Rating Tests Guide

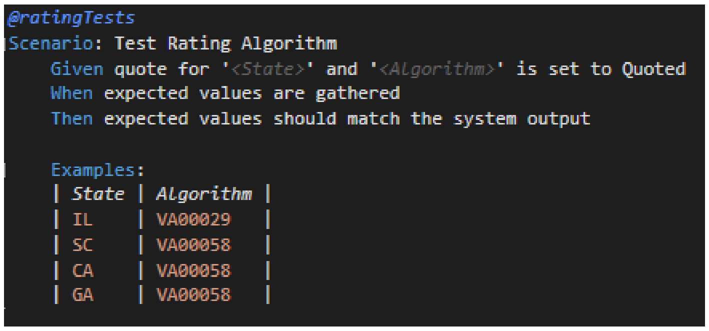
The purpose of this user guide is to help testers use execute and analyze rating tests in Apollo.

Table of Contents

Description

How it works

1. **Scenario Setup:**
   To run any tests, we first need a fully elaborated quote (risk, coverage, question answers & premium), that is the purpose of the step "**Given** quote for 'state' and 'algorithm' is set to Quoted".
   Using two parameters: State & Algorithm (e.g. VA00029).

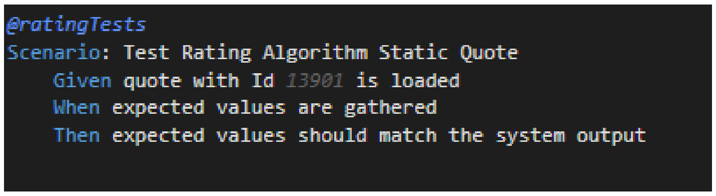


```
@ratingTests
Scenario: Test Rating Algorithm
    Given quote for '<State>' and '<Algorithm>' is set to Quoted
    When expected values are gathered
    Then expected values should match the system output

    Examples:
    | State | Algorithm |
    | IL    | VA00029   |
    | SC    | VA00058   |
    | CA    | VA00058   |
    | GA    | VA00058   |
```

   Details: this step will use Rest API to go through the quote process in order to get a premium meeting the both of the parameters. This uses default question answers from "`ApolloQA.Data.TestData.AnswersHydrator`".

   Alternatively, we can use the desired quote to run tests on by using the quote Id. (ApplicationId)



```
@ratingTests
Scenario: Test Rating Algorithm Static Quote
    Given quote with Id 13901 is loaded
    When expected values are gathered
    Then expected values should match the system output
```

2. **Gathering expected premium along with all its factors**
   There is a Rating Engine in place (`ApolloQA.Data.Rating.Engine), this class uses Factors(e.g. Class Code Factor) which uses Knownfields (E.g. Class Code) both resolving to a single value to then be used on the algorithm formula(E.g. VA00029) in the rating engine.`

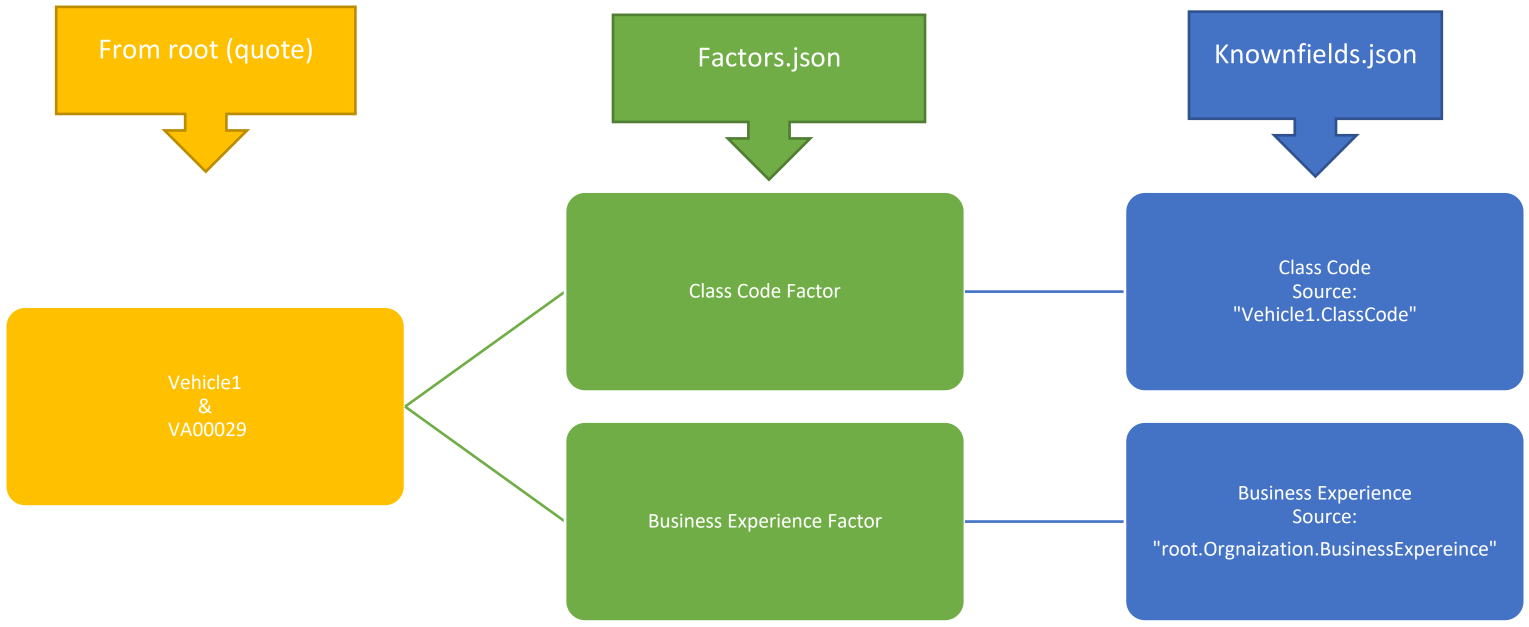
   `The manual used to output a premium comes from the PDF manual which is parsed into tables. Each state's manual can be found in "ApolloQA.Data.RatingManual".`

