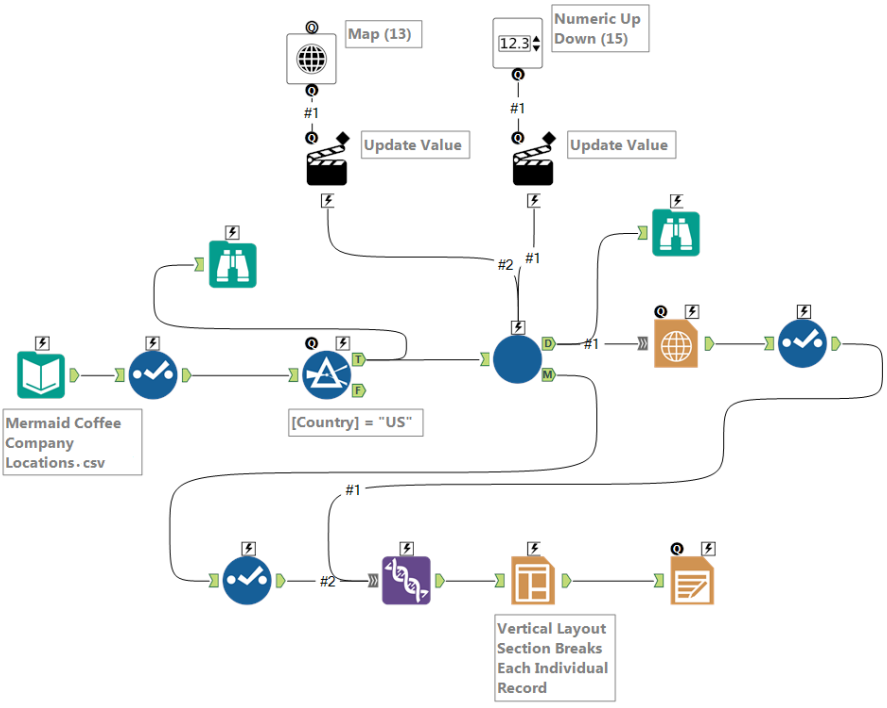
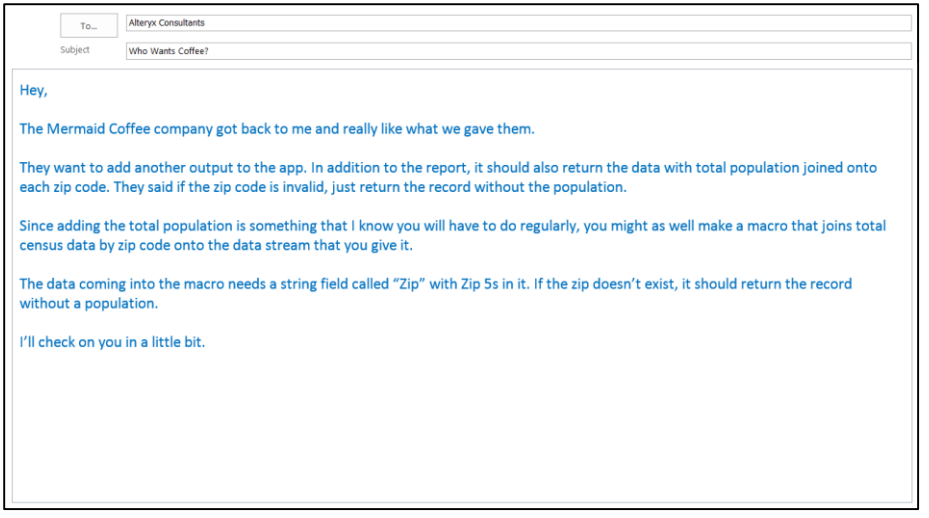


Figure 9-46 - Mermaid Coffee Final data stream when complete

9.10 Who Wants Coffee?


To... Alteryx Consultants

Subject Who Wants Coffee?

Hey,

The Mermaid Coffee company got back to me and really like what we gave them.

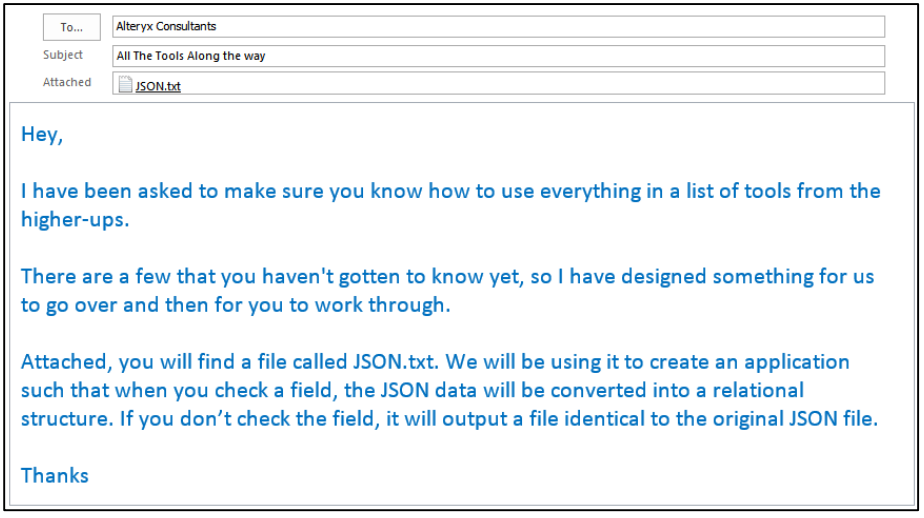
They want to add another output to the app. In addition to the report, it should also return the data with total population joined onto each zip code. They said if the zip code is invalid, just return the record without the population.

Since adding the total population is something that I know you will have to do regularly, you might as well make a macro that joins total census data by zip code onto the data stream that you give it.

The data coming into the macro needs a string field called "Zip" with Zip 5s in it. If the zip doesn't exist, it should return the record without a population.

I'll check on you in a little bit.

Chapter 10
Meta-Morphosis


To... Alteryx Consultants
Subject All The Tools Along the way
Attached JSON.txt

Hey,

I have been asked to make sure you know how to use everything in a list of tools from the higher-ups.

There are a few that you haven't gotten to know yet, so I have designed something for us to go over and then for you to work through.

Attached, you will find a file called JSON.txt. We will be using it to create an application such that when you check a field, the JSON data will be converted into a relational structure. If you don't check the field, it will output a file identical to the original JSON file.

Thanks

10.1 Tools & Concepts

Tools	**Concepts**
Check Box	Alternative Data Prep
Detour	JSON Files
Detour End	Documentation
Dynamic Rename	
JSON Parse	
Message	

10.2 Check Box

Figure 10-1 – Check Box

The *Check Box* interface tool allows us to create a check box question.

Group	Input	Output
Interface	None	None

Special Note: This *Check Box* returns a Boolean value.

The bottom *Black Question Anchor* connects to question anchors on top of other tools to pass the results to the subsequent tool.

Properties Window:

The *Check Box Configuration* window has three components.

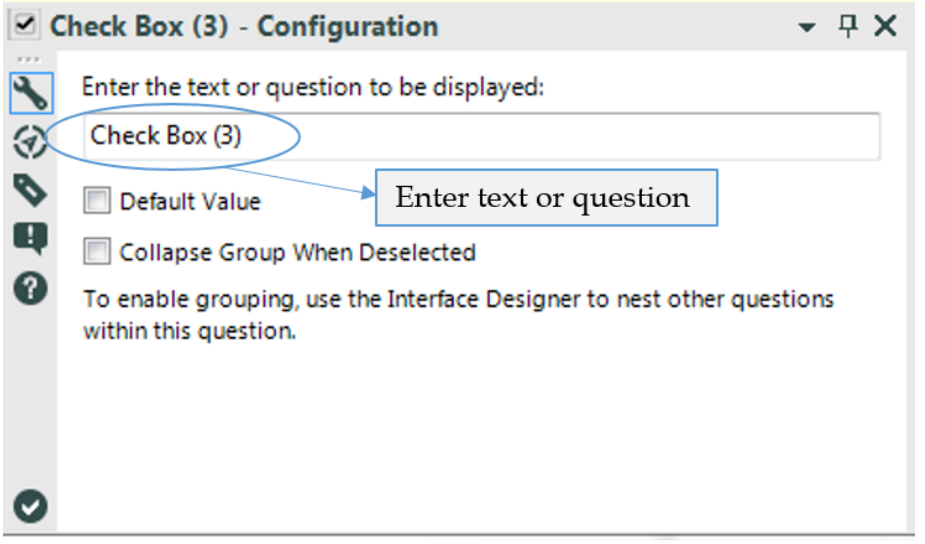


Figure 10.2 – Check Box Configuration

- *Enter the text or question to be displayed* is a text box that will be the question prompt.
- *Default Value* can either be checked or unchecked to determine what the default result will be.
- *Collapse Group When Deselected* allows us to hide the relevant group in the interface designer.

10.3 Detour

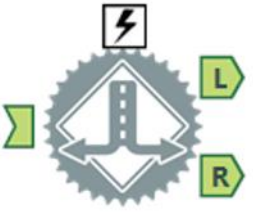


Figure 10.3 - Detour

The *Detour* tool allows us to create alternative paths dependent on the user input.

Group	Input	Output
Developer	Any data stream	Output L & R sections below

Note: Detour is useful if we are developing flexible tools that allow choices in data inputs or outputs. This tool should almost always be used with an action tool. Only the Left or Right stream will be active.

An Action tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output L: Nothing or the input data stream.

Output R: Nothing or the input data stream.

Properties Window:

The *Detour Configuration* window has only one component.

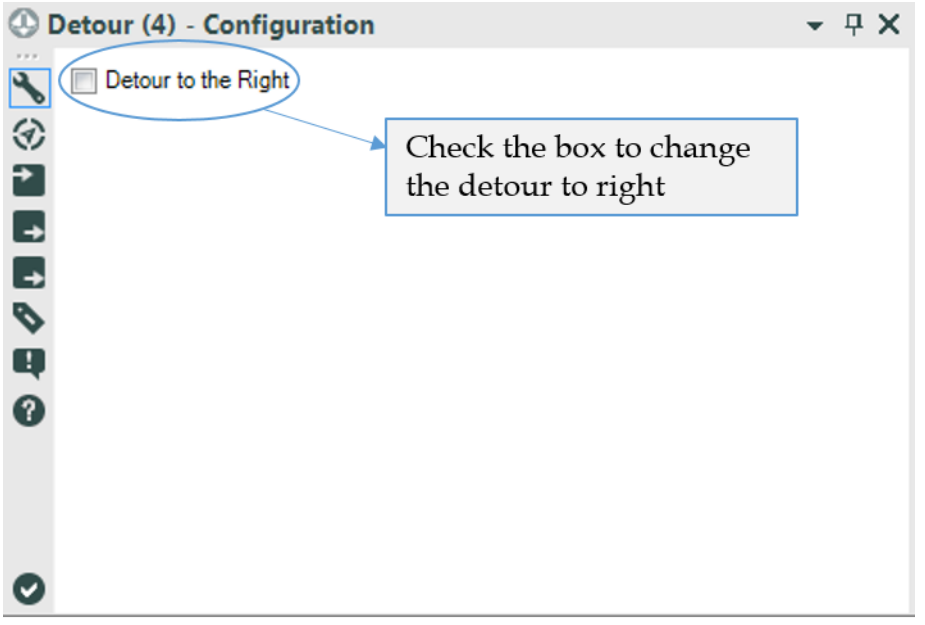


Figure 10-4 – Detour Configuration

- The *default* behavior for the *detour* tool is to always divert to the left (L) data stream.
- *Detour to the Right* allows us to reverse this default behavior.

10.4 Detour End

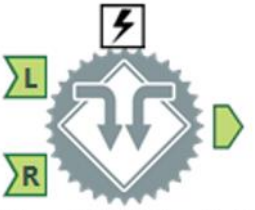


Figure 10.5 – Detour End

The *Detour End* tool rejoins two optional paths.

Group	Input	Output
Developer	Input L & Input R below	Any data Stream

Note: The *Detour End* tool should be used after we have used a Detour tool in order to bring the data streams back together. Only the Left or Right stream can be active.

Input L: Nothing or any data stream.

Input R: Nothing or any data stream.

An Action tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Detour End Configuration* window has no configuration.

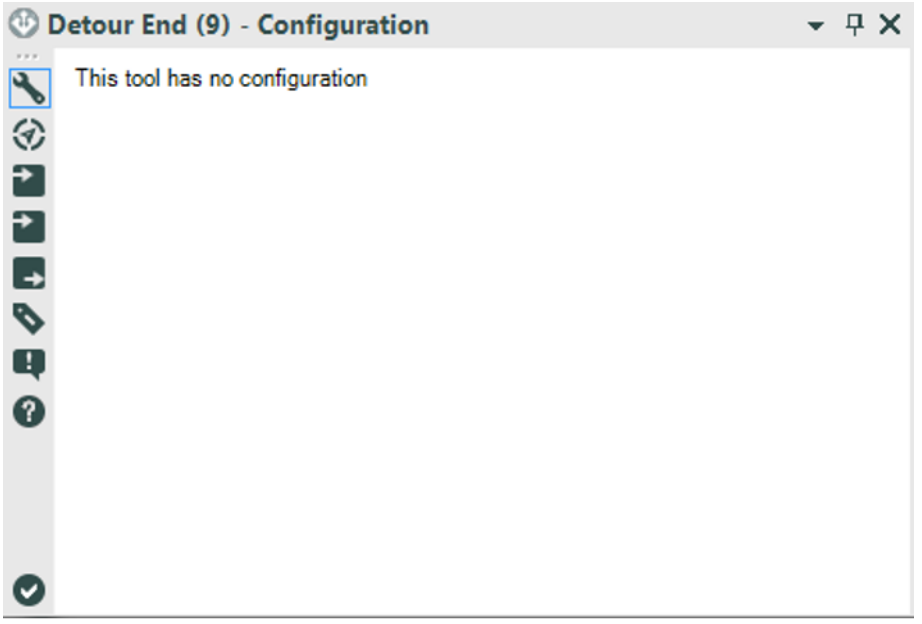


Figure 10-6 – Detour End Configuration

10.5 Dynamic Rename

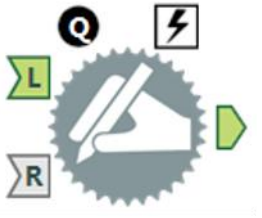


Figure 10.7 – Dynamic Rename

The *Dynamic Rename* tool allows us to systematically rename fields.

Group	Input	Output
Developer	Input L & Input R below	Any data stream

Note: The *Dynamic Rename* tool is useful in properly identifying column headers in data that either have header rows before the column names or no column names.

Input L: Any data stream.

Input R: This input is optional, and its structure depends on the Rename Mode.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the top *Black Question Anchor* to use answers in the tool.

Properties Window:

The *Dynamic Rename Configuration* window has three core components and eight different modes.

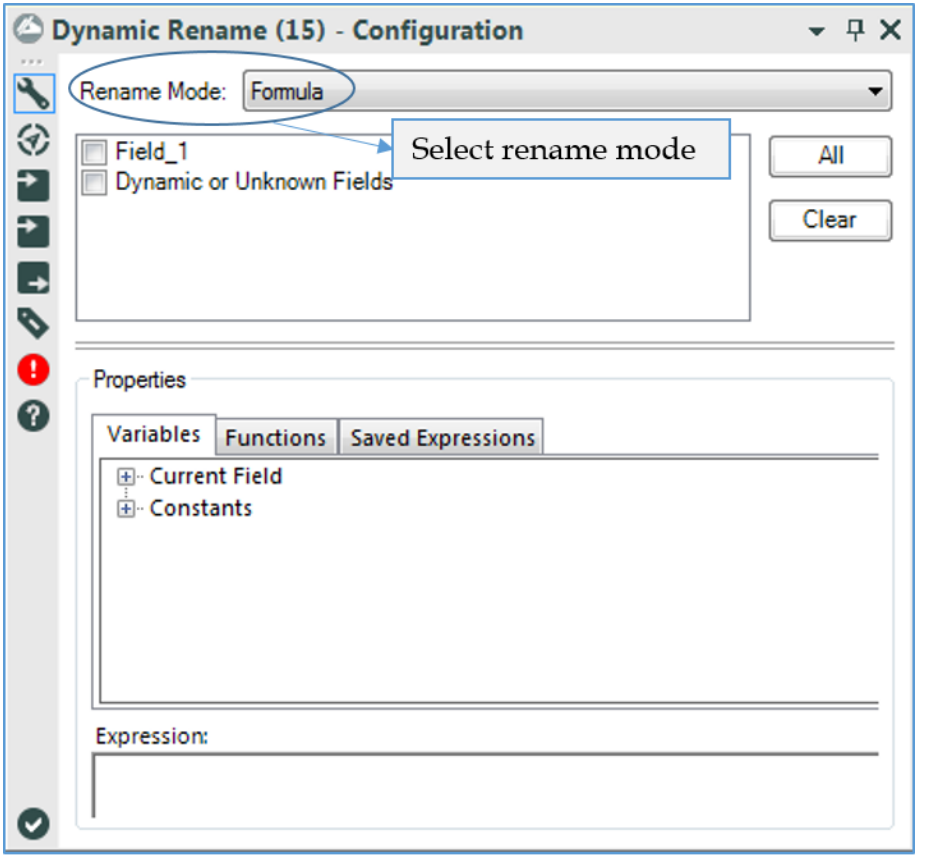


Figure 10-8 – Dynamic Rename Configuration

- *Rename mode* allows us to select the method used to rename the fields in the left (L) input.

