Test Case Design in TOSCA

How to design an automated TestCase in TOSCA

Author:	Archana Singh
Version:	1.2
Version date:	29.08.2017

0 Organisational Topics

0.1 Obsolescence

Version	Date	Contact	Change	Reviewer
V1.0	29.06.2017	Archana Singh	Initial creation of document.	Subroto Roy
V1.1	29.06.2017	Archana Singh	Review comments by Subroto.	Avinash More
V1.2	29.06.2017	Archana Singh	Review comments by Avinash and Subroto.	Subroto Roy

Contents

1. Overview

1.1 Goal

This Document covers the recurrent test case designing in Tosca and also the whole designing process followed by the Automation and Functional teams in the ABS Test center.

1.2 Tools Used

ALM- Designer can take the reference for the scripts to be designed from ALM.

TOSCA- The scripts are designed and updated in TOSCA.

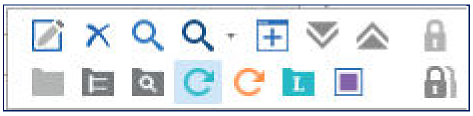
2. Template Test Case Designing

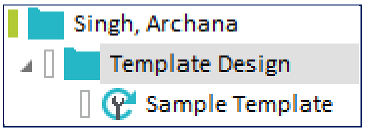
2.1 Why do we need Template Test case Designing

Template test case design comes into account with the data driven concept, when there is a chunk of test cases with similar functionality.

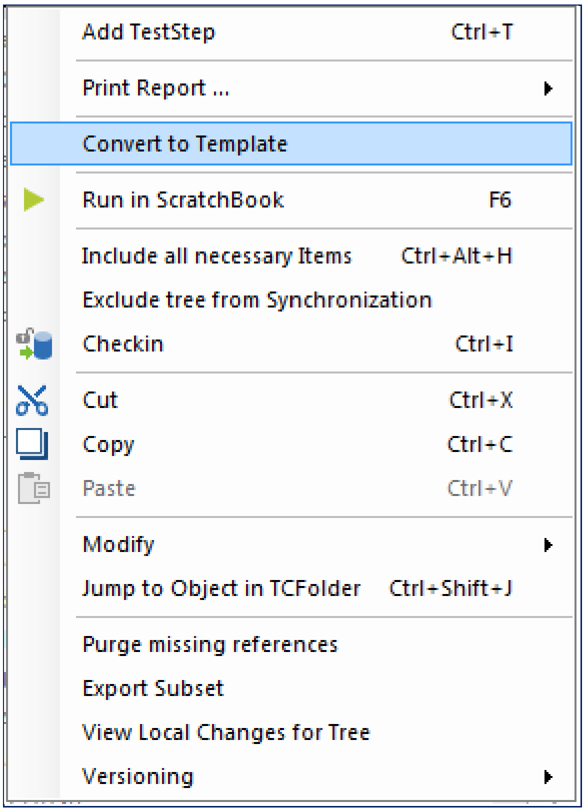
2.2 How to create Templates

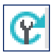
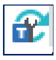
1. Right click the desired folder (it should be checked out) where you want to create your template, and create an empty test case and name it with a logical name.



