

Advanced Techniques for Automated Deployment and Testing

A Major Qualifying Project Report:

submitted to the faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science

by:

Ryan Svoboda

Date: April 30th, 2009

Approved:

Professor Neil T. Heffernan, Major Advisor

1. Continuous Integration
2. Process Automation
3. Web Application Testing

Abstract

The goal of the project is to introduce the software development methodology of Continuous Integration into the Assistment lab. Continuous Integration is a collection of software engineering practices which aim to reduce the time spent resolving issues arising from integrating development code into release software. By automating manual processes we can decrease the amount of time it takes for new features to reach the final product and detect errors earlier. If errors are found, the developers are notified and it becomes a priority to fix the problem. This leads to a reduction of risk by removing bugs as they are introduced. This will allow developers to focus their work on writing functional and well tested code and help to release new features faster.

Acknowledgements

I would like to thank Neil Heffernan for sponsoring the project, and for the use of the necessary server equipment for this project. I would also like to thank Joseph Politz and Jozsef Patvarczki for their input and insightful feedback.

Table of Contents

Abstract	i
Acknowledgements	ii
Table of Contents	3
Introduction	1
Background	4
Methodology	10
Results and Analysis	19
Future Work	21
Conclusions	22
Appendix A: Installing and setting up CruiseControl.rb	23
Appendix B: Installing and using Selenium	27
Appendix C: Selenium Test Suite	29
login.rb	29
complete_assignment.rb	31
create_class.rb	33
join_class.rb	35
assign_work.rb	37
Appendix D: Ruby on Rails Unit Test Suite	39
action_hint_test.rb	39
action_problems_test.rb	40
action_test.rb	41
answer_test.rb	42
assignment_answers.rb	44
assistent_info_test.rb	45
assistent_ownership_test.rb	46
assistent_sequence_test.rb	47
assistent_test.rb	51
class_assignment_test.rb	54
class_report_comment_test.rb	56
content_creator_test.rb	57
daily_student_report_test.rb	58
db_files_test.rb	59
district_test.rb	60
enrollment_test.rb	62
hint_test.rb	63
image_test.rb	64
logged_action_test.rb	65
problem_test.rb	66
problem_type_test.rb	70
progress_test.rb	75
report_generator_test.rb	77

right_test.rb	78
role_test.rb.....	79
scaffold_test.rb	80
school_test.rb	81
section_test.rb.....	83
sequence_ownership_test.rb.....	106
sequence_test.rb	107
student_class_test.rb.....	110
student_files_test.rb	114
student_parent_test.rb	114
student_report_comment_test.rb	115
student_test.rb	116
teacher_class_test.rb.....	119
teacher_parent_message_item_report_reference_test.rb	120
teacher_test.rb	121
tutor_strategy_test.rb.....	124
user_detail_test.rb	125
user_role_test.rb	126
user_test.rb	127
Appendix E: Sample Build Log.....	132
References.....	160

Introduction

In all software development projects as the number of users grow, the time that must be spent integrating code from individual developers into a single code base increases. This process consists of a series of steps that can consume a large amount of time and can become a hindrance to future development if all of the steps must be manually completed. First the development code must be tested locally until the developer is confident the code is ready to be pushed to the production environment. The new code must then be merged into the existing code base. If conflicts within files are realized, the developer must manually decide which set of changes are appropriate.

The newly combined code must then be tested in an integration environment, to ensure that no new bugs have been introduced, and to detect any other errors the programmer may have missed. When these tests have been successfully completed, the code should be tested in a production environment, one that is as close a copy as possible of the actual environment that the final product will be run on. If all of these stages are successfully completed, the new code can be introduced into the project.

The amount of time spent integrating code can become a difficult impediment to the development of new features since even the smallest changes must go through rigorous testing to ensure that as few errors as possible propagate to the final project. This

is why it is critical to automate as much of this process as is possible. It must be as simple as possible to build the software; it should be able to be compiled with a single command. This allows new developers to quickly have a running system on their development machine, all they need to do is check out the source code and run the build script.

Much of the automation of the integration process can be handled by setting up an integration server. This server should automatically build and test all new versions of the software, thus removing this requirement from the developer. If all tests pass the program can then be tested in a production environment to simulate how it would run in a typical environment. If there are problems with the build or tests are failing, the developers should be automatically notified so that they may immediately work on fixing the issue. An important factor of all software projects is to always maintain a working build of your software in the mainline of the source control repository. By maintaining a working copy of your latest software updates it is always possible to obtain the latest improvements and features easily, which increases confidence within the development team, and also allows the customer to see how the project is improving in near real time.

The remaining sections of this document are organized thusly. The “Background” section describes the Continuous Integration methodology in detail. The “Methodology” section covers the processes and software that was implemented to assist in the development process. The “Results and Analysis” section describes the accomplishments

and examines how the implementation affected the development process for the project. Finally, the “Future Work and Conclusions” section details how the development team can continue to utilize and improve upon the work completed.

Background

The procedure of integrating code from individual developers into a shared repository is different at every software development company. The most basic and difficult integration process is one where this step of development is mostly overlooked until the deadline for the project is nearly upon the team. At this point the team is forced to enter an integration phase, taking all of the code written by each developer and put it all together at the same time. This process can introduce a new type of error into the system, commonly referred to as “integration bugs”. The development team must then spend as much time as necessary to test the software and fix any new bugs that are found. Often this stage is difficult to estimate the amount of time it will take to complete, since errors between different components can be difficult to track down, especially in view of the fact that all of the changes to the codebase are compounded at the same time. This often permits bugs that the developers miss to make their way into the final product.

Software integration is not necessarily a difficult event, but if it is left until the end of the project, it can take down even the most well written application by introducing unforeseen problems immediately before important deadlines. Steps must be taken to prevent a so called “integration hell”. It can be avoided by a shift in development practices. By having all of the developers on the team integrate their code often,

integration bugs can be detected both earlier in the development process, and are usually easier to fix. By integrating small changes into the repository often and testing for errors as soon as possible it reduces the amount of risk taken when you otherwise would have had to hope everything works together well at the end of the development cycle.

Often on a large project the team understands that the current process of deferring integration causes difficulties, but they say it would be even more difficult to introduce new practices to remedy it. The fallacy in this is that the majority of work that is currently being done is often repetitive and easy to automate. The first step is to create a script that can automatically build the latest version of the software. “A build is much more than a compile (or its dynamic language variations). A build may consist of the compilation, testing, inspection, and deployment—among other things. A build acts as the process for putting source code together and verifying that the software works as a cohesive unit.”

(Duvall, Matyas, & Glover, 2007)

The build script can be a simple program that can save time for all of the developers on a project. By having a script to build the program, it makes it so that the build process is the same for everyone on the team, so that problems with building the software are much easier to detect and fix. The script runs the build the same way every time, so that there are no longer problems with individual developers attempting to build the software and failing, simply because they forgot to do a small step in the build process.

Once the build process has been simplified into a singular command, one of the next steps a software development team will do is to create “nightly builds”; automated builds of the software that occur regularly. They are usually run over night so that all changes from the previous day have been committed and the developers are not currently at work. The next day the first thing the developers see is if the build from the previous night worked. If it has failed then the first priority is to correct the problems in the failed build so the current copy of the software builds without error. This is a good first step in reducing the risk taken in deferred integration, but it can still be vastly improved upon. Simply building the software can only detect the most obvious bugs, such as syntax errors and missing dependencies. To make the automated builds more useful, we need to introduce automated testing.

Manually testing a software project is a long and tedious process that can easily be overlooked by the developers on a team. The excuse for not running the tests is often simple “I didn’t have the time to run all of the tests, so I just ran mine.” This may help detect local errors, but it does not determine if the new changes have affected any of the other aspects of the project. Having the tests run locally on the development machine also makes it much more difficult to keep a log of all of the test records for each version of the software. By automating the testing process, it ensures that testing is completed on a regular basis, helping to detect errors earlier so they can be fixed before more bugs become compounded upon smaller and easier to fix errors.

There are many different types of tests to verify software is working correctly and to detect errors and the easiest of these to automate is the “unit test”. “A true unit tests should run to completion (successfully) in a fraction of a second.” (Duvall, Matyas, & Glover, 2007) These tests should be simple and run fast since they should not be reliant upon outside sources that can increase the amount of running time such as databases. Unit tests are a very fast to detect minor errors in code, such as logic errors and unfulfilled boundary conditions, since they can be run quickly and test small components individually.

“A good way to catch bugs more quickly and efficiently is to include automated tests in the build process. Testing isn't perfect, of course, but it can catch a lot of bugs - enough to be useful.” (Fowler, 2006) This causes a regularly scheduled run of the tests to be run against the code, and if the developers monitor the status of the tests then they can see what needs to be fixed as soon as it has a failing test. Even if the logs for testing are not monitor constantly, they become a valuable asset to the development team, because they can go back and see when a test started failing, and that makes it much easier to isolate the changes which caused the test to fail.

Other types of tests that are run manually can be automated as well, such as component tests, which verify that larger components interact with each other properly, and functional tests, which verify that the application behaves the way it was intended to.

Adding these types of tests to the automated build and test phase can increase the amount and type of problems detected by testing, again enabling the developers to fix problems earlier in the development cycle which leads to a better quality and more reliable software product when it comes time to release a final version.

Nightly builds are a good first step in automating the build and testing the application, but on most software projects, multiple changes are made daily, and it would be advantageous to run this automated process for every change that is committed to the central repository. This requires monitoring the source control repository for changes, and running the automated scripts whenever a change has been detected. One of the easiest ways to do this is to set up a server dedicated to running these processes. This has been coined a “Continuous Integration server” since it attempts to resolve integration problems as soon as possible.

The Continuous Integration server handles all of the aspects discussed earlier. It monitors the source control repository for changes, and when it detects a new version of the software it begins the automated build. When that is completed, it begins to run the automated test suite, starting with the fastest and simpler unit tests before moving on to the longer running component and functional tests. The CI server is also responsible for keeping logs of all of these events, so that every developer has a central place to look to see if the current build works, and to easily check the status of the test suite ran against

the latest versions of the software. It would even be easy to set up automated deployment to a production environment on the condition that the build passes and the entirety of the tests pass, however, not all projects chose to deploy new software immediately and this is not necessary as it does not improve the final quality of the software, but just ease the deployment process.

The key feature of Continuous Integration is that everything is completely automated. Using a CI server effectively can completely eliminate the need for an integration phase at the end of a project, since there is continually testing being run on every version of the software. This lets the developers work on integration during the development process, because they receive feedback immediately upon changing the software, they are much more motivated to fix any new errors that they may have created with the new changes. It becomes much easier to locate bug because there are no gaps in the logs where individual developers forgot to run some of the tests, so it is much easier to see what minor changes made the test begin to fail. “If you have continuous integration, it removes one of the biggest barriers to frequent deployment. Frequent deployment is valuable because it allows your users to get new features more rapidly, to give more rapid feedback on those features, and generally become more collaborative in the development cycle.” (Fowler, 2006)

Methodology

“The Assistment project (<http://www.assistment.org/>) is a web based tutoring program for 4th to 10th grade mathematics. In 2007-2008, just over 4000 students used the system, mostly near Worcester, MA and Pittsburgh, PA. Math teachers assign problem sets to their students to do on the computer, and students are tutored on the items they get wrong. Teachers log on to the System and study detailed reports about their students’ difficulties and strengths. Teachers can also use content already developed at WPI or write their own content.” (Assistment Overview, 2009) Development began in 2003 with a few developers, and has since grown over time having more than 30 WPI students having worked on the system since its creation. Currently the core development team consists of full time and part time students working on core system development, as well as many students working on various projects associated with the Assistment system.

The current development process for the Assistment project begins by checking out the latest version of the code from the source control repository into a personal branch. The developer then works on new features and checks in updates periodically into their own branch. When it is determined that the code is stable and has been satisfactorily tested by the developer, the developer must ask the lab manager to merge the code from the individual branch into the main repository.

The lab manager inspects the code for any obvious defects and if the code is approved then they must manually complete a source control merge from the development branch into the main repository, thus adding the new code into the most recent version of the main software. After the merge operation is completed then the server is started and the application is tested via manual interaction with the web browser. If the functional testing of the new code goes smoothly, the code can be deployed to the production environment, which is completed by checking out the new version of the code and then running a script which restarts the web servers.

This process is a significant barrier to the development of new features and improvements upon the current application. Since work is done in individual branches, other developers do not have access to changes that have not yet been integrated into the mainline. Having this separation causes integration problems to go undetected while new work is in development, instead the problems will not manifest until the code has been completed and merged into the main development branch. It is recommended that all developers are given access to the main repository so that they are able to commit small changes thus allowing the other developers to receive these updates and detect any conflicts as early as possible.

The reliability of the system is always in question because testing is scarce at best. Often unit tests are not written for new code, and even when they are written, they do not

get run often. Testing new code usually consists of functional testing; manually interacting with the web browser and checking to ensure that the correct results occur when a specific series of actions is taken within the application. Even when deploying to the production environment, new features are tested from only the view of the end user. This allows an amount of errors to go undetected because every series of interactions cannot possibly be manually tested.

At the exposition of this project the goal was to introduce automated builds and testing to the Assistent development lab. The first process looked into was nightly builds; setting up a script to run at night when the developers have left work to check out the latest code and run a series of tests against them. The developers would come in the next day and see the results of the nightly build, including if the build was successful and the status of each test. It was quickly decided to improve upon this idea by introducing Continuous Integration into the development lab.

Continuous Integration is similar to nightly builds in that automated building and testing occurs, but the main difference is that these builds are ran when new changes are detected in the main source repository. This allows for much more rapid feedback from the automated builds, as well as testing every change. If multiple changes are committed in the same day, nightly builds would only test the last commit of the day. This allows for much more rapid feedback from the automated builds, as well as testing every change. If

multiple changes are committed in the same day, nightly builds would only test the last commit of the day.

CruiseControl.rb (<http://cruisecontrolrb.thoughtworks.com/>) is an open source Continuous Integration server written in Ruby on Rails. There are two main parts to CruiseControl.rb, the builder and the dashboard. The builder is a background process that monitors the source control repository for changes. When the builder detects that someone has checked in new changes to the source control repository, it will check out these changes to update its copy of the project, run the build for the project, and reports on the outcome of the build. The dashboard is the web based front end for CruiseControl.rb, which allows the developers to monitor the status of builds, request new builds, and help with troubleshooting errors that have occurred.

Cruise Control was chosen to be used as the Continuous Integration server for several reasons. CruiseControl.rb is an open source Ruby on Rails version of the popular Cruise Control application. Since the Assistment project is also written in Ruby on Rails maintenance of the CruiseControl.rb server would be familiar to the developers in the Assistment lab. The open source nature of Cruise Control allows for easy customization of the Continuous Integration server. CruiseControl.rb is designed to work with any software written in any language, but it works out of the box with Ruby or Ruby on Rails

projects. CruiseControl.rb supports SVN as its source control repository, which is the current source control management software in use by the Assistment development lab.

CruiseControl.rb also provides support for Selenium web application. Selenium tests are functional tests recorded as macros in Firefox. These tests interact directly with the browser, manipulating it in the same way that the end user does. It records clicks, typing, and other actions and can play them back later. This will allow us to create functional and smoke tests for the Assistment project. Selenium was chosen over other web application scripting tools such as WATIR due to the support built into CruiseControl.rb. The Selenium tests that have been created test basic functionality of the website, covering things such as logging in or out, creating new user accounts, and completing problem sets.

