TESTINGWHIZ
by Cygnet.One

RELEASE NOTES VERSION 9.1.5

New Features

1. Background Execution for TestingWhiz Executor
    - The TestingWhiz executor has been upgraded to support **Background Execution**. This enhancement allows the executor to run seamlessly in the background, enabling users to maintain productivity and perform other tasks on their workstations without interrupting the automated execution process.
2. Intelligent CI/CD Pipeline: Selective Test Re-Execution
    - We have introduced **Selective Failed Test Re-Execution** for Azure CI/CD pipelines. This feature allows for the targeted retry of failed test cases with configurable retry limits and isolated reporting.
    - **Impact:** Dramatically reduces pipeline execution time and resource consumption while effectively isolating flaky tests for faster developer feedback.
3. Advanced Auto-Healing Enhancements
    - The Auto-Healing engine has been significantly upgraded across the entire ecosystem (Tool, Executor, and Scheduler).
    - **Dynamic Property Updates:** Automatically identifies and updates shifting object properties including IDs, Names, cssPaths, and XPaths.
    - **Visual Validation:** Healed objects are now highlighted directly on the web interface during execution.
    - **Enhanced Traceability:** Detailed logs of all healing actions are now captured in the GUI Console and final reports.

Improvements & Enhancements

1. SAP Recorder & Build Optimization
    - We have successfully **decoupled the SAP Recorder from third-party dependencies**. By internalizing this logic, we have improved the overall stability of the build and reduced the external footprint of the application. Build data and IDs have been synchronized to reflect this architecture update.
2. IE Mode Support via Edge Chromium

- To support legacy enterprise applications, we have optimized test execution to run within **Internet Explorer (IE) Mode** inside the Edge Chromium browser. This ensures continued compatibility for IE-dependent scripts while leveraging the performance and security of a modern browser environment.
- **Enhanced Configuration:** Users can now manually define the **Edge Chromium** and **IE Driver** paths directly within the whiz.ini configuration file.

3. Smarter SAP script execution
    - SAP script execution is now more **resilient, with improved handling of locator changes** and configuration flexibility.
    - SAP script execution has been validated to remain stable across key scenarios. Scripts continue to **run successfully even when locator numbers are changed or entered incorrectly**, when required Element Type and Object Class configurations are defined in the properties file, and when objects are referenced through variables or data tables.
4. Expanded SAP Interaction Support
    - The SAP keypress library now includes **Tab Key support**. This allows for more natural navigation and precise interaction within SAP windows, mirroring manual user workflows.

Bugs Fixes (Critical/High)

1. **Variable Initialization:** Resolved a critical initialization issue where the default value for **Set Variable Length** was not correctly set upon new installations. This fix ensures that any command action requiring variable configuration functions correctly without manual intervention.
2. **Browser Driver Management:** Fixed an issue preventing the **Edge Web Driver** from updating automatically. Furthermore, we resolved a conflict that caused scripts to hang on the driver update message, ensuring stable execution on Edge Chromium.
3. **Configuration Integrity:** Addressed a race condition that caused the **".whiz" folder** to become corrupted in multi-configuration (e.g., database, SMTP, etc.) Environments, ensuring system stability during startup.
4. **Unified SAP Click Logic:** Resolved multiple issues where **Click and Double-Click actions** failed to execute on target objects, list items (e.g., Transaction MD04), and dialog controls (buttons/list boxes). The engine now ensures verified interaction before proceeding to the next step.

5. **Object Visibility & Timing:** Fixed a **401-timeout error** affecting the "Wait for Element" command. The tool now accurately detects object visibility in the Object Repository, eliminating false negatives for visible elements.
6. **JSON Data Integration:** Re-engineered the **JSON import process** to ensure data integrity and a seamless transition of external data sets into the tool.
7. **Reporting Accuracy:** Resolved a rendering issue where **HTML reports** appeared blank despite containing valid data. This ensures high-fidelity reporting even in environments with multiple tool instances.
8. **Backup Reliability:** Fixed the backup utility to ensure it generates **multiple rotating backup files** as configured, rather than overwriting a single file. Additionally, backup **TWIZX files** now correctly inherit self-healing properties from their source files.
9. **Enhanced Connectivity Validation:** Optimized the reachability detection logic for **URL-Reachable commands**. This ensures that actions such as *Verify, Validate, and Check* accurately report the status of accessible websites.
10. **Extension Compatibility:** Restored seamless compatibility between the **TestingWhiz Chrome Extension** and the core application, ensuring users can leverage the latest web store updates without disruption.
11. **Execution Consistency:** Resolved a "NOT RUN" status bug that caused the scheduler to skip test cases during re-runs. This ensures that the execution sequence follows the user-defined configuration without skipping logs or suites.
12. Resolved an issue where the **Enter Authentication command failed to authenticate users**, preventing login across multiple websites even after providing valid **URL, username, and password**. This release ensures **successful authentication and reliable login behavior** when using the Enter Authentication command in the Test Editor.