



**LAND REGISTRY**

---

# Test Automation

## Behavior Driven Development (BDD)

---



## Table of Contents

|                          |    |
|--------------------------|----|
| Table of Contents        | 1  |
| Introduction to Ruby     | 1  |
| Introduction to Cucumber | 18 |
| More Cucumber            | 34 |
| Ruby Gems                | 46 |
| HTML, XML and XPATH      | 57 |
| Jenkins Driven Test      | 67 |

## Introduction to Ruby

You will require Ruby to be installed on your computer, please navigate to the link below and follow the steps:

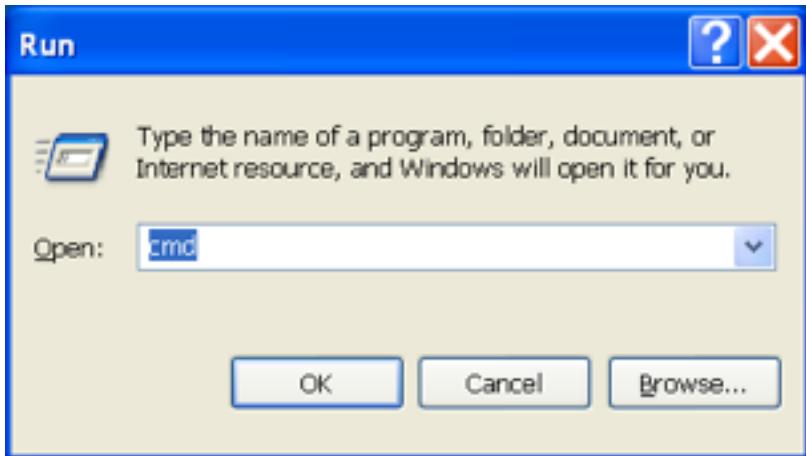
[http://lrg00095593/devwiki/index.php/Settings\\_Up\\_Cucumber](http://lrg00095593/devwiki/index.php/Settings_Up_Cucumber)

Now that ruby is installed:

Ruby is a scripting language. A language simple enough for beginners to use and powerful enough for experienced programmers to have all the tools they need. The best way to understand Ruby is to have a go! Really, have a go, follow the guide and see for yourself.

### 1.1 Open Command Prompt Screen

- Press both the Windows button and the letter ‘r’ on your keyboard at the same time. The prompt below should appear.
- When prompted, type “cmd” and click OK.



- C:\Documents and Settings\CS522BH> will be displayed. (Obviously your own CS sign on will be displayed).
- The Command Prompt Screen will respond as follows:

```
Start Command Prompt with Ruby - ansicon
ruby 1.9.3p362 (2012-12-25) [i386-mingw32]
C:\Documents and Settings\CS522BH>ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\CS522BH>
```

- Go to the C drive root, by changing the directory.
- C:\Documents and Settings\CS522BH> **cd..** press enter.
- Repeat the command C:\Documents and Settings> **cd..** press enter.
- The next response will be C:>



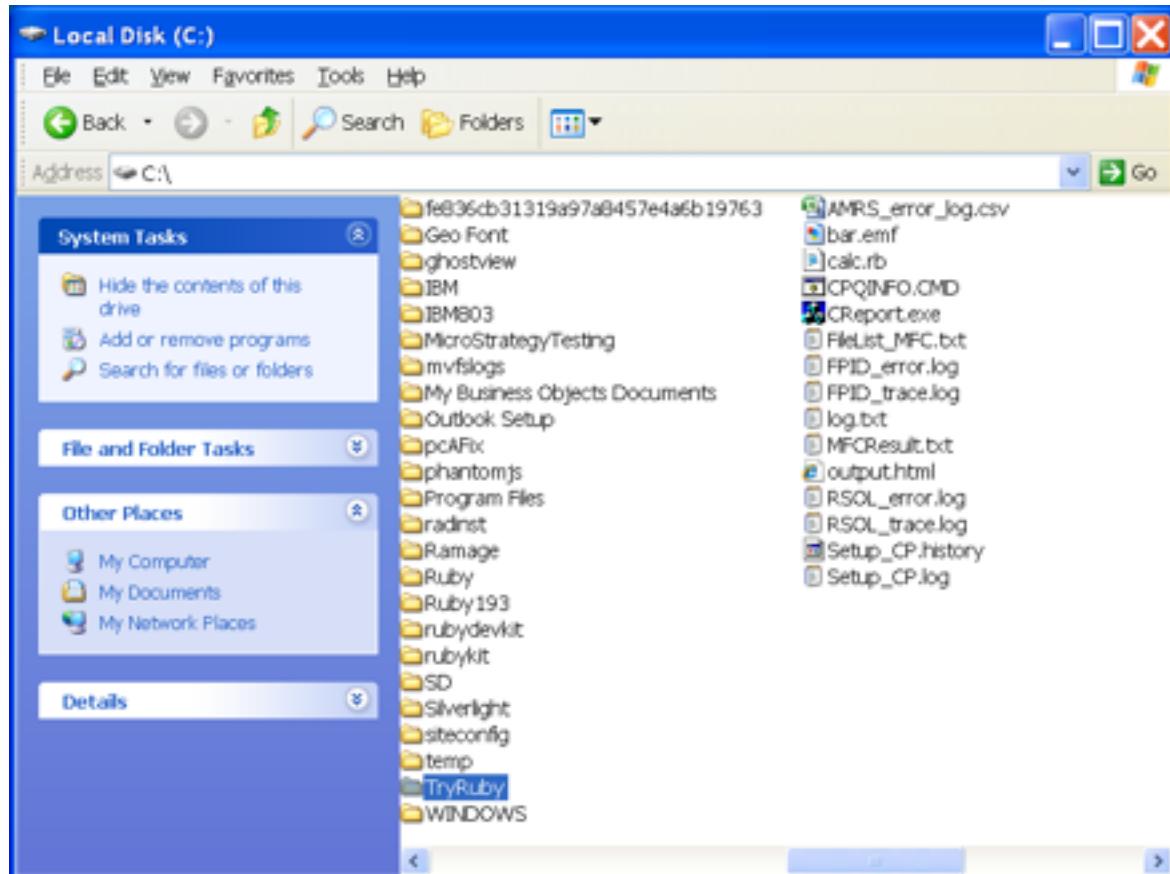
Start Command Prompt with Ruby - ansicon

```
ruby 1.9.3p362 (2012-12-25) [i386-mingw32]
C:\Documents and Settings\CS522BH>ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\CS522BH>cd..
C:\Documents and Settings>cd..
C:\>
```

## 1.2 Create a Directory called TryRuby

- C:\> [mkdir TryRuby](#), press enter.
- Within the C drive a folder named Try Ruby has been created, double click on it and open the folder.





- Command the pc to go to the Try Ruby folder by entering C:\> cd TryRuby, press enter.

```
C:\WINDOWS\system32\cmd.exe - ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\CS522BH>cd ..

C:\Documents and Settings>cd ..

C:\>ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>mkdir TryRuby
C:\>cd TryRuby
C:\TryRuby>
```

Within the TryRuby folder (Directory) you now need to create an .rb file, where Ruby Commands will be inserted.

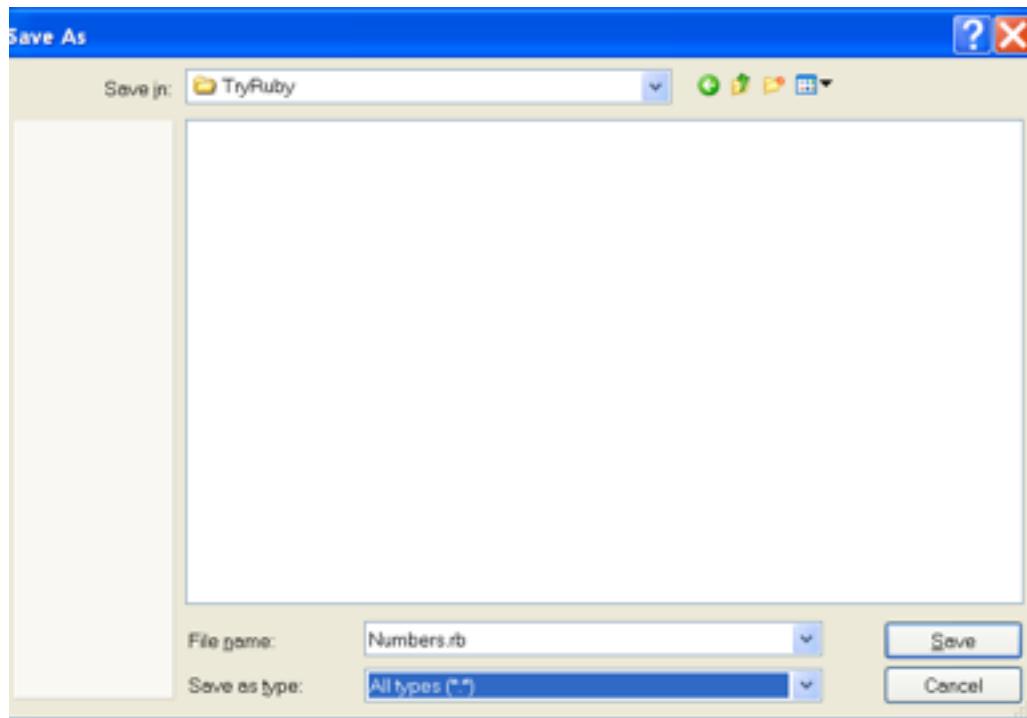
### 1.3 Create a .rb file called Numbers

- Open Notepad
- Alternatively you can use Notepad ++:  
<http://notepad-plus-plus.org/download/v6.3.3.html>
- Type in the following command: puts 2+6

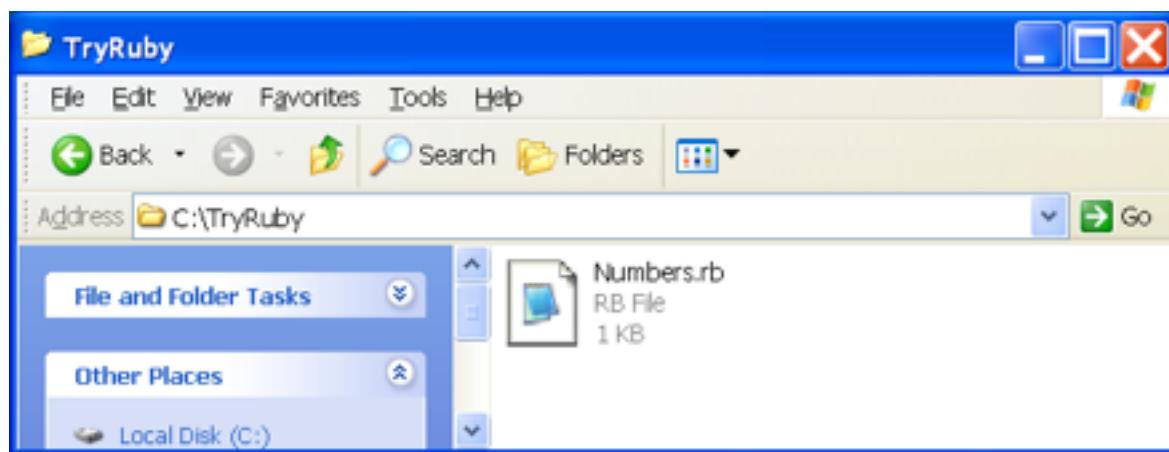
```
Untitled - Notepad
File Edit Format View Help
puts 2+6
```



- Go to file and save as, then navigate to the TryRuby folder in the C Drive
- Type Numbers.rb into File name and All Types into Save as Type, then click Save.



- The file should look as the screen shot below





## 1.4 Run Ruby

- Go to Command Prompt Screen and enter C:\TryRuby> [Ruby Numbers.rb](#), press enter.
- The result of running Ruby reveals that the answer to 2+6 is 8

```
C:\WINDOWS\system32\cmd.exe - ansicon
C:\Documents and Settings>cd ..
C:\>ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\>mkdir TryRuby
C:\>cd TryRuby
C:\TryRuby>Ruby Numbers.rb
8
C:\TryRuby>
```

- Lets try some more simple maths. Add the following commands to the Numbers.rb file

```
Numbers.rb - Notepad
File Edit Format View Help
puts 2+6
puts 4*10
puts 5-12
puts 40/4
```

- Save the file.

### Information

$4 \times 10$  is the same as 4 times 10.

$40/4$  is the same as 40 divided by 4

### Useful Tip

Press the up arrow to cycle through the previous commands to quickly run the previous one!



The answer to the commands (mathematical questions) are instantly returned

```
C:\>mkdir TryRuby
C:\>cd TryRuby
C:\TryRuby>Ruby Numbers.rb
8
40
-7
10
C:\TryRuby>
```

## 1.5 Create an .rb file called Names

Computers are great at handling mathematical commands, but let's investigate further using letters.

A **name** is a *string* of letters.

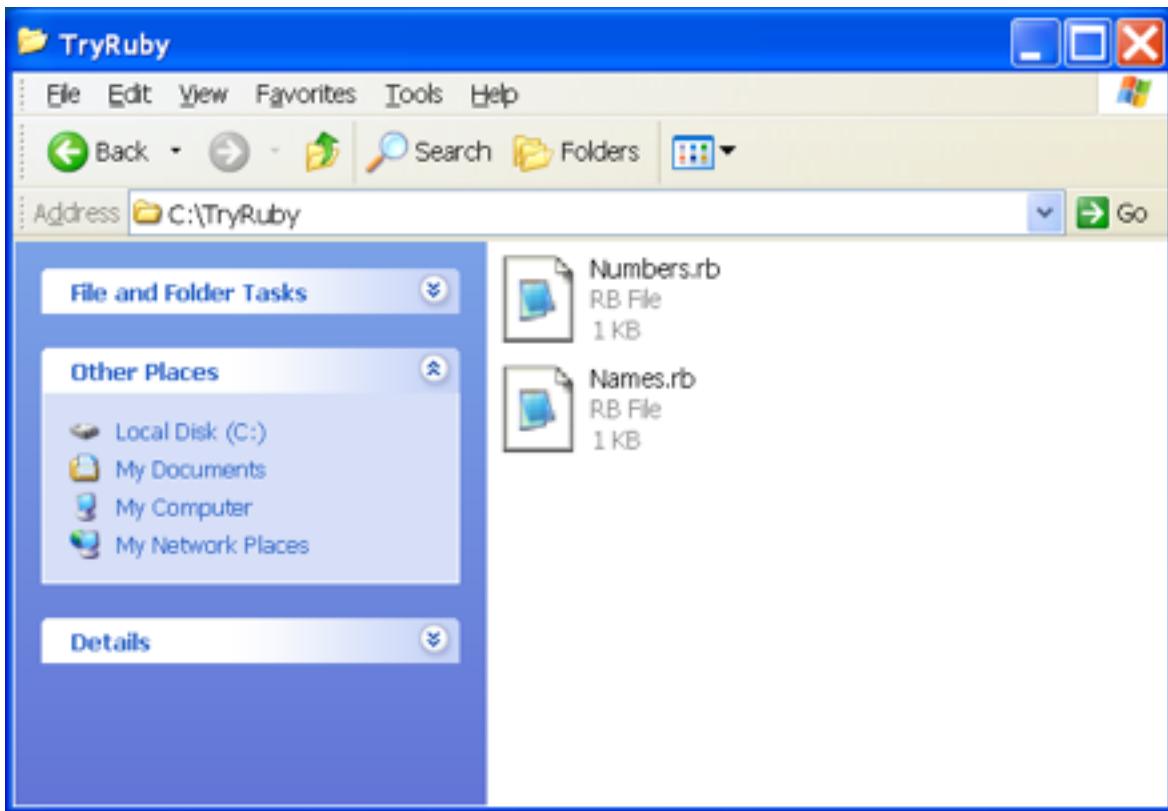
A *string* is a set of characters the computer can process.

