

# Manage String Lengths

## Contents:

- *Test String Length*
- *Truncate Strings*
- *Specialized String Lengths*
- *Use Rightmost Values*
- *Substring Values*
- *Additional String Functions*

In this example, your target system has a limit on the maximum length for the First Name and Last Name fields. You can use the following transforms to evaluate and truncate your strings based on their length.

## Test String Length

You can use the following command to write a `TOO LONG` message when the length of the `first_name` field exceeds 32 characters:

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	String_test
<b>Parameter: Formula</b>	IF(LEN(first_name) > 32, 'TOO LONG',String_test)

## Truncate Strings

The above test allows you to evaluate individual strings that are too long to see if they are errors or can somehow be shortened. For a large dataset in which you cannot easily solve these problems, you can simply choose to cut off the length of a string at 32 characters:

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	*
<b>Parameter: Formula</b>	LEFT(\$col,32)

In the above, you can use a wildcard to match all columns in the dataset. The replacement value is defined to be the first 32 characters of the source column (`$col`). By definition of the `LEFT` function, columns that are shorter than 32 characters in length are untouched.

**Tip:** If the field you are truncating is used as a key to your dataset, you should verify that your key still contains unique values after you have applied the truncation. For example, if the combination of `first_name` and `last_name` is a unique identifier in your dataset, you should verify that the column containing these identifiers contains unique values.

## Specialized String Lengths

In some cases, you might want to limit the lengths of text strings. In this example, your dataset contains a column of zip code values, some of which are in Zip+4 format. Your source data might look like the following:

zip\_code

94104
94104-2218
94105

For consistency, you might want to limit the column to use just the first five digits of the zip code.

**Steps:**

1. Select the first five digits of one of the nine-digit zip codes.
2. In the suggestion cards, select the Extract card.
3. Select the following variation:

<b>Transformation Name</b>	Extract text or pattern
<b>Parameter: Column to extract from</b>	zipcode
<b>Parameter: Option</b>	Custom text or pattern
<b>Parameter: Text to extract</b>	`{zip}`
<b>Parameter: Start extracting after</b>	`{start}`

4. Click **Add**.

The above solution references two Trifacta® patterns to identify elements of the cell value. For more information, see *Text Matching*.

For a more generalized approach, you can use some of the following string functions to limit your data length. Values that are shorter than the designated string length are left untouched.

**NOTE:** Transforms that cut down the size of a value might generate mismatched or missing values based on the column's data type. You should verify that you are not creating new missing or mismatched values.

**Use Rightmost Values**

Use the following transform to reduce a string to the rightmost 6 characters in any value:

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	prodID
<b>Parameter: Formula</b>	RIGHT(prodID, 6)

**Substring Values**

The SUBSTRING function enables you to designate a specific subset of the string's characters to use. You specify the index of the first character in the values and the number of subsequent characters to include. For example, when applied to the value `United States of America` in the `countries` column, the following transform sets the new value to be `States`.

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	countries

<b>Parameter: Formula</b>	SUBSTRING( <i>countries</i> , 7, 6)
---------------------------	-------------------------------------

Note that the index value begins at zero; to extract from the beginning of the value, replace 7 above with 0.

## Additional String Functions

Wrangle supports other functions, which can be used to transform string values. See *String Functions*.