If we are using one of the single input rename modes, we will see a field selection window.

- The *Properties* section changes, depending on the Rename Mode.
- *Formula Rename Mode* allows us to define an expression that will name our columns.

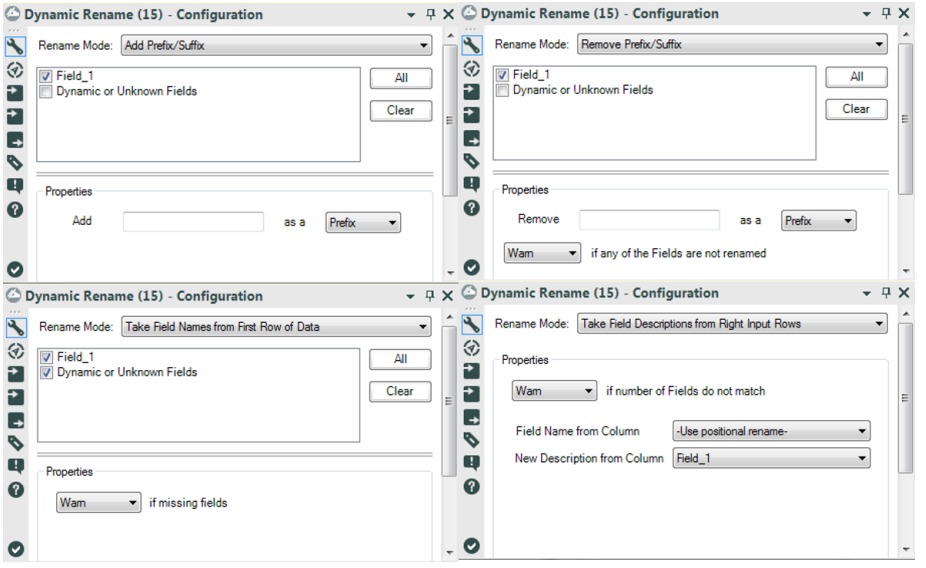


Figure 10-9a – Dynamic Rename (contd.)

- *Add Prefix/Suffix Rename Mode* allows us to define a string to add to the beginning or end of specific fields.
- *Remove Prefix/Suffix Mode* allows us to remove a common string from the beginning or end of specific fields.
- *Take Field Names from First Row of Data* allows us to replace the column headers with the first record of our data.
- *Take Field Descriptions from Right Input Rows* allows us to use a definition file to redefine the fields.

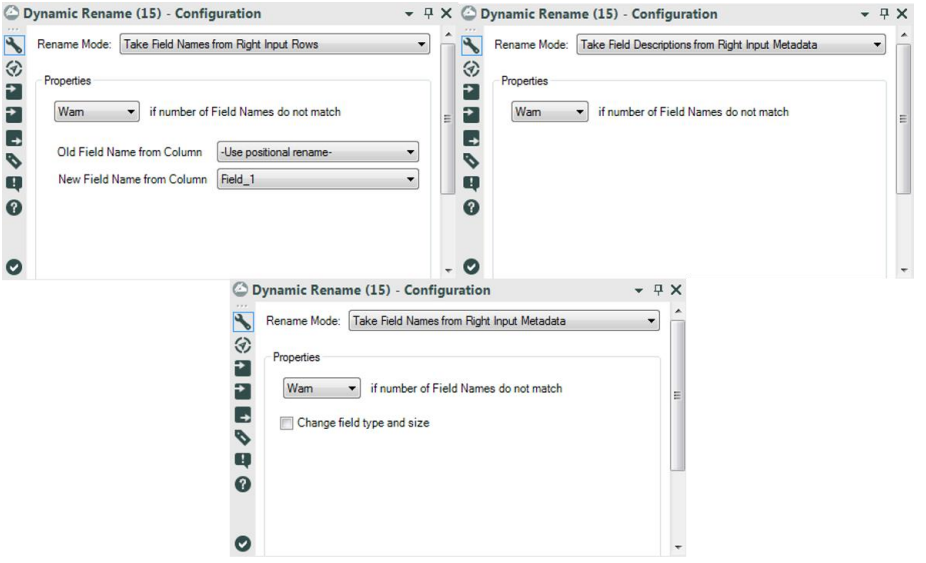


Figure 10-9b – Dynamic Rename (contd.)

Take Field Names from Right Input Rows allows us to use a definition file to rename the fields.

Take Field Names from Right Input Metadata allows us to use a data stream that we know matches to rename the fields.

Take Field Descriptions from Right Input Metadata allows us to use a data stream that we know matches to redefine the fields.

10.6 JSON Parse

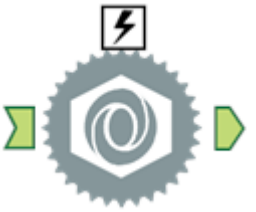


Figure 10-10 – JSON Parse

The *JSON Parse* tool allows us to read JSON formatted files into relational tables.

Group	Input	Output
Developer	JSON formatted data stream	See below

Note: For those unfamiliar with JSON: JSON stands for JavaScript Object Notation, which is a structured data format typically used in data transfer.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream in relational data format.

Properties Window:

The *JSON Parse Configuration* window has three components.

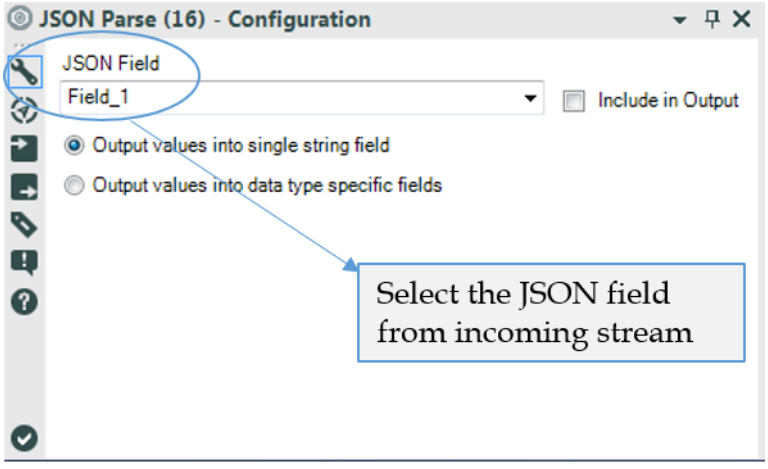


Figure 10-11 – JSON Parse Configuration

- *JSON Field* is the field in the incoming data stream that has JSON data written with one complete record in the cell.
- *Include in Output*, when checked, will maintain the JSON field when it is output from this tool.
- *The Output Values* options allow us to decide if each of the data values should be included in a single string field or in a field defined by the JSON data type.

10.7 Message

Figure 10-12 - Message

The *Message* tool allows us to create a message in the output window.

Group	Input	Output
Developer	Any data stream	Any data stream

Note: Message helps find out where something went wrong and log the settings during a run. It also provides updates about parts of our workflow that have completed.

Application questions can be connected to the top *Black Question Anchor* to use those answers in this tool.

An Action tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Message Configuration* window has five basic components.

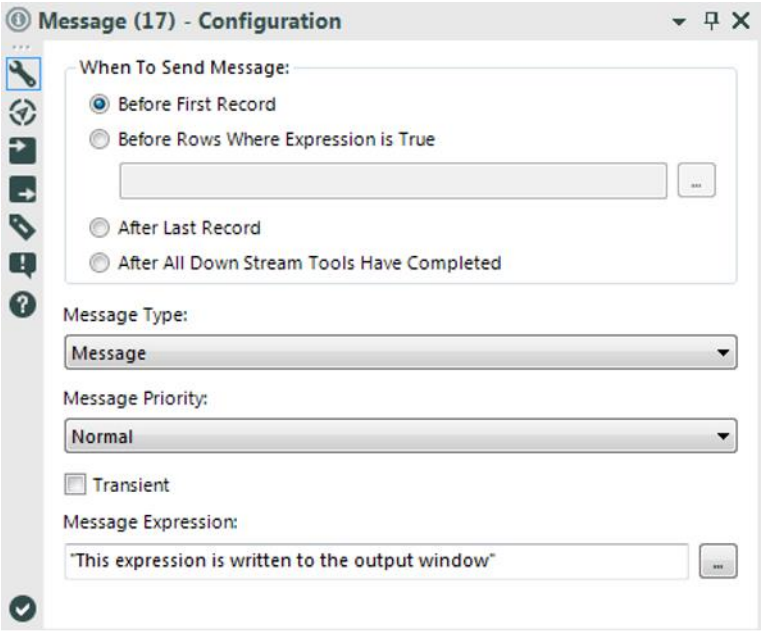


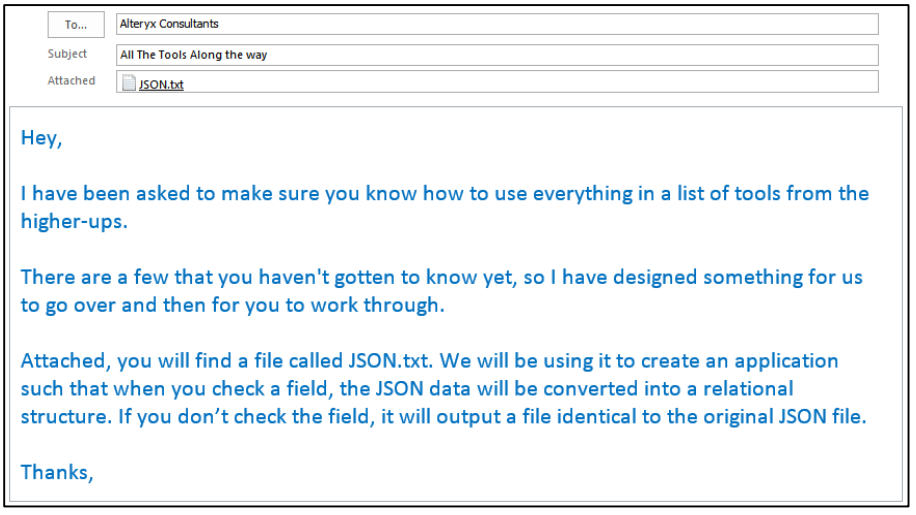
Figure 10-13 – Message Configuration

- *When To Send Message* defines what the trigger for writing out the message is. It can be just before the first record, before every record where a Boolean expression is true, after the last record has passed through the tool, or when the data stream has terminated.
- *Message Type* defines what type of output should be written out and if it should stop the data flow (see Output Window in The Interface for more details).
- *Message Priority* determines what the message would do by default if it were in a macro. Normal will show the message if the tool is in the workflow. Medium will show

the message if the tool is directly in a macro used in the workflow. High will show the message no matter how many macros deep the message occurred.

- *Transient*, when checked, means that if the Message tool writes more than one message, it will replace the single message instead of creating a new one every time.
- *Message Expression* is an expression field that will output the result to the output window.

10.8 All the Tools Along the Way


To... Alteryx Consultants
Subject All The Tools Along the way
Attached JSON.txt

Hey,

I have been asked to make sure you know how to use everything in a list of tools from the higher-ups.

There are a few that you haven't gotten to know yet, so I have designed something for us to go over and then for you to work through.

Attached, you will find a file called JSON.txt. We will be using it to create an application such that when you check a field, the JSON data will be converted into a relational structure. If you don't check the field, it will output a file identical to the original JSON file.

Thanks,

The option for a tool to essentially just copy the file seems a little strange, but it will make a more sense when we start working on the follow-up assignment. We can write the file to a .csv instead of a .txt file at the end of the workflow. Let us start by connecting to the data.

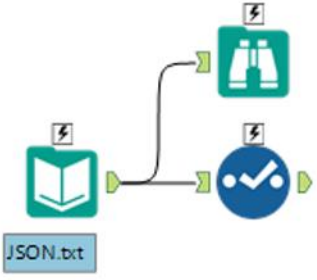


Figure 10-14 – Open JSON File

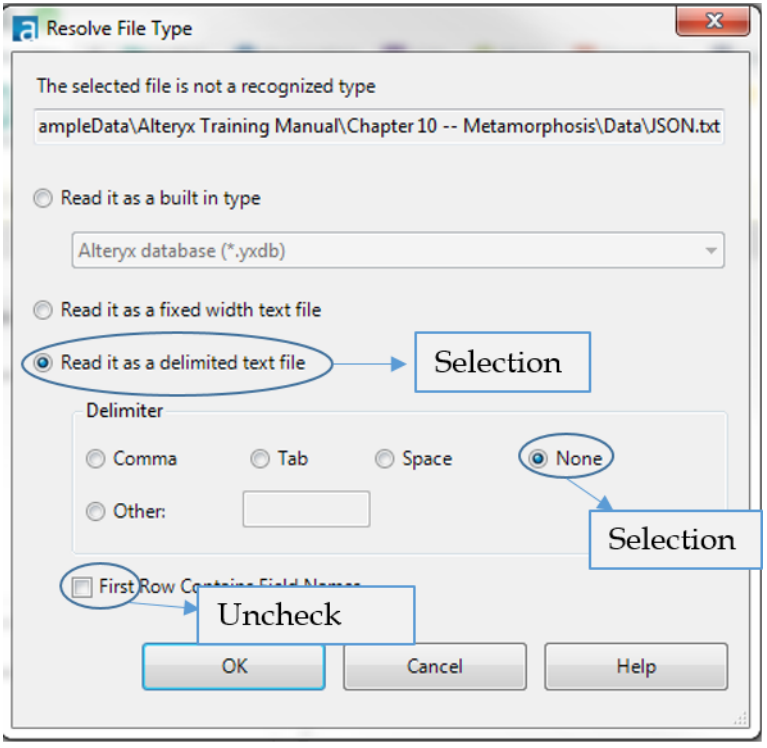


Figure 10-15 – Read JSON File

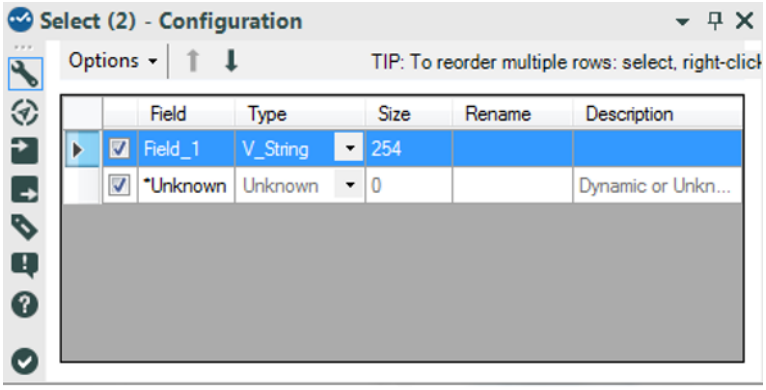


Figure 10-16 – JSON Select

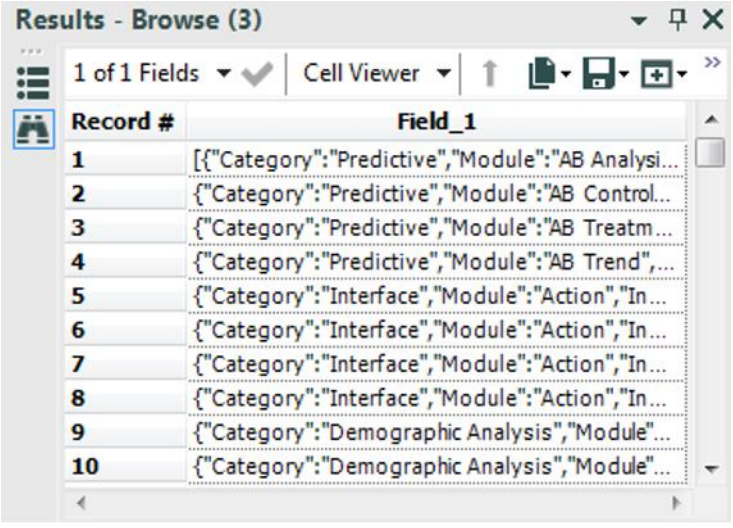


Figure 10-17 – JSON Browse

When we connect to this file, because it is *.txt*, Alteryx needs to know how it should read it. This is by design because file formats like JSON need to be read so that all data is in a single field and ideally one record per row. Because we want these settings, we can choose to have the file delimited with no delimiter so that nothing will break the field. If we take a look at the file, we can see there is no header row. This is because JSON

carries the header names in every record and thus does not need a devoted row.