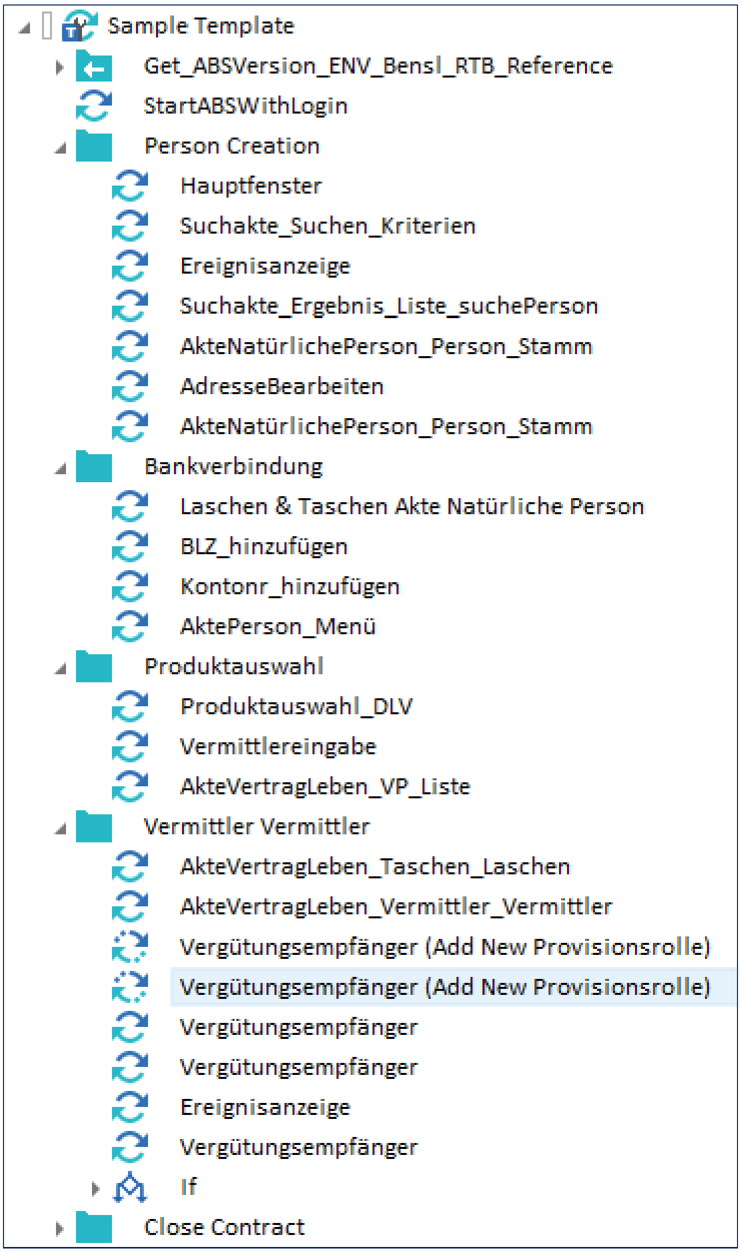


2. Now, design your whole test case with all the required test steps and then right click your designed test case and select ‘convert to template’ option.



Once you select ‘convert to template’ the sign of test case automatically gets changed from  to .