Imagine the letters in a name are on a string of laundry line and the quotes are the clothes pegs holding the ends.

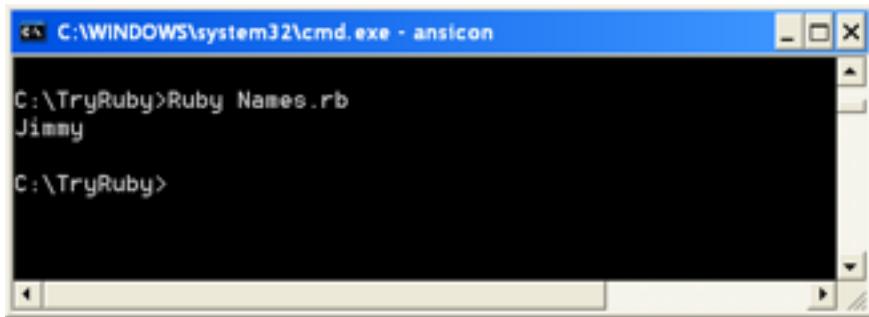
- Open Notepad and type the following.

```
Names.rb - Notepad
File Edit Format View Help
puts "Jimmy"
```

- Save the file within the TryRuby folder. File name: Names.rb and Save as type: All Types.



- Run Ruby. C:\TryRuby>[Ruby Names.rb](#) and press enter



- The result returned from the command screen is that the string of letters is displayed.

## 1.6 Ruby Commands

- As this is just a string of characters, we can write commands to manipulate the character string



- To reverse the string, type `puts "Jimmy".reverse` in the Names.rb file, save the file and run Ruby. Type `C:\TryRuby>Ruby Names.rb` and press enter.

A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.exe - ansicon". The window shows the command `C:\TryRuby>Ruby Names.rb` being run. The output is "Jimmy" followed by its reverse, "ymmiJ". The prompt then changes to "C:\TryRuby>".

- To count how many letters are in the string, type `puts "Jimmy".length` in the Names.rb file
- Save the file and run Ruby.

A screenshot of a Notepad window titled "Names.rb - Notepad". The window contains the following Ruby code:

```
puts "Jimmy"
puts "Jimmy".reverse
puts "Jimmy".length
```

- The result returned for the length of characters in the string is 5.

A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.exe - ansicon". The window shows the command `C:\TryRuby>Ruby Names.rb` being run. The output is "Jimmy" followed by "ymmiJ", then "5", which is the length of the string. The prompt then changes to "C:\TryRuby>".

- To multiply the string, type `puts "Jimmy" * 5` in the Names.rb file, save and run Ruby



A screenshot of a Windows Notepad window titled "Names.rb - Notepad". The window contains the following Ruby code:

```
puts "Jimmy"
puts "Jimmy".reverse
puts "Jimmy".length
puts "Jimmy" * 5
```

The result will be to print the string of characters five times

A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.exe - ansicon". The command "Ruby Names.rb" is run, and the output is:

```
C:\TryRuby>Ruby Names.rb
Jimmy
yimmij
5
JimmyJimmyJimmyJimmyJimmy
C:\TryRuby>
```

#### Learning Point

Numbers and strings are Ruby's math and text objects. You've used English-language commands like **reverse** and symbolic commands like **\*** (multiplication). Commands are actions.

## 1.7 Strings, Integers and Arrays

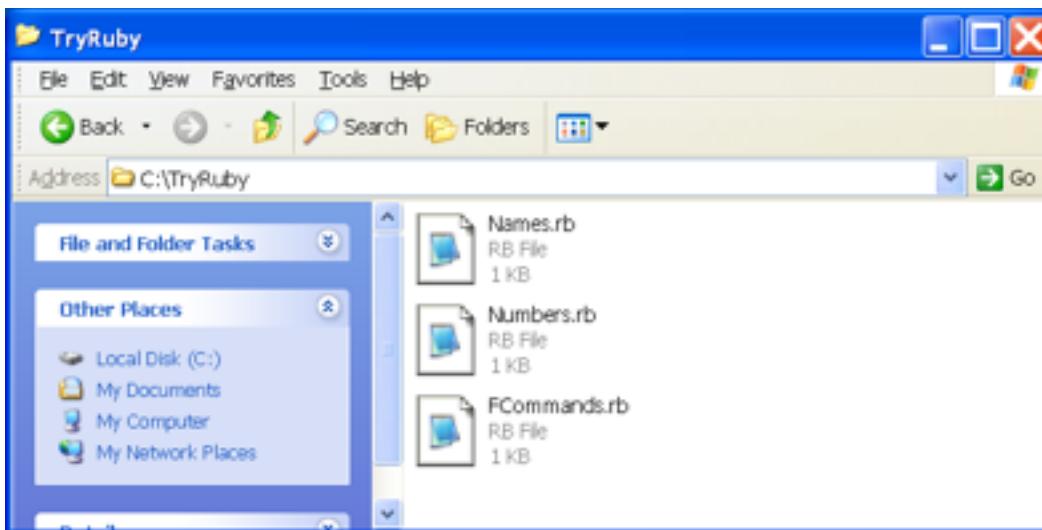
We have used some very simple examples of ruby commands, now it's time to look at some further commands.

- Create another .rb file within TryRuby named FCommands. Open Notepad and enter the following



```
FCommands.rb - Notepad
File Edit Format View Help
puts 40.reverse
```

- Your files within TryRuby should now look like this



- Run ruby. Type C:\TryRuby>[Ruby FCommands.rb](#) and press enter.
- Notice the results

```
C:\WINDOWS\system32\cmd.exe
C:\TryRuby>Ruby FCommands.rb
FCommands.rb:2:in `<main>': undefined method `reverse' for 40:Fixnum (NoMethodError)
C:\TryRuby>
```

- Undefined method 'reverse' – we have an error.  
You can't reverse the number forty, it just doesn't make sense.



- Ruby is telling you there is no command `reverse` for numbers.
- We have to turn the number into a string to reverse
- To convert the number to a string we need to use the command `to_s`
- Let's try it by amending the `FCommands.rb` file, saving the file and running Ruby.

```
File Edit Format View Help
puts 40.to_s.reverse
```

- Let's try it by amending the `FCommands.rb` file, saving the file and running Ruby.

```
C:\WINDOWS\system32\cmd.exe
C:\TryRuby> Ruby FCommands
Ruby: No such file or directory -- FCommands (LoadError)

C:\TryRuby>Ruby FCommands.rb
FCommands.rb:2:in `<main>': undefined method `reverse' for 40:Fixnum (NoMethodError)

C:\TryRuby>Ruby FCommands.rb
04

C:\TryRuby>_
```

- The number 40 has been reversed because it has been turned into a string.

### Information

We have discovered Ruby's "to" commands:

`to_s` converts things to `strings`

`to_i` converts things to `integers (numbers)`

`to_a` converts things to `arrays`

There's another example to try. Let's amend the `FCommands.rb` file, save the file and run ruby.



```
FCommands.rb - Notepad
File Edit Format View Help
puts "9" * 2
puts "9".to_i * 2
```

```
C:\WINDOWS\system32\cmd.exe - ansicon
C:\>cd TryRuby
C:\TryRuby>ruby FCommands.rb
99
18
```

The reason the first command outputs the value of 99 is because it treats it as a string, it just duplicates the value, this is the same as the “puts “Jimmy” \* 5” example you did earlier.

The second line converts the string of 9 into an integer and then times it by 2.

What are arrays? They are lists and they are always put in square brackets [ ].

A list of numbers, for example, could be a list of lottery numbers 40, 3, 21, 9, 7 and 26.

- Amend the FCommands.rb file

```
FCommands.rb - Notepad
File Edit Format View Help
puts [40, 3, 21, 9, 7, 26]
```

- Save the file and run Ruby again.



C:\WINDOWS\system32\cmd.exe

```
C:\TryRuby>Ruby FCommands.rb
40
3
21
9
7
26
C:\TryRuby>_
```

- The list of numbers is shown exactly as we typed them in the list.
- Which number is the highest? Amend the FCommands.rb file

FCommands.rb - Notepad

```
File Edit Format View Help
puts [40, 3, 21, 9, 7, 26].max
```

- Save the file and run Ruby

C:\WINDOWS\system32\cmd.exe

```
C:\TryRuby>Ruby FCommands.rb
40
3
21
9
7
26
C:\TryRuby>Ruby FCommands.rb
40
```

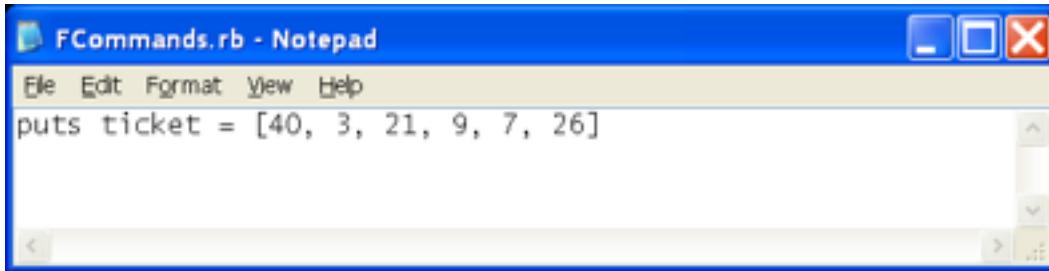
From the results you can see that 40 is the highest number.

## 1.8 Variables and Sorts.

It would be really annoying to have to keep typing the list of numbers every time you referred to them, there must be a way of referring to the lottery numbers, a way of storing them. We can give the list of numbers a name which is a *variable*.



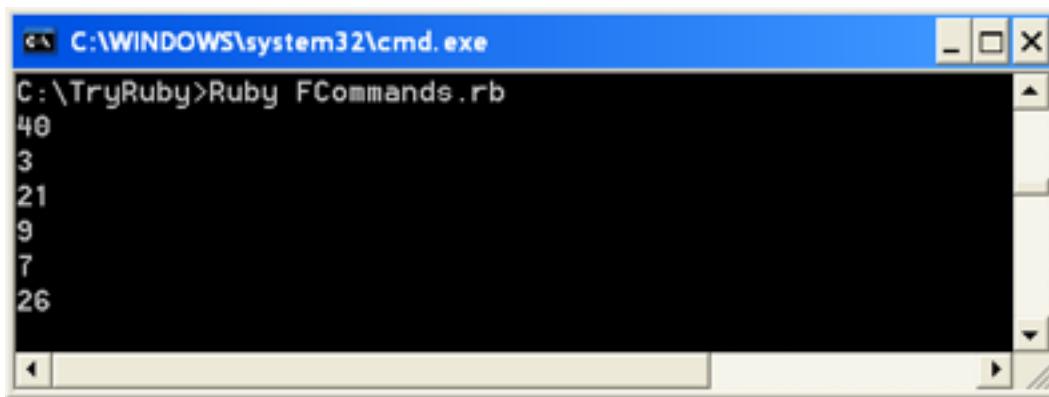
- Let's call the lottery numbers, ticket. Amend the FCommands.rb file



A screenshot of a Windows Notepad window titled "FCommands.rb - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main text area contains the following Ruby code:

```
puts ticket = [40, 3, 21, 9, 7, 26]
```

- Save the file and run Ruby.

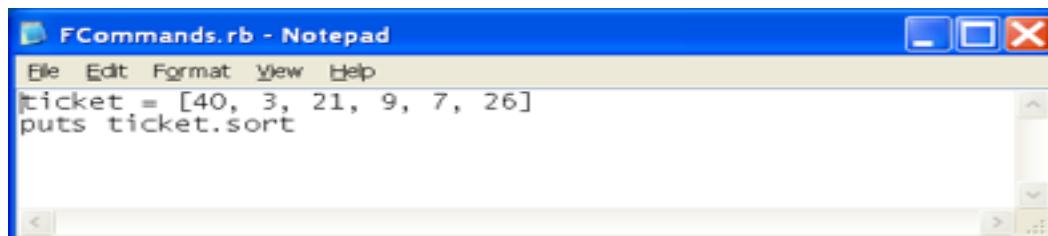


A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.exe". The command line shows "C:\TryRuby>Ruby FCommands.rb". The output displays the six lottery numbers listed vertically:

```
40  
3  
21  
9  
7  
26
```

Ruby has displayed the numbers in the variable named ticket.

- The numbers in the variable named ticket can now be sorted. Amend the FCommands.rb



A screenshot of a Windows Notepad window titled "FCommands.rb - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main text area contains the following Ruby code, which includes a sorting command:

```
ticket = [40, 3, 21, 9, 7, 26]
puts ticket.sort
```

- Save the file and run Ruby.



## Chapter

## 2

A screenshot of a Windows Command Prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command entered is 'C:\TryRuby>Ruby FCommands.rb'. The output shows the numbers 3, 7, 9, 21, 26, and 40, which have been sorted numerically.

```
C:\TryRuby>Ruby FCommands.rb
3
7
9
21
26
40
```

Ruby has displayed the numbers in the ticket in numerical order, in this instance.

- To permanently sort the numbers in the variable, so that they are in numerical order **every time** we refer to them we place an exclamation mark at the end of the word sort.

A screenshot of a Notepad window titled 'FCommands.rb - Notepad'. The file contains the following Ruby code:

```
ticket = [40, 3, 21, 9, 7, 26]
puts ticket.sort!
```

- Save the file and run Ruby.
- Ruby displays the numbers in numerical order, again. This time you cannot see any difference on the command screen but the variable named ticket is now stored in numerical order.



C:\WINDOWS\system32\cmd.exe

```
C:\TryRuby>Ruby FCommands.rb
3
7
9
21
26
40

C:\TryRuby>
```



# Introduction to Cucumber

This chapter will guide you on how to create your very first cucumber script. This will cover the folder structure, writing a gherkin scenario, writing capybara code and then executing and troubleshooting.

