

Azure-Data-Factory

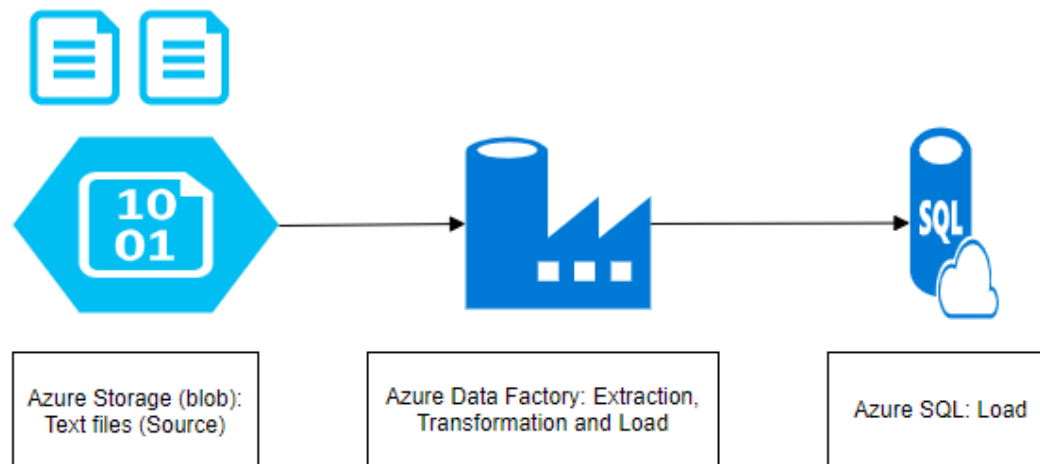
Azure Data Factory

Problem Statement:

XYZ Company has requested to get a file that lists all the products the company sells (source). They also requested the model description which is in a different table (source & transform - we will use lookup & select). XYZ requested shipping weight needs to be calculated by padding the actual weight plus 10% to account for packing (transformation - we will use calculation & derive column). Finally, they want to load the data to Azure SQL by list price descending (sort and sink)

Introduction:

To solve this, we need to use Azure Data Factory (ADF) that will extract text files from azure storage, transform it in ADF and load it to azure SQL Database.





Getting Started:

For this project to work, we need

- a) Azure blob storage
- b) Azure Data Factory
- c) Azure SQL

Source:

In Azure Blob, there are couple of text files which we will process them using ADF.

← → ∨ ↑ Active blobs (default) container								
Name ^	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining
 ProductModel_Lookup.txt	Cool		10/24/2019, 10:04:13 AM	Block Blob	text/plain	2.7 KB	Active	
 Products_All.txt	Cool		10/24/2019, 10:04:13 AM	Block Blob	text/plain	70.3 KB	Active	

I created source for Product_All txt file below.

Set properties

Name

DS_Blob_ProductAll

Linked service *

AZ_Blob_DataFactory

[Edit connection](#)

File path

container

/

Directory

/

Products_All.txt

[Browse](#)

