TRICENTIS

Test Design Specialist Level 2

STUDENT EXERCISE WORKBOOK

TEST DESIGN SPECIALIST LEVEL 2

Test Design Specialist Level 2 Student Exercise Workbook

Designed to be used with Tricentis Tosca version 12.x

Student Exercise Workbook

This exercise workbook is designed to provide a collection of exercises on the methods and concepts covered in the Tricentis Tosca Test Design Specialist Level 2 Training.

Legal Notice

Tricentis GmbH
Leonard-Bernstein-Straße 10
1220 Vienna
Austria

Tel.: +43 (1) 263 24 09
Fax: +43 (1) 263 24 09-15
Email: academy@tricentis.com

TABLE OF CONTENTS

PREFACE

About this workbook

This workbook is specifically designed to supplement training of the **Test Design Specialist Level 2 12.x** course.

The workbook is arranged in sections. Each section contains a number of exercises which give detailed instructions on how to perform certain functions in Tosca.

Tricentis recommends completing the exercises before continuing to the next section and to take the related online exams in order to achieve high impact learning.

For each exercise there will be a lesson video that explains how to complete the exercise. Most exercises will also have a solution video that explains how to complete the exercise in full.

This workbook is not aiming to be a complete manual.

Recommended learning material

In addition to this workbook, it is necessary to have Tosca Server installed (version specific) in order to complete the exercises successfully.

PREPARING YOUR PROJECT FOR USING TDS

Exercise 01A | Enabling TDS through Tosca Server

Objective

By the end of this exercise, you will be able to install and enable the Test Data Service (TDS) on the Tricentis Tosca Server.

Why is this important?

The Test Data Service (TDS) must be enabled to establish the TDS connection on the Tosca Server through its endpoint.

Instructions

1.	To download "**Tosca Server**", go to the "**Tricentis Support Portal**" and click on "**Downloads**". Make sure to download the version of the Tosca Server which matches the version of Tosca you are currently using. **Note**: The TDS feature has been available from Tosca version 11.0 onwards.	☐
2.	Follow the step by step instructions on installing the "**Tosca Server**" via the "**InstallShield Wizard**" in the Knowledge Base article "Install Tosca Server" available in the "**Support Portal**": https://support.tricentis.com/community/manuals_detail.do?lang=en&version=12.0.0&url=tosca_server/server_component_setup.htm Throughout the InstallShield Wizard installation process, you will be directed to the "**Tricentis Tosca Feature Configuration**". Select the "**Test Data Service**" as the feature to be installed and deselect all other features.	☐
3.	Once "**Tosca Server**" is installed via the "**InstallShield Wizard**", the landing page of the "**Test Data Service**" will automatically launch with the icon of the "**Tosca Test Data Object Viewer**" visible. To access this webpage, type the below **endpoint request** into the browser window: http://<SERVER IP or localhost>:<port-number> E.g. If you have to configure the Tosca Server locally on your system at port number 81, then the endpoint request would be: http://localhost:81	☐
4.	Verify that you are redirected to the landing page of the repositories where a default TDS repository named "**data**" is present.	☐
5.	Download the **Subset** "**TDS2_Base.tsu**" and save it to your local drive.	☐
6.	In Tosca Commander, create a new single user workspace named "**TestDataService**" and select the option "**Use workspace template**".	☐
7.	Navigate to where you saved "T**DS2_Base.tsu**" and select it to import the elements contained within that Subset into your workspace.	☐
8.	Click "**OK**" to finish creating the new Project.	☐

Hints

- If Tosca Server is already installed, make sure the Test Data Service is installed as a feature. If this feature is not installed, navigate to your Windows Control Panel, select "Tosca Server" and then right click on "Modify". It will then be possible to install the required feature on the Tosca Server.
- It is not always mandatory to configure the TestDataService (TDS) at port number 81 on the Tosca Server. Any port that is available and unoccupied can be used.
- Please contact your IT department to ensure that no firewall is blocking your configured port number for further communication between Tosca Commander and Tosca Server.

Exercise 01B | Creation of TDS Repository

Objective

By the end of this exercise, you will be able to create a TDS repository to store and manage your test data.

Why is this important?

It is necessary to create a TDS repository in order to store and manage your test data in a centralized database.

Business Context

The necessary details required to create any TDS repository in Test Data Management should be:

- **Name** = Define a unique name for the new repository
- **Type** = Type of repository
- **Location** = Define the path to the location where the database will be saved .The default location is %PROGRAMDATA%\TestDataService\<reponame>.db
- **Description** = The database type **SQLite** is used for purposes of this training. (It is also possible to create an **In Memory** repository type where test data is deposited temporarily in the computer's memory. Use the property **Location** for unique identification.)

Instructions

1.	Open the "**Tosca Server**" URL (**http://<SERVER IP or Localhost>:<Port-number>**) in your browser. i.e http://localhost:81 You will be navigated to the landing page of the "**Tosca Server**".	☐
2.	Click on the "**Test Data Object Viewer**" icon on the Tosca Server home page. You will be navigated to the **Test Data Management** page, which displays all the repositories.	☐
3.	Click on "**Create Repository**".	☐
4.	Provide the necessary details below: • **Name** = TDS2 • **Description** = This repository is created for the TDS2 certification course **Note**: The "**Type**" and "**Location**" fields have default Values. There is no need to modify them.	☐
5.	Save your changes. This will automatically direct you to the TDS2 repository that you just created.	☐
6.	By clicking on your newly created TDS2 repository on the "**Test Data Management**" page, you will be able to see all your test records in a table-like structure.	☐

Exercise 01C | Connect to TDS Repository

Objective

By the end of this exercise, you will be able to establish a connection between the TDS repository and the Tosca Client (Commander) through TDS Configurations.

Why is this important?

