Rating Test User Guide

This guide does not explain technical details how Rating test solution works (please refer to developers guide if needed).

Goal of this guide is to give basic understanding on following items:

[Rating toolkit function and parameters 2](#_Toc414377818)

[How to use Rating feature in tests. 2](#_Toc414377819)

[Execution parameters (JVM parameter) in eclipse and Jenkins 4](#_Toc414377820)

[In Eclipse: 4](#_Toc414377821)

[!!! Enabling Screenshot acquiring feature. 5](#_Toc414377822)

[In Jenkins 6](#_Toc414377823)

[!!! Enabling Screenshot acquiring feature. 6](#_Toc414377824)

[Resulting artifacts and usage 7](#_Toc414377825)

[Console output: 7](#_Toc414377826)

[Screen shots: 8](#_Toc414377827)

[Results files located in *target/test-output*. 9](#_Toc414377828)

[Defect creation 11](#_Toc414377829)

[Appendix 1 – Rating request/response log – how to convert it to usual XML structure 12](#_Toc414377830)

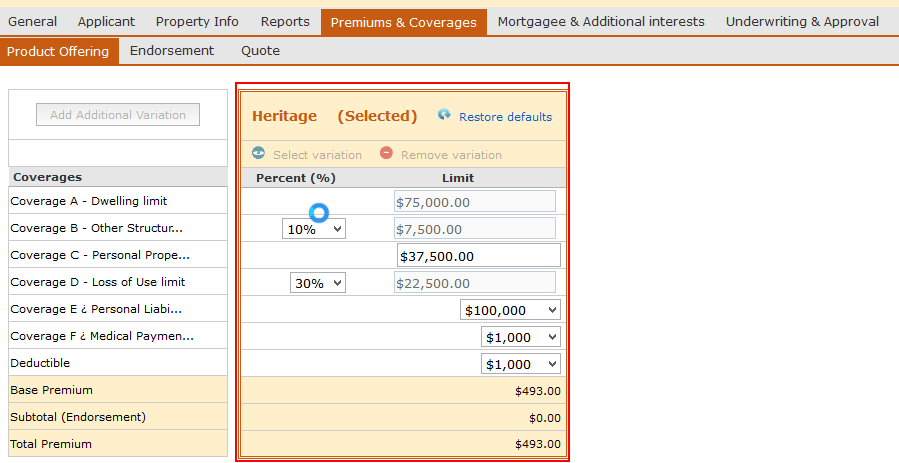
# Rating toolkit function and parameters

## How to use Rating feature in tests.

1. When you create tests for Rating testing you need to use Rating Helper.
   1. String policyNumber = **RatingHelper**.*createHssPolicy*(*getDataSet*());
2. Next important parameter is bundle. That you can see on Premium tab which to use.
   1. String bundle = RatingHelper.*getBundleHO3Policy*(*getDataSet*());

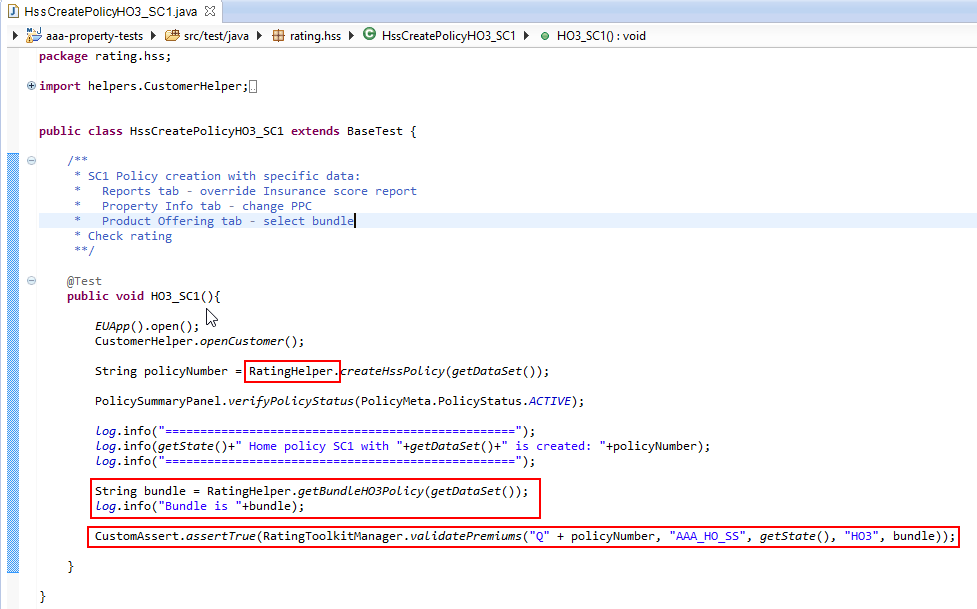
*log*.info("Bundle is "+bundle);

