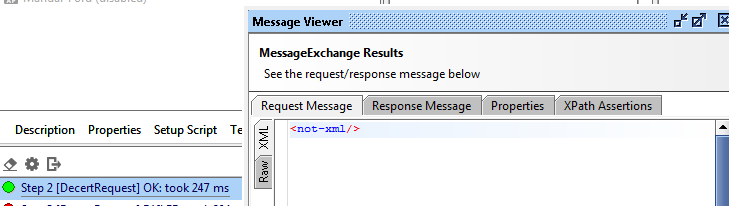
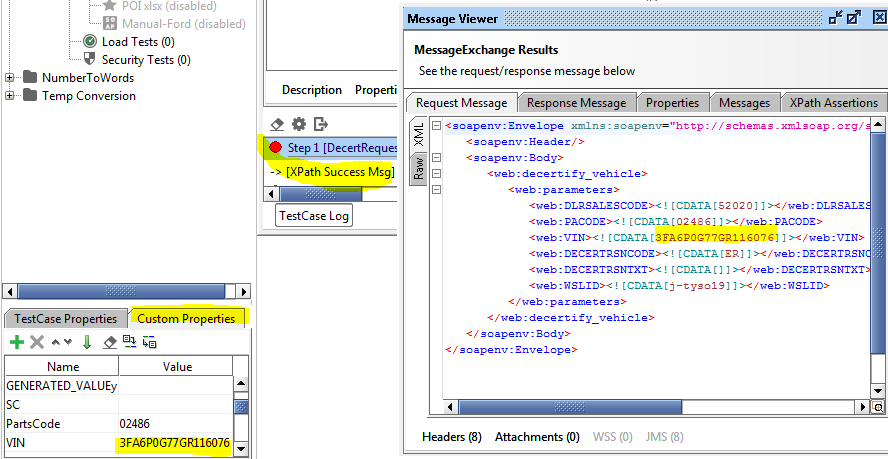
SoapUI Gotchas

1. Do not run scripts with looping when the Request/Response window is open. It usually hangs SoapUI and it must be killed thru Task Manager. Any unsaved changes are lost.
2. SoapUI can run out of memory and return empty Requests and Responses. You won't know this until you examine the R/R. Restart SoapUI. It may be a timing issue, since later tests will contain results. 
3. Assertions
   1. Empty Excel rows and assertions. Humans can't tell the difference between a blank Excel row and empty row. XPath will fail a blank row since it doesn't match a NULL row.
   2. Certification – Success Msg vs Fault Msg
      1. A successful operation never contains a Fault Msg. When empty rows (different from NULL rows) in Excel existed, a Fault msg was returned but it was not caught as a Fail. Was this because the Expected Result as a space in Excel and a space char existed in the error msg returned? That prompted the move to XPath Contains assertions.
      2. Because the Success Msg always returns a value – either True or False – this should be the determining test factor.
      3. The regular "Contains" assertion will not produce a FAIL when blank input data (i.e. a blank Excel row) is compared to a Response returned with data. From the error msg in SoapUI, it appears that the input sent from a blank Excel row is a space and since the error msg "contains" space chars, it is incorrectly marked as a PASS.
      4. XPath assertions do not have the 'ignore case' option.
4. The burp - Running at the Test Case level appears to run the last test case twice, which usually causes a failure if you are certifying or decertifying a vehicle.
5. Custom properties remain in memory and can cause a false fail on the next test case run. SOLUTION: Add this to the test case TearDown Script section:

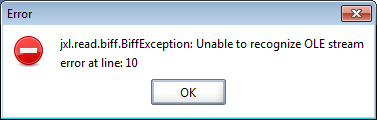
context.testCase.properties.each {

context.testCase.properties[it.key].value = ''

}

1. If you leave other Test Steps enabled, they will be executed which may cause bad or misleading results. You will notice that a lot more test steps in the results than you normally get.
2. Sometimes test data from a previously run test appears in the first test request. Clear out Custom Properties (see below – VIN 3FA6P...was not in the DD input sheet)
3. If you forgot to add jxl.jar to the proper folder or path isn't set correctly, you will see an error msg like this:

org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed: Script1.groovy: 5: unable to resolve class Workbook @ line 5, column 10. Workbook workbook = Workbook.getWorkbook(new File("c:\\SoapUI Tutorial\\numbers.xls"))//Data file to be read ^ org.codehaus.groovy.syntax.SyntaxException: unable to resolve class Workbook @ line 5, column 10. at org.codehaus.groovy.ast.ClassCodeVisitorSupport.addError(ClassCodeVisitorSupport.java:149) at

1. If you try to read a .xlsx sheet, you will get this error msg:  Apache POI allows the use of .xlsx sheets.