

Macros

Macros

- Shorter Name given to Constants
- Macro name starts with a \$
- Macro name can contain letters, numbers or underscores (_)
- Values are substituted at runtime.

| | A | B | C |
|---|-------------------------------------|---|---|
| 1 | Macro Name | Value | Comment |
| 2 | \$URL | http://google.com | Giving the URL input which the browser will open. |
| 3 | \$Browser | firefox | Specifying which browser to open. |
| 4 | \$Login_Verify_Text | Google | Specifying the text which should be verified in the webpage . |

| TestCase ID | Description | property | Step | Action | ActionArg_1 | ActionArg_2 | ActionArg_3 | Verify | VerifyArg_1 |
|-------------|-------------|----------|------|------------------------------|-------------|-------------|-------------|--------------------------------|-------------------------------------|
| TC001 | | | | &OpenUrl | Handle | \$Browser | \$URL | &VerifyUrl | \$Login_Verify_Text |

Macros: Where can you declare them?

- On the Macro worksheet of the testsuite
 - Referenced directly
 - e.g. \$URL
- On the macro worksheet of an external file
 - Referenced through the name of the external file
 - e.g. \$extfile.URL

Macros: Environment specific values

- Macro values can be environment specific
 - e.g. \$DBUserName can be
 - user_qa1 on QA1
 - user_qa2 on QA2
 - user_qa3 on QA3
- Values can be given in separate columns
- Values that are not environment specific, only a single value need be specified
- Choice of column to use is made at runtime
 - -macrocolumn=file identifier:column number

Macros: What they cannot do

- **Limitations of Macros:**

- Cannot change the values of Macros at run time
- Atoms cannot alter the values of Macros

Context Variables

Context Variables

- **Purpose**

- Can change the value of at runtime
- Atoms can access and modify the Contents

ContextVars: Declarations

- Set/UnSet ContextVars
- Passing ContextVars by reference and by value

By Reference

| | A | B | C | D | E | F | G | H | I | J | K |
|---|-------------|-------------|----------|------|------|-------------------------|-------------|----------------------|--------------|--------------------|---------------------|
| 1 | TestCase ID | Description | property | User | Step | Action | ActionArg_1 | ActionArg_2 | ActionArg_3 | Verify | VerifyArg_1 |
| 2 | | | | | | | | | | | |
| 3 | Login | | | | | SetContextVar | handle | | | | |
| 4 | | | | | | @OpenBrowserWithUrl.BAT | \$URL | handle | | @VerifyContent.BAT | \$LOGIN_TEXT |
| 5 | | | | | | @SetTextByName.BAT | %handle% | \$USERNAME_FIELDNAME | \$MYUSERNAME | | |
| 6 | | | | | | @SetTextByName.BAT | %handle% | \$PASSWORD_FIELDNAME | \$MYPASSWORD | | |
| 7 | | | | | | @ClickButtonByName.BAT | %handle% | \$LOGIN_BUTTON_NAME | | @VerifyContent.BAT | \$WELCOME_PAGE_TEXT |
| 8 | | | | | | | | | | | |

By Value

ContextVariables and Atoms

- Reading And Altering Context Variable values from Atoms (ZugAPI.DLL)
 - Reading the value:
 - GetContextVar: `value = GetContextVar(name)`
 - Modifying the value:
 - AlterContextVar: `void AlterContextVar(name, value)`
 - Writing to the logfile from an atom
 - Log: `void Log("loglevel", "message")`

ContextVars: How are they special?

- Provide easy mechanism to share information between testsuite and atoms implemented in any language
- Stored in a SQLite Database
- Associated with Zug instance creating them
- Scope is global within a testsuite

Exercise : ContextVars

- Write a test case to write a macro value to file and then read the content of the file and store it in a ContextVar, and verify the content of the file.

AppendToContextVar

- Built-in atom
- Concatenates multiple values
- Any number of values can be appended

Example : AppendToContextVar

- Creating a URL by appending the domain name and the page name.

| TestCase ID | Description | property | User | Step | Action | ActionArg_1 | ActionArg_2 | ActionArg_3 | ActionArg_4 |
|-------------|-------------|----------|------|------|-------------------------|----------------|----------------------|--------------|------------------|
| | | | | | | | | | |
| Login | | | | | SetContextVar | Handle | | | |
| | | | | | SetContextVar | URL=http:// | | | |
| | | | | | AppendToContextVar | contextvar=URL | domain=\$DOMAIN | slash=/' | page=\$PAGE_NAME |
| | | | | → | @OpenBrowserWithUrl.BAT | %URL% | Handle | | |
| | | | | | @SetTextByName.BAT | %Handle% | \$USERNAME_FIELDNAME | \$MYUSERNAME | |
| | | | | | @SetTextByName.BAT | %Handle% | \$PASSWORD_FIELDNAME | \$MYPASSWORD | |
| | | | | | @ClickButtonByName.BAT | %Handle% | \$LOGIN_BUTTON_NAME | | |
| | | | | | UnSetContextVar | Handle | | | |
| | | | | | UnSetContextVar | URL | | | |
| | | | | | | | | | |

Exercise : AppendToContextVar

- Write a test case to create refined search query by appending values to it.
 - For example if the first query is "test" then the second query can be "test automation"