▼

First row as header



Import schema



From connection/store



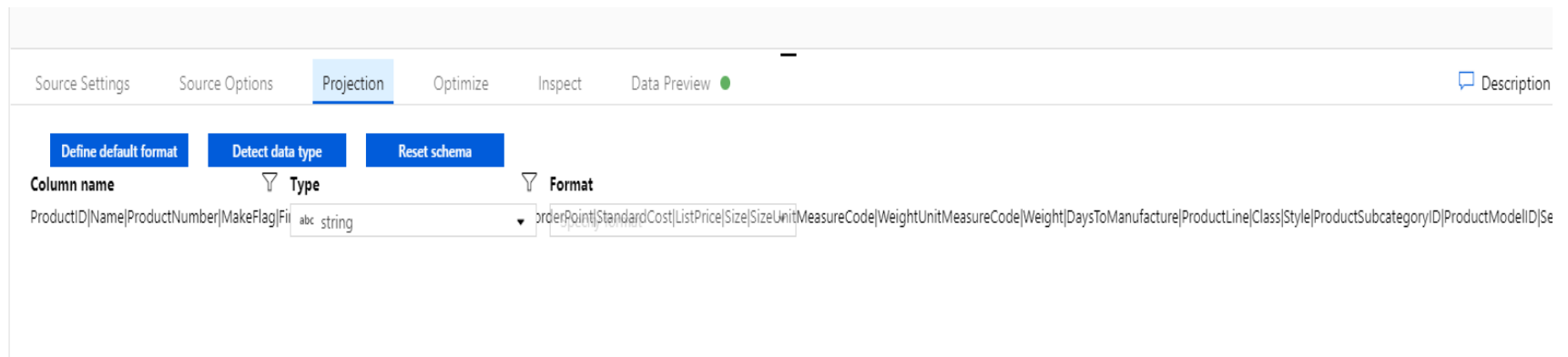
From sample file



None

► Advanced

Now if we look at the data, they are not parsed correctly



I will go to the data source to change the column comma delimiter to pipe delimiter.



DelimitedText
DS_Products_All

General


Connection


Schema

Parameters

Linked service *

 AZ_Blob_DataFactory ▼

 Test connection

 Edit

+ New

File path *

container / Directory / Products_All.txt

Browse



 Preview data

Compression type

none ▼

Column delimiter

Pipe (|) ▼

☐ Edit

Row delimiter

Auto detect (\r,\n, or \r\n) ▼

☐ Edit

Encoding

Default(UTF-8) ▼

Escape character

Backslash (\) ▼

☐ Edit

Quote character

Double quote (") ▼

☐ Edit

First row as header



We have to select import schema to refresh source data schema.



DelimitedText

ds_Blob_ProductsAll

General

Connection

Schema

Parameters

Import schema

Clear

Column name

ProductID

Name

ProductNumber

MakeFlag

FinishedGoodsFlag

Color

SafetyStockLevel

ReorderPoint

StandardCost

ListPrice

Size

SizeUnitMeasureCode

WeightUnitMeasureCode

Weight

DaysToManufacture

Type

String

String

String

String

String

String

String

String

String

String

String

String

String

String

String

Now, if we preview the data, it is parsed correctly.

Data Preview




Linked service: AZ_Blob_DataFactory

Object: Products_All.txt

ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint	Sta
1	Adjustable Race	AR-5381	False	False	\N	1000	750	0.0
2	Bearing Ball	BA-8327	False	False	\N	1000	750	0.0
3	BB Ball Bearing	BE-2349	True	False	\N	800	600	0.0
4	Headset Ball Bearings	BE-2908	False	False	\N	800	600	0.0
316	Blade	BL-2036	True	False	\N	800	600	0.0
317	LL Crankarm	CA-5965	False	False	Black	500	375	0.0
318	ML Crankarm	CA-6738	False	False	Black	500	375	0.0
319	HL Crankarm	CA-7457	False	False	Black	500	375	0.0

We will modify the data types for ProductsAll.text file

Weight	1.2 double	▼	Specify format	▼
DaysToManufacture	123 integer	▼	Specify format	▼
ProductLine	abc string	▼	Specify format	▼
Class	abc string	▼	Specify format	▼
Style	abc string	▼	Specify format	▼
ProductSubcategoryID	12s short	▼	Specify format	▼
ProductModelID	12s short	▼	Specify format	▼
SellStartDate	 timestamp	▼	Specify format	▼
SellEndDate	 timestamp	▼	Specify format	▼
DiscontinuedDate	 timestamp	▼	Specify format	▼

Now we will create second data source for Product Model table.

Set properties

Name

ds_Blob_ProductModel

Linked service *

AZ_Blob_DataFactory

[Edit connection](#)

File path

container

/

Directory

/

ProductModel_Lookup:

[Browse](#)

▼

First row as header



Import schema



From connection/store



From sample file



None

► Advanced

Columns are separated by commas, so they are parsed correctly for second data source.

Data Preview



Linked service: AZ_Blob_DataFactory

Object: ProductModel_Lookup.txt

ProductModelID	Name
122	All-Purpose Bike Stand
119	Bike Wash
115	Cable Lock
98	Chain
1	Classic Vest
2	Cycling Cap
121	Fender Set - Mountain
102	Front Brakes
103	Front Derailleur

Transformation:

I will change the ProductModelID to short data type, so we can compare them with the first source file ProductsAll.text file.

Define default format

Detect data type

Reset schema

Column name	Type	Format
ProductModelID	12s short	Specify format
Name	abc string	Specify format

Now I will do a lookup comparison based on our two sources with comparing ProductModelIds.

