

 **TestArchitect™ 8.3**

February 15<sup>th</sup> 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](http://www.logigear.com)

## Contents

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

**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

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

## TestArchitect Client

### New features

- **Introduction of new [XML test result model](#):**
  - TestArchitect saves test results of each test run in the XML format, so that you can view them in any web browser.  
XML result model provides the following advantages when compared to the previous result model.
    - **Less memory footprint:** The amount of memory that TestArchitect uses or references while opening and viewing test results is less.
    - **Smaller parsing times:** The operation process is faster, such as open test results, import/export test results, etc.
    - **Smaller test result compression:** Test results take up less hard drive space; therefore, the repository's size is reduced.
  - **[Test result customization](#):** You now can customize how your test results appear by modifying the corresponding XSL file, a language for transforming XML test results.
  - **[XML Result Conversion tool](#):** Rather than [converting one-by-one test results](#) into XML format, you can perform this operation via a command line tool to quickly convert entire test results of a given repository.
  - Numerous bug fixes.
- **[Better troubleshooting of automation errors](#):** If there are any automation errors in the test result, TestArchitect provides a hyperlink to direct to the respective online troubleshooting topic which describes common causes and possible solutions.

### Enhancements to existing features

- **Improved picture checks:**
  - **[Importing external image sources](#):** TestArchitect now lets you easily import image files (i.e. .PNG format) for new picture checks or additional new baseline images.
  - **[Resolving unverified picture checks](#):** You are now able to upload unverified picture checks to the repository, and then resolve those unverified picture checks.

## TestArchitect Automation

### New features

- **[Support for web service testing](#)**: Web services are applications that can be published and called over the Internet by client applications. It is vital to perform testing for web services during the pre-deployment stage. This helps to identify and resolve defects as early as possible. Also, in the regression testing during post-deployment stage it ensures ongoing quality of service and availability. TestArchitect now supports functional testing of web services.
  - TestArchitect includes new built-in actions that let you perform web service testing:
    - **[create http request](#)**: Instantiate a new HTTP request.
    - **[authenticate](#)**: Specify an HTTP authenticate scheme and its information before starting to work with web services. The user must be authenticated to the services.
    - **[add http header](#)**: Add a name/value pair as an HTTP header field that you might want to include with your request.
    - **[add http body](#)**: Add a message body into an HTTP request.
    - **[add http parameter](#)**: Add a parameter including a pair of name/value to a GET request. Note that, this parameter will be added to the query string in the requester URL.
    - **[send http request](#)**: Send an HTTP request to a given resource.
    - **[parse http response](#)**: Parse the response of the request to status, header and body.
  - TestArchitect also enables you to test web services that use **Basic HTTP authentication** and **OAuth 2.0 authentication**.
- **Built-in actions for [JSON checkpoints](#) supported:**

**Note:** These built-in actions can also be used to verify JSON response returned by the tested web service.

  - **[check json query result](#)**: Check a JSON query result against an expected string. The result is **Passed** if the strings match; otherwise it has **Failed**.
  - **[get json query result](#)**: Query a JSON document by JSONPath.
- **Added beta support for automation testing in [WebDriver-based cloud services \(Appium enabled\)](#)**: It delivers cost saving, time saving, high productivity, and increased test coverage across different mobile device types, etc.
  - With TestArchitect, you can now run mobile web app testing on real devices in WebDriver based cloud services.
  - TestArchitect includes new built-in actions that let you test in the cloud services.
    - **[assign cloud device](#)**: Assign a logical name for a given capability of a target cloud device. This name may then be used in the action **connect cloud device**.

- [connect cloud device](#): Logically connect to a cloud-based device and make that device the target for the automation.
- [use cloud device](#): Use a cloud device, as specified by its session ID, to run automated tests.
- [get cloud device session id](#): Retrieve session ID of the connected cloud device.
- [disconnect cloud device](#): Logically disconnect the target cloud device.
- [use host machine](#): After running automated tests on cloud services, switch back to the host machine to continue to run the remaining tests.

## System Requirements

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

### Recommended system requirements

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

## Additional Resources

For more information, please visit our website at [www.testarchitect.com](http://www.testarchitect.com) or email us at [sales@logigear.com](mailto:sales@logigear.com).