



# Table Steering

## Hands-on Guide

# Contents

What is this document for? -	3
Disclaimer -	3
Table Steering – Use Cases -	4

## What is this document for?

This hands-on guide is designed to help you explore the process of **Table Steering** in Tricentis Tosca and enhance your functional knowledge of the product.

In this document, you are provided with detailed instructions to walk you through the steps to achieve the desired result(s) for the listed assignment(s).

We encourage you to use this document to support your knowledge of Tosca, compare it with your alternate practical implementation method(s), and improve your overall know-how of the tool.

## Disclaimer

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose without the express written permission of Tricentis GmbH.

© 2022 by Tricentis GmbH

# Table Steering – Use Cases

## + Objective

By the end of these exercise(s), you will be able to demonstrate how to:

- ◆ Identify the process of Table Steering using various techniques in Tosca
- ◆ Explain the difference between the notations (\$) and (#) with respect to Table Steering in Tosca
- ◆ Recognize how to deal with specific occurrence of the cell value in Tosca
- ◆ Explore how to use Regular Expressions (REGEX) to steer table rows with similar values in Tosca

## + Why is this important?

Making use of proper Table Steering helps to :

- ◆ Create TestCases with the least amount of automation steps possible
- ◆ Reduce execution duration of TestCases
- ◆ Ensure that the TestCases are resilient to change and as maintainable as possible

## + Steps to perform

Here are the steps you need to perform for Table Steering in Tosca. Please remember to follow the steps in sequence to achieve the desired results.

### - Section I: Prepare the SUT



#### Note

As mentioned in the pre-requisites, you can choose to import the base subset either at the project root level or in a new component folder to begin with the steps below.

- 1 | Login to the **Demo Webshop** Application using the link - <http://demowebshop.tricentis.com/>
- 2 | Navigate to the Accessories page **COMPUTERS** —→ **Accessories**
- 3 | Add all the listed products on the Accessories page, to the Shopping cart except TCP Coaching Day product
- 4 | Navigate to the **Digital Download** Page and add products titled **Music 2** to the Shopping cart
- 5 | Navigate to the **Shopping cart** Page by clicking on the Shopping cart link at the top right corner of the page

# Table Steering – Use Cases

6 | Ensure that your Shopping cart has all the products as shown in the image below:

Remove		Product(s)	Price	Qty.	Total
<input type="checkbox"/>	 NO IMAGE AVAILABLE	TCP Instructor Led Training	9000.00	<input type="text" value="1"/>	9000.00
<input type="checkbox"/>	 NO IMAGE AVAILABLE	TCP Public Complete	3000.00	<input type="text" value="1"/>	3000.00
<input type="checkbox"/>	 NO IMAGE AVAILABLE	TCP Public MT/AT	1700.00	<input type="text" value="1"/>	1700.00
<input type="checkbox"/>	 NO IMAGE AVAILABLE	TCP Public RPA/TCD	1700.00	<input type="text" value="1"/>	1700.00
<input type="checkbox"/>	 NO IMAGE AVAILABLE	TCP Self-Paced Training	400.00	<input type="text" value="1"/>	400.00
<input type="checkbox"/>	 NO IMAGE AVAILABLE	TCP Self-Paced Training additional month	400.00	<input type="text" value="1"/>	400.00
<input type="checkbox"/>	 Music 2		10.00	<input type="text" value="1"/>	10.00
<input type="checkbox"/>	 Music 2		3.00	<input type="text" value="1"/>	3.00

## Section II: Steer the table using \$ notation

In continuation to the last section, go to your Tosca Commander and follow the steps below.