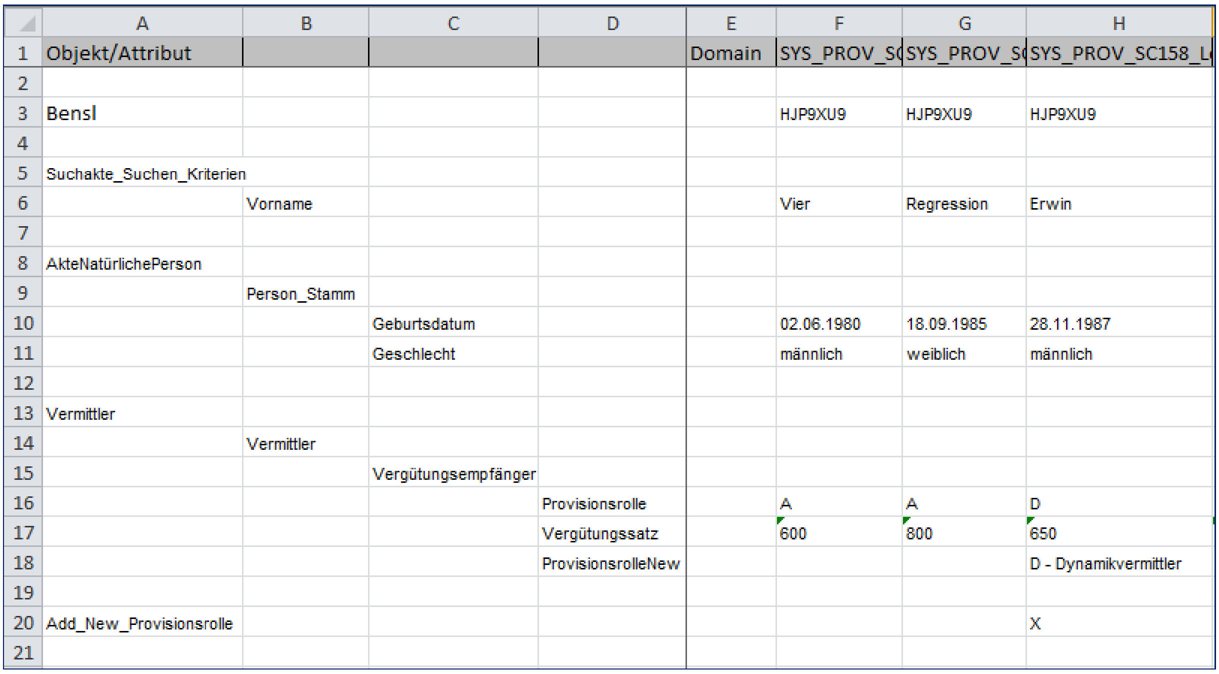
3. Now, as template test case designing uses the data driven concept hence we will add some extra test steps with different functionality in the template with some conditions. Concept of conditions is explained in further steps in this section.
   In the below template, we have added test steps which will be required in all test cases of a chunk of similar test cases with different “Provision Rolle”.



- Sample Template
  - Get_ABSVersion_ENV_Bensl_RTB_Reference
  - StartABSWithLogin
  - Person Creation
    - Hauptfenster
    - Suchakte_Suchen_Kriterien
    - Ereignisanzeige
    - Suchakte_Ergebnis_Liste_suchePerson
    - AkteNatürlichePerson_Person_Stamm
    - AdresseBearbeiten
    - AkteNatürlichePerson_Person_Stamm
  - Bankverbindung
    - Laschen & Taschen Akte Natürliche Person
    - BLZ_hinzufügen
    - Kontonr_hinzufügen
    - AktePerson_Menü
  - Produktauswahl
    - Produktauswahl_DLV
    - Vermittlereingabe
    - AkteVertragLeben_VP_Liste
  - Vermittler Vermittler
    - AkteVertragLeben_Taschen_Laschen
    - AkteVertragLeben_Vermittler_Vermittler
    - Vergütungsempfänger (Add New Provisionsrolle)
    - Vergütungsempfänger (Add New Provisionsrolle)
    - Vergütungsempfänger
    - Vergütungsempfänger
    - Ereignisanzeige
    - Vergütungsempfänger
    - If
  - Close Contract

4. Create a schema:
   After creation of your template, create a schema file from where you can fetch the test step values.
   - Schema file should always be in .xlsx or .xls format.
   - In order to increase the readability of a schema file, we put required input in hierarchal way.
   - In schema file, first 5 columns are reserved for attributes and from sixth column with first row represent the test case name.