Selenium IDE allows you to export your recorded tests in many programming languages, including Ruby. This allows the developers to tweak the tests after they have been recorded to provide more comprehensive test coverage. Since the majority of actions taken by the end user have an associated side effect, it is important to limit the functionality of each test to individual components. The individual tests can be combined easily to mock the way the user interacts with the application. Here is an example of a Selenium test written to log in as a test user. The first example is the basic Selenium

source, output as a series of selenium commands in an xml format, and the second is the same test, but written as a Ruby program.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="" />
<title>New Test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">New Test</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>http://localhost:3000/</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>Assistment</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>link=Log in</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>Assistment » Account » Accounts</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>login</td>
<td>seleniumtest@wpi.edu</td>
</tr>
<tr>
<td>type</td>
<td>password</td>
<td>Selenium Test</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>Assistment » Teacher » Student Classes</td>
<td></td>
</tr>
</tbody></table>
</body>
```

```

require "selenium"
require "test/unit"

class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleniumDriver.new("localhost", 4444,
      "**chrome", "localhost:3000", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "http://localhost:3000/"
    assert_equal "Assistment", @selenium.get_title
    @selenium.click "link=Log in"
    @selenium.wait_for_page_to_load "30000"
    assert_equal "Assistment » Account » Accounts", @selenium.get_title
    @selenium.type "login", "seleniumtest@wpi.edu"
    @selenium.type "password", "Selenium Test"
    @selenium.click "commit"
    @selenium.wait_for_page_to_load "30000"
    assert_equal "Assistment » Teacher » Student Classes", @selenium.get_title
  end
end

```

This test opens the browser to the Assistment application running locally, loads the home page, clicks on the log in button, fills in the username and password for a test user and clicks the login button. The test then checks to see if the user has successfully logged in by checking the title of the page after the user tries to log in. If the login failed, then this assertion will fail because the user will not be directed to the correct page. The Selenium test written as a Ruby program is much easier to read and edit.

Installing and setting up CruiseControl.rb is simple, download and unpack the application into any directory, add the URL for the SVN repository of the project to CruiseControl.rb, and then start the application. The default location for the CruiseControl.rb server is localhost: 3333. If you navigate there in your browser you can access the dashboard, which displays the list of projects that have been added, as well as the status of the most recent builds. From there you can investigate details for each build, which display the changes since the previous build, results from running the tests, logs from the build process, and the settings for this build. For more detailed information on installing CruiseControl.rb, see Appendix A.

Results and Analysis

A computer has been acquired in the lab and set up as a Continuous Integration server. It is running CruiseControl.rb, which monitors the Assistment SVN repository. When an update has been detected CruiseControl.rb checks out the latest version of the code and begins building. The build consists of tearing down the previous database and then setting it up with any new changes if necessary. Default users and roles are added, and then the unit tests that have been written for the Assistment project are run. After the tests are completed the server is started, so that the latest version of the Assistment project is always available on the server.

Before the introduction of the Continuous Integration server the unit tests that existed were rarely run. Out of one hundred seventy one (171) there were fourteen (14) failures and sixty four (64) errors. Currently the number of tests has remained the same, but the number of errors has decreased to forty eight (48), which is a twenty five (25) percent decrease in the number of errors thrown when the test suite is run. Simply by running the test suite and making the results visible to the developers, it was realized that it is necessary to maintain the test suite. As the Continuous Integration server begins to become a more central part of the development process in the Assistment lab, the logs of the testing results will become a valuable asset to the development team, assisting in

detecting newly introduced errors as well as helping to pinpoint when an error was introduced.

The benefits of a Continuous Integration server are difficult to measure immediately, but it will be an asset to the development team as it has begun the process of automating many important aspects of the software development process. Tests are being run against every new change in the system, with logs being recorded and saved for future use. It is simple to see the current state of the project, you simply need to navigate to the dashboard of CruiseControl.rb and there you may view detailed logs for the latest build.

Future Work

The Continuous Integration server will serve as a valuable asset for the development team as it is currently configured, but it has the potential to become much more useful. Selenium tests have been written and should be added to the build and testing process spawned when new changes are detected. These tests will help to guarantee future functionality of the application; if something does not respond in the correct way as defined in the test, it will cause a failure and report this to the development team. These types of errors are usually hard to detect manually, but if the same set of tests is run for every change to the system it will help in detecting what changes caused the failure.

CruiseControl.rb can also be configured to generate documentation for the Assistment project using RDoc. RDoc is a documentation generation utility for Ruby on Rails applications which extracts pertinent information from the source code and outputs html files containing the documentation for the project. The Assistment lab currently uses RDoc markup in some of the code, so it would not be difficult to set up CruiseControl.rb to also run the RDoc utility when changes are detected. The server could also be set up to host the documentation, providing a central repository for information about the Assistment codebase.

Conclusions

A Continuous Integration server has been set up to assist in the development process for the Assistment lab. Its usefulness is being realized as tests that have not been run regularly are now being run with every change to the codebase. As the development team continues to use the server it will become the benefits will become more realized and it will be a useful tool in aiding the reliability of the overall project. The server contains many possibilities for expansion, and will only become more useful to the development team as its functionality is expanded upon.

Appendix A: Installing and setting up CruiseControl.rb

Cruise Control requires that both Ruby and Subversion must be installed before Cruise Control can be installed.

1. Download and unpack Cruise Control (currently can be found at http://rubyforge.org/frs/?group_id=2918)

2. In the directory that Cruise Control was unpacked into run the command:

```
./cruise add your_project --url location_of_svn_repository
```

This creates a directory at `$HOME/.cruise` which is where Cruise Control keeps all of its data. For each project a working directory is created at

```
$HOME/.cruise/projects/your_project/work/
```

3. From the install directory run `./cruise start`.
4. Browse to <http://localhost:3333>.

Your browser should display the Cruise Control dashboard

5. Navigate to `$HOME/.cruise/projects/your_project/work/` and make the build pass. For a regular Rails app, this involves creating a test database, editing `database.yml` to connect to that database, performing `rake RAILS_ENV=test`

`db:migrate`, running `rake test` and making sure that it passes. If you have a build script then make sure that it will successfully complete when ran.

6. Configure Cruise Control by editing the file at

`$HOME/.cruise/projects/your_project/cruise_config.rb`

You can tell Cruise Control to use your build script, as well as set up email configuration and polling intervals in this configuration file.

7. Press the “build now” button on the Cruise Control dashboard to rebuild your project.

This should build your project and place build output logs into

`$HOME/.cruise/projects/your_project/build-[revision-number]`

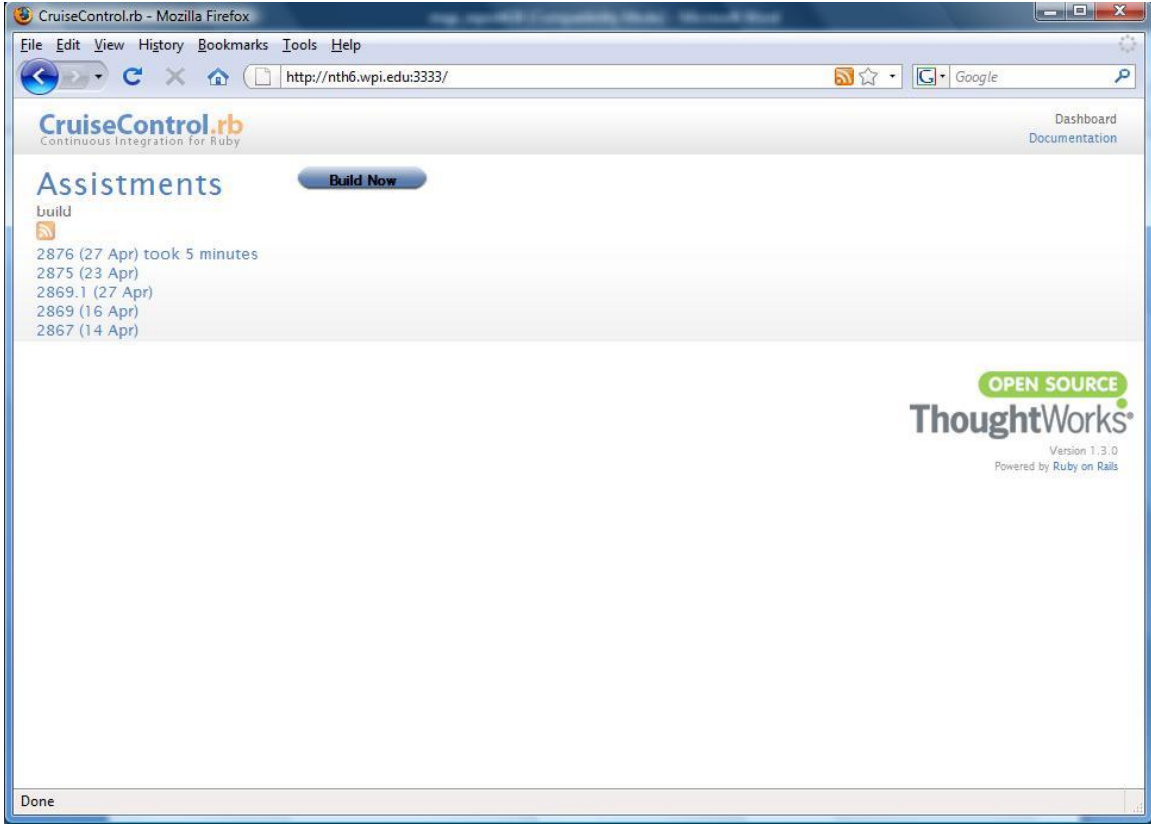


Figure 1: The CruiseControl.rb dashboard

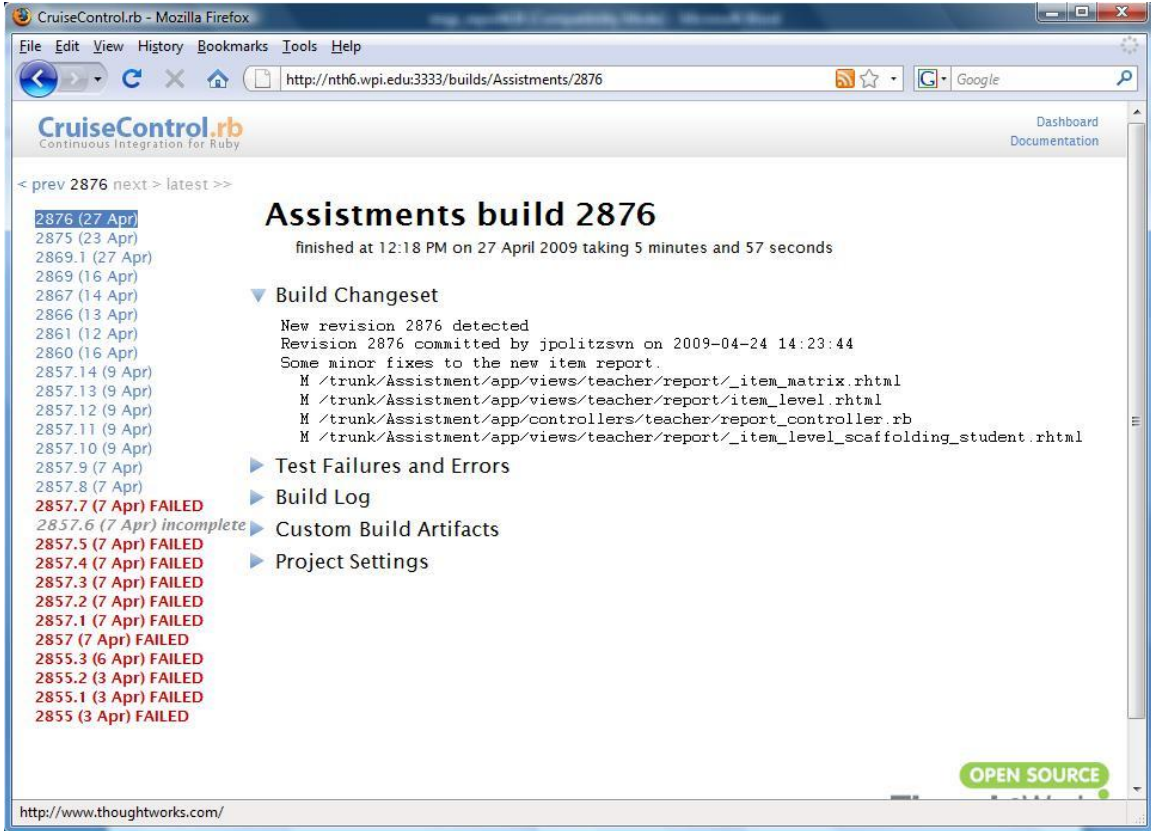


Figure 2: Cruise Control build reports

Appendix B: Installing and using Selenium

Selenium is a Firefox add-on which requires that Firefox is installed previous to installing the Selenium IDE add-on. Navigate in Firefox to <http://seleniumhq.org/download/> and select the Download link next to Selenium IDE. This should prompt you to install the Selenium IDE add-on, which will require Firefox to be restarted once it is finished installing.

To open the Selenium IDE, select Tools>Selenium IDE from the Firefox toolbar. To begin recording a test case click on the red record button on the top right of the IDE. All actions taken in Firefox while the record option is enabled will be recorded to the current test. To stop recording a test disable the red record button by clicking on it again.

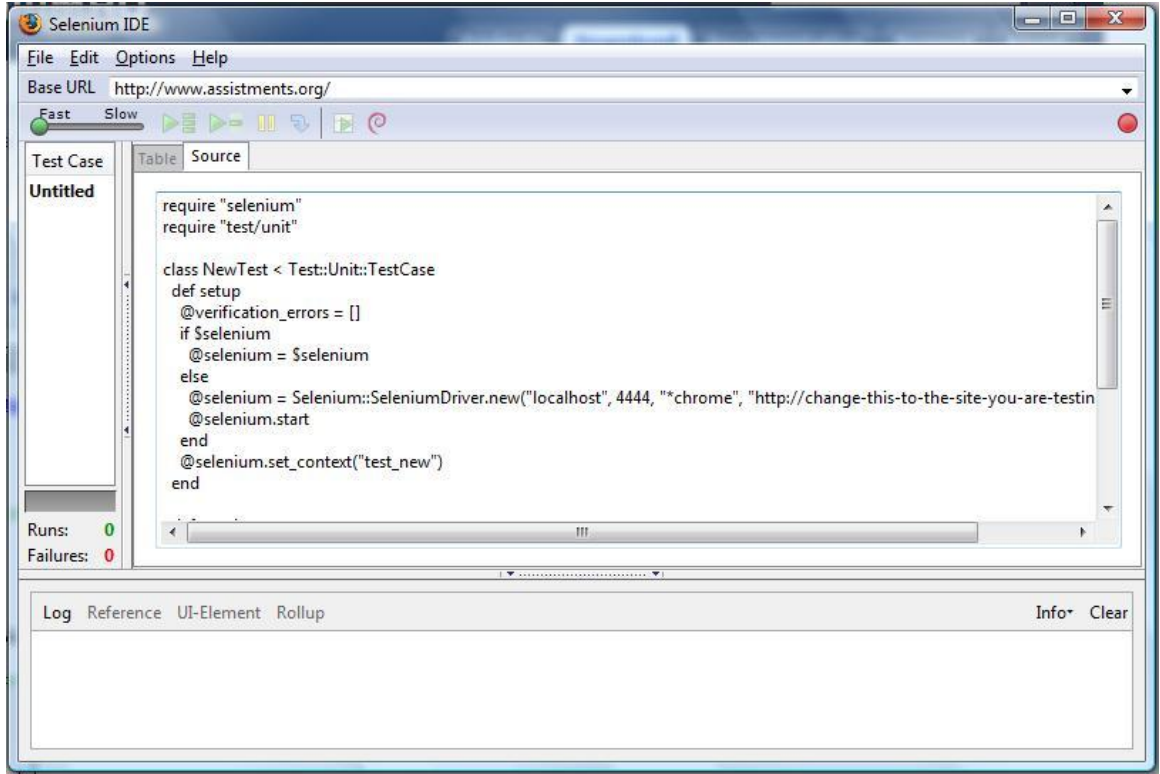


Figure 3: The Selenium IDE

Appendix C: Selenium Test Suite

login.rb

```
require "selenium"
require "test/unit"

class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleniumDriver.new("localhost", 4444, "*chrome",
"http://localhost:3000/", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "http://localhost:3000/"
    assert_equal "Assistment", @selenium.get_title
    @selenium.click "link=Log in"
    @selenium.wait_for_page_to_load "30000"
    assert_equal "Assistment » Account » Accounts", @selenium.get_title
  end
end
```

```
@selenium.type "login", "seleniumtest@wpi.edu"  
@selenium.type "password", "Selenium Test"  
@selenium.click "commit"  
@selenium.wait_for_page_to_load "30000"  
assert_equal "Assistment » Teacher » Student Classes", @selenium.get_title  
end  
end
```

complete_assignment.rb

