

**Tosca Migration to AscentialTest** Matryxsoft Tech LLP 04 May 2023

#### **Background and Introduction:**

Tosca has been used by many customers across the world as one of the test automation tools for supporting various kinds of applications but lately the support for PowerBuilder is declining and a few customers of PowerBuilder, have made several requests to Zeenyx for migrating their tests from Tosca to AscentialTest as it supports all the latest controls of PowerBuilder applications.

Matryxsoft Tech, a Robust Automation Testers well known for migration and building reliable tests analyzed the requirement from the customers and also the need from IT market arena on test automation migration of Tosca. Being the distributors of AscentialTest (AT), came up with a plan and worked with Zeenyx Software, Inc (Product Owner) to migrate Tosca to AT. The purpose of this paper is to provide a detailed understanding on how the conversion was successfully done from Tosca to AscentialTest w.r.t to Object Repository, Steps/Functions, DataTable and Tests.

## Setup:

Ascential Test and Tricentis Tosca Tool licenses and Project built on Tosca are the basic requirement for Conversion.

The entire Tosca project including the modules, libraries, Configuration tests, plans and input data reading from external databases and etc. to be provided. All the files will be taken segregated according to the conversion required. One can go for the conversion with the utilities and the equivalent methods that are prepared for migration but for a better conversion rate it is good to understand the Tosca architecture and the framework built around for the application under test.

Application Knowledge and the manual flow testcase document will help in fine tuning the conversion at a faster rate.



# **AppObjects**

Tosca Objects are recognized by the path statements with multiple attributes like id, name, text, name, class and so on. At first all the modules having the screenshots containing the app objects are noted screen by screen and the same objects are captured by using AscentialTest Snapshot Technology creating app objects with unique paths will be identified. This task will be simple as both the product go with screenshots.

Example: AppObjects conversion is as shown in the AT Screenshot.

## **TOSCA Object Repository Screenshot**

ails	Self-Healing Properties				
	Name	lmage	TechnicalID	TypeInfoDescription	ValueRange
Cre	eate Bank Account Trans				
<b>a</b> ]	Currency				{Click};{Doublec
<b>a</b> ]	Amount				{Click};{Doublec
<b>a</b> ]	Payment Type				{Click};{Doublec
<u>a</u> ]	Initiating Company				{Click};{Doublec
<u>a</u> ]	Initiating Account				{Click};{Doublec
	From Company				
	From Account				
aŢ	To Company				{Click};{Doublec
aŢ	To Account				{Click};{Doublec
OK	Cancel				{Click}
OK	Discard				{Click}



## **AscentialTest Migrated Screenshot:**

```
App Objects.z ×
□ app WebPage Web_CreateBankAccountTransfer
  o path [@Title=="Create Bank Account Transfer for Settlement
 ■ WebElem Txt_Currency

    disabled path WebBlock[@ElemClass=="css-748hnh-SearchBarContainer edt0qdt23"][1]

   o path [@Label=="Currency"]
 o path [@Label=="Amount"]

    □ WebElem Txt PaymentType

    path [@Label=="Payment Type"]

    □ WebElem Txt_InitiatingCompany

   o path [@Label=="Initiating Company"]
 o path [@Label=="Initiating Account"]
 o path [@ElemClass=="WOTO"][1]
 o path [@ElemClass=="css-1flo3fb-PlainInstance evn577i3"][1]

    path [@ElemClass=="css-1flo3fb-PlainInstance evn577/3"][2]

 ■ WebElem Txt_ToCompany
path WebBlock[@ElemId=="pil-2501$35"]

    □ WebTextField Txt_ToAccount
    ○ path [@Label=="Account"]

 ■ WebPushButton Btn_Cancel
   path [@Text=="Cancel"]
 ■ WebPushButton Btn_Discard
   o path [@Text=="Discard"]
```

## Steps:

Tosca functions are captured in the form of modules. There could be 'n' number of modules with 'n' number of parameters. Each module is converted to AT steps by understanding Module flow into AscentialTest code by using the steps conversion utilities. With the help of these Utilities, have to convert Modules of Tosca to AT Steps.