   `Details: The system uses Rest API, Cosmos & SQL to gather all the data points directly from their source. The source of each known field is configured in the source property of each object in "ApolloQA.Data.Rating.Knownfields.json".`

3. **Testing that expected results are identical to actual results from the system.**
   Expected Results: Json object generated by "`ApolloQA.Data.Rating.Engine.Run()`".
   `Actual Results: Json object generated by the system upon rating. (Rating Worksheet object)`

   `The step iterates through every Vehicle & Coverage from the expected results object, Failing if the premium is different.`
   `Also, it will iterate through every single Factor and compare both resulting factor. (e.g. 1.05)`

Rating Engine (Visual)



Vehicle & Coverage:

- engine to iterate through every vehicle and all coverages associated. (policy & vehicle level)

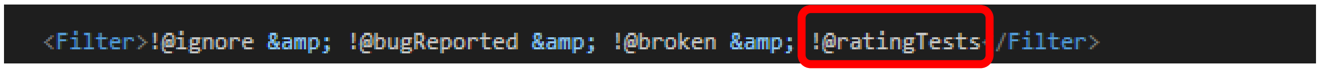
Factors:

- factors are read from algorithm formula (e.g. VA00029). The system will attempt to find each factor listed in `ApolloQA.Data.Rating.Factors.json`.
- `Each factor is instantiated as a ApolloQA.Data.Rating.Factor object.`
- `Each factor is resolved one of two ways:`
    1. **`(Default)`** `resolving each Knownfield associated (must exist in knownfields.json), then it will find the factor in its corresponding table (e.g. VA00029.ClassCodeFactor) from the Rating Manual.`
    2. `CustomCalculation (triggered if property “CustomCalculation”=true in Factors.json) The system will attempt to find a the source property value in Apollo.Data.Rating.Factor as a private member.`
- `Each knownfield is resolved by resolving the value of its source property. (e.g. root.Organization.BusinessExperience)`

Execution

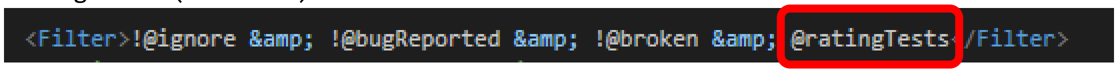
1. Because rating tests are too extensive to run on a regular basis, therefore they're tagged with "`@ratingTests`" `which is being ignored in default.srprofile.`

   `Ignored (with '!'):`



```
<Filter>!@ignore &amp; !@bugReported &amp; !@broken &amp; !@ratingTests</Filter>
```