Default bundle “Heritage” – see screenshot.



1. And finally the main line in such test is following; this will do the main task – compare premiums.
   1. CustomAssert.*assertTrue*(RatingToolkitManager.*validatePremiums*("Q" + policyNumber, "AAA\_HO\_SS", *getState*(), "HO3", bundle));

Exigen Rating test as an Example.



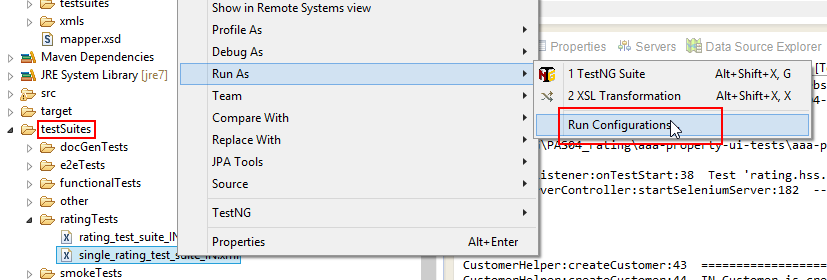
You can examine other tests in **src/test/java/rating/hss.**

# Additional Execution parameters in eclipse and Jenkins (JVM and Screenshots)

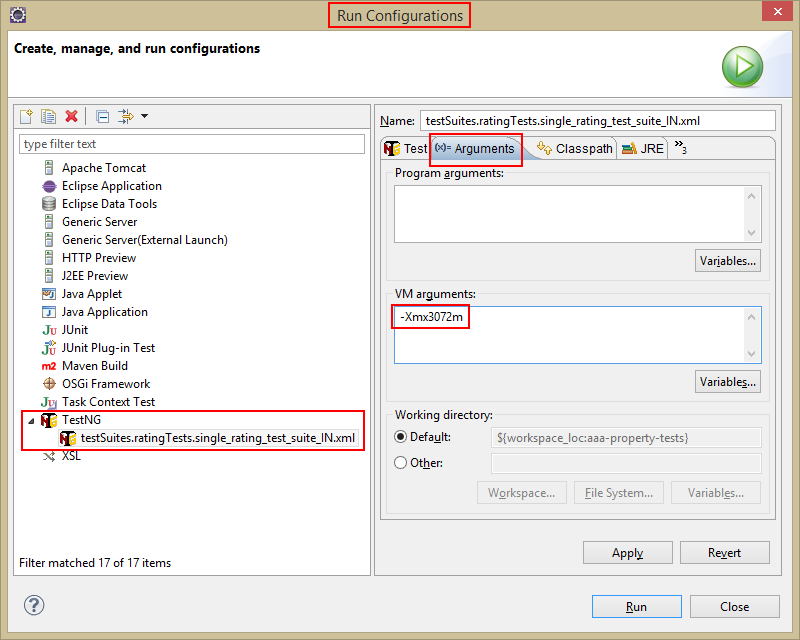
## In Eclipse:

To be able to execute rating test locally in eclipse you need to follow these simple steps.

