

ZUG Training

Day 2

Topics

- **Molecules**
- **Atoms**
- **Macros & Context Variables**
- **Debugging**

CHEMICAL PARADIGM

- **Molecules**
- **Atoms**

What are molecules?

- **A group of test steps carrying out a logical piece of work, which may be either actions, verifications or both**
- **A molecule has a name, and can be invoked from another molecule or a testcase**

Why do we need molecules?

- **Limitations of Test cases**

- Difficult to reuse the test case logic
- Difficult to maintain
- Testcases become unreadable

- **Purpose**

- Reusability of the test case logic

Molecules are

- Collection of several steps
- Has a name it can be referenced by
- Can be invoked by a test case, or another molecule
- Accept input arguments, and can also pass data back to the caller
- May invoke another molecule, an external, or a "built-in" atom

Example : Invoking a Molecule

Molecules are invoked by placing an "&" before the name

TestCase ID	Description	property	User	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4
Test005					SetContextVar	Handle			
					&Login	Handle	\$URL_YAHOO	\$ZERMATT_USER	\$ZERMATT_PASSWD

Molecule ID	Description	Property	User	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
Login					@OpenBrowserWithUrl.BAT	\$input_arg2	\$input_arg1	
					@SetTextByName.BAT	%input_arg1%	\$YAHOO_USER_FIELDNAME	\$input_arg3
					@SetTextByName.BAT	%input_arg1%	\$YAHOO_PASSWD_FIELDNAME	\$input_arg4
					@ClickButtonByName.BAT	%input_arg1%	\$YAHOO_LOGIN_BUTTON_NAME	

Molecule

- **Argument Passing**

- Positional argument passing
- Named argument passing

Molecule: Passing positional arguments

Positional argument Passing

TestCase ID	Description	property	Step	Action	ActionArg_1	ActionArg_2
comment	Using a positional argument where the argument is a context variable that has been passed in by reference					
P001ARG				SetContextVar	Name=John Smith	
				&mol_pos	Name	\$NameVerify

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2
mol_pos				#define_arg	name	verify_name
				Print	#name is	##name%
				Zstring.Compare	##name%	#verify_name

```
Working on Test Case Variable Combination mol pos
[P001ARG_MOL_POS] Executed Action Print with values [Name is, John Smith]
[P001ARG_MOL_POS] Execution Started Action Zstring.Compare with values [John Smith, John Smith]
[P001ARG_MOL_POS] Action ZSTRING.COMPARE SUCCESSFULLY Executed
```

Molecule: Passing by value & reference

Passing by value and reference

TestCase ID	Description	property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
comment	Using a named argument in a AppendToContextVar, where the named arg has been passed in as reference and value						
P003ARG				SetContextVar	Name=John Smith		
				SetContextVar	String		
				&mol_pos3	Name	%Name%	String
				Zstring.Compare	%String%	\$Name	

Passing by Value

Passing by Reference

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4
mol_pos3				#define_arg	name	value	stringVar	
				Print	#name is	#value		
				AppendToContextVar	#stringVar	##name%	can only be	#value

```

Working on Test Case Variable Combination mol_pos3
[P003ARG_MOL_POS3] Executed Action Print with values [Name is, John Smith]

[P003ARG_MOL_POS3] Action APPENDTOCONTEXTVAR Execution STARTED With Arguments contextvar = String valueToAppend = John Smith can only be John Smith
[P003ARG_MOL_POS3] Action APPENDTOCONTEXTVAR SUCCESSFULLY Executed

***** Molecule P003ARG_mol_pos3 Execution Finished *****

[P003ARG] Execution Started Action Zstring.Compare with values [John Smith can only be John Smith, John Smith can only be John Smith]
[P003ARG] Action ZSTRING.COMPARE SUCCESSFULLY Executed
    
```

Molecule: Passing Named arguments

Arguments passed as Key=Value pair

TestCase ID	Description	property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
comment	Using a named argument where the argument is a context variable that has been passed in by reference and value						
N001ARG				SetContextVar	Name=John Smith		
				&mol_pos	name=Name	verify_name=\$NameVerify	

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2
mol_pos				#define_arg	name	verify_name
				Print	#name is	##name%
				Zstring.Compare	##name%	#verify_name

Molecule: Passing by value & reference

Passing by value and reference

TestCase ID	Description	property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3
comment	Using a named argument in a AppendToContextVar, where the named arg has been passed in as reference and value						
N003ARG				SetContextVar	Name=John Smith		
				SetContextVar	String		
				&mol_pos3	name=Name	value=%Name%	stringVar=String
				Zstring.Compare	%String%	\$Name	

Pass by Value

Pass by Ref

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4
mol_pos3				#define_arg	name	value	stringVar	
				Print	#name is	#value		
				AppendToContextVar	#stringVar	##name%	can only be	#value

ture

Molecule: Referencing input arguments

#define_args allows the molecule to associate an input argument with a name, e.g. **#define_args value**

The input argument can then be referenced by prefixing a # to the name, e.g. **#value**

