



Emtech Group, Inc

QMT Release Notes

Version 2.0



QMT

Emtech QMT is a high-performance model-based quality engineering software for validating the insurance value chain. QMT automatically generates test cases and test data along with test scenario execution. By testing the end-to-end process of insurance systems, and the integrations between them, Carriers, Insurtechs and software vendors can drive quality into product launches and eliminate embarrassing errors experienced by distributors and customers post-launch. QMT allows the development team to shift-left by finding defects earlier, thereby reducing the QA cycle. QMT takes the guess work out of test scenario creation, allowing QA to automate end-to-end business flow testing, speeding product launch delivery & driving cost efficiency.

What's New in Version 2.0.0

QMT Version 2.0 features new capabilities to efficiently model and manage the key interactions of the testing process. Model building in QMT Version 2.0 is made more powerful and efficient through a focus on reusability, “intra-model” data sharing, simplified handling of complex operators in conditional flows and pioneering Dynamic XPath.

Reusability

The **Object Repository** is a centralized hub for managing and organizing reusable project objects. An Object acts as a unique identifier - or Object Locator - for web elements. This facilitates their reuse not just within a single model but also across multiple models within the same project.

The **Vault** is used to securely store reusable values such as username and password combinations.

Intra-model data sharing

/obj

Objects in the **Object Repository** act as a unique identifier for web elements. To directly reference an object from the object repository, a reference to the object can be entered into an XPath field using the following syntax:

Example:

'/[obj]=[Project_name/Category_name/Object_name]'

/obj=Emtech Life Insurance/Underwriting_app/txt_income

This creates an association between the object and the nodes in which it is used. Changes to the object in the repository automatically propagate to all nodes that are associated with the object. An object reference can be copied

from the **Object Repository** by selecting the object, Right-click on it, select **Copy Reference** and paste into an XPath field.

/i

The **“/i”** reference is used to reference node inputs, that is, any string that was directly entered into a field on a node. Inputs are ‘known’ at test generation time. All nodes are able to use **“/i”** to reference a previous node’s input. References to the outputs of nodes are handled through **“/n”** references. The auto-complete feature in QMT helps users to identify whether a value is an output **“/n”** or an input **“/i”**.

Example: /i=Open URL : url

/n

The **“/n”** reference is used to reference node outputs, that is, any value that was generated or created when a node was executed. Outputs are ‘known’ at test execution time. All nodes are able to use **“/n”** to reference a previous node’s output.

Example: /n=City : city

/vf

The **“/vf”** reference is used to reference secured values from the Vault such as username and password combinations.

Example: /vf=EAppLogin:value

Conditional Flows

Business processes with conditional or reflexive flows can be modeled with QMT’s Enable IF node. Enable IF groups conditional nodes in the model and manages the flow to these nodes based on the result, True or False, of an evaluated criterion.

Dynamic XPath

Dynamic XPath increase reusability of complex XPath within a model by using a variable tag in the XPath format and replacing the tag with a value at runtime. QMT’s Checkbox, Radio button, and Dropdown nodes support the use of Dynamic XPath.

Dynamic XPath format uses mustache formatting **“{{value}}”** tags to indicate the location(s) in the XPath where value replacement will occur at runtime.

Example: Dynamic XPath format: `//span[text()='{{value}}']`

Sample **Value:** 'Male'

Result at execution: "`{{value}}`" is replaced by the **Value** "Male":

`//span[text()='Male']`

Test Data Generator

Test Data Generator embeds automatically generated, business relevant test data into models. QMT supports data generation for 16 key data inputs. Users can tailor the data generated for each data input by specifying parameters within their corresponding generate nodes.

Example: Generate Date of Birth

Parameters: Youngest Age, Oldest Age, Date Format

Test Case Generator

QMT's test case generation uses Emtech's proprietary Minimum and Maximum algorithms to provide the flexibility to build all combinations of test paths or construct all essential paths at least once for a modelled business process.

Known Issues:

Closing the application

- Pressing escape while running test Generation will immediately close the application. It will not save any work in progress and will not ask the user to confirm that they want to close the application.
- Closing the application using Close Project will immediately close the application and the Project Home Page will be displayed. It will not save any work in progress and will not ask the user to confirm that they want to close the application.
- Closing the application using X or any other option (beside close project and the above escape) will ask the user to confirm that they want to close the application, if execution is running, the user will then be asked if they want to stop execution, if they say no, execution will continue but the editor will close. If the user says yes, then execution will stop gracefully (writing the results of the most recently completed test case) and QMT will close.
- Closing the application using X or any other option (beside close project and the above escape) will, if the user chooses to close the application also close the Project Home Page. The user must restart the application in order to open a different project.

Sleep/Suspend Behaviour

- If the editor is left idle for a significant amount of time and the computer is allowed to sleep/suspend, then the editor will be slow to refresh itself when the computer is woken. It may also not respond correctly to user input. If this happens, it is best to close the application. The user may have to force quit the application via the Task Manager.
- Test execution cannot proceed while a computer sleeps or is suspended as QMT needs to access the network and interact with external websites. External web sites may also time out or data transfer may be corrupted due to the sleep/suspend cycle. This is not a bug or known issue. Adjust your sleep/suspend settings so that the computer will not sleep during test execution.

Test Execution

- If the user closes the browser during test execution, QMT will hang.
- If the user clicks on the cancel execution or the x button on the execution while test execution is shutting down, the confirmation dialog box is presented, and the user will have to click OK again.
- In some cases, test execution completes and the report has been created successfully, but the console indicates that it is waiting for report to be written. The work around is to click stop execution (or the X); if the confirmation window does not appear, then test execution has completed. If the confirmation window appears, click OK, wait a short while (it should be less than a minute) and try again.
- If the user uses the Advanced Settings to select specific test cases to execute, report generation will fail if test case 1 is not included in the list of test cases. The workaround is to modify the query to include the test case with "tc_id=1"

Nodes

- If the formula in an Expected Condition/Expected result node starts with a space, it may be treated as a syntax error
- Scrolling screenshot may not work depending on the user's file access permissions due to an issue storing the temporary images. The work around is to use the other screenshot types.
- Screenshots behave differently on different browsers and web pages. Three screenshot types are offered so that the user can choose the screenshot that produces the best image.
- The Enable If group will allow one to reference, as the Value to Check or the Value Check, the value of a node inside the group. Such a reference should not be allowed and the option will be removed. Supplying such

a reference may result in a test suite with inappropriate test coverage. Avoid referencing nodes that are contained in an Enable If group from the Enable If's Value To Check or Value Check parameters.