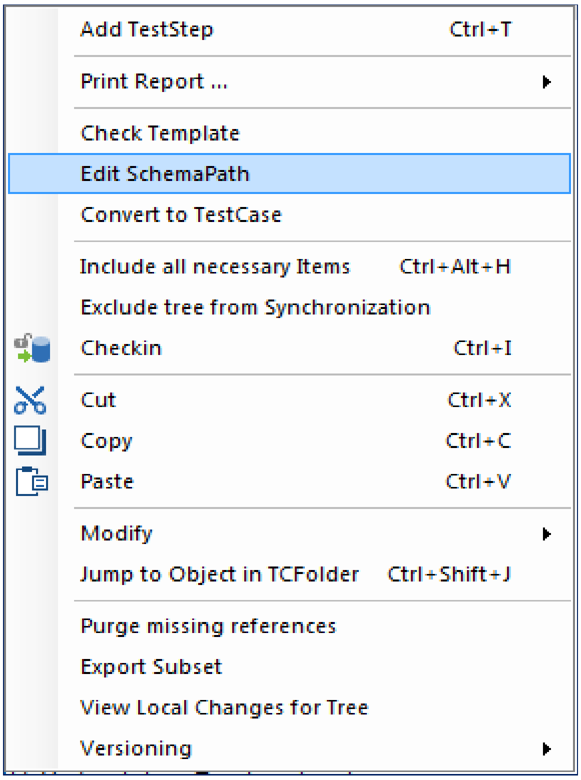
As shown in below screenshot, column A has the tasche name (Vermittler), column B has its corresponding Lasche (Vermittler), Column C has the attribute ‘Vergütungsempfänger’ present on this tasche/lasche and this hierarchy continues till column D.



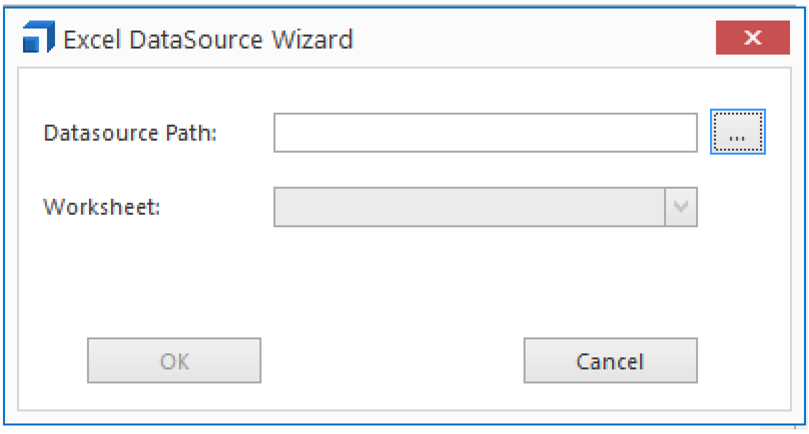
	A	B	C	D	E	F	G	H
1	Objekt/Attribut				Domain	SYS_PROV_S(	SYS_PROV_S(	SYS_PROV_SC158_L(
2								
3	Bensl					HJP9XU9	HJP9XU9	HJP9XU9
4								
5	Suchakte_Suchen_Kriterien							
6		Vorname				Vier	Regression	Erwin
7								
8	AkteNatürlichePerson							
9		Person_Stamm						
10			Geburtsdatum			02.06.1980	18.09.1985	28.11.1987
11			Geschlecht			männlich	weiblich	männlich
12								
13	Vermittler							
14		Vermittler						
15			Vergütungsempfänger					
16				Provisionsrolle		A	A	D
17				Vergütungssatz		600	800	650
18				ProvisionsrolleNew				D - Dynamikvermittler
19								
20	Add_New_Provisionsrolle							X
21								

