

TestArchitect™ 8.3

Update 2
August 31st 2017

Release Notes for macOS

LogiGear Corporation

4100 E 3rd Ave - Suite 150
Foster City, CA, 94404

Phone (650) 572-1400

(800) 322-0333

Fax (650) 572-2822

Web www.logigear.com

Contents

What's New in TestArchitect 8.3 Update 2.....	3
New features.....	3
Enhancements to existing features	3
TestArchitect Automation.....	4
New features.....	4
Enhancements to existing features	5
System Requirements	5
Recommended system requirements.....	5
Additional Resources	6

Please note:

- Before upgrading to the latest version of TestArchitect, it is strongly recommended that you back up all repositories.
- A TestArchitect client and any repository server to which it connects must be of the same version.

What's New in TestArchitect 8.3 Update 2

The following new features and improvements are introduced in the latest version of TestArchitect running under macOS.

New features

- **[Modern TestArchitect command-line interface:](#)**
 - **Flexibility:** With new TestArchitect's command line tool, tests can be launched from a command shell. When incorporated into batch files, the tool significantly extends the flexibility of test execution, especially from a scheduling standpoint.
 - **Standards for command-line interfaces:** In order to provide a consistent, cohesive user experience, the new TestArchitect command-line interface has been standardized on the following aspects:
 - Consistent argument naming conventions
 - Consistent error messages and exit codes
 - Consistent command syntax
 - **Ease of use:** The presence of optional arguments in batches files is no longer required. When the optional arguments are absent, their default value will be automatically applied.
- **[A separate TestArchitect Help package:](#)** The TestArchitect Help package has been removed from TestArchitect installer. If you would like to continue to open local help documentation in offline mode, please follow this [link](#).

Enhancements to existing features

- **[Adding test results to the repository:](#)**
 - In the CI/CD process, after automated tests are entirely finished, you are able to automatically add local test results into the repository, even if the specified result folder might not exist beforehand.
 - At the moment, TestArchitect is capable of dynamically creating result folders on-the-fly. Name of the result folders is determined based on the following predefined macro variables.

Macro variables	Description	Output (example)
{year}	Gets the current year	2017
{month}	Gets the current month	08
{day}	Get the current date	09
{today}	Get today's date in the form of <i>mm-dd-yyyy</i>	08-09-2017

- **Exporting test results into various formats:**
 - **Export results into .TAREXPORT format:**
 - Test run results now can be converted and [exported to .TAREXPORT](#), an archive filename extension developed by LogiGear. This new extension is extremely convenient for test result query purposes, especially in the CI/CD process. Specifically, upon test completion, you can aggregate all test results quickly by just simply querying all returned .TAREXPORT files.
 - .TAREXPORT then also can be [imported](#) back into TestArchitect.
 - **Export results into HTML format:**
 - [XSLT template](#): Ability to export HTML results with your own view layout, customized by your XSLT template.
 - [Directory structure output](#): Ability to output different types of directory structure folder which store HTML files.
 - **Folder structure**: TestArchitect creates a hierarchical tree structure, or subdirectories to store HTML results.
 - **Flat structure**: TestArchitect does not create subdirectories. There is only a single top-level directory which contains all HTML results.
 - **Create output folders on-the-fly**: TestArchitect automatically creates a folder to store HTML files if the specified folder does not exist.

TestArchitect Automation

New features

- **Testing in cloud services:**
 - **Official release for automation testing in [WebDriver based cloud services](#)**: It delivers cost saving, time saving, remote/mobile access to information, high productivity, and increased test coverage across different mobile device types, etc.
 - **Execution on multiple cloud devices**: You are now able to automate tests on multiple cloud devices concurrently. The cloud devices will be defined via desired capabilities, specified in [JSON files](#).
 - New built-in [disconnect all cloud devices](#) action added: Logically disconnect all currently-connected cloud devices.
 - [web element matching](#) built-in setting: Specify which technique is used to perform web UI element matching.
- **[Web service testing](#)**:
 - New built-in actions supported:
 - [configure web service settings](#): Send JSON string request to a web service in order to apply new configuration settings.
 - **Proxy**: Defines a manual proxy server.
 - **SSL**: Defines SSL configurations.
 - **Cookies**: Specifies HTTP cookies that are sent back to the server.
 - **Encoder**: Specifies configurations for the encoder.

- **Decoder:** Specifies configurations for the decoder.
 - **Redirect:** Configures the redirect settings
 - [attach file](#): Attach given file(s) to a web service.
- **Common HTTP request methods are fully supported** for the [send http request](#) action including:
 - [GET](#): Request data from a specified resource.
 - [POST](#): Submit data to be processed to a specified resource.
 - [PUT](#): Replace all the current representations of the target resource with the uploaded content.
 - [DELETE](#): Remove all the current representations of the target resource given by URI.
 - [PATCH](#): The PATCH method is used to apply partial modifications to a resource.
 - [HEAD](#): The HEAD method asks for a response identical to that of a GET request, but without the response body.
 - [OPTIONS](#): The OPTIONS method is used to describe the communication options for the target resource.
- A new method of **AbtEntity** for a user-scripted action (applicable to Java and C# harnesses only):
 - [ExecuteScript](#): Executes JavaScript code of a specified web page, or inline JavaScript.

Enhancements to existing features

- **iOS testing:**
 - **Support for [iOS 10.3.2](#):** You can now run tests for applications on iOS devices running this iOS version.

System Requirements

The following table lists the recommended hardware and software requirements for TestArchitect.

Recommended system requirements

	TestArchitect Client/Controller with small-scale Repository Server	Production-scale Repository Server (up to 15 concurrent connections, 1GB repository)
Operating system	El Capitan (10.11), Yosemite (10.10) (Learn more)	Windows 2008/2012 Server
RAM	4 GB	8 GB
HDD	At least 2.5GB available space	Available space of at least 2 GB plus total aggregate size of all databases
CPU	2 GHz dual-core or faster	3.1 GHz quad-core

Additional Resources

For more information, please visit our website at www.testarchitect.com or email us at sales@logigear.com.