```
require "selenium"
```

```
require "test/unit"
```

```
class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleniumDriver.new("localhost", 4444, "*chrome",
"http://change-this-to-the-site-you-are-testing/", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "http://localhost:3000/tutor"
    assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
    @selenium.click "link=Tutor"
    @selenium.wait_for_page_to_load "30000"
    assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
    @selenium.click "link=Show Assignments"
    @selenium.wait_for_page_to_load "30000"
  end
end
```

```
assert_equal "Assistment » Tutor » Class Assignments", @selenium.get_title
@selenium.click "//a[contains(text(),'1 - Quick Assignment 04/07/09')]"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Tutor » Class Assignments", @selenium.get_title
@selenium.type "answer_2", "answer"
@selenium.click "submit_answer_link_2"
@selenium.click "assignments_link"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Tutor » Class Assignments", @selenium.get_title
end
end
```

create_class.rb

```
require "selenium"
require "test/unit"

class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleniumDriver.new("localhost", 4444, "*chrome",
"http://change-this-to-the-site-you-are-testing/", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "http://localhost:3000/teacher"
    assert_equal "Assistment » Teacher » Student Classes", @selenium.get_title
    @selenium.click "link=New Class"
    @selenium.select "student_class_sections_size", "label=2"
    @selenium.select "student_class_grade", "label=8"
    @selenium.select "student_class_sections_size", "label=1"
    @selenium.type "student_class_sections_name_0", "Selenium grade 8"
  end
end
```

```
@selenium.click "commit"
@selenium.click "link=Account"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Account » Accounts", @selenium.get_title
@selenium.click "//input[@value='Give me a student account']"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Account » Accounts", @selenium.get_title
@selenium.click "link=Build"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Build » Assistments", @selenium.get_title
@selenium.click "link=Tutor"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
@selenium.click "link=Enroll in a class"
@selenium.select "student_class_id", "label=Selenium T Test: Math Grade 8"
@selenium.click "link=Assess"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Teacher » Student Classes", @selenium.get_title
@selenium.click "link=Tutor"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
end
end
```

join_class.rb

```
require "selenium"
require "test/unit"

class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleniumDriver.new("localhost", 4444, "*chrome",
"http://change-this-to-the-site-you-are-testing/", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "http://localhost:3000/tutor"
    assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
    @selenium.click "link=Enroll in a class"
    @selenium.select "student_class_id", "label=Selenium T Test: Math Grade 8"
    @selenium.select "section_id", "label=Selenium grade 8"
    @selenium.click "submit_button"
  end
end
```



```
end
```

```
require "selenium"
```

```
require "test/unit"
```

```
class NewTest < Test::Unit::TestCase
```

```
  def setup
```

```
    @verification_errors = []
```

```
    if $selenium
```

```
      @selenium = $selenium
```

```
    else
```

```
      @selenium = Selenium::WebDriver.new("localhost", 4444, "*chrome",  
"http://change-this-to-the-site-you-are-testing/", 10000);
```

```
      @selenium.start
```

```
    end
```

```
    @selenium.set_context("test_new")
```

```
  end
```

```
  def teardown
```

```
    @selenium.stop unless $selenium
```

```
    assert_equal [], @verification_errors
```

```
  end
```

assign_work.rb

```
def test_new
  @selenium.open "http://localhost:3000/teacher"
  assert_equal "Assistment » Teacher » Student Classes", @selenium.get_title
  @selenium.click "link=Assignments"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Teacher » Student Classes", @selenium.get_title
  @selenium.click "link=New Assignment"
  @selenium.select "assignment", "label=1 - Quick Assignment 04/07/09"
  @selenium.click "link=Tutor"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
  @selenium.click "link=Show Assignments"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Tutor » Class Assignments", @selenium.get_title
  @selenium.click "//a[contains(text(),'1 - Quick Assignment 04/07/09')]"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Tutor » Class Assignments", @selenium.get_title
  @selenium.click "link=Tutor"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
  @selenium.click "link=Build"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Build » Assistments", @selenium.get_title
  @selenium.click "//div[@id='assistment_index_message']/div[3]/a/em"
  @selenium.wait_for_page_to_load "30000"
  assert_equal "Assistment » Build » Assistments", @selenium.get_title
  @selenium.click "//div[@id='assistment_2']/span[2]/a[2]"
  @selenium.wait_for_pop_up "preview_assistment", "30000"
```

```
@selenium.click
"/a[@onclick=\"window.open(this.href,'preview_assistment','height=600,width=800,resizable=yes,scrollbars=yes');return false;\"]"
@selenium.wait_for_pop_up "preview_assistment", "30000"
@selenium.click "link=Tutor"
@selenium.wait_for_page_to_load "30000"
assert_equal "Assistment » Tutor » Student Classes", @selenium.get_title
end
end
```

Appendix D: Ruby on Rails Unit Test Suite

action_hint_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ActionHintTest < Test::Unit::TestCase
```

```
  fixtures :action_hints
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

action_problems_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ActionProblemTest < Test::Unit::TestCase  
  fixtures :action_problems
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

action_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ActionTest < Test::Unit::TestCase  
  fixtures :actions
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

answer_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class AnswerTest < Test::Unit::TestCase
  fixtures :answers, :assignment_answers, :problems
```

```
  def test_problem
    Answer.find(:all).each do |answer|
      assert_respond_to(answer, :problem)
      assert_respond_to(answer, :problem_id)
      assert_kind_of Problem, answer.problem
      assert_equal answer.problem_id, answer.problem.id
    end
  end
end
```

```
  def test_acts_as_list
    assert_acts_as_list(:list => Answer, :scope => Problem, :fixture =>
problems(:rank))
  end
end
```

```
  def test_incorrect_message
    assert answers(:has_incorrect_message).incorrect_message?
    assert !answers(:no_incorrect_message).incorrect_message?
    assert !answers(:empty_incorrect_message).incorrect_message?
  end
end
```

```
  def test_correct?
    assert answers(:correct).correct?, "Answer should be correct"
    assert !answers(:incorrect).correct?, "Answer should be incorrect"
  end
end
```

end

assignment_answers.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class AssignmentAnswerTest < Test::Unit::TestCase  
  fixtures :assignment_answer
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

assistent_info_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class AssistentInfoTest < Test::Unit::TestCase  
  fixtures :assistent_infos
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

assistent_ownership_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class AssistentOwnershipTest < Test::Unit::TestCase  
  fixtures :assistent_ownerships
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

assistment_sequence_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class AssistmentSequenceTest < Test::Unit::TestCase
```

```
  def test_SingleProblemOperation
```

```
    s = AssistmentSequence.new(3)
```

```
    assert(1==s.size)
```

```
    assert(1==s.length)
```

```
    assert(3==s.pop[0])
```

```
    assert(nil==s.pop)
```

```
  end
```

```
  def test_concatination
```

```
    s0 = AssistmentSequence.new(1)
```

```
    s0.concatinate(AssistmentSequence.new(2))
```

```
    s0.concatinate(3)
```

```
    assert(3==s0.size)
```

```
    assert(3==s0.length)
```

```
    s1 = AssistmentSequence.new(4)
```

```
    s1.concatinate(AssistmentSequence.new(5))
```

```
    s1.concatinate(s0)
```

```
    assert(5==s1.size)
```

```
    assert(5==s1.length)
```

```
    assert(4==s1.pop[0])
```

```
    assert(5==s1.pop[0])
```

```
assert(1==s1.pop[0])
assert(2==s1.pop[0])
assert(3==s1.pop[0])
assert(nil==s1.pop)
end
```

```
def test_append
  s0 = AssistentSequence.new(1)
  s0.append(AssistentSequence.new(2))
  s0.append(3)

  assert(3==s0.size)
  assert(3==s0.length)

  s1 = AssistentSequence.new(4)
  s1.concatinate(AssistentSequence.new(5))
  s1.append(s0)

  assert(3==s1.size)
  assert(5==s1.length)

  assert(4==s1.pop[0])
  assert(5==s1.pop[0])
  assert(1==s1.pop[0])
  assert(2==s1.pop[0])
  assert(3==s1.pop[0])
  assert(nil==s1.pop)
end
```

```
def test_join
```

```
s0 = AssistmentSequence.new(1)
s0.append(AssistmentSequence.new(2))
s0.append(3)

s1 = AssistmentSequence.new(4)
s1.concatinate(AssistmentSequence.new(5))
```

```
s0.join(s1)
```

```
assert(2==s0.size)
assert(5==s0.length)
```

```
assert(1==s0.pop[0])
assert(2==s0.pop[0])
assert(3==s0.pop[0])
assert(4==s0.pop[0])
assert(5==s0.pop[0])
assert(nil==s0.pop)
```

```
end
```

```
def test_pop
```

```
s = AssistmentSequence.new(AssistmentSequence.new(1))
assert(1==s.pop[0])
assert(nil==s.pop)
```

```
end
```

```
def test_length_size
```

```
s =
AssistmentSequence.new(AssistmentSequence.new(AssistmentSequence.new(Assistmen
tSequence.new(1).concatinate(2))))
```

```
assert(2==s.length)
assert(1==s.size)
end
end
```

assistment_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class AssistmentTest < Test::Unit::TestCase
  fixtures :assistments, :problems
```

```
  def test_before_save_create_info
    assistment = Assistment.new
    assert_nil assistment.assistment_info
    assistment.save!
    assert_not_nil assistment.assistment_info
  end
```

```
  def test_before_save_create_timestamp
    assistment = Assistment.new
    assert_nil assistment.created_at
    assistment.save!
    assert_not_nil assistment.created_at
  end
```

```
  def test_after_create_problem
    assistment = Assistment.new
    assert_equal 0, assistment.problems.size
    assistment.save!
    assert_equal 1, assistment.problems.size
  end
```

```
  def test_problems
    Assistment.find(:all).each do |assistment|
```



```

    assert_respond_to assistment, :problems
    assert_kind_of Array, assistment.problems
    assistment.problems.each do |problem|
      assert_kind_of Problem, problem
    end
  end
end

def test_name_and_problem_body_nil
  assert_equal "Assistment ##{assistments(:all_nil).id}", assistments(:all_nil).name,
    "Assistment name is not defaulting to ID display."
end

def test_problem_body_nil
  assert_equal "Assistment ##{assistments(:problem_body_nil).id}",
assistments(:problem_body_nil).name,
    "Assistment name is not defaulting to ID display."
end

def test_problem_body_empty
  assert_equal "Assistment ##{assistments(:problem_body_empty).id}",
assistments(:problem_body_empty).name,
    "Assistment name is not defaulting to ID display."
end

def test_name_nil
  assert_equal "#{assistments(:name_nil).id} - This is a questio...",
assistments(:name_nil).name,
    "Assistment name is not defaulting to truncated main problem body"
end

def test_name_empty
  assert_equal "#{assistments(:name_empty).id} - This is a questio...",
assistments(:name_empty).name,
    "Assistment name is not defaulting to truncated main problem body"
end

```

```
end

def test_name_custom
  assert_equal "#{assistments(:first).id} - My First Assistent",
assistments(:first).name,
  "Assistent name should be custom"
end

def test_problem
  problem = assistments(:first).problem
  assert_not_nil problem,
  "Assistent has no main problem"
  assert_nil problem.scaffold_id,
  "Main problem is part of a scaffold"
  assert_equal assistments(:first).id, problem.assistent.id,
  "Main problem does not belong to correct assistent"
end
```

class_assignment_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ClassAssignmentTest < Test::Unit::TestCase
```

```
  fixtures :class_assignments, :actions
```

```
  def test_relationships
```

```
    has_many_test class_assignments(:not_started), :actions, Action,  
    "class_assignment"
```

```
  end
```

```
  def test_not_started
```

```
    assert class_assignments(:not_started).not_started?
```

```
    assert !class_assignments(:in_progress).not_started?
```

```
    assert !class_assignments(:complete).not_started?
```

```
  end
```

```
  def test_in_progress
```

```
    assert !class_assignments(:not_started).in_progress?
```

```
    assert class_assignments(:in_progress).in_progress?
```

```
    assert !class_assignments(:complete).in_progress?
```

```
  end
```

```
  def test_complete
```

```
    assert !class_assignments(:not_started).complete?
```

```
    assert !class_assignments(:in_progress).complete?
```

```
    assert class_assignments(:complete).complete?
```

```
  end
```

```
  def test_status
```

```
assert :not_started, class_assignments(:not_started).status
assert :in_progress, class_assignments(:in_progress).status
assert :complete, class_assignments(:complete).status
end
end
```

class_report_comment_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ClassReportCommentTest < Test::Unit::TestCase  
  fixtures :class_report_comments
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

content_creator_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ContentCreatorTest < Test::Unit::TestCase  
  fixtures :content_creators
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

daily_student_report_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class DailyStudentReportTest < Test::Unit::TestCase  
  fixtures :daily_student_reports
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

db_files_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class DbFilesTest < Test::Unit::TestCase  
  fixtures :db_files
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```


district_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class DistrictTest < Test::Unit::TestCase
```

```
  fixtures :districts
```

```
  def test_name_and_actual_state_required
```

```
    d = District.new
```

```
    assert !d.valid?
```

```
    assert d.errors.invalid?(:name)
```

```
    assert d.errors.invalid?(:state)
```

```
    d.name = "Hurray"
```

```
    assert !d.valid?
```

```
    assert d.errors.invalid?(:state)
```

```
    d.state = "MZ"
```

```
    assert !d.valid?
```

```
    assert d.errors.invalid?(:state)
```

```
    d.state = "MA"
```

```
    assert d.valid?
```

```
  end
```

```
  def test_find_districts_by_state
```

```
    worcester = districts(:worcester)
```

```
    boston_north = districts(:boston_north)
```

```
    goffstown = districts(:goffstown)
```

```
    assert_equal(2, District.find_all_by_state(worcester.state).size)
```

```
    assert_equal(2, District.find_all_by_state(boston_north.state).size)
```

```
    assert_equal(1, District.find_all_by_state(goffstown.state).size)
```

```
  end
```

end

enrollment_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class EnrollmentTest < Test::Unit::TestCase
```

```
  fixtures :enrollments
```

```
  def test_relationships
```

```
    belongs_to_test_enrollments(:brian_woods_1), :student, Student, "enrollment"
```

```
    belongs_to_test_enrollments(:brian_woods_1), :student_class, StudentClass,  
"enrollment"
```

```
  end
```

```
end
```

hint_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class HintTest < Test::Unit::TestCase
```

```
  fixtures :hints, :problems
```

```
  def test_problem
```

```
    belongs_to_test hints(:first), :problem, Problem, "hint"
```

```
  end
```

```
  def test_action_hints
```

```
    has_many_test hints(:first), :action_hints, ActionHint, "hint"
```

```
  end
```

```
  def test_text
```

```
    assert_equal hints(:first).text, hints(:first).text
```

```
  end
```

```
  def test_acts_as_list()
```

```
    assert_acts_as_list(:list => Hint, :scope => Problem, :fixture => problems(:rank))
```

```
  end
```

```
end
```

image_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ImageTest < Test::Unit::TestCase  
  fixtures :images
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

logged_action_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class LoggedActionTest < Test::Unit::TestCase  
  fixtures :logged_actions
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

problem_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ProblemTest < Test::Unit::TestCase
  fixtures :problems, :scaffolds, :answers, :problem_types, :action_problems,
  :assignment_answers

  def test_hints
    has_many_test(problems(:first_assistment_main), :hints, Hint, "problem")
  end

  def test_answers
    has_many_test(problems(:first_assistment_main), :answers, Answer, "problem")
  end

  def test_scaffold
    has_one_test(problems(:first_assistment_main), :scaffold, Scaffold, "problem")
  end

  def test_assistment
    belongs_to_test(problems(:first_assistment_main), :assistment, Assistment,
"problem")
  end

  def test_action_problems
    has_many_test(problems(:first_assistment_main), :action_problems,
ActionProblem, "problem")
  end

  def test_assignment_answers
    has_many_test(problems(:first_assistment_main), :assignment_answers,
AssignmentAnswer, "problem")
  end

  def test_problem_type
    belongs_to_test(problems(:first_assistment_main), :problem_type, ProblemType,
"problem")
  end
end
```