A Sample of the module written in Tosca is migrated to AT as shown in the below pictures,



## Tosca Module,

Details	Test configuration (	Control Flow Diagram			
Name		Value	ActionMode	DataType	WorkState
△ 🕜 Provide data					PLANNED
▲ Create Bank Account Tr					
<b>a</b> ]	Currency	CLICK	Input	String	
aI	Currency	CP Currency	Input	String	
a	Currency	SENDKEYS {ENTER}	Input	String	
<b>a</b> [	Amount	CLICK	Input	String	
<b>a</b> [	Amount	CP Amount	Input	String	
<u>a</u> ]	Amount	SENDKEYS {ENTER}	Input	String	
<u>a</u> [	Payment Type	CLICK	Input	String	
<b>a</b> [	Payment Type	CP PaymentType	Input	String	
a	Payment Type	SENDKEYS {ENTER}	Input	String	
a	Initiating Company	CLICK	Input	String	
aI	Initiating Company	CP InitiatingCompany	Input	String	
<b>a</b> [	Initiating Company	SENDKEYS {ENTER}	Input	String	
a	Initiating Account	CLICK	Input	String	
<u>a</u> ]	Initiating Account	CP InitiatingAccount	Input	String	
<u>a</u> [	Initiating Account	SENDKEYS {ENTER}	Input	String	
	From Company	CLICK	Input	String	
	From Company	InnerText = CP InitiatingCompany	Verify	String	
	From Account	InnerText = CP InitiatingAccount	Verify	String	



### **AscentialTest Step**

```
Z Steps.z* x
□ class ProvideData_CreateBankAccountTransferPage : Step

    %StepInfo[Desc="Check able to provide data to create bank account transfer page.",Group="BankAccount_Transfer_For_S

    parameter String sCurrency

    parameter String sPaymentType

    parameter String sAmount

    parameter String sInitiatingCompany

    parameter String sInitiatingAccount

 ■ Main()
   //variable Declaration

    String sFromCompany,sFromAccount

    Web_CreateBankAccountTransfer.WaitUntilExists (3)

    Web_CreateBankAccountTransfer.Txt_Currency.Click ()

    Web_CreateBankAccountTransfer.Txt_Currency.TypeKeys (sCurrency)

    Web_CreateBankAccountTransfer.WaitWhileExists (3)

    Web_CreateBankAccountTransfer.Txt_Amount.Click ()

    Web_CreateBankAccountTransfer.Txt_Amount.DoubleClick()

    Web_CreateBankAccountTransfer.WaitWhileExists (3)

    Web_CreateBankAccountTransfer.Txt_PaymentType.Click ()

    Web_CreateBankAccountTransfer.Txt_PaymentType.TypeKeys (sPaymentType)

    Web_CreateBankAccountTransfer.WaitWhileExists (3)

    Web_CreateBankAccountTransfer.Txt_InitiatingCompany.Click ()

    Web_CreateBankAccountTransfer.Txt_InitiatingCompany.TypeKeys (sInitiatingCompany)

    Web_CreateBankAccountTransfer.WaitWhileExists (3)

    Web_CreateBankAccountTransfer.Txt_InitiatingAccount.Click ()

    Web_CreateBankAccountTransfer.Txt InitiatingAccount.TypeKeys (sInitiatingAccount)

    Web_CreateBankAccountTransfer.WaitWhileExists (3)

    sFromCompany = Web_CreateBankAccountTransfer.Txt_FromCompany.GetAttr ("Text")

    Print (sFromCompany)

    Web_CreateBankAccountTransfer.WaitWhileExists (3)

   sFromAccount = Web_CreateBankAccountTransfer.Txt_FromAccount.GetAttr ("Text")
   Print (sFromAccount)
```

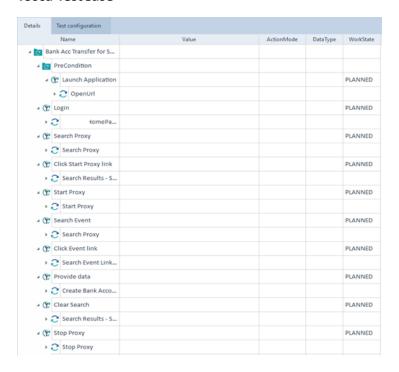
#### **Testcases to Tests:**

Tosca Tests are created by dragging the required modules and libraries in a sequential manner with the test data been passed either by parameterized or hard coded or taken from the test sheet. The flow of the test is clearly visible which helps us to drag the same steps/functions in AscentialTest to create the equivalent tests. By using standard TBox method, the precondition which helps to start the test and Postcondition which closes the tests. Where in AT concept of AppState that has OnStart and OnFinish which works similar way of Tosca Precondition and Postcondition.



