QMT Release Notes

Version 2.3

QMT Version 2.3 Release Notes

QMT

Emtech's QMT is a high-performance model-based quality engineering platform for validating complex enterprise systems and business processes. QMT supports structured model creation, deterministic test generation, automated execution, reporting, and system intelligence to help teams identify defects earlier, improve test coverage, accelerate quality engineering activities, and reduce manual effort.

What's New in Version 2.3

QMT Version 2.3 is a major capability release that introduces several new areas of functionality to QMT. The release introduces Assisted Model Generation (AMG), Mainframe/Legacy Application Testing, QMT Oracle, Custom Test Cases, enhanced test generation with additional algorithms and automatic algorithm recommendation, and parallel test execution. Together, these capabilities broaden QMT beyond traditional manual model creation and sequential test execution, while preserving deterministic model-based test generation as the foundation of the platform.

New Capabilities

Assisted Model Generation (AMG)

QMT 2.3 introduces **Assisted Model Generation (AMG)** as a new capability for accelerating the creation of QMT models while users interact with an application. Instead of requiring users to create every model element manually, AMG assists with mapping application interactions and converting them into structured QMT model nodes and connections.

Assisted Web AMG

For web applications, Assisted AMG supports guided mapping of user interactions into QMT model segments.

Key capabilities include:

- Mapping web application interactions and converting them into QMT model segments.
- Selecting application elements during mapping to identify and create the appropriate QMT action or input node.
- Generating locators used by QMT for later automated execution.
- Supporting iterative model construction so users can continue adding new segments to an existing model.
- Support for conditional paths using **Flow If**.
- LLM-assisted processing where configured through QMT Global Configuration.

Playback

As part of Assisted Web AMG, QMT 2.3 introduces **Playback** to help users continue mapping from a specific point in an application flow without manually navigating back through every preceding step.

- Playback re-executes an already mapped path to return the application to the required state.
- The user can select the point from which mapping should continue.
- Once QMT reaches that point, the user can map a new segment or a different conditional branch.
- This is particularly useful when the same application path leads to multiple branches that need to be mapped into the QMT model.

Mainframe / Legacy Application Testing

QMT 2.3 introduces **Mainframe / Legacy Application Testing** as a new capability for modeling and testing terminal-based applications. This adds support for Telnet/ANSI/VT-style systems and brings legacy application workflows into the same QMT model-based testing approach used for modern applications.

Capabilities include:

- Manual creation of QMT models for terminal/mainframe application flows.
- Creating and opening Telnet terminal sessions from QMT.
- **Assisted Terminal AMG** for mapping terminal interactions and accelerating model creation.
- Mapping entered text and terminal keystrokes as part of the application flow.
- Selecting terminal elements while building a model.
- Creating terminal model segments from mapped interactions.
- Defining **Expected Condition** and **Expected Result** validation steps.
- Supporting present/absent text validations.
- Using **Flow If** to represent menu selections and conditional terminal paths.
- Executing terminal/mainframe tests in headed or headless mode.

QMT Oracle

QMT 2.3 introduces **QMT Oracle** as a new capability for interacting with and analyzing the structured information represented by QMT models and related execution results. QMT Oracle provides a conversational intelligence layer over the QMT Digital Blueprint, allowing users to ask questions and obtain insights about the represented system.

QMT Oracle capabilities include:

- Model insights and model exploration.
- Execution-result analysis.
- Business rule extraction and explanation.
- Data usage analysis.
- Path and coverage analysis.
- Dependency analysis.
- Change-impact analysis.
- Requirement traceability support where requirement information is available in the model or related artifacts.
- Conversation history so users can review previous questions and Oracle interactions.
- User-configurable LLM API credentials through QMT Global Configuration.
- Selection among LLM models validated by Emtech engineering.
- Configurable reasoning level and service level where supported by the selected provider/model.
- LLM usage visibility, including token consumption and estimated usage cost.

QMT Oracle operates on QMT model and execution information; it is not used as the mechanism for mapping application elements during model construction.

Custom Test Cases

QMT 2.3 introduces Custom Test Cases for users who need to create specific deterministic test scenarios in addition to automatically generated coverage-based test suites.

Custom Test Cases are available under **Test Generation > Custom Test Cases** and support:

- Creating named custom test suites associated with a selected QMT model.
- Creating individual custom test cases as columns within a suite.
- Assigning a unique name to each custom test case.
- Adding model nodes and selecting the exact values that must be covered by the test.
- Selecting a model node only once within the same custom test case.
- Defining required **Flow If** branches so only the selected conditional path is generated.
- Supporting end nodes and conditional branches when defining the desired scenario.
- Detecting duplicate custom test cases based on the complete scenario definition and warning the user when an equivalent test already exists.
- Importing and exporting custom test suites.
- Managing and deleting custom test cases.