1. Take test suite you created click “Run As” -> “Run Configuration”. You will see if your test suite is visible in section “Test NG”.

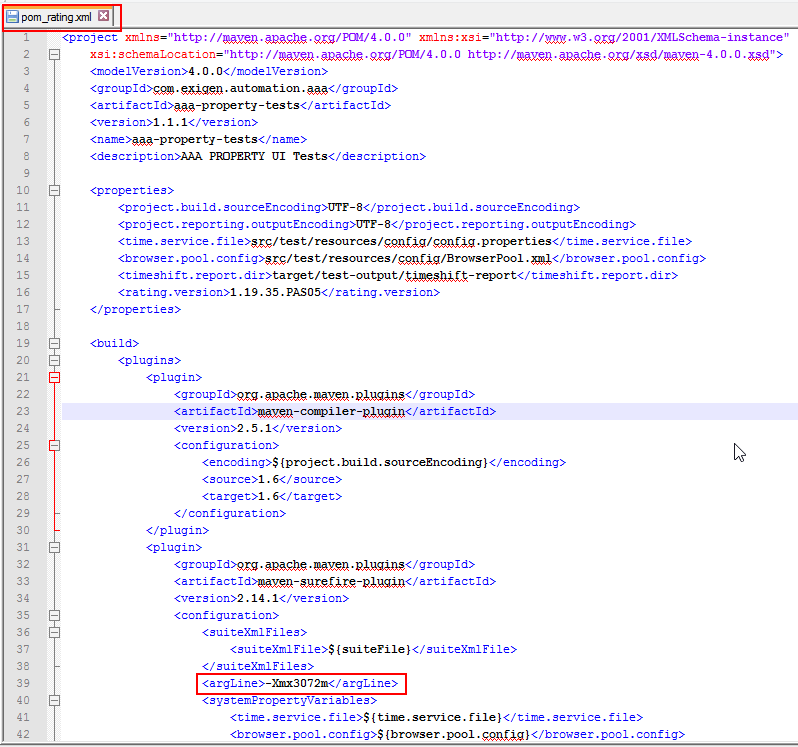


1. If the test suite is not visible, then you need to Run this suite, in fact just start execution and then stop it, after that you should see that suite file as in screenshot below.



1. After you get this suite file in the list you need to go to tab “Arguments” tab and set VM arguments value “-Xmx3072m”. If this is not done test will fail with Java Heap space out of memory error.

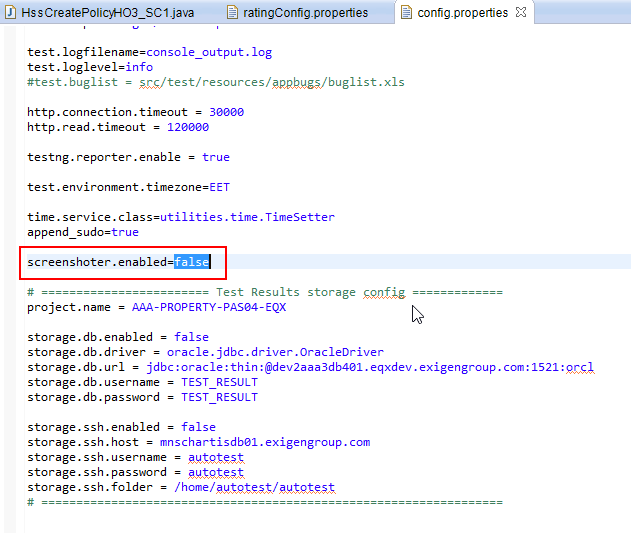
Same parameter is also added in pom\_rating.xml, see screenshot below.



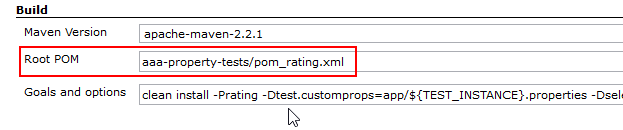
### 

### !!! Enabling Screenshot acquiring feature (by default it’s turned off).

For local test execution change “screenshoter.enabled” parameter to “true”.