## 1. Open Command Prompt Screen

- If not already open, open the command screen as you did before in the previous chapter
- Go to the C drive root, by changing the directory.
- C:\Documents and Settings\CS522BH> **cd..** press enter.
- Repeat the command C:\Documents and Settings> **cd..** press enter.
- The next response will be C:\>

```
ruby 1.9.3p362 (2012-12-25) [i386-mingw32]
C:\Documents and Settings\CS522BH>ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\CS522BH>cd..
C:\Documents and Settings>cd..
C:\>
```

### INFORMATION

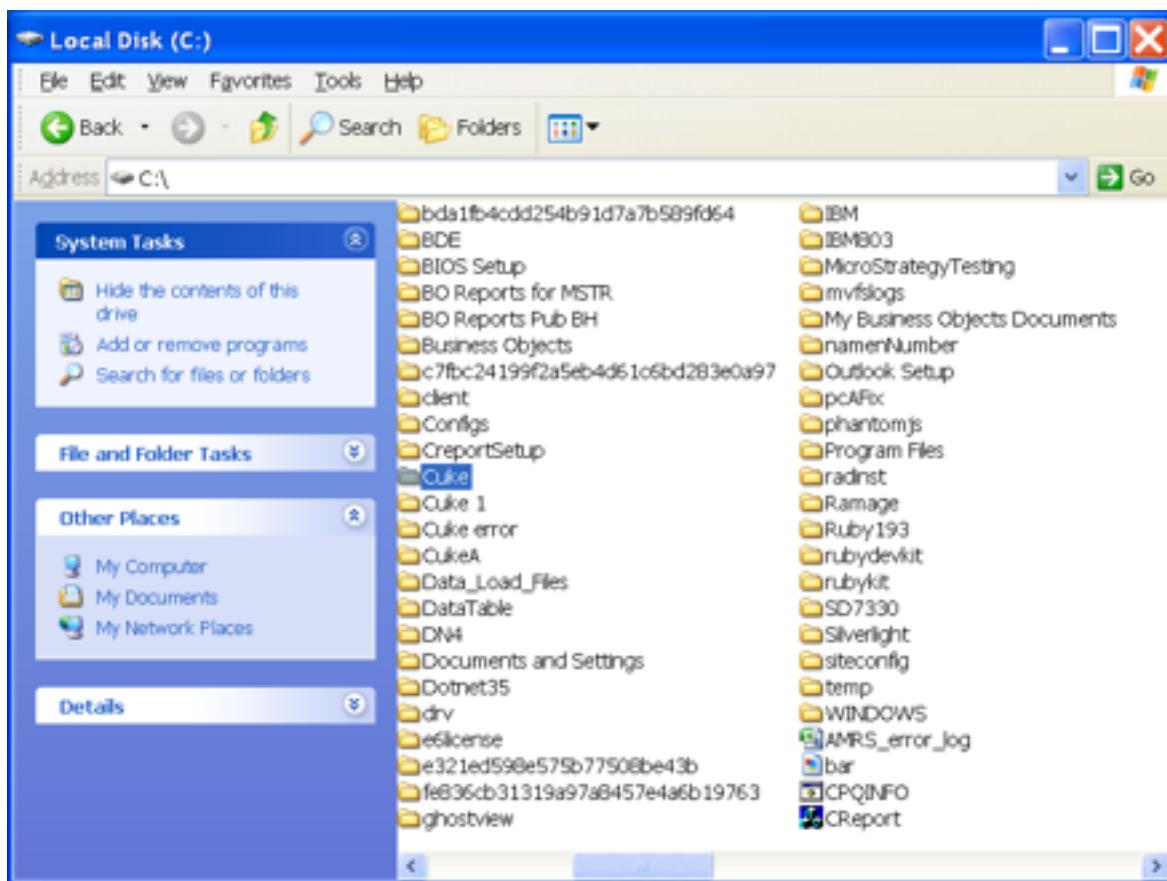
All folders and files are stored in the C Drive which is why we have navigated the pc to C:>

Open a window with your C Drive displayed, that way you can see folders and



## 2. Create a Directory called Cuke

- `C:\> mkdir Cuke` press enter.
- Within the C drive a folder named Cuke has been created, double click on it and open the folder.



- Command the pc to go to the Cuke folder by entering `C:\> cd Cuke`, press enter



```
Start Command Prompt with Ruby - ansicon
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\CS522BH>cd..

C:\Documents and Settings>cd..

C:\>mkdir Cuke

C:\>cd Cuke

C:\Cuke>
```

### 3. Run Cucumber

- Go to Command Prompt Screen and enter C:\Cuke> [cucumber](#) then press enter.

```
Start Command Prompt with Ruby - ansicon
C:\Documents and Settings>cd..

C:\>mkdir Cuke

C:\>cd Cuke

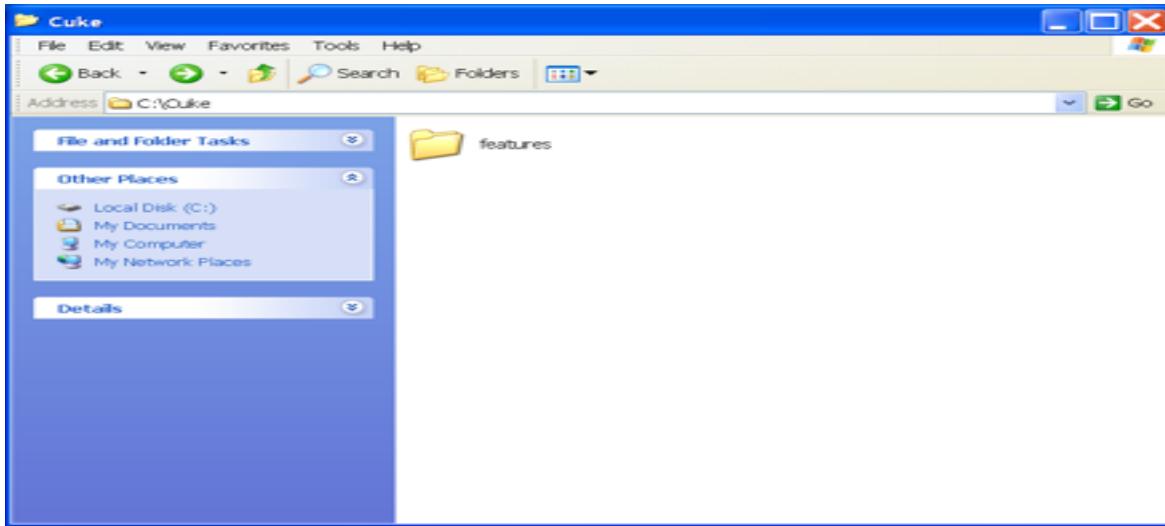
C:\Cuke>cucumber
You don't have a 'features' directory. Please create one to get started.
See http://cukes.info/ for more information.

C:\Cuke>
```

- The result of running cucumber reveals that we do not have a features directory. You may not have exactly the same message as above but it still means we do not have a features directory.

### 4. Create a Features Directory

- C:\ Cuke> [mkdir features](#) then press enter.



### INFORMATION

Open the Cuke folder in the C drive and you will have a new folder named features, double click on the folder and open it.

- Run cucumber again, C:\ Cuke> cucumber, press enter

```
Start Command Prompt with Ruby - ansicon
C:\Cuke>cucumber
You don't have a 'features' directory. Please create one to get started.
See http://cukes.info/ for more information.

C:\Cuke>mkdir features

C:\Cuke>cucumber
0 scenarios
0 steps
0m0.000s

C:\Cuke>_
```

### INFORMATION

Cucumber found a features directory however we have not created any scenarios or steps to run within the feature directory. To do this you will need to open Notepad ++ and create a scenario.



**2.5 Open Notepad** (You may still have this open from downloading it in the previous chapter).

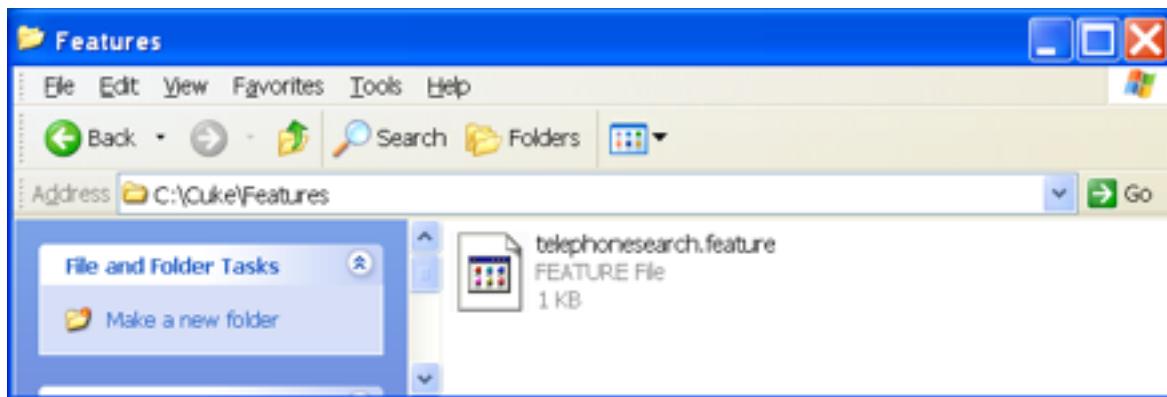
## 2.6 Create a Scenario

- Type the following into notepad

```
telephonesearch.feature - Notepad
File Edit Format View Help
Feature: Intranet Telephone Search
Scenario: Search for surname in the Quick Telephone Search
  Given I navigate to the corporate home page
  When I enter the name of Howell into the telephone search field
  Then the text BARBARA appears on the screen
```

## 2.7 Save the Feature File

- To do this go to File, Save As, navigate to the open features folder and save using 'telephonesearch.feature' as the filename and 'All Types' as the type of file.



**IMPORTANT**

Be **very** careful how you save files, file extensions must be precise.

- Run Cucumber, C:\ Cuke> [cucumber](#), press enter.



```
C:\RubySetup\ansi153\x86\ansicon.exe

C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search      # features\t
elephonesearch.feature:3
    Given I navigate to the corporate home page                  # features\t
elephonesearch.feature:4
    When I enter the name of Howell into the telephone search field # features\t
elephonesearch.feature:5
    Then the text BARBARA appears on the screen                 # features\t
elephonesearch.feature:6

1 scenario (1 undefined)
3 steps (3 undefined)
0m0.000s

You can implement step definitions for undefined steps with these snippets:

Given(/^I navigate to the corporate home page$/) do
  pending # express the regexp above with the code you wish you had
end

When(/^I enter the name of Howell into the telephone search field$/) do
  pending # express the regexp above with the code you wish you had
end

Then(/^the text BARBARA appears on the screen$/) do
  pending # express the regexp above with the code you wish you had
end

If you want snippets in a different programming language,
just make sure a file with the appropriate file extension
exists where cucumber looks for step definitions.

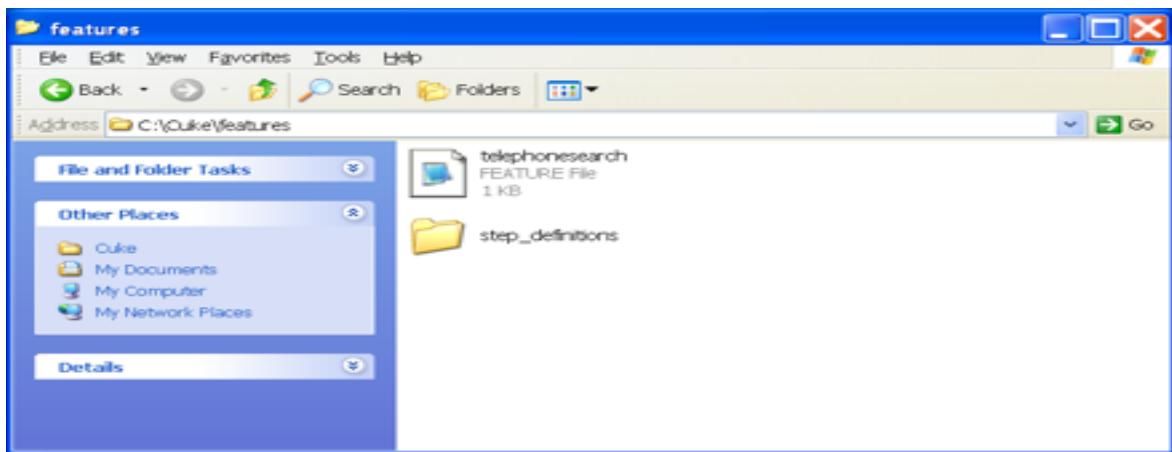
C:\Cuke>
```

You can see from the response that the scenario is undefined and there are 3 undefined steps. Directly underneath this information cucumber has provided snippets which can be used to create step definitions.

## 2.8 Implement Step Definitions



- Change direction to the features directory
- Make a new directory named `step_definitions`



- Open the step-definitions folder.

## 2.9 Create Step Definitions File

- The step definitions can be copied from the snippets. Copy the text from the Command Prompt Screen and paste to Notepad++ by
  - o Click on the black icon (top left)
  - o Scroll to edit
  - o Click on mark
  - o Highlight the text you need to copy



Select C:\RubySetup\ansi153\x86\ansicon.exe

```
1 scenario (1 undefined)
3 steps (3 undefined)
0m0.000s

You can implement step definitions for undefined steps with these snippets:

Given(/^I navigate to the corporate home page$/)
  pending # express the regexp above with the code you wish you had
end

When(/^I enter the name of Howell into the telephone search field$/)
  pending # express the regexp above with the code you wish you had
end

Then(/^the text BARBARA appears on the screen$/)
  pending # express the regexp above with the code you wish you had
end

If you want snippets in a different programming language,
just make sure a file with the appropriate file extension
exists where cucumber looks for step definitions.
```

- o Right click on the area you selected
- o Go to Notepad, right click and paste.

step\_telephonesearch.rb - Notepad

```
File Edit Format View Help
Given /AI navigate to the corporate home page$/ do
  pending # express the regexp above with the code you wish you had
end

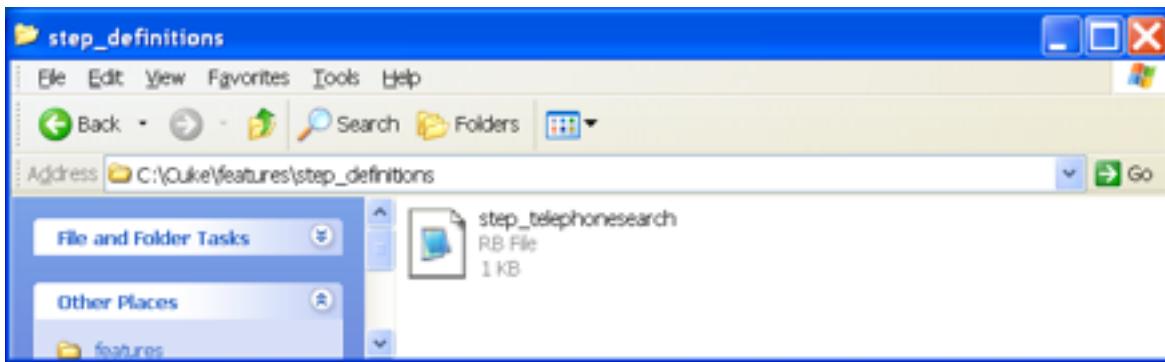
When /AI enter the name of Howell into the telephone search field$/ do
  pending # express the regexp above with the code you wish you had
end

Then /At the text BARBARA appears on the screen$/ do
  pending # express the regexp above with the code you wish you had
end
```



## 2.10 Save the Ruby File

- Step definition files are saved as .rb files (ruby files).
- Go to file, click on save as, then navigate to step\_definitions folder and save using file name '[step\\_telephonesearch.rb](#)' and the file type is 'All types'.



