



HammerDB HTML Interface

This guide gives you an introduction to using HammerDB with HTML interfaces such as middle-tier web applications.

Introduction to HTML Testing	1
Writing an HTML Test	1
Support and Questions	3

Introduction to HTML Testing

HammerDB is focused upon testing Databases and interfacing directly with those databases under test. Ongoing HammerDB development does not focus primarily on testing the web tier. Nevertheless there are circumstances when it is necessary to interact with the middle-tier whilst testing databases and therefore HammerDB includes the [Tclwebtest](#) package “that provides a simple API for issuing http requests, dealing with the result and assume specific response values, while taking care of the details such as redirects and cookies”. For using Tclwebtest the [API page](#) is here.

Writing an HTML Test

For a simple example the following script searches Google for a random choice of words from HammerDB, Oracle, SQL Server, PostgreSQL, MySQL and Redis. Note that new HammerDB scripts should always refer to the TCL engine on the first line and then call the tclwebtest package as shown on line 2. This example then defines the procedure randlist to select a search term at random before using tclwebtest commands to interact with the Google search page. It is a requirement for Google in particular that the browser type be defined and this is done with the command user_agent_id as shown. The script then selects a search term, enters it into the Google search field and submits the form.

```
#!/usr/local/bin/tclsh8.6
package require tclwebtest

proc randlist {list} {
  set ranval [ lindex $list [expr {int(rand()*[expr [llength $list]])}]]
  return $ranval
}

::tclwebtest::init
::tclwebtest::user_agent_id "Custom mozilla"
::tclwebtest::do_request {http://www.google.com}
::tclwebtest::assert text {Google}
::tclwebtest::form find ~n {f}
set searchlist [ split {HammerDB:Oracle: SQL
Server:PostgreSQL:MySQL:Redis} : ]
set searchterm [ randlist $searchlist ]
puts "Searching Google for $searchterm"
::tclwebtest::field fill $searchterm ;# ~n {q} ;# type of field = text
```

```
::tclwebtest::form submit
```

Enter the completed form in the HammerDB script editor as shown in Figure 1.

