

TeamDev

JExplorer Case Study

JExplorer enriched iTKO LISA application with Internet Explorer functionality

"In case you are wondering, why am I trying to get your product to do what I've already gotten other products to do? Answer: Your product is stable! And your support is superb."

John Michelsen, iTKO Inc.

Summary

iTKO is a Dallas-based company designing and building mission-critical enterprise software for world-class clients. They provide their own automated testing solution called LISA designed for a variety of technologies focused on SOA-type applications.

Industry	Software
Tools & Technologies	JExplorer, Sun JDK 1.5
Project Size	129 man-hours

Challenge

Primarily the customer decided to add some additional functionality to LISA using one of our products JExplorer.

iTKO needed a solution to record and restore test cases in their application. This is a typical task for this kind of software. LISA was already using a pure-Java browser component, but needed native rendering performance and features of a real widely used browser, such as Internet Explorer. Having investigated some other alternatives they were impressed with JExplorer and technical support level and decided to use it for their project.

The task was to integrate JExplorer into LISA and to enhance LISA with DOM event level record and playback capabilities as well as provide the customer with the ability to programmatically manage browser cookies. This was a new development as they were currently capturing at the HTTP protocol level and playing back without managing cookies. Having convinced of our technical expertise and from a timing standpoint they decided to outsource the integration and the new development to us.

Solution

For integration of Internet Explorer into Java and for implementing required tasks we used JExplorer—one of our products from the Java Platform Integration Family that allows tight integration of Internet Explorer into Java applications as either a visual or non-visual component. It ensures deep integration with DOM functionality by using a low-level Internet Explorer API which allows to easily implement record/playback capabilities.

Technical Implementation

Full access to the native browser API was provided with JExplorer. Its Browser component provided LISA with an Internet Explorer API to browse the content and the DOM API to work with the DOM of web pages in many ways. In order to listen to and capture browser events (such as navigation, document loading completion or new window) event handlers provided with JExplorer were used.

The customer also needed to control cookies received during the recording of a given test case. For that purpose a new Cookie Management API was added to JExplorer that allowed managing cookies for a user-specified session (create, modify, find all cookies available and delete the chosen ones programmatically, get and set their name, value, domain and other information). Later, we released this API publicly in JExplorer version 1.9.

During the implementation we faced several issues successfully solved by our development team.

LISA uses browser component in 2 different modes, a view-only mode, and a recorder mode. Each time the user was required to use a Browser component there was created a new browser instance. It became a real problem because large test cases could require the creation of a great many browser instances that could cause memory leaks. A shared browser was proposed and successfully implemented in this case to avoid extensive memory usage.

For rigorous testing we needed to set the proxy just for a given instance of the browser. So if the Windows default IE browser is set to no proxies, the JExplorer-driven browser should have customizable proxy settings that do not affect the global proxy registration in Windows. The problem was that you can only set the default system-wide proxy configuration in the registry. You can not set the proxy settings only for a given instance of the Browser without affecting the global proxy registration. In Windows all browsers retrieve proxy settings from one source. The solution we found allowed setting the proxy settings for just a current process without affecting global setting. In the end the system was able to change proxy settings for a given process on-the-fly. At the specified time the system sets the proxy settings for a current process, and at the appropriate time, it sets them back to their previous (global) settings. The API that allows such functionality can be found in JExplorer starting from version 1.8.

Benefits

We successfully implemented all required functionality on time and in accordance with all customer's requirements and specifications. This functionality provided LISA with a possibility to record and replay test cases and ensured full reality of testing. Owing to JExplorer it was done fast and easy.