```
end
```

```
def test_correct?
```

```
  problem = problems(:first_assistment_main)
```

```
  assert problem.correct?(:body => "42")
```

```
  assert !problem.correct?(:body => "11")
```

```
end
```

```
def test_question_text
```

```
  problem = problems(:first_assistment_main)
```

```
  assert_equal problem.body, problem.question_text,
```

```
    "Question body and question text are not the same"
```

```
end
```

```
def test_has_scaffolds?
```

```
  assert problems(:has_scaffold).has_scaffolds?,
```

```
    "Problem has_scaffold should have scaffolds!"
```

```
  assert !problems(:no_scaffold).has_scaffolds?,
```

```
    "Problem no_scaffold should have no scaffolds!"
```

```
  assert !problems(:disabled_scaffold).has_scaffolds?,
```

```
    "Problem disabled_scaffold has scaffolds, but they are disabled ... so it should  
still show no scaffolds."
```

```
end
```

```
def test_parent_scaffold
```

```
  assert_nil problems(:first_assistment_main).parent_scaffold,
```

```
    "Main problem shouldn't have a parent scaffold"
```

```
  assert_equal scaffolds(:first).parent_scaffold,
```

```
    "Problem does not have the correct parent scaffold"
```



```
    assert_equal problems(:first_assistment_main).assistment.id,  
problems(:scaffold_first).assistment.id,  
    "All scaffolding questions should have the same assistment_id as their  
parent_scaffold's problem"  
end
```

```
def test_is_part_of_scaffold?  
  assert problems(:scaffold_first).is_part_of_scaffold?,  
    "Scaffolding problem should have it's scaffold_id set"  
  assert !problems(:first_assistment_main).is_part_of_scaffold?,  
    "Main problems cannot be part of a scaffold!"  
end
```

```
def test_acts_as_list  
  assert_equal problems(:scaffold_second), problems(:scaffold_first).next,  
    "The second scaffolding problem does not come after the first"  
  assert_equal problems(:scaffold_third), problems(:scaffold_second).next,  
    "The third scaffolding problem does not come after the second"  
  assert_nil problems(:scaffold_third).next,  
    "Last problem is not last!"  
  assert_nil problems(:first_assistment_main).next,  
    "Main problem should not have a next problem"  
end
```

```
def before_create_initialize_problem_type  
  problem = Problem.new :name => "Test"  
  assert_nil problem.problem_type  
  problem.save!  
  assert_not_nil problem.problem_type  
  assert_equal "choose_1", problem.problem_type.name
```

end

end

problem_type_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ProblemTypeTest < Test::Unit::TestCase
```

```
  fixtures :problem_types, :problems, :answers
```

```
  def test_problems
```

```
    has_many_test(problem_types(:choose_1), :problems, Problem, "problem_type")
```

```
  end
```

```
  def test_description
```

```
    assert_equal "Multiple Choice", problem_types(:choose_1).description,
```

```
      "ProblemType description is set, but description doesn't return it"
```

```
    assert_equal "Algebra", problem_types(:algebra).description,
```

```
      "ProblemType does not have description set, and it's not generating a correct  
description"
```

```
  end
```

```
  def test_choose_1_correct
```

```
    setup_choose_1
```

```
    answer = {:body => 42}
```

```
    correct = @type.correct?(@problem, answer)
```

```
    assert correct, "Answer #{answer[:body]} should be correct"
```

```
  end
```

```
  def test_choose_1_invalid_choice
```

```
    setup_choose_1
```

```
    answer = {:body => -5}
```

```
    assert_equal false, @type.correct?(@problem, answer),
```

```
      "Answer #{answer[:body]} should be incorrect, even though it's not an option"
```

```
  end
```

```

def test_choose_1_incorrect
  setup_choose_1
  answer = { :body => 11 }
  assert_equal false, @type.correct?(@problem, answer),
    "Answer #{answer[:body]} should be incorrect"
end

def test_choose_n_correct
  setup_choose_n
  answers = { 10 => 1, 11 => 1, 12 => 0, 13 => 0 }
  assert @type.correct?(@problem, answers),
    "Answers #{answers_sentence(@problem.correct_answers)} should be correct"
end

def test_choose_n_invalid_choice
  setup_choose_n
  # TODO: change this to be invalid, not just wrong
  answers = { 10 => 0, 11 => 1, 12 => 0, 13 => 0 }
  assert_equal false, @type.correct?(@problem, answers),
    "One of the answers, #{answers_sentence(@problem.answers)}, isn't a valid
choice, but should still mark wrong"
end

def test_choose_n_one_incorrect
  setup_choose_n
  answers = { 10 => 0, 11 => 1, 12 => 0, 13 => 0 }
  assert_equal false, @type.correct?(@problem, answers),
    "One of the answers, #{answers_sentence(@problem.correct_answers)}, isn't
correct, so should return number of correct"
end

def test_choose_n_all_incorrect
  setup_choose_n

```

```
answers = { 10 => 0, 11 => 0, 12 => 1, 13 => 1 }
assert_equal false, @type.correct?(@problem, answers),
  "All answers, #{answers_sentence(@problem.correct_answers)}, should be
incorrect"
end
```

```
def test_rank_correct
  setup_rank
  answer = { 16 => 1, 17 => 2, 14 => 3, 15 => 4 } # should be id's of answers
  assert @type.correct?(@problem, answer),
    "Ranking #{answer} should be correct"
end
```

```
def test_rank_incorrect
  setup_rank
  answer = { 16 => 1, 14 => 2, 15 => 3, 16 => 4 }
  assert_equal false, @type.correct?(@problem, answer),
    "Ranking #{answer} should be incorrect"
end
```

```
def test_fill_in_correct
  setup_fill_in
  answer = { :body => "Neil Heffernan" }
  assert @type.correct?(@problem, answer),
    "Answer #{answer[:body]} should be correct"
end
```

```
def test_fill_in_incorrect
  setup_fill_in
  answer = { :body => "Anything else" }
  assert_equal false, @type.correct?(@problem, answer),
    "Answer #{answer[:body]} should be incorrect"
```

```
end
def test_fill_in_expected_incorrect
  setup_fill_in
  answer = {:body => "Jimmy Schementi"}
  assert_equal false, @type.correct?(@problem, answer)
  "Answer #{answer[:body]} should be expectedly incorrect"
end
```

```
def test_algebra_correct
  setup_algebra
  answer = {:body => 100}
  assert @type.correct?(@problem, answer),
    "Answer #{answer[:body]} should be correct"
end
```

```
def test_algebra_incorrect
  setup_algebra
  answer = {:body => 99}
  assert_equal false, @type.correct?(@problem, answer),
    "Answer #{answer[:body]} should be incorrect"
end
```

private

```
# makes it possible to call "setup_algebra" and have the algebra
# problem type and test problem loaded into respective instance variables.
# this works for any problem and problem type that are both named after
# the problem type name
def method_missing(method_id, *args)
  begin
    super
```

```
rescue => e
  name = method_id.id2name.split("_")
  if name.first == "setup"
    setup_test(name[1..-1].join("_").to_sym)
  else
    raise e
  end
end
end
def setup_test(type)
  @type = problem_types(type)
  @problem = problems(type)
end

def answers_sentence(answers)
  answers.collect{|answer| answer.value}.to_sentence
end

end
```

progress_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ProgressTest < Test::Unit::TestCase
```

```
  fixtures :progresses
```

```
  #How sad that this test is the most important one for any of the models?
```

```
  def test_CheckCRUD
```

```
    p = Progress.create!
```

```
    assert(!p.is_done)
```

```
    p.is_done = true
```

```
    assert(p.is_done)
```

```
    p.save!
```

```
    id = p.id
```

```
    p = Progress.find(id)
```

```
    assert(p.is_done)
```

```
    p.destroy
```

```
    begin
```

```
      p = Progress.find(id)
```

```
      assert(false)
```

```
    rescue
```

```
      assert(true)
```

```
    end
```

```
    p = Progress.create! :is_done=>true
```

```
    p.state[:section][0] = 1
```

```
    p.save!
```



```
assert(1 == p.state[:section][0])

id = p.id
p = Progress.find(id)
#print p.to_yaml
assert(1 == p.state[:section][0])
end
end
```

report_generator_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ReportGeneratorTest < Test::Unit::TestCase  
  fixtures :report_generators
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

right_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class RightTest < Test::Unit::TestCase  
  fixtures :rights
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

role_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class RoleTest < Test::Unit::TestCase  
  fixtures :roles
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    flunk
```

```
  end
```

```
end
```

scaffold_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class ScaffoldTest < Test::Unit::TestCase
```

```
  fixtures :scaffolds, :problems
```

```
  def test_relationships
```

```
    belongs_to_test scaffolds(:first), :problem, Problem, "scaffold"
```

```
    has_many_test scaffolds(:first), :problems, Problem, "scaffold"
```

```
  end
```

```
end
```

school_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class SchoolTest < Test::Unit::TestCase
```

```
  fixtures :schools, :user_roles, :enrollments, :student_classes, :teacher_classes,  
  :districts
```

```
  def test_name_and_district_required
```

```
    school = School.new
```

```
    assert !school.valid?
```

```
    assert school.errors.invalid?(:name)
```

```
    assert school.errors.invalid?(:district_id)
```

```
    school.name = "Some school"
```

```
    school.district = District.find :first
```

```
    assert school.valid?
```

```
  end
```

```
  def test_classes
```

```
    @wems = schools(:wems)
```

```
    teachers = @wems.teachers
```

```
    classes = @wems.classes
```

```
    teachers.each do |teacher|
```

```
      teacher.classes.each do |klass|
```

```
        classes.member?(klass)
```

```
      end
```

```
    end
```

```
  end
```

```
  def test_teachers
```

```
@wems = schools(:wems)
teachers = @wems.teachers
assert teachers.member?(user_roles(:woods))
assert teachers.member?(user_roles(:forrest))
assert teachers.member?(user_roles(:black))
end
```

```
def test_students
```

```
  @wems = schools(:wems)
  assert_respond_to(@wems, :students)
  assert_kind_of(Array, @wems.students)
  @wems.students.each do |student|
    assert_kind_of(Student, student)
  end
  assert @wems.students.member?(user_roles(:brian))
  assert @wems.students.member?(user_roles(:shawn))
end
```

```
end
```

section_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
def withing_small_margin(a,b)
```

```
  e = 0.08
```

```
  return ((a+e)>b) && ((a-e)<b)
```

```
end
```

```
class SectionTest < Test::Unit::TestCase
```

```
  fixtures :sections
```

```
  fixtures :section_links
```

```
  TestPaths = {:r5_w5 => [{18 => true},{1 => true},{7 => true},{8 => true},{24 => true},{23 => false},{5 => false},{17 => false},{35 => false},{31 => false},{39 => false}],
```

```
    :z0 => [{18 => false},{17 => true},{5 => true},{10 => true},{26 => true},{20 => true},{2 => false},{32 => true},{14 => true},{33 => false},{27 => true},{11 => false},{29 => false}],
```

```
    :first_prob => [{18 => true}],
```

```
    :all_wrong => [{18 => false},{17 => false},{35 => false},{31 => false},{39 => false},{19 => false},{28 => false},{5 => false},{16 => false},{6 => false},{36 => false}],
```

```
    :every_other => [{18 => true},{1 => false},{17 => true},{37 => false},{3 => true},{13 => false},{34 => true},{38 => false},{15 => true},{30 => false},{33 => true}],
```

```
    :all_correct => [{18 => true},{1 => true},{7 => true},{8 => true},{24 => true},{23 => true},{25 => true},{12 => true},{22 => true},{4 => true},{21 => true}]}
```

```
  # Replace this with your real tests.
```

```
  def test_CheckCRUD
```

```
    s = LinearSection.create! :sequence_id => 1
```

```
  end
```



```
def test_CAT_CRUD
  s = CatThreePMaxInformationSection.create! :sequence_id => 1
end
```

```
def follow_CAT_path(seq)
  s = sections(:CAT_NEW)
  p = Progress.new.set_defaults
  s.initProgress(p)
```

```
  seq.each { |h|
    k = h.keys #this is type weirdness? not sure, but it thinks it is a Fixnum without
this step.
```

```
    p_id = k[0]
    correctness = h[p_id]
    r = s.nextSequence(p)
    prob = r.pop[0]
    print "Case (" + p_id.to_s + " => " + correctness.to_s + "): " + prob.to_s + "\n"
    assert(p_id == prob)
    p.state[:answer_history][prob] = correctness
  }
end
```

```
def test_CAT_path_5r_5w
  print "r5 w5\n"
  follow_CAT_path(self.class::TestPaths[:r5_w5])
end
```

```
def test_CAT_path_z0
  print "z=0\n"
```

```
    follow_CAT_path(self.class::TestPaths[:z0])
end

def test_CAT_path_first_prob
  print "Check First Problem\n"
  follow_CAT_path(self.class::TestPaths[:first_prob])
end

def test_CAT_path_all_correct
  print "All Correct\n"
  follow_CAT_path(self.class::TestPaths[:all_correct])
end

def test_CAT_path_all_wrong
  print "All Wrong\n"
  follow_CAT_path(self.class::TestPaths[:all_wrong])
end

def test_CAT_path_every_other
  print "Every Other\n"
  follow_CAT_path(self.class::TestPaths[:every_other])
end

# def test_CAT3_proper_estimator
#   s = sections(:CAT_CV)
#
#   print "Start CAT proper\n"
#
#   (-3..3).each { |i|
#     print "For i="+i.to_s
```

```

# p = Progress.new.set_defaults
# s.initProgress(p)
# while (r = s.nextSequence(p))
#   assert(1==r.size)
#   prob = r.pop[0]
#   if s.Pof(p.state[:section][s.id]['assigned_problems'][prob],1)> 0.50
#     p.state[:answer_history][prob] = true
#   else
#     p.state[:answer_history][prob] = false
#   end
#   #print "Theta hat = " + s.theta_hat(p).to_s+"\n"
#   #print "standard error = " + s.SE(p).to_s+"\n"
#   end
#
# print " we got "+s.theta_hat(p).to_s+".\n"
# assert(withing_small_margin(i,s.theta_hat(p)))
# }
# end

```

#Sample run: (Target z=0.5)

#: response, estimate, standard error

_: _____, 0.5

1: correct, 3.0, 6.0

2: incorrect, -3.0, 1.65

3: correct, 1.06, 1.34

4: incorrect, 0.67, 0.74

5: incorrect, 0.31, 0.48

6: correct, 0.42, 0.37

7: correct, 0.56, 0.30

#

```

def test_CAT3_perfect
  s = sections(:CAT3)
  p = Progress.new.set_defaults
  s.initProgress(p)

  print "Start CAT perfect\n"

  assert(withing_small_margin(0,s.theta_hat(p)))
  #print s.I(0,0).to_s+"\n"

  r = s.nextSequence(p)
  assert(1==r.size)
  assert(1==r.pop[0])
  p.state[:answer_history][1] = true
  print "Theta hat = " + s.theta_hat(p).to_s+"\n"
  assert(withing_small_margin(3,s.theta_hat(p)))
  #print "standard error = " + s.SE(p).to_s+"\n"
  #assert(withing_small_margin(6,s.SE(p)))

  r = s.nextSequence(p)
  assert(1==r.size)
  assert(2==r.pop[0])
  p.state[:answer_history][2] = false
  print "Theta hat = " + s.theta_hat(p).to_s+"\n"
  assert(withing_small_margin(-3,s.theta_hat(p)))
  #print "standard error = " + s.SE(p).to_s+"\n"
  #assert(withing_small_margin(1.65,s.SE(p)))

  r = s.nextSequence(p)
  assert(1==r.size)

```

```
assert(3==r.pop[0])
p.state[:answer_history][3] = true
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(1.06,s.theta_hat(p)))
```

```
r = s.nextSequence(p)
assert(1==r.size)
assert(4==r.pop[0])
p.state[:answer_history][4] = false
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(0.67,s.theta_hat(p)))
```

```
r = s.nextSequence(p)
assert(1==r.size)
assert(5==r.pop[0])
p.state[:answer_history][5] = false
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(0.31,s.theta_hat(p)))
```

```
r = s.nextSequence(p)
assert(1==r.size)
assert(6==r.pop[0])
p.state[:answer_history][6] = true
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(0.42,s.theta_hat(p)))
```

```
r = s.nextSequence(p)
assert(1==r.size)
assert(7==r.pop[0])
p.state[:answer_history][7] = true
```

