Split Column

For many recipes, the first step is to split data from a single column into multiple columns. This section describes the various methods that can be used for splitting a single column into one or more columns, based on character- or pattern-matching or position within the column's values.

Split by Delimiter

When data is initially imported into Cloud Dataprep by TRIFACTA® INC., data in each row may be split on a single delimiter. In the following example, you can see that the tab key is a single clear delimiter:

```
^MSIDN^IMEI>       DATETTIME/TIMEZONE OFFSET/DURATION      MSWCNT:BASCNT^BASTRA    C
70097665881^13011330554^011808005351311>   2014-12-12T00:06:13/-5/1.55     MSC001:BS
70097665881^13011330554^011808005351311>   2014-12-12T02:27:26/-5/0.00     MSC001:BS
170-097665881^13011330554^011808005351311> 2014-12-12T03:24:20/-5/0        MSC001:BS
```

However, when this data is imported, it may be rendered in the data grid in the following structure:

column2	column3	colu
<IMSI^MSIDN^IMEI>	DATETTIME/TIMEZONE OFFSET/DURATION	MSW
<310170097665881^13011330554^011808005351311>	2014-12-12T00:06:13/-5/1.55	MSC
<310170097665881^13011330554^011808005351311>	2014-12-12T02:27:26/-5/0.00	MSC
<310-170-097665881^13011330554^011808005351311>	2014-12-12T03:24:20/-5/0	MSC

Notes:

- When the data is first imported, all of it is contained in a single column named column1. The application automatically splits the columns on the tab character for you and removes the original column1.

Tip: This auto-split does not appear in your recipe by default. For most formats, a set of initial steps is automatically applied to the dataset. Optionally, you can review and modify these steps, but you must deselect Detect Structure during the import. See Initial Parsing Steps (/dataprep/docs/html/Initial-Parsing-Steps_57344625).

- Because the application was unable to determine clear headers for each column's data, generic ones are used. So, before you apply a header to your data, you must split out the data within each column.
- The delimiters within each column vary.
  - column2 uses the caret, while column3 uses the forward slash.
  - column4 and column5 use multiple delimiters.
- There is sparseness in the data. Note that in column5, the second row contains the value `11` at the end, while the other two data rows do not have this value.

Split on single delimiter

For column2, you can split the column into separate columns based on the caret delimiter:

Transformation Name	`Split by delimiter`
Parameter: Column	`column2`
Parameter: Option	`By delimiter`
Parameter: Delimiter	`'^'`
Parameter: Number of columns to create	2

NOTE: The Number of columns to create value reflects the total number of new columns to generate.

Results:

Below is how the data in column2 is transformed:

column1	column6	column7

<IMSI	MSIDN	IMEI>
<310170097665881	13011330554	011808005351311>
<310170097665881	13011330554	011808005351311>
<310-170-097665881	13011330554	011808005351311>

- Since column1 was unused as a name, it re-appears here. column6 and column7 are the next available generic column names.
- There is a small bit of cleanup to do in column1 and column7 to remove the symbols at the beginning and end of these column values. You can do this cleanup before the split in the original column2 if desired.

For column3, suppose that you want to keep the DATETIME and TIMEZONE OFFSET values in the same column, preserving the forward slash to demarcate these two values. The DURATION values are to be split into a separate column:

Transformation Name	`Split by delimiter`
Parameter: Column	`column2`
Parameter: Option	`By delimiter`
Parameter: Delimiter	`'/'`
Parameter: Start to split after	`` `/(-{digit}|{digit})` ``

- The above uses Cloud Dataprep patterns, which are simplified versions of regular expressions for matching patterns.
  - In this case, the expression is the following:

    ```
    `/(-{digit}|{digit})`
    ```

  - For the Start to split after value, the above indicates that the application should start to look for matches on the delimiter (forward slash) only after the above pattern has

been detected in the column values.

- In this case, the pattern describes values that appear after a forward slash and could be a negative digit or a positive digit, which matches the pattern for the TIMEZONE OFFSET values in the column.
- For more information on how to use Cloud Dataprep patterns, see Text Matching (/dataprep/docs/html/Text-Matching_57344767).

- Since you are splitting the column into two columns, you do not need to specify the number of new columns to create. The default is `1`.