The screenshot displays a data integration workflow in a tool. At the top, there are three data streams: 'ProductsAll' (Import data from ds_Blob_ProductsAll), 'ProductModel' (Import data from ds_Blob_ProductModel), and 'ModelNameLookup' (Columns: 25 total). The 'ProductsAll' and 'ProductModel' streams are connected to the 'ModelNameLookup' stream via a blue line, indicating a lookup operation. Below the streams is a dashed box labeled 'Add Source'. At the bottom, there is a tabbed interface with four tabs: 'Lookup Settings' (selected), 'Optimize', 'Inspect', and 'Data Preview' (with a green status indicator). The 'Lookup Settings' tab contains the following configuration:

- Output stream name *: ModelNameLookup [Documentation](#)
- Primary stream *: ProductsAll
- Lookup stream *: ProductModel
- Lookup conditions *: **Left: ProductsAll's column** 12s ProductModelID == **Right: ProductModel's column** 12s ProductModelID

As, one of our requirements is to add actual weight plus 10% of weight so we can add packaging weight for the whole package.

The screenshot displays a data flow interface with three stages in a pipeline:

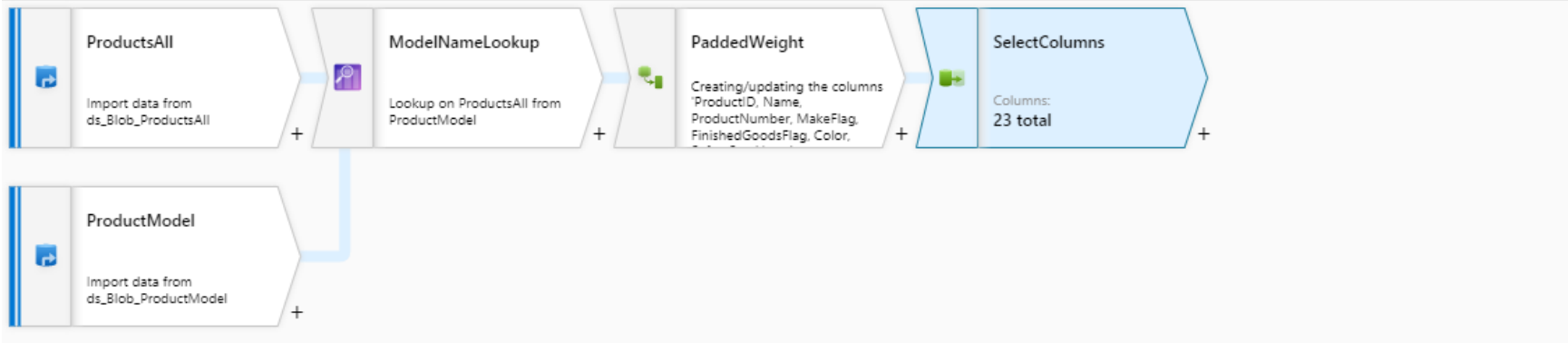
- ProductsAll**: Import data from ds_Blob_ProductsAll
- ModelNameLookup**: Lookup on ProductsAll from ProductModel
- PaddedWeight**: Columns: 26 total

The **PaddedWeight** stage is selected, and its settings are shown in the bottom panel:

- Output stream name**: PaddedWeight
- Incoming stream**: ModelNameLookup
- Columns**: PaddedWeight (selected), Weight*1.10 (1.2)