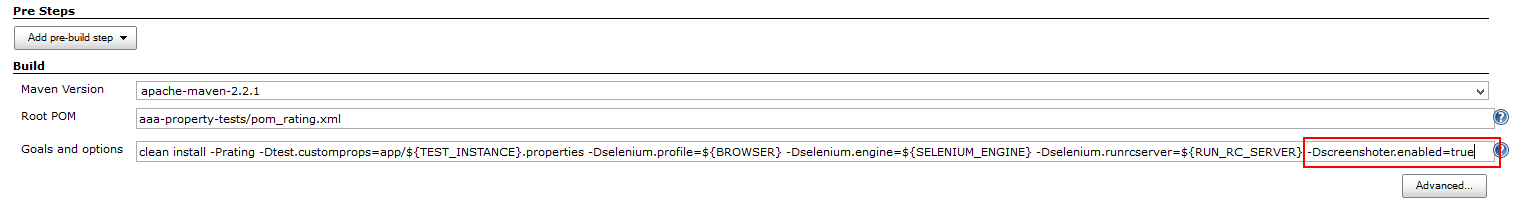


## In Jenkins

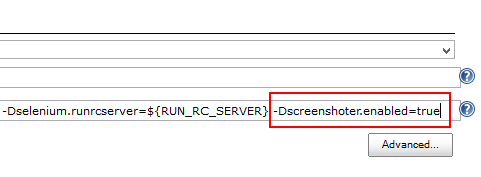
For Jenkins job you need to set which pom.xml to use, for rating tests it should be “pom\_rating.xml:

### !!! Enabling Screenshot acquiring feature.

Add additional parameter in Jenkins job to enable screenshot capture for every tab:



Enlarged:



# Resulting artifacts and usage

After you successfully executed test there are lot of results to work with:

### Console output:

BaseTest:<clinit>:95 Test instance: http://rix-websphere06.exigenservices.com:9081/aaa-app/

BaseTest:<clinit>:96 Build: EIS\_5.2.23.5-AAA\_PAS04-20150312-63745

[TestNG] Running:

E:\TestNG\PAS04\_rating\aaa-property-ui-tests\aaa-property-tests\testSuites\ratingTests\single\_rating\_test\_suite\_IN.xml

CustomTestListener:onTestStart:38 Test 'rating.hss.CreateAccount.createAccount(IN)' has STARTED

SeleniumServerController:startSeleniumServer:182 -------> Selenuim Server Started

CustomerHelper:createCustomer:43 ============================

CustomerHelper:createCustomer:44 IN Customer is created: 700027266

CustomerHelper:createCustomer:45 ============================

CustomTestListener:onTestSuccess:45 Test 'rating.hss.CreateAccount.createAccount(IN)' has PASSED

CustomTestListener:onTestStart:38 Test 'rating.hss.HssCreatePolicyHO3\_SC1.HO3\_SC1(IN)' has STARTED

HssCreatePolicyHO3\_SC1:HO3\_SC1:36 ==================================================

HssCreatePolicyHO3\_SC1:HO3\_SC1:37 IN Home policy SC1 with HO3\_Dataset1 is created: INH3910068903

HssCreatePolicyHO3\_SC1:HO3\_SC1:38 ==================================================

HssCreatePolicyHO3\_SC1:HO3\_SC1:41 Bundle is Heritage

Ssh:downloadFile:89 SSH: File '/logs/aaa-rating-engine-app-soap.log' was downloaded to 'E:\TestNG\PAS04\_rating\aaa-property-ui-tests\aaa-property-tests\target\test-output'.