5. How to link a schema File to Template:

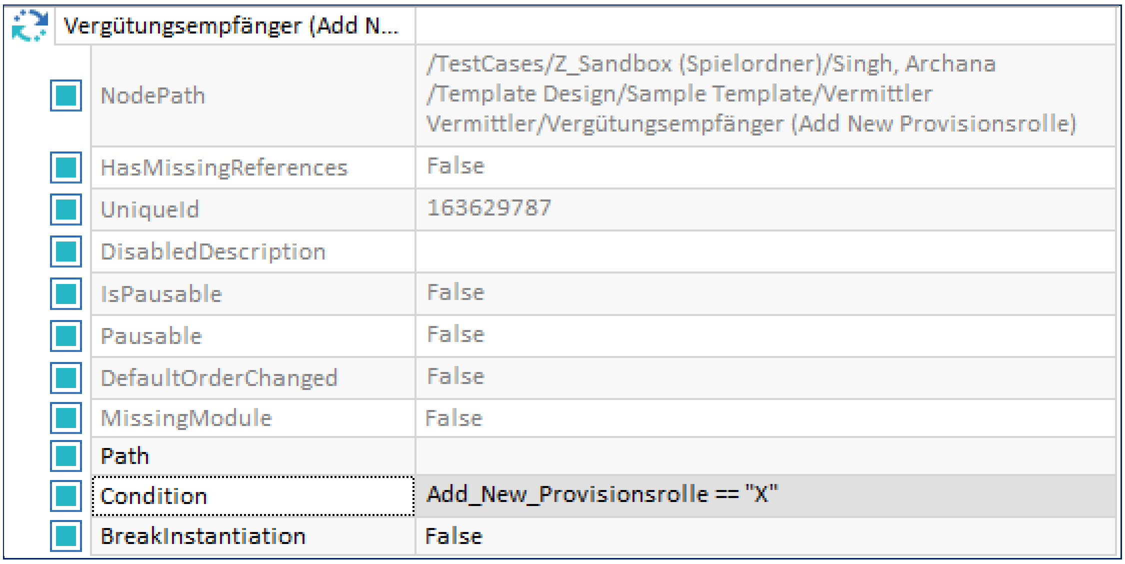
Once creation of schema file is done, we have to link the same with the test case. We can see the path of schema in attributes “SchemaPath” and “SchemaName” under Properties tab of the Template. To link a schema file, first right click the template and then select “Edit SchemaPath”.



A Popup 'Excel DataSource Wizard' opens. Now browse the schema path and related worksheet and click on OK button.



6. Once linked, we will start with applying conditions on the test steps where we want some changes or where the functionality is slightly different from other test steps.



Vergütungsempfänger (Add N...

NodePath	/TestCases/Z_Sandbox (Spielordner)/Singh, Archana /Template Design/Sample Template/Vermittler Vermittler/Vergütungsempfänger (Add New Provisionsrolle)
HasMissingReferences	False
UniqueId	163629787
DisabledDescription	
IsPausable	False
Pausable	False
DefaultOrderChanged	False
MissingModule	False
Path	
Condition	Add_New_Provisionsrolle == "X"
BreakInstantiation	False

In the above screenshot, First go to the properties tab of test step then put the condition (Add_New_Provisionsrolle == "X") under Condition field. We have different ways or methods to apply a condition. As in the above condition we used an attribute name with value 'X'. Here, it will look for exact value 'X' in the schema file with row name Add_New_Provisionsrolle.

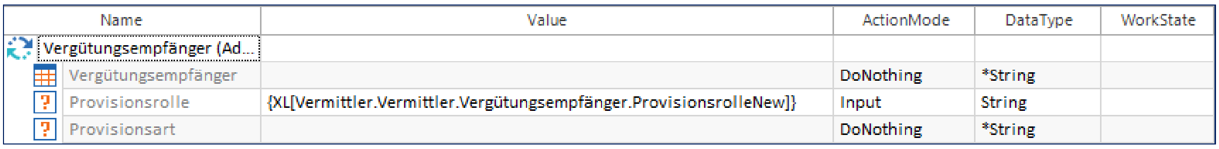
We can also use NULL datatype in the conditions. For example, if we use NULL for the same condition, we have to just put NULL in place of X. [Add_New_Provisionsrolle!=NULL]. Which means if the respective column contains NULL value then that particular test step will be ignored and Vice versa.

7. Now, go to the Detail tab of test step.

We follow a procedure to call a value from schema file.

For example: {XL[Vermittler.Vermittler.Vergütungsempfänger.ProvisionsrolleNew]}

In this we are using dot (.) operator until the final field called. Hence, first Vermittler is the Tasche, second Vermittler is Lasche, Vergütungsempfänger is one Field under tashe lasche and finally ProvisionsrolleNew is the sub value or whose value we actually want from the schema file.



