AUTOMATING TIME SHIFT TESTING WITH TIME MACHINE AND SELENIUM
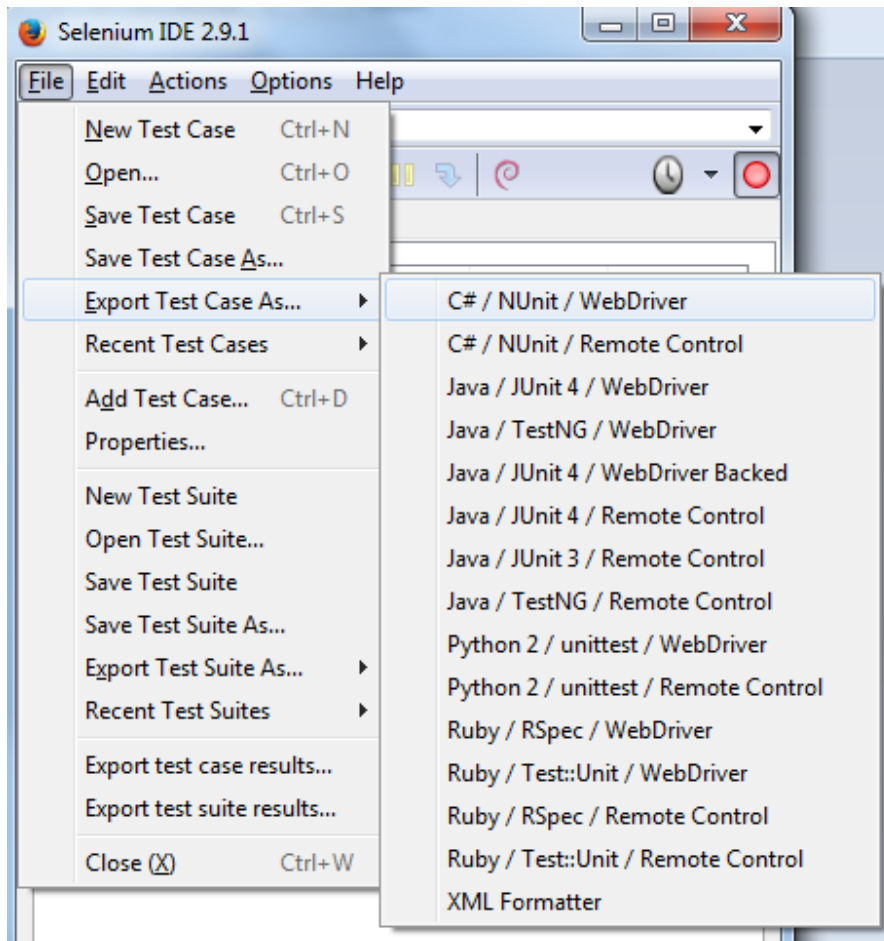
TIME MACHINE® AND TIME MACHINE SYNC SERVER

Time Machine facilitates the creation of software virtual clocks. These virtual clocks enable time shift testing without requiring a change to the system clock on the servers that make up the architecture of a deployed application. The Time Machine Sync Server allows for the broadcast of a virtual clock to specified targets that may be spread across different computing nodes in the enterprise. These groups of related targets are known as Sync Groups. This paper describes how to leverage the Time Machine Sync Server API during test automation using Selenium.

TEST AUTOMATION WITH SELENIUM

Selenium is the most commonly used open source automation framework. Selenium is capable of supporting a variety of programming languages such as Python, Java, C/C++, Ruby as well a variety of browsers such as IE, Safari, Firefox, Chrome. Hence, software developers and QA engineers have more freedom and ability to create sophisticated, data rich automation test scripts that may depend on complex environments.

The test scripts can be generated by the Selenium IDE, a Firefox plugin that allows the tester to record and play back user interactions in the browser. The recorded scripts can be exported to a target language that can then be used for other types of tests like unit testing, stress testing, or performance testing.