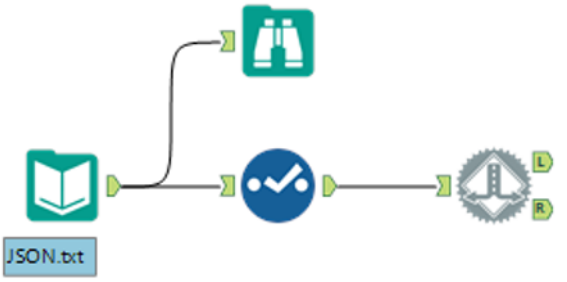


Figure 10-18 – JSON Detour

Our next step will be to add a *Detour* tool. This tool allows us to use a question in order to decide which data stream to follow. The *Detour to the Right* option is so that while we are testing our workflow, we can properly check both sides. By design, the Detour tool detours to the left unless updated.

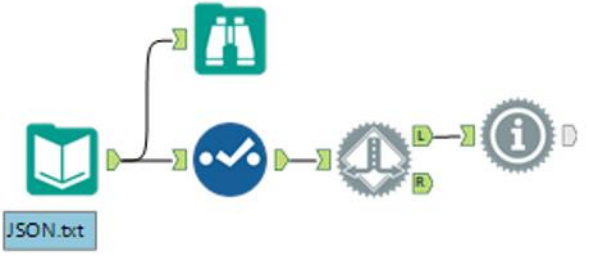


Figure 10-19 – JSON Detour to the Left

Let's start by designing the left (L) data stream. We are going to make this the side where we do not convert the data to a relational structure.

We are going to use *Message* tools to record what will happen to the data. The Message tool allows us to write something to the output window. In this case, we are going to write *The data will remain in JSON format* so that it will be in the output log.

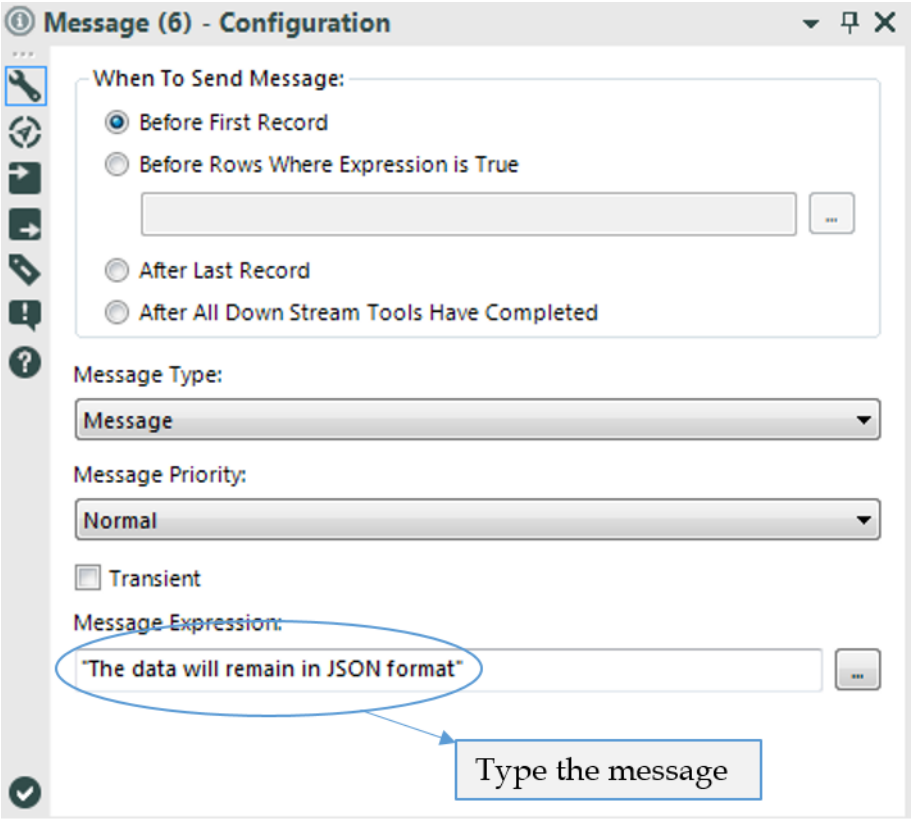


Figure 10-20 – JSON Message Configuration

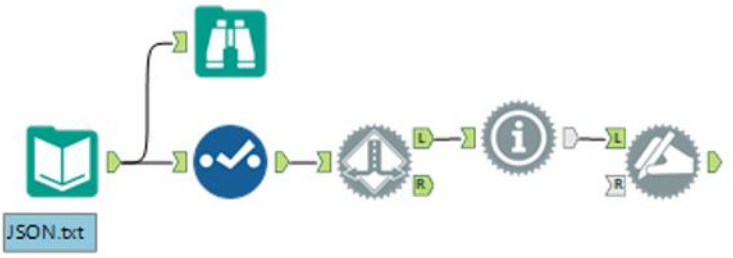


Figure 10-21 – JSON Left data stream

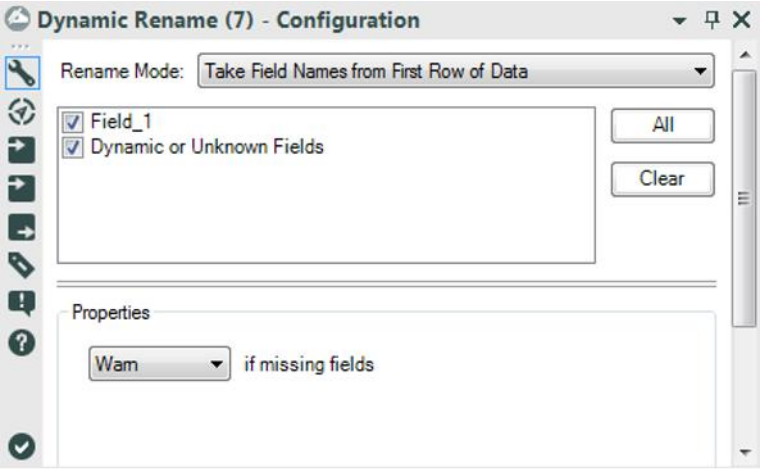


Figure 10-22 – JSON Dynamic Rename

Now let us add a *Dynamic Rename* tool. This is because once we bring the data back together to write, we will need to write the header. Since we don't want a header for this option on the data stream, we will modify it such that the header is actually the first record of data. This will complete all of the data prep we need before rejoining the data streams to be written out.

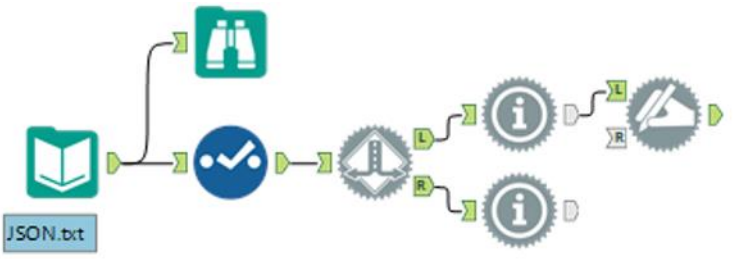


Figure 10-23 – JSON Right data stream

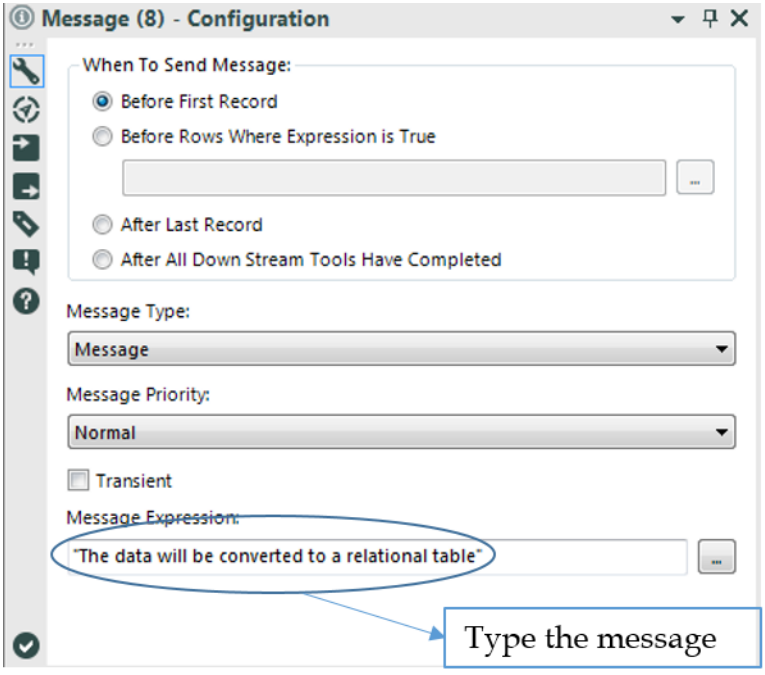


Figure 10-24 – JSON Message configuration for Right Side data stream

We are going to start the right (R) side of the data stream by creating a parallel message to the left that reads *The data will be converted to a relational table*.

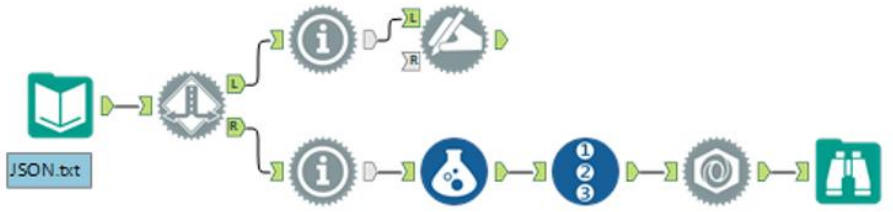


Figure 10-25 - JSON Right side data stream completed

We will introduce a *Formula* tool now and convert each row of the incoming data into an array by introducing '[' and ']' symbols at the start and end of each row.

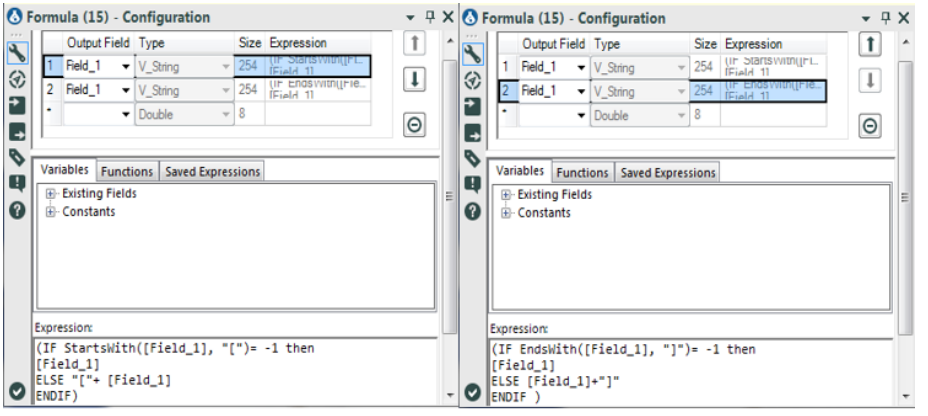


Figure 10-25a – Formula Field Calculation

We then introduce Record ID tool to number each row which will eventually help us in grouping the data as we shall see. Now, we will bring the *JSON Parse* tool onto the canvas. This tool is designed to look at JSON data and convert it into relationally structured data. Unfortunately, if we just put the tool at the end of the data stream, it does not recognize any fields. This is because the *Detour* is diverting all information to the left, including the metadata. In order to configure the tools on this side of the *Detour,* we need to divert the tool to the right.

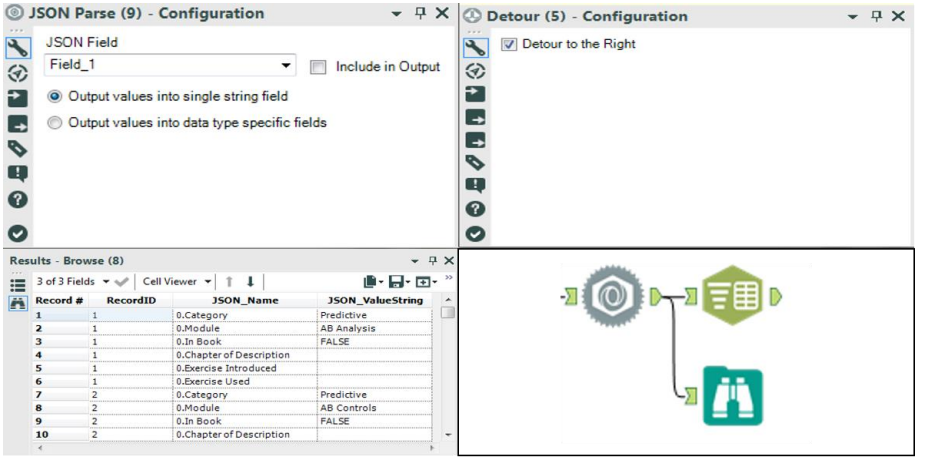


Figure 10-26 – JSON Parse converting data to relationally structured data

Now that we have restructred the data, we should take a look at the result so we know what our next steps should be. We can see the JSON Parse tool has created two fields: *JSON_Name*, which has the record number and field name separated by a period, and *JSON_ValueString*, which has the values associated with each record and variable.

Note: If we wanted to convert data into JSON format using the *JSON Build* tool, this is the format it should be in.

Since we know we want a relational dataset that is easy to work with, we should make it tidy by creating a single column for each variable.

In order to do this, we first need to identify the unique column names by splitting them off of the record number. We can use the *Text To Columns* tool in order to create a separate column for both the record number and the column name so we can use the two of them to restructure the data.

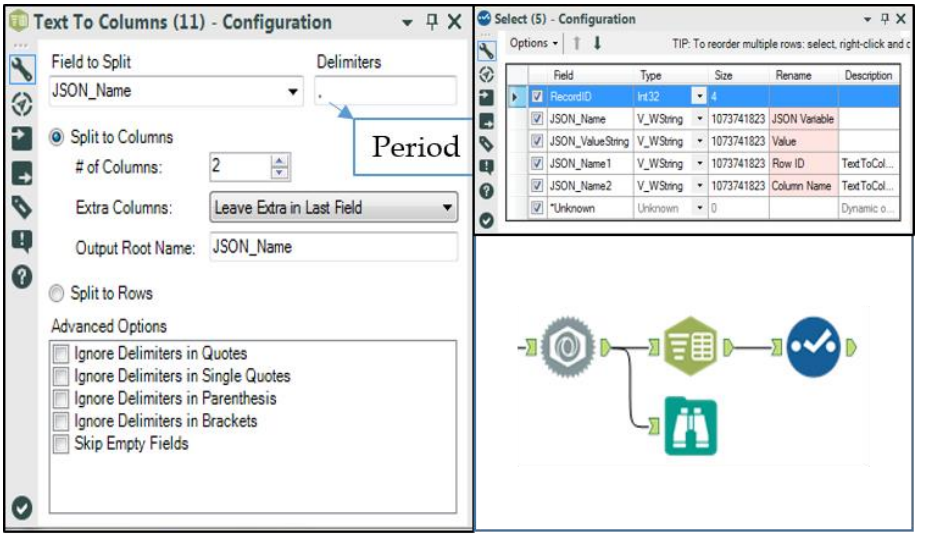


Figure 10-27 – JSON – Splitting the columns and renaming them

In order to make the data easier to understand we should rename our fields.

Figure 10-28 – JSON – Relationally structured data

Now that we have our fields well named, we can pivot the data into a more usable format. We are going to group the data by the *Record ID*, use the *Field Names* as our columns, and use the value field as the data in the intersection. We need to set a methodology. However, since we are using a unique combination of header and grouping fields, we do not need to worry about clashing data.