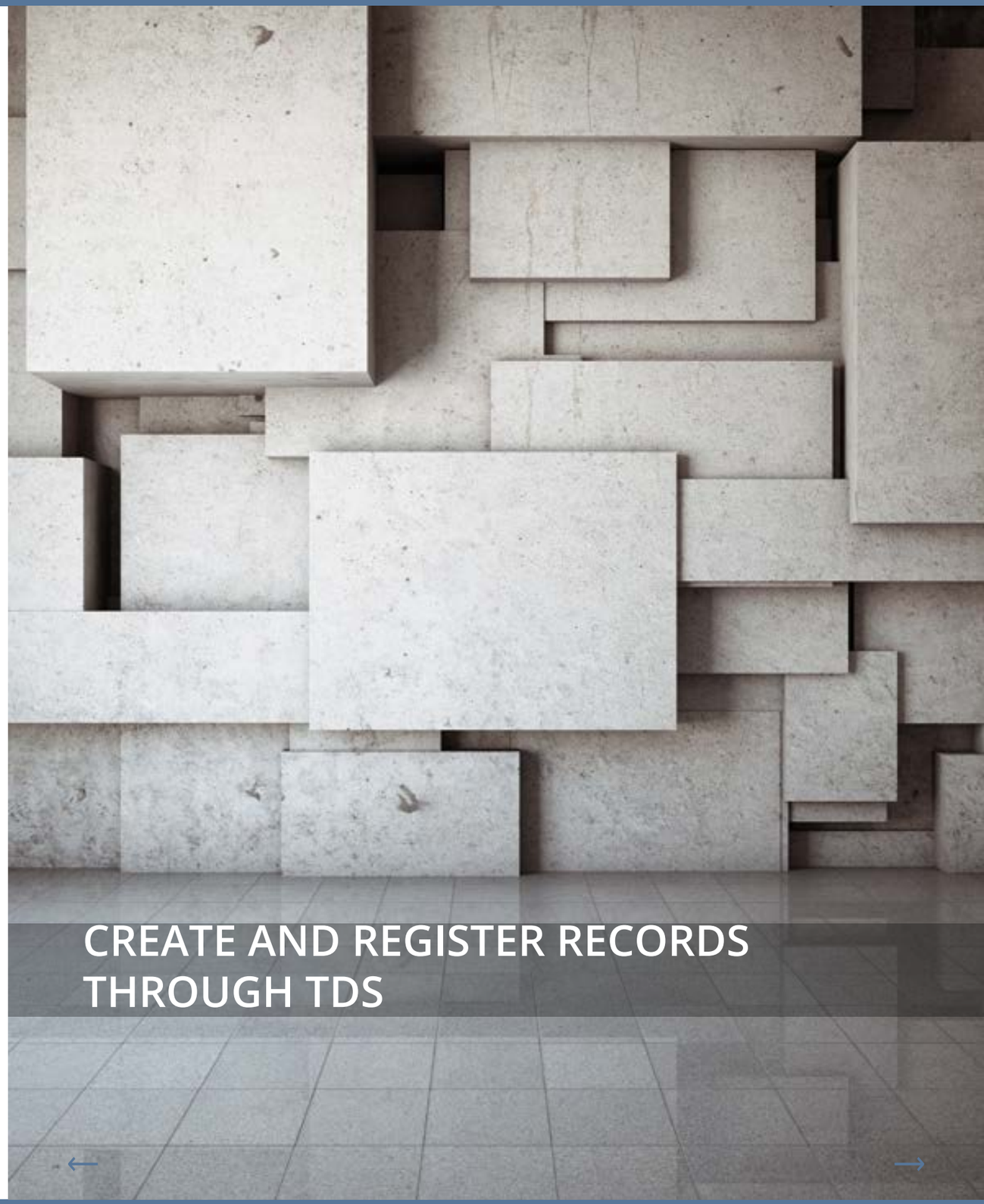
In order to manage the test data in the TDS repository, it is necessary to establish a connection between the database and the Tosca Client. This is achieved using TDS configurations in Tosca Commander.

Instructions

1.	In your opened Project "**TestDataService**" in "**Tosca Commander**", navigate to the "**Configurations**" Folder by clicking on the "**Home**" ribbon and select "**Project**".	☐
2.	Navigate to "**TestDataService Configurations**". Drag and drop this "**Configurations**" onto the TestCases Root Folder and then switch to the "**Test configuration**" tab.	☐
3.	Define the values of the TDS Test configuration parameters as shown below: **TestDataEndPoint** = http://localhost:81/TestDataService **TestDataRepository** = TDS2	☐
4.	Save the changes.	☐

Hints

- **TestDataEndPoint** is the TDS endpoint request which was created in Exercise 01A. For your reference, it is (http://<SERVER IP or localhost>:<port-number>/TestDataObjectViewer)
- **TestDataRepository** is the name of the repository which the user created in Exercise 01B.
- If the user did not link any of the TDS configurations to the TestCase Root Folder, then the following exception will pop up: "*The Setting or TestConfigurationParameter 'TestDataEndpoint' was not found!*"
- If the user entered an incorrect Value of the Configuration Parameter TestDataEndPoint, then the following exception will pop up: "*The server returned an error that could not be interpreted. Please make sure the endpoint specified in the test configuration parameter 'TestDataEndpoint' is reachable and the server is configured properly.*"



CREATE AND REGISTER RECORDS THROUGH TDS

Exercise 02A | Register Test Records through TDS

Objective

By the end of this exercise, you will be able to create test records in the TDS repository by using TDS Standard Modules.

Why is this important?

In order to enter test records into the database of the TDS repository, we have to create a TestCase using TDS Standard Modules. This exercise will focus on creating a simple TestCase for you to get used to the concept of creating new test records through the TDS Standard Modules.

Business Context

The name provided for the “Existing or new TDS type*“ will be the name displayed under “Types“ on the Test Data Management page.
“Alias name (record)“ is a name assigned to specify a certain record in a dataset. This field is optional. If left empty, the TDS type will be used as the Alias name. The main purpose of using an Alias name is to distinguish between record types.

