Test complete tool is a commercial tool by smart bear .It used for web applications and windows applications. We have project items such as Tested Apps, stores, Name mapping, keyword tests.

**Version of the Test Complete Tool:**

The latest version of Test Complete is 12.50

**Tested Apps:**

It means we are adding a tested application to this project item. Tested application might be Browser application or Windows application.

If it is browser application we can add like this in test script:

1.TestedApps. https\_\_\_www\_facebook\_com\_.Run ()

We can do cross browser testing in the test complete by selecting browser option.

If it is windows application we can add like this in test script:

1. TestedApps. calc.Run()

**Stores:**

Stores is a list of items that are stored in Test complete project for future comparison and for creating check points.

We can store the following :

1. Image Files (jpeg, png, gif)

2. General files

3. Data base tables.

4. XML files

**Name mapping:**

Name mapping contains all objects related to applications which are used in the automated tests.

**How to do?**

**Keyword Tests:**

Keyword driven tests are called keyword tests. Each keyword is equal to one action.

**Automating the Calculator application:**

1. To automate the Calculator application, first we select the calculator as Tested Application.
2. We select the calc.exe file in the application as Run mode parameters.

2. We use the object spy for choosing the buttons in the Calculator and Click() method.

**Automating the Toolsqa Application:**

1. We can choose the browser options in the Run mode parameters.

a) We can choose the URL in Run mode parameters as [**http://toolsqa.com/automation-practice-form/**](http://toolsqa.com/automation-practice-form/)

b) Choose the web browser as internet explorer.

1.To enter some words in the Edit box we use “SetText()” method

EX: Sys.process().SetText(“Hello Vidyanath”)

2. To choose file option also we use “Click()” method

3.For dropdown options we use “Keys()” method

EX: Sys.process().Keys(“Europe”)

4.In selection options we use “ClickItem()” method.

Ex: Sys.process().ClickItem(1).Where 1 is an index

5.To rotate the mouse wheel we are using the “MouseWheel()”

We use –ve value to rotate the mouse wheel in down direction.

We use +ve value to rotate the mouse wheel in up direction.

Ex: Sys.process().MouseWheel(-50,0)-For down

**We can automate the things in 3 ways.**

1. Descriptive way

2. Record and Play

I did this

3. Object Spy

**1. Descriptive way:**

If object is changing dynamically then to overcome this problem we use descriptive way of testing.

a.Find

b.FindAll

c.FindChild

**a. Find:**

Find method is used to search the desired object in the object hierarchy. This method is used to search the object with specified values and specified properties.

Testobj.Find (property names, property values, depth, refresh)

**b.FindAll:**

Findall method used to search the object in the object hierarchy.i.e in the child and parent objects.

Search the object with specified values and specified properties.

Testobj.FindAll (property names, property values, depth, refresh)

**c.FindChild:**

Find child method is used to search the object in the child objects.

Search the object with specified values and specified properties.

Testobj.FindChild (property names, property values, depth, refresh)

**Data Driven Testing:**

def CurDriverExample():

# Creates a driver

DDT.ExcelDriver ("C:\\MyFiles\\MyFile.xls", "Sheet1")

# Iterates through records while not

DDT.CurrentDriver.EOF ():

# Gets a value from the storage and posts it to the log

Log.Message(DDT.CurrentDriver.Value[0])

DDT.CurrentDriver.Next()

# closes the driver

DDT.CloseDriver(DDT.CurrentDriver.Name)