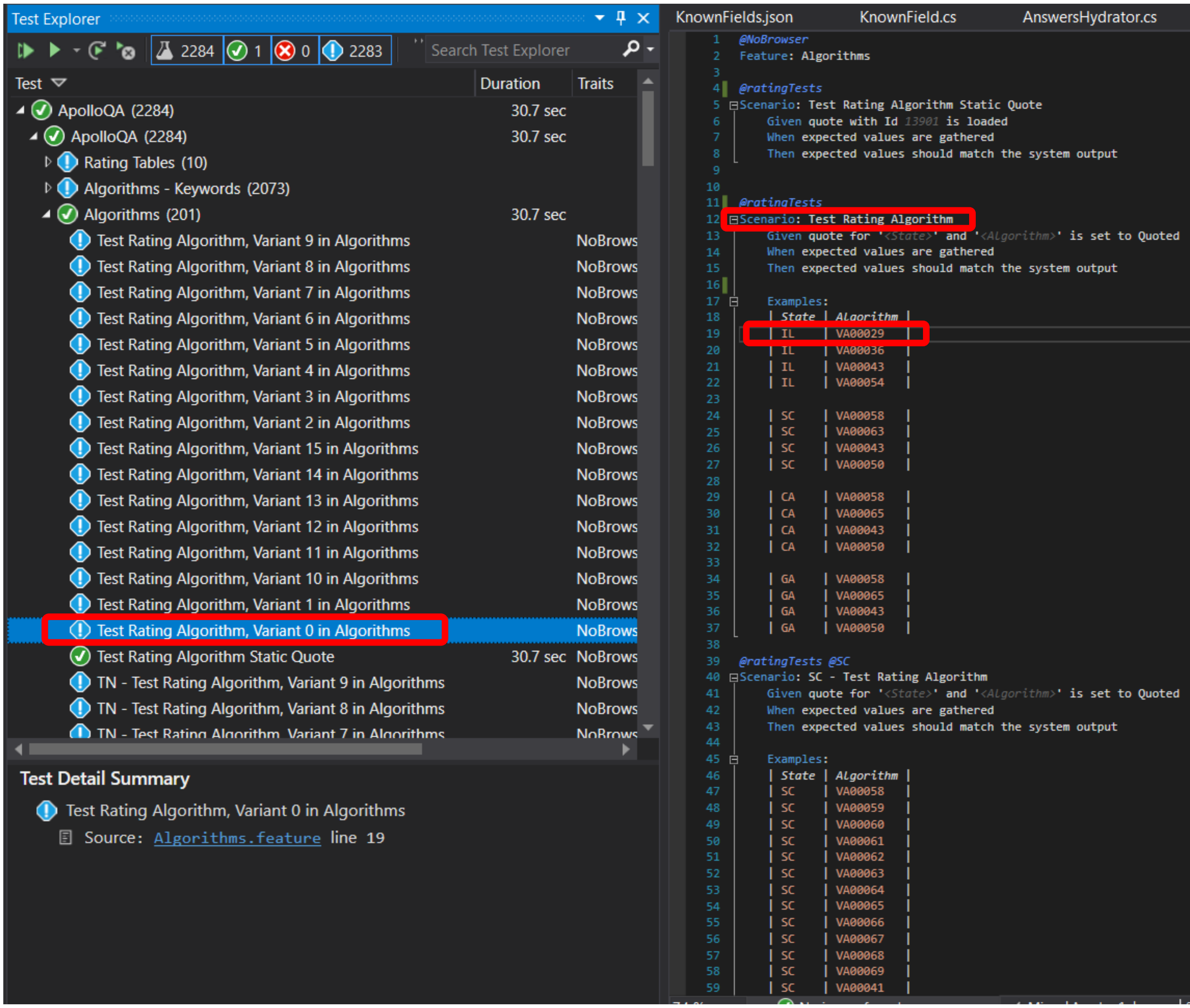
   `Not Ignored (without:):`



```
<Filter>!@ignore &amp; !@bugReported &amp; !@broken &amp; @ratingTests</Filter>
```

   `note: make sure to rebuild solution for the above changes to take effect`
   `Important: never push to master without ignoring the rating tests because execution is really extensive when triggered in the pipeline.`