ReflectionServiceFactoryBean:buildServiceFromClass:445 Creating Service {http://home.property.rating.aaa.exigen.com/}RatingService from class com.exigen.aaa.rating.property.home.RatingService

Java5TypeCreator:getOrCreateParameterizedType:181 Could not map TypeVariable named T from java.util.Map<java.lang.String, T> with initial mapping null to a known class. Using Object.

ReflectionServiceFactoryBean:buildServiceFromClass:445 Creating Service {http://home.property.rating.aaa.exigen.com/}RatingService from class com.exigen.aaa.rating.property.home.RatingService

Java5TypeCreator:getOrCreateParameterizedType:181 Could not map TypeVariable named T from java.util.Map<java.lang.String, T> with initial mapping null to a known class. Using Object.

RatingToolkitManager:validatePremiums:43 Log for policy #QINH3910068903 has been successfully parsed.

RaterPremiumCalculatorHOUTImpl:fillRater:225 Comparison for policy #QINH3910068903 was success.

CustomTestListener:onTestSuccess:45 Test 'rating.hss.HssCreatePolicyHO3\_SC1.HO3\_SC1(IN)' has PASSED

===============================================

IN Rating Tests Suite

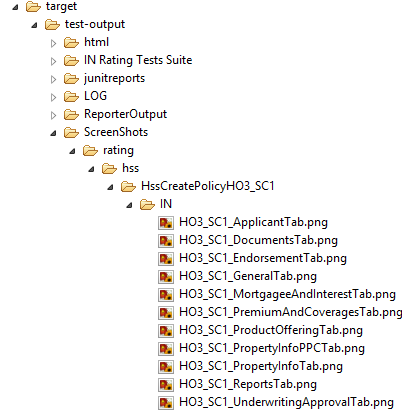
Total tests run: 2, Failures: 0, Skips: 0

===============================================

### Screen shots:

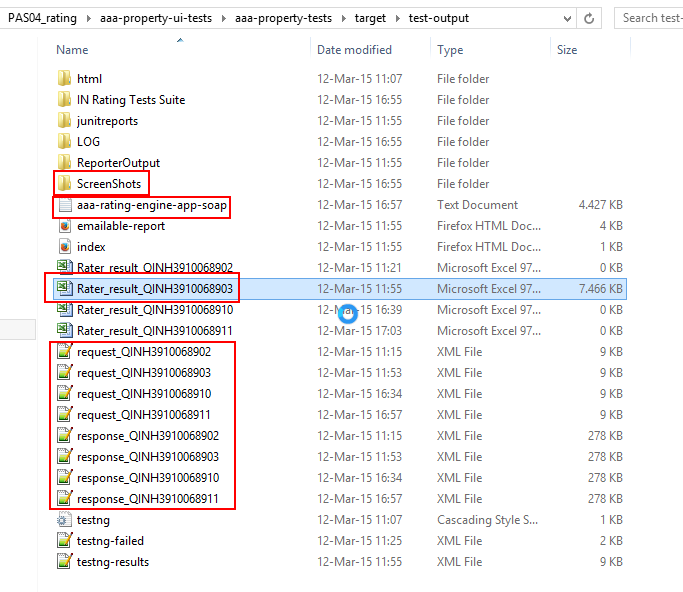
Screenshots are taken for each tab because that can be used later upon defect creation. This is very important generated output for later analysis in case there are any issues.

This is how it looks in test output folder.