AUTOMATING TIME SHIFT TESTING WITH TIME MACHINE AND SELENIUM

HOW TIME MACHINE AND SELENIUM WORK TOGETHER

Selenium and *Selenium Webdriver* scripts provide the backbone for the automated test. These scripts are created to simulate user actions. By calling the Time Machine Sync Server API from the test script, one enhances the capabilities of testing the date and time centric logic across multiple nodes, environments and time zones.

A few examples of a Selenium generated script that has been modified to enable time shift testing of an application is seen below. In the examples, the script is calling the Time Machine Sync Server API to enable and disable the Sync Group. This essentially turns on and off the virtual clock on all targets defined in the Sync Group. The examples also show a new virtual clock being specified at the same time as the Sync Group is being enabled. Using the Selenium Webdriver, the following samples can be executed with any supported browser. Solution-Soft would like to point out that not all applications will continue to execute when time moves backward. This is essentially what happens when you disable the Sync Group, removing the virtual clocks. If this becomes an issue, rather than disable the Sync Group, move on to the next test case that requires a virtual clock that is the same or that moves time forward. Simply calling the enable function and specifying the next virtual clock in the future should suffice.

EXAMPLE 1: Code snippet to enable a Sync Group called "***dbCloud***"

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import Select

class TestSeleniumInfoWeb2(unittest.TestCase):
def setUp(self):
#definition of Webdriver
self.driver = webdriver.Firefox()

def test_selenium_tnfo_web2(self):

driver = self.driver

# Lets enable TM Sync Group called dbCloud
data = {'action': 'EnableSyncGroup','name':'dbCloud'}
ws_url = 'http://SYNCGROUP SERVER:4600/tmSyncServer?'
r=requests.get(ws_url,params=data)
print(r.url)
time.sleep(5)
```

EXAMPLE 2: Code snippet to disable a Sync Group called "***dbCloud***"

```
data = {'action': 'DisableSyncGroup','name':'dbCloud'}
 ws_url = 'http://SYNCGROUP SERVER:4600/tmSyncServer?'
r=requests.get(ws_url,params=data)
print(r.url)
time.sleep(5)
```

EXAMPLE 3: Code snippet enables Sync Group "***DotNetServers***", sets an absolute virtual clock and sets the speed of that virtual clock to twice as fast as the system clock

```
data =
{'action':'EnableSyncGroup','name':' DotNetServers ','abs_time':'070120182130','speed': '2'}
ws_url = 'http://SYNCGROUP SERVER:4600/tmSyncServer?'
r=requests.get(ws_url,params=data)
print(r.url)
time.sleep(5)
```

ADVANTAGES OF AUTOMATION WITH SELENIUM AND TIME MACHINE

Benefits of Automation

Save Costs: Reduced operational costs
Boost Productivity*:* Increased productivity
Improved quality*:* Increased quality of product
Successful Deployments*:* Can produce scalable test scripts on demand, browser independent. Supports multiple programming languages.
Application: Can be used in any type of test (regression, integration, performance, unit testing)

Features & Benefits of Time Machine

Save Costs: Hardware and software maintenance costs
Boost Productivity: Boost engineering team productivity
Successful Deployments: Ensure large scale software projects finish on time and under budget
Mitigate Failures: Mitigate risks for mission-critical application failures
Travel to Past, Future, running or frozen clocks: Virtual clocks can be set to the past or future, running or frozen in time, and specified by absolute or relative values
Variable Speed Clock: Using the Variable Speed Clock function, applications can run up to 1000 times faster or slower than real time.
One System can perform multiple simultaneous tests: Concurrent testing of multiple virtual clocks enable one test environment to become multiple test environments.

Solution-Soft

SolutionSoft Systems, Inc.

2350 Mission College Blvd.

Suite #777

Santa Clara, CA 95054

U.S.A.

Toll Free: 1.888.884.7337

Phone: 1.408.346.1400

Fax: 1.408.346.1499

www.solution-soft.com