Conversion of the tests is as shown in the below pictures.

#### Tosca TestCase



## **Tosca Precondition**

Details	Test configuration				
	Name	Value	ActionMode	DataType	WorkState
△ PreCondition					
4 (P)	aunch Application				PLANNED
4 €	OpenUrl				
	C Url	CP URL	Input	String	
	C UseActiveTab	True	Input	String	
•	WebDriverBrowser Arguments			String	



#### **AT Test Conversion**

```
2 Tests.z* ×
□ class BankAccount_Transfer_For_Settlement_Event_TS_01: Main_PreviewSignAppState
 %TestInfo[Table="Dt_BankAccountTransfer"]

    parameter String RowId

    Dt_BankAccountTransferRecord dt_BankAccountTransferRecord = DataStore.QueryTableRow ("Dt_BankAccountTransfer", "RowId", RowId)

   ♦ //Step:1

    //Check User is able to Login to application

    step LoginPage

    sUserName:dt_BankAccountTransferRecord.sUserName

   //To check user is able to enter data in search field

☐ step Search_Proxy Search_Proxy

    sSearchProxy:dt_BankAccountTransferRecord.lsSearchName[1]

    //Check able to click on Search item link

    step Click_SearchItemLink Click_StartProxyLink

   ♦ //Step:4

    //Check able to enter data in ActAs Field

□ step Enter_Data_ActAs_Field

    sActAs:dt_BankAccountTransferRecord.lsActAs[1]

   ♦ // Step:5
   //To check user is able to enter data in search field
   step Search_Proxy Search_Proxy

    sSearchProxy:dt_BankAccountTransferRecord.lsSearchName[3]

   ♦ // Step:6

    // Check User Able to click on event link page

   step Click_EventLink_Page
   ♦ // Step:7

\( // Check able to provide data to create bank account transfer page.
\( \)
```

## AppState:

```
Z Tests.z ×
□ class Main_:PreviewSignAppState : Test
            □ OnStart()

\( // Use Project/Data... to change the BrowserType \( \)
\( // Use Project/Data... to Change the BrowserType \( // Us
                         WebBrowserType BrowserType = DataStore.Query ("BrowserType")

    Web.SetBrowser(BrowserType)

☐ if Web_WorkDay.WaitUntilExists (20)

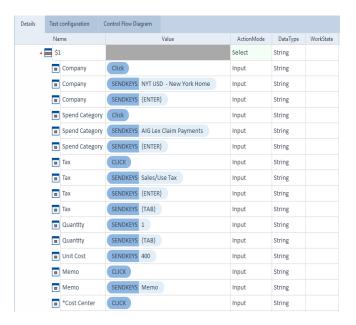
                                      return
                         Web.SetBaseState ("{DataStore.Query("StartURL")}")
           □ OnFinish()
                         ♦ Web.CloseAll()
```



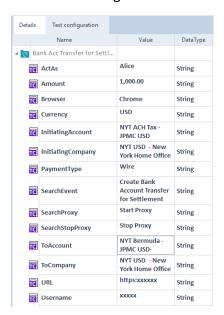
#### DataTable:

In Tosca, data to the Tests are passed directly by hardcoding or by parameterizing using a variable or by passing through the Test Sheet or an external database. These data are converted to AscentialTest Datatable.

#### Tosca Hardcoded Values:

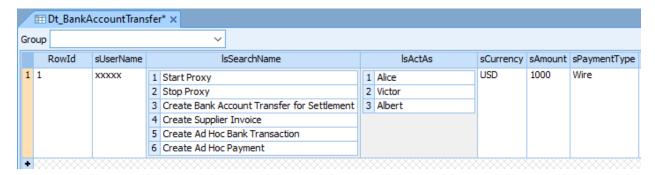


### Tosca Test Configuration:





#### AT DataTable:



### **Conclusion:**

Over years of effort built over Tosca project which is currently unstable/unreliable to maintain larger tests can be migrated quickly and efficiently to AscentialTest using the utilities created by Matryxsoft in a very short span of time which improves the reliability and quality of the tests in turn improves the software quality, the utmost priority for one and all.

For more information on migrating Tosca to AT test automation tool visit Matryxsoft Tech at www.matryxsoft.com.