```
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(0.56,s.theta_hat(p)))
```

```
end
```

```
# #: response, estimate, standard error
```

```
# _: _____, 0.5,
```

```
# 1: correct, 3.0, 6.0
```

```
# 2: incorrect, -3.0, 1.65
```

```
# 3: correct, 1.06, 1.34
```

```
# 4: incorrect, 0.67, 0.74
```

```
# 5: incorrect, 0.31, 0.48
```

```
# 6: incorrect, 0.01, 0.46
```

```
# (8): correct, 0.10, 0.37
```

```
def test_CAT3_deviation
```

```
  s = sections(:CAT3)
```

```
  p = Progress.new.set_defaults
```

```
  s.initProgress(p)
```

```
  print "Start CAT deviation\n"
```

```
  assert(withing_small_margin(0,s.theta_hat(p)))
```

```
  r = s.nextSequence(p)
```

```
  assert(1==r.size)
```

```
  assert(1==r.pop[0])
```

```
  p.state[:answer_history][1] = true
```

```
  print "Theta hat = " + s.theta_hat(p).to_s+"\n"
```

```
  assert(withing_small_margin(3,s.theta_hat(p)))
```

```
  #print "standard error = " + s.SE(p).to_s+"\n"
```

```

#assert(withing_small_margin(6,s.SE(p)))

r = s.nextSequence(p)
assert(1==r.size)
assert(2==r.pop[0])
p.state[:answer_history][2] = false
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(-3,s.theta_hat(p)))
#print "standard error = " + s.SE(p).to_s+"\n"
#assert(withing_small_margin(1.65,s.SE(p)))

r = s.nextSequence(p)
assert(1==r.size)
assert(3==r.pop[0])
p.state[:answer_history][3] = true
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(1.06,s.theta_hat(p)))

r = s.nextSequence(p)
assert(1==r.size)
assert(4==r.pop[0])
p.state[:answer_history][4] = false
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(0.67,s.theta_hat(p)))

r = s.nextSequence(p)
assert(1==r.size)
assert(5==r.pop[0])
p.state[:answer_history][5] = false
print "Theta hat = " + s.theta_hat(p).to_s+"\n"

```

```
assert(withing_small_margin(0.31,s.theta_hat(p)))
```

```
r = s.nextSequence(p)
```

```
assert(1==r.size)
```

```
assert(6==r.pop[0])
```

```
p.state[:answer_history][6] = false
```

```
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
```

```
assert(withing_small_margin(0.01,s.theta_hat(p)))
```

```
r = s.nextSequence(p)
```

```
assert(1==r.size)
```

```
assert(8==r.pop[0])
```

```
p.state[:answer_history][8] = true
```

```
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
```

```
assert(withing_small_margin(0.10,s.theta_hat(p)))
```

```
end
```

```
# #: response, estimate, standard error
```

```
# _: _____, 0.5,
```

```
# 1: incorrect, -3.0, 6.0
```

```
# 3: correct, -1.32, 4.09
```

```
def test_CAT3_early_deviation
```

```
  s = sections(:CAT3)
```

```
  p = Progress.new.set_defaults
```

```
  s.initProgress(p)
```

```
  print "Start CAT early deviation\n"
```

```
  assert(withing_small_margin(0,s.theta_hat(p)))
```



```
r = s.nextSequence(p)
assert(1==r.size)
assert(1==r.pop[0])
p.state[:answer_history][1] = false
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(-3,s.theta_hat(p)))
#print "standard error = " + s.SE(p).to_s+"\n"
#assert(withing_small_margin(6,s.SE(p)))
```

```
r = s.nextSequence(p)
assert(1==r.size)
assert(3==r.pop[0])
p.state[:answer_history][3] = true
print "Theta hat = " + s.theta_hat(p).to_s+"\n"
assert(withing_small_margin(-1.32,s.theta_hat(p)))
```

```
end
```

```
def test_ProblemSectionDoesNotReturnID
```

```
  s = sections(:zero)
  p = Progress.new.set_defaults
  s.initProgress(p)
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(747==r.pop[0])
  assert(nil==s.nextSequence(p))
```

```
end
```

```
def test_ProblemSection
```

```
s = sections(:one)
p = Progress.new.set_defaults
s.initProgress(p)
r = s.nextSequence(p)
assert(1==r.size)
assert(1==r.pop[0])
assert(nil==s.nextSequence(p))
end
```

```
def test_LinearSection
s = sections(:Linear1_2_3)
p = Progress.new.set_defaults
s.initProgress(p)
r = s.nextSequence(p)
assert(1==r.size)
assert(1==r.pop[0])
r = s.nextSequence(p)
assert(1==r.size)
assert(2==r.pop[0])
r = s.nextSequence(p)
assert(1==r.size)
assert(3==r.pop[0])
assert(nil==s.nextSequence(p))
end
```

```
#this test runs a random curriculum twice,
#each time it runs it, it checks that each number
#came up that was in it
#and that the curriculum was out after that
#and that the curriculum returned the right number of assistments.
```

```
#It then, having stored which request each assistment id came up on
#makes sure that the runs where not identical.
#
#Admittedly, there exists no good way to test for randomness.
#This doesn't even make a good attempt at checkign that something
#wasn't a fluke, it DOES check that it doesn't repeat.
#If by chance it does, it doesn't handle that.
def test_RandomOrderSection
  s = sections(:Random1_to_9)

  p = Progress.new.set_defaults
  s.initProgress(p)
  run1 = Hash.new
  (1..9).each do |i|
    r = s.nextSequence(p)
    assert(1==r.size)
    run1[r.pop[0]]=i
  end
  assert(nil==run1[0])
  assert(nil!=run1[1])
  assert(nil!=run1[2])
  assert(nil!=run1[3])
  assert(nil!=run1[4])
  assert(nil!=run1[5])
  assert(nil!=run1[6])
  assert(nil!=run1[7])
  assert(nil!=run1[8])
  assert(nil!=run1[9])
  assert(nil==s.nextSequence(p))
end
```

```
p = Progress.new.set_defaults
s.initProgress(p)
run2 = Hash.new
(1..9).each do |i|
  r = s.nextSequence(p)
  assert(1==r.size)
  run2[r.pop[0]]=i
end
assert(nil==run2[0])
assert(nil!=run2[1])
assert(nil!=run2[2])
assert(nil!=run2[3])
assert(nil!=run2[4])
assert(nil!=run2[5])
assert(nil!=run2[6])
assert(nil!=run2[7])
assert(nil!=run2[8])
assert(nil!=run2[9])
assert(nil==s.nextSequence(p))

#make sure they differed
times_not_equal = 0
(1..9).each do |i|
  if(run1[i]!=run2[i])
    times_not_equal += 1
  end
end

assert(0<times_not_equal)
end
```

```
def test_NoRepeatsSection
  s = sections(:NoRepeat0)
  p = Progress.new.set_defaults
  s.initProgress(p)
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(1==r.pop[0])
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(2==r.pop[0])
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(3==r.pop[0])
  assert(nil==s.nextSequence(p))
end
```

```
def test_NumericLimitSection
  s = sections(:NumericLimit0)
  p = Progress.new.set_defaults
  s.initProgress(p)
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(1==r.pop[0])
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(2==r.pop[0])
  r = s.nextSequence(p)
  assert(1==r.size)
  assert(3==r.pop[0])
end
```

```

    assert(nil==s.nextSequence(p))
end

def test_ChooseConditionSection
  s = sections(:ChooseCondition0)

  values = {}
  #make sure it occasionally chooses all of them
  (0..15).each { |i|
    p = Progress.new.set_defaults
    s.initProgress(p)
    r = s.nextSequence(p)
    assert(1==r.size)
    values[r.pop[0]] = 1
    assert(nil==s.nextSequence(p))
  }

  (1..3).each { |i|
    assert(1==values[i])
  }
end

```

```

def test_ChooseConditionSectionWeighted
  s = sections(:ChooseConditionW)

  values = []
  values[1] = 0
  values[2] = 0
  #make sure it occasionally chooses all of them
  (1..1000).each { |i|

```

```

p = Progress.new.set_defaults
s.initProgress(p)
r = s.nextSequence(p)
assert(1==r.size)
values[r.pop[0]] += 1
assert(nil==s.nextSequence(p))
}

assert((values[1]>50) && (values[1]<150))
assert((values[2]>850) && (values[2]<950))
assert(1000==(values[1]+values[2]))
end

def test_GroupSequencesSectionPartials
s = sections(:GroupSequencesFromChildPartials)

#First make sure it returns everything
p = Progress.new.set_defaults
s.initProgress(p)

r = s.nextSequence(p)
assert(2==r.length)
assert(2==r.size)
assert(1==r.pop[0])
assert(2==r.pop[0])

r = s.nextSequence(p)
assert(1==r.size)
assert(3==r.pop[0])
assert(nil==s.nextSequence(p))

```

```
end
```

```
def test_GroupSequencesSectionNoPartials
```

```
  s = sections(:GroupSequencesFromChildNoPartials)
```

```
  #First make sure it returns everything
```

```
  p = Progress.new.set_defaults
```

```
  s.initProgress(p)
```

```
  r = s.nextSequence(p)
```

```
  assert(2==r.length)
```

```
  assert(2==r.size)
```

```
  assert(1==r.pop[0])
```

```
  assert(2==r.pop[0])
```

```
  assert(nil==s.nextSequence(p))
```

```
end
```

```
def test_TemporalLimitSection
```

```
  s = sections(:TemporalLimit0)
```

```
  #First make sure it returns everything
```

```
  p = Progress.new.set_defaults
```

```
  s.initProgress(p)
```

```
  r = s.nextSequence(p)
```

```
  assert(1==r.size)
```

```
  assert(1==r.pop[0])
```

```
  r = s.nextSequence(p)
```

```
  assert(1==r.size)
```

```
  assert(2==r.pop[0])
```



```
r = s.nextSequence(p)
assert(1==r.size)
assert(3==r.pop[0])
assert(nil==s.nextSequence(p))

#Then that it stops after time correctly
p = Progress.new.set_defaults
s.initProgress(p)
r = s.nextSequence(p)
assert(1==r.size)
assert(1==r.pop[0])
sleep(1)
assert(nil==s.nextSequence(p))

#Then that it always starts
p = Progress.new.set_defaults
s.initProgress(p)
sleep(1)
r = s.nextSequence(p)
assert(1==r.size)
assert(1==r.pop[0])
r = s.nextSequence(p)
assert(1==r.size)
assert(2==r.pop[0])
sleep(1)
assert(nil==s.nextSequence(p))

end

def test_TestGettingOfDefaultParameter
```

```
s = sections(:NumericLimit1)
p = Progress.new.set_defaults
s.initProgress(p)
assert(nil==s.nextSequence(p))
end
```

```
def test_SectionCreation_ChildForLeaf
  s = ProblemSection.new
  c = LinearSection.new
```

```
  errored = false
```

```
  begin
    s.addChild(c)
  rescue
    errored = true
  end
```

```
  assert(errored)
end
```

```
def test_SectionCreation_ChildForNonLeaf
  s = LinearSection.new
  c = LinearSection.new
```

```
  worked = true
```

```
  s.addChild(c)
```

```
    assert(worked)
  end

  def test_SectionCreation_SectionParameters
    s = GroupSequencesSection.new

    failed = false
    begin
      s.parameter("NumberToGroup", 2)
    rescue
      failed = true
    end
    assert(failed)

    worked = true
    begin
      s.parameter("NumToGroup", 2)
    rescue
      print $!,"\n"
      worked = false
    end
    assert(worked)

  end

  #need test for Termiante On Empty
  #Ech! horrible test case!
  #Uses build for performance reasons
  def test_RandomIterSection
    s = sections(:RandomIterNoWeights)
```

```

counts = Hash.new(0)

#make sure it has the right items and number of them in a valid sequence
#500 times means it should hit all the paths
(1..12000).each { |u|
  p = Progress.new(:state=>{:section => {}})
  s.initProgress(p)

  entries = Hash.new(false)
  entries[1] = true
  entries[2] = true
  entries[3] = true
  entries[4] = true
  entries[5] = true

  seq_rec = ""

  (1..5).each { |i|
    r = s.nextSequence(p)
    assert(nil!=r)
    val = r.pop[0]
    assert(entries[val])
    entries[val]=false

    #There was only one there
    assert(nil==r.pop)

    seq_rec << val.to_s
  }
}

```

```

assert(nil==s.nextSequence(p))
counts[seq_rec] += 1
}

#make sure they are all within a close range of each other
counts.each { |seq,count|
  #so large because the ruby random nubmer generator sucks.
  assert((count>=60) && (count<=140), seq+"("+count.to_s+")")
}
end

#need test for Termiante On Empty
#Ech! horrible test case!
#Uses build for performance reasons
def test_RandomIterSectionWeighted
  s = sections(:RandomIterWeights)

  counts = Hash.new(0)

  #make sure it has the right items and number of them in a valid sequence
  #500 times means it should hit all the paths
  (1..10000).each { |u|
    p = Progress.new(:state=>{:section => {}})
    s.initProgress(p)

    entries = Hash.new(false)
    entries[1] = true
    entries[2] = true
    entries[3] = true
    entries[4] = true
  }
}

```

```
entries[5] = true

seq_rec = ""

(1..2).each { |i|
  r = s.nextSequence(p)
  assert(nil!=r)
  val = r.pop[0]
  assert(entries[val])
  entries[val]=false

  #There was only one there
  assert(nil==r.pop)

  seq_rec << val.to_s
}
assert(nil==s.nextSequence(p))
counts[seq_rec] += 1
}
assert((counts["12"]<1250) && (counts["12"]>750))
assert((counts["21"]<9250) && (counts["21"]>8750))
end
end
```

sequence_ownership_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class SequenceOwnershipTest < Test::Unit::TestCase  
  fixtures :sequence_ownerships
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

sequence_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class SequenceTest < Test::Unit::TestCase
  fixtures :sequences
  fixtures :sections
  #fixtures :progresses

  # Replace this with your real tests.
  #def test_CheckCRUD
  # s = Sequence.new(:name => "Blank Test Sequence")
  #end

  #make sure a curriculum can find it's head problem,
  #generate a progress and get a problem.
  #This is the very basic functionality.
  def test_SimpleSingleSection
    c = sequences(:problem_section_1)
    s = Student.create!
    assert(1==c.nextProblem(s)[0])
    assert(nil==c.nextProblem(s))
  end

  #This tests what happens when a section contains it's self as a child.
  #This is to make sure that if an error DOES occure, the code handles it
  #and doesn't halt the system
  # def test_SectionLoop
  # c = curriculums(:Loop)
  # assert(nil==c.nextProblem(nil))
end
```



```
# end
```

```
def test_SequenceSequenceUsage
  c = sequences(:SequencedCurriculum)
  print c.nextProblem(nil)
  print c.nextProblem(nil)
  print c.nextProblem(nil)
  assert(1==c.nextProblem(nil))
  assert(2==c.nextProblem(nil))
  assert(nil==c.nextProblem(nil))
end
```

```
def test_SequenceSequenceUsage
  c = sequences(:PrettyCuric0)
  #c.extend SequenceBuilderUtils
  print c.DOTfile
  assert(c.DOTfile ==
    "digraph g {\n"+
    " graph [ rankdir = \"LR\" ];\n"+
    " node [ fontsize = \"16\" shape = \"ellipse\" ];\n"+
    " edge [ ];\n"+
    " section_id_93 [ label = \"<fName>Section #93 <fType>ProblemSection\" shape
= \"record\" ];\n"+
    " section_id_94 [ label = \"<fName>Section #94 <fType>NumericLimitSection
<f1>\" shape = \"record\" ];\n"+
    " section_id_90 [ label = \"<fName>Section #90 <fType>LinearSection <f1>
<f2> <f3> <f4>\" shape = \"record\" ];\n"+
    " section_id_91 [ label = \"<fName>Section #91 <fType>ProblemSection\" shape
= \"record\" ];\n"+
    " section_id_92 [ label = \"<fName>Section #92 <fType>ProblemSection\" shape
= \"record\" ];\n"+
```

```
" section_id_94:f1 -> section_id_92:fName [ id = 1 ];\n"+  
" section_id_90:f1 -> section_id_91:fName [ id = 2 ];\n"+  
" section_id_90:f2 -> section_id_92:fName [ id = 3 ];\n"+  
" section_id_90:f3 -> section_id_94:fName [ id = 4 ];\n"+  
" section_id_90:f4 -> section_id_93:fName [ id = 5 ];\n"+  
"};");  
end  
end
```

