

# EXAMPLE - Splitting with Different Delimiter Types

This example shows how you can split data from a single column into multiple columns using delimiters.

- **single-pattern delimiter:** One pattern is applied one or more times to the source column to define the delimiters for the output columns
- **multi-pattern delimiter:** Multiple patterns, in the form of explicit strings, character index positions, or fixed-width fields, are used to split the column.

## Source:

In this example, your CSV dataset contains status messages from a set of servers. In this case, the data about the server and the timestamp is contained in a single value within the CSV.

```
Server|Date Time,Status
admin.examplecom|2016-03-05 07:04:00,down
webapp.examplecom|2016-03-05 07:04:00,ok
admin.examplecom|2016-03-05 07:04:30,rebooting
webapp.examplecom|2016-03-05 07:04:00,ok
admin.examplecom|2016-03-05 07:05:00,ok
webapp.examplecom|2016-03-05 07:05:00,ok
```

## Transformation:

When the data is first loaded into the Transformer page, the CSV data is split using the following two transformations:

|                            |                 |
|----------------------------|-----------------|
| <b>Transformation Name</b> | Split into rows |
| <b>Parameter: Column</b>   | column1         |
| <b>Parameter: Split on</b> | \n              |

|  |              |
|--|--------------|
| <b>Transformation Name</b>               | Split column |
| <b>Parameter: Column</b>                 | column1      |
| <b>Parameter: Option</b>                 | On pattern   |
| <b>Parameter: Match pattern</b>          | ' , '        |
| <b>Parameter: Ignore matches between</b> | \"           |

You might need to add a header as the first step:

|                              |                                  |
|------------------------------|----------------------------------|
| <b>Transformation Name</b>   | Rename column with row(s)        |
| <b>Parameter: Option</b>     | Use row(s) as column names       |
| <b>Parameter: Type</b>       | Use a single row to name columns |
| <b>Parameter: Row number</b> | 1                                |

At this point, your data should look like the following:

| Server_Date_Time                      | Status |
|---------------------------------------|--------|
| admin.example.com 2016-03-05 07:04:00 | down   |

|  |           |
|--|-----------|
| webapp.example.com 2016-03-05 07:04:00 | ok        |
| admin.example.com 2016-03-05 07:04:30  | rebooting |
| webapp.example.com 2016-03-05 07:04:30 | ok        |
| admin.example.com 2016-03-05 07:05:00  | ok        |
| webapp.example.com 2016-03-05 07:05:00 | ok        |

The first column contains three distinct sets of data: the server name, the date, and the time. Note that the delimiters between these fields are different, so you should use a multi-pattern delimiter to break them apart:

|                            |                      |
|----------------------------|----------------------|
| <b>Transformation Name</b> | Split column         |
| <b>Parameter: Column</b>   | Server Date Time     |
| <b>Parameter: Option</b>   | Sequence of patterns |
| <b>Parameter: Pattern1</b> | ' , '                |
| <b>Parameter: Pattern2</b> | ' - '                |

When the above is added, you should see three separate columns with the individual fields of information. Note that the source column has been automatically dropped.

Now, you decide that it would be useful to break apart the date information column into separate columns for year, month, and day. Since the column delimiter of this field is consistently a dash (-), you can use a single-pattern delimiter with the following transformation:

|   |                    |
|---|--------------------|
| <b>Transformation Name</b>                    | Split by delimiter |
| <b>Parameter: Column</b>                      | Server Date Time2  |
| <b>Parameter: Option</b>                      | By delimiter       |
| <b>Parameter: Delimiter</b>                   | ' - '              |
| <b>Parameter: Number of columns to create</b> | 2                  |

## Results:

After you rename the generated columns, your dataset should look like the following. Note that the source timestamp column has been automatically dropped.

| server             | year | month | day | time     | Status    |
|--------------------|------|-------|-----|----------|-----------|
| admin.example.com  | 2016 | 03    | 05  | 07:04:00 | down      |
| webapp.example.com | 2016 | 03    | 05  | 07:04:00 | ok        |
| admin.example.com  | 2016 | 03    | 05  | 07:04:30 | rebooting |
| webapp.example.com | 2016 | 03    | 05  | 07:04:30 | ok        |
| admin.example.com  | 2016 | 03    | 05  | 07:05:00 | ok        |
| webapp.example.com | 2016 | 03    | 05  | 07:05:00 | ok        |