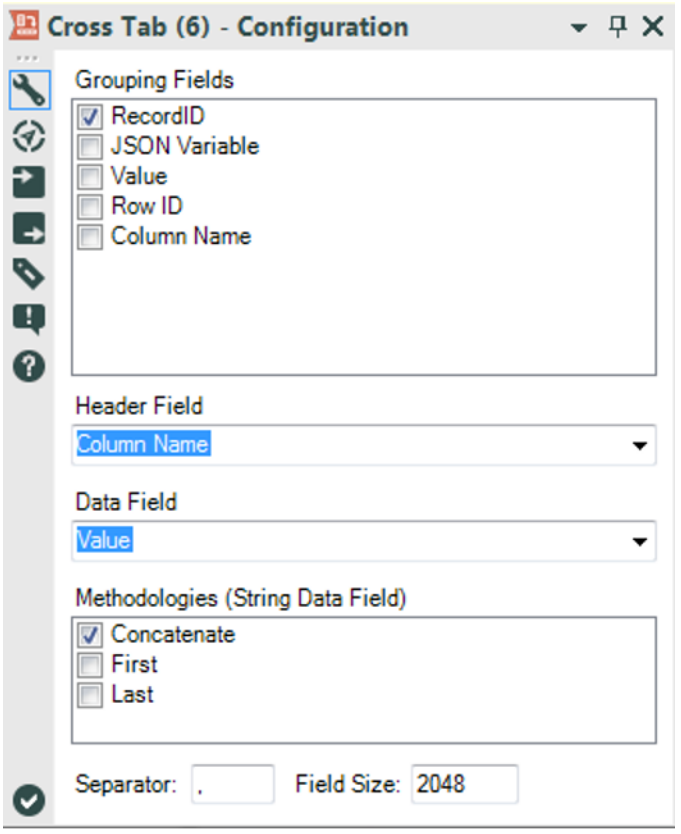
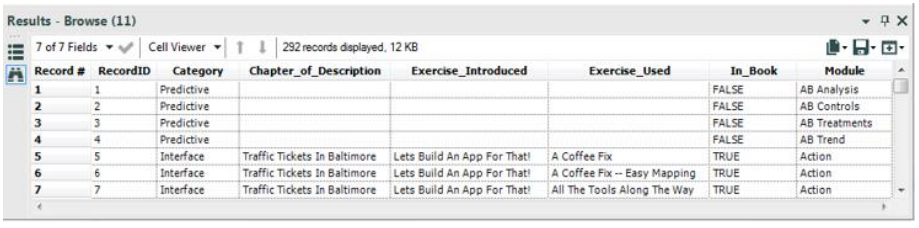


Figure 10-29 – JSON – Data grouping by Record ID



Record #	RecordID	Category	Chapter_of_Description	Exercise_Introduced	Exercise_Used	In_Book	Module
1	1	Predictive				FALSE	AB Analysis
2	2	Predictive				FALSE	AB Controls
3	3	Predictive				FALSE	AB Treatments
4	4	Predictive				FALSE	AB Trend
5	5	Interface	Traffic Tickets In Baltimore	Lets Build An App For That!	A Coffee Fix	TRUE	Action
6	6	Interface	Traffic Tickets In Baltimore	Lets Build An App For That!	A Coffee Fix -- Easy Mapping	TRUE	Action
7	7	Interface	Traffic Tickets In Baltimore	Lets Build An App For That!	All The Tools Along The Way	TRUE	Action

Figure 10-30 – JSON – Grouped dataset

If we take a look at the data, we can see we have a tidy dataset, which means we can end this specialized portion of the workflow and bring the two data streams back together.

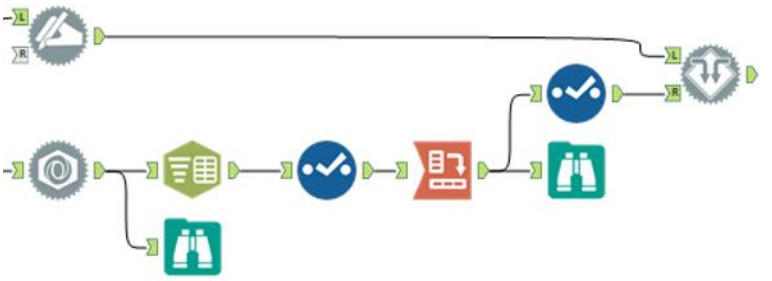


Figure 10-31 – JSON – Bringing both data streams together

Now we can connect both sides of the data stream to a *Detour End* tool to bring them back together so we can use a single output.

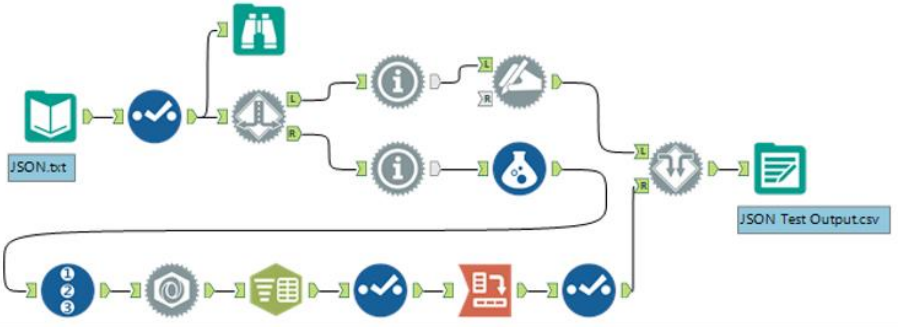


Figure 10-32 – JSON – Using Detour End Tool to bring together the data streams

Once we add the Output Data tool, we are almost done. As we can see, we have a fully operational data stream. The only problem we have is that we have no way to control if the file will be written as JSON or a relational table.

We are going to add a *Check Box* tool to the workflow so we can give the option of selecting between converting the data or not. If we connect the question directly to the *Detour* tool, the action will be created exactly how we want it to work, except that

we should take a look at the behavior of this action because it doesn't just change the direction, like implied by the action type.

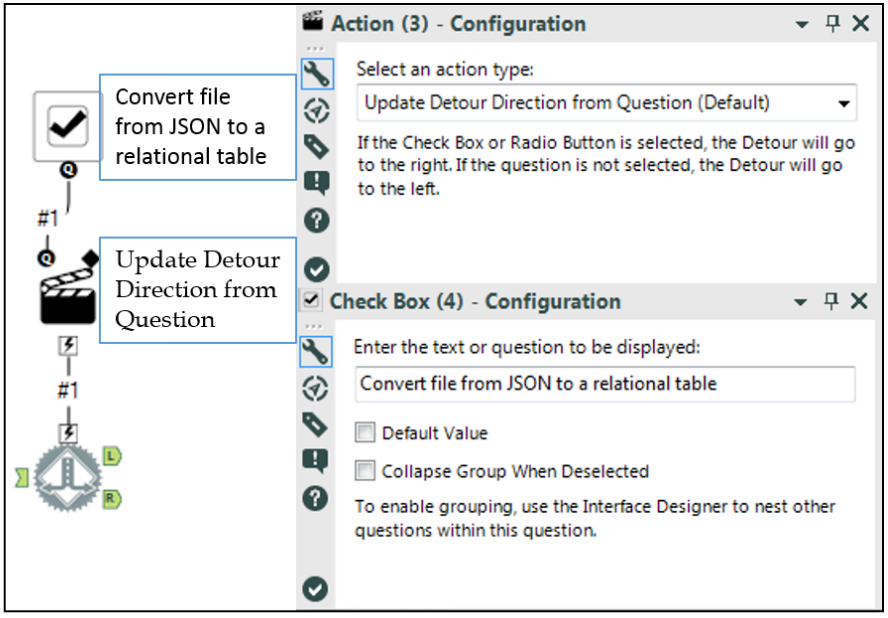


Figure 10-33 – JSON – Adding Check Box to provide an option for conversion

Any time we are running the Detour tool as an application, the default path is to go to the left. When we have an action to update this behavior, it will change the result to divert to the right. We should name the question to reflect this behavior.

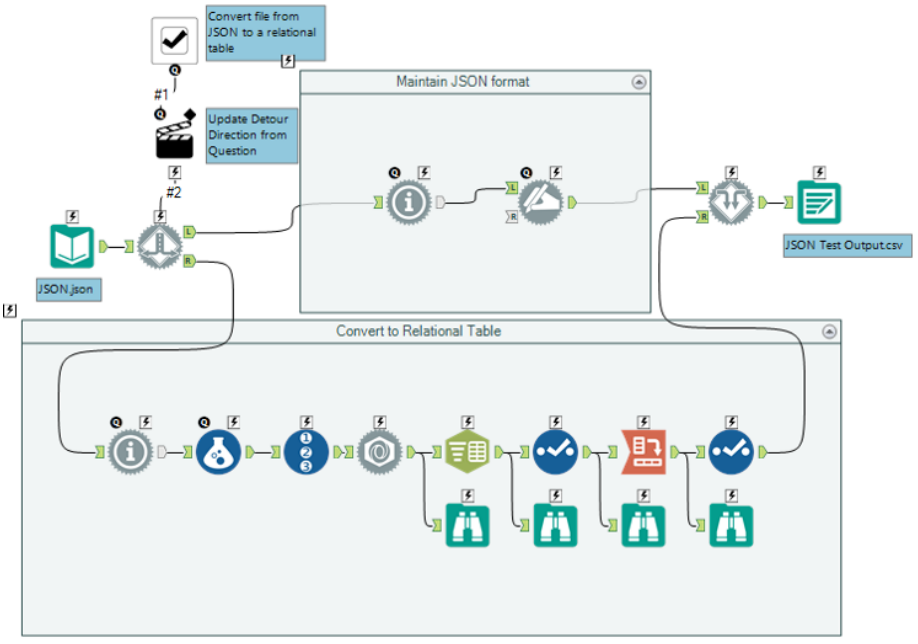
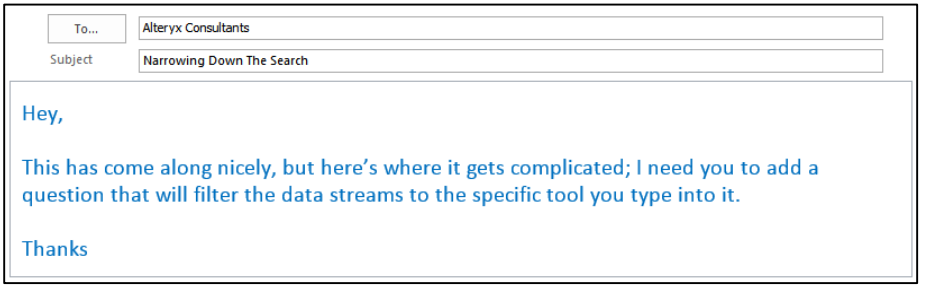


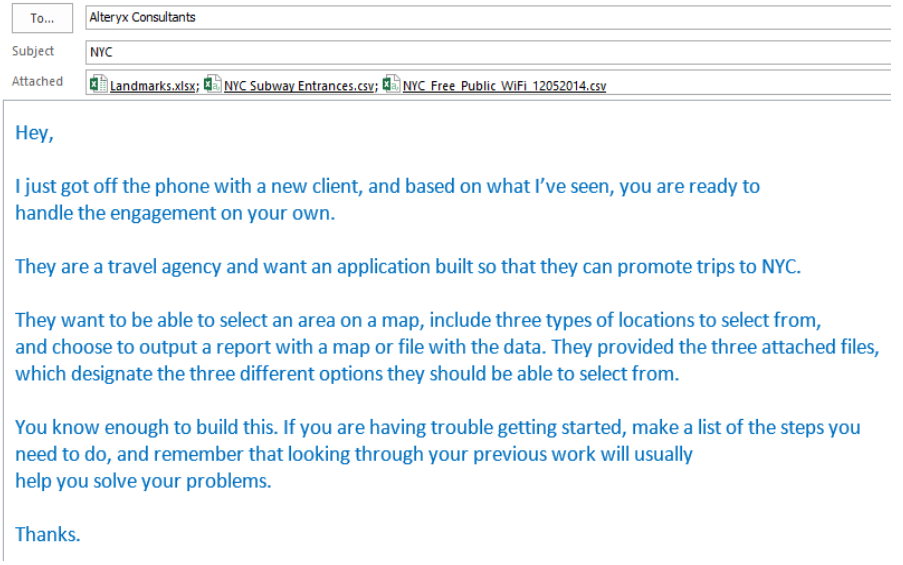
Figure 10-34 – JSON data stream on completion

10.9 Narrowing Down the Search



CHAPTER 11
NYC

Congratulations! You have made it through all of the basic training and are ready to take on a client of your own. Let me know how it goes when you're done.


To... Alteryx Consultants

Subject NYC

Attached Landmarks.xlsx; NYC Subway Entrances.csv; NYC_Free_Public_WiFi_12052014.csv

Hey,

I just got off the phone with a new client, and based on what I've seen, you are ready to handle the engagement on your own.

They are a travel agency and want an application built so that they can promote trips to NYC.

They want to be able to select an area on a map, include three types of locations to select from, and choose to output a report with a map or file with the data. They provided the three attached files, which designate the three different options they should be able to select from.

You know enough to build this. If you are having trouble getting started, make a list of the steps you need to do, and remember that looking through your previous work will usually help you solve your problems.

Thanks.

CHAPTER 12
Self-Guided Solutions

Readers are strongly encouraged not to use these solutions when solving the self-guided examples. There are many approaches to answering these problems, and figuring out how to do it on your own is the goal of these exercises. These screenshots of solutions are provided so that if you're stuck, you can refer to the logic of how the examples direct you to a solution.

12.1 The Games: The Almighty Dollar

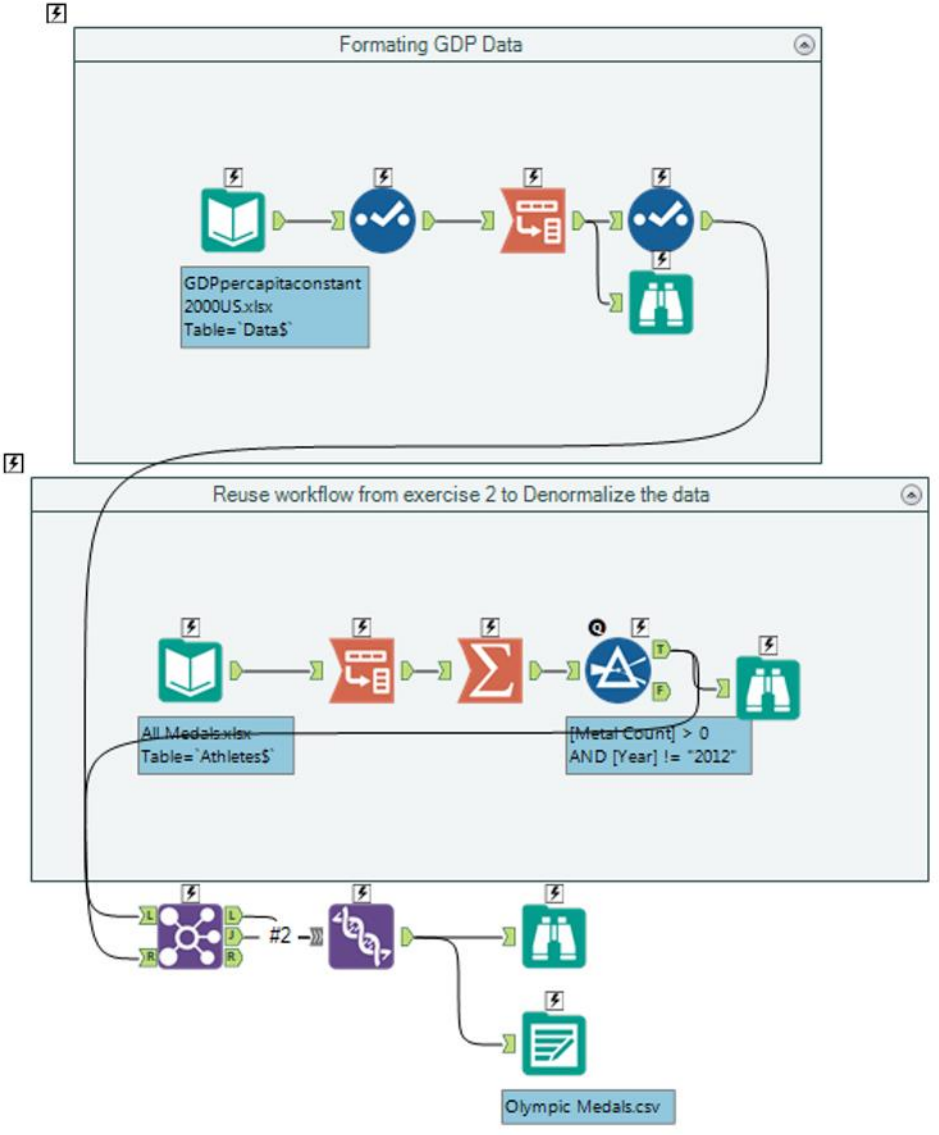


Figure 12-1 – Almighty Dollar

12.2 Unisex Baby Names: What's In a Name?

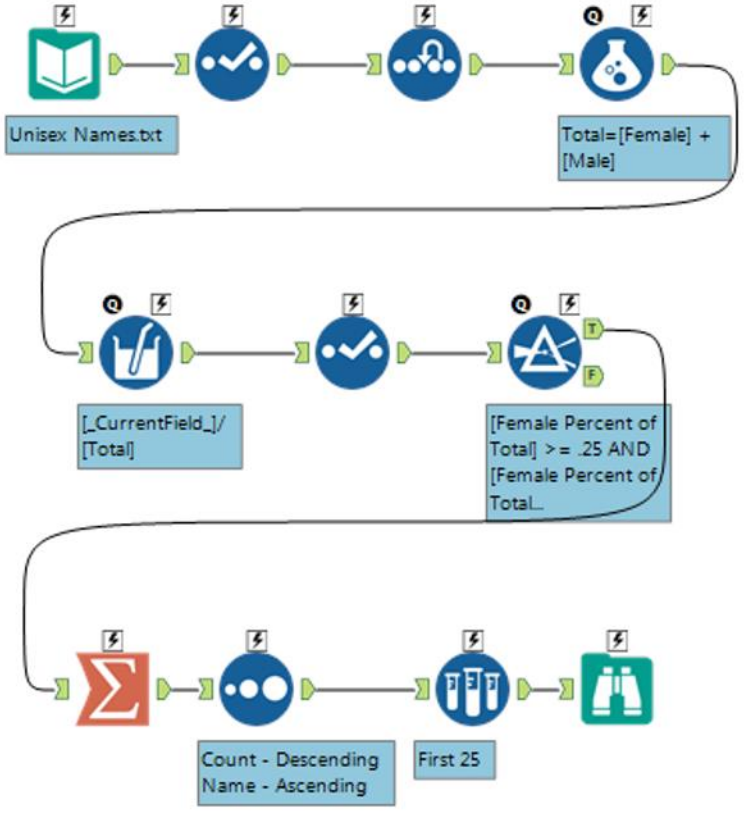


Figure 12-2 – What's in a name?