student_class_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'

class StudentClassTest < Test::Unit::TestCase
  fixtures :student_classes, :user_roles, :enrollments,
          :teacher_classes, :class_assignments, :sequences

  def test_schools

  end

  def test_name_and_grade_required
    student_class = StudentClass.new
    assert !student_class.valid?
    assert student_class.errors.invalid?(:name)
    assert student_class.errors.invalid?(:grade)
    student_class.name = "Class numero uno"
    student_class.grade = "6"
    assert student_class.valid?
  end

  def test_find_students
    woods1 = student_classes(:woods_1)
    woods2 = student_classes(:woods_2)
    forrest1 = student_classes(:forrest_1)
    forrest2 = student_classes(:forrest_2)

    brian = user_roles(:brian)
    shawn = user_roles(:shawn)
```

```
assert woods1.students.member?(brian)
assert woods1.students.member?(shawn)

assert woods2.students.member?(shawn)

assert forrest1.students.member?(brian)
assert !forrest1.students.member?(shawn)

assert forrest2.students.member?(brian)
assert forrest2.students.member?(shawn)

end

def test_find_teachers
  woods1 = student_classes(:woods_1)
  woods2 = student_classes(:woods_2)
  forrest1 = student_classes(:forrest_1)
  forrest2 = student_classes(:forrest_2)

  woods = user_roles(:woods)
  forrest = user_roles(:forrest)

  assert woods1.teachers.member?(woods)
  assert !woods1.teachers.member?(forrest)

  assert forrest1.teachers.member?(forrest)
  assert !forrest1.teachers.member?(woods)

  assert forrest2.teachers.member?(woods)
```

```
    assert forrest2.teachers.member?(forrest)
end

def test_find_assignments
  get_classes
  get_teachers
  get_sequences

  assert @woods1.sequences.member?(@seq1)
  assert @woods1.sequences.member?(@seq2)
  assert @woods2.sequences.member?(@seq3)

  assert_member(@forrest1.sequences, @seq4)
  assert_member(@forrest1.sequences, @seq5)
  assert_member(@forrest2.sequences, @seq6)

  assert @woods1.sequences[0] = @seq2
  assert @woods1.sequences[1] = @seq1

  assert @forrest1.sequences[0] = @seq4
  assert @forrest1.sequences[1] = @seq5

  # If we can get ordering to work right, we can run these:
  #  assert @forrest1.sequences[0].first?
  #  @forrest1.sequences[1].move_up
  #
  #  assert @forrest1.sequences[0] = @seq5
  #  assert @forrest1.sequences[1] = @seq4
end
```

```
private
  def assert_member (list, item)
    assert list.member?(item)
  end

  def get_classes
    @woods1 = student_classes(:woods_1)
    @woods2 = student_classes(:woods_2)
    @forrest1 = student_classes(:forrest_1)
    @forrest2 = student_classes(:forrest_2)
  end

  def get_teachers
    @woods = user_roles(:woods)
    @forrest = user_roles(:forrest)
  end

  def get_sequences
    @seq1 = sequences(:seq1)
    @seq2 = sequences(:seq2)
    @seq3 = sequences(:seq3)
    @seq4 = sequences(:seq4)
    @seq5 = sequences(:seq5)
    @seq6 = sequences(:seq6)
  end
end
```

student_files_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class StudentFilesTest < Test::Unit::TestCase  
  fixtures :student_files
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

student_parent_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class StudentParentTest < Test::Unit::TestCase  
  fixtures :student_parents
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

student_report_comment_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class StudentReportCommentTest < Test::Unit::TestCase  
  fixtures :student_report_comments
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```


student_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class StudentTest < Test::Unit::TestCase
```

```
  fixtures :user_roles, :student_classes, :teacher_classes, :enrollments, :users
```

```
  def setup
```

```
    @student = user_roles(:jimmy)
```

```
  end
```

```
  def test_school
```

```
    belongs_to_test @student, :school, School, "student"
```

```
  end
```

```
  def test_user
```

```
    belongs_to_test @student, :user, User, "student"
```

```
  end
```

```
  def test_enrollments
```

```
    has_many_test @student, :enrollments, Enrollment, "student"
```

```
  end
```

```
  def test_classes
```

```
    has_many_test @student, :classes, StudentClass, "student", :through =>  
:enrollments
```

```
  end
```

```
  def test_joinable_classes
```

```
    assert_equal (@student.school.classes - @student.classes),  
@student.joinable_classes,
```

```
    "Students joinable classes are suppose to be the classes they are not in."  
end
```

```
def test_find_classes  
  brian = user_roles(:brian)  
  shawn = user_roles(:shawn)  
  woods1 = student_classes(:woods_1)  
  woods2 = student_classes(:woods_2)  
  assert brian.classes.member?(woods1), "Student has a incorrect class"  
  assert shawn.classes.member?(woods1), "Student has a incorrect class"  
  assert shawn.classes.member?(woods2), "Student has a incorrect class"  
end
```

```
def test_teachers  
  brian = user_roles(:brian)  
  shawn = user_roles(:shawn)  
  woods = user_roles(:woods)  
  forrest = user_roles(:forrest)  
  
  # Ensure that Brian & Shawn know Mr. Woods and Ms. Forrest as their teachers  
  assert brian.teachers.member?(woods), "Student has a incorrect teacher"  
  assert brian.teachers.member?(forrest), "Student has a incorrect teacher"  
  assert shawn.teachers.member?(woods), "Student has a incorrect teacher"  
  assert shawn.teachers.member?(forrest), "Student has a incorrect teacher"  
  
  # Ensure Mr. Woods only shows up once for Shawn  
  assert_equal(2, brian.teachers.size)  
  assert_equal(2, shawn.teachers.size)  
end
```

```
def test_before_save_role
  user = User.create :login => "george",
                    :email => "george@jungle.org",
                    :password => "ilovetoswing",
                    :password_confirmation => "ilovetoswing"
  user.user_roles
end

def test_belongs_to_class
  @student = user_roles(:brian)
  assert @student.belongs_to_class?(teacher_classes(:woods_woods1))
  assert !@student.belongs_to_class?(teacher_classes(:woods_woods2))
end

end
```

teacher_class_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class TeacherClassTest < Test::Unit::TestCase
```

```
  fixtures :teacher_classes, :user_roles, :student_classes
```

```
  def test_teacher
```

```
    belongs_to_test teacher_classes(:woods_woods1), :teacher, Teacher,  
"teacher_class"
```

```
  end
```

```
  def test_student_class
```

```
    belongs_to_test teacher_classes(:woods_woods1), :student_class, StudentClass,  
"teacher_class"
```

```
  end
```

```
end
```

teacher_parent_message_item_report_reference_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class TeacherParentMessageItemReportReferenceTest < Test::Unit::TestCase  
  fixtures :teacher_parent_message_item_report_references
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

teacher_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class TeacherTest < Test::Unit::TestCase
```

```
  fixtures :user_roles, :teacher_classes, :student_classes, :enrollments, :users, :schools
```

```
  def setup
```

```
    @teacher = user_roles(:woods)
```

```
  end
```

```
  def test_create_role
```

```
    user = User.create :login => "teacher_test", :password => "teacher",  
:password_confirmation => "teacher", :email => "teacher@wpi.edu"
```

```
    user.user_roles << Teacher.create( :school_id => schools(:wems))
```

```
    teacher = user.user_roles.first
```

```
    assert_not_nil(teacher.role)
```

```
    assert_equal(Role.find_by_name("Teacher"), teacher.role)
```

```
  end
```

```
  def test_teacher_classes
```

```
    assert_kind_of Array, @teacher.teacher_classes
```

```
    @teacher.teacher_classes.each do |klass|
```

```
      assert_kind_of Teacher, klass.teacher
```

```
      assert_equal @teacher.id, klass.teacher.id
```

```
    end
```

```
  end
```

```
  def test_student_classes
```

```
    assert_respond_to @teacher, :student_classes
```

```
    assert_kind_of Array, @teacher.student_classes
```

```
@teacher.student_classes.each do |klass|
  assert_kind_of StudentClass, klass
end
end

def test_school
  assert_respond_to @teacher, :school_id
  assert_respond_to @teacher, :school
  assert_kind_of School, @teacher.school
  assert_equal @teacher.school_id, @teacher.school.id
end

def test_classes
  assert_equal @teacher.student_classes(true), @teacher.classes,
    "classes is suppose to be an alias for student_classes"
end

def test_students
  classes.inject([]) { |students, klass| students.push(*klass.students) }.uniq
end

def test_name
  assert_equal @teacher.user.name, @teacher.name
end

def test_students_in_class
  [student_classes(:woods_1), student_classes(:woods_2)].each do |student_class|
    assert user_roles(:woods).classes.member?(student_class),
      "Teacher should be associated with this Class"
  end
end
```

end

def test_students

[user_roles(:brian), user_roles(:shawn)].each do |student|

assert user_roles(:woods).students.member?(student),

"Student should be associated with this teacher"

end

assert_equal 2, @teacher.students.size, "Class should only have a size of 2"

end

end

tutor_strategy_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class TutorStrategyTest < Test::Unit::TestCase  
  fixtures :tutor_strategies
```

```
  # Replace this with your real tests.
```

```
  def test_truth
```

```
    assert true
```

```
  end
```

```
end
```

user_detail_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class UserDetailTest < Test::Unit::TestCase
```

```
  fixtures :user_details
```

```
  def test_relationships
```

```
    belongs_to_test user_details(:jimmy), :user, User, "user_detail"
```

```
  end
```

```
  def test_name
```

```
    assert_equal "James M Schementi", user_details(:jimmy).name
```

```
    assert_equal "Test Person", user_details(:no_middle).name
```

```
  end
```

```
  def test_full_name
```

```
    assert_equal "James Michael Schementi", user_details(:jimmy).full_name
```

```
  end
```

```
  def test_birthday
```

```
    assert_equal user_details(:jimmy).birthdate, user_details(:jimmy).birthdate
```

```
  end
```

```
end
```

user_role_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class UserRoleTest < Test::Unit::TestCase
```

```
  fixtures :user_roles
```

```
  def setup
```

```
    @user_role = user_roles(:jimmy)
```

```
  end
```

```
  def test_role
```

```
    belongs_to_test @user_role, :role, Role, "user_role"
```

```
  end
```

```
  def test_user
```

```
    belongs_to_test @user_role, :user, User, "user_role"
```

```
  end
```

```
end
```

user_test.rb

```
require File.dirname(__FILE__) + '/../test_helper'
```

```
class UserTest < Test::Unit::TestCase
```

```
  # relationship tests
```

```
#####  
#
```

```
  def test_teacher; has_one_test users(:jimmy), :teacher, Teacher, "user"; end
```

```
  def test_student; has_one_test users(:jimmy), :student, Student, "user"; end
```

```
  def test_content_creator; has_one_test users(:jimmy), :content_creator,  
ContentCreator, "user"; end
```

```
  def test_administrator; has_one_test users(:jimmy), :administrator, Administrator,  
"user"; end
```

```
  def test_user_roles; has_many_test users(:jimmy), :user_roles, UserRole, "user";  
end
```

```
  def test_roles; has_many_test users(:jimmy), :roles, Role, "user", :through =>  
:user_roles; end
```

```
  def test_user_detail; has_one_test users(:jimmy), :user_detail, UserDetail, "user";  
end
```

```
  def test_actions; has_many_test users(:jimmy), :actions, Action, "user"; end
```

```
  def test_progresses; has_many_test users(:jimmy), :progresses, Progress, "user";  
end
```

```
  # validation tests
```

```
#####  
###
```

```
  def test_require_login
```

```

assert_no_difference User, :count do
  u = create_user(:email => nil, :login => nil)
  assert u.errors.on(:login)
end
end
def test_login_length
  assert_no_difference User, :count do
    u = create_user :login => "t"
    assert u.errors.on(:login)
    u = create_user :login => (1..102).to_a.collect{|i|j}.join
    assert u.errors.on(:login)
  end
end
def test_unique_login
  u = create_user :login => "quire"
  assert_no_difference User, :count do
    u = create_user :login => "qUire"
    assert u.errors.on(:login)
  end
end
def test_email_length
  assert_no_difference User, :count do
    u = create_user :email => "#{(1..100).to_a.collect{|i|j}.join}@bar.com"
    u.errors.on(:email)
    u = create_user :email => "a@b.c"
    u.errors.on(:email)
  end
end
def test_unique_email

```

```
u = create_user
assert_no_difference User, :count do
  u = create_user(:email => "quIre@example.com")
  assert u.errors.on(:email)
end
end
def test_require_email
  assert_no_difference User, :count do
    u = create_user(:email => nil)
    assert u.errors.on(:email)
  end
end
def test_require_password
  assert_no_difference User, :count do
    u = create_user(:password => nil)
    assert u.errors.on(:password)
  end
end
def test_require_password_confirmation
  assert_no_difference User, :count do
    u = create_user(:password_confirmation => nil)
    assert u.errors.on(:password_confirmation)
  end
end
def test_password_size
  assert_no_difference User, :count do
    u = create_user(:password => (1..3).to_a.collect{|i|'a'}.join,
:password_confirmation => (1..3).to_a.collect{|i|'a'}.join)
    assert u.errors.on(:password)
  end
end
```

```

    u = create_user(:password => (1..41).to_a.collect{|i|'a'}.join,
      :password_confirmation => (1..41).to_a.collect{|i|'a'}.join)
    assert u.errors.on(:password)
  end
end
def test_password_confirmation
  assert_no_difference User, :count do
    u = create_user(:password => "12345", :password_confirmation => "123456")
    assert u.errors.on(:password)
  end
end

# creation tests
#####

def test_create_user
  assert_difference User, :count do
    user = create_user
    assert !user.new_record?, "#{user.errors.full_messages.to_sentence}"
  end
end

# password tests
#####

def test_reset_password
  users(:quentin).update_attributes(:password => 'new password',
:password_confirmation => 'new password')
  assert_equal users(:quentin), User.authenticate('quentin@something.com', 'new
password')
end

```

```
end

def test_not_rehash_password
  users(:quentin).update_attributes(:email => 'quire22@bar.com')
  assert_equal users(:quentin), User.authenticate('quire22@bar.com', 'test')
end

# authentication tests
#####

def test_set_remember_token
  users(:quentin).remember_me
  assert_not_nil users(:quentin).remember_token
  assert_not_nil users(:quentin).remember_token_expires_at
end

def test_unset_remember_token
  users(:quentin).remember_me
  assert_not_nil users(:quentin).remember_token
  users(:quentin).forget_me
  assert_nil users(:quentin).remember_token
end

protected
def create_user(options = {})
  User.create({:email => 'quire@example.com', :password => 'quire',
:password_confirmation => 'quire' }.merge(options))
end
end
```


Appendix E: Sample Build Log

```
/home/ryansv/.cruise/projects/Assistments/work ryansv$ ../build_assistments.sh
```

```
/home/ryansv/.cruise/projects/Assistments/work/config/boot.rb:25:Warning:  
Gem::SourceIndex#search support for String patterns is deprecated (in  
/home/ryansv/.cruise/projects/Assistments/work)
```

```
/home/ryansv/.cruise/projects/Assistments/work/config/boot.rb:25:Warning:  
Gem::SourceIndex#search support for String patterns is deprecated (in  
/home/ryansv/.cruise/projects/Assistments/work)
```

Roles created

rake aborted!

Validation failed: Email has already been taken (See full trace by running task with --trace)

```
/home/ryansv/.cruise/projects/Assistments/work/config/boot.rb:25:Warning:  
Gem::SourceIndex#search support for String patterns is deprecated
```

```
sh: dropdb: command not found
```

```
sh: createdb: command not found /usr/local/bin/ruby -
```

```
I"/home/ryansv/.cruise/projects/Assistments/work/lib" -
```

```
I"/home/ryansv/.cruise/projects/Assistments/work/test"
```

```
"/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader.rb"
```

```
"test/unit/student_files_test.rb" "test/unit/action_problems_test.rb"
```

```
"test/unit/tutor_strategy_test.rb" "test/unit/content_creator_test.rb"
```

```
"test/unit/assistment_ownership_test.rb" "test/unit/teacher_class_test.rb"
```

