**Working with SoapUI Properties**

**There are two types of properties in SoapUI:**

1. **Default Properties**: included in the SoapUI installation. We can edit some of the default properties but not all.
2. **Custom/user-defined properties**: These are defined by us at any level needed, such as global, project, test suite, test case or test step.

*Most often, properties are used to****store and retrieve the data****while executing the test cases. Internally property will store the value in key pair format.*

**For example**, in the below statement, “Local\_Property\_FromCurrency” is a key name and “USD” refers value. To access property value, we need to use property name.

*testRunner.testCase.testSteps[“Properties”].setPropertyValue*  
*( “****Local\_Property\_FromCurrency****“,****‘USD’****)*

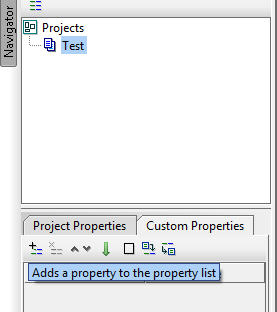
### Various property levels in SoapUI Pro

Let us discuss the various property levels in SoapUI Pro. In SoapUI there three levels of properties available.

**Level #1. Project and Custom Properties**

In this level, properties are divided into two sections. They are project properties and custom properties. These will appear at the bottom of the navigator panel when we click on the Project name. Project properties section has default properties which are created during the project creation for example, Name, Description, File etc.

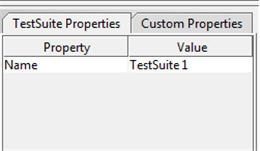
In order to create our own properties, we can use custom properties tab. Click on the plus icon to create properties:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/WORKING-WITH-PROPERTIES-new-1.jpg)

There are many other options available such as remove, move up, move down and sorting next to add. Any number of custom properties can be added and used by any sections(test suite, test cases) within the project.

**Level #2. Test Suite and Custom Properties**

These properties are visible only under the test suite. A test suite can contain any number of properties and they can be accessed from any test steps which belong to the said test suite.  
Test suite properties appear when click on the respective test suite name under the project. To add custom properties as needed, click on custom properties tab and click on the ‘+’ sign under it.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties2.jpg)

**Property #3. Test Case and Custom Properties**

Test case properties are accessible within the test case. They are not accessible by other test case steps or even test suite under the project.

### More details about properties with examples

Properties can store end points, login details, header information, and domain etc. even though we have discussed about writing and reading data to/from the properties, we are yet to discuss this topic in details with example.

The above discussed levels of properties are used in scripting to read the data.

**#1. Reading properties:**

We will look at how we can read properties in groovy script. In order to access different level properties, the following is the syntax:

**Project: Syntax:** ${# Project Name # Value }  
**Example:**

**def**projectPro = testRunner.testC***a***se.testSuite.project.getPropertyValue  
( “Project\_Level\_Property” )  
“Project\_Level\_Property” )  
log.info (projectPro)

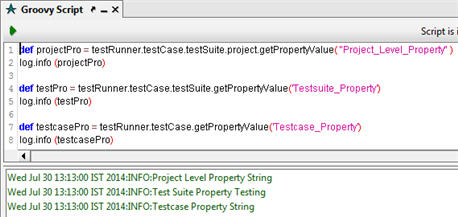
**Test suite: Syntax:** ${# TestSuite # Value }  
**Example:**

**def**testPro = testRunner.testCase.testSuite.getPropertyValue(‘Testsuite\_Property’)  
log.info (testPro)

**Test case: Syntax:** $ {# TestCase # Value }  
**Example:**

**def**testcasePro = testRunner.testCase.getPropertyValue(‘Testcase\_Property’)  
log.info (testcasePro)

Refer the screenshot below:

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties3.jpg)

**#2. Writing to properties:**

To do this, we have to use **setPropertyValue** method.

**Syntax: setPropertyValue (“property name”,”value”)**

If we assign values to unknown properties, then SoapUI will create these properties newly. For existing properties will receive the values during the assignment.