- Run cucumber, C:\ Cuke> [cucumber](#), press enter.

```
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search      # features\telephonesearch.feature:3
    Given I navigate to the corporate home page                  # features\step_definitions\step_telephonesearch.rb:1
      TODO (Cucumber::Pending)
      ./features/step_definitions/step_telephonesearch.rb:2:in `I navigate to
      the corporate home page$'
        features\telephonesearch.feature:4:in `Given I navigate to the corporate h
ome page'
        When I enter the name of Howell into the telephone search field # features\ste
p_definitions\step_telephonesearch.rb:5
        Then the text BARBARA appears on the screen                 # features\ste
p_definitions\step_telephonesearch.rb:9

1 scenario (1 pending)
3 steps (2 skipped, 1 pending)
0m0.016s

C:\Cuke>
```



## 2.11 Enter Ruby Code

- The next step is to replace the pending and skipped statements with ruby code and to insert a Header into the step\_telephonesearch.rb file.

```

step_telephone_search.rb - Notepad
File Edit Format View Help
### START HEADER - DON'T TOUCH ###

require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end

### END HEADER - DON'T TOUCH ###

Given(/^I navigate to the corporate home page$/) do
  pending # express the regexp above with the code you wish you had
end

when(/^I enter the name Howell into the telephone search field$/) do
  pending # express the regexp above with the code you wish you had
end

Then(/^the text BARBARA appears on the screen$/) do
  pending # express the regexp above with the code you wish you had
end

```

- Save the file.
- Each step needs to have ruby code inserted.  
The code will replace the line 'pending # express the regexp above with the code you wish you had'.
- Each step will be coded separately in this guide so we can see any error messages and consider them as they arise.
- Update the first step with the following code:



The screenshot shows a Microsoft Notepad window titled "step\_telephone search.rb - Notepad". The content of the file is a Ruby script for a Capybara test. It includes require statements for 'capybara', 'capybara/dsl', and 'rspec'. It sets up Capybara with 'localhost:4567' as the host, 'css' as the default selector, 'selenium' as the default driver, and a wait time of 10 seconds. It registers the 'selenium' driver and specifies Internet Explorer as the browser. The script then defines three scenarios: navigating to the corporate home page, entering 'Howell' into the telephone search field, and checking if 'BARBARA' appears on the screen. The code uses pending blocks for the second and third scenarios.

```
step_telephone search.rb - Notepad
File Edit Format View Help
### START HEADER - DON'T TOUCH ###
require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end

### END HEADER - DON'T TOUCH ###

Given(/^I navigate to the corporate home page$/) do
  visit "http://intranet/cs/defaultin.asp"
end

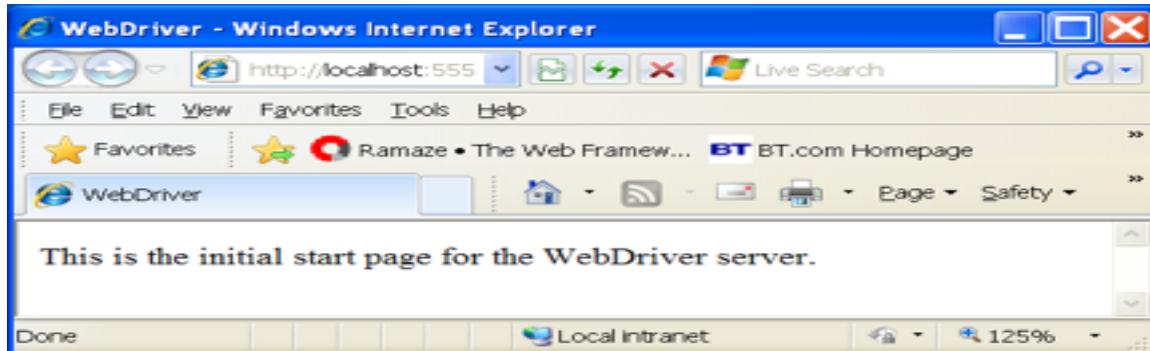
When(/^I enter the name Howell into the telephone search field$/) do
  pending # express the regexp above with the code you wish you had
end

Then(/^the text BARBARA appears on the screen$/) do
  pending # express the regexp above with the code you wish you had
end
```

The line visit "<http://intranet/corp/>" simply means the pc opens the browser and goes to the web page.



- Save the file. Run cucumber, C:\Cuke> **cucumber**, press enter.
- You will see that the pc opens Windows Internet Explorer but stops before it performs a search **and** that the test has failed in Cucumber.  
(see both snap shots below)



```
xx Start Command Prompt with Ruby - ansicon
C:\Cuke>
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search      # features\telephonesearch.feature:3
  Started InternetExplorerDriver server (32-bit)
  2.29.0.0
  Listening on port 5555
    Given I navigate to the corporate home page                # features\step_definitions\step_telephonesearch.rb:21
      Unexpected error launching Internet Explorer. Browser zoom level was set to 125%. It should be set to 100% (Selenium::WebDriver::NoSuchDriverError)

      ./features/step_definitions/step_telephonesearch.rb:22:in `I navigate to the corporate home page$'
        features\telephonesearch.feature:4:in `Given I navigate to the corporate home page'
          When I enter the name of Howell into the telephone search field # features\step_definitions\step_telephonesearch.rb:25
            Then the text BARBARA appears on the screen               # features\step_definitions\step_telephonesearch.rb:29

  Failing Scenarios:
  cucumber features\telephonesearch.feature:3 # Scenario: Search for Howell in the Quick Telephone Search

  1 scenario (1 failed)
  3 steps (1 failed, 2 skipped)
  0m1.875s

C:\Cuke>
```

## 2.12 Error Messages

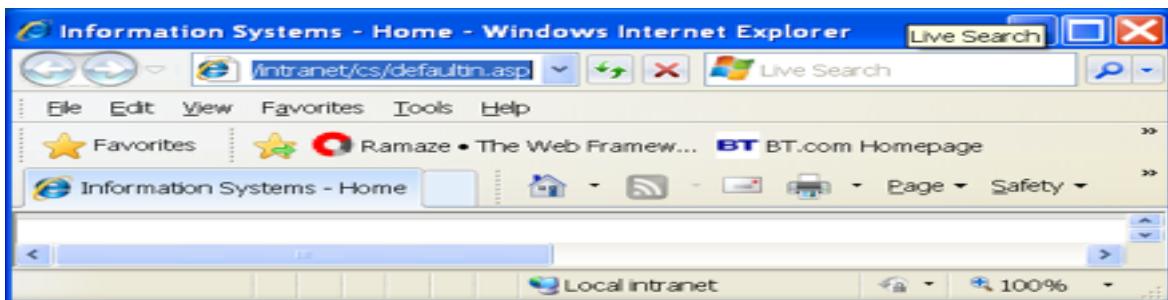
### IMPORTANT TIP

Always read any Error Message.

The error message will state the exact problem which will allow you to go and fix the code before you run cucumber again.



- The error message states: **Unexpected error launching Internet Explorer. Browser zoom level was set to 125%. It should be set to 100%** Change the zoom level on Internet explorer window to 100%



- Run cucumber again. C:\Cuke> **cucumber**, press enter.

```
PS Start Command Prompt with Ruby - ansiicon
Failing Scenarios:
cucumber features\telephonesearch.feature:3 # Scenario: Search for Howell in the
Quick Telephone Search

1 scenario (1 failed)
3 steps (1 failed, 2 skipped)
0m1.875s

C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search      # features\t
elephonesearch.feature:3
  Started InternetExplorerDriver server (32-bit)
  2.29.0.0
  Listening on port 5555
    Given I navigate to the corporate home page                  # features\s
tep_definitions\step_telephonesearch.rb:21
    When I enter the name of Howell into the telephone search field # features\s
tep_definitions\step_telephonesearch.rb:25
      TODO (Cucumber::Pending)
      ./features/step_definitions/step_telephonesearch.rb:26:in `/^I enter the n
ame of Howell into the telephone search field$/'
      features\telephonesearch.feature:5:in `When I enter the name of Howell int
o the telephone search field'
      Then the text BARBARA appears on the screen                 # features\s
tep_definitions\step_telephonesearch.rb:29

1 scenario (1 pending)
3 steps (1 skipped, 1 pending, 1 passed)
0m3.688s

C:\Cuke>_
```

- The first step passed. The PC has successfully navigated to and opened Internet Explorer.



- Now it's time to write the When statement, enter ruby code into the next step.
- We need to tell the pc where to input the surname Howell. We also need the code for the location of the Quick telephone search box so the pc fills in the correct location.

A screenshot of a Microsoft Notepad window titled "step\_telephone\_search.rb - Notepad". The window contains Ruby code for a Cucumber feature. The code includes setup for Capybara, a scenario for navigating to a corporate home page, a 'when' step to enter 'Howell' into a surname field, and a 'then' step to check for the text 'BARBARA' on the screen. The code uses the Capybara DSL and Selenium driver.

```
step_telephone_search.rb - Notepad
File Edit Format View Help
### START HEADER - DON'T TOUCH ###
require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end

### END HEADER - DON'T TOUCH ###

Given(/^I navigate to the corporate home page$/) do
  visit "http://intranet/cs/defaultin.asp"
end

When(/^I enter the name Howell into the telephone search field$/) do
  fill_in('surname', :with => 'Howell')
  click_button('Go')
end

Then(/^the text BARBARA appears on the screen$/) do
  pending # express the regexp above with the code you wish you had
end
```



Start Command Prompt with Ruby - ansiicon

```
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search          # features\telephonesearch.feature:3
    Started InternetExplorerDriver server (32-bit)
    2.29.0.0
    Listening on port 5555
      Given I navigate to the corporate home page                      # features\step_definitions\step_telephonesearch.rb:21
      When I enter the name of Howell into the telephone search field # features\step_definitions\step_telephonesearch.rb:25
        Then the text BARBARA appears on the screen                   # features\step_definitions\step_telephonesearch.rb:31
        TODO (Cucumber::Pending)
        ./features/step_definitions/step_telephonesearch.rb:32:in `Then the text BARBARA appears on the screen$'
          features\telephonesearch.feature:6:in `Then the text BARBARA appears on the screen'

  1 scenario (1 pending)
  3 steps (1 pending, 2 passed)
  0m6.312s

C:\Cuke>_
```

- 2 steps have now passed, the Given and the When.
- To run the final step we need to write the code for the Then statement, we need to enter the following ruby code

step\_telephone\_search.rb - Notepad

```
File Edit Format View Help
require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end

### END HEADER - DON'T TOUCH ###

Given(/I navigate to the corporate home page$/) do
  visit "http://intranet/cs/defaultin.asp"
end

When(/I enter the name of Howell into the telephone search fields$/) do
  fill_in('surname', :with => 'Howell')
  click_button('Go')
end

Then(/the text BARBARA appears on the screen$/) do
  page.should have_content ('BARBARA')
end
```

- Save the file. Run cucumber. C:\ Cuke> **cucumber**, press enter.



```
65 Start Command Prompt with Ruby - ansiicon
1 scenario (1 pending)
3 steps (1 pending, 2 passed)
0m6.312s

C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search For Howell in the Quick Telephone Search      # features\t
    telephonesearch.feature:3
    Started InternetExplorerDriver server (32-bit)
    2.29.0.0
    Listening on port 5555
      Given I navigate to the corporate home page                # features\d
      step_definitions\step_telephonesearch.rb:21
      When I enter the name of Howell into the telephone search field # features\d
      step_definitions\step_telephonesearch.rb:25
      Then the text BARBARA appears on the screen                 # features\d
      step_definitions\step_telephonesearch.rb:31

1 scenario (1 passed)
3 steps (3 passed)
0m7.406s

C:\Cuke>
```

Notice that all the text is green.

## WELL DONE

- 1 scenario passed and 3 steps passed.
- Internet Explorer opens, Howell is typed in the search field and BARBARA appears on the screen.

End of the first simple test.



# More Cucumber

Now we have successfully passed a very simple test, we want to build on it and make the underlying code more reusable. This chapter will cover a number of techniques that will allow you to expand your cucumber scenarios but still maintaining a small step definition file.

What if we want to search for another name in the Quick Telephone Search?

## 3.1 Create another Scenario

- Go to the telephonesearch FEATURE file within the features folder and copy the scenario, then amend the names within the scenario
- Put speech marks around the surname and first name

The screenshot shows a Windows Notepad window titled "telephonesearch - Notepad". The content is a Cucumber feature file:

```
Feature: Intranet Telephone Search
  Scenario: Search for Howell in the Quick Telephone Search
    Given I navigate to the corporate home page
    When I enter the name of "Howell" into the telephone search field
    Then the text "BARBARA" appears on the screen

  Scenario: Search for Moore in the Quick Telephone Search
    Given I navigate to the corporate home page
    When I enter the name of "Moore" into the telephone search field
    Then the text "Andy" appears on the screen
```

**Information** Notice the double quotes around the “surname” and “first name”.

- Save the file. Run cucumber, C:\ Cuke> **cucumber**, press enter.
- The response tells you that you can add more snippets into the step definitions.



```
ss C:\WINDOWS\system32\cmd.exe - ansicon
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search      # features
  \telephonesearch.feature:3
  Started InternetExplorerDriver server (32-bit)
  2.29.0.0
  Listening on port 5555
    Given I navigate to the corporate home page      # features
    /step_definitions/step_telephonesearch.rb:21
    When I enter the name of "Howell" into the telephone search field # features
    \telephonesearch.feature:5
      Then the text "BARBARA" appears on the screen      # features
    \telephonesearch.feature:6

  Scenario: Search for Moore in the Quick Telephone Search      # features\
  telephonesearch.feature:8
    Given I navigate to the corporate home page      # features/
    step_definitions/step_telephonesearch.rb:21
    When I enter the name of "Moore" into the telephone search field # features\
  telephonesearch.feature:10
      Then the text "Andy" appears on the screen      # features\
  telephonesearch.feature:11

2 scenarios (2 undefined)
6 steps (4 undefined, 2 passed)
0m6.812s

You can implement step definitions for undefined steps with these snippets:

When /^I enter the name of "(.*)" into the telephone search field$/ do |arg1|
  pending # express the regexp above with the code you wish you had
end

Then /^the text "(.*)" appears on the screen$/ do |arg1|
  pending # express the regexp above with the code you wish you had
end

C:\Cuke>_
```

**Information** If you are reusing the same ruby file as chapter 2, then delete the previous When and Then chunks of code.

- Not all the scenarios and steps have passed. The response is that the step definitions need to be updated.



### 3.2 Update the step definitions file

- Copy the snippets to the bottom of the step\_telephonesearch RB File (C:\Cuke\features\step\_definitions)

```
xx Select C:\WINDOWS\system32\cmd.exe
Scenario: Search for Moore in the Quick Telephone Search      # features\BHT
elephonesearch.feature:8
  Given I navigate to the corporate home page                # features/ste
p_definitions/step_telephone_search.rb:20
  When I enter the name "Moore" into the telephone search field # features\BHT
elephonesearch.feature:10
  Then the text "Andy" appears on the screen                  # features\BHT
elephonesearch.feature:11

2 scenarios (2 undefined)
6 steps (4 undefined, 2 passed)
0m3.984s

You can implement step definitions for undefined steps with these snippets:

When(/^I enter the name "(.*)" into the telephone search field$/) do |arg1|
  pending # express the regexp above with the code you wish you had
end

Then(/^the text "(.*)" appears on the screen$/) do |arg1|
  pending # express the regexp above with the code you wish you had
end

C:\Cuke>
```

- The response has given some code, (.\*) and |arg 1| at the end, to work with.
- Edit the given, when, then statement with the code as below.



step\_telephone\_search.rb - Notepad

```

File Edit Format View Help
### START HEADER - DON'T TOUCH ###

require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end

### END HEADER - DON'T TOUCH ###

Given(/^I navigate to the corporate home page$/) do
  visit "http://intranet/cs/defaultin.asp"
end

When(/^I enter the name "(.*?)" into the telephone search field$/) do |arg1|
  fill_in('surname', :with => arg1)
  click_button('Go')
end

Then(/^the text "(.*?)" appears on the screen$/) do |arg1|
  page.should have_content(arg1)
end

```

### INFORMATION

**(.\*?)** is a *regular expression* (also known as regex).

This particular regular expression will catch all text within the “marks and save the value as a variable called arg1. This is then used to fill in the surname value.

Regular Expressions can be really complex (See below for an example of a post code validation regular expression), however generally cucumber will provide you the code for this.