12.3 The Direct Approach: Where, Oh Where Have My Three Files Gone?

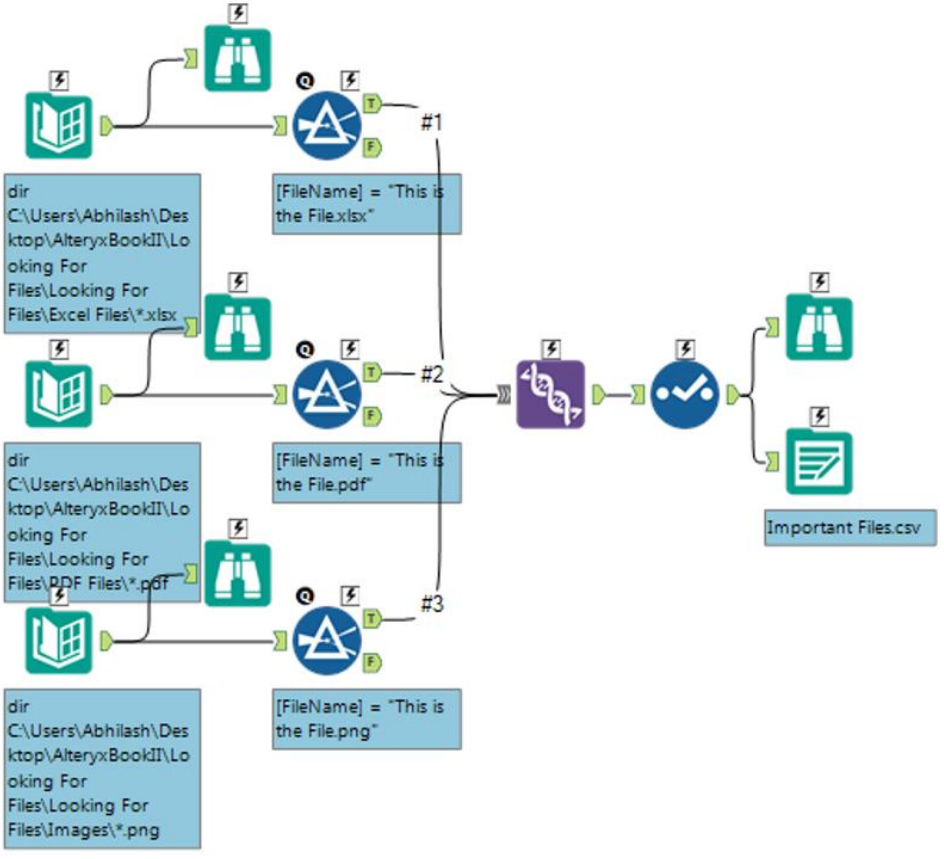


Figure 12-3 – Where, Oh Where have the three files gone?

12.4 Cultural Musing: Culturally Divided

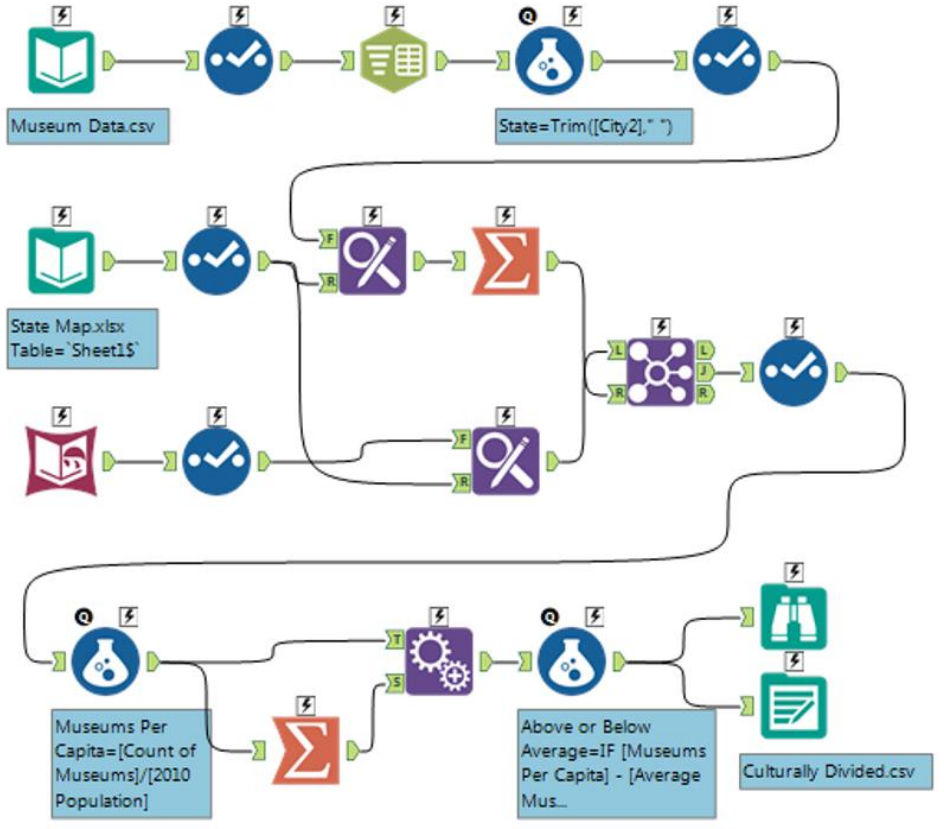


Figure 12-4 – Culturally Divided

12.5 The Sport Report: FOOTBALL!

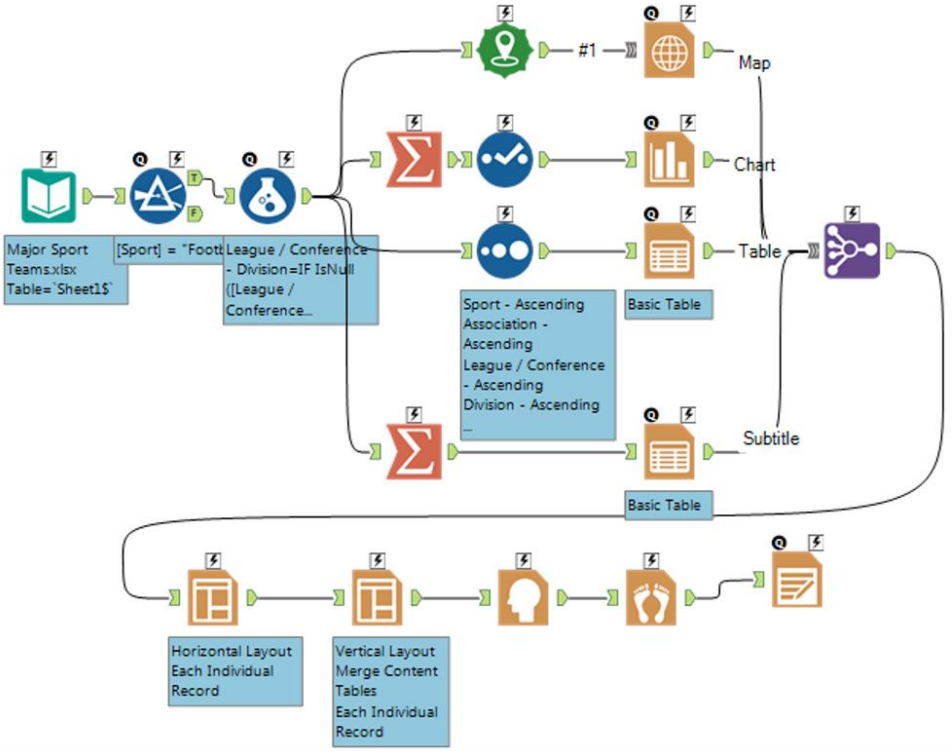


Figure 12-5 - Football

12.6 Expensive Beauty Products: More Flags

Since the complete workflow is large, it is broken down into segments for better understanding.

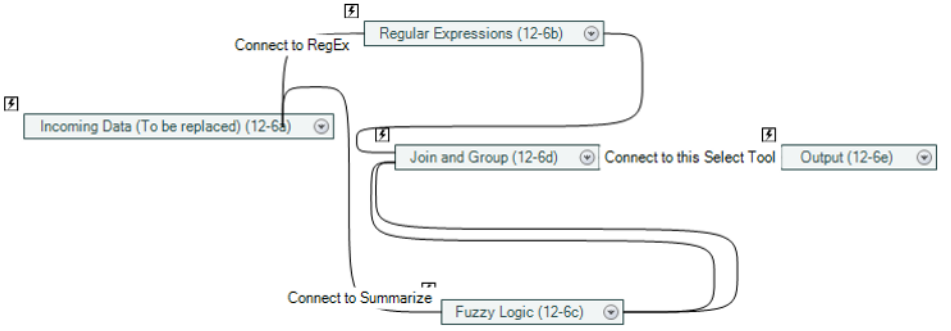


Figure 12-6 – More Flags

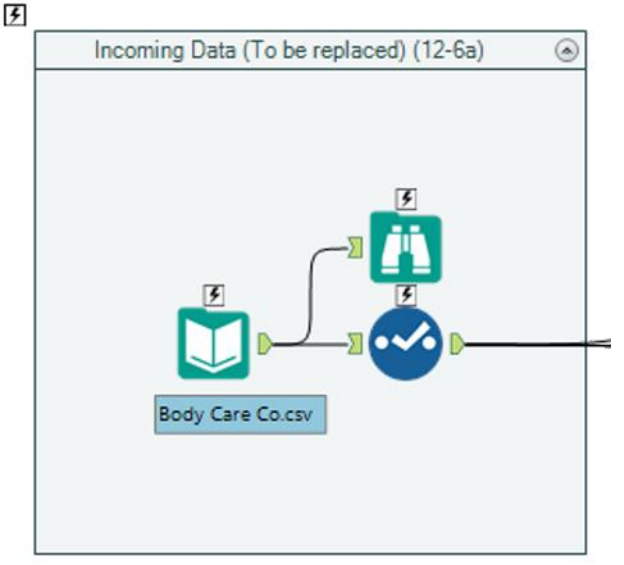


Figure 12-6a – Incoming Data

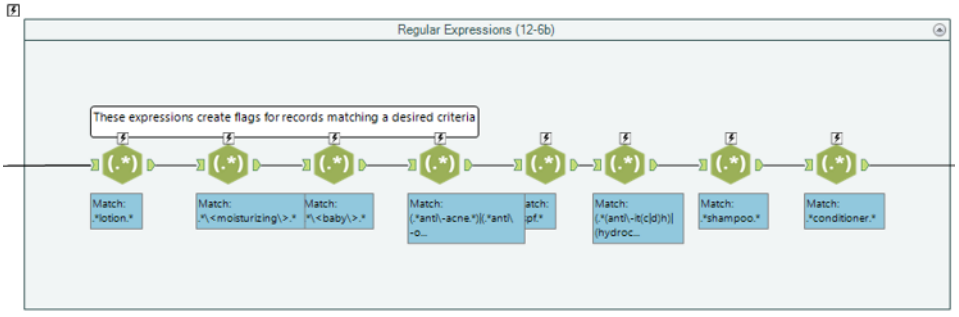


Figure 12-6b – Regular Expressions

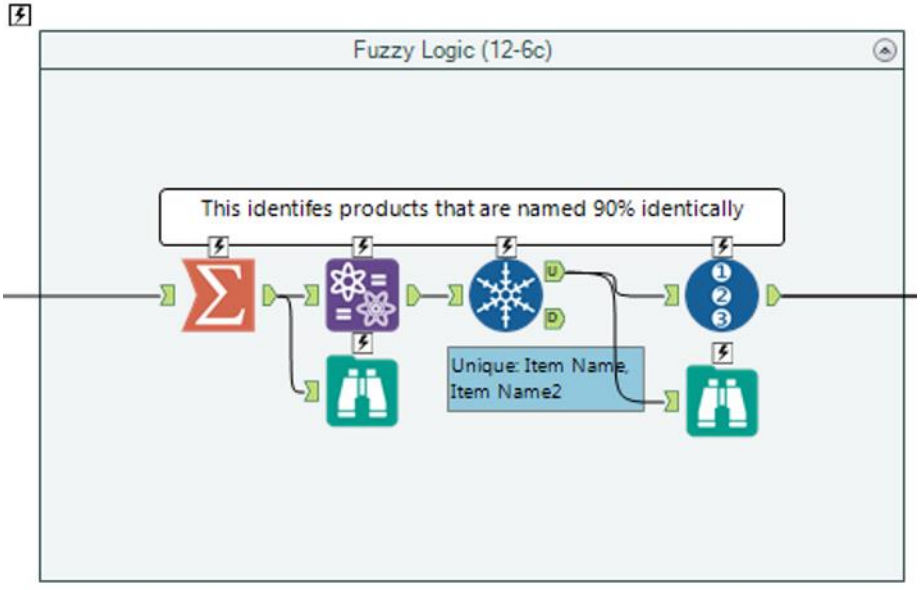


Figure 12-6c – Fuzzy Logic

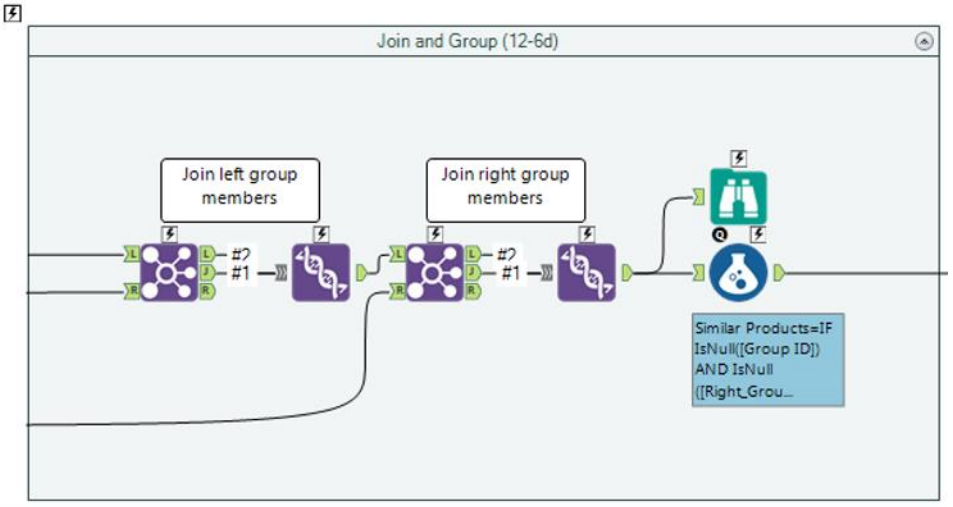


Figure 12-6d – Join & Group

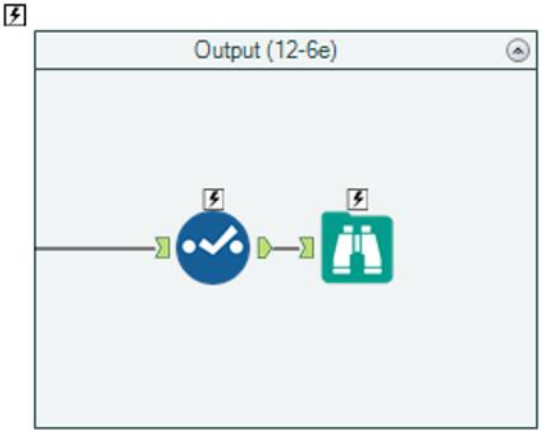


Figure 12-6e - Output

12.7 Applications Wanted: To Summarize or Not to Summarize: That is the Question.

Since the complete workflow is large, it is broken down into segments for better understanding.