**#3. Removing Properties through Script:**

This can be done by right-clicking on the property name from the property panel. Then click on the Remove option from the context menu.  
To do this using script for removing the custom properties use the following statements for project, test suite or test case levels respectively:

testRunner.testCase.testSuite.project.removeProperty( “Testcase\_Property” );  
testRunner.testCase.testSuite.removeProperty( “Testcase\_Property” );  
testRunner.testCase.removeProperty( “Testcase\_Property” );

The above scripts are not optimum when we have multiple properties in each level as these steps have to be repeated several times for each property. An alternative is to iterate the properties through the script as below:

testRunner.testCase.properties.each  
{  
           key,value ->  
           testRunner.testCase.removeProperty(key)  
}

The above script will iterate till the last property available under the test case. “**Key**” refers the name of the property where as “**value**” denotes actual value of the property. We can modify the above script to remove the bulk property list present in various levels.

**#4. Add property:**

**AddProperty** method is used for this whose syntax is:

addProperty ( property name );

This can be adapted for each level as below:

testRunner.testCase.testSuite.project.addProperty(‘ProjectProperty1’)  
testRunner.testCase.testSuite.addProperty(‘TestsuiteProperty1’)  
testRunner.testCase.addProperty(‘TestcaseProperty1’)

After executing the above scripts, click on the project/test suite/test case name. Check the custom properties tab in property panel and the created property appears here. See below for reference:

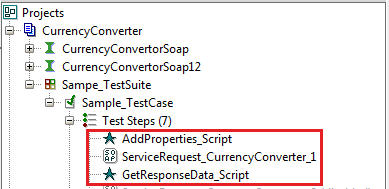
------------

### Using properties in services

In this section, we will learn how we can use the properties in services and we are going to use the above scripts for adding, assigning, retrieving property data with currency converter web service.

**Integrating Properties in Service:**

Let us start adding test steps as shown in the below screenshot.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties5.jpg)

In the above screenshot, AddProperties\_Script test step contains the following script which adds two properties such as **Property\_FromCurrency** and **Property\_ToCurrency**.

// Add Properties  
testRunner.testCase.addProperty(‘Property\_FromCurrency’)  
testRunner.testCase.addProperty(‘Property\_ToCurrency’)

// Assign values to the Properties  
testRunner.testCase.setPropertyValue(‘Property\_FromCurrency’,’USD’)  
testRunner.testCase.setPropertyValue(‘Property\_ToCurrency’,’INR’)

In the **ServiceRequest\_CurrencyConverter\_1** contains the request with input parameters as seen below:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties6.jpg)

Assigned values in the properties will be transferred to these parameters during the execution. Following this test step, **GetResponseData\_Script** test step has the script that will get the response value and show the result in the log. Here’s the script.

// Get Response data from the service  
def response = context.expand( ‘${ServiceRequest\_Currency  
Converter\_1#Response}’ )  
def parsedResponse = new XmlSlurper().parseText(response)  
String convertedValue = parsedResponse.Body.ConversionRateResponse.  
ConversionRateResult.text()  
log.info(convertedValue)

Once all the steps are ready, double click on the test suite name and run the test suite. Then, double click on the **ServiceRequest\_CurrencyConverter\_1**and see the response section.

This is what we would find:

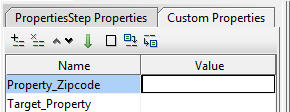
* Response will be received
* Open the script log to see the resultant data that is converted based on input parameters

This is how we can pass the parameters to the input request and get the response through the script using properties. Going further we can also pass the response value to another service as input.

### ****Property Transfer****

The property transfer test step transfers the property data from one property to another during the execution. Let us see in brief how we can create property transfer test step and how the property value is transferred between two properties.

