1. CA Lisa Registry

<http://www.ca.com/us/support/ca-support-online/product-content/knowledgebase-articles/tec604116.aspx>

2) Lisa version

3) types of records

4) how to create service virtualization

**iTKO LISA**

**http://www.itko.com/solutions/ui\_web\_ria\_testing.jsp**

LISA can test Internet Application (RIA) interfaces, all the way through to the services, content and implementation layers that support them for complete test coverage.

iTKO LISA supports both the creation and staging of Web Services tests.

Lisa separates test cases from staging documents.

Test cases define which components are tested.

Staging documents define how those tests are run.

The staging documents can be reused across the project and allowed us to define scenarios with different numbers of users and periods of time for average and peak loads.

This feature of load testing makes it easier to find bottlenecks by determining each subsystem's performance under various load conditions.

I have exp using Lisa for functional testing.

I used to Create scripts in Lisa Workstation.

For web application I used to record the scripts using HTTP Proxy Recorder and DOM Events Recorder.

I added data sets if required and do parameterization. And added filters to output values

And Assertions to check conditions.

I used ITR (Interactive Test Run) mode to Playback the recorded/created script.

I used to develop Test Suites by group of 2 or more test cases and run parallel by running suite

**Codeless UI Recording and Testing:** Testers can record and playback the entire Web UI testing workflow within a native browser emulation (IE, Firefox, Safari, etc.), and get sustainable test assets out of the process. LISA captures all user actions, from mouse clicks and data entry, to drag-and-drop functions in the web UI, and then provides several ways to leverage that test, and add dynamic data validation points without needing to re-script the test.

**Web 2.0-Ready Testing:** LISA supports direct testing of DTHML/JavaScript, AJAX, Java Applet, Swing, Microsoft ActiveX, Oracle Apps and most commonly known approaches to delivering UI functionality into the browser, in addition to standard testing of HTTP-level traffic, validating and simulating all of the transactions that occur between the web page and the web server.

**Extend Test Cases from UI to Implementation:** Testers no longer need to use separately acquired or developed tools to test the user interfaces, and all of the other implementation layers that feed them. Users can use LISA to model a single test case with steps that invoke and verify web UI behavior and performance, as well as validating database calls, SOAP messages and many more types of transactions.

**High-Performance UI Testing:** LISA efficiently conducts high-performance user testing of RIA browser applications. LISA uses a shared set of virtual users that can test any technology, leveraging pre-existing functional test cases, and dynamically staging them as high-capacity load tests. Test overhead is very low, with a typical CPU able to stage dozens of browser instances, Java apps or other simulators, and the load can be distributed across multiple machines in a local or remote computing network. Upon staging, load tests can simulate an instance of the browser for each virtual user, or run "headlessly" to simulate only the transactions that occur between objects to conserve test system overhead.

**Synchronized User Simulation:** Unlike other tools that simulate variable user behavior and "think time" only by setting an arbitrary time frame for each successive test step, LISA allows testers to flexibly synchronize suites of tests with the real-world conditions and timing produced by each step of the test case.

**Virtual Service Environment:** Use LISA Virtualize to capture and simulate the needed services and data required for a complete test bed, even if these dependencies are unavailable or incomplete. By using Virtual Services in conjunction with LISA's UI testing capabilities, teams get 24/7 access to all needed components and data, at a fraction of the cost of building or maintaining custom test environments.