```
"test/unit/sequence_ownership_test.rb" "test/unit/assistment_sequence_test.rb"
```

```
"test/unit/district_test.rb" "test/unit/db_files_test.rb" "test/unit/problem_test.rb"
```

```
"test/unit/sequence_test.rb" "test/unit/class_report_comment_test.rb"
```

```
"test/unit/progress_test.rb" "test/unit/assistment_test.rb" "test/unit/problem_type_test.rb"
```

```
"test/unit/teacher_test.rb" "test/unit/student_parent_test.rb" "test/unit/hint_test.rb"
```

```
"test/unit/teacher_parent_message_item_report_reference_test.rb"
```

```
"test/unit/scaffold_test.rb" "test/unit/action_hint_test.rb" "test/unit/enrollment_test.rb"
```

```
"test/unit/assistment_info_test.rb" "test/unit/user_role_test.rb" "test/unit/section_test.rb"
```

```
"test/unit/student_report_comment_test.rb" "test/unit/student_test.rb"
```

```
"test/unit/user_detail_test.rb" "test/unit/right_test.rb" "test/unit/school_test.rb"
```

```
"test/unit/student_class_test.rb" "test/unit/answer_test.rb"
```

```
"test/unit/logged_action_test.rb" "test/unit/user_test.rb"
```

```
"test/unit/report_generator_test.rb" "test/unit/action_test.rb"
```

"test/unit/daily_student_report_test.rb" "test/unit/role_test.rb" "test/unit/image_test.rb"
"test/unit/class_assignment_test.rb" (in /home/ryansv/.cruise/projects/Assistments/work)

[/home/ryansv/.cruise/projects/Assistments/work/config/boot.rb:25](#): Warning:
Gem::SourceIndex#search support for String patterns is deprecated

Loaded suite /usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader

Started

FFFF...F.....FFF...EEE.EEE.E...EEEE..EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEFEFE.EEEE
EFE....

Start CAT deviation

Theta hat = 2.999999999999996
Theta hat = -3
Theta hat = 1.029999999999998
Theta hat = 0.6699999999999981
Theta hat = 0.299999999999998
Theta hat = 0.009999999999998032
Theta hat = 0.09999999999999803

Start CAT early deviation

Theta hat = -3
Theta hat = -1.3900000000000002

Start CAT perfect

Theta hat = 2.999999999999996
Theta hat = -3
Theta hat = 1.029999999999998
Theta hat = 0.6699999999999981
Theta hat = 0.299999999999998
Theta hat = 0.4199999999999981
Theta hat = 0.5599999999999981

..r5 w5 Case (18 => true): 18 Case (1 => true): 1 Case (7 => true): 17 FALL Wrong Case
(18 => false): 18 Case (17 => false): 17 Case (35 => false): 35 Case (31 => false): 31
Case (39 => false): 39 Case (19 => false): 19 Case (28 => false): 28 Case (5 => false): 5
Case (16 => false): 16 Case (6 => false): 6 Case (36 => false): 36 .Every Other Case (18
=> true): 18 Case (1 => false): 1 Case (17 => true): 17 Case (37 => false): 37 Case (3 =>
true): 3 Case (13 => false): 13 Case (34 => true): 34 Case (38 => false): 38 Case (15 =>
true): 35 FCheck First Problem Case (18 => true): 18 .z=0 Case (18 => false): 18 Case
(17 => true): 17 Case (5 => true): 5 Case (10 => true): 10 Case (26 => true): 26 Case (20
=> true): 20 Case (2 => false): 2 Case (32 => true): 32 Case (14 => true): 14 Case (33 =>
false): 33 Case (27 => true): 27 Case (11 => false): 11 Case (29 => false): 29

.....

```
digraph g { graph [ rankdir = "LR" ]; node [ fontsize = "16" shape = "ellipse" ]; edge [ ];
};FE.....E.E...F.....E.....EE.....F...
```

Finished in 337.520154 seconds.

- 1) Failure: test_truth(ActionHintTest) [[./test/unit/action_hint_test.rb:8](#)]: Flunked.
- 2) Failure: test_truth(ActionProblemTest) [[./test/unit/action_problems_test.rb:8](#)]: Flunked.
- 3) Failure: test_truth(ActionTest) [[./test/unit/action_test.rb:8](#)]: Flunked.
- 4) Failure: test_acts_as_list(AnswerTest) [[./test/unit/./test_helper.rb:106](#):in `belongs_to_test' [./test/unit/./test_helper.rb:48](#):in `assert_acts_as_list' [./test/unit/answer_test.rb:16](#):in `test_acts_as_list']: answer does not have a problem method. <false> is not true.
- 5) Failure: test_truth(AssistmentInfoTest) [[./test/unit/assistment_info_test.rb:8](#)]: Flunked.
- 6) Failure: test_name_empty(AssistmentTest) [[./test/unit/assistment_test.rb:54](#)]: Assistment name is not defaulting to truncated main problem body. <"10 - This is a questio..."> expected but was <"Assistment #10">.
- 7) Failure: test_name_nil(AssistmentTest) [[./test/unit/assistment_test.rb:50](#)]: Assistment name is not defaulting to truncated main problem body. <"9 - This is a questio..."> expected but was <"Assistment #9">.
- 8) Failure: test_problem(AssistmentTest) [[./test/unit/assistment_test.rb:64](#)]: Assistment has no main problem. <nil> expected to not be nil.
- 9) Error: test_complete(ClassAssignmentTest): NoMethodError: undefined method `complete?' for #<ClassAssignment:0xb6a5b1c4>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in `method_missing' [./test/unit/class_assignment_test.rb:23](#):in `test_complete'
- 10) Error: test_in_progress(ClassAssignmentTest): NoMethodError: undefined method `in_progress?' for #<ClassAssignment:0xb6a50648>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in `method_missing' [./test/unit/class_assignment_test.rb:17](#):in `test_in_progress'
- 11) Error: test_not_started(ClassAssignmentTest): NoMethodError: undefined method `not_started?' for #<ClassAssignment:0xb6a437f4>

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' ./test/unit/class_assignment_test.rb:11:in `test_not_started'
```

```
12) Error: test_status(ClassAssignmentTest): NoMethodError: undefined method `status'
for #<ClassAssignment:0xb6a1ca64> /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/base.rb:1860:in `method_missing'
./test/unit/class_assignment_test.rb:29:in `test_status'
```

```
13) Error: test_truth(ClassReportCommentTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42P01 Mrelation "class_report_comments" does not exist
Fnamespace.c L221 RRangeVarGetRelid: DELETE FROM class_report_comments
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:162:in `update'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:47:in
`delete' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:285:in `delete_existing_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:256:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
14) Error: test_truth(ContentCreatorTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42P01 Mrelation "content_creators" does not exist
Fnamespace.c L221 RRangeVarGetRelid: DELETE FROM content_creators
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
```

```
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:162:in `update'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:47:in  
`delete' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:285:in `delete_existing_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in  
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:256:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in  
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in  
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
15) Error: test_truth(DbFilesTest): ActiveRecord::StatementInvalid: RuntimeError:  
ERROR C42703 Mcolumn "db_file_id" of relation "db_files" does not exist P29  
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO db_files ("id", "db_file_id",  
"data") VALUES (2, 2, 'data') /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
```

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
16) Error: test_action_hints(HintTest): ActiveRecord::StatementInvalid: RuntimeError:
ERROR C42703 Mcolumn "problem_id" of relation "hints" does not exist P20
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO hints ("problem_id", "id",
"value") VALUES (1, 2, 'This is another hint')
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
17) Error: test_acts_as_list(HintTest): ActiveRecord::StatementInvalid: RuntimeError:
ERROR C42703 Mcolumn "problem_id" of relation "hints" does not exist P20
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO hints ("problem_id", "id",
"value") VALUES (1, 2, 'This is another hint')
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
```

```
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in  
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
18) Error: test_problem(HintTest): ActiveRecord::StatementInvalid: RuntimeError:  
ERROR C42703 Mcolumn "problem_id" of relation "hints" does not exist P20  
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO hints ("problem_id", "id",  
"value") VALUES (1, 2, 'This is another hint')  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
```

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
19) Error: test_text(HintTest): ActiveRecord::StatementInvalid: RuntimeError: ERROR
C42703 Mcolumn "problem_id" of relation "hints" does not exist P20 Fparse_target.c
L767 RcheckInsertTargets: INSERT INTO hints ("problem_id", "id", "value") VALUES
(1, 2, 'This is another hint') /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
20) Error: test_action_problems(ProblemTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
```



```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
21) Error: test_acts_as_list(ProblemTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
```

```
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
22) Error: test_answers(ProblemTest): ActiveRecord::StatementInvalid: RuntimeError:  
ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist P32  
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name", "enabled",  
"problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in  
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
23) Error: test_assignment_answers(ProblemTest): ActiveRecord::StatementInvalid:  
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist  
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",  
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
```

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
24) Error: test_assistment(ProblemTest): ActiveRecord::StatementInvalid: RuntimeError:
ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist P32
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name", "enabled",
"problem_id", "id") VALUES ('Problem Types', 't', 9, 4)
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
```

```
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
25) Error: test_correct?(ProblemTest): ActiveRecord::StatementInvalid: RuntimeError:  
ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist P32  
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name", "enabled",  
"problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in  
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
26) Error: test_has_scaffolds?(ProblemTest): ActiveRecord::StatementInvalid:  
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist  
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",  
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
```

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
27) Error: test_hints(ProblemTest): ActiveRecord::StatementInvalid: RuntimeError:
ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist P32
Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name", "enabled",
"problem_id", "id") VALUES ('Problem Types', 't', 9, 4)
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
```

```
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
28) Error: test_is_part_of_scaffold?(ProblemTest): ActiveRecord::StatementInvalid:  
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist  
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",  
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in  
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
29) Error: test_parent_scaffold(ProblemTest): ActiveRecord::StatementInvalid:  
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist  
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",  
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
```

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
30) Error: test_problem_type(ProblemTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
```

```
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
31) Error: test_question_text(ProblemTest): ActiveRecord::StatementInvalid: RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist  
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name", "enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in  
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in  
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in  
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in  
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

```
32) Error: test_scaffold(ProblemTest): ActiveRecord::StatementInvalid: RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist  
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name", "enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
```



```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in
`insert_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in
`each' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

33) Error: test_algebra_correct(ProblemTypeTest): NoMethodError: undefined method
`correct?' for #<ProblemType:0xb6a64940>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:97](#):in `test_algebra_correct'

34) Error: test_algebra_incorrect(ProblemTypeTest): NoMethodError: undefined method
`correct?' for #<ProblemType:0xb6a5b7dc>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:103](#):in `test_algebra_incorrect'

35) Error: test_choose_1_correct(ProblemTypeTest): NoMethodError: undefined method
`correct?' for #<ProblemType:0xb6a4976c>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:20](#):in `test_choose_1_correct'

36) Error: test_choose_1_incorrect(ProblemTypeTest): NoMethodError: undefined
method `correct?' for #<ProblemType:0xb6a40734>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:32](#):in `test_choose_1_incorrect'

37) Error: test_choose_1_invalid_choice(ProblemTypeTest): NoMethodError: undefined
method `correct?' for #<ProblemType:0xb6a33c50>

/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:26](#):in `test_choose_1_invalid_choice'

38) Error: test_choose_n_all_incorrect(ProblemTypeTest): NoMethodError: undefined
method `correct?' for #<ProblemType:0xb6a23350>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:58](#):in `test_choose_n_all_incorrect'

39) Error: test_choose_n_correct(ProblemTypeTest): NoMethodError: undefined method
`correct?' for #<ProblemType:0xb6a15d68>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:39](#):in `test_choose_n_correct'

40) Error: test_choose_n_invalid_choice(ProblemTypeTest): NoMethodError: undefined
method `correct?' for #<ProblemType:0xb6a0b7f0>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:46](#):in `test_choose_n_invalid_choice'

41) Error: test_choose_n_one_incorrect(ProblemTypeTest): NoMethodError: undefined
method `correct?' for #<ProblemType:0xb6a02a88>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:52](#):in `test_choose_n_one_incorrect'

42) Failure: test_description(ProblemTypeTest) [[./test/unit/problem_type_test.rb:11](#)]:
ProblemType description is set, but description doesn't return it. <"Multiple Choice">
expected but was <"Multiple choice">.

43) Error: test_fill_in_correct(ProblemTypeTest): NoMethodError: undefined method
`correct?' for #<ProblemType:0xb69ebbe4>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:78](#):in `test_fill_in_correct'

44) Error: test_fill_in_expected_incorrect(ProblemTypeTest): NoMethodError: undefined
method `correct?' for #<ProblemType:0xb69e3728>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:90](#):in `test_fill_in_expected_incorrect'

45) Error: test_fill_in_incorrect(ProblemTypeTest): NoMethodError: undefined method
`correct?' for #<ProblemType:0xb69dc5e0>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' [./test/unit/problem_type_test.rb:84](#):in `test_fill_in_incorrect'

46) Error: test_rank_correct(ProblemTypeTest): NoMethodError: undefined method `correct?' for #<ProblemType:0xb69bd5dc>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in `method_missing' [./test/unit/problem_type_test.rb:65](#):in `test_rank_correct'

47) Error: test_rank_incorrect(ProblemTypeTest): NoMethodError: undefined method `correct?' for #<ProblemType:0xb69b0b0c>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in `method_missing' [./test/unit/problem_type_test.rb:71](#):in `test_rank_incorrect'

48) Error: test_CheckCRUD(ProgressTest): Errno::ENOENT: No such file or directory - /home/ryansv/.cruise/projects/Assistments/work/config/./test/fixtures/progresses
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:344:in `open' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:344:in `entries'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:344:in `read_fixture_files' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:281:in `initialize'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:251:in `new' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:251:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:250:in `map' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:250:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in `silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in `load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'

49) Error: test_truth(ReportGeneratorTest): ActiveRecord::StatementInvalid: RuntimeError: ERROR C42P01 Mrelation "report_generators" does not exist
Fnamespace.c L221 RRangeVarGetRelid: DELETE FROM report_generators
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:162:in `update'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:47:in

```
`delete' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:285:in `delete_existing_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

50) Error: test_truth(RightTest): ActiveRecord::StatementInvalid: RuntimeError:
ERROR C42P01 Mrelation "rights" does not exist Fnamespace.c L221

RRangeVarGetRelid: DELETE FROM rights

```
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:162:in `update'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:47:in
`delete' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:285:in `delete_existing_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
```

```
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

51) Failure: test_truth(RoleTest) [[./test/unit/role_test.rb:8](#)]: Flunked.

```
52) Error: test_relationships(ScaffoldTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42703 Mcolumn "enabled" of relation "scaffolds" does not exist
P32 Fparse_target.c L767 RcheckInsertTargets: INSERT INTO scaffolds ("name",
"enabled", "problem_id", "id") VALUES ('Problem Types', 't', 9, 4)
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:290:in `insert_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:289:in `insert_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:257:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in `transaction'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in `silence'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in `load_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:538:in `setup'
```

53) Failure: test_CAT_path_5r_5w(SectionTest) [[./test/unit/section_test.rb:40](#):in `follow_CAT_path' [./test/unit/section_test.rb:33](#):in `each' [./test/unit/section_test.rb:33](#):in `follow_CAT_path' [./test/unit/section_test.rb:47](#):in `test_CAT_path_5r_5w']: <false> is not true.

54) Failure: test_CAT_path_every_other(SectionTest) [[./test/unit/section_test.rb:40](#):in `follow_CAT_path' [./test/unit/section_test.rb:33](#):in `each' [./test/unit/section_test.rb:33](#):in `follow_CAT_path' [./test/unit/section_test.rb:72](#):in `test_CAT_path_every_other']: <false> is not true.

55) Failure: test_SequenceSequenceUsage(SequenceTest)
[./test/unit/sequence_test.rb:60]: <false> is not true.

56) Error: test_SimpleSingleSection(SequenceTest): ArgumentError: wrong number of arguments (1 for 2) ./test/unit/sequence_test.rb:19:in `nextProblem'
./test/unit/sequence_test.rb:19:in `test_SimpleSingleSection'

57) Error: test_truth(StudentReportCommentTest): ActiveRecord::StatementInvalid:
RuntimeError: ERROR C42P01 Mrelation "student_report_comments" does not exist
Fnamespace.c L221 RRangeVarGetRelid: DELETE FROM student_report_comments
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract_adapter.rb:128:in `log'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:152:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/postgresql_adapter.rb:162:in `update'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:47:in
`delete' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:285:in `delete_existing_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:256:in `each'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:256:in
`create_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:255:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:867:in
`silence' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:248:in `create_fixtures'
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/fixtures.rb:593:in
`load_fixtures' /usr/local/lib/ruby/gems/1.8/gems/activerecord-
1.15.3/lib/active_record/fixtures.rb:538:in `setup'

58) Error: test_belongs_to_class(StudentTest): NoMethodError: undefined method
`belongs_to_class?' for #<Student:0xb6c029a0>
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1857:in
`method_missing' ./test/unit/student_test.rb:68:in `test_belongs_to_class'

59) Failure: test_joinable_classes(StudentTest) [[./test/unit/student_test.rb:27](#)]: Students joinable classes are suppose to be the classes they are not in.