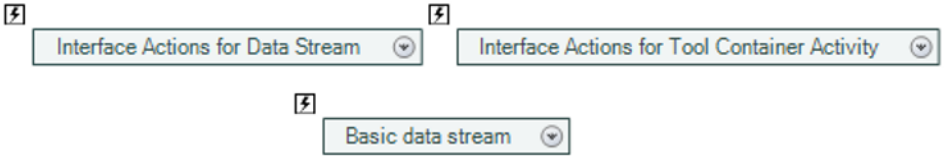


Figure 12-7 – To Summarize or not to Summarize

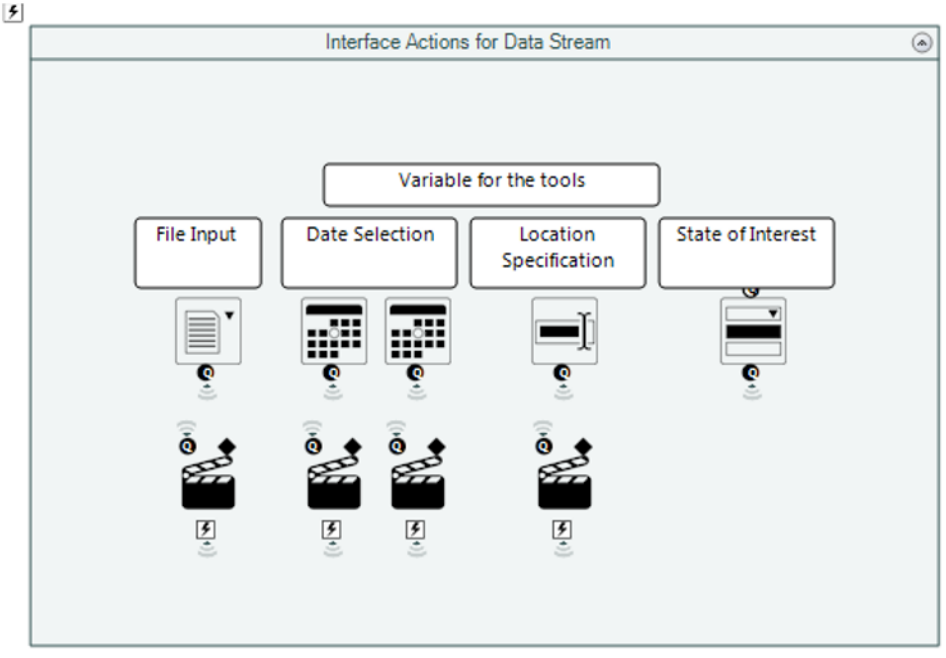


Figure 12-7a – Interface Actions for Data Stream

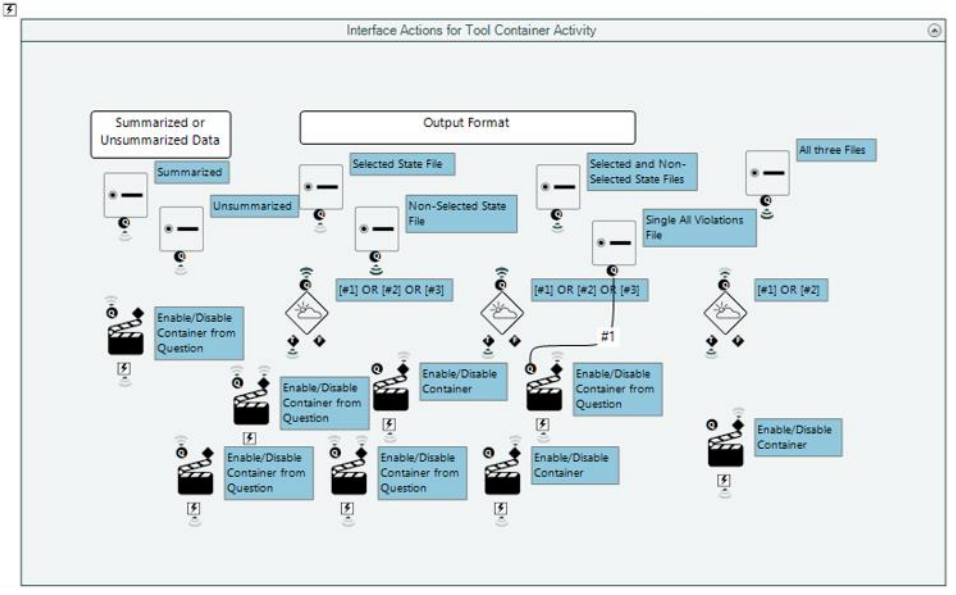


Figure 12-7b – Interface Actions for Tool Container Activity

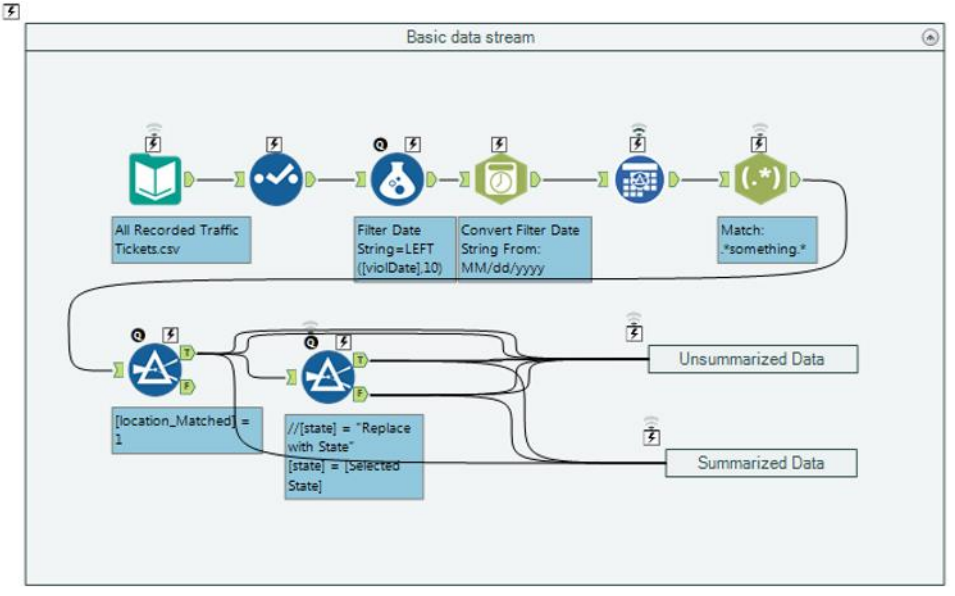


Figure 12-7c – Basic data stream

12.8 Where's The Joe? Zip 5 Total Population Macro and Workflow

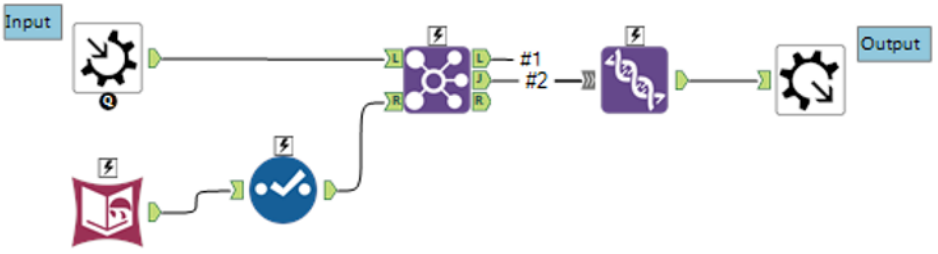


Figure 12-8a – Who wants coffee? - Macro

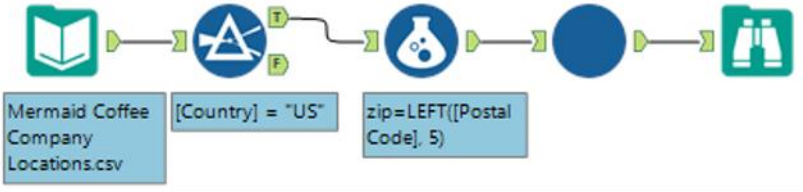


Figure 12-8b – Who wants coffee? - Workflow

12.9 Metamorphosis: Narrowing Down the Search

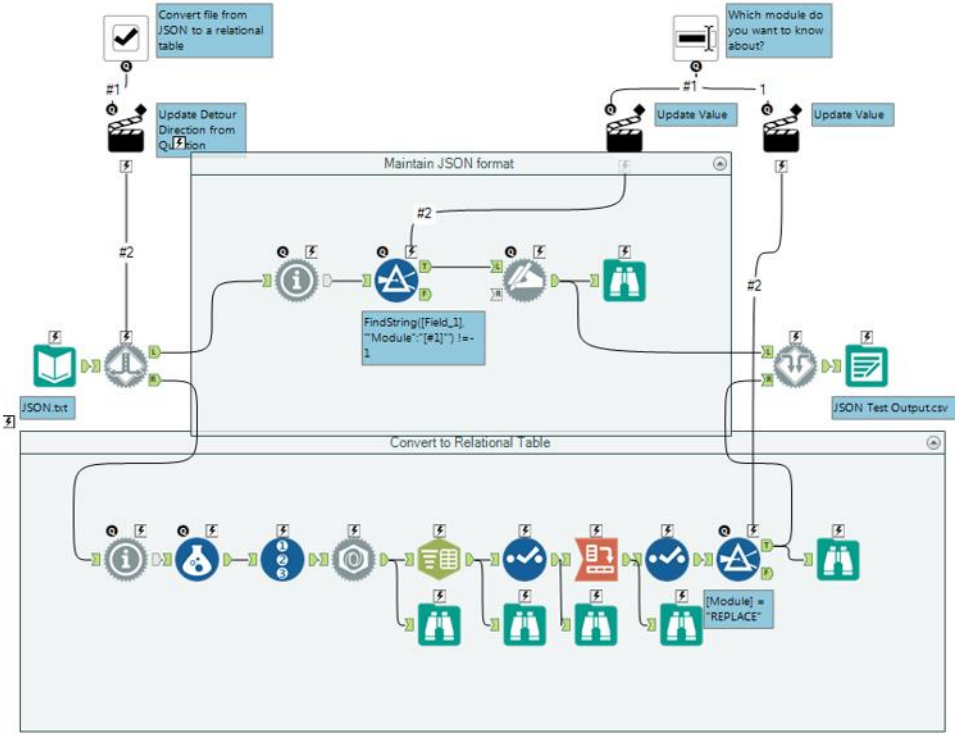


Figure 12-9 - Metamorphosis

12.10 Capstone: NYC

Since the entire workflow is large, it is broken down into segments for better understanding.

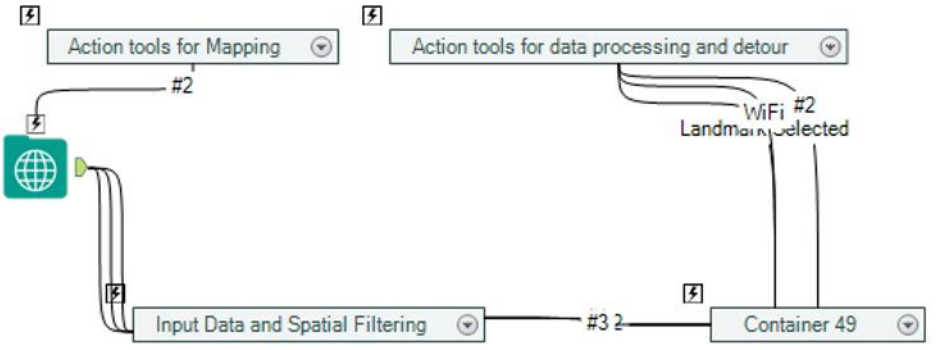


Figure 12-10 - NYC

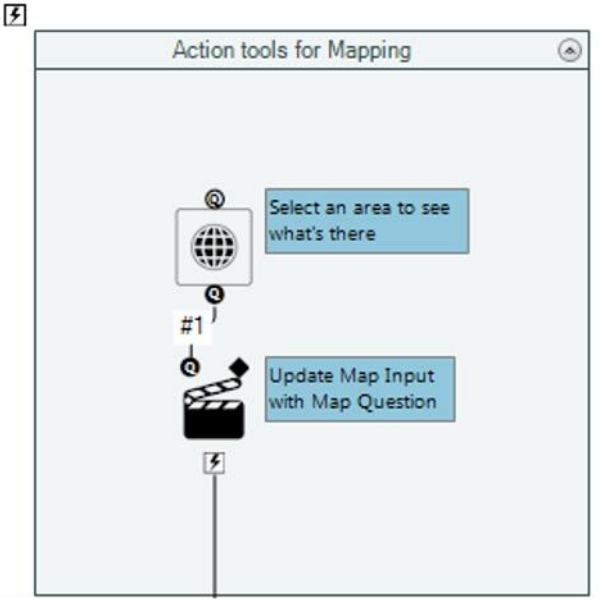


Figure 12-10a – Action tools for mapping

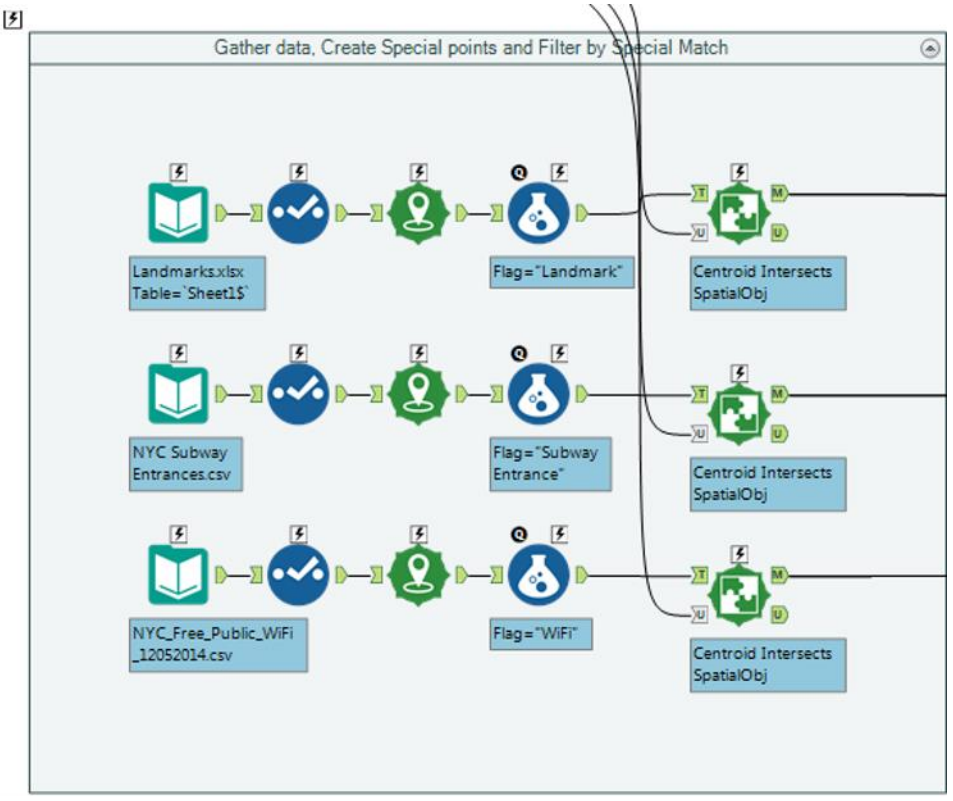


Figure 12-10b – Gather data, Create Special Points and Filter by Special Match

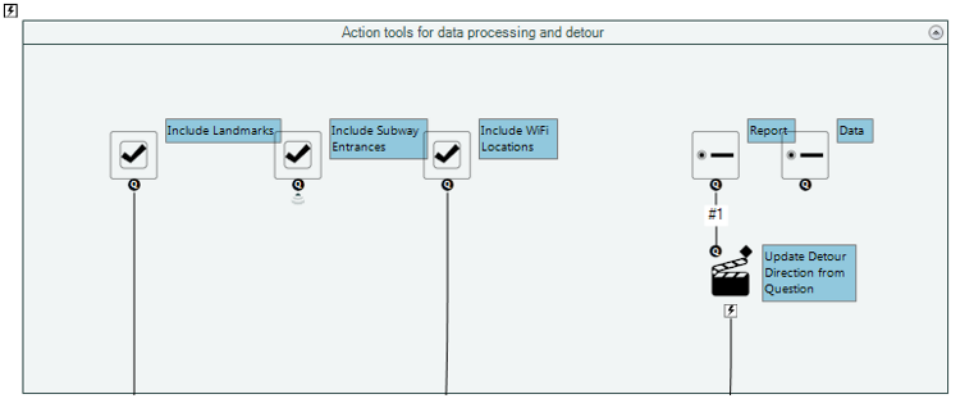


Figure 12-10c – Action tools for Data processing and Detour

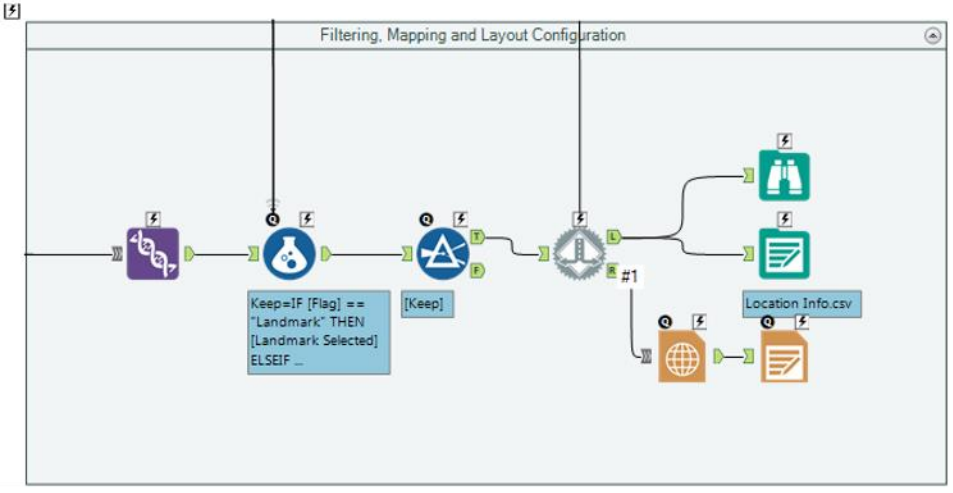


Figure 12-10d – Filtering, Mapping and Layout Configuration

Appendices

Appendix A – File Types

.aws	Allocate Workspace
.cydb	Calgary database
.pcxml	XML data output type that allows the end user to select from multiple file types for a final output when using Alteryx Gallery
.yxdb	Alteryx database
.yxft	Alteryx Header File
.yxlc	Alteryx License File
.yxmc	Alteryx Macro File
.yxmd	Alteryx Workflow File
.yxwv	Alteryx Analytic App Values File which allows predefined selections to be made.
.yxwz	Alteryx Analytic App File
.yxzp	Packaged Alteryx File