Instructions

1. Create a **TestCase** in the Folder "**Exercise 02A**" named “**Register TestData through TDS**“.
   Add the TDS Standard Module "**TestData – Create & provide new record**" into this TestCase and rename the TestStep to “**Create MyAddress**“. ☐
2. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
Create MyAddress	Existing or new TDS type*	MyAddress	Input
	Expand the Data structure and enter the Values for the TDS Attributes as below:		Select
	Name	Mr Test	Input
	Address	Training Street 1	Input
	Zipcode	1234	Input

3. Run the TestCase in the **Scratchbook**. ☐
4. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐

Hints

» Please note, the objects within the TestStepValue column such as “Name“, “Address“ and “Zipcode“ are referred to as **TestStepValues** in Tosca. However, these same objects are also referred to as **TDS Attributes** within the TDS repository. Depending on the context, these two terminologies are used interchangeably. This naming convention applies to all objects displayed in the TestStepValue column when working with TDS.

Exercise 02B | View Test Records in Test Data Management

Objective

By end of this exercise, you will know where to view the registered test records in the TDS repository by familiarizing yourself with the Test Data Object Viewer and the Test Data Management page.

Why is this important?

To view all your created test records in the TDS repository, you must access the Test Data Object Viewer. This directs you to the Test Data Management webpage, which contains all your data repositories and test records within them.

Business Context

The Types column on the left displays all the database types, such as “MyAddress“ which was created by the TestCase in the previous exercise. The green number displayed next to the TDS Type indicates the number of records (or data entries) of this type. You can view these records by clicking on the respective type in the "Types" column.

Instructions

1. Open the landing page of the "**Test Data Object Viewer**" through your configured ToscaServer URL in the web browser:
   http://<SERVER IP or localhost>:<port-number>/TestDataObjectViewer
   For example, the sample URL would look like: http://localhost:81/TestDataObjectViewer ☐
2. Click on the "**TDS2 repository**" created as part of the previous exercise. ☐
3. Underneath the "**Types**" column on the left, you will see the TDS Type "**MyAddress**". This contains a table-like structure with all the data entries or records from the previous exercise. ☐

Hints

» By default, the TDS records created in the respective TDS type are in a locked state. After you have retrieved a record, it is locked. This means, no one else in a distributed environment can use that test record in a TestStep. The record, however, becomes automatically unlocked once you have executed your test.

Exercise 02C | Register a Customer in the Web Shop

Objective

By end of this exercise, you will register a new customer in the DemoWebShop application and enter this new customer data into the TDS repository through Test Data Services. This exercise is an extension of Exercise 02A.

Why is this important?

In order to create and enter test records of newly registered customers in the TDS repository, we will be using the DemoWebShop application which will be the building block for the rest of the course.

Business Context

Some customer data such as FirstName, LastName and Email have already been saved as Buffers as part of the imported TDS2 Base Subset, as this is an important part of the registration process for the DemoWebShop.
Once the TestCase "Register new customer" is executed, this will register one customer in the DemoWebShop application. This will also create a new data entry in the TDS repository, namely, a customer with the Status "New".
The Values of the TDS Attributes NewsLetter, Items, OrderNumber, OrderStatus have been left blank as they will be updated later in subsequent exercises. These TDS Attributes will also be used later as search criteria when updating them progressively.

Instructions

1. Navigate to the **TestCase** "**Register new customer**" in the "**Exercise 02C**" Folder. ☐
2. Run the **TestCase** in the **Scratchbook** as part of the imported **TDS2 Base Subset**. ☐
3. Navigate to the TestStepFolder "**Write customer details in TDS**" and add the TDS Standard Module: "**TestData - Create & provide new record**" ☐
4. Rename the "**TestData - Create & provide new record**" TestStep to "**Provide the customer details to be entered in TDS**". ☐
5. Enter the data as per the table below: ☐

TestStep Name	TestStep Value	Value	Action Mode
TestStepFolder **Write customer details in TDS**			
Provide the customer details to be entered in TDS	Existing or new TDS type*	CustomerDetails	*Set the correct Action Mode*
	Expand the Data structure and enter the Values for the TDS Attributes as below:		*Set the correct Action Mode*
	FirstName	*Enter the correct Buffer*	*Set the correct Action Mode*
	LastName	*Enter the correct Buffer*	*Set the correct Action Mode*
	Email	*Enter the correct Value*	*Set the correct Action Mode*
	NewsLetter	*Leave blank*	*Leave blank*
	Status	*Enter the correct Value*	*Set the correct Action Mode*
	Items	*Leave blank*	*Leave blank*
	OrderNumber	*Leave blank*	*Leave blank*
	OrderStatus	*Leave blank*	*Leave blank*

6. Run the TestCase in the **Scratchbook**. ☐
7. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
8. Refresh the **Test Data Management** page in the web browser to **view** the newly created test records. ☐

Hints

» The TDS Attribute "Status" will be used later as search criteria to look for customers in that state.

Exercise 02D | Create Multiple Customers

Objective

By the end of this exercise, you will be able to create multiple records of customer data in the TDS repository using the DemoWebShop application.

Why is this important?

This exercise demonstrates how to create a large amount of customer records in the TDS repository.

