Release Notes for Linux

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

Version 8.2 | August 2015

Contents

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 8.2

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

TestArchitect Client

- **Time-traveling Execution**
  Time-traveling execution provides support for retroactive execution, that is, the ability to execute a particular historical snapshot, or set of revisions, of project items in a given project. Given its ability to find the right set of test assets for a particular past version of an application, time-traveling execution is well-suited for regression testing on AUT versions that precede the current release.

  Note:
  - Implementation of time-traveling execution has necessitated structural changes to TestArchitect repository databases. An upgrade tool is provided to allow you to take advantage of time traveling on repositories created prior to TestArchitect version 8.2.
  - However, if your current databases belong to TestArchitect version 6.4 or earlier, you must perform the upgrade in stages:
    1. If your TestArchitect version is 6.4 or earlier, upgrade TestArchitect to any version between 7.3 and 8.1.
    2. From a TestArchitect 7.3-8.1 installation, run the repository upgrade tool. This upgrades your repository databases to support name change propagation, a prerequisite for time traveling support.
    3. Upgrade to TestArchitect version 8.2
    4. Run the 8.2 repository upgrade tool. This upgrade provides time traveling support to your databases.

- **Support for Time zones**
  To help teams spread across different time zones, TestArchitect can now automatically display date and time values for project assets relative to the time zone in which they are being viewed. As an example, during the months of Daylight Savings Time, Vietnam is 14 hours ahead of the U.S. Pacific Time Zone. Hence, if a team in California checks in a project item at 3:00pm on January 1, Vietnamese engineers see the item as having a **Last update date** of 5:00am, January 2.

Enhancements to existing features

- **Support for creating a branch in variations**
  Branches are most typically used to represent the test changes implemented to accommodate series of patches or hot fixes, usually involving such things as bug fixes or performance enhancements for a given version of an AUT. Note that you may also create subtrees of version nodes under branch nodes.

- **Support for picture checks in project subscription**
  Picture checks stored within the **Picture Checks** node can now be shared across multiple projects via subscription. In other words, your project can reuse picture checks that reside in other projects.

- **Enhanced support for Revision Control**
  TestArchitect Revision Control saves pertinent information on a project item every time it is checked in. Each revision is now identified by the following information:
  - timestamp (that is, the checked-in date and time)
  - assigned AUT versions
  - host project name
  - project item type
  - project item name
  - user who checked in the item
  - check-in comment.

- **New UI for the *VERSION HISTORY* dialog box.**

- **Support for Context-sensitive help of built-in settings**
  Positioning your cell pointer over a built-in setting in the test editor and pressing `F1` opens TestArchitect Help with the associated reference topic displayed.

TestArchitect Automation

- **Introduction of XPath property**
  TestArchitect uses a special new TA property, called **XPath** and derived from the XPath language, to uniquely locate UI controls within tested web applications. Specifically, TestArchitect depicts the UI controls running on web applications as a hierarchical representation of elements and provides the **XPath** TA property for each control to uniquely identify it within that hierarchy.

- **Support for escape sequences during string processing**
  A new built-in setting, **escape sequences**, allows this functionality to be enabled or turned off.

Known Issues

- TestArchitect captures a web element based on the **id** attribute, and sometimes other attributes, and then uses those attributes to form an `XPath` expression, which it applies to the element's **XPath** TA property. If those attributes are non-static, it might not be possible to locate the element during playback. Under such circumstances, the **XPath** TA property is not a good choice for mapping web elements.

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	Red Hat Linux 5.8 and 6.3	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*.
- On a system with TestArchitect installed, help is available at: **/usr/local/logigear/testarchitect/tahelp/webhelp/index.html**