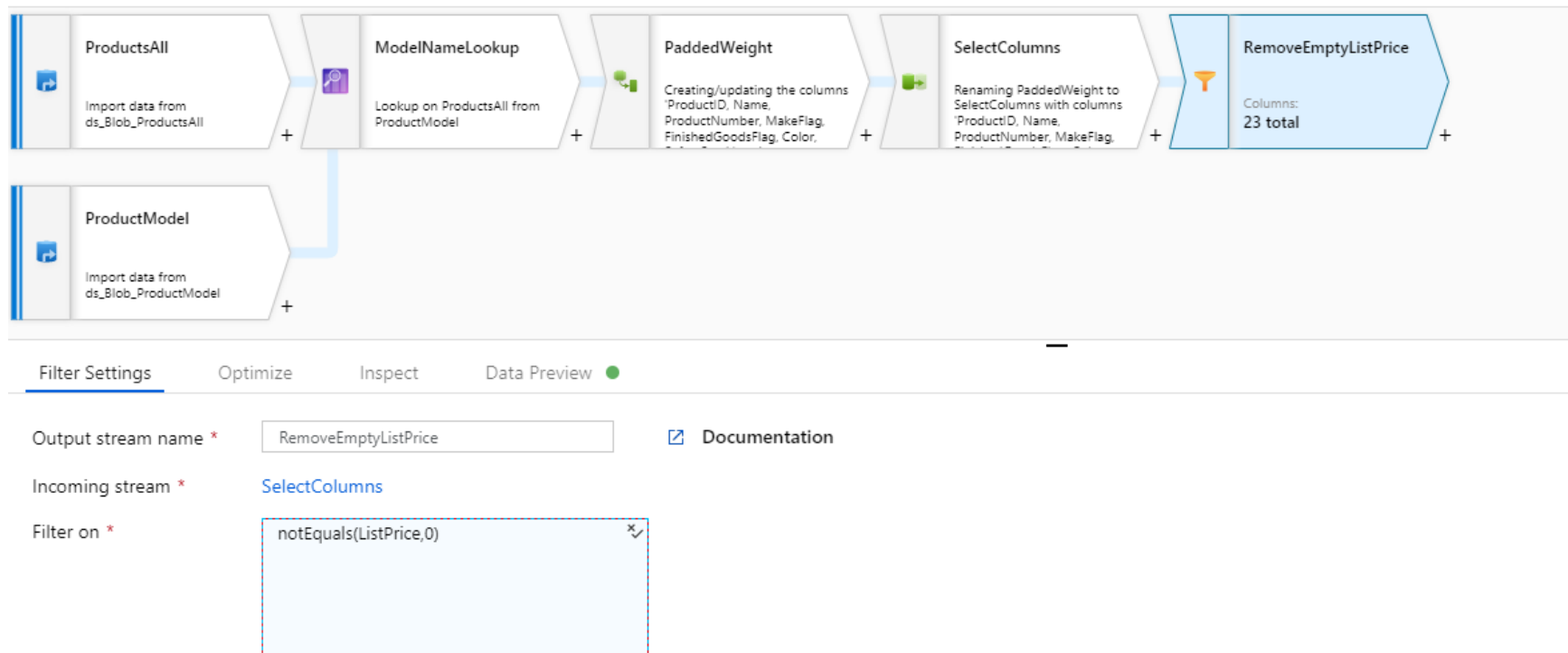
The bottom panel also includes tabs for **Derived column's settings**, **Optimize**, **Inspect**, and **Data Preview** (selected). A **Documentation** link is also present.

As we have 26 columns in our data flows, we will remove few columns which are needed by the business.



