

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:

AscentialTest 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

Details		Self-Healing Properties			
	Name	Image	TechnicalID	TypeInfoDescription	ValueRange
4	✚ Create Bank Account Trans...				
	 Currency				{Click};{Doublec...
	 Amount				{Click};{Doublec...
	 Payment Type				{Click};{Doublec...
	 Initiating Company				{Click};{Doublec...
	 Initiating Account				{Click};{Doublec...
	 From Company				
	 From Account				
	 To Company				{Click};{Doublec...
	 To Account				{Click};{Doublec...
	 Cancel				{Click}
	 Discard				{Click}

AscentialTest Migrated Screenshot:

```
App Objects.z x
└─ app WebPage Web_CreateBankAccountTransfer
  └─ path [@Title=="Create Bank Account Transfer for Settlement"]
    └─ WebElem Txt_Currency
      └─ disabled path WebBlock[@ElemClass=="css-748hnh-SearchBarContainer edt0qdt23"][1]
        └─ path [@Label=="Currency"]
      └─ WebTextField Txt_Amount
        └─ path [@Label=="Amount"]
      └─ WebElem Txt_PaymentType
        └─ path [@Label=="Payment Type"]
      └─ WebElem Txt_InitiatingCompany
        └─ path [@Label=="Initiating Company"]
      └─ WebTextField Txt_InitiatingAccount
        └─ path [@Label=="Initiating Account"]
      └─ WebText FromAccount
        └─ path [@ElemClass=="WOTO"][1]
      └─ WebText Txt_FromCompany
        └─ path [@ElemClass=="css-1flo3fb-PlainInstance evn577i3"][1]
      └─ WebText Txt_FromAccount
        └─ path [@ElemClass=="css-1flo3fb-PlainInstance evn577i3"][2]
      └─ WebElem Txt_ToCompany
        └─ path WebBlock[@ElemId=="p#-2501$35"]
      └─ WebTextField Txt_ToAccount
        └─ path [@Label=="Account"]
      └─ WebPushButton Btn_Cancel
        └─ path [@Text=="Cancel"]
      └─ WebPushButton Btn_Discard
        └─ 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
4	Provide data					PLANNED
4	Create Bank Account Tr...					
	Currency	CLICK		Input	String	
	Currency	CP Currency		Input	String	
	Currency	SENDKEYS {ENTER}		Input	String	
	Amount	CLICK		Input	String	
	Amount	CP Amount		Input	String	
	Amount	SENDKEYS {ENTER}		Input	String	
	Payment Type	CLICK		Input	String	
	Payment Type	CP PaymentType		Input	String	
	Payment Type	SENDKEYS {ENTER}		Input	String	
	Initiating Company	CLICK		Input	String	
	Initiating Company	CP InitiatingCompany		Input	String	
	Initiating Company	SENDKEYS {ENTER}		Input	String	
	Initiating Account	CLICK		Input	String	
	Initiating Account	CP InitiatingAccount		Input	String	
	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

```

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

Details		Test configuration			
	Name	Value	ActionMode	Data Type	WorkState
└	Bank Acc Transfer for S...				
└	PreCondition				
└	Launch Application				PLANNED
└	OpenUrl				
└	Login				PLANNED
└	homePa...				
└	Search Proxy				PLANNED
└	Search Proxy				
└	Click Start Proxy link				PLANNED
└	Search Results - S...				
└	Start Proxy				PLANNED
└	Start Proxy				
└	Search Event				PLANNED
└	Search Proxy				
└	Click Event link				PLANNED
└	Search Event Link...				
└	Provide data				PLANNED
└	Create Bank Acco...				
└	Clear Search				PLANNED
└	Search Results - S...				
└	Stop Proxy				PLANNED
└	Stop Proxy				

Tosca Precondition

Details		Test configuration			
	Name	Value	ActionMode	Data Type	WorkState
└	PreCondition				
└	Launch Application				PLANNED
└	OpenUrl				
└	Url	CP URL	Input	String	
└	UseActiveTab	True	Input	String	
└	WebDriverBrowser Arguments			String	

AT Test Conversion

```

Tests.z* x
class BankAccount_Transfer_For_Settlement_Event_TS_01 : Main_PreviewSignAppState
  %TestInfo[Table="Dt_BankAccountTransfer"]
  parameter String RowId
  Main()
    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
    //Step:2
    //To check user is able to enter data in search field
    step Search_Proxy Search_Proxy
      sSearchProxy:dt_BankAccountTransferRecord.lsSearchName[1]
    //Step:3
    //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:

```

Tests.z x
class Main_PreviewSignAppState : Test
  OnStart()
    // Use Project/Data... to change the BrowserType
    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:

Details					
Test configuration					
Control Flow Diagram					
Name	Value	ActionMode	Data Type	WorkState	
\$1		Select	String		
Company	Click	Input	String		
Company	SENDKEYS NYT USD - New York Home	Input	String		
Company	SENDKEYS {ENTER}	Input	String		
Spend Category	Click	Input	String		
Spend Category	SENDKEYS AIG Lex Claim Payments	Input	String		
Spend Category	SENDKEYS {ENTER}	Input	String		
Tax	CLICK	Input	String		
Tax	SENDKEYS Sales/Use Tax	Input	String		
Tax	SENDKEYS {ENTER}	Input	String		
Tax	SENDKEYS {TAB}	Input	String		
Quantity	SENDKEYS 1	Input	String		
Quantity	SENDKEYS {TAB}	Input	String		
Unit Cost	SENDKEYS 400	Input	String		
Memo	CLICK	Input	String		
Memo	SENDKEYS Memo	Input	String		
*Cost Center	CLICK	Input	String		

Tosca Test Configuration:

Details			
Test configuration			
Name	Value	Data Type	
Bank Acc Transfer for Sett...			
ActAs	Alice	String	
Amount	1,000.00	String	
Browser	Chrome	String	
Currency	USD	String	
InitiatingAccount	NYT ACH Tax - JPMC USD	String	
InitiatingCompany	NYT USD - New York Home Office	String	
PaymentType	Wire	String	
SearchEvent	Create Bank Account Transfer for Settlement	String	
SearchProxy	Start Proxy	String	
SearchStopProxy	Stop Proxy	String	
ToAccount	NYT Bermuda - JPMC USD-	String	
ToCompany	NYT USD - New York Home Office	String	
URL	https:xxxxxx	String	
Username	xxxxxx	String	

AT DataTable:

Dt_BankAccountTransfer* x							
Group							
RowId	sUserName	IsSearchName	IsActAs	sCurrency	sAmount	sPaymentType	
1	1	xxxxx	1 Start Proxy 2 Stop Proxy 3 Create Bank Account Transfer for Settlement 4 Create Supplier Invoice 5 Create Ad Hoc Bank Transaction 6 Create Ad Hoc Payment	1 Alice 2 Victor 3 Albert	USD	1000	Wire

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.