

# Agile GUI Testing software

## devdaily.com

### Version

This document was released on May 29, 2010.

### Introduction

Our Agile GUI Testing (AGT) software lets you write scripts to perform GUI functional regression tests on your software applications.

### Goals

When creating the AGT software, we had several goals in mind:

1. Create a GUI testing framework that works on many platforms.
2. Make GUI testing fast, as in fast to create, and fast to update.
3. Create a GUI “Recorder”, and a separate GUI “Player”.
4. Develop an easy but powerful scripting language.
5. Do whatever is needed so tests aren’t based on “timings”, as timings are prone to errors.
6. Have the ability to run as many unattended tests as desired in succession.

We have done well with most of these goals, and we’re working on the others.

As one important note, the idea of a “Recorder” sounded like just what we wanted when we began developing the software, but as we learned what we really wanted, we learned that a Recorder is really just best as a training tool, so there isn’t much development happening with that right now.

### Prerequisites

AGT scripts are written in the Ruby programming language, and specifically require the JRuby interpreter, which also gives you access to the Java programming language.

To run AGT scripts you will need JRuby and Java installed. Both of these programming environments are easy to install. You can find more information at these websites:

- [www.jruby.org](http://www.jruby.org)
- [www.java.com](http://www.java.com)

## A sample script

The best way to learn about the AGT software is to see some sample scripts, and then begin adopting those scripts for your own needs.

Here's the source code for a simple AGT script that brings the Google Chrome browser to the foreground on a Mac OS X system, and then opens the [devdaily.com](http://devdaily.com) URL in the browser:

```
# three boilerplate/required lines
require 'java'
require 'AgileGuiTesting'
require 'AGTMacOsX'

comment 'bring chrome to the foreground'
foreground 'Google Chrome'
wait 2000

comment 'get the mouse out of the way'
move_mouse 30, 200
wait 250

comment 'put focus in the url field'
apple VK_L

comment 'go to the devdaily website'
type 'www.devdaily.com \n'
```

As you can see, because we use Ruby/JRuby as our scripting language, a script like this can be very clean, and also compact. Possibly the only unusual line in that script is this one:

```
apple VK_L
```

which means, press the [Apple] key (also known as the [Command] key) and the letter 'L'. It can be written more simple like this:

```
apple 'l'
```

but it's easy to confuse that lowercase letter 'L' for the number one (1), so for a simple demo, VK\_L is a little less confusing. As you can see, either syntax can be used.

## Installation

Installing AGT is very simple. Once you have JRuby and Java installed and working on your system, just download the last version of AGT, which will contain a “scripts” folder.

The current method for using AGT is to simply create your own scripts in that AGT folder. This approach will probably change in the future to make upgrading easier, but for now, this is a very easy way to make sure everything you need can be found by the JRuby interpreter.

Optionally, you can put our “scripts” folder in a subdirectory of your tests, and then change our “require” statements to look like this:

```
require 'java'  
require 'scripts/AgileGuiTesting'  
require 'scripts/AGTMacOsX'
```

I haven't tried that yet, but I think it will work.

## Comments

In that example, you also saw the “comment” command shown, like this:

```
comment 'bring chrome to the foreground'
```

There are two important notes to share about comments. First, you can shorten that line, like this:

```
c 'bring chrome to the foreground'
```

Second -- and this is very important -- comments are saved in a stack, so if your test fails, the comments will be unrolled and printed out when the test fails. In this manner, you can see exactly where your test failed.

So, whenever you're tempted to put a comment in your code like this (which is the normal Ruby comment syntax):

```
# bring chrome to the foreground
```

use our comment syntax instead. In this way, you'll be rewarded with a much easier debugging process if/when your test fails.

## Common commands (“methods”)

The AGT programming language is still evolving at this time, but at present, these are the most important and commonly-used commands in our language:

Command	Description
click (x, y)	Perform a single-click at the given x/y coordinates. (This is a left-click.)
doubleclick (x, y)	Perform a double-click at the given x/y coordinates. (Again, this is a left-click.)
comment (text)	Add a comment. Comments are printed when a test fails.
type (text)	Type a text string, such as <a href="http://www.devdaily.com">www.devdaily.com</a> \n  Note that you can use backslash characters, like \n for newline, and \t for tab.
move_mouse (x, y)	Move the mouse cursor to the given (x,y) coordinate.
wait (time in ms)	Pause the system for the given number of milliseconds.
wait_for_xycolor (x, y, r, g, b)	Wait for the RGB color at the given pixel (x,y) to appear. (See the docs for more details, including the “timeout” field.)
wait_for_xycolor_to_go_away (x, y, r, g, b)	Wait for the RGB color at the given pixel (x,y) to go away. (See the docs for more details, including the “timeout” field.)
(x,y) = find_image_click_coordinates (image_file_name)	This method finds the (x,y) coordinates of the given image on screen. For instance, it can find the Acrobat logo if Adobe Acrobat is currently displayed on screen.
apple	Invoke the [Apple] or [Command] key on a Mac OS X system
ctrl	Invoke the [Control] key

Command	Description
tab	A convenience method. Use this as a shortcut when you want to type the [Tab] character.

Beyond these, there are several other commands currently available, so please see our included HTML documentation for information on all the available commands.

### Reserved keywords

Our vocabulary is also going to grow over time, so please consider the following