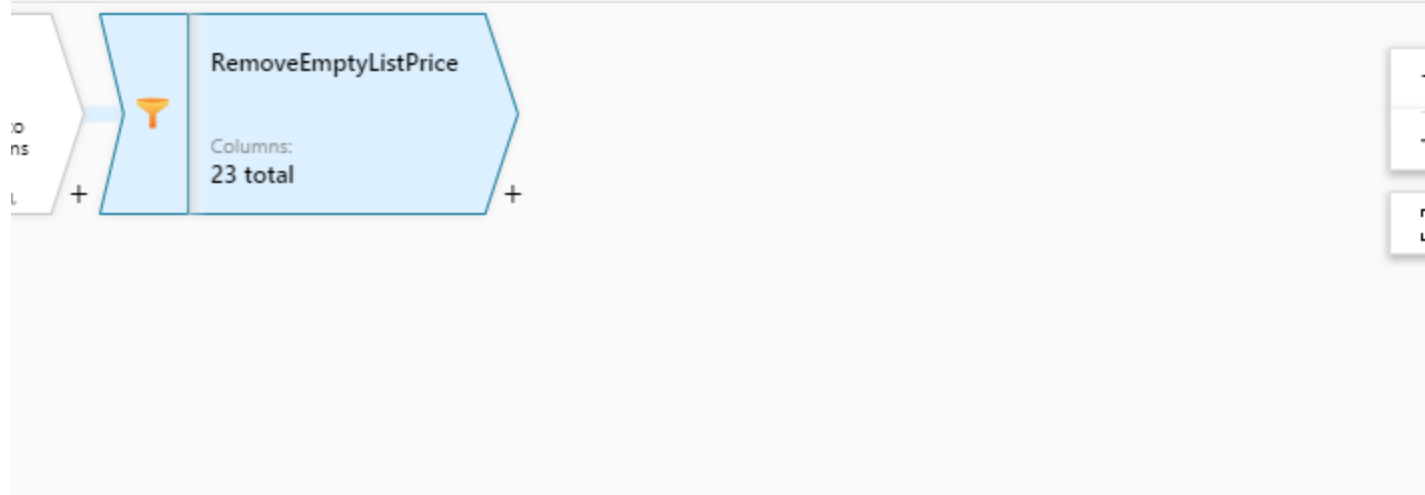
Select Settings	Optimize	Inspect	Data Preview
<input type="checkbox"/>	abc StandardCost	StandardCost	+
<input type="checkbox"/>	abc ListPrice	ListPrice	+
<input type="checkbox"/>	abc Size	Size	+
<input type="checkbox"/>	abc SizeUnitMeasureCode	SizeUnitMeasureCode	+
<input type="checkbox"/>	abc WeightUnitMeasureCode	WeightUnitMeasureCode	+
<input type="checkbox"/>	123 DaysToManufacture	DaysToManufacture	+
<input type="checkbox"/>	abc ProductLine	ProductLine	+
<input type="checkbox"/>	abc Class	Class	+
<input type="checkbox"/>	abc Style	Style	+
<input type="checkbox"/>	125 ProductSubcategoryID	ProductSubcategoryID	+
<input type="checkbox"/>	125 ProductsAll@ProductModelID	ProductModelID	+
<input type="checkbox"/>	SellStartDate	SellStartDate	+
<input type="checkbox"/>	SellEndDate	SellEndDate	+
<input type="checkbox"/>	DiscontinuedDate	DiscontinuedDate	+
<input type="checkbox"/>	abc ProductModel@Name	ModelName	+
<input type="checkbox"/>	1.2 PaddedWeight	PaddedWeight	+

We will also remove empty list price using filter.





After using the filter, our number of rows are reduced to 304 rows.

 Hide graph  Script  Columns



 Descript

 UPSERT 0

 LOOKUP 91

TOTAL 304



Now our last requirement is to sort list price in descending order.

The screenshot displays a data pipeline configuration interface. The pipeline consists of the following steps:

- ProductsAll**: Import data from ds_Blob_ProductsAll.
- ModelNameLookup**: Lookup on ProductsAll from ProductModel.
- Join**: A join step with Reference: 1 and Columns: 26 total.
- SelectColumns**: Renaming PaddedWeight to SelectColumns with columns ProductID, Name, ProductNumber, MakeFlag.
- RemoveEmptyListPrice**: Filtering rows using expressions on columns 'ListPrice'.
- ListPriceSorting**: Columns: 23 total.

Below the pipeline, the **Sort Settings** tab is active. The configuration for the **ListPriceSorting** step is shown:

- Output stream name ***: ListPriceSorting
- Incoming stream ***: RemoveEmptyListPrice
- Options ***:
 - ☐ Case insensitive
 - ☐ Sort only within partition
- Sort conditions ***:

RemoveEmptyListPrice's column	Order	Nulls first
1.2 ListPrice	Descending	<input type="checkbox"/>

Destination:

Now I will create a destination table and load the data in Azure SQL.



Sink Settings Mapping Optimize Inspect Data Preview

Output stream name * [Documentation](#)

Incoming stream * [ListPriceSorting](#)

Sink dataset *

Load_SQL_AZ_PRDBI

Edit + New

Options

- ☒ Allow schema drift ⓘ
- ☐ Validate schema ⓘ

Now we will create a new pipeline to run this data flow.

Adding Data Flow

☒ Use existing Data Flow ☐ Create new Data Flow

Existing Data Flow *

df_Product_Calculation ▼

The screenshot displays a software interface for managing data flows. At the top, a 'Data Flow' panel is visible, containing a component named 'df_Product_Calculation' with a small green status indicator. Below this, a toolbar includes icons for adding (+), removing (-), locking (lock), zooming in ([100%]), zooming out (magnifying glass), panning (hand), and other navigation tools. A tabbed interface at the bottom shows the 'General' tab selected, displaying the component's name 'df_Product_Calculation' and a 'Documentation' link.

