Automating UFT Test from Azure DevOps

Prerequisite:

1) A Virtual Machine with a UFT One Testing Automation Software Installed.
2) An Azure DevOps account
3) Test Scripts for Testing

Step 1: Creating of a Self-Hosted Agent

1) Go to Azure DevOps and create a Personal Access Token. Save the Token at Safe Place.
2) Go to your Project Settings and to Agent pool section-> Default pool.
3) Click on New Agent and download the agent.
4) Copy the Agent to UFT One VM, and keep it under download folder
5) Execute the following scripts from Powershell on "c:\>"

```
PS C:\> mkdir agent ; cd agent
PS C:\agent> Add-Type -AssemblyName System.IO.Compression.FileSystem ;
[System.IO.Compression.ZipFile]::ExtractToDirectory("$HOME\Downloads\vst
s-agent-win-x64-2.170.1.zip", "$PWD")
```

6) Run below command

```
PS C:\agent> .\config.cmd
```

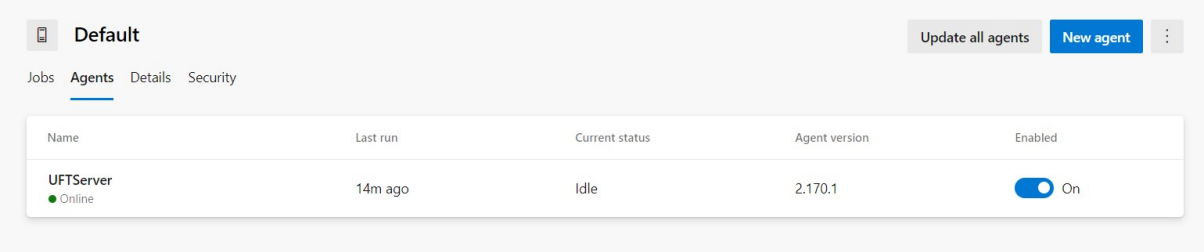
7) Enter Server URL - https://dev.azure.com/<Your DevOps Organisation Name >

8) Press enter for PAT
9) Enter your Personal Access token which you saved on step 1
10) Enter the agent pool – Default
11) Press enter for NT Authority access
12) Run below command, and make sure it keeps running .This is the command which keeps machine agent active

```
PS C:\agent> .\run.cmd
```

12) Your Agent Machine is ready now

In Azure DevOps Agent pool you will see the Agent will Show Online



Step 2: Create a BAT file and VSB Script file for UFt Automation inside your UFT Test Solution.

1) In VS Code inside UFT Test Root folder create a “mytest.bat” file and copy and paste following script.(You can use any name for the file)

```
cd c:\windows\syswow64

cscript.exe "myvbs.vbs"
exit
```

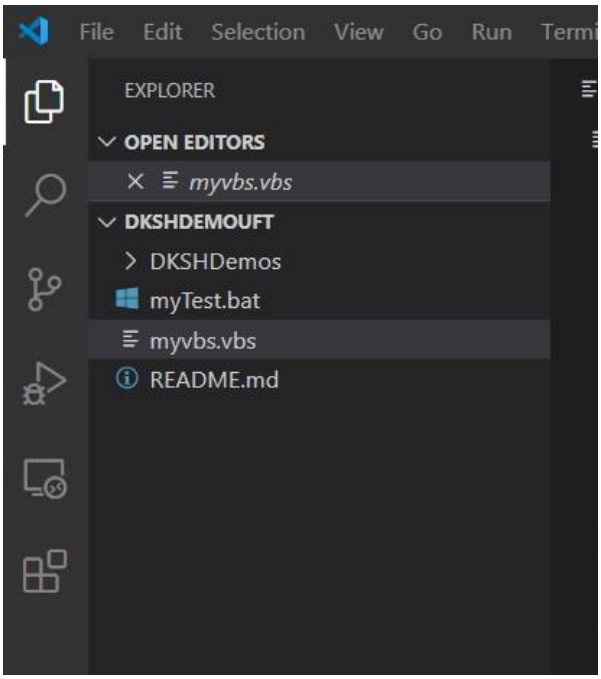
2) In Same root folder , create myvbs.vbs file and paste following scripts. (in Test Path -> Put the Folder name of your Test which is in root Directory under same tree as this file)

```
testPath = "DKSHDemos"
```

```
Dim objFSO
Set objFSO = CreateObject("Scripting.FileSystemObject")
DoesFolderExist = objFSO.FolderExists(testPath)
Set objFSO = Nothing
If DoesFolderExist Then
Dim qtApp
Dim qtTest
Dim qtResultsOpt
Set qtApp = CreateObject("QuickTest.Application")
qtApp.Launch
qtApp.Visible = True
qtApp.Open testPath, False
Set qtTest = qtApp.Test
Set qtResultsOpt = CreateObject("QuickTest.RunResultsOptions")
qtResultsOpt.ResultsLocation = "C:\Users\dkshuser\Desktop\Reports"
qtTest.Run qtResultsOpt,True
qtTest.Run
qtTest.Close
qtApp.Quit
Else
End If
```

For the **`qtResultsOpt.ResultsLocation`** , Create that folder inside the VM agent and copy that path for this variable. This is where your Test Report will be stored.