1 | Create a new TestCase in the TestCase folder and name it **Table steering**

2 | Drag and drop the Shopping cart Module from the path mentioned below to the TestCase Table steering:

Modules —→ Webshop —→ Shopping cart

3 | Rename the Shopping cart TestStep to **Understanding the difference between \$ and #**

4 | Specify the Shopping cart products table TestStep values as mentioned below:

TestStepValue	Value	ActionMode
<Row>: Rename it to \$1		Select
<Cell>: Choose Product(s)	Product	Buffer

# Table Steering – Use Cases

 For a visual representation of the previous step(s) in Tosca, refer to the image below :

Details	Control Flow Diagram	Name	Value	ActionMode	DataType	WorkState
		Understanding the difference between \$ and #				
		Shopping cart products		Select	String	
		#1		Select	String	
		Product(s)	Product	Buffer	String	

5| Execute the TestStep in the ScratchBook and verify the results

**Expected result** – You will now notice that Tosca has created the buffer **Product** with the value “TCP Instructor Led Training”

## Section III: Steer the table using # notation

In continuation to the last section, follow the steps below.

1| Update the Shopping cart products table TestStep values as mentioned below:

TestStepValue	Value	ActionMode
<Row>: Rename it to #1		Select
<Cell>: Choose Product(s)	Product	Buffer

 For a visual representation of the previous step(s) in Tosca, refer to the image below :

Details	Control Flow Diagram	Name	Value	ActionMode	DataType	WorkState
		Understanding the difference between \$ and #				
		Shopping cart products		Select	String	
		#1		Select	String	
		Product(s)	Product	Buffer	String	

2| Execute the TestStep in the ScratchBook and verify the results

**Expected result** – You will now notice that Tosca has updated the buffer **Product** value as “Product(s)”

## – Section IV: Modify HeaderRow and observe the change

In continuation to the last section, follow the steps below.

- 1 | Right-click on the TestStep **Understanding the difference between \$ and #** and select **Jump to** from the context menu
- 2 | Open the **Properties** section of the Module attribute **Shopping cart products** by clicking the expand arrow on the right-hand pane 
- 3 | Locate the steering parameter **HeaderRow** and change its value to **2**
- 4 | Navigate back to the TestCase and execute it in the ScratchBook

**Expected Result** – You will now notice that Tosca shows an error stating  
“Cell 'Product(s)' was not found.”

**Reason for the error** – Tosca couldn't locate the Product(s) column because the table header starts from row# 2, which doesn't have any column with the name Product(s). Therefore, steering via column name is not possible.

Remove	Product(s)	Price	Qty.	Total
<input type="checkbox"/>	 TCP Instructor Led Training	9000.00	<input type="text" value="2"/>	18000.00
<input type="checkbox"/>	 TCP Public Complete	3000.00	<input type="text" value="2"/>	6000.00

- 5 | Now, update the **Shopping cart products** table TestStep values as mentioned below:

TestStepValue	Value	ActionMode
<Row>: Rename it to \$1		Select
<Cell>: Rename it to \$3	Product	Buffer

## Table Steering – Use Cases

 For a visual representation of the previous step(s) in Tosca, refer to the image below :

Details	Control Flow Diagram	Name	Value	ActionMode	DataType	WorkState
		Understanding the difference between \$ and #				
		Shopping cart products		Select	String	
		#1		Select	String	
		\$3	Product	Buffer	String	

6 | Execute the TestStep in the ScratchBook and verify the results

**Expected result** – You will now notice that Tosca has updated the buffer **Product** value as "TCP Public Complete"

7 | Next, update the **Shopping cart products** table TestStep values as mentioned below:

TestStepValue	Value	ActionMode
<Row>: Rename it to #1		Select
<Cell>: Rename it to \$3	Product	Buffer

 For a visual representation of the previous step(s) in Tosca, refer to the image below :

Details	Control Flow Diagram	Name	Value	ActionMode	DataType	WorkState
		Understanding the difference between \$ and #				
		Shopping cart products		Select	String	
		#1		Select	String	
		\$3	Product	Buffer	String	

8 | Execute the TestStep in the ScratchBook and verify the results

**Expected result** – You will now notice that Tosca has updated the buffer **Product** value as "Product(s)"

## Table Steering – Use Cases

### – Section V: Remove HeaderRow and observe the change

In continuation to the last section, follow the steps below.

