



SUGARCRM[®]
THE CLOUD IS OPEN



Soda Introduction

A gentle-ish introduction to Soda



Introduction

- This is just a quick start guide for Soda
 - Supported Platforms
 - Installation
 - Configuration
 - Your First Test
 - Data-Driven Testing
 - Reporting
 - Pitfalls
 - Advanced Topics
 - Resources

Supported Platforms

● Windows

- Recommended – we run XP in production.
- Windows 7 is not tested but should work.
- Ruby 1.8.6 patchlevel 287
- Firefox 3.5+ only – IE support in early beta.

● Linux

- Supported but not recommended – slowness
- Ruby 1.8.7 patchlevel 249
- Firefox 3.5+ only.

● Macintosh – not supported.

Installation – part 1 (Watir)

- Run these commands:

1. `gem update --system`

2. `gem install -r libxml-ruby`

3. `gem install activesupport --version="2.3.8"`

4. On Windows: `gem install watir`

On Linux: `gem install firewatir`

5. `cd C:\Ruby\lib\ruby\gems\1.8\gems\watir-1.6.5\lib\watir`

6. `regsvr32 /s autoitx3.dll`

- On Windows, you may need to install `vcredist_x86.exe`, available from [this page at Microsoft](#).

Installation – part 2 (Firefox)

- Install the appropriate version of JSSH from [this page on the Watir wiki](#).
- Configure Firefox to open popups in new tabs instead of new windows:
 - Type about:config in the FireFox address bar
 - Search for "newwindow"
 - Set browser.link.open_newwindow = 3
 - Set browser.link.open_newwindow.restriction = 0
 - Restart Firefox.

Installation – part 2 (IE)

- Configure IE to open popups in new tabs instead of new windows:
 - Click on Tools->Internet Options.
 - On the General tab, in the Tabs section, click on the Settings button.
 - Check the checkbox called "Always switch to new tabs when they are created".
 - Click the radio button called "Always open pop-ups in a new tab".
 - Click the OK button in the Tabbed Browsing Settings dialog.
 - Click the OK button in the Internet Options dialog.

Installation – part 3 (Soda)

- Browse to www.github.com/sugarcrm/SODA.
- Check out the master branch.

Configuration

- Soda uses a config file which can supply defaults for certain command line parameters. This config file is an XML file which can contain a number of different tags:
 - `<gvar>` tags represent global variables which can be accessed from within a test.
 - `<cmdopt>` tags represent command line options which can be overridden through the file. Currently `–browser` is the only one supported.
 - `<errorskip>` allows certain classes of errors to be disregarded for debugging purposes. Currently “css” is the only type which is supported by `<errorskip>`.

Configuration (cont'd)

- Open soda-config.xml in a text editor.
- Change the contents of the `<gvar name="url">` tag to the URL of the SugarCRM instance you wish to test against. This should be a fresh instance with no demo data installed.
- Change the contents of the `<gvar name="scriptsdir">` tag to the path to the directory where you plan to store your tests. Note that this is relative to the Soda directory.

- None of these are required, but they are recommended for convenience.
- Command line arguments supersede these settings.

Your First Test

- Create a new XML document to begin writing a Soda test!
- All Soda tests must begin with a `<soda>` tag on a line by itself and end with a `</soda>` tag on a line by itself.
- Inside the `<soda>` tags you can place elements which interact with the browser or the web page. The most common actions are clicking on page elements and asserting page contents for verification.
- For starters, we will log into SugarCRM as an existing user and verify that the login was successful.

Your First Test

```
<soda>
  <puts text="Test: login.xml starting." />

  <!-- Navigate to the login page and verify that it loaded. -->
  <browser url="{@global.url}" />
  <browser assert="Please enter your user name and password" />

  <!-- Fill in the login form. -->
  <textfield id="user_name" set="admin" />
  <textfield id="user_password" set="admin" />

  <!-- Click on the login button and verify that we are logged in. -->
  <button id="login_button" click="true" />
  <browser assert="Welcome, " />

  <browser action="close" />

  <puts text="Test: login.xml finished." />
</soda>
```

Your First Test

- Save your XML file in the directory where you will store your tests. In this example, we will assume it is a subdirectory of Soda named scripts.
- Open a command prompt in the SODA directory.
- `ruby SodaSuite.rb --debug --test="scripts/login.xml"`
- Watch your test run!

- Note: the above assumes you are using the soda-config.xml configuration file as described above. If you skipped that step, additional command line options may be required.

Data-Driven Testing

- Soda supports data-driven testing through the use of CSV data files.
- The example below assumes the existence of two users -- be sure to create them before running it.

Your First Data-Driven Test

- login.csv

```
username,password
admin,admin
qauser,QAuser1
```

- login.xml

```
<soda>
  <puts text="Test: login.xml starting." />

  <!-- Navigate to the login page and verify that it loaded. -->
  <browser url="{@global.url}" />
  <browser assert="Please enter your user name and password" />

  <!-- Loop through the CSV file, executing this block of tags once for each row. -->
  <csv file="login.csv" var="info">
    <!-- Fill in the login form. -->
    <textfield id="user_name" set="{@info.username}" />
    <textfield id="user_password" set="{@info.password}" />

    <!-- Click on the login button and verify that we are logged in. -->
    <button id="login_button" click="true" />
    <browser assert="Welcome, " />

    <!-- Log out. -->
    <link text="Log Out" />
    <browser assert="Please enter your user name and password" />
  </csv>

  <browser action="close" />

  <puts text="Test: login.xml finished." />
</soda>
```

Reporting

- Folders in the SODA dir with the name of the test and the timestamp. E.g.: login-11-10-2010-11-41-results
 - HTML logs
 - Raw logs
 - Saved HTML pages (optional)
- --savehtml saves HTML pages on assert fails.
- --resultdir allows you to override the default results directory.

Pitfalls

- Ruby's activesupport gem is no longer compatible with Watir out of the box and must be downgraded during the install process.
- The Visual C redistributable is required for testing against Firefox on older versions of Windows.
- On Windows, Soda should only be run in a real Windows command shell, not Cygwin.
- Watir's text entry is painfully slow on Linux!
- Watir accesses browser DOM elements different on Firefox vs. IE.
- JS alert dialogs can only be interacted with using the alert="true"/"false" hack

Pitfalls

- The new Confirm dialog in SugarCRM 6.1 requires alert="true" (alert="false" is not supported with the onbeforeunload JavaScript event).
- A new version of Watir is on the horizon and the install procedure will likely change soon. Soda presently only supports the latest version of Watir, 1.6.5, and will not run with other versions.
- HTML <tr> and <td> tags must be accessed with the Soda tags <row> and <cell> respectively.
- All text in the XML file must have proper XML escape codes for special characters.
- All relative paths are relative to the where Soda is executed from.

Advanced Topics

- The `<script>` tag.
- Suites
- Asserts: `assertnot` and `regexes`.
- The “Test: “ string and reporting.
- For more information on Soda command line options, type: `ruby SodaSuite.rb --help`

Resources

- If you find a bug in Soda, file an issue on Github.
- Web Resources
 - <http://www.github.com/sugarcrm/SODA>
 - <http://watir.com/>
 - <http://wiki.openqa.org/display/WTR/FireWatir+Installation>
- Email
 - Ruby/Watir/Soda interpreter issues
Trampus Richmond trichmond@sugarcrm.com
 - Soda test design issues
David Safar dsafar@sugarcrm.com