ZUG

Day3

automature

Topics

- MVMs & MVCVs
- Debugger
- Built-in Variables
- Built-in Atoms
- Step Properties

automature

MVM & MVCV

- **Limitations of current technologies:**
  - Data driven testing is complex
- **Purpose:**
  - Allows user to test with multiple input data sets
  - Increases the coverage of a test scenario by automatically exercising all combinatorial variants

automature

Multi-valued Macros (MVM)

- Multi-value macros (MVM) contain lists of values

SYNTAX:

$$TestType={manual, automation}

- Primarily used as *iteration vehicles*
- Can be used inside a test case or a molecules

automature

MVM in Test Case

The same test case can test with multiple search queries

TestCase ID	Description	property	User	Step	Action	ActionArg_1	ActionArg_2
Test007					SetContextVar	Handle	
					&SearchGoogle	Handle	$$QUERY

Molecule ID	Description	Property	User	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
SearchGoogle					@OpenBrowserWithUrl.BAT	$GOOGLE_URL	$input_arg1	
					@SetTextByName.BAT	%input_arg1%	$GOOGLE_SEARCH_FIELD_NAME	$input_arg2
					@ClickButtonByName.BAT	%input_arg1%	$GOOGLE_SEARCH_BUTTON_NAME	

automature

MVM in Molecule

Test Case

TestCase ID	Description	property	User	Step	Action	ActionArg_1	ActionArg_2
Test008					SetContextVar	Handle	
					&SearchGoogle	Handle	$QUERY
					UnSetContextVar	Handle	

Molecule

Molecule ID	Description	Property	User	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
SearchGoogle					@OpenBrowserWithUrl.BAT	$GOOGLE_URL	$input_arg1	
					@SetTextByName.BAT	%input_arg1%	$GOOGLE_SEARCH_FIELD_NAME	$input_arg2
					@ClickButtonByName.BAT	%input_arg1%	$$GOOGLE_SEARCH_BUTTON_NAMES	

automature

MVM Features

- Values can be comma-separated lists
  - e.g. {QA1,QA2,QA3}
- Can also be a contiguous range of values
  - e.g. {1..5}, same as {1,2,3,4,5}
- Test Case identifications are auto-generated
  - e.g. Test001_1, Test001_2 and so on.
- An MVM can be passed as a scalar, and interpreted within the molecule as a vector
  - e.g. $HOSTS = {OHIO, IOWA, MAINE}
  - &EXEC_SHELL $HOST, $COMMAND
  - EXEC_SHELL:
    - @DO_COMMAND ##HOST, #COMMAND

automature

Exercise : MVM

- Write a test case to search in Google with different queries in a single testcase, using an MVM

automature

Cartesian MVM

- Cartesian product over the set of values of all the MVM

Ex

$FILE_NAME={File1.txt, File2.rtf}
$LOCATION={C:\TempFolder, \\Network\SharedFolder}

referencing both these MVMs in a single testcase as
$$ FILE_NAME, and $$LOCATION will generate 4 test case

1. [File1.txt, C:\TempFolder],
2. [File2.rtf, C:\TempFolder],
3. [File1.txt, \\Network\SharedFolder],
4. [File2.txt, \\Network\SharedFolder]

automature

Cartesian MVM

- **Search all the combinations of Source to Destinations**

TestCase ID	Description	propert	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4	ActionAr	ActionArg_6
Init	Creating a variable named Handle			SetConte>	Handle					
	Open a browser			Zbrowser	Handle	$Browser				
TC004	Selecting a Way using Cartesian MVM			&Travel	BrowserHandle=%Handle%	Origin=$$Source_	Dest=$$Destination_	WaitTime=30	Url=$URL	Page_Title=$Result_Title

Macros declared for Cartesian MVM

automature

Indexed MVM

- Performs indexing of two MVM.
- Both MVMs must have same number of elements.
- Ex:
  - $FILE_NAME={File1.txt, File2.rtf}
  - $LOCATION={C:\TempFolder, \\Network\SharedFolder}
  - Index both the MVM
    - $FILE_NAME_LOCATION={$$FILE_NAME, $$LOCATION}

automature

Indexed MVM

- Using index MVM
  - $$FILE_NAME_LOCATION#FILE_NAME and
  - $$FILE_NAME_LOCATION#LOCATION expand to
    - [File1.txt, C:\TempFolder]
    - [File2.txt, \\Network\SharedFolder]

	TestCase ID	Description	propert	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4	ActionAr	ActionArg_6
2											
3	Init	Creating a variable named Handle			SetConte	Handle					
4		Open a browser			Zbrowser	Handle	$Browser				
5											
6	TC005	Selecting a Way using Indexed MVM			&Travel	BrowserHandle=%Handle%	Origin=$$Indexed#Source_Cities	Dest=$$Indexed#Dest_Cities	WaitTime=30	Url=$URL	Page_Title=$Result_Title
7											

Declaring Marcros for Indexed MVM

automature

Indexed MVM: When to use?

- Typical use case for Indexed MVM is when you want to iterate over multiple user credentials
  - e.g. a user has multiple attributes, viz.
    - Username
    - Password
    - Role
- Here is how you would declare this:
  - $USERNAMES = {tom, dick, harry}
  - $PASSWORDS = {txxx, dyyy, hzzz}
  - $ROLES = {ADMIN, VENDOR, CUSTOMER}
  - $USER = {$$USERNAMES,$$PASSWORDS,$$ROLES}
- Here is how you would reference any attribute:
  - $$USER#USERNAMES

automature

Multi-valued Context Variable

- **Limitation of MVM**
  - Iterates on predefined values
  - Cannot change the values at run time
- **MVCV**
  - **Can change the value at run time**
  - **Can generate a MVCV at run time.**

automature

Multi-valued Context Variable

- **Context Variable contains a list of values**

Action	ActionArg_1
SetContextVar	mvcv1={value1,value2}
SetContextVar	mvcv2={valueA,valueB}

- **Test Case/Molecule executed for each value**

TestCase ID	Description	property	Step	Action	ActionArg_1
comment	Printing the values of a context variable as mvcv				
MVCV004				print	$$%mvcv1%

```
TestCase ID      Status          Time Taken(In milli-seconds)    Comments

MVCV004_value1   pass            25
MVCV004_value2   pass            23
**********************************************************************************
```

automature

Multi-valued Context Variables

- **Used in Molecule**
  - **Passing by Value**

TestCase ID	Description	property	Step	Action	ActionArg_1	ActionArg_2
comment	cartesian product in a molecule using named argument					
MVCV006				SetContextVar	StringCartesian	
				&MoleculeMVCV	list1=%mvcv1%	list2=%mvcv2%

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
MoleculeMVCV				#define_arg	list1	list2	
				AppendToContextVar	contextvar=StringCartesian	v1=##list1	v2=##list2

automature

Multi-valued Context Variables

- **Used in Molecule**
  - **Passing by Reference**

TestCase ID	Description	property	Step	Action	ActionArg_1	ActionArg_2
comment	cartesian product in a molecule using named argument when passed as reference					
MVCV007				SetContextVar	StringCartesian	
				&MoleculeMVCV2	list1=**mvcv1**	list2=**mvcv2**

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
MoleculeMVCV2				#define_arg	list1	list2	
				AppendToContextVar	StringCartesian	**$$%#list1%**	**$$%#list2%**

automature