Business Context

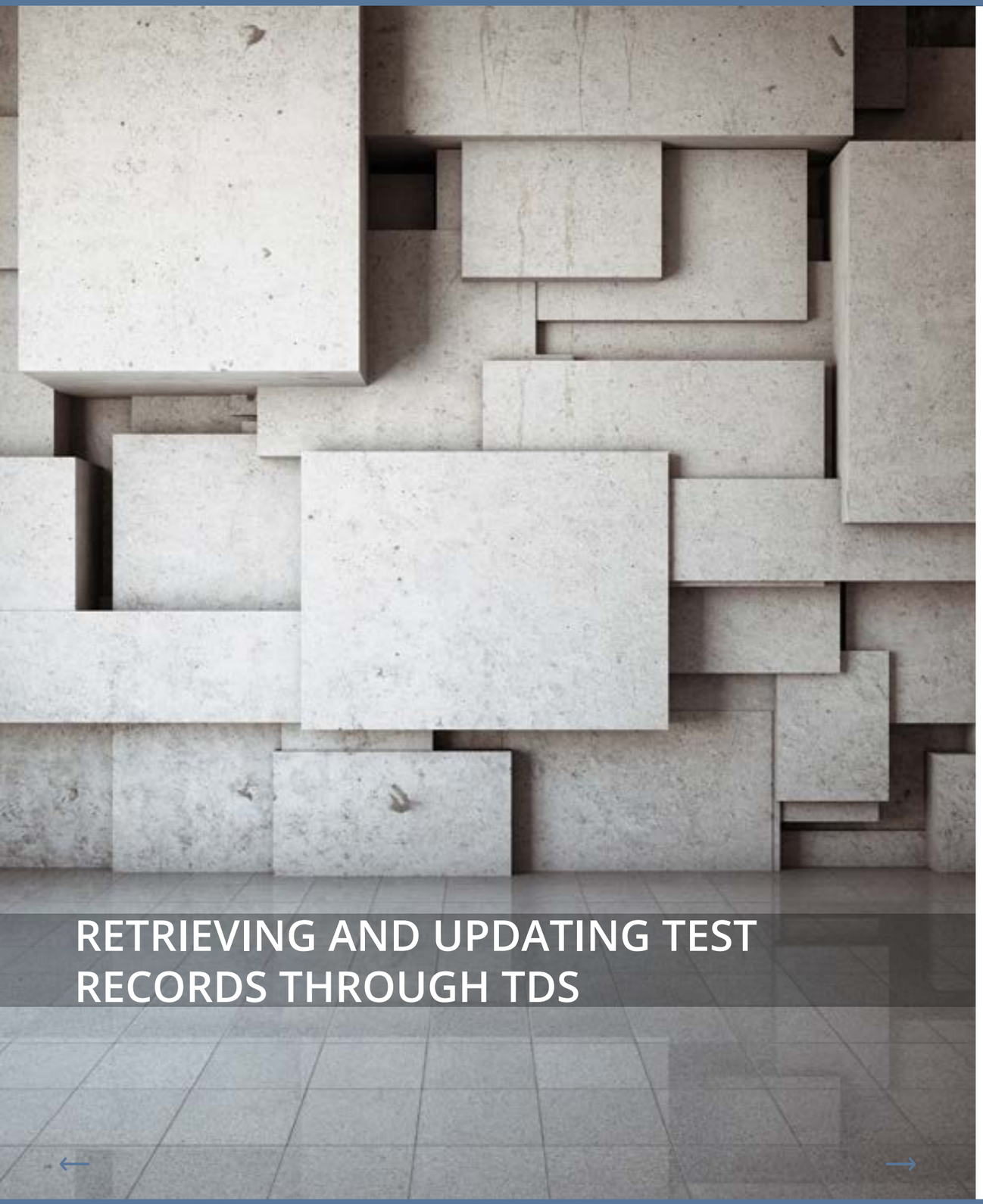
Using Buffers in the TestCase as well as modifying the Repetitions property Value allows you to generate a large amount of test data very easily for Test Data Management.

Instructions

1. Create an "**ExecutionList**" called "**Multiple customers**" within the "Exercise 02D" ExecutionList Folder. ☐
2. Drag and drop the **TestCase** "**Register new customer**" onto this ExecutionList. ☐
3. Navigate to the newly created **ExecutionEntry** "**Register new customer**", and in the context menu, select the command **Modify -> Repetitions**. ☐
4. Enter a **Value** of 5 in the "**Replace with**" field and click on "**Modify All**". ☐
5. Run the **ExecutionList**. Refresh/reopen the **Test Data Management page** in the web browser to view the newly created test records. ☐
6. Refresh the **Test Data Management** page in the web browser to **view** the newly created test records. ☐
7. Save your changes. ☐

Hints

» The Repetitions Value can also be changed in the Properties pane when selecting the relevant ExecutionEntry.



RETRIEVING AND UPDATING TEST RECORDS THROUGH TDS

RETRIEVING AND UPDATING TEST RECORDS THROUGH TDS

Exercise 03A | Retrieve Test Records through TDS

Objective

By the end of this exercise you will able to retrieve test records which are stored in the TDS repository with the help of the TDS Standard Module "TestData - Find & provide record".

Why is this important?

In order to succesfully update a test record, we must first find or retrieve it in the TDS repository, and then perform an update on it. In this exercise, we will look for a customer with the Status “New“.

Business Context

In this exercise, we will look for a customer with the Status “New“.
Once we retrieve a test record, it becomes automatically locked, meaning that no other tester can use this test record in their TestCase. This ensures consistency and accuracy with the state of the data when working in a distributed environment. Once you have executed your test, the record become automatically unlocked.

Instructions

1. Navigate to the Folder **“Exercise 3A“** in the TestCase section. ☐
2. Add the TDS Standard Module **"TestData - Find & provide record“** into the TestStepFolder **“Find for New Customer in TDS records“** and rename it to **“Find TDS record“**. ☐
3. Enter the data in the table below to find/retrieve a customer with the Status "New": ☐

TestStep Name	TestStepValue	Value	Action Mode
Find TDS record	Existing or new TDS type*	*Enter the Value for the TDS Type*	*Set the correct Action Mode*
	Expand the Data search filter and enter the Values for the TDS Attributes as below:		*Set the correct Action Mode*
	Status	*Enter the correct Value*	*Set the correct Action Mode*

4. Navigate to the TestStepFolder **“Login with New Customer in Web Shop“** and jump to the TestStep **“Login Page“**. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
Login Page	Email	***Enter t**he correct Value using the {TD[….]} function*	*Set the correct Action Mode*

5. Run the TestCase in the **Scratchbook**. ☐
6. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
7. Refresh the **Test Data Management** page in the web browser to **view** the changes in the TDS repository. ☐

Hints

» The {TD[….]} function is a command used to retrieve a specific record from the TDS repository. The syntax is similar to the syntax used to retrieve variables from the Buffer. The Value in the square brackets relates to the record you wish to retrieve. In this case, it is the Value stored under the TDS type “CustomerDetails“ with the Value “Email“.

Exercise 03B | Update Test Records through TDS

Objective

By the end of this exercise, you will able to update the existing TDS records present in the TDS repository with the help of the TDS Standard Module "TestData - Update Record".

Why is this important?

This exercise demonstrates how to update test records in your database after necessary changes have been made in the DemoWebShop application.