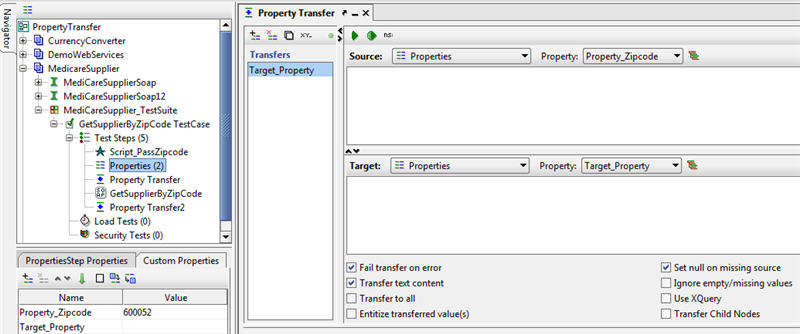
* Right click on the test case name under the test suite
* Click **Add Step** and then click **Properties**option from context menu
* Repeat the above steps to create the second property. See the below screenshot:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties7.jpg)

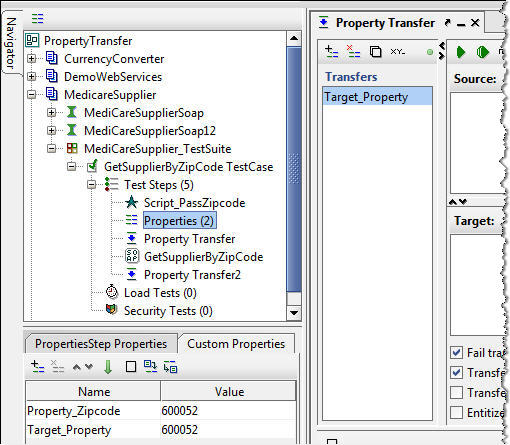
* Now we have to add property transfer test step.
* Right click on the test case name and click property transfer option from the context menu
* Enter your desired property transfer name and then click OK
* Click Add i.e. plus sign in the property transfer tool bar
* Specify the transfer name and then click OK button
* In the right side panel, there are two sections available: Source and Target.

Choose the source as **Properties** and property as **Property\_Zipcode**. Do the same in the target section. Choose **Target\_Property** from the property drop down. When run icon, the property value will be transferred from **Property\_Zipcode** to **Target\_Property**.

(Click on image for enlarged view)

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/WORKING-WITH-PROPERTIES-new-3.jpg)

See the transferred value as shown in the below screenshot.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/WORKING-WITH-PROPERTIES-new-4.jpg)

**Note:** Source property should contain the default value.

In addition to this, there are many options available in the property transfer screen.

* Fail Transfer on Error
* Transfer Text Content
* Transfer to All
* Entitize Transferred Value(s)
* Set Null on Missing Source
* Ignore Empty/Missing Values
* Use XQuery
* Transfer Child Nodes

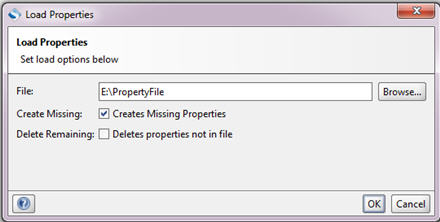
**Load Properties from External Source:**

To load properties from an external source, follow these steps.

* Add Properties test step under the test case
* Enter the property step name and then click OK
* In the property panel under the navigation panel, click Custom Properties tab
* Click [Working with Properties](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties.jpg) icon to load the properties from the external property file

**Note:**Property file should be saved or present on your computer. To save the properties click  icon.

Then, go to the respective drive and pick the property as shown below:

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2015/05/Working-with-Properties9.jpg)

On OK, we can see the loaded properties and their values in the Custom Properties tab.

### ****Conclusion****

Well, that’s properties for us!

Each level properties have their own characteristics. During your SoapUI practice, try to include properties whenever possible with the groovy script test step for adding, removing, assigning and retrieving property data. This is not only useful when you practice with the services but also critical for real application testing as this technique is very helpful to [assert your test cases](http://www.softwaretestinghelp.com/soapui-tutorial-5-soapui-assertions/).

Transferring properties among test steps is easier than writing repeated scripts to create new. SoapUI also gives the wonderful feature to import and export properties. This feature will be useful when we are using common properties such as login details, session details etc., for multiple projects. That way, we don’t have to create the same properties again and again for multiple projects. We can simply change in the property value against the properties based on the projects.