Appendix B – Hot Keys

Ctrl+Alt+B	Show/Hide Toolbar
Ctrl+Alt+D	Show/Hide Interface Designer
Ctrl+Alt+O	Show/Hide Output
Ctrl+Alt+P	Show/Hide Properties
Ctrl+Alt+T	Show/Hide Tool Palette
Ctrl+Alt+V	Show/Hide Overview
Ctrl+C	Copy
Ctrl+F	Fine
Ctrl+F4	Close workflow
Ctrl+N	New workflow
Ctrl+O	Open workflow
Ctrl+R	Run/Cancel Workflow
Ctrl+S	Save workflow
Ctrl+Shift+B	Add a Browse tool after all selected tools that a Browse tool can be connected to
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Y	Redo
Ctrl+Z	Undo
Ctrl++	Vertically align selected tools
Ctrl+-	Horizontally align selected tools
Del (Delete)	Deletes selected part of workflow
F1	Open Help menu
F5	Refresh Configuration

Appendix C – Downloads/Content

US 2010 Census SF1
http://downloads.alteryx.com/data.html

USGS North America Map
http://downloads.alteryx.com/data.html

Visual Analytics Kit (Tableau)
http://alteryx.com/kit

Visual Analytics Kit (Qlik)
http://pages.alteryx.com/VisualAnalyticsKitforQlik_Reg-LP.html

Alteryx Gallery
https://gallery.alteryx.com

Appendix D – Field Types

Type	Class	Description
Bool	Boolean	A field with two values, True and False
Byte	Numeric	A field with integers between 0 and 255
Int16	Numeric	A field with integers between -32,768 and 32,767
Int32	Numeric	A field with integers between -2,147,483,648 and 2,147,483,647
Int64	Numeric	A field with integers between –9,223,372,036,854,775,808 and 9,223,372,036,854,775,807
Fixed Decimal	Numeric	A field with a specific width including the number of decimal places
Float	Numeric	A field with up to 7 digits of accuracy between -3.4^38 and 3.4^38
Double	Numeric	A field with up to 15 digits of accuracy between -1.7^308 and 1.7^308
String	String	A field with up to 8192 characters
WString	String	A field with up to 8192 characters that will accept Unicode characters
V_String	String	A field that will adjust the size depending on the length of the strings in it

V_WString	String	A field that will adjust the size depending on the length of the strings in it and will accept Unicode characters
Date	String	A field in the format "yyyy-mm-dd"
Time	String	A field in the format "hh:mm:ss"
DateTime	String	A field in the format "yyyy-mm-dd hh:mm:ss"
Blob	Blob	A field with image or sound files
SpatialObj	Blob	A field with points, lines, polylines, or polygons.

Appendix E- Properties Window

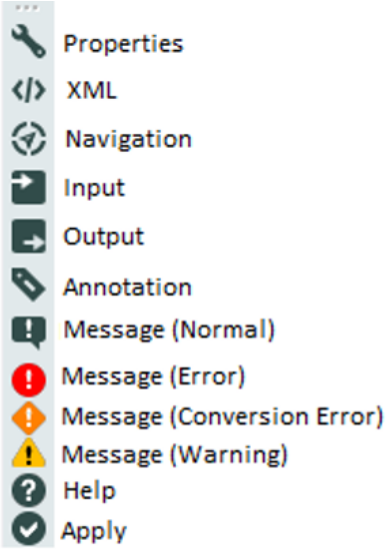


Figure 12-11 – Properties Window

Appendix F – Boolean Expressions

Expression	Meaning
A = B	Expression A equals expression B
A != B	A does not equal B
A > B	A is greater than B
A >= B	A is greater than or equal to B
A < B	A is less than b
A <= B	A is less than or equal to B
A	A is True
NOT A	A is False
A IN B	A is in the set of B things
A NOT IN B	A is not in the set of B things
A OR B	Expression A is True or expression B is True or both expressions are True
A AND B	Expression A is True and expression B is True
A OR (B AND C)	Expression A is True or expression B and expression C are both True or expression A and expression B and C are True
A AND (B OR C)	Expression A is True and expression B, expression C, or expressions B and C are True

Basic Boolean expressions can be combined to create more complex logic by replacing A, B or C with any basic Boolean Expression.

Appendix G – Data Components

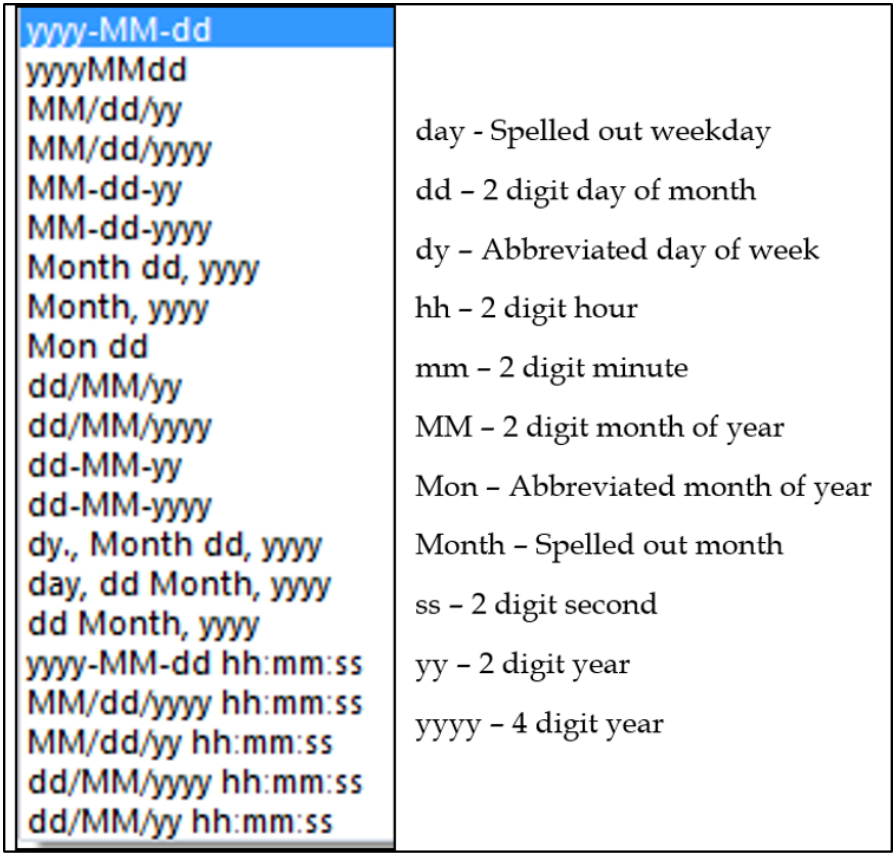


Figure 12-12 – Date Components

Appendix H – Date/Time Units

Keywords
Years
Months
Days
Hours
Minutes
Seconds

Appendix I – RegEx Cheat Sheet

.	Any Character
$	End of a line
()	Marked Group
(?:)	Unmarked Group
*	Repeat the previous thing 0 or more times
+	Repeat the previous thing 1 or more times
[ ]	A set to be used for optional lists of single characters
[^]	A set to be used for optional lists of single characters to be excluded
[[:alpha:]]	Any letter
\	Escape the following symbol
\<	Beginning of a word
\>	End of a word
\d	Digit
\l	Lowercase letter
\n	New line character
\s	Space
\t	Tab character
\u	Uppercase letter
\w	Word characters
\W	Non-word characters
^	Beginning of a new line or “not” depending on context
|	Or

Appendix J – Action Tool Sets

These tools do not receive connections from an Action tool.

Action	Date	Folder Browse	Numeric Up Down
Checkmark	Drop Down	List Box	Radio Button
Comment	Error Message	Macro Input	Text Box
Condition	Explorer Box	Macro Output	Tree
Control Parameter	File Browse	Map	

All other tools have at least these basic methods

Delete Tool from Workflow
Update Field List
Update Raw XML with Formula
Update Select with Reverse Field Map
Update Value
Update Value with Formula

These tools only have the basic methods.

AB Analysis	Data Stream In	Join In-DB	Report: Comparison
AB Controls	Data Stream Out	Join Multiple	Report: Detail
AB Treatments	DataSift Connector	JSON Build	Report: Rank

AB Trend	Date Filter	JSON Parse	Run Command
Allocate Append	Date Time Now	K-Centroids Cluster Analysis	Running Total
Allocate Input	DateTime	K-Centroids Diagnostics	Sample
Allocate Metainfo	Decision Tree	Layout	Sample In-DB
Allocate Report	Detail Fields	Lift Chart	Scatterplot
Amazon S3 Download	Detour End	Linear Regression	Score
Amazon S3 Upload	Distance	Logistic Regression	Select In-DB
API Output	Distribution Analysis	Make Columns	Select Records
Append Cluster	Dynamic Input	Make Grid	Smooth
Append Fields	Dynamic Rename	Make Group	Spatial Info
ARIMA	Dynamic Replace	Map Legend Builder	Spatial Match
Arrange	Dynamic Select	Map Legend Splitter	Spatial Process
Association Analysis	Email	Marketo Append	Spearman Correlation
Auto Field	ETS	Marketo Input	Spline Model
Base 64 Encoder	Field Info	Marketo Output	Stepwise
Behavior Metainfo	Field Summary	MB Inspect	Street Geocoder

Blob Convert	Filter	MB Rules	Summarize
Blob Output	Filter In-DB	Message	Summarize In-DB
Block Until Done	Find Nearest	Multi-Field Binning	Support Vector Machine
Boosted Model	Find Nearest Neighbors	Multi-Field Formula	Table
Browse	Find Replace	Multi-Row Formula	Tableau Workbook Macro
Browse Data In-DB	Footer	Naïve Bayes Classifier	Test
Buffer	Forest Model	Nested Test	Test of Means
Business Match (US)	Formula	Neural Network	Text To Columns
Calgary Input	Formula In-DB	Non Overlap Drivetime	Throttle
Calgary Join	Foursquare Search	Overlay	Tile
Calgary Loader	Frequency Table	Oversample Field	Transpose
Canada Geocoder	Fuzzy Match	Parse Address	TS Compare
CASS	Gamma Regression	Pearson Correlation	TS Covariate Forecast
Charting	Generalize	Plot of Means	TS Filler
Cluster Code	Generate Rows	Poly-Build	TS Forecast
Compare Behavior	Gnip Search	Poly-Split	TS Plot

Connect In-DB	Google Analytics	Principal Components	Twitter Search
ConsumerView Matching	HDFS Input	Profile Input	Union
Contingency Table	HDFS Output	Profile Output	Union In-DB
Count Records	Header	R	Unique
Count Regression	Heat Map	Random % Sample	US Geocoder
Create Points	Heat Plot	Record ID	US Zip 9 Coder
Create Profile	Histogram	RegEx	Violin Plot
Create Samples	Household File Matching	Render	Weighted Average
Cross Count	Image	Report Map	Write In-DB
Cross Count Append	Imputation	Report Text	XML Parse
Cross Tab	Join		

The following tools have the basic tools plus

Update Password

Download	MongoDB Input	Salesforce Input	SharePoint List Input
	MongoDB Output	Salesforce Output	SharePoint List Output

The following tools have the basic tools plus

Update Input Data Tool

Blob Input	Input Data

The following tools have the basic tools plus unique methods

Detour	Update Detour Direction from Condition Update Detour Direction from Question
Directory	Update Directory with Folder Browse
Map Input	Update Map Input with Map Question
Output Data	Update Output Data Tool
Select	Update Select with multi-select ListBox
Sort	Update Sort Order
Text Input	Update Cell
Tool Container	Enable/Disable Container From Condition Enable/Disable Container From Question
Trade Area	Update Trade Area Radius

Appendix K – Directory Tool Data

FullPath	The direct path to open the file.
Directory	The direct path to the folder the file is in.
FileName	The actual name and extension of the file.
ShortFileName	A short file name provided for compatibility with legacy software.
CreationTime	This indicates the first time the file was created.
LastAccessTime	This indicates the last time the file was opened.
LastWriteTime	This indicates the last time the file was saved.
Size	This is the total size of the file in bytes.
AttributeArchive	This indicates if the file is archived.
AttributeCompressed	This indicates if the file is compressed.
AttributeEncrypted	This indicates if the file is encrypted.
AttributeHidden	This indicates if the file is hidden.
AttributeNormal	This indicates if the file has normal attributes.
AttributeOffline	This indicates if the data is unavailable.
AttributeReadonly	This indicates if the file is read only.
AttributeResparsePoint	This indicates that the file is associated with a resparse point.
AttributeSparseFile	This indicates that the file is sparse.
AttributeSystem	This indicates that the file is a system file.
AttributeTemporary	This indicates that the file is a temporary file.

Credited Original Data Sources

The Games

- Medals:
  - Access through Tableau Public and assumed to be in the public domain https://public.tableau.com/profile/a.m.5517#!/vizhome/OlympicMedals2000-2012_0/Dashboard2 published by A.M.
- Nobel Laureates:
  - Access through Tableau Public and assumed to be in the public domain: http://tableausoftware.co.nz/public/gallery/nobelprizes published by Ross Perez
- GDP per Capita:
  - Accessed through GitHub https://github.com/cschin/ipython_d3_mashup/blob/master/ipython_13_vis_example/gm_data/GDPpercapitaconstant2000US.csv published by **Jason Chin**

Unisex Baby Names

- Unisex Names:
  - Access through Tableau Public and assumed to be in the public domain http://tableausoftware.co.nz/public/gallery/unisex-baby-names published by Steve Ruble

The Direct Approach

- Created by Author
- Special Thanks to http://watchout4snakes.com/ for use of their random word generator in creating the names of the folders in the exercise Where, Oh Where Have The Three Files Gone?