Business Context

In the DemoWebShop, a customer becomes “Active“ once they have added an address. These changes are reflected in the TDS repository, with the Status of that customer changing from "New" to "Active".

Instructions

1. Navigate to the TestCase “**Update customer address record**“ in the Folder “**Exercise 3B**“. ☐
2. Copy and paste the TestStep from the TestStepFolder “**Find for New Customer in TDS records**“ in "**Exercise 3A**" into the TestStepFolder "**Find for New Customer in TDS records**" in Exercise 3B. ☐
3. In the TestStepFolder “**Add an Address**“, navigate to the TestStep “**Addresses | New address**“ and retrieve the test records for “**FirstName**“ and “**LastName**“ from the TDS repository to enter those customer details in the DemoWebShop application, as per below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Add an Address**			
Addresses | New address	FirstName	*Enter the correct Value using the {TD[....]} function*	*Set the correct Action Mode*
	LastName	*Enter the correct Value using the {TD[....]} function*	*Set the correct Action Mode*

4. Add the TDS Standard Module "**TestData - Update Record**“ to the TestStepFolder “**Update TDS record with Active Status**“ and rename the TestStep to "**Update TDS record**“.

   Enter the appropriate Values for the TestStepValues **Existing alias name (record)*** and **Status** to update the customer details to "Active" once they have added an address. ☐
5. Run the TestCase in the **Scratchbook**. ☐
6. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
7. Refresh the **Test Data Management** page in the web browser to **view** the changes in the TDS repository. ☐

Hints

» Updating any TDS records in your database does not work solely with using the TDS Standard Module "TestData - Update Record“. First, you must find and retrieve the record with the help of the TDS Standard Module "TestData - Find & provide record“, and then perform an update on it.

Exercise 03C | Add Items to the Shopping Cart

Objective

By the end of this exercise, you will be able to add items to the shopping cart in the DemoWebShop application and simultaneously update the Item’s record for the relevant Active customer in the TDS repository.

Why is this important?

This exercise further demonstrates how to update test records in the TDS repository when a customer adds items to the shopping cart using TDS Standard Modules in Tosca.

Business Context

Some data such as "Items" has been saved as a Buffer as part of the imported TDS2 Base Subset. When the TestCase is run, a dynamic Value is buffered using {XB[Items]}, which stores the number of items added to the shopping cart.
In order to update the "Items" column in the TDS repository, we must first create a TestStep in Tosca that finds/retrieves an "Active" customer test record. Subsequently, we update that test record, namely the "Items" column in the TDS repository using the appropriate TDS Standard Module.

Instructions

1. Navigate to the TestCase “**Add items to shopping cart**“ in the Folder “**Exercise 3C**“. ☐
2. In the TestStepFolder “**Search for Active Customer in TDS records**“, add the TDS Standard Module "**TestData - Find & provide record**“ and rename the TestStep to “**Search TDS record with Active Status**“.

   Enter the appropriate Values for the TestStepValues **Existing alias name (record)*** and **Status** to find/retrieve the appropriate test record (i.e. constrain the search for an "Active" customer) in the TDS repository. ☐
3. Add the relevant **TDS Standard Module** in the TestStepFolder **“Update TDS record with Items in Shopping cart“** and rename the TestStep to **“Update TDS record with buffered Items“**.

   Enter the appropriate Values to update the Buffered number of items in the TDS repository ☐
4. Run the TestCase in the **Scratchbook**. ☐
5. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
6. Refresh the **Test Data Management** page in the web browser to **view** the changes in the TDS repository. ☐

Exercise 03D | Purchase Items in the DemoWebShop

Objective

By the end of this exercise, you will be able to purchase the items that have already been added to the shopping cart in the DemoWebShop application. You will simultaneously update the OrderNumber generated and set the OrderStatus to Pending in the TDS repository for an Active customer.

Why is this important?

This exercise demonstrates how to perform two search criteria: 1) Active customer 2) Buffered items added into the cart. This is used in order to find those relevant TDS records and complete the order process of an Active customer. Once this is completed, an update is performed in the test records for Items, OrderNumber and OrderStatus in the TDS repository.

Business Context

The data for "Items" and "OrderNumber" have been saved as Buffers in the TestSteps as part of your imported base Subset. We will need to change the "OrderStatus" to "Pending" to indicate the beginning of the customer purchasing process.

Instructions

1. Navigate to the TestCase "**Purchase items in the web shop"** in the Folder "**Exercise 3D**". ☐
2. In the TestStepFolder "**Search for Active Customer in TDS records with Items in Cart**", add the TDS Standard Module "**TestData - Find & provide record**" to constrain the search for an "Active" customer and the Buffered number of items in the shopping cart. Rename the TestStep to "**Search Active customers with Items in cart**". ☐
3. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder "**Search for Active Customer in TDS records with Items in Cart**"			
Search Active customers with Items in cart	Existing TDS type*	*Enter the Value for the TDS Type*	*Set the correct Action Mode*
	Expand the Data structure and enter the Values for the TDS Attributes as below:		*Set the correct Action Mode*
	Status	*Enter the correct Value*	*Set the correct Action Mode*
	Items	*Enter the correct Value*	*Set the correct Action Mode*

4. In the TestStepFolder "**Update TDS records with Order Number, Order Status & Items in Cart**", add the relevant **TDS Standard Module** to update the test record and rename the TestStep to "**Update TDS records**".

   Enter the appropriate Value and Action Mode for the TestStepValue **Existing alias name (record)***. Buffer the TestStepValues **Items** as well as **OrderNumber**, and set the **OrderStatus** to "Pending". ☐
5. Run the TestCase in the **Scratchbook**. ☐
6. Update the **WorkState** of TestCase to **COMPLETED** and save your changes. ☐
7. Refresh the **Test Data Management** page in the web browser to **view** the changes in the TDS repository. ☐

Exercise 03E | Process Payments through Non-UI

Objective

By the end of this exercise, you will able to process the customer's order payment in the DemoWebShop created in the previous exercises to its desired state through non-UI (API) testing.

Why is this important?

In order to further validate the created order, we need to process the customer's payments to its desired state using non-UI (API) Modules.