2. Select the desired test to run



Test Explorer

2284 | 1 | 0 | 2283 | Search Test Explorer

Test	Duration	Traits
ApolloQA (2284)	30.7 sec	
ApolloQA (2284)	30.7 sec	
Rating Tables (10)		
Algorithms - Keywords (2073)		
Algorithms (201)	30.7 sec	
Test Rating Algorithm, Variant 9 in Algorithms		NoBrows
Test Rating Algorithm, Variant 8 in Algorithms		NoBrows
Test Rating Algorithm, Variant 7 in Algorithms		NoBrows
Test Rating Algorithm, Variant 6 in Algorithms		NoBrows
Test Rating Algorithm, Variant 5 in Algorithms		NoBrows
Test Rating Algorithm, Variant 4 in Algorithms		NoBrows
Test Rating Algorithm, Variant 3 in Algorithms		NoBrows
Test Rating Algorithm, Variant 2 in Algorithms		NoBrows
Test Rating Algorithm, Variant 15 in Algorithms		NoBrows
Test Rating Algorithm, Variant 14 in Algorithms		NoBrows
Test Rating Algorithm, Variant 13 in Algorithms		NoBrows
Test Rating Algorithm, Variant 12 in Algorithms		NoBrows
Test Rating Algorithm, Variant 11 in Algorithms		NoBrows
Test Rating Algorithm, Variant 10 in Algorithms		NoBrows
Test Rating Algorithm, Variant 1 in Algorithms		NoBrows
Test Rating Algorithm, Variant 0 in Algorithms		NoBrows
Test Rating Algorithm Static Quote	30.7 sec	NoBrows
TN - Test Rating Algorithm, Variant 9 in Algorithms		NoBrows
TN - Test Rating Algorithm, Variant 8 in Algorithms		NoBrows
TN - Test Rating Algorithm, Variant 7 in Algorithms		NoBrows

Test Detail Summary

Test Rating Algorithm, Variant 0 in Algorithms

Source: Algorithms.feature line 19

KnownFields.json | KnownField.cs | AnswersHydrator.cs

```
1   @NoBrowser
2   Feature: Algorithms
3
4   @ratingTests
5   Scenario: Test Rating Algorithm Static Quote
6       Given quote with Id 13901 is loaded
7       When expected values are gathered
8       Then expected values should match the system output
9
10
11  @ratingTests
12  Scenario: Test Rating Algorithm
13      Given quote for '<State>' and '<Algorithm>' is set to Quoted
14      When expected values are gathered
15      Then expected values should match the system output
16
17      Examples:
18      | State | Algorithm |
19      | IL    | VA00029   |
20      | IL    | VA00036   |
21      | IL    | VA00043   |
22      | IL    | VA00054   |
23
24      | SC    | VA00058   |
25      | SC    | VA00063   |
26      | SC    | VA00043   |
27      | SC    | VA00050   |
28
29      | CA    | VA00058   |
30      | CA    | VA00065   |
31      | CA    | VA00043   |
32      | CA    | VA00050   |
33
34      | GA    | VA00058   |
35      | GA    | VA00065   |
36      | GA    | VA00043   |
37      | GA    | VA00050   |
38
39  @ratingTests @SC
40  Scenario: SC - Test Rating Algorithm
41      Given quote for '<State>' and '<Algorithm>' is set to Quoted
42      When expected values are gathered
43      Then expected values should match the system output
44
45      Examples:
46      | State | Algorithm |
47      | SC    | VA00058   |
48      | SC    | VA00059   |
49      | SC    | VA00060   |
50      | SC    | VA00061   |
51      | SC    | VA00062   |
52      | SC    | VA00063   |
53      | SC    | VA00064   |
54      | SC    | VA00065   |
55      | SC    | VA00066   |
56      | SC    | VA00067   |
57      | SC    | VA00068   |
58      | SC    | VA00069   |
59      | SC    | VA00041   |
```

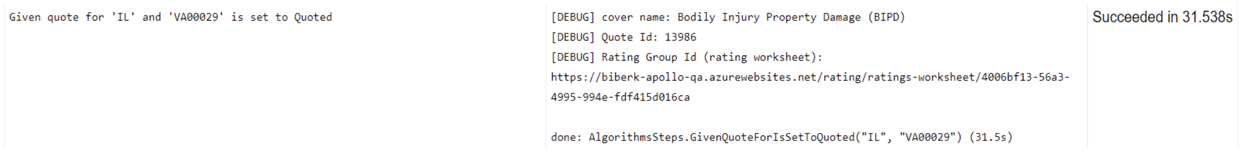
Test result analysis

Test results are found in ./`ApolloQA/TestResults/TestRunReport.html`

Step Results:

1. `Given quote for 'State' and 'Algorithm' is set to quoted.`

   `Output:`
   `Quote Id`
   `Link to Rating Worksheet`



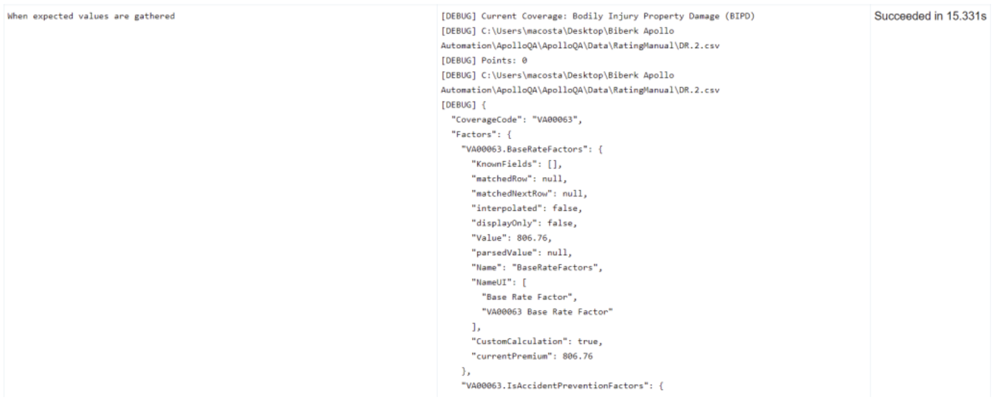
```
Given quote for 'IL' and 'VA00029' is set to Quoted
```

```
[DEBUG] cover name: Bodily Injury Property Damage (BIPD)
[DEBUG] Quote Id: 13986
[DEBUG] Rating Group Id (rating worksheet):
https://biberk-apollo-qa.azurewebsites.net/rating/ratings-worksheet/4006bf13-56a3-
4995-994e-fdf415d016ca

done: AlgorithmsSteps.GivenQuoteForIsSetToQuoted("IL", "VA00029") (31.5s)
```

Succeeded in 31.538s

2. When expected values are gathered.
   Visual of the entire expected object that was used for the test
   Output:
   Matched Row in the rating worksheet.
   Resulting Value for each Factor.
   Resulting Value for each Knownfield.



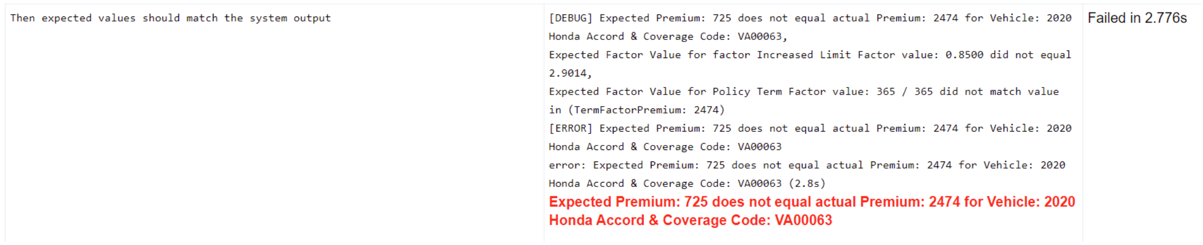
```
When expected values are gathered
```

```
[DEBUG] Current Coverage: Bodily Injury Property Damage (BIPD)
[DEBUG] C:\Users\macosta\Desktop\Biberk Apollo
Automation\ApolloQA\ApolloQA\Data\RatingManual\DR.2.csv
[DEBUG] Points: 0
[DEBUG] C:\Users\macosta\Desktop\Biberk Apollo
Automation\ApolloQA\ApolloQA\Data\RatingManual\DR.2.csv
[DEBUG] {
  "CoverageCode": "VA00063",
  "Factors": {
    "VA00063.BaseRateFactors": {
      "KnownFields": [],
      "matchedRow": null,
      "matchedNextRow": null,
      "interpolated": false,
      "displayOnly": false,
      "Value": 806.76,
      "parsedValue": null,
      "Name": "BaseRateFactors",
      "NameUI": [
        "Base Rate Factor",
        "VA00063 Base Rate Factor"
      ],
      "CustomCalculation": true,
      "currentPremium": 806.76
    },
    "VA00063.IsAccidentPreventionFactors": {
```

Succeeded in 15.331s

3. Then expected values should match the system output
    result of comparison of both objects

   Output:
   Difference in premiums.
   Difference in Factors.
   Any factors expected but not found in the rating worksheet.
   e.g.



```
Then expected values should match the system output
```

```
[DEBUG] Expected Premium: 725 does not equal actual Premium: 2474 for Vehicle: 2020
Honda Accord & Coverage Code: VA00063,
Expected Factor Value for factor Increased Limit Factor value: 0.8500 did not equal
2.9014,
Expected Factor Value for Policy Term Factor value: 365 / 365 did not match value
in (TermFactorPremium: 2474)
[ERROR] Expected Premium: 725 does not equal actual Premium: 2474 for Vehicle: 2020
Honda Accord & Coverage Code: VA00063
error: Expected Premium: 725 does not equal actual Premium: 2474 for Vehicle: 2020
Honda Accord & Coverage Code: VA00063 (2.8s)
Expected Premium: 725 does not equal actual Premium: 2474 for Vehicle: 2020
Honda Accord & Coverage Code: VA00063
```

Failed in 2.776s