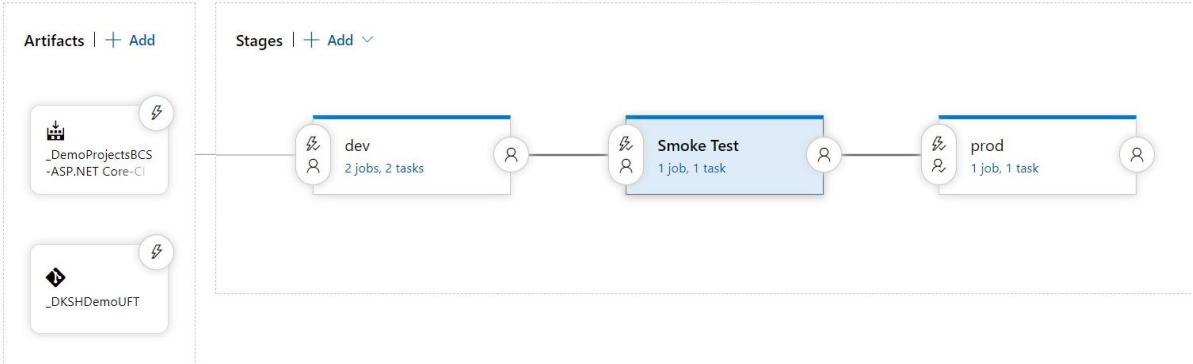
3) Your Tree Structure should look Like this :-



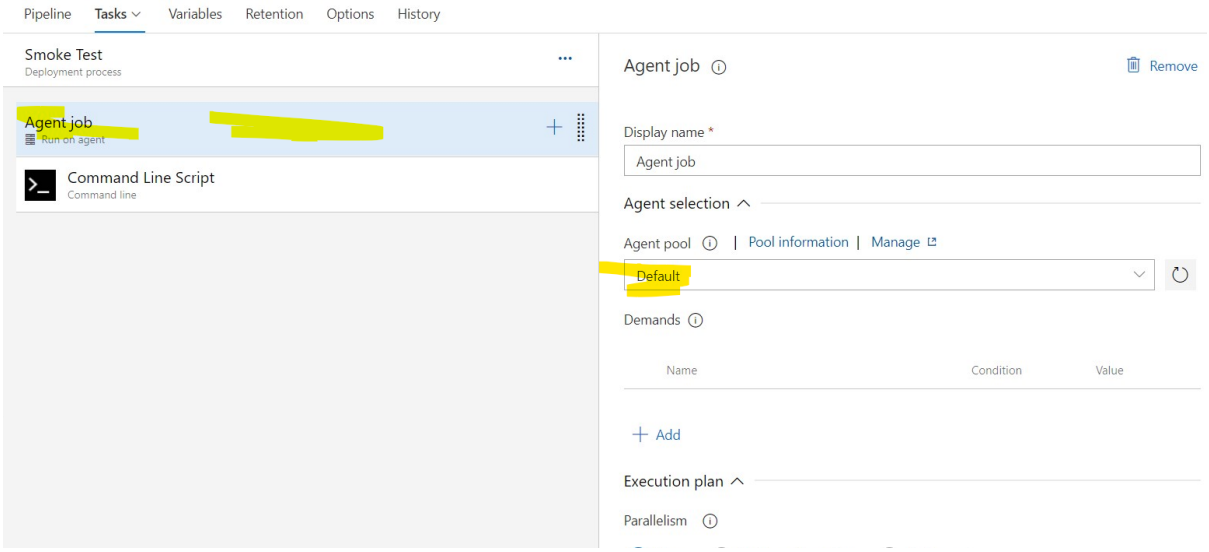
4) Checkin and push your code to a Repository in Azure Repo

Step 3: Create a Stage/Pipeline for executing the test scripts .

1) Go to the Pipeline of application for which you want to run this test.
2) Add the UFT Test artifact, by selecting Azure Repo-> UFT Test Repo -> master branch -> latest . Save the artifact
3) Add a Stage after the dev stage named as Smoke(or any test) test you have the script of.



4) Click on Smoke test stage on Job link in blue
5) Select the Agent Job, and Choose Agent Pool as “default”, to make your UFT Virtual machine as agent machine to run this test.



6) Now click on + and add a command line task

7) In CMD Task, you have to execute the VB Script which you have pushed in your Repository which will be there in Agent Machine after deploy.
8) Add the path

   $(Agent.WorkFolder)\r1\a\<Your Test Artifact> Name\myvbs.vbs

   In my case the Artifact name is _DKSHDemoUFT, So my Script command will be

   $(Agent.WorkFolder)\r1\a_DKSHDemoUFT\myvbs.vbs

   Remember there can be different value instead of r1, that can only check once you run the pipeline, you can go to Agent Machine **C:\agent_work** folder and see the artifact name where your TEST artifact is stored, Usually it is r1 if there is 1 pipeline, r2 if there is 2nd pipeline …likewise

9) Run the Pipeline and watch the VM, you will see the UFT test will run successfully.

10) Browse to the `C:\Users\dkshuser\Desktop\Reports` location, which you put to have the report and check the Html file for the Report. You can decide how you want to show this report. This folder can be pushed to a public location to be accessible by any team for test report, or you can make it as a website to show the report or you can decide to create a mail task to send this report via mail to target users.

I have hosted this in IIS and, and can see report as a webpage:-