Name	Value	ActionMode	DataType	WorkState
Vergütungsempfänger (Ad...				
Vergütungsempfänger		DoNothing	*String	
Provisionsrolle	{XL[Vermittler.Vermittler.Vergütungsempfänger.ProvisionsrolleNew]}	Input	String	
Provisionsart		DoNothing	*String	

8. Conditions
   In Tosca we generally use OR and AND operator for conditions. These operator are Boolean type which provides result in True or False. Hence one condition can be either true or false. We can use more than one condition using logical operators AND/OR.
   Example: Art==NULL OR Rolle==NULL
   Art==NULL AND Rolle==NULL
   We can also use the combination of OR and AND in our conditions.

3. Design Process:

1. Before 4th week of Month ‘M’, functional team highlights all the test cases to be automated in Pipeline (status is set to ’06. Reviewed’). By 4th week of month ‘M’, AFEs create the design plan for month ‘M’ + 1 for each module and then provide this design plan to onshore, offshore TM’s for review.

2. Onshore/Offshore TM’s approve/ disapprove the design plan accordingly.

3. IF the planning is rejected, then TM resend the resource planning to the AFE.

4. Once approved, status of manual test cases is set to ‘Automation Planned’ from ’06. Reviewed’ in ALM.

5. Once the status is set to ‘Automation Planned’ test designers start the designing of planned test cases. During the designing of test cases the status is changed to ‘Automation in progress’ from ‘Automation Planned’ in ALM. At the same time the status of ‘TestCaseWorkState’ is set to ‘In-work,’ keeping the Automation status blank in Tosca.

6. Once the designing of test cases is done (including Unit Testing), AFE’s perform the technical review in week1-week 4 of month ‘M’. The status of ‘TestCaseWorkState’ is set to ‘Completed,’ keeping the ‘Automation Status’ as ‘Automated-Unattended’ in Tosca. While the status of manual TC in ALM remains the same.

7. After the technical review of test cases, functional review is performed by the functional team. Once sent, status should be changed to ‘Handed over for functional review’ from ‘Automation in Progress’ in ALM. While the status remains the same (‘TestCaseWorkState’- Completed, ‘Automation status’- Automated-Unattended) in Tosca.

8. Once the functional review is done, Functional team move the test cases to production by Friday of every week of month ‘M’ and send the weekly progress note to AFE. While moving the test cases to production, status of all the manual test cases is changed to Automated and all the newly automated test cases are marked as ’06 Reviewed’ in ALM. These tasks are performed by functional team.

9. AFE send the weekly and monthly Consolidated Automation design to all the stakeholders.

Keep below points in mind:

- Folders in Tosca:
  - TestCase Design folder:

  [\TestCases\ABS_17.0\08 Testautomation\QA for automated TestCases\01_Design\Release 170\{Team Name}]
  - Execution Lists format:

  [\Execution\ExecutionLists\ALM\AzD_Standard\ABS_17.0\08 Testautomation\QA for automated TestCases\01_Design\Release 170\{Team Name}]
  - Once designing is completed, designer will perform unit testing and move the Test cases to below mentioned path :

  [\TestCases\ABS_17.0\08 Testautomation\QA for automated TestCases\02 Functional review folders \Release 170\{Team Name}]

- Once the functional review is complete (review comments incorporated), Test cases and Execution lists will be moved to production folders (03 / 04 folders) as per ALM structure, by functional teams.
- Only after the above steps are completed, functional teams will sync the test cases to ALM (sync the ExecutionList)

Note:

- Only one Execution list should exist for each Test case.
- Only one Execution entry must be present in each execution list.
- Execution lists should not be moved to ALM (synced) unless both TestCase and ExecutionList are in corresponding production folders in Tosca. If this rule is violated, Automated Script updater will not run on these test cases from Tosca, and corresponding changes done in Tosca will not be reflected in ALM.
- ExecutionList has to be synced at least once, else test cases will not be reflected in ALM.