EXAMPLE - Settype Transform

This example illustrates how to clean up data that has been interpreted as numeric in nature, when it is actually String or a structured string type, such as Gender. This example uses:

- `settype` - defines the data type for a column or columns. See *Settype Transform*.
- `merge` - merges two String type columns together. See *Merge Transform*.

Source:

The following example contains customer ID and Zip code information in two columns. When this data is loaded into the Transformer page, it is initially interpreted as numeric, since it contains all numerals.

The four-digit `ZipCode` values should have five digits, with a `0` in front.

CustId	ZipCode
4020123	1234
2012121	94105
3212012	94101
1301212	2020

Transform:

CustId column: This column needs to be retyped as String values. You can set the column data type to String through the column drop-down, which is rendered as the following transform:

```
settype col:CustId type:'String'
```

While the column is now of String type, future transforms might cause it to be re-inferred as Integer values. To protect against this possibility, you might want to add a marker at the front of the string. This marker should be removed prior to execution.

The basic method is to create a new column containing the customer ID marker (`C`) and then merge this column and the existing `CustId` column together. It's useful to add such an indicator to the front in case the customer identifier is a numeric value that could be confused with other numeric values. Also, this merge step forces the value to be interpreted as a String value, which is more appropriate for an identifier.

```
merge col:'C', CustId
```

You can now delete the `CustId` columns and rename the new column as `CustId`.

ZipCode column: This column needs to be converted to valid Zip Code values. For ease of use, this column should be of type String:

```
settype col:ZipCode type:'Zipcode'
```

The transform below changes the value in the `ZipCode` column if the length of the value is four in any row. The new value is the original value prepended with the numeral `0`:

```
set col: ZipCode value: if(len($col) == 4, merge(['0',$col]), $col)
```

This column might now be re-typed as Zipcode type.

Results:

CustId	ZipCode
C4020123	01234
C2012121	94105
C3212012	94101
C1301212	02020

Remember to remove the `C` marker from the `CustId` column. Select the `C` value in the `CustId` column and choose the `replace` transform. You might need to re-type the cleaned data as String data.