Cultural Musings

- GDP per Capita:
  - List of U.S. states by GDP per capita. (2015, March 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 00:01, May 12, 2015, fromhttp://en.wikipedia.org/w/index.php?title=List_of_U.S._states_by_GDP_per_capita&oldid=653869588
- Museum Data:
  - Data retrieved from Data.Gov (http://data.gov) Direct Link: http://catalog.data.gov/dataset/museums-universe-data-file-mudf-fy-2014-3rd-quarter/resource/d4b36e0f-1d9d-4c2a-8882-138c15763717
- State Map:
  - Created by Author

Sport Report

- Major US Sport Teams:
  - The author created this data using Wikipedia pages and tools
    - Primary Wikipedia Article

- List of professional sports teams in the United States and Canada. (2015, March 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:34, January 7, 2015, from http://en.wikipedia.org/w/index.php?title=List_of_professional_sports_teams_in_the_United_States_and_Canada&oldid=650502141

    - The Geocoded locations of each of the sites were accessed by using the coordinates link embedded in their web pages with utilizes a Wikimedia Tool called GeoHack which can be accessed directly at http://tools.wmflabs.org/geohack/

- Turner Field. (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:02, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Turner_Field&oldid=653079052
- Marlins Park. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Marlins_Park&oldid=649550406
- Citi Field. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Citi_Field&oldid=653482419
- Citizens Bank Park. (2015, March 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Citizens_Bank_Park&oldid=649435153
- Nationals Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Nationals_Park&oldid=655186324
- Wrigley Field. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:04, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Wrigley_Field&oldid=655275853
- Great American Ball Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:04, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Great_American_Ball_Park&oldid=655285631
- Miller Park (Milwaukee). (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:04, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Miller_Park_(Milwaukee)&oldid=655121188
- PNC Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:05, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=PNC_Park&oldid=654838712
- Busch Stadium. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:05, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Busch_Stadium&oldid=649550165
- Chase Field. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:05, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Chase_Field&oldid=655057408
- Coors Field. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:09, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Coors_Field&oldid=654538824
- Dodger Stadium. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:09, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Dodger_Stadium&oldid=654869845
- Petco Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:10, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Petco_Park&oldid=654538750
- AT&T Park. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:10, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Park&oldid=654538692

- Oriole Park at Camden Yards. (2015, March 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Oriole_Park_at_Camden_Yards&oldid=651898983
- Fenway Park. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Fenway_Park&oldid=653504103
- Yankee Stadium. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Yankee_Stadium&oldid=654863343
- Tropicana Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Tropicana_Field&oldid=655160166
- Rogers Centre. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:12, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Rogers_Centre&oldid=654913445
- U.S. Cellular Field. (2015, March 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:12, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=U.S._Cellular_Field&oldid=649856024
- Progressive Field. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:12, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Progressive_Field&oldid=653499953
- Comerica Park. (2015, March 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Comerica_Park&oldid=651832995
- Kauffman Stadium. (2015, March 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Kauffman_Stadium&oldid=650377832
- Target Field. (2015, February 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Target_Field&oldid=647933510
- Minute Maid Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Minute_Maid_Park&oldid=655303381
- Angel Stadium of Anaheim. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Angel_Stadium_of_Anaheim&oldid=654460841
- O.co Coliseum. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=O.co_Coliseum&oldid=653612043
- Safeco Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Safeco_Field&oldid=655148228
- Globe Life Park in Arlington. (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Globe_Life_Park_in_Arlington&oldid=652959003
- TD Garden. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=TD_Garden&oldid=653403579
- Barclays Center. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Barclays_Center&oldid=654631943
- Madison Square Garden. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Madison_Square_Garden&oldid=654828742
- Wells Fargo Center (Philadelphia). (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Wells_Fargo_Center_(Philadelphia)&oldid=649738895

- Air Canada Centre. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Air_Canada_Centre&oldid=655130412
- United Center. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=United_Center&oldid=655268263
- Quicken Loans Arena. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Quicken_Loans_Arena&oldid=655272955
- The Palace of Auburn Hills. (2015, February 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=The_Palace_of_Auburn_Hills&oldid=648272487
- Bankers Life Fieldhouse. (2014, December 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Bankers_Life_Fieldhouse&oldid=640160898
- BMO Harris Bradley Center. (2015, February 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=BMO_Harris_Bradley_Center&oldid=645646883
- Philips Arena. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Philips_Arena&oldid=654538478
- Time Warner Cable Arena. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Time_Warner_Cable_Arena&oldid=653373673
- American Airlines Arena. (2015, February 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=American_Airlines_Arena&oldid=646426983
- Amway Center. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Amway_Center&oldid=655268871
- Verizon Center. (2015, March 14). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Verizon_Center&oldid=651299892
- Pepsi Center. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Pepsi_Center&oldid=653560944
- Target Center. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Target_Center&oldid=655045980
- Chesapeake Energy Arena. (2015, March 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Chesapeake_Energy_Arena&oldid=650053351
- Moda Center. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Moda_Center&oldid=652902770

- EnergySolutions Arena. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=EnergySolutions_Arena&oldid=654209905
- Oracle Arena. (2015, February 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Oracle_Arena&oldid=648884284
- Staples Center. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Staples_Center&oldid=654943633
- Talking Stick Resort Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:19, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Talking_Stick_Resort_Arena&oldid=654788366
- Sleep Train Arena. (2015, February 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:19, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Sleep_Train_Arena&oldid=648040550
- AT&T Center. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:40, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Center&oldid=654184289
- Smoothie King Center. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:40, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Smoothie_King_Center&oldid=653411690
- FedExForum. (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:40, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=FedExForum&oldid=649655797
- Toyota Center. (2015, February 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:41, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Center&oldid=649249468
- American Airlines Center. (2015, March 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:41, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=American_Airlines_Center&oldid=654000621
- Sleep Train Arena. (2015, February 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:19, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Sleep_Train_Arena&oldid=648040550
- Philips Arena. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Philips_Arena&oldid=654538478
- Allstate Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Allstate_Arena&oldid=654835689
- Mohegan Sun Arena. (2015, March 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Mohegan_Sun_Arena&oldid=650179511
- Bankers Life Fieldhouse. (2014, December 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Bankers_Life_Fieldhouse&oldid=640160898
- Madison Square Garden. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Madison_Square_Garden&oldid=654828742
- Verizon Center. (2015, March 14). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Verizon_Center&oldid=651299892

- Staples Center. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Staples_Center&oldid=654943633
- Target Center. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Target_Center&oldid=655045980
- Talking Stick Resort Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Talking_Stick_Resort_Arena&oldid=654788366
- AT&T Center. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Center&oldid=654184289
- KeyArena. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=KeyArena&oldid=653237504
- BOK Center. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=BOK_Center&oldid=654259841
- Santander Arena. (2015, February 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Santander_Arena&oldid=648485678
- Odeum Expo Center. (2015, February 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Odeum_Expo_Center&oldid=648219741
- The Gardens Ice House. (2014, December 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=The_Gardens_Ice_House&oldid=636303525
- Tiger Arena. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Tiger_Arena&oldid=653415784
- Dow Event Center. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Dow_Event_Center&oldid=654204684
- York City Ice Arena. (2014, October 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=York_City_Ice_Arena&oldid=629018943
- Talking Stick Resort Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Talking_Stick_Resort_Arena&oldid=654788366
- Honda Center. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Honda_Center&oldid=654822150
- Alamodome. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Alamodome&oldid=654207806
- Moda Center. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Moda_Center&oldid=652902770
- SAP Center at San Jose. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=SAP_Center_at_San_Jose&oldid=654649938
- Spokane Veterans Memorial Arena. (2015, March 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Spokane_Veterans_Memorial_Arena&oldid=654081587

- Jacksonville Veterans Memorial Arena. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Jacksonville_Veterans_Memorial_Arena&oldid=652941155
- CFE Arena. (2015, February 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=CFE_Arena&oldid=646752542
- Amalie Arena. (2015, February 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Amalie_Arena&oldid=646850271
- Quicken Loans Arena. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Quicken_Loans_Arena&oldid=655272955
- Smoothie King Center. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Smoothie_King_Center&oldid=653411690
- Wells Fargo Center (Philadelphia). (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Wells_Fargo_Center_(Philadelphia)&oldid=649738895
- Tim Hortons Field. (2015, March 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Tim_Hortons_Field&oldid=651817295
- Percival Molson Memorial Stadium. (2015, January 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Percival_Molson_Memorial_Stadium&oldid=642808624
- TD Place Stadium. (2015, February 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=TD_Place_Stadium&oldid=648816970
- Rogers Centre. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Rogers_Centre&oldid=654913445
- BC Place. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=BC_Place&oldid=653675532
- McMahon Stadium. (2015, February 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=McMahon_Stadium&ol

Gillette Stadium. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Gillette_Stadium&oldid=654585813did=649207077

- Commonwealth Stadium (Edmonton). (2015, March 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Commonwealth_Stadium_(Edmonton)&oldid=649791012
- Mosaic Stadium at Taylor Field. (2015, February 15). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Mosaic_Stadium_at_Taylor_Field&oldid=647219959

- Investors Group Field. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Investors_Group_Field&oldid=655386188
- Ralph Wilson Stadium. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Ralph_Wilson_Stadium&oldid=655091171
- Sun Life Stadium. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Sun_Life_Stadium&oldid=652918431
- MetLife Stadium. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=MetLife_Stadium&oldid=655501155
- M&T Bank Stadium. (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=M%26T_Bank_Stadium&oldid=638418974
- Paul Brown Stadium. (2014, December 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:46, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Paul_Brown_Stadium&oldid=637052857
- FirstEnergy Stadium (Cleveland). (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:46, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=FirstEnergy_Stadium_(Cleveland)&oldid=654531854
- Heinz Field. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Heinz_Field&oldid=653482405
- NRG Stadium. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=NRG_Stadium&oldid=655407096
- Lucas Oil Stadium. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Lucas_Oil_Stadium&oldid=655376758
- EverBank Field. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=EverBank_Field&oldid=653161581
- LP Field. (2014, November 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=LP_Field&oldid=634894166
- Sports Authority Field at Mile High. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Sports_Authority_Field_at_Mile_High&oldid=653126660
- Arrowhead Stadium. (2015, March 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Arrowhead_Stadium&oldid=654006833
- O.co Coliseum. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=O.co_Coliseum&oldid=653612043
- Qualcomm Stadium. (2015, March 27). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Qualcomm_Stadium&oldid=653741585
- AT&T Stadium. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Stadium&oldid=654731734

- MetLife Stadium. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=MetLife_Stadium&oldid=655501155
- Lincoln Financial Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Lincoln_Financial_Field&oldid=655256338
- FedExField. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=FedExField&oldid=654429456
- Soldier Field. (2015, March 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Soldier_Field&oldid=651010515
- Ford Field. (2015, March 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Ford_Field&oldid=653890809
- Lambeau Field. (2015, March 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Lambeau_Field&oldid=650772069
- TCF Bank Stadium. (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=TCF_Bank_Stadium&oldid=649686714
- Georgia Dome. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Georgia_Dome&oldid=653308324
- Bank of America Stadium. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Bank_of_America_Stadium&oldid=653374205
- Mercedes-Benz Superdome. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Mercedes-Benz_Superdome&oldid=654152346
- Raymond James Stadium. (2015, February 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Raymond_James_Stadium&oldid=649202435
- University of Phoenix Stadium. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=University_of_Phoenix_Stadium&oldid=653559579
- Edward Jones Dome. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Edward_Jones_Dome&oldid=654167126
- Levi's Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Levi%27s_Stadium&oldid=655210318
- CenturyLink Field. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=CenturyLink_Field&oldid=652787982
- Toyota Park. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:30, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Park&oldid=654161546
- Mapfre Stadium. (2015, March 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Mapfre_Stadium&oldid=652025679
- Robert F. Kennedy Memorial Stadium. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Robert_F._Kennedy_Memorial_Stadium&oldid=652754664

- Saputo Stadium. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Saputo_Stadium&oldid=653113798
- Gillette Stadium. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Gillette_Stadium&oldid=654585813
- Yankee Stadium. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Yankee_Stadium&oldid=655513009
- Red Bull Arena (New Jersey). (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Red_Bull_Arena_(New_Jersey)&oldid=653149464
- Orlando Citrus Bowl Stadium. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Orlando_Citrus_Bowl_Stadium&oldid=652829636
- PPL Park. (2015, March 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=PPL_Park&oldid=651972956
- BMO Field. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BMO_Field&oldid=653369346
- Dick's Sporting Goods Park. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Dick%27s_Sporting_Goods_Park&oldid=655566313
- Toyota Stadium (Texas). (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Stadium_(Texas)&oldid=638418651
- BBVA Compass Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BBVA_Compass_Stadium&oldid=655235636
- StubHub Center. (2015, March 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=StubHub_Center&oldid=651093792
- Providence Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Providence_Park&oldid=654711739
- Rio Tinto Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Rio_Tinto_Stadium&oldid=654304261
- Avaya Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Avaya_Stadium&oldid=655230355
- CenturyLink Field. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=CenturyLink_Field&oldid=652787982
- Sporting Park. (2015, March 13). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sporting_Park&oldid=651202621

- BC Place. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BC_Place&oldid=653675532
- Harvard Stadium. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Harvard_Stadium&oldid=654508027
- Village of Lisle-Benedictine University Sports Complex. (2014, September 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Village_of_Lisle-Benedictine_University_Sports_Complex&oldid=624371085
- BBVA Compass Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BBVA_Compass_Stadium&oldid=655235636
- Providence Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Providence_Park&oldid=654711739
- Memorial Stadium (Seattle). (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Memorial_Stadium_(Seattle)&oldid=652979450
- Yurcak Field. (2014, October 11). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Yurcak_Field&oldid=629218625
- Maryland SoccerPlex. (2014, October 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Maryland_SoccerPlex&oldid=629989133
- Sahlen's Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sahlen%27s_Stadium&oldid=654403301
- Atlanta Silverbacks Park. (2014, November 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Atlanta_Silverbacks_Park&oldid=632159843
- WakeMed Soccer Park. (2015, February 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=WakeMed_Soccer_Park&oldid=647633122
- Clarke Stadium. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Clarke_Stadium&oldid=654678732
- Lockhart Stadium. (2014, November 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Lockhart_Stadium&oldid=632454890
- Baseball Grounds of Jacksonville. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Baseball_Grounds_of_Jacksonville&oldid=649550076
- IU Michael A. Carroll Track & Soccer Stadium. (2014, December 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=IU_Michael_A._Carroll_Track_%26_Soccer_Stadium&oldid=636779965

- National Sports Center. (2014, December 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=National_Sports_Center&oldid=636289847
- James M. Shuart Stadium. (2015, January 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=James_M._Shuart_Stadium&oldid=640688511
- Toyota Field. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Field&oldid=654448149
- Al Lang Stadium. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Al_Lang_Stadium&oldid=652776072
- Blackbaud Stadium. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Blackbaud_Stadium&oldid=653227162
- American Legion Memorial Stadium. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=American_Legion_Memorial_Stadium&oldid=654180309
- Skyline Sports Complex. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Skyline_Sports_Complex&oldid=654929230
- Louisville Slugger Field. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Louisville_Slugger_Field&oldid=655428425
- Saputo Stadium. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Saputo_Stadium&oldid=653113798
- Highmark Stadium. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Highmark_Stadium&oldid=654950422
- City Stadium (Richmond). (2015, February 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=City_Stadium_(Richmond)&oldid=647801967
- Sahlen's Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sahlen%27s_Stadium&oldid=654403301
- St. Louis Soccer Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=St._Louis_Soccer_Park&oldid=654795723
- Ontario Soccer Centre. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:35, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Ontario_Soccer_Centre&oldid=649486610
- Legion Stadium. (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:35, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Legion_Stadium&oldid=638361469
- Scottsdale Stadium. (2015, March 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Scottsdale_Stadium&oldid=652123780

- House Park. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=House_Park&oldid=654152211
- Sand Creek Stadium. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sand_Creek_Stadium&oldid=654238924
- StubHub Center. (2015, March 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=StubHub_Center&oldid=651093792
- Bishop McGuinness Catholic High School (Oklahoma). (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Bishop_McGuinness_Catholic_High_School_(Oklahoma)&oldid=655587468
- Anteater Stadium. (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Anteater_Stadium&oldid=638362970
- Merlo Field. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Merlo_Field&oldid=654210299
- Rio Tinto Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Rio_Tinto_Stadium&oldid=654304261
- Bonney Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Bonney_Field&oldid=655241105
- Starfire Sports. (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Starfire_Sports&oldid=653021031
- ONEOK Field. (2015, January 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=ONEOK_Field&oldid=644849140
- Thunderbird Stadium. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:39, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Thunderbird_Stadium&oldid=655408047

Expensive Beauty Products

- Created by Author

Applications Wanted

- All Recorded Traffic Tickets:
  - Data provided by the Baltimore Department of Finance on Open Baltimore (https://data.baltimorecity.gov/) Direct Link: https://data.baltimorecity.gov/Transportation/All-Time/ks7u-tsjz (accessed on

Where's the Joe?

- Mermaid Coffee Company: Data modified to exclude "Starbucks" from everywhere in the data.
  - Open Data https://opendata.socrata.com/Business/All-Starbucks-Locations-in-the-World/xy4y-c4mk provided by Starbucks Corporation

Meta-morphosis

- Created by Author

NYC

- Landmarks:
  - Created By Author
- NYC Subway Entrances:
  - NYC Open Data https://data.cityofnewyork.us/Transportation/Subway-Entrances/drex-xx56 provided by Metropolitan Transportation Authority (MTA)
- NYC_Free_Public_WiFi_12052014:
  - NYC Open Data https://data.cityofnewyork.us/Social-Services/NYC-Wi-Fi-Hotspot-Locations/a9we-mtpn provided by Department of Information Technology & Telecommunications (DoITT)

Content:

The Alteryx Help documentation was used extensively in the research for this manual. Alteryx Help can be accessed through the Help Menu, Properties Window, or with the Hot Key F1 in Alteryx or can be accessed directly at http://help.alteryx.com/.

USEREADY

USEReady is a leading provider of advanced business intelligence, data management, and network security solutions. Our mission is to help businesses succeed by fast-tracking their business performance - through our industry, domain and technology expertise, agile global delivery practices, and our customer fanaticism. For more information, visit: www.useready.com.

"The book has created a great launching point for beginners who want a manual, in addition to the real world use cases, so you can easily learn how best to use Alteryx Designer. By the time you finish working through this manual, someone who has never opened the Alteryx Designer before, will be able to create workflows, design reports, develop applications, and write macros to solve any of their data needs."

- Laura Sellers,
Vice President,
Product Management,
Alteryx

Authored By

Abhilash Ramanathan
Anand Vadul
Ashish Tergaonkar
Jagdeesh Narayanan
Rahul Upadhye
Rima Upadhye
Vijai Narasimha
Vijay Gowtham

Published by

USEREADY

www.useready.com
contact@useready.com

ISBN 978-0-692-44795-6
90000>
9 780692 447956