`^(A-PR-UWYZ0-9)[A-HK-Y0-9][AEHMNPRTVXY0-9]?[ABEHMNPRVWXY0-9]? {1,2}[0-9][ABD-HJLN-UW-Z]{2}|GIR 0AA$`

**arg1** is an *argument*.

This particular argument will save the surname as a variable and Cucumber has called it arg1. Arguments are shown between two vertical line symbols - |arg1|.



- Save the file. Then run cucumber, C:\Cuke> **cucumber**, press enter.

```
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search          # features
  \telephonesearch.feature:3
  Started InternetExplorerDriver server (32-bit)
  2.29.0.0
  Listening on port 5555
    Given I navigate to the corporate home page                      # features
    /step_definitions/step_telephonesearch.rb:22
      When I enter the name of "Howell" into the telephone search field # features
    /step_definitions/step_telephonesearch.rb:26
        Then the text "BARBARA" appears on the screen                 # features
    /step_definitions/step_telephonesearch.rb:31

  Scenario: Search for Moore in the Quick Telephone Search          # features\
  telephonesearch.feature:8
    Given I navigate to the corporate home page                      # features/
    step_definitions/step_telephonesearch.rb:22
      When I enter the name of "Moore" into the telephone search field # features/
    step_definitions/step_telephonesearch.rb:26
        Then the text "Andy" appears on the screen                   # features/
    step_definitions/step_telephonesearch.rb:31

  2 scenarios (2 passed)
  6 steps (6 passed)
  0m12.905s

C:\Cuke>_
```

To search another name, add another scenario to the feature – try this for yourself and you should get 3 scenarios (3 passed) and 9 steps (9 passed).

TIP - Don't forget to save the file before you run cucumber.

The final result should look like the screen shot below.



```
Start Command Prompt with Ruby - ansiicon
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario: Search for Howell in the Quick Telephone Search          # Features
    \telephonesearch.feature:3
    Started InternetExplorerDriver server (32-bit)
    2.29.0.0
    Listening on port 5555
      Given I navigate to the corporate home page                      # Features
      /step_definitions/step_telephonesearch.rb:22
        When I enter the name of "Howell" into the telephone search field # Features
        /step_definitions/step_telephonesearch.rb:26
          Then the text "BARBARA" appears on the screen                  # Features
        /step_definitions/step_telephonesearch.rb:31

      Scenario: Search for Moore in the Quick Telephone Search          # Features
        telephonesearch.feature:8
        Given I navigate to the corporate home page                      # Features
        step_definitions/step_telephonesearch.rb:22
          When I enter the name of "Moore" into the telephone search field # Features
          step_definitions/step_telephonesearch.rb:26
            Then the text "Andy" appears on the screen                  # Features
          step_definitions/step_telephonesearch.rb:31

      Scenario: Search for Dart in the Quick Telephone Search          # Features\t
        telephonesearch.feature:13
        Given I navigate to the corporate home page                      # Features\s
        step_definitions/step_telephonesearch.rb:22
          When I enter the name of "Dart" into the telephone search field # Features\s
          step_definitions/step_telephonesearch.rb:26
            Then the text "JULIA" appears on the screen                  # Features\s
          step_definitions/step_telephonesearch.rb:31

  3 scenarios (3 passed)
  9 steps (9 passed)
  0m18.202s

C:\Cuke>_
```

- Now create an error by changing one of the names to something that doesn't exist in the telephone directory e.g Andy Grainger



### 3.3 Outline Scenario and Examples

So far, a new scenario has been written for each name search. The more names searched the larger the feature file becomes.

Another way of doing this would be to turn the scenario into a Scenario outline, which lets us specify multiple scenarios using a table (or examples).

### 3.4 Create an Outline Scenario

- Open the `telephonesearch.feature` file with the feature directory. Look at the line starting with ‘When’. The surname you are searching will become the input and the first name that appears on the screen will become the output. The ‘Examples’ will be a table of surnames and first names.

```
telephonesearch.feature - Notepad
File Edit Format View Help
Feature: Intranet Telephone Search

Scenario outline: Search for surname in the Quick Telephone Search
Given I navigate to the corporate home page
When I enter the name of "<input>" into the telephone search field
Then the text "<output>" appears on the screen

Examples:
| input      | output    |
| Howell    | BARBARA |
| Moore     | Andy     |
| Dart      | JULIA   |
```

**Note:** Where there was “Scenario:” before, this has now changed to “Scenario Outline.”

Run this by typing `cucumber`



```
PS C:\WINDOWS\system32\cmd.exe - ansicon
C:\Cuke>cucumber
Feature: Intranet Telephone Search

  Scenario Outline: Search for Howell in the Quick Telephone Search    # Feature
  s\telephonesearch.feature:3
    Given I navigate to the corporate home page                      # Feature
  s\step_definitions\step_telephonesearch.rb:22
    When I enter the name of "<input>" into the telephone search Field # Feature
  s\step_definitions\step_telephonesearch.rb:26
      Then the text "<output>" appears on the screen                 # Feature
  s\step_definitions\step_telephonesearch.rb:31

  Examples:
  | input | output |
  | Howell | BARBARA |
  | Moore | Andy |
  | Dart | JULIA |

  3 scenarios (3 passed)
  9 steps (9 passed)
  0m21.562s

C:\Cuke>
```

As you can see, 3 scenario's and 9 steps have passed just like before.

### 3.5 Argument Exercise

Let's go back to the previous **Information** box and look at the **argument**. Cucumber has called the argument **arg1** (obviously because it's the 1<sup>st</sup> argument that it's encountered).

Can you think of an appropriate word to call the argument? **TIP** - The word cannot start with a capital letter, all arguments start with lowercase.

Try it! Run Cucumber and look at your results.



### 3.6 Tags

As the number of feature and scenarios grow we may want to run a single, a couple or just a few specific scenarios.

The simplest way to do this is to tag scenarios.

You tag a scenario by putting a word prefixed with the @ character on the line before the Scenario keyword, for example like this:

A screenshot of a Windows Notepad window titled "telephonesearch.feature - Notepad". The window contains the following text:

```
Feature: Intranet Telephone Search

@FastRun
Scenario: Search for Howell in the Quick Telephone Search
  Given I navigate to the corporate home page
  When I enter the name of "Howell" into the telephone search field
  Then the text "BARBARA" appears on the screen
```

Please Note: There is no gap between the @ and the name of the tag e.g. @FastRun.

If we were testing lots of different types of searches on the corporate homepage, we could tag all the scenarios that were most important, name them '@FastRun' and run those tagged scenarios in Cucumber.

We could take other scenarios with less importance and tag them with '@LongRun' and run them separately.

To run Tagged scenarios in Cucumber the command you would need to use in the command screen would be:

C:\ Cuke> cucumber - -tags @FastRun or C:\ Cuke> cucumber - -tags @LongRun.

### 3.7 Running Specific Scenarios



Feature will normally contain multiple scenarios, which can be quite time consuming to run through, especially if you're trying to make the last scenario pass!

- Create a new feature file in `c:\cuke\features` called `incomplete.feature`
- Copy in the following content:

```
Incomplete.feature - Notepad
File Edit Format View Help
Feature: Intranet Telephone Search
Scenario: Incomplete Scenario
Given I navigate to the corporate home page
When I do something

Scenario: Incomplete Scenario 2
Given I navigate to the corporate home page
When I do something else
```

- If you now run the `cucumber` command, it will run through all the tests and report back the missing step.

```
C:\WINDOWS\system32\cmd.exe
0m26.245s

You can implement step definitions for undefined steps with these snippets:

When(/^I do something$/) do
  pending # express the regexp above with the code you wish you had
end

When(/^I do something else$/) do
  pending # express the regexp above with the code you wish you had
end

C:\Cuke>_
```

In this example it took 26 seconds to run (your time may vary), it ran both the `telephonesearch.feature` and `incomplete.feature`. This time build up as you add more feature files.

There are ways to be more specific on the scenarios you want to run:

- This time run the command `cucumber features/incomplete.feature`



```
PS C:\WINDOWS\system32\cmd.exe
C:\Cuke>cucumber features/incomplete.feature
including Capybara::DSL in the global scope is not recommended!
Feature: Intranet Telephone Search

  Scenario: Incomplete Scenario          # features\incomplete.feature:3
    Started InternetExplorerDriver server (32-bit)
    2.31.0.0
    Listening on port 5555
      Given I navigate to the corporate home page # features/step_definitions/step_search.rb:22
        When I do something                      # features\incomplete.feature:5

  Scenario: Incomplete Scenario 2         # features\incomplete.feature:7
    Given I navigate to the corporate home page # features/step_definitions/step_search.rb:22
      When I do something else                 # features\incomplete.feature:9

2 scenarios (2 undefined)
4 steps (2 undefined, 2 passed)
0m3.850s

You can implement step definitions for undefined steps with these snippets:

When(/^I do something$/) do
  pending # express the regexp above with the code you wish you had
end

When(/^I do something else$/) do
  pending # express the regexp above with the code you wish you had
end

C:\Cuke>_
```

This now only ran scenarios in the `incomplete.feature` file and completed in 3 seconds.

You can extend this even further to just run specific scenarios in a feature file, this is done by appending the line number of the scenario to the command.

Notepad isn't brilliant for displaying this information. Alternatively you can use notepad++ which will give this information:

**Notepad++ is available from <http://notepad-plus-plus.org/>**



## Chapter

## 4

The screenshot shows a Notepad++ window with the file 'incomplete.feature' open. The content of the file is:

```
1 Feature: Intranet Telephone Search
2
3 Scenario: Incomplete Scenario
4 Given I navigate to the corporate home page
5 When I do something
6
7 Scenario: Incomplete Scenario 2
8 Given I navigate to the corporate home page
9 When I do something else
```

The status bar at the bottom of the Notepad++ window displays: length : 239 lines : Ln : 9 Col : 25 Sel : 0 | 0 Dos\Windows ANSI

Incomplete Scenario 2 is on line 7.

- To run only this scenario, type the following: `cucumber features/incomplete.feature:7`



The screenshot shows a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command 'cucumber features/incomplete.feature:7' is run, which includes a warning about using Capybara::DSL in the global scope. It then runs a feature named 'Intranet Telephone Search' with one scenario named 'Incomplete Scenario 2'. This scenario has two steps: 'Given I navigate to the corporate home page' and 'When I do something else'. Both steps are marked as undefined. The output shows 1 scenario (1 undefined) and 2 steps (1 undefined, 1 passed). A note at the bottom suggests implementing step definitions for undefined steps with snippets, and provides a pending step definition example.

```
C:\Cuke>cucumber features/incomplete.feature:7
including Capybara::DSL in the global scope is not recommended!
Feature: Intranet Telephone Search

  Scenario: Incomplete Scenario 2          # features\incomplete.feature:7
    Started InternetExplorerDriver server (32-bit)
    2.31.0.0
    Listening on port 5555
      Given I navigate to the corporate home page # features/step_definitions/step_search.rb:22
      When I do something else                      # features\incomplete.feature:9

  1 scenario (1 undefined)
  2 steps (1 undefined, 1 passed)
  0m3.242s

You can implement step definitions for undefined steps with these snippets:

When(/^I do something else$/) do
  pending # express the regexp above with the code you wish you had
end

C:\Cuke>
```

As you can see, it only ran 1 scenario, which is the one we specified by the line number.

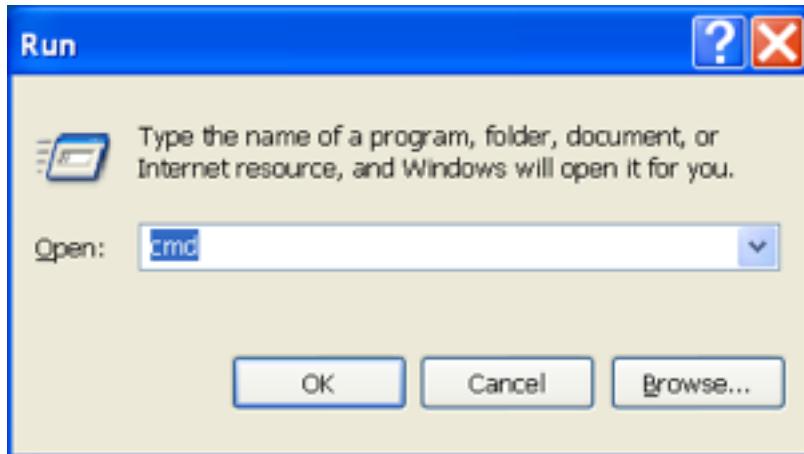
## Ruby Gems

Ruby has a very large function library available, which are called Gems. These Gems allow you to extend your ruby/cucumber scripts further to allow for additional functionality in your scripts.

We have already used two gems, Cucumber and Capybara. Cucumber gives us the framework to use the Given, When and Then statements. Capybara gives us the ability to drive web sites.

### 4.1 Installed Gems

- Press both the Windows button and the letter 'r' on your keyboard at the same time. The prompt below should appear.
- When prompted, type "cmd" and click OK.



- C:\Documents and Settings\CS811AM> will be displayed. (Obviously your own CS sign on will be displayed).
- The Command Prompt Screen will respond as follows:



A screenshot of a Microsoft Windows XP Command Prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window displays the following text:  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
C:\Documents and Settings\CS811AM>~

- Go to the C drive root, by changing the directory.
- Type `cd..` press enter.
- Repeat the command `cd..` press enter.
- Now you should be at the base of the C Drive

A screenshot of a Microsoft Windows XP Command Prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window displays the following text:  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
C:\Documents and Settings\CS811AM>cd..  
C:\Documents and Settings>cd..  
C:\>

- Type `gem query --local` (that is `double dash` then the word 'local' with no gaps)



The screenshot shows a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The command 'gem query --local' has been run, displaying a list of locally installed Ruby gems and their versions. The output is as follows:

```
C:\>gem query --local

*** LOCAL GEMS ***

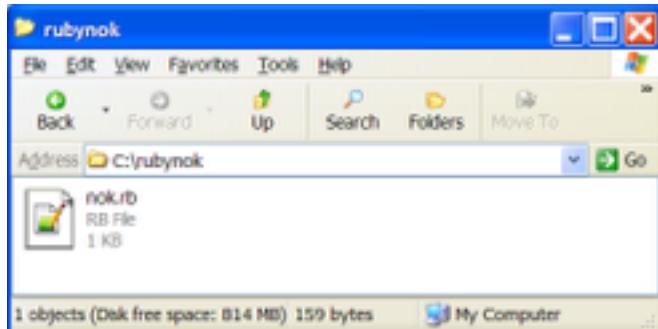
actionmailer (3.2.13)
actionpack (3.2.13)
activemodel (3.2.13)
activerecord (3.2.13)
activerecord-sqlserver-adapter (3.2.10)
activeresource (3.2.13)
activesupport (3.2.13)
akami (1.2.0)
annotate (2.5.0)
arel (3.0.2)
bcrypt-ruby (3.0.1 x86-mingw32)
bigdecimal (1.2.1, 1.1.0)
bootstrap-sass (2.1.0.0)
bootstrap-will_paginate (0.0.6)
builder (3.0.4)
bundler (1.3.4)
capybara (2.2.1, 2.1.0, 1.1.2)
capybara-webkit (0.14.2)
childprocess (0.3.9)
clipboard (1.0.5)
coffee-rails (3.2.2)
coffee-script (2.2.0)
coffee-script-source (1.6.2, 1.6.1)
cucumber (1.3.10, 1.3.2, 1.3.1)
cucumber-rails (1.2.1)
database_cleaner (0.7.0)
debugger-ruby_core_source (1.2.2)
diff-lcs (1.2.4, 1.1.3)
epoxy (0.3.1)
erubis (2.7.0)
eventmachine (1.0.3 x86-mingw32)
```