- 1 | Go to the **Shopping cart products** Module attribute and navigate to its **Properties** by clicking the expand arrow on the right-hand pane 
- 2 | Locate and delete the **HeaderRow** steering parameter: Right-click —→ Delete
- 3 | Go back to the TestCase **Understanding the difference between \$ and #** and execute it in the ScratchBook and verify the results

**Expected result** – You will now notice that Tosca successfully executes the TestCase. In both the cases, Tosca captures the value of the buffer **Product** value as "Product(s)"

**Reason for successful execution** – Tosca treated \$ and # as the same when the HeaderRow steering parameter was not specified

### – Section VI: Use constraint to steer specific instance of cell

In continuation to the last section, follow the steps below.

- 1 | Go to the **Shopping cart products** Module attribute and navigate to its **Properties** by clicking the expand arrow on the right-hand pane 
- 2 | Add a Steering Parameter **HeaderRow** and add its value as 1
- 3 | Drag and drop the Shopping cart Module again to the TestCase **Understanding the difference between \$ and #** from the path mentioned below:  
  
Modules —→ Webshop —→ Shopping cart
- 4 | Rename the TestStep to **Dealing with Specific occurrence of the cell value**
- 5 | In the TestStep **Dealing with Specific occurrence of the cell value**, specify the **Shopping cart products** table TestStep values as mentioned below:

TestStepValue	Value	ActionMode
<Row>: Rename it to #2		Select
<Cell>: Choose Product(s)	Music 2	Constraint
<Row>: Choose Price	Price	Buffer

# Table Steering – Use Cases



For a visual representation of the previous step(s) in Tosca, refer to the image below :

Details	Control Flow Diagram	Name	Value	ActionMode	DataType	WorkState
		Dealing with Specific occurrence of the cell value				
		Shopping cart products		Select	String	
		#2		Select	String	
		Product(s)	Music 2	Constraint	String	
		Price	Price	Buffer	String	

6 | Execute the TestStep in the Scratchbook and verify the results

**Expected result** – You will now notice that Tosca stores the price of the second **Music 2** product in buffer **Price**. The value is "**Price: 3.00**" (i.e., the Price of the 2nd Music product)

## Section VII: Use REGEX for table steering

In continuation to the last section, follow the steps below.

1 | Drag and drop the **Shopping cart** Module to the TestCase Understanding the difference between \$ and # from the path mentioned below

Modules —→ Webshop —→ Shopping cart

2 | Rename the TestStep to **Using REGEX to steer table rows with similar values**

3 | Update the **Shopping cart products** table TestStep values as mentioned below:

TestStepValue	Value	ActionMode
<Row>: Rename it to #1		Select
<Cell>: Choose Product(s)	{REGEX["TCP Public MT/AT TCP Public RPA/TCD"]}	Constraint



For a visual representation of the previous step(s) in Tosca, refer to the image below :

Details	Control Flow Diagram	Name	Value	ActionMode	DataType	WorkState
		Using REGEX to steer table rows with similar values				
		Shopping cart products		Select	String	
		#1		Select	String	
		Product(s)	{REGEX["TCP Public MT/AT TCP Public RPA/TCD"]}	Constraint	String	

## Note

You can choose to copy the TestStep value here {REGEX["TCP Public MT/AT|TCP Public RPA/TCD"]} for the update.

- 4 | Execute the TestStep in the Scratchbook and verify the results

**Expected result** – You will now notice that Tosca searches for the presence of either of the values and the results will be successful only if it finds the accurate name for at least one of the products.

## + Observation

- 1 | When # is used in the <Row> attribute, it steers the specific row number irrespective of the HeaderRow steering parameter value
- 2 | When \$ is used in the <Row> attribute, it always steers the content row(s) which are below the table header based on the value defined for the HeaderRow steering parameter
- 3 | Constraint can limit the search to a specific value(s)
- 4 | Tosca allows using regular expressions to minimize automation efforts