## Result files located in *target/test-output*.

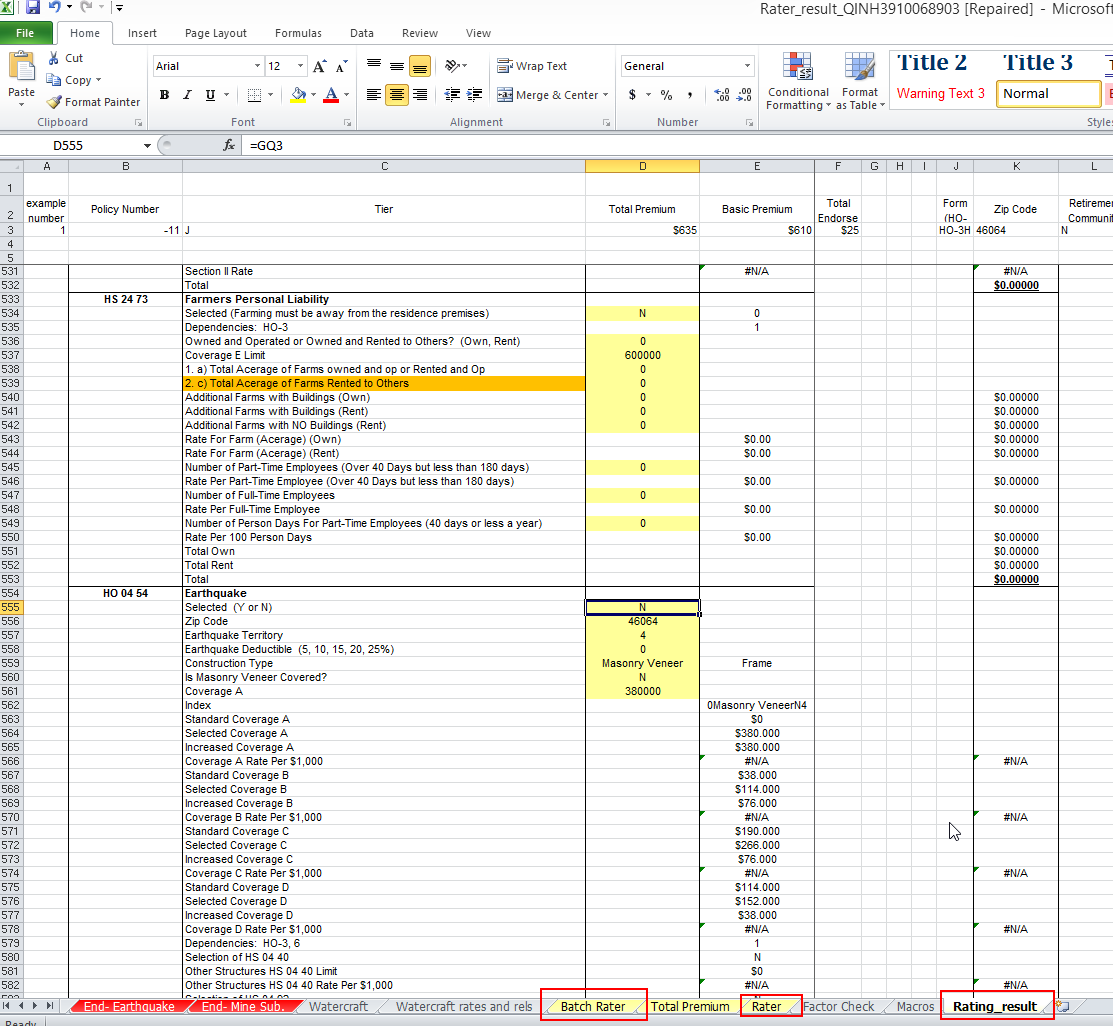
All the results are found in test-output folder – see screenshot below.



Full Rating log is also stored – aaa-rating-engine-app-soap.txt.

Rater Results xls files are prepared for each policy.

Also request and response XMLs from the whole rating log are cut out and stored separately for each policy.

s

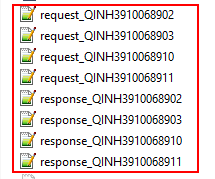
# Defect creation

If there is some issue in Rating\_results sheet then you have everything you need to create defect.

