

CLD Exercise 15: CSV File Text String Parsing

Objective

Develop a data parsing application using LabVIEW and the given application front panel (Figure 1). The utility reads a Comma Separate Value (.csv) file, parses, processes, formats, and saves the reformatted data in a new CSV file.

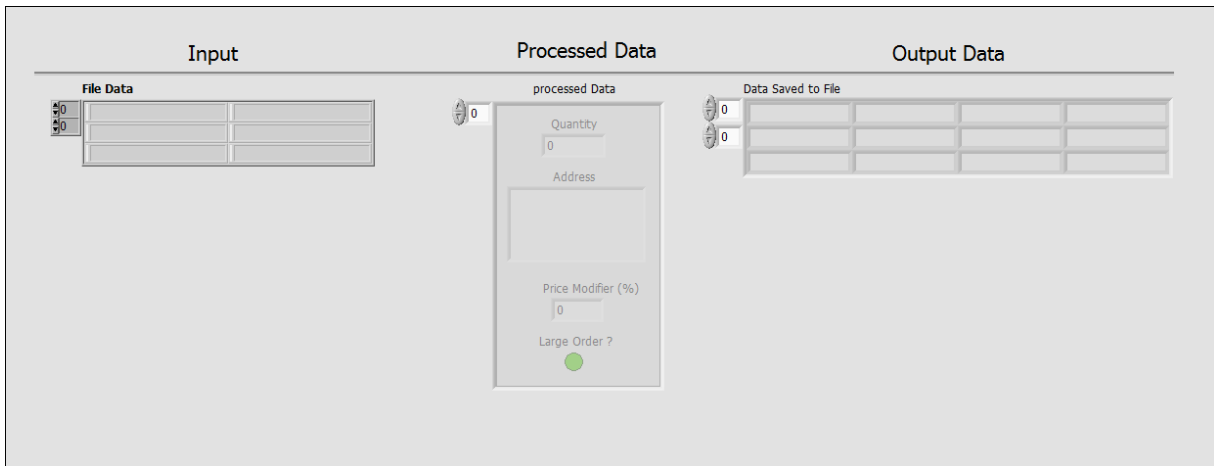


Figure 1. Application Front Panel

General Operation

The parser application reads the data file `CLD 15 CSV File.csv` for customer order information. This file contains order data for the product “tubes”. The data is parsed, reformatted, and then saved in the results file `CLD 15 Results file.csv`.

Data Formats and Requirements

Data File

The data file is comprised of rows of two strings separated by a comma. The first column is the Quantity, and the second column is the Customer Information. The file contains the following three rows:

- 091,TX78701Travis+02
- 256,CA94501Alameda-03
- 185,TX78710Williamson+10

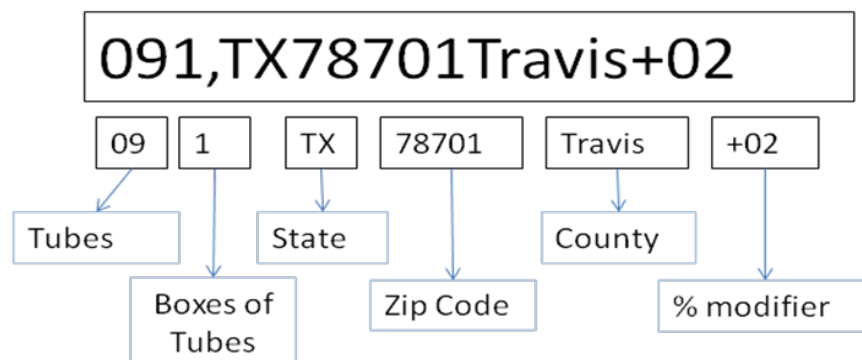
Data File Format

The data file format is described below.

Data Example	Data Name	Data Format
091	Quantity	First two digits represent the number of tubes. Last digit represents the number of boxes. There are 50 tubes in each box.
TX78701Travis+02	Customer	The first <u>two letters</u> represent the state, the <u>middle numbers</u> represent the zip code, the following text represents the county, and the last digits are the percent modifiers (either positive (+) or negative (-) percent)

Table 1. Data Code Format Table

Format Example: Here is a graphical breakdown of the record: “091,TX78701Travis+02”



- **Column One:** 091, there are 09 tubes and 1 box of 50 tubes. (Boxes hold 50 tubes)
- **Column Two:** TX is the state, 78701 is the Zip, Travis is the county, and there is a positive 2% modifier on the price.

Data Conversion

The data in the strings must be converted to provide the total number of tubes, the address, the price modifier, and the comparison of the total price to the “large order” threshold value.

The following is the list of conversion information.

- 1 box is 50 tubes.
- Each tube is \$10.
- The quantity should be in number of tubes.
- The price modifier is a decimal multiplier against the price.
 - For example: “-3” percent becomes the price modifier of .97
- An order is considered large if the total cost is over \$1000.
- Address format:
 - County, state, zip code (including commas)

For example: Travis, TX, 78731

Results File Format

Save the data in a file named `CLD 15 Results file.csv`.

The converted data should be saved in a CSV format of rows of six values. The values are in the order of the cluster.

Data Order: Number of Tubes, County, State, Zip Code, price modifier, Large Order (TRUE / FALSE)

For example, based on the data file example from above, the following results should result.

- 59 ,Travis ,TX ,78701 ,1.02 ,FALSE
- 325 ,Alameda ,CA ,94501 ,0.97 ,TRUE
- 268 ,Williamson ,TX ,78710 ,1.1 ,TRUE

Initialization

The front panel has no specific initialization requirements.

Operation

VI Run

When the application is run it will:

- Read the file data and display the file contents in file data
- Process the data and display in Processed Data
- Format the processed data and display in the Data Saved to File

Challenge Exercise

Develop the same application as a Functional Global Variable with the states, Initialize, Read File, Process data, and Write file. Use the given application front panel (Figure 2).

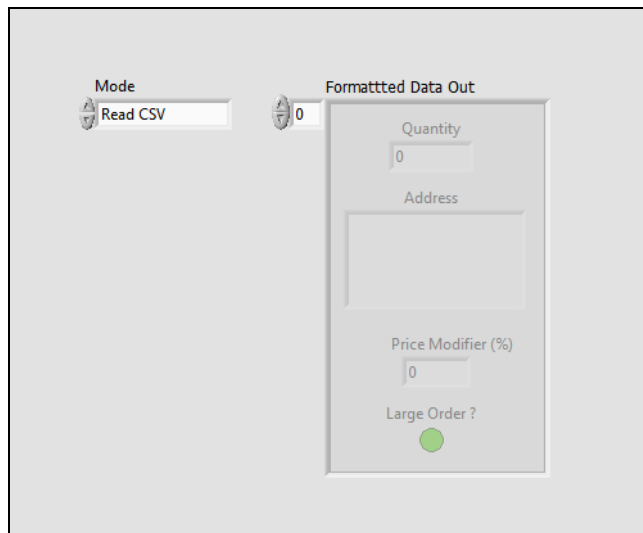


Figure 2. Application Front Panel

General Operation

The application shows the **Formatted Data Out** for the Process and Write steps.