Input arguments can also be dereferenced, e.g. **%#name%**

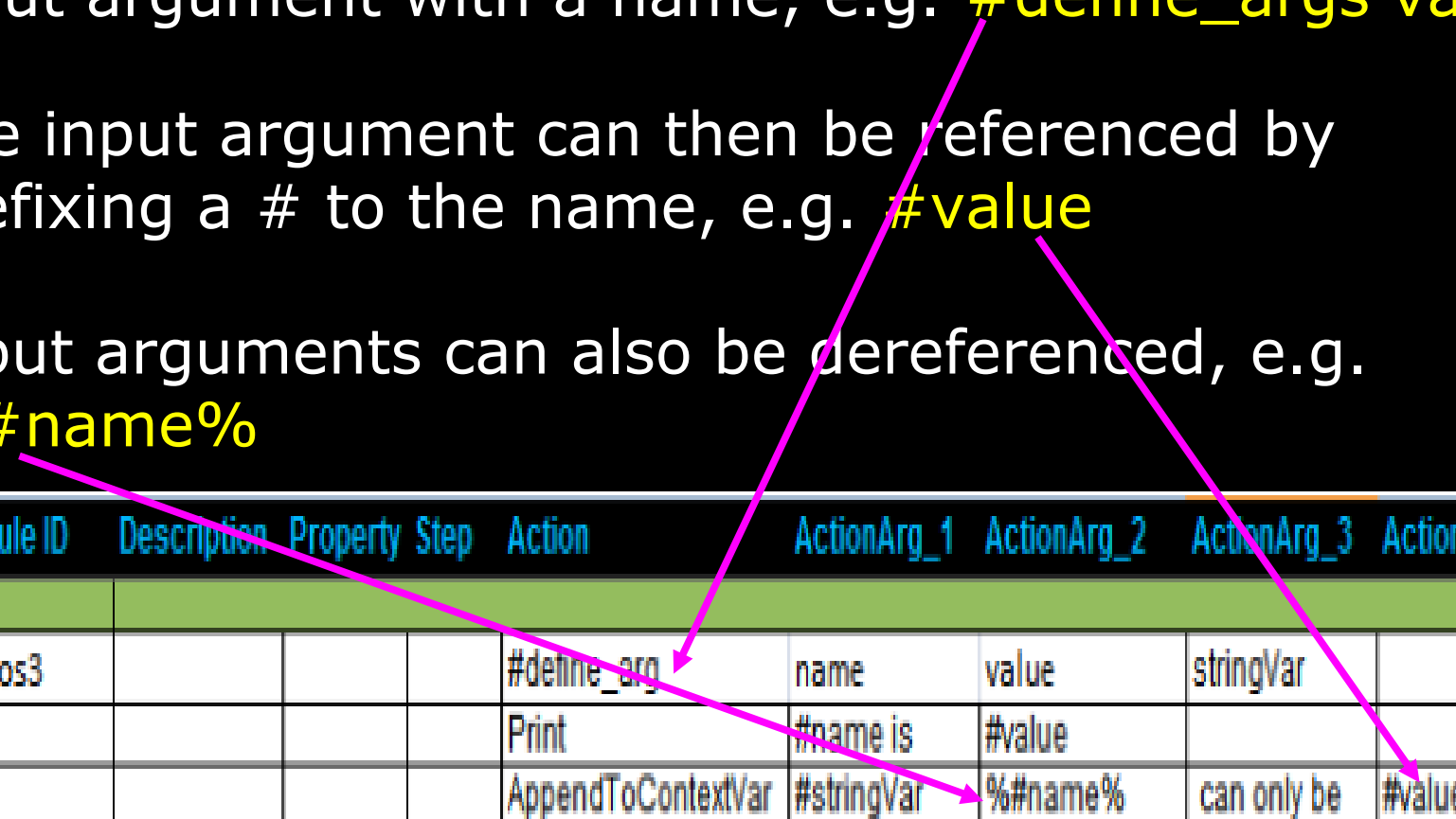


Diagram illustrating the flow of argument referencing:

- A pink arrow points from the text **#define_args value** to the **#define_args** cell in the table.
- A pink arrow points from the text **#value** to the **#value** cell in the table.
- A pink arrow points from the text **%#name%** to the **%#name%** cell in the table.

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2	ActionArg_3	ActionArg_4
mol_pos3				#define_args	name	value	stringVar	
				Print	#name is	#value		
				AppendToContextVar	#stringVar	%#name%	can only be	#value

What are atoms?

- ◉ **Smallest units of execution, that carry out desired operations**
 - An executable program or a script, that can be launched in a command prompt
 - A method in a dynamically linked library
 - A “built-in” function inside ZUG

ATOMS: How to invoke

- **In-Process**
 - **PackageName.Atomname**
- **Out-of-Process**
 - **@atomname**
- **Built-In**
 - **Atomname**

ATOMS: Passing arguments

- **Similar to passing arguments to molecules**
 - But, cannot pass named arguments
- **Pass by value**
 - `$macroname`
 - `%variablename%`
- **Pass by reference**
 - Name of a variable, e.g. `browser_handle`

ATOMS: Examples

- In-Process
- Out-of-Process
- Built-In

Molecule ID	Description	Property	Step	Action	ActionArg_1	ActionArg_2
mol_pos				#define_arg	name	verify_name
				Print	#name is	##name%
				Zstring.Compare	##name%	#verify_name

Built-In

In-Process

Login				SetContextVar	handle				
				@OpenBrowserWithUrl.BAT	\$URL	handle		@VerifyContent.BAT	\$LOGIN_TEXT

Out-of-Process

Exercise : Molecules

- Write a molecule to login to a site with a username and password
- Write two test cases using the same molecule to login to two different sites
- First testcase should call the molecule using Named argument passing.
- Second testcase should call the molecule using Positional argument passing.