This lists all the gems installed and their version. In the example above, you can see that three different versions of the cucumber gem is installed (versions: 1.3.10, 1.3.2, 1.3.1)

**Information** The Gems listed above may differ from the Gems listed for your pc.

## 4.2 Installing new Gem

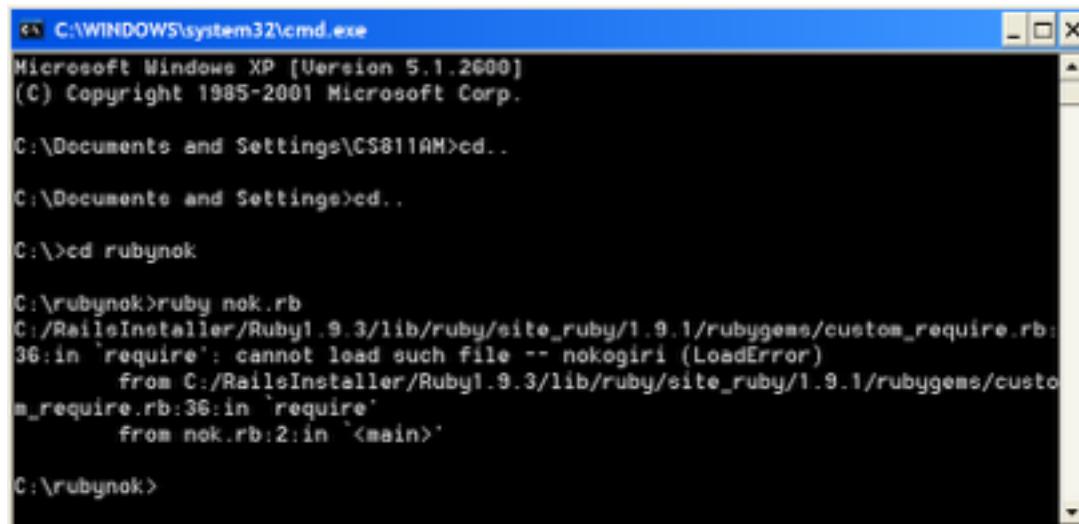
- Now create a new ruby file (e.g. [nok.rb](#)). Save this file in [c:\rubynok](#)



- Now open this file and type the follow:



- Open a command prompt and navigate to the [rubynok](#) folder (using the cd commands). Then type [ruby nok.rb](#) and press enter



gentl reported an error message saying that it could not load the nokogiri gem. We now need to install this gem.

- Now type: [gem install nokogiri](#) and press enter (nokogiri is all lower case)

**Note:**

Gem names are case sensitive



```
ex C:\WINDOWS\system32\cmd.exe
C:\>cd rubynok

C:\rubynok>ruby nok.rb
C:/RailsInstaller/Ruby1.9.3/lib/ruby/site_ruby/1.9.1/rubygems/custom_require.rb:36:in `require': cannot load such file -- nokogiri (LoadError)
    from C:/RailsInstaller/Ruby1.9.3/lib/ruby/site_ruby/1.9.1/rubygems/custom_require.rb:36:in `require'
    from nok.rb:2:in `<main>'

C:\rubynok>gem install nokogiri
Fetching: nokogiri-1.6.1-x86-mingw32.gem (100%)
Successfully installed nokogiri-1.6.1-x86-mingw32
1 gem installed
Installing ri documentation for nokogiri-1.6.1-x86-mingw32...
Installing RDoc documentation for nokogiri-1.6.1-x86-mingw32...

C:\rubynok>
```

- This has now installed the nokogiri gem.
- Now run the ruby script again by typing: `ruby nok.rb`

```
ex C:\WINDOWS\system32\cmd.exe
36:in `require': cannot load such file -- nokogiri (LoadError)
    from C:/RailsInstaller/Ruby1.9.3/lib/ruby/site_ruby/1.9.1/rubygems/custom_require.rb:36:in `require'
    from nok.rb:2:in `<main>'

C:\rubynok>gem install nokogiri
Fetching: nokogiri-1.6.1-x86-mingw32.gem (100%)
Successfully installed nokogiri-1.6.1-x86-mingw32
1 gem installed
Installing ri documentation for nokogiri-1.6.1-x86-mingw32...
Installing RDoc documentation for nokogiri-1.6.1-x86-mingw32...

C:\rubynok>ruby nok.rb
<title>
  Land Registry - Land Registry for England and Wales
</title>

C:\rubynok>
```

The script has connected to the landregistry.gov.uk website and extract the title of that web site using the nokogiri function library.

#### 4.3 Installing an old version of a gem

Gems normally depend on other gems, and as these gems can change often, you may not want to install the latest version of a gem. This part will explain how to install an old version of a gem:

- Open a command prompt and navigate to the `rubynok` folder (using the `cd` commands)



- Then type `gem install nokogiri -v 1.5` and press enter

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\CS811AM>cd..

C:\Documents and Settings>cd..

C:\>cd rubynok

C:\rubynok>gem install nokogiri -v 1.5
Fetching: nokogiri-1.5.0-x86-mingw32.gem (100%)
Successfully installed nokogiri-1.5.0-x86-mingw32
1 gem installed
Installing ri documentation for nokogiri-1.5.0-x86-mingw32...
Installing RDoc documentation for nokogiri-1.5.0-x86-mingw32...

C:\rubynok>
```

This will install the version 1.5 gem of Nokogiri.

- Now edit the `nok.rb` file as follows:

```
nok.rb - Notepad
File Edit Format View Help
require 'rubygems'
gem "nokogiri", "~> 1.5"
require 'nokogiri'
require 'open-uri'

page = Nokogiri::HTML(open("http://www.landregistry.gov.uk"))
puts page.xpath('//title')
```

**Note:**

By default, ruby will use the latest gem. You need to specify the gem name and version of the old gem before you require the gem.

- Now type: `gem install nokogiri` and press enter (nokogiri is all lower case)

Although it appears no different, it is instead using an older version of the nokogiri gem.

- Type `gem query --local` (that is double dash then the word local)



```
C:\> C:\WINDOWS\system32\cmd.exe
C:\rubynok>gem query --local

*** LOCAL GEMS ***

actionmailer (3.2.13)
actionpack (3.2.13)
activemodel (3.2.13)
activerecord (3.2.13)
activerecord-sqlserver-adapter (3.2.10)
activeresource (3.2.13)
activesupport (3.2.13)
akami (1.2.0)
annotate (2.5.0)
archive-tar-minitar (0.5.2)
arel (3.0.2)
bcrypt-ruby (3.0.1 x86-mingw32)
bigdecimal (1.2.1, 1.1.0)
```

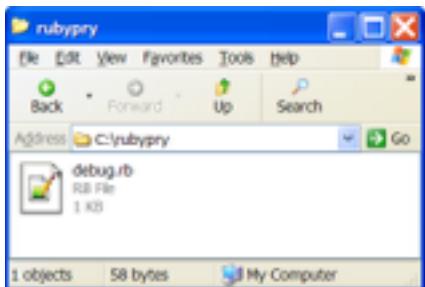
You can see the nokogiri line saying that two versions are installed:

```
C:\> C:\WINDOWS\system32\cmd.exe
multi_json (1.7.7, 1.7.3, 1.7.1)
multi_test (0.0.2)
net-ssh (2.6.8)
nokogiri (1.6.1 x86-mingw32, 1.5.0 x86-mingw32)
nori (2.3.0, 2.1.0)
pg (0.14.1 x86-mingw32, 0.12.2 x86-mingw32)
poltergeist (1.3.0)
```

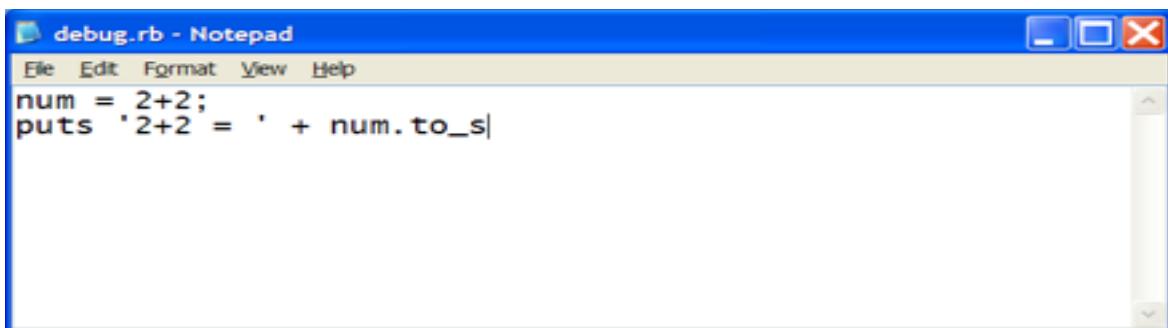


#### 4.4 Ruby Debugging Gem

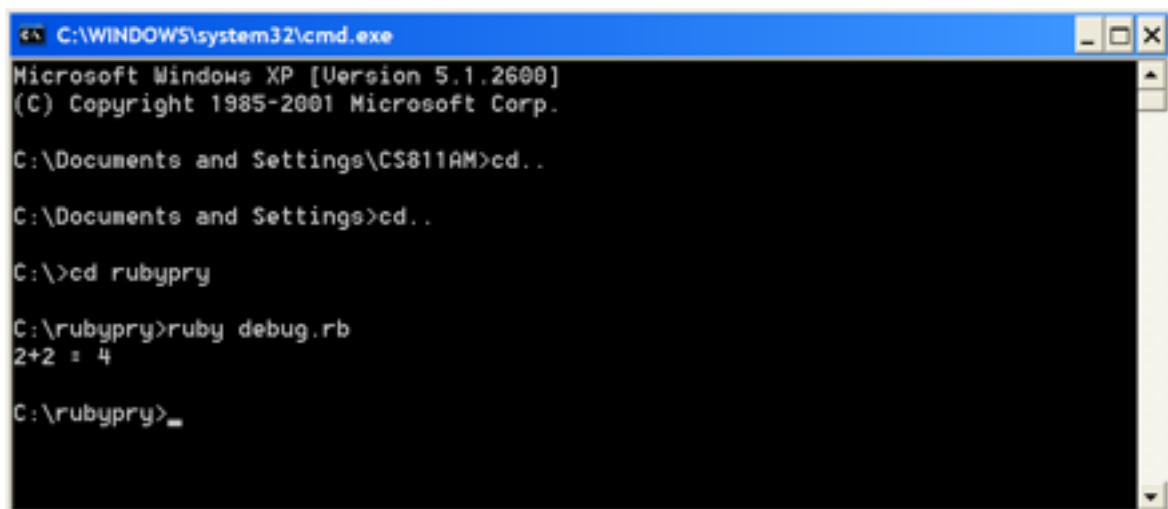
- Now create a new ruby file (e.g. `debug.rb`). Save this file in `c:\rubypy`



- Now open this file and type the follow:



- Open a command prompt and navigate to the `rubypy` folder (using the `cd` commands)
- Then type `ruby debug.rb` and press enter



This has put the value of 4 out into the console



- Now install the pry gem using the command: `gem install pry`

```
C:\rubypy>gem install pry
Fetching: coderay-1.1.0.gem (100%)
Fetching: slop-3.4.7.gem (100%)
Fetching: method_source-0.8.2.gem (100%)
Fetching: win32console-1.3.2-x86-mingw32.gem (100%)
Fetching: pry-0.9.12.6-i386-mingw32.gem (100%)
Successfully installed coderay-1.1.0
Successfully installed slop-3.4.7
Successfully installed method_source-0.8.2
Successfully installed win32console-1.3.2-x86-mingw32
Successfully installed pry-0.9.12.6-x86-mingw32
5 gems installed
Installing ri documentation for coderay-1.1.0...
Installing ri documentation for slop-3.4.7...
Installing ri documentation for method_source-0.8.2...
Installing ri documentation for win32console-1.3.2-x86-mingw32...
Installing ri documentation for pry-0.9.12.6-x86-mingw32...

RDoc::Parser::Ruby failure around line 25 of
lib/pry/commands/ls.rb

Before reporting this, could you check that the file you're documenting
has proper syntax:

c:/Ruby193/bin/ruby.exe -c lib/pry/commands/ls.rb
```

**Information** You may have slightly different wording depending on what is already installed on your pc, the main object is that some gems have been installed.

- Modify your `debug.rb` to look like the following:

```
debug.rb - Notepad
File Edit Format View Help
require 'pry'

num = 2+2;

binding.pry

puts '2+2 = ' + num.to_s
```

- In the command prompt, type `ruby debug.rb` again and press enter:



```
C:\WINDOWS\system32\cmd.exe - ruby debug.rb
C:\rubypy>ruby debug.rb

From: C:/rubypy/debug.rb @ line 5 :

  1: require 'pry'
  2:
  3: num = 2+2;
  4:
=> 5: binding.pry
  6:
  7: puts '2+2 = ' + num.to_s

[1] pry(main)> _
```

The script has paused at the “binding.pry” command. You can now execute ruby code to manipulate the environment.

- Type the command: `num = 6` and press enter

```
C:\WINDOWS\system32\cmd.exe - ruby debug.rb

From: C:/rubypy/debug.rb @ line 5 :

  1: require 'pry'
  2:
  3: num = 2+2;
  4:
=> 5: binding.pry
  6:
  7: puts '2+2 = ' + num.to_s

[1] pry(main)> num = 6
=> 6
[2] pry(main)>
```

We have now changed the value of num to be 6, this is after the command that initially set num to be 4 (`2+2`). So if we let the script finish, it should no longer say `2+2 = 4`, but instead say `2+2 = 6`.

- Type the command `exit` and press enter.



The screenshot shows a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.exe'. The window contains the following Ruby code:

```
1: require 'pry'
2:
3: num = 2+2;
4:
=> 5: binding.pry
6:
7: puts '2+2 = ' + num.to_s

[1] pry(main)> num = 6
=> 6
[2] pry(main)> exit
2+2 = 6

c:\rubypy>
```

After typing `exit`, the script will continue to run any remaining code, as you can see, it now says `2+2 = 6`.

#### 4.5 More Gems

You can find more ruby gems at: <http://rubygems.org/>

## HTML, XML and XPATH

What is HTML?

“HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.”

What is XPATH

“XPATH is a query language for selecting nodes from an XML or HTML document.”

HTML and XML are very similar and fairly human readable, they comprise of a set of tags that a machine understands.



The difference between XML and HTML is that with XML you can create your own tags, HTML have defined tags for display web pages.

XPATH is a way of accessing the tags on a HTML/XML document, and is very useful when using capybara with cucumber. Depending how the application has been developed the XPATH paths can be relatively simple, or very complex.

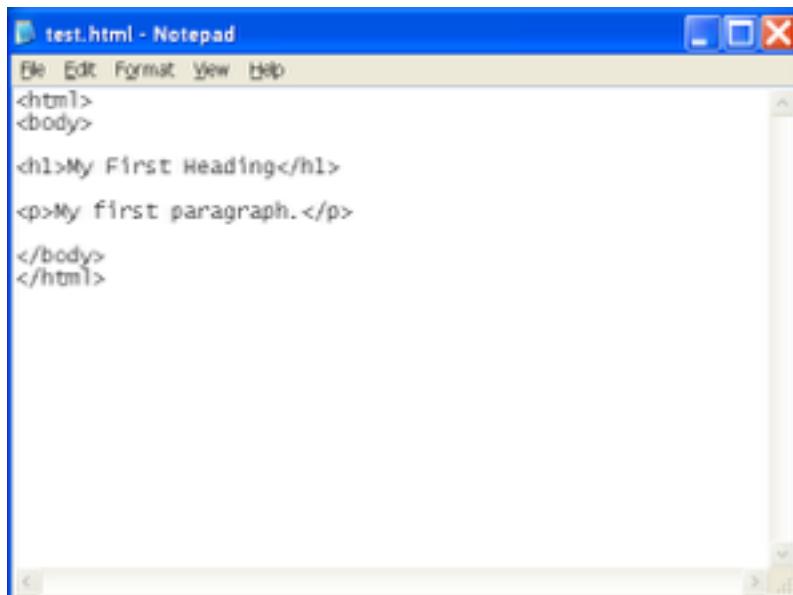
### **Software requirements:**

- Firefox Installed
- Firebug addon
- FirePath addon

### **5.1 HTML**

Using HTML we will guide you through creating your own web page.

- Create a folder on your F Drive called “HTML”.
- Open notepad and write the following:



A screenshot of a Windows Notepad window titled "test.html - Notepad". The window shows the following HTML code:

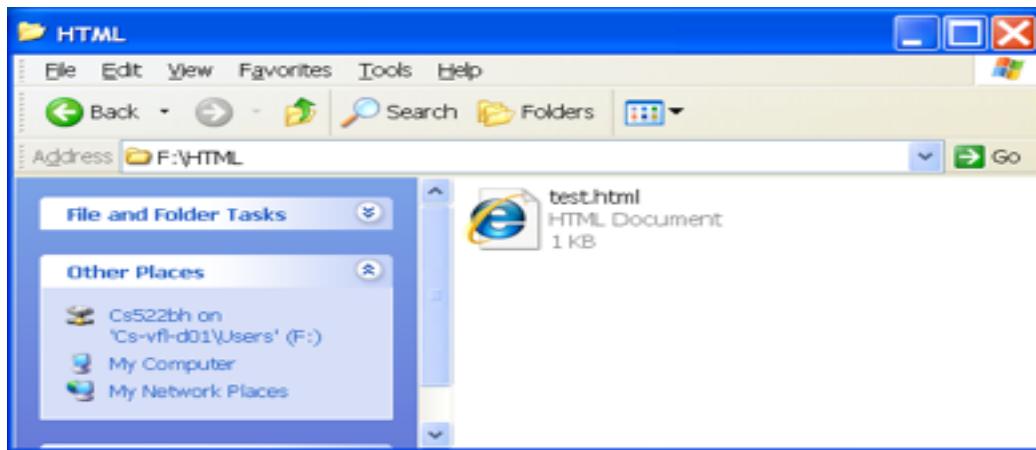
```
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

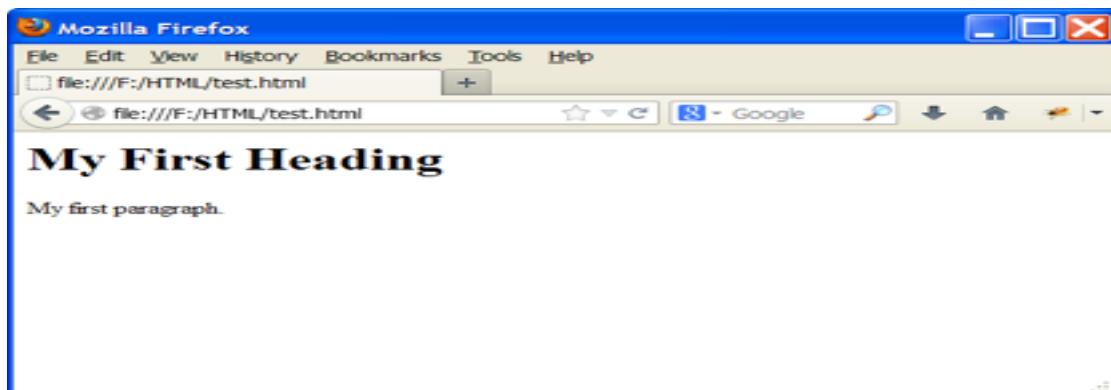
</body>
</html>
```



- Save the document as File name: test.html, Save as Type: All Files within the HTML Folder in your F Drive.
- To edit the document, right click and choose edit
- Once saved the document should look like this:



- Open up Firefox and navigate to the url: <file:///F:/HTML/test.html> and this is what you should see.



The web browser has interpreted the HTML code and then translated this to display a web page.

HTML uses nesting of tags, below is an outline of how the code is structured:



```
<html>  
<body>  
<h1>This a heading</h1>  
  
<p>This is a paragraph.</p>  
</body>  
</html>
```



### Information

All html documents have a <html> and <body> which encompasses all the html code on the page. This is a defined structure for html documents.

The <h1> and <p> tags inform the web browser that the text between the start tag (e.g. <h1>) and end the end tag (</h1>) will be displayed in the style that the tag dictates.

- Open up the test.html document in notepad.
- Add after the </p> tag the following code:

```

```

- The test.html file now looks like:

A screenshot of a Windows Notepad window titled "test.html - Notepad". The window shows the following HTML code:

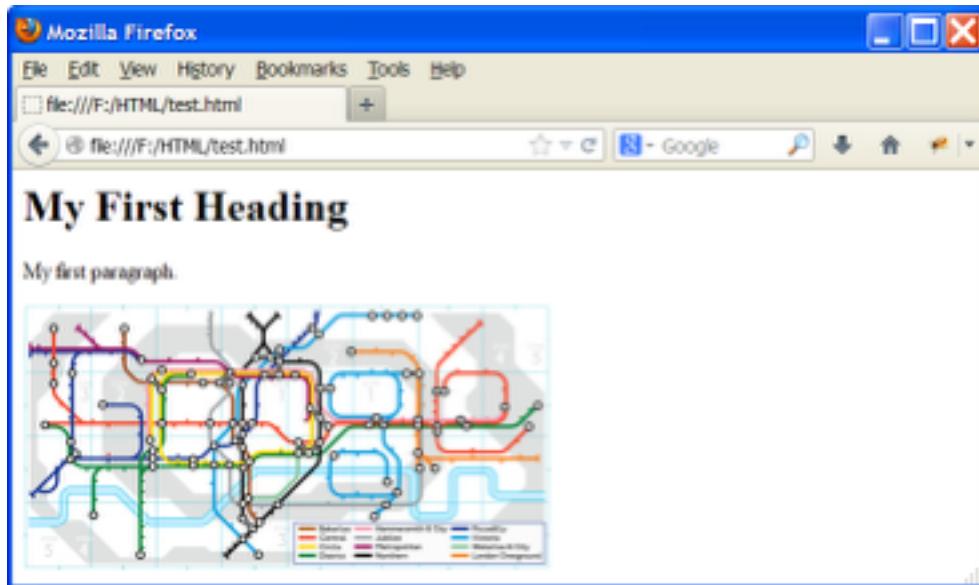
```
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>


</body>
</html>
```

The code includes an image tag with a URL pointing to a Google logo for the 150th anniversary of the London Underground.

- Reopen firefox and again navigate to <file:///F:/HTML/test.html> it will display as:



If we now try to make the google image a link to go to www.google.com we do:

- Amend the text.html file so it looks like:

```
test.html - Notepad
File Edit Format View Help
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
<a href="http://www.google.com">

</a>
</body>
</html>
```

We have wrapped the image around a hyperlink (<a> tag).

All the text between <a href..."> and </a> will now be a link.

- Refresh your web page.

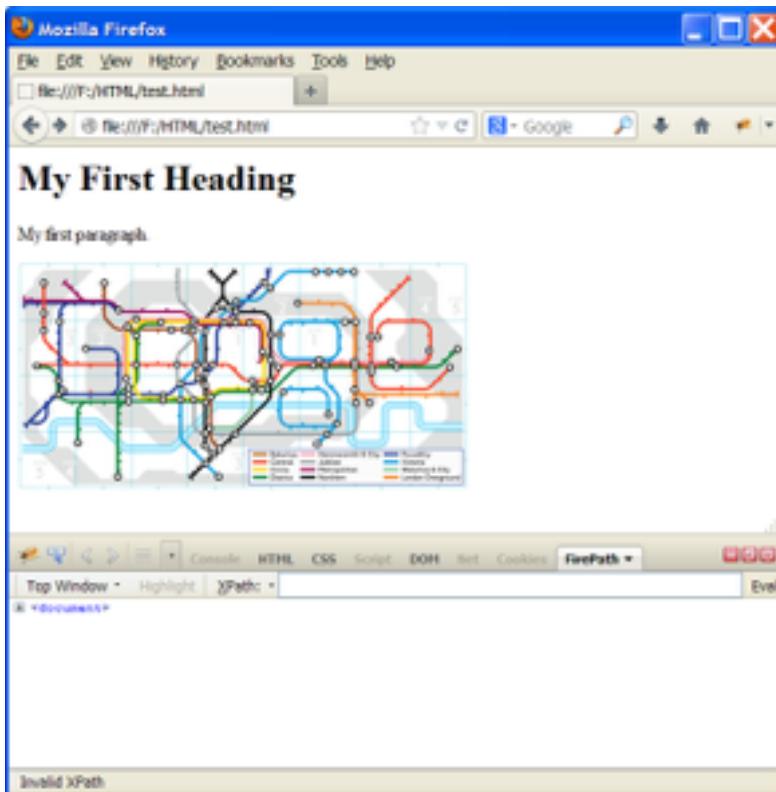
It should appear no different, **but** you can now click on the image and it will take you to google.com



## 5.2 XPATH

Capybara sometimes has difficulties in identifying what you want to do, an example is automating the clicking of the image to take you to google will involve using XPATH.

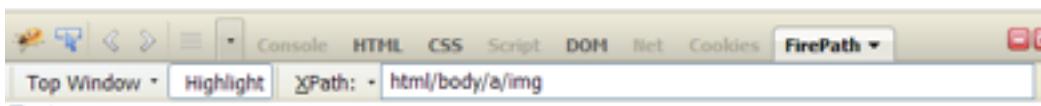
- Using firefox visit the web page you just created and press F12.
- A debug window will appear at the bottom of the screen.
- Click on the tab called FirePath. (Final tab on the toolbar)



Using the second icon to the left you can now select items on the page to find their location.



- Hover over the image, it will appear as:



The image has an xpath of: `html/body/a/img`

- Now create a ruby script using capybara to drive this web page:

```
rubytest.rb - Notepad
File Edit Format View Help
### START HEADER - DON'T TOUCH ###
require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end

### END HEADER - DON'T TOUCH ###

visit "http://intranet/cs/Solutions/testPractice/xpath/example1.html"
find(:xpath, "html/body/a/img").click
```

(Due to restrictions in running local files using capybara, I have put the same code on the intranet, this is why the visit path doesn't match the code you saved on your F Drive).

As explained in the previous part of the guide, the header part configures capybara to use internet explorer and requires the gems necessary to execute capybara.



The code, below the header line, visits the web page and then finds the image on the page and then clicks on it. It uses the xpath to find where the image is. If you save the script as rubytest.rb and then using the command line, navigate (using the 'cd' command) and execute the script by running: `ruby rubytest.rb` .

You should see that it drives the web page and takes you to google.

Although this works, the xpath is dependent on the page not changing. If you were to move the image to another location then the ruby script would break. Also it is very difficult to understand from the line "`find(:xpath, "html/body/a/img").click`" what exactly it is doing.

The solution to this is to modify our code to add an id to the image. If we open up test.html and amend the image tag:

```
test.html - Notepad
File Edit Format View Help
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
<a href="http://www.google.com">

</a>
</body>
</html>
```

Save that and open it in firefox, again press F12 and use the firepath tab. Highlight the image using the button and it should appear as:

You can now see that it has picked up the id of the image, and has used that as the xpath instead.

- Now update our ruby script with this new id:

A screenshot of a Windows Notepad window titled "rubytest.rb - Notepad". The window contains Ruby code for a Capybara test setup. The code includes require statements for 'capybara', 'capybara/dsl', and 'rspec', and sets up Capybara with 'localhost:4567' as the host, CSS as the default selector, Selenium as the default driver, and a 10-second default wait time. It also registers Selenium as the driver for Internet Explorer.

```
### START HEADER - DON'T TOUCH ###
require 'capybara'
require 'capybara/dsl'
require 'rspec'

include Capybara::DSL

Capybara.app_host = 'localhost:4567'
Capybara.default_selector = :css
Capybara.default_driver = :selenium
Capybara.default_wait_time = 10

Capybara.register_driver :selenium do |app|
  Capybara::Selenium::Driver.new(app, :browser => :internet_explorer)
end
```

**Important**

Also change the wording in the rubytest.rb – Notepad.

Where the previous file stated visit "<http://intranet/cs/Solutions/testPractice>xpath example1.html>" to "<http://intranet/cs/Solutions/testPractice>xpath example2.html>"

The script is slightly more descriptive in what it is doing. And also if the image was to move on the page, the script would still work.



# Jenkins Driven Test

Jenkins is a powerful but simple to use tool. It will build code, run tests and report results in a visual way.

In a nutshell, Jenkins provides an easy-to-use so-called continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

The automated, continuous build increases the productivity.

## 6.1 Background Information

- A ClearCase project has been set up named Jenkins\_Dev\_Int.
- A project named ***Training Env X*** has been created but not configured in Jenkins.

## 6.2 Navigate to Jenkins

- Navigate to <http://lrg00101613:8080> in a web browser. Jenkins looks like this



Dashboard [Jenkins] - Windows Internet Explorer  
<http://192.168.1.13:8080/>

File Edit View Favorites Tools Help

Favorites | [Free Hotmail](#) [HP ALM - Quality Center 11](#) [HP ALM - Quality Center 11 ...](#) [ISG Test - DevWiki](#) [Uniform Test Practice - Dev...](#)

[Dashboard \[Jenkins\]](#)

Jenkins

ENABLE AUTO REFRESH [add description](#)

New Job People Build History Message Jenkins Wall Display

Build Queue No builds in the queue.

Build Executor Status 1 idle

| S | W | Name                 | Last Success       | Last Failure       | Last Duration |
|---|---|----------------------|--------------------|--------------------|---------------|
|   |   | RTC_CI               | N/A                | 4 hr 50 min (±1)   | 13 sec        |
|   |   | RTC_Sandbox          | 21 days (±5)       | 21 days (±1)       | 1 min 45 sec  |
|   |   | RTC_DataShare        | N/A                | N/A                | N/A           |
|   |   | malista.initial.test | 3 mo 25 days (±10) | 3 mo 25 days (±5)  | 1 min 51 sec  |
|   |   | E2SS_SharePoint      | 2 mo 21 days (±12) | 2 mo 21 days (±12) | 1 min 58 sec  |
|   |   | E2SS_OCRM_Test       | 4 mo 9 days (±11)  | 2 mo 26 days (±15) | 13 sec        |
|   |   | ScalQ_Project        | 5 mo 7 days (±13)  | 2 mo 2 days (±12)  | 2 min 24 sec  |
|   |   | Training_Env_1       | N/A                | 1 day 2 hr (±2)    | 1 min 40 sec  |
|   |   | Training_Env_10      | N/A                | 2 days 1 hr (±1)   | 14 min        |
|   |   | Training_Env_11      | N/A                | 1 day 1 hr (±1)    | 2 min 15 sec  |
|   |   | Training_Env_12      | N/A                | 2 days 0 hr (±1)   | 15 min        |
|   |   | Training_Env_2       | N/A                | 1 day 0 hr (±1)    | 2 min 10 sec  |
|   |   | Training_Env_3       | N/A                | 1 day 1 hr (±2)    | 2 min 20 sec  |
|   |   | Training_Env_4       | N/A                | 1 day 2 hr (±2)    | 2 min 2 sec   |
|   |   | Training_Env_5       | N/A                | 2 days 2 hr (±1)   | 16 min        |
|   |   | Training_Env_6       | N/A                | 1 day 20 hr (±1)   | 1 min 45 sec  |
|   |   | Training_Env_7       | N/A                | 1 day 3 hr (±2)    | 1 min 43 sec  |

Local intranet 100% [View Source](#)

- Let's have a look at the information displayed

### 6.3 Status, Weather and Name



| All | Radiator Tab | wall  |
|-----|--------------|---|
| S   | W            | Name ↓  |
|     |              | <a href="#">Agricultural Credits Dev (CI)</a>             |
|     |              | <a href="#">Agricultural Credits Test 1</a>               |
|     |              | <a href="#">AutoDeploy Proof Of Concept - Dont Delete</a> |

- ‘S’ is for Status.  
It’s a simple traffic light indicator of what’s going on with your project.  
**Red** - failed.  
**Blue** - passed.  
Grey - obsolete.
- ‘W’ is for Weather report.  
It’s a very rough guide, the sunnier the weather the better your project is going. Hovering over the symbols will tell you about build stability, a summary of how many builds have failed or passed in recent history.
- ‘N’ is for Name which is self explanatory.

The last three columns refer to test runs. It states when the last test run was successful, when it last failed and how long it took.

| Last Success        | Last Failure        | Last Duration |
|---------------------|---------------------|---------------|
| 3 mo 26 days (#698) | 1 mo 26 days (#711) | 42 min        |
| N/A                 | 1 mo 26 days (#2)   | 7 min 10 sec  |
| 3 mo 28 days (#25)  | 1 mo 26 days (#27)  | 1 min 54 sec  |



## 6.4 Select your project

- Single click over the project name (Project Training Env X), your view should shift to:

### Information

This screen gives you basic information and is a read only page, to get a bit further you need to log-in.

## 6.5 Log-in

- Select the log in icon at the top right hand side of the page use your usual RACF and password.
- select the log in button and this page should be displayed:



## 6.6 Configure a project

The next step is to configure the project, the configure icon is at the bottom of the list on the top left hand side of the screen.



- Select the Configure Icon

## 6.7 Project Name

- The first thing displayed is Project Name, the project name should be in place by default.



Training Env X Config [Jenkins] - Windows Internet Explorer  
<http://lrg00381613:8080/job/training%20env%20/configure>

File Edit View Favorites Tools Help

Favorites Free Hotmail HP ALM - Quality Center 11... HP ALM - Quality Center 11... ISG Test - DevWiki Uniform Test Practice - Dev...

Training Env X Config [Jenkins]

**Jenkins**

Project name: Training Env X

Description: A description has been added

Back to Dashboard Status Changes Workspace Save Apply

Error on page Local intranet 100%

## 6.8 Source Code Management

- The next thing to look at is the **Source Code Management**.

### Warning

**Do not amend the settings; changing some settings can cause serious**

#### Source Code Management

|   |   |
|---|---|
| <input type="radio"/> Base ClearCase                |   |
| <input type="radio"/> CVS                           |   |
| <input type="radio"/> CVS Projectset                |   |
| <input type="radio"/> ClearCase UCM                 |   |
| <input type="radio"/> None                          |   |
| <input type="radio"/> Subversion                    |   |
| <input checked="" type="radio"/> UCM ClearCase      |   |
| View tag  | Jenkins_dev_Int_1   |
| View path   | Jenkins_dev_Int   |
| Stream selector                                     | Jenkins_dev_Int@\LR_pvob  |
| <input type="checkbox"/> Define load rules manually |   |
| Use update  | <input checked="" type="checkbox"/> If checked, Hudson will use 'cleartool update' whenever possible, r                               |
| Changestet  | <input type="radio"/> No History<br><input checked="" type="radio"/> Current stream<br><input type="radio"/> Current stream + rebases |

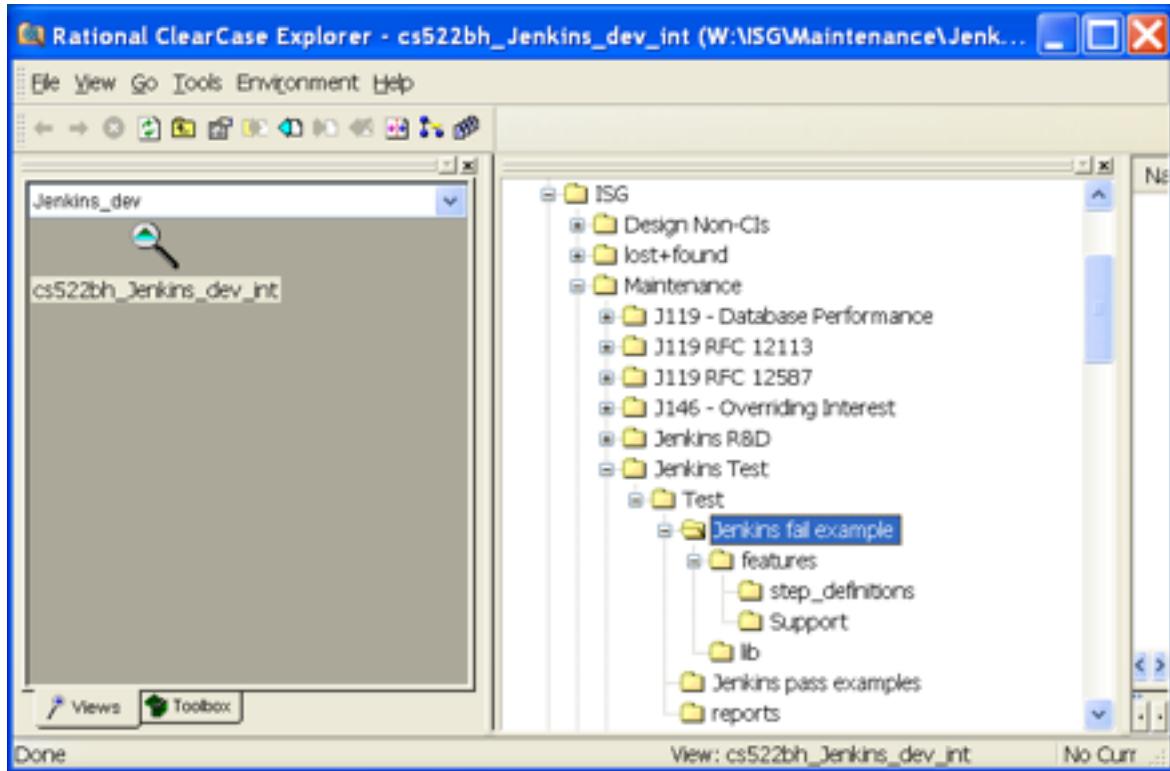
### Important

**You do not need to change the values above, these have been configured for you.**

- The source code is UCM ClearCase . Take a look at the project in ClearCase.

### Warning

Clearcase may not work in the training environment. Clearcase is just used to show where the tests are stored. This step can be skipped.



### Information

For now, just take a look at the file structure. Note how the `step_definitions` folder is placed inside the `Features` folder and the `step_karate.rb` file is placed inside the `step definitions` folder.

## 6.9 Add a Build Step

- Return to Jenkins, scroll down and pass Build Triggers, Build Environment and Build.
- Click on Add a build step. A new menu is displayed.



The screenshot shows the Jenkins configuration interface for a job named 'Ian Test Config'. The 'Build' section contains a step to execute a Windows batch command:

```
cd "Jenkins_dev_1st\200-Maintenance\Jenkins Test\Test\Jenkins fail example"
cucumber --format junit --out ../reports/
```

The 'Post-build' section shows a dropdown menu with options like 'Execute Windows batch command', 'Execute shell', 'Execute shell script on remote host using ssh', 'HP Quality Center', 'Inject environment variables', 'Invoke Ant', 'Invoke Rake', and 'Invoke top-level Maven targets'. The 'Execute Windows batch command' option is currently selected.

- Click on Execute Window Batch Command.

## 6.10 Enter a Command

- Fill in the Build step as you would create a command line.



Training Env 12 Config [Jenkins] - Windows Internet Explorer  
 http://hq00101613:8080/job/Training%20Env%2012/configure

File Edit View Favorites Tools Help

Favorites Training Env 12 Config [Jenkins]

Jenkins > Training Env 12 > configuration

Execute Windows batch command

Command: cd "Jenkins\_Dev\_Inst\IDS\Maintenance\Jenkins Test\Test\Jenkins fail example"  
 mvn test --fail-on-error --format junit --out ..\reports\ || Exit 0

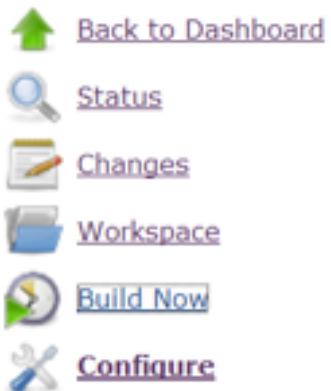
Save Apply

Error on page.

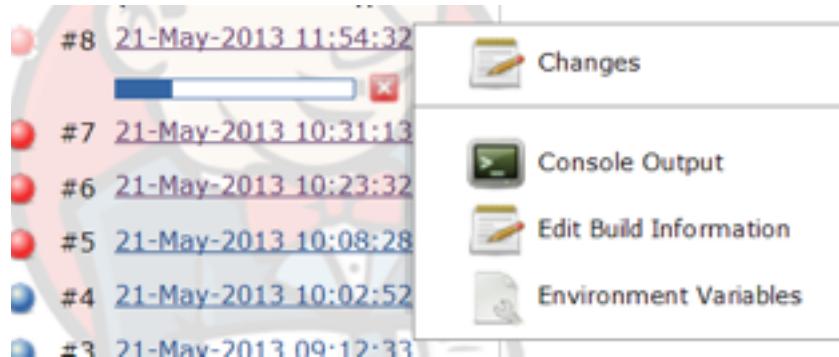
- To complete the configuration step simply select Save at the bottom of the page.

## 6.11 Running the Test

- Select the Build Now icon



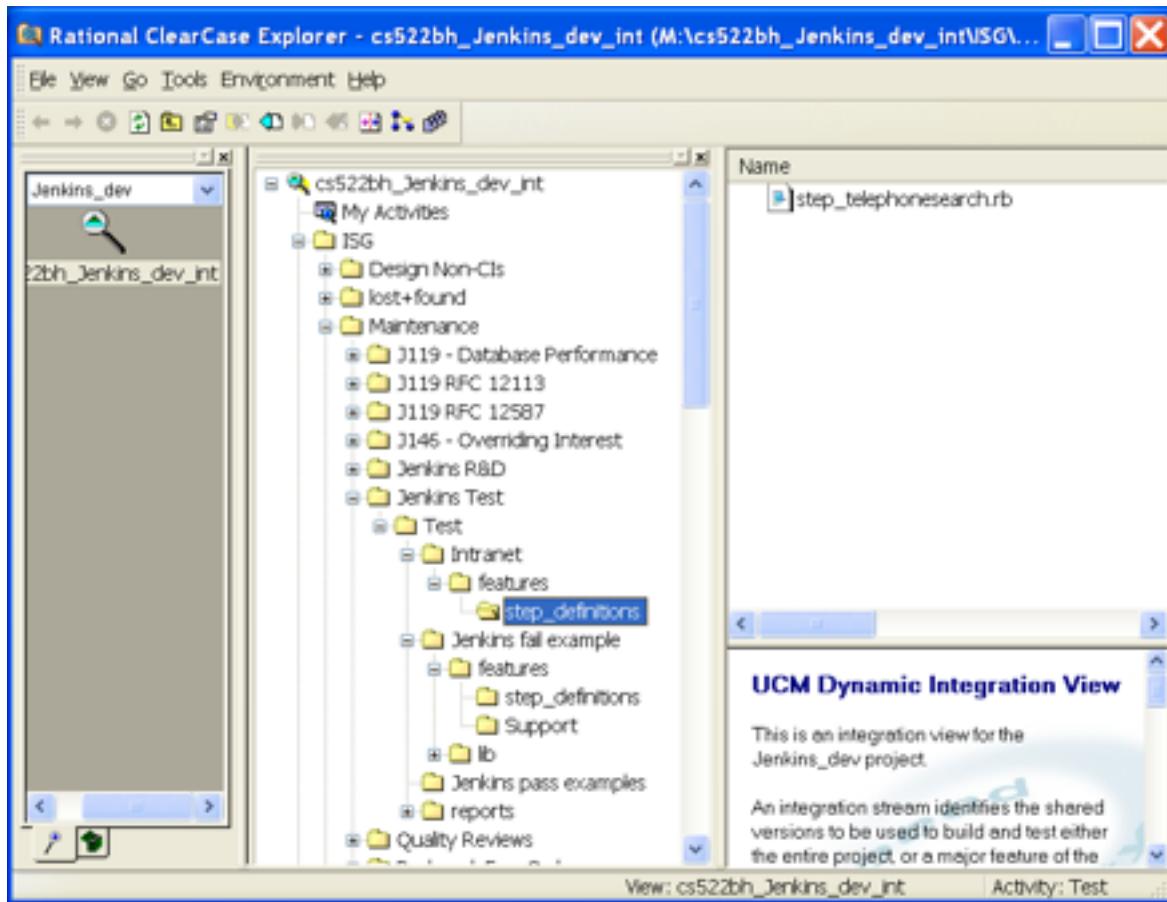
- Hover over the Build History information box, which will be shown by a flashing traffic light
- Click and Select Console Output



- The results are printed on the page as they are returned.
- At the end of the results you will see Finished: SUCCESS or FAILURE, see extract below:

```
[xUnit] [INFO] - The total number of tests for this category exceeds the specified 'failure' threshold value.
[xUnit] [INFO] - Setting the build status to FAILURE
[xUnit] [INFO] - Stopping recording.
Finished: FAILURE
```

- This particular test failed.
- **6.12 Successful Jenkins Run.**
- Let's run another test in Jenkins. Hopefully this time it will be successful. The first thing we need to do is add another test to Jenkins\_Dev\_Int project in ClearCase.
- We know that the test we created in Cucumber to search names on the Intranet was successful so we will use that.
- In ClearCase there is another folder created in the Test folder named Intranet (located at Jenkins\_dev\_int\ISG\Maintenance\Jenkins Test\Test\Intranet).



- The next step is to configure the project, click the configure icon.

Click on Add build step.

- Click on the Execute Windows batch command
- Enter cd "Jenkins\_dev\_Int\ISG\Maintenance\Jenkins Test\Test\Intranet"  
cucumber --format junit --out ..\reports\ || Exit 0
- Click on the Save button at the bottom of the page
- Click on Build now
- Click on the flashing red traffic light and go to the console to see the results

You will see that the result says FAILURE. The reason it says failure is because the first test 'Jenkins Fail Example' has still failed, as it was supposed to.

The second test has passed – how do we know this?



- Go back to the Ian Test tab at the top of the screen
- Look at the graph named Test Result Trend

The red shows the test that are failing but we now have blue colouring on the graph which shows that the second test has been SUCCESSFUL.