Split column by multiple delimiters

After splitting column3, the data resembled the following:

column3
DATETTIME/TIMEZONE OFFSET
2014-12-12T00:06:13/-5
2014-12-12T02:27:26/-5
2014-12-12T03:24:20/-5

Suppose you want to break down the components of this date-time data into separate columns for year, month, day, hour, minute, second, and offset. The following could be use to do so:

Transformation Name	`Split by delimiter`
Parameter: Column	`column2`
Parameter: Option	`By multiple delimiters`
Parameter: Delimiter 1	`'-'`
Parameter: Delimiter 2	`'-'`
Parameter: Delimiter 3	`'T'`

Parameter: Delimiter 4	':'
Parameter: Delimiter 5	':'
Parameter: Delimiter 6	'/'

- Each delimiter is entered on a separate row.
- Delimiters are processed in the listed order.

Split column between delimiters

Suppose that for column4, you want to split the column such that the middle part section is removed. You could use the previous transformation and then delete the middle column. You can also use the following transformation, which identifies that starting and editing delimiters that demarcate the separator between fields, effectively removing the middle column:

Transformation Name	`Split by delimiter`
Parameter: Column	`column4`
Parameter: Option	`By two delimiters`
Parameter: Start delimiter	':'
Parameter: Include as part of split	`Selected`
Parameter: End delimiter	'^'
Parameter: Include as part of split	`Selected`

- The separator between the columns is all of the content between the forward slashes. This content is removed from the dataset.
- The two selected options include the forward slashes as part of the separator, which removes them from the dataset.

Split by Position

You can also perform column splits based on numerical positions in column values. These splitting options are useful for highly regular data that is of consistent length.

Suppose you have the following coordination information in three dimensions (x, y, and z). Note that the data is very regular, with leading zeroes for values that are less than 1000.

column1
POSXPOSYPOSZ
000100040001
012405210555
100220046554
202056789011
379274329832

Split column by positions

The above data could be split based on positions within a column's value:

Transformation Name	`Split by character position`
Parameter: Column	`column1`
Parameter: Option	`By positions`
Parameter: Position 1	`4`
Parameter: Position 2	`8`

Results:

column2	column3	column4

POSX	POSY	POSZ
0001	0004	0001
0124	0521	0555
1002	2004	6554
2020	5678	9011
3792	7432	9832

Split columns between positions

Suppose that you wish to split the above source data such that the middle column is removed:

Transformation Name	`Split by character position`
Parameter: Column	`column1`
Parameter: Option	`Between two positions`
Parameter: Position 1	`4`
Parameter: Position 2	`8`

Results:

column2	column3
POSX	POSZ
0001	0001
0124	0555
1002	6554
2020	9011
3792	9832

Split column at regular interval

The above transformation could be simplified even further, since the splits happen at regular intervals:

Transformation Name	`Split by character position`
Parameter: Column	`column1`
Parameter: Option	`At regular interval`
Parameter: Interval	`4`
Parameter: Number of times to split	`2`

Results:

The results would be the same as the first example.

Encoding Issues

If you are attempting to split columns based on non-ASCII characters that appear in the dataset, your transformations may fail.

In these cases, you should change the encoding that is applied to the dataset.

Steps:

1. In the Import Data page, select the dataset to import.
2. When the dataset card appears in the right column, click the Edit Settings link.
3. From the drop-down, select a more appropriate encoding to apply to the file.
4. Import the data and wrangle.
5. Try your split transformation on the dataset.

Splitting Rows

When a dataset is imported, the application attempts to split the data into individual rows, based on any available end of line delimiters. This transformation is performed automatically and is not included in your initial set of steps.

If the data is not consistently formatted, the rows may not be properly split. If so, you can disable the automatic splitting of rows.

Steps:

1. In the Import Data page, select the dataset to import.
2. When the dataset card appears in the right column, click the Edit Settings link.
3. Deselect the Detect Structure checkbox.
4. Import the data and wrangle.

The steps used to detect structure are listed as the first steps of your recipe, which allows you to modify them as needed. For more information, see Initial Parsing Steps (/dataprep/docs/html/Initial-Parsing-Steps_57344625).

See Import Data Page (/dataprep/docs/html/Import-Data-Page_57344837).

Last updated 2020-07-13 UTC.