

Scripting Web Tests

with Watir and Ruby

A One-Day Tutorial

July 2005

Copyright © 2004-2005 Bret Pettichord. Permission granted to reproduce with attribution.



Objectives

- ◆ Learn Test Automation Concepts
- ◆ Dispel Common Myths
- ◆ Gain Hands-on Experience



2

Method

- ◆ Allow people to use their own computers
- ◆ Provide a tool they can keep
- ◆ We found open-source tools and improved them
- ◆ All class materials are open-source and can be shared with others



3

Watir

- ◆ WATIR = Web Application Testing In Ruby
- ◆ Watir is the tool that developed from this class (which we've taught since May 2003)
- ◆ Since Jan 2005, Watir has become popular in its own right
- ◆ But our purpose is still to teach concepts that apply whether you use Watir or another tool – open-source or commercial



4

Watir Testimonials



- ◆ **Watir is the most compelling alternative for filling the automated acceptance testing need** – *Ward Cunningham, comparing Watir to Fit*
- ◆ **Watir Rocks! I have played with it for only about an hour and accomplished what I had wanted to do, but did not have the wherewithal, for ages.** Truly awesome!!
- ◆ **I was able to write a few unit tests using Watir in about 15 minutes** and I've looked at Ruby code for all around 48 hours.
- ◆ Up until now, I'd managed to avoid learning the Ruby scripting language.... **But after seeing what Watir could do and how neat Ruby is, I became a convert.**
- ◆ I've been looking for an easy to use automated test tool for quite some time now. Something went wrong with everything that I tried. When I first came across Watir, I expected the same disappointment. However, Watir blew my freakin mind. Everything worked, no fuss, no muss, no bother. I had never used Ruby or Watir, and within a half a day I was gliding through my application with the greatest of ease. **I can't overstate how much this application is going to change my life**

<http://wtr.rubyforge.org/wiki/wiki.pl?WatirTestimonials>

5

Agenda



1. Show you a web application
2. Teach you how to drive it with a program
3. Turn these programs into tests

6

A Series of Labs



1. Manually test Timeclock
2. Use the Ruby interactive shell (irb)
3. Write scripts
4. Verify results
5. Use Assertions
6. Use a Test Harness

End Result: A Suite of Tests for Timeclock

7

Enough Talk



- ◆ Let's see some action

8

Example Script



```
require 'watir'
site = 'http://www.google.com' # set a variable
ie = Watir::IE.new             # open the IE browser
ie.goto(site)                  # go to the google site
# enter 'pickaxe' in the search text field.
# "q" is the name of the search field
ie.text_field(:name, "q").set("pickaxe")
# click the 'Google Search' button
# "btnG" is the name of the Search button
ie.button(:name, "btnG").click
# see if the Programming Ruby book appears on the page
if ie.contains_text("Programming Ruby")
  puts "Test Passed"
else
  puts "Test Failed!"
end
```

9

Expect to Struggle



- ◆ We don't necessarily teach you everything you need to know to complete each lab
- ◆ But the information you need is documented somewhere
- ◆ Ask questions if you are stuck
- ◆ Feel free to experiment

10