Business Context

In order to process the customer's payment, we must first search for a customer with a "Pending" OrderStatus. Then, we will use non-UI Modules to change the OrderStatus to "Processing" in the backend of the DemoWebShop.

Instructions

1. Navigate to the TestCase "**Process payments through non-UI"** in the Folder "**Exercise 3E**". ☐
2. In the TestStepFolder "**Search for Pending OrderStatus in TDS records"**, add the relevant "**TDS Standard Module"** to constrain the search for the "Pending" OrderStatus (different than Status) and rename the TestStep to "**Search for pending OrderStatus"**. Enter the appropriate Values and Action Modes. ☐
3. Navigate to the TestStepFolder "**Payments are processed**" which contains the non-UI TestSteps. ☐
4. Expand all the items under the TestStep of the Request node named "**SetOrderPaymentPaid**" and enter the relevant test data from the TDS records as per the table below:

   Note: It is not necessary to modify any other information in this TestCase. ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Payments are processed**			
SetOrderPaymentPaid	orderId	{TD[CustomerDetails.OrderNumber]}	Insert
	usernameOrEmail	{TD[CustomerDetails.Email]}	Insert
	userPassword	Tricentis1	Insert

5. Run the TestCase in the **Scratchbook**. ☐
6. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐

Hints

» Be careful not to confuse "OrderStaus" (the status of the order) with "Status" (the status of a customer).

Exercise 03F | Process Purchase Order

Objective

By the end of this exercise, you will use TDS Standard Modules to update the OrderStatus from "Pending" to "Processing" in the TDS Repository in order to indicate that the customer's order payments have been received. We will also change the Status of that customer to "Used" in the TDS repository.

Why is this important?

This exercise demonstrates the final steps of the End-to-End customer order process. As the earlier non-UI TestCase verified the customer's payment and changed the OrderStatus to "Processing" in the backend of the DemoWebShop, we must now reflect these changes in the TDS repository. This is particularly important when working in a distributed environment to make other testers aware of the used/consumed TDS records and keep the data stateful.

Business Context

To reflect the changes in the TDS repository, we must first change the OrderStatus from "Pending" to "Processing". We must also change the Status of that customer to "Used" to show that their order has been successfully processed.

Instructions

1. Navigate to the TestCase "**Process purchase order**" in the Folder "**Exercise 3F**". ☐
2. In the TestStepFolder "**Search for Pending OrderStatus in TDS records**", add the relevant **TDS Standard Module** to constrain the search to a "Pending" OrderStatus (different than Status) and rename the TestStep to "**Search for pending OrderStatus**". Enter the appropriate Values and Action Modes. ☐
3. In the TestStepFolder "**Update TDS records with Processing Order Status**", add the relevant **TDS Standard Module** to update the TDS record and rename the TestStep to "**Update Status & OrderStatus**".

   Enter the appropriate Values for the TestStepValues **Existing alias name (record)***, "**OrderStatus**" and "**Status**" with their appropriate Action Modes to indicate the relevant updates of the completed customer order process. ☐
4. Run the TestCase in the **Scratchbook**. ☐
5. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
6. Refresh the **Test Data Management** page in the web browser to **view** the changes in the TDS repository. ☐

Exercise 03G | Move used records to other TDS Type

Objective

By the end of this exercise, you will move the "Used" customer records from the existing TDS type to another TDS type using the TDS Standard Module "TestData - Move record to TDS type".

Why is this important?

Once data has been consumed after the End-to-End process has been completed, it is important to move those test records to an new TDS type for efficient test record management in a distributed environment. This also ensures consistency and accuracy in the test data, so that used or modified test records do not get reused, which would create errors in the TestCase.

Business Context

Here, we first search for the customer records with the Status "Used" and move those test records to a new TDS Type named "CustomerDetails_Used". As the Alias name (record) field is kept blank, it automatically takes the existing TDS Type as an Alias name. This means, when moving those test records, the Existing alias name (record)* will take the same Value as the existing TDS Type.