Data Flow

df_Product_Calculation

+ - [100%] [magnifying glass] [hand] [lock] [refresh]

General Settings Parameters User properties

Name * df_Product_Calculation [Documentation](#)

From above graph, we can see successful execution of pipeline is completed.

Data Flow

df_Product_Calculation

<div><div></div><div></div><div></div><div>100%</div><div></div><div></div><div></div><div></div></div>						
<div>GeneralParametersVariablesOutput</div>						
<div>Pipeline run ID: 4e349d5c-b66c-4627-8e3f-4ce6696a0c46 [@] <div></div> ⓘ</div>						
NAME	TYPE	RUN START	DURATION	STATUS	ACTIONS	RUN ID
df_Product_Calculation	ExecuteDataFlow	10/24/2019 4:08 PM	00:01:01	<div></div> Succeeded	<div></div> <div></div> <div></div>	e01ed140-4ec5-4cb8-97fe-107bcb23269d

Data is uploaded to the table below.

```

SELECT TOP (1000) [ProductID]
, [Name]
, [ProductNumber]
, [MakeFlag]
, [FinishedGoodsFlag]
, [Color]
, [SafetyStockLevel]
, [ReorderPoint]
, [StandardCost]
, [ListPrice]
, [Size]
, [SizeUnitMeasureCode]
, [WeightUnitMeasureCode]
, [DaysToManufacture]
, [ProductLine]
, [Class]
, [Style]
, [ProductSubcategoryID]
, [ProductModelID]
, [SellStartDate]
, [SellEndDate]
, [DiscontinuedDate]
, [ModelName]
, [PaddedWeight]
FROM [Products].[ModelProductsLoad]

```

100 %

Results Messages

	ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint	StandardCost	ListPrice	Size	SizeUnitMeasureCode	WeightUnitMeasureCode	DaysToManufacture	ProductL
1	771	Mountain-100 Silver, 38	BK-M82S-38	True	True	Silver	100	75	1912.1544	3399.99	38	CM	LB	4	M
2	772	Mountain-100 Silver, 42	BK-M82S-42	True	True	Silver	100	75	1912.1544	3399.99	42	CM	LB	4	M
3	773	Mountain-100 Silver, 44	BK-M82S-44	True	True	Silver	100	75	1912.1544	3399.99	44	CM	LB	4	M
4	774	Mountain-100 Silver, 48	BK-M82S-48	True	True	Silver	100	75	1912.1544	3399.99	48	CM	LB	4	M
5	749	Road-150 Red, 62	BK-R93R-62	True	True	Red	100	75	2171.2942	3578.27	62	CM	LB	4	R
6	750	Road-150 Red, 44	BK-R93R-44	True	True	Red	100	75	2171.2942	3578.27	44	CM	LB	4	R
7	751	Road-150 Red, 48	BK-R93R-48	True	True	Red	100	75	2171.2942	3578.27	48	CM	LB	4	R
8	752	Road-150 Red, 52	BK-R93R-52	True	True	Red	100	75	2171.2942	3578.27	52	CM	LB	4	R
9	753	Road-150 Red, 56	BK-R93R-56	True	True	Red	100	75	2171.2942	3578.27	56	CM	LB	4	R
10	775	Mountain-100 Black, 38	BK-M82B-38	True	True	Black	100	75	1898.0944	3374.99	38	CM	LB	4	M
11	776	Mountain-100 Black, 42	BK-M82B-42	True	True	Black	100	75	1898.0944	3374.99	42	CM	LB	4	M
12	777	Mountain-100 Black, 44	BK-M82B-44	True	True	Black	100	75	1898.0944	3374.99	44	CM	LB	4	M
13	778	Mountain-100 Black, 48	BK-M82B-48	True	True	Black	100	75	1898.0944	3374.99	48	CM	LB	4	M
14	954	Touring-1000 Yellow, 46	BK-T79Y-46	True	True	Yellow	100	75	1481.9379	2384.07	46	CM	LB	4	T
15	955	Touring-1000 Yellow, 50	BK-T79Y-50	True	True	Yellow	100	75	1481.9379	2384.07	50	CM	LB	4	T