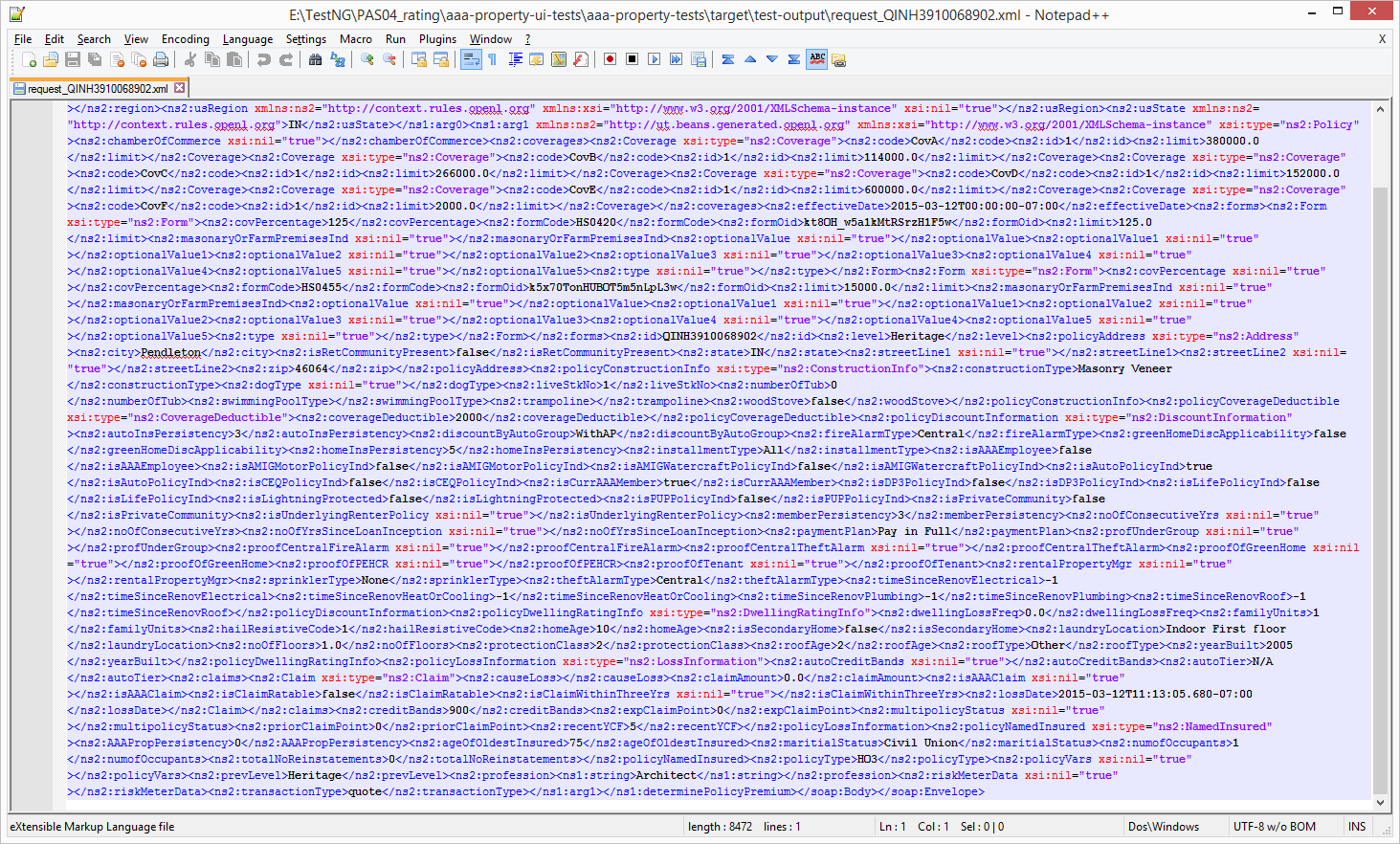
1. Test/Scenario;
2. Screenshots for all the inputs on UI;
3. Request / Response XML files;
4. Batch Rater - Results XLS file.

**!** All this is mandatory to create defect.

# Appendix 1 – Rating request/response log – how to convert it to usual XML structure



This is how it looks if you open in notepad++



You can easily convert this XML to pretty print. You need to put Plugin XML Tools.