- Automatically starting deterministic test generation after the custom scenario is defined.
- Keeping custom test cases associated with the model so model changes can be reflected in the custom test definition.
- Preserving custom test case names in execution status and reporting.

Default Values for Model Nodes

QMT 2.3 adds model-level default values that can be reused during test design and Custom Test Case creation.

- Option-type nodes can contain one or more default values.
- Text/data nodes can contain one or more static default values.
- Users can generate using only configured defaults or include all available permutations.
- Defaults are stored at model level.

Test Generation Enhancements

Expanded Test Generation Algorithms

QMT 2.3 supports the following test generation algorithms:

- **Minimum Resolution**
- **Maximum Resolution**
- **Cached Min Coverage**
- **Graph Coverage**
- **CP-SAT**

These algorithms allow users to select different generation strategies depending on the desired coverage, model characteristics, and optimization objective.

Auto Algorithm Recommendation

A new **Auto** option analyzes model characteristics and recommends the most appropriate optimized generation algorithm.

- QMT can select the recommended algorithm automatically.
- The Auto option can present only the optimized algorithms considered appropriate for the current model.
- Depending on the model, Auto may identify a single preferred algorithm.

Advanced Test Generation Settings

QMT 2.3 adds or expands advanced generation settings, including:

- **Only Enabled Options** to restrict generation to enabled values/options where applicable.

- **CP-SAT Time Limit** to control the maximum optimization time used by CP-SAT generation.

Test Execution Enhancements

Parallel Executions

QMT 2.3 introduces parallel test execution, allowing multiple test cases to run concurrently.

- Maximum concurrency is configured under **Test Execution > Advanced Settings**.
- The appropriate value depends on CPU, memory, browser/application behavior, and the environment running QMT.
- A modern laptop may often support approximately 20–25 concurrent executions, but the effective value should be determined for each environment.

Terminal Execution

As part of the new Mainframe / Legacy Application Testing capability, QMT 2.3 supports terminal execution in both:

- **Headed execution**, where the terminal session is visible.
- **Headless execution**, where terminal tests execute without displaying the session UI.

Model Engineering and Locator Enhancements

XPath and CSS Locator Support

QMT 2.3 expands locator handling for web model nodes with support for:

- XPath locators.
- CSS selectors.

iFrame and Shadow DOM Support

QMT 2.3 improves interaction with modern web applications by supporting elements located within:

- iFrames.
- Shadow DOM structures.

Conditional Modeling

Flow If

Flow If remains the supported mechanism for representing conditional branches in QMT models and is used by model creation, Custom Test Cases, and terminal/mainframe mapping.

Global Configuration Enhancements

QMT 2.3 expands **Edit > Global Config** to support AI-enabled QMT capabilities.

Configuration may include:

- User-provided LLM API credentials.
- Selection of a validated LLM model.
- Reasoning level.
- Service level.
- Other provider-specific settings supported and validated by Emtech.

Validation and Specialized Testing

TruePDF and TrueXML

QMT 2.3 continues to provide TruePDF and TrueXML capabilities as part of the QMT platform experience, allowing teams to include document and XML validation within broader automated business-process testing.

CLI Enhancements

QMT 2.3 adds support for configuring parallel execution from the command line:

```
--parallel-executions <int>
```

- Default value: `1`
- The value defines the maximum number of test cases that QMT may execute concurrently for that run.

Migration / Compatibility Notes

Assisted Model Generation

Models created through Assisted AMG remain regular QMT models after creation. Users can review, edit, extend, generate tests from, and execute those models using the same QMT model-based workflow used for manually created models.

Parallel Execution Capacity

Existing projects can continue using sequential execution. Parallel execution is optional and should be configured based on the available machine/server capacity and the behavior of the system under test.

Beta Capabilities

Autonomous Web Model Generation (Beta)

QMT 2.3 makes **Autonomous Web Model Generation** available as a **beta capability**. This capability is still in the development phase and is not yet considered production-ready, but it is available for users to try, evaluate, and explore its potential.

Autonomous Web Model Generation is intended to reduce the amount of direct user involvement required during application mapping by allowing QMT to explore a web application and incrementally build model segments.

Current beta capabilities may include:

- Definition of an application entry point and mapping instructions.
- AI-assisted autonomous exploration of the application.
- Incremental construction of QMT model segments.
- Model segment merge and node matching.
- Integration with QMT Oracle.
- Live mapping visualization where available.

Because the capability remains under active development, behavior and functionality may continue to evolve based on engineering progress and user feedback.