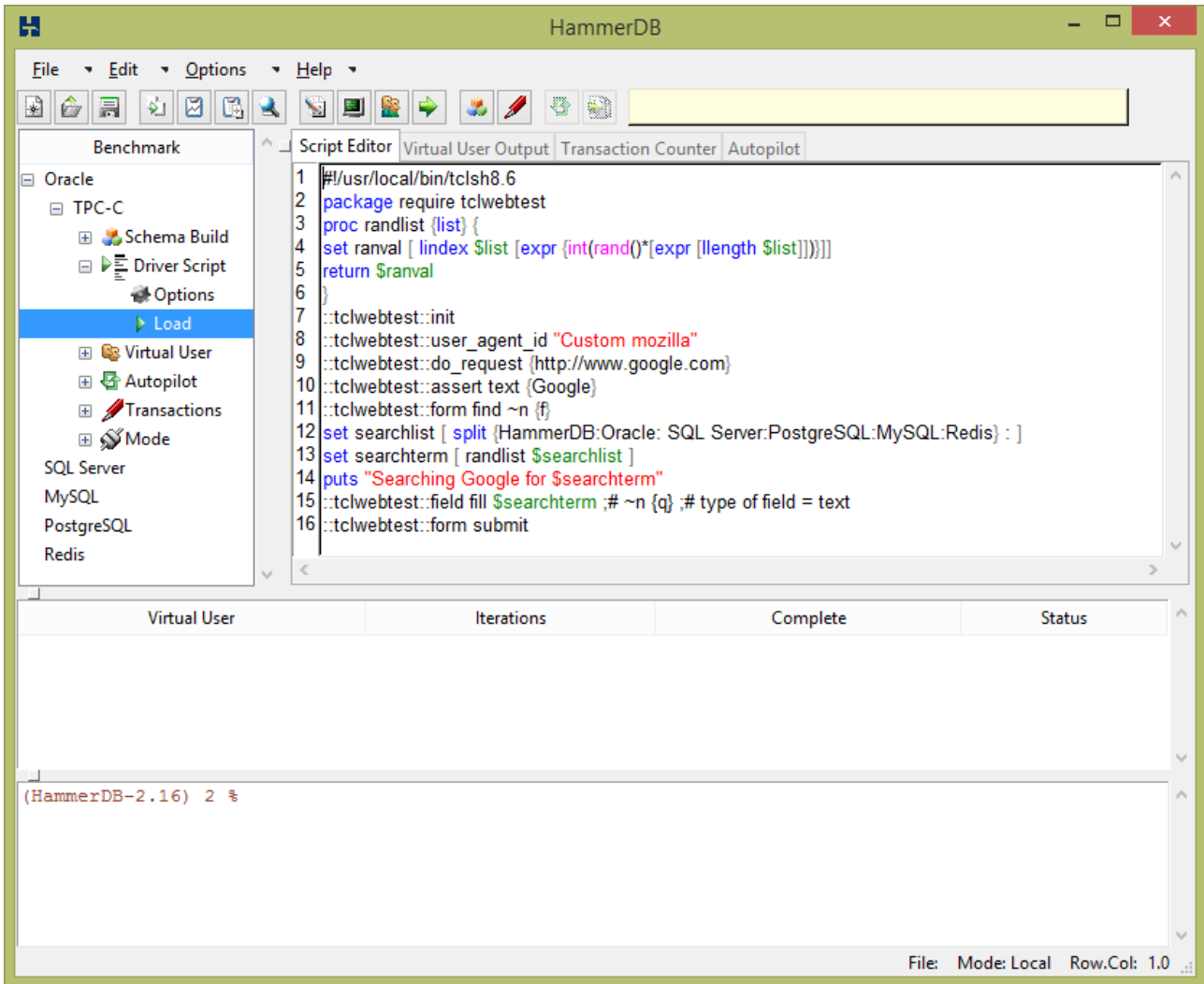


Figure 1 Completed Test Script

Create a number of virtual users and run the test script, you will now be able to replay your recorded web session with multiple users as shown in Figure 2.

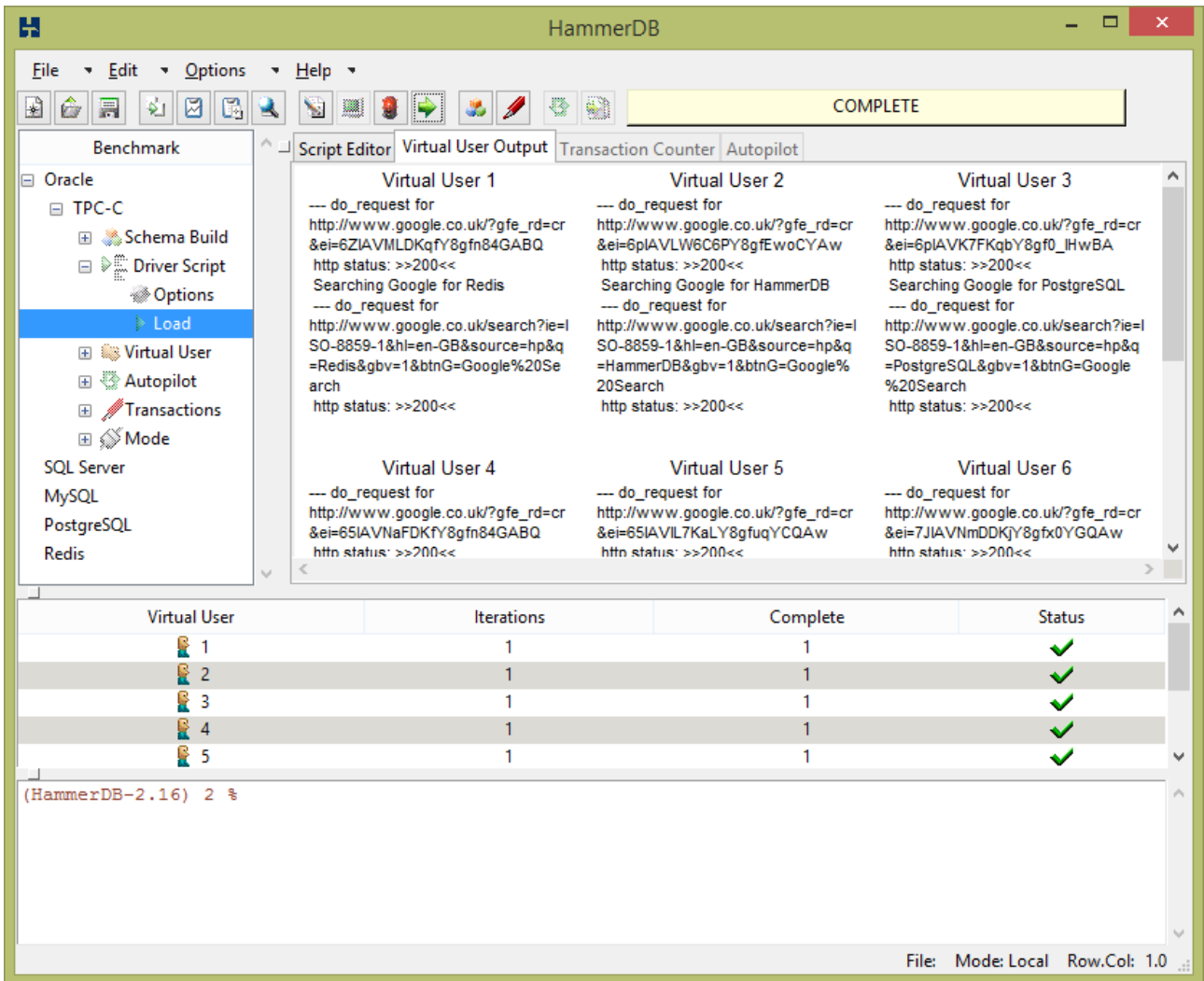


Figure 2 Test Script Run

Use the link to Tclwebtest API to create more sophisticated tests to interact with your web-tier whilst testing database environments.

Support and Questions

For help use the HammerDB Sourceforge forum available at the HammerDB sourceforge project.