Instructions

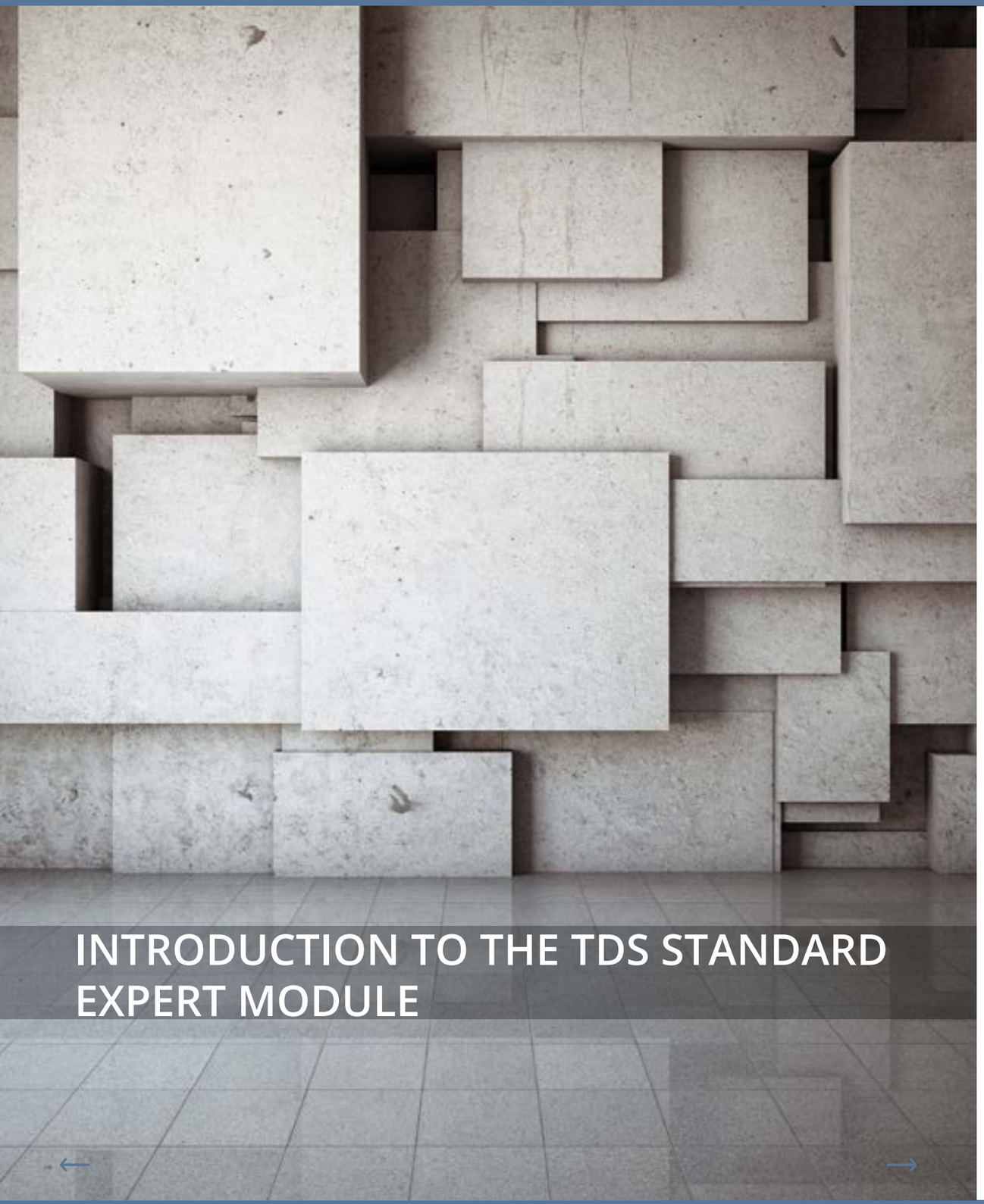
1. Navigate to the TestCase "**Move used record to other TDS Type**" in the Folder "**Exercise 3G**". ☐
2. In the TestStepFolder "**Search for used TDS records**", add the relevant **TDS Standard Module** to constrain the search to a customer with the "Used" Status and rename the TestStep to "**Search for used customer status**". Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStep Folder **Search for used TDS records**			
Search for used customer status	Existing or new TDS type*	*Enter the Value for the TDS Type*	*Set the correct Action Mode*
	Expand the Data search filter and enter the Values for the TDS Attributes as below:		*Set the correct Action Mode*
	Status	*Enter the correct Value*	*Set the correct Action Mode*

3. In the TestStepFolder "**Move used records to other TDS type**", add the relevant TDS Standard Module to move the TDS records to a new TDS Type. Rename the TestStep to "**Move used customer records**" and enter the appropriate Values and Action Modes as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStep Folder **Move used records to other TDS type**			
Move used customer records	Existing or new TDS type*	CustomerDetails_Used	*Set the correct Action Mode*
	Existing alias name (record)*	*Enter the correct Value*	*Set the correct Action Mode*

4. Run the TestCase in the **Scratchbook**. ☐
5. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
6. Refresh the **Test Data Management** page in the web browser to **view** the newly created TDS Type. ☐



INTRODUCTION TO THE TDS STANDARD EXPERT MODULE

INTRODUCTION TO THE TDS STANDARD EXPERT MODULE

Exercise 04A | Introduction to the TestData - Expert Module (1)

Objective

By the end of this exercise, you will learn how to use the TestData – Expert Module which combines the functionalities of all the other stand-alone TDS Standard Modules that we have covered so far. Additionally, this exercise demonstrates the usage of the Create, ReadOnly, Find and Update properties of the TestStepValue Test data task* when working with TestData – Expert Module.

Why is this important?

Instead of using separate TDS Standard Modules for creating, finding and updating test records, it is possible to use the TestData – Expert Module as a unique Standard Module for serving all our operations under one hood.

Business Context

To create multiple orders, we must change the Repetition property Value on the relevant TestStepFolder. The same Value must be entered when reading that many number of test records in the TDS repository.
As we will be cancelling a single order first, we must first find the relevant OrderStatus according to YearofManufacture and then update that OrderStatus to Cancelled. In the second scenario, we find the remaining orders according to OrderStatus "Ordered" and update their OrderStatus accordingly.

Instructions

1. Navigate to the "**Exercise 4A**" Folder in the TestCase section. ☐
2. Add the TDS Standard Module "**TestData - Expert Module**" into the "**Create Multiple Orders**" TestStepFolder and rename the TestStep to "**Create Order**". ☐
3. On the TestStepFolder "**Create Multiple Orders**", change the Repetition property to a Value of 3 in order to generate that many number of orders. ☐
3. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Create Multiple Orders**			
Create Order	Test data task*	Create	Input
	TDS type	SampleOrders	Input
	Alias name (record)	order	Input
	Expand the Data structure and enter the Values for the TDS Attributes as below:		Select
	OrderID	123456	Input
	Product	TricentisHoodie	Input
	Quantity	2	Input
	Total	$60	Input
	OrderStatus	Ordered	Input
	YearofManufacture	2019	Input

4. Run the TestStepFolder "**Create Multiple Orders**" in the Scratchbook and verify the results in the **Test Data Management page** of **TDS**. ☐
5. Add the TDS Standard Module **"TestData - Expert Module**" into the "**Read Created Order**" TestStepFolder and rename it to "**Read Order**". ☐
6. Change the Repetition property on the relevant TestStepFolder to the appropriate Value in order to read that many number of orders. ☐

7. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Read Created Order**			
Read Order	Test data task*	ReadOnly	Input
	TDS type	SampleOrders	Input
	Alias name (record)	order	Input
	Expand the Data structure and enter the Values for the TDS Attributes as below:		Select
	OrderID	123456	Verify
	Product	TricentisHoodie	Verify
	Quantity	2	Verify
	Total	$60	Verify
	OrderStatus	Ordered	Verify
	YearofManufacture	2019	Verify

8. Run the TestStepFolder "**Read Created Order**" in the Scratchbook. ☐

9. Save your changes. ☐

10. Navigate to the TestStepFolder "**Find and Update the first Order to Cancelled state"** and add the TDS Standard Module "**TestData - Expert Module**" into the "**Find Created Order by YearofManufacture**" subfolder and rename the TestStep to "**Find Order**". ☐

11. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Find and Update the first Order to Cancelled state**			
Sub TestStepFolder **Find Created Order by OrderStatus**			
Find Order	Test data task*	Find	Input
	TDS Type	SampleOrders	Input
	Alias name (record)	order	Input
	Expand the Data structure attibute and Enter below TDS attributes and its values		Select
	YearofManufacture	>2010	Constraint
	YearofManufacture	<2020	Constraint

12. Navigate to the subfolder "**Update Order Status to Cancelled"** and add the TDS Standard Module "**TestData - Expert Module**" and rename the TestStep to "**Update OrderStatus**". ☐

13. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Find and Update the first Order to Cancelled state**			
Sub TestStepFolder **Find Created Order by YearofManufacture**			
Find Order	Test data task*	Update	Input
	TDS Type	SampleOrders	Input
	Alias name (record)	order	Input
	Expand the Data structure attibute and Enter below TDS attributes and its values		Select
	OrderStatus	Cancelled	Input

14. Run the TestStepFolder "**Find and Update the first Order to Cancelled state**" in the Scratchbook and verify the results in **Test Data Management page** of the **TDS**. ☐

15. Navigate to the TestStepFolder "**Find and Update the subsequent Orders to Cancelled state**" and add the TDS Standard Module "**TestData - Expert Module**" into the "**Find Created Order by OrderStatus**" subfolder and rename the TestStep to "**Find Order**".

    Enter the appropriate Values to constrain the search to the OrderStatus "Ordered". ☐

    Note: On the TestStepFolder "**Find and Update the subsequent Orders to Cancelled state**", change the Repetition property to a Value of 2 in order to find and update that many number of orders.

16. Add the TDS Standard Module "**TestData - Expert Module**" into the "**Update Order Status to Cancelled**" subfolder and rename the TestStep to "**Update OrderStatus**".

    Enter the appropriate Values to update the OrderStatus to "Cancelled".

17. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐

18. Run the TestStepFolder "**Find and Update the subsequent Orders to Cancelled state**" in the Scratchbook and verify the results in **Test Data Management page** of the **TDS**. ☐

Hints

- The Test data task* "Update" property works always in conjuction with the Test data task* "Find" property. This means, the relevant TDS record is first found and retrieved from the database and then an update is performed on it.
- For the mathematical comparison operators (<,>,==,!=,>=,<=), the DataType must be changed from String to Numeric. These operators can be adjusted by clicking on the blue dropdown arrow when selecting the relevant Value.
- The Alias name (record) used here is unique name for your record for further usage.

Exercise 04B | Introduction to the TestData - Expert Module (2)

Objective

By the end of this exercise, you will learn how to use additional functionalities of the TestData – Expert Module such as deleting TDS records and deleting TDS types. This exercise demonstrates the usage of the "DeleteRecord" and "DeleteType" properties of the Test data task* TestStepValue when working with the TestData – Expert Module.

Why is this important?

It is useful to delete specific test records from your database which might have been entered incorrectly or if its state is no longer of use, such as a cancelled order in the DemoWebShop. In such cases, we will first find the relevant record and perform the "DeleteRecord" operation. It is also useful to know how to delete an entire TDS type which may no longer be of use, such as records containing all cancelled orders.

Business Context

To delete multiple cancelled orders, we must find all the cancelled orders and change the Repetition property Value on the relevant TestStepFolder. The same Value must be entered when deleting that many number of test records in the TDS repository.

Instructions

1. Navigate to the "**Exercise 4B**" Folder in the TestCase section. ☐
2. Navigate to the TestStepFolder "**Find and Delete all Cancelled Orders**" and add the TDS Standard Module "**TestData - Expert Module**" into the "**Find Order by OrderStatus**" subfolder and rename the TestStep to "**Find Order**". ☐
3. Change the Repetition property on the "**Find and Delete all Cancelled Orders**" TestStepFolder to the Value of 3 to find that many number of orders. ☐
4. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Find and Delete all Cancelled Orders**			
Sub TestStepFolder **Find Order by OrderStatus**			
Find Order	Test data task*	Find	Input
	TDS Type	SampleOrders	Input
	Alias name (record)	order	Input
	Expand the Data structure and enter the Values for the TDS Attributes as below:		Select
	OrderStatus	Cancelled	Input

5. Add the TDS Standard Module "**TestData - Expert Module**" into the "**Delete Cancelled Order**" subfolder and rename the TestStep to "**Delete Order**". ☐
6. Enter the data as per the table below:

TestStep Name	TestStepValue	Value	Action Mode
Sub TestStepFolder **Delete Cancelled Order**			
Delete Order	Test data task*	DeleteRecord	Input
	TDS Type	SampleOrders	Input
	Alias name (record)	order	Input
	Expand the Data structure and enter the Values for the TDS Attributes as below:		Select
	OrderStatus	Cancelled	Input

7. Run the TestStepFolder "**Find and Delete all Cancelled Orders**" in the Scratchbook and verify the results in **Test Data Management page** of the **TDS**. ☐
8. Save your changes. ☐
9. Add the TDS Standard Module "**TestData - Expert Module**" into the "**Delete SampleOrders TDS type**" TestStepFolder and rename the TestStep to "**Delete TDS Type**". ☐
10. Enter the data as per the table below: ☐

TestStep Name	TestStepValue	Value	Action Mode
TestStepFolder **Delete SampleOrders TDS type**			
Delete TDS Type	Test data task*	DeleteType	Input
	TDS Type	SampleOrders	Input
	Alias name (record)	order	Input

11. Update the **WorkState** of the TestCase to **COMPLETED** and save your changes. ☐
12. Run the TestStepFolder "**Delete SampleOrders TDS type**" in the Scratchbook and verify the results in **Test Data Management page** of the **TDS**. ☐

Hints

» The property "DeleteAll" of the TestStepValue **Test data task*** needs to be used cautiously as it deletes all the contents of the Tricentis Tosca Test Data Service database and basically formats the TDS repository. When executing this function, there is no need to specify the Values for the TDS Type and Alias name (record) TestStepValues respectively.