```
<[#<StudentClass:0xb6b6f6dc @attributes= {"report_students_min"=>"5",  
"name"=>"Mr. Woods' first class", "report_summary_hour"=>nil, "grade"=>"8",  
"weight_by_category"=>"f", "parameters"=>{}, "class_type_id"=>"1",  
"uncategorized_weight"=>"0", "enabled"=>"t", "id"=>"1", "created_at"=>nil}>,  
#<StudentClass:0xb6b6f68c @attributes= {"report_students_min"=>"5", "name"=>"Mr.  
Woods' second class", "report_summary_hour"=>nil, "grade"=>"8",  
"weight_by_category"=>"f", "parameters"=>{}, "class_type_id"=>"1",  
"uncategorized_weight"=>"0", "enabled"=>"t", "id"=>"2", "created_at"=>nil}>,  
#<StudentClass:0xb6b6f600 @attributes= {"report_students_min"=>"5", "name"=>"Ms.  
Forrest' second class", "report_summary_hour"=>nil, "grade"=>"6",  
"weight_by_category"=>"f", "parameters"=>{}, "class_type_id"=>"1",  
"uncategorized_weight"=>"0", "enabled"=>"t", "id"=>"4", "created_at"=>nil}>,  
#<StudentClass:0xb6b68558 @attributes= {"report_students_min"=>"5", "name"=>"Mr.  
Black's Period One Class", "report_summary_hour"=>nil, "grade"=>"10",  
"weight_by_category"=>"f", "parameters"=>{}, "class_type_id"=>"1",  
"uncategorized_weight"=>"0", "enabled"=>"t", "id"=>"5", "created_at"=>nil}>,  
#<StudentClass:0xb6b68530 @attributes= {"report_students_min"=>"5", "name"=>"Ms.  
Forrest' first class", "report_summary_hour"=>nil, "grade"=>"6",  
"weight_by_category"=>"f", "parameters"=>{}, "class_type_id"=>"1",  
"uncategorized_weight"=>"0", "enabled"=>"t", "id"=>"3", "created_at"=>nil}>,  
#<StudentClass:0xb6b5c7f8 @attributes= {"report_students_min"=>"5", "name"=>"Mr.  
Black's Period Two Class", "report_summary_hour"=>nil, "grade"=>"10",  
"weight_by_category"=>"f", "parameters"=>{}, "class_type_id"=>"1",  
"uncategorized_weight"=>"0", "enabled"=>"t", "id"=>"6", "created_at"=>nil}>]>  
expected but was <[]>.
```

60) Error: test_students(TeacherTest): NoMethodError: undefined method `students' for #<Teacher:0xb6855514> /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:1860:in `method_missing' [./test/unit/teacher_test.rb:63](#):in `test_students' [./test/unit/teacher_test.rb:62](#):in `each' [./test/unit/teacher_test.rb:62](#):in `test_students'

61) Error: test_require_email(UserTest): NoMethodError: You have a nil object when you didn't expect it! The error occurred while evaluating nil.downcase [/home/ryansv/.cruise/projects/Assistments/work/config/./app/models/user.rb:223](#):in `update_login' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/callbacks.rb:333:in `send' /usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/callbacks.rb:333:in `callback' /usr/local/lib/ruby/gems/1.8/gems/activerecord-


```
1.15.3/lib/active_record/callbacks.rb:330:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:330:in `callback'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:296:in `valid?'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/validations.rb:751:in `save_without_transactions'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:129:in `save'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:95:in `transaction'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:121:in `transaction'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:129:in `save'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:451:in  
`create' ./test/unit/user_test.rb:131:in `create_user' ./test/unit/user_test.rb:60:in  
`test_require_email'  
/home/ryansv/.cruise/projects/Assistments/work/config/./lib/authentication/authenticated  
test\_helper.rb:38:in `assert\_difference'  
/home/ryansv/.cruise/projects/Assistments/work/config/./lib/authentication/authenticated  
test\_helper.rb:43:in `assert\_no\_difference' ./test/unit/user\_test.rb:59:in  
`test_require_email'
```

62) Error: test_require_login(UserTest): NoMethodError: You have a nil object when you didn't expect it! The error occurred while evaluating nil.downcase

```
/home/ryansv/.cruise/projects/Assistments/work/config/./app/models/user.rb:223:in  
`update_login' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:333:in `send'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:333:in `callback'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:330:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:330:in `callback'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/callbacks.rb:296:in `valid?'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/validations.rb:751:in `save_without_transactions'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-
```



```
1.15.3/lib/active_record/transactions.rb:129:in `save'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/connection_adapters/abstract/database_statements.rb:59:in  
`transaction' /usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:95:in `transaction'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:121:in `transaction'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-  
1.15.3/lib/active_record/transactions.rb:129:in `save'  
/usr/local/lib/ruby/gems/1.8/gems/activerecord-1.15.3/lib/active_record/base.rb:451:in  
`create' ./test/unit/user_test.rb:131:in `create_user' ./test/unit/user_test.rb:23:in  
`test_require_login'  
/home/ryansv/.cruise/projects/Assistments/work/config/./lib/authentication/authenticated  
test\_helper.rb:38:in `assert_difference'  
/home/ryansv/.cruise/projects/Assistments/work/config/./lib/authentication/authenticated  
test\_helper.rb:43:in `assert_no_difference' ./test/unit/user_test.rb:22:in  
`test_require_login'
```

```
63) Failure: test_unique_login(UserTest) [./test/unit/user\_test.rb:39:in `test_unique_login'  
/home/ryansv/.cruise/projects/Assistments/work/config/./lib/authentication/authenticated  
test\_helper.rb:38:in `assert_difference'  
/home/ryansv/.cruise/projects/Assistments/work/config/./lib/authentication/authenticated  
test\_helper.rb:43:in `assert_no_difference' ./test/unit/user\_test.rb:37:in  
`test_unique_login']: <nil> is not true.
```

171 tests, 264706 assertions, 15 failures, 48 errors

```
/usr/local/bin/ruby -I"/home/ryansv/.cruise/projects/Assistments/work/lib" -  
I"/home/ryansv/.cruise/projects/Assistments/work/test"  
"/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader.rb"  
"test/functional/admin/district_controller_test.rb"  
"test/functional/admin/teacher_controller_test.rb"  
"test/functional/admin/rights_role_controller_test.rb"  
"test/functional/admin/action_controller_test.rb"  
"test/functional/admin/account_controller_test.rb"  
"test/functional/admin/assignment_answer_controller_test.rb"  
"test/functional/admin/right_controller_test.rb"  
"test/functional/admin/action_hint_controller_test.rb"  
"test/functional/admin/role_controller_test.rb"  
"test/functional/admin/school_controller_test.rb"  
"test/functional/admin/about_controller_test.rb"  
"test/functional/admin/student_controller_test.rb"
```

"test/functional/admin/action_problem_controller_test.rb"
"test/functional/admin/user_role_controller_test.rb"
"test/functional/admin/user_detail_controller_test.rb"
"test/functional/teacher/print_controller_test.rb"
"test/functional/teacher/multiple_class_controller_test.rb"
"test/functional/teacher/student_report_controller_test.rb"
"test/functional/teacher/student_class_controller_test.rb"
"test/functional/teacher/homework_controller_test.rb"
"test/functional/teacher/assignment_controller_test.rb"
"test/functional/teacher/open_response_controller_test.rb"
"test/functional/teacher/preview_controller_test.rb"
"test/functional/teacher/teacher_class_controller_test.rb"
"test/functional/teacher/enrollment_controller_test.rb"
"test/functional/account_controller_test.rb" "test/functional/stats_controller_test.rb"
"test/functional/build/scaffold_controller_test.rb"
"test/functional/build/sequence_controller_test.rb"
"test/functional/build/problem_controller_test.rb"
"test/functional/build/tag_controller_test.rb"
"test/functional/build/section_controller_test.rb"
"test/functional/build/hint_controller_test.rb"
"test/functional/build/strategy_controller_test.rb"
"test/functional/build/assistment_section_controller_test.rb"
"test/functional/build/image_controller_test.rb"
"test/functional/build/assistment_controller_test.rb"
"test/functional/build/answer_controller_test.rb"
"test/functional/build/preview_controller_test.rb"
"test/functional/homework_controller_test.rb"
"test/functional/tutor/class_assignment_controller_test.rb"
"test/functional/tutor/student_class_controller_test.rb"
"test/functional/tutor/class_homework_controller_test.rb"
"test/functional/about_controller_test.rb" "test/functional/graph_controller_test.rb"
"test/functional/home_controller_test.rb"

[/home/ryansv/.cruise/projects/Assistments/work/config/boot.rb:25](#): Warning:

Gem::SourceIndex#search support for String patterns is deprecated
/usr/local/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:31:in
`gem_original_require': no such file to load -- admin/teacher_controller
(MissingSourceFile) from
/usr/local/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:31:in `require' from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-
1.4.2/lib/active_support/dependencies.rb:495:in `require' from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-

```
1.4.2/lib/active_support/dependencies.rb:342:in `new_constants_in' from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-
1.4.2/lib/active_support/dependencies.rb:495:in `require' from
./test/functional/admin/teacher_controller_test.rb:2 from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-
1.4.2/lib/active_support/dependencies.rb:488:in `load' from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-
1.4.2/lib/active_support/dependencies.rb:488:in `load' from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-
1.4.2/lib/active_support/dependencies.rb:342:in `new_constants_in' from
/usr/local/lib/ruby/gems/1.8/gems/activesupport-
1.4.2/lib/active_support/dependencies.rb:488:in `load' from
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader.rb:5 from
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader.rb:5:in `each' from
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader.rb:5
/usr/local/bin/ruby -I"/home/ryansv/.cruise/projects/Assistments/work/lib" -
I"/home/ryansv/.cruise/projects/Assistments/work/test"
"/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/rake_test_loader.rb" Command
failed with status (1): [/usr/local/bin/ruby -I"/home/ryansv/.cruis...]
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:969:in `sh'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:982:in `call'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:982:in `sh'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:1070:in `sh'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:1005:in `ruby'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:1070:in `ruby'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/testtask.rb:117:in `define'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:1088:in `verbose'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake/testtask.rb:102:in `define'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:617:in `call'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:617:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:612:in `each'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:612:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:578:in `invoke_with_call_chain'
/usr/local/lib/ruby/1.8/monitor.rb:238:in `synchronize'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:571:in `invoke_with_call_chain'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:564:in `invoke'
/home/ryansv/.gem/ruby/1.8/gems/rails-1.2.3/lib/tasks/testing.rake:45
/home/ryansv/.gem/ruby/1.8/gems/rails-1.2.3/lib/tasks/testing.rake:43:in `collect'
/home/ryansv/.gem/ruby/1.8/gems/rails-1.2.3/lib/tasks/testing.rake:43
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:617:in `call'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:617:in `execute'
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:612:in `each'
```

```
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:612:in `execute'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:578:in `invoke_with_call_chain'  
/usr/local/lib/ruby/1.8/monitor.rb:238:in `synchronize'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:571:in `invoke_with_call_chain'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:564:in `invoke'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:2027:in `invoke_task'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:2005:in `top_level'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:2005:in `each'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:2005:in `top_level'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:2044:in  
`standard_exception_handling' /usr/local/lib/ruby/gems/1.8/gems/rake-  
0.8.4/lib/rake.rb:1999:in `top_level' /usr/local/lib/ruby/gems/1.8/gems/rake-  
0.8.4/lib/rake.rb:1977:in `run' /usr/local/lib/ruby/gems/1.8/gems/rake-  
0.8.4/lib/rake.rb:2044:in `standard_exception_handling'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/lib/rake.rb:1974:in `run'  
/usr/local/lib/ruby/gems/1.8/gems/rake-0.8.4/bin/rake:31 /usr/local/bin/rake:19:in `load'  
/usr/local/bin/rake:19 rake aborted! Test failures (See full trace by running task with --  
trace)
```

[./script/./config/boot.rb:25](#):Warning: Gem::SourceIndex#search support for String patterns is deprecated **

```
Starting Mongrel listening at 0.0.0.0:3000 /usr/local/lib/ruby/gems/1.8/gems/mongrel-  
1.0.1/lib/mongrel/tcpback.rb:12:in `initialize_without_backlog': Address already in use -  
bind(2) (Errno::EADDRINUSE) from /usr/local/lib/ruby/gems/1.8/gems/mongrel-  
1.0.1/lib/mongrel/tcpback.rb:12:in `initialize' from  
/usr/local/lib/ruby/gems/1.8/gems/mongrel-1.0.1/lib/mongrel.rb:560:in `new' from  
/usr/local/lib/ruby/gems/1.8/gems/mongrel-1.0.1/lib/mongrel.rb:560:in `initialize' from  
/usr/local/lib/ruby/gems/1.8/gems/mongrel-1.0.1/lib/mongrel/configurator.rb:128:in `new'  
from /usr/local/lib/ruby/gems/1.8/gems/mongrel-1.0.1/lib/mongrel/configurator.rb:128:in  
`listener' from /usr/local/lib/ruby/gems/1.8/gems/mongrel-1.0.1/bin/mongrel_rails:98:in  
`cloaker_' from /usr/local/lib/ruby/gems/1.8/gems/mongrel-  
1.0.1/lib/mongrel/configurator.rb:51:in `call' from  
/usr/local/lib/ruby/gems/1.8/gems/mongrel-1.0.1/lib/mongrel/configurator.rb:51:in  
`initialize' ... 14 levels... from /home/ryansv/.gem/ruby/1.8/gems/rails-  
1.2.3/lib/commands/server.rb:39 from  
/usr/local/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:31:in  
`gem_original_require' from  
/usr/local/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:31:in `require' from  
script/server:3 => Booting Mongrel (use 'script/server webrick' to force WEBrick) =>  
Rails application starting on http://0.0.0.0:3000 => Call with -d to detach => Ctrl-C to  
shutdown server Exiting
```

References

Martin Fowler, “Continuous Integration”. May 1st, 2006. Retrieved April 29, 2009 from

<<http://martinfowler.com/articles/continuousIntegration.html>>

Martin Fowler, “Continuous Integration (original version)”. Retrieved April 29, 2009

from <<http://martinfowler.com/articles/originalContinuousIntegration.html>>

“CruiseControl.rb”. 2007. ThoughtWorks. Retrieved April 29, 2009 from

<<http://cruisecontrolrb.thoughtworks.com/>>

“CruiseControl.rb”. 2007. ThoughtWorks. Retrieved April 29, 2009 from

<http://cruisecontrolrb.thoughtworks.com/documentation/getting_started>

“Selenium web application testing system”. OpenQA. Retrieved April 29, 2009 from

<<http://seleniumhq.org/>>

“Assistment Overview – Assistment TeacherWiki” April 2nd, 2009. Retrieved April 29,

2009 from

<http://teacherwiki.assistment.org/index.php?title=Assistment_Overview&oldid=4091>

Duvall, Paul, Steve Matyas, and Andrew Glover. *Continuous Integration: Improving*

Software Quality and Reducing Risk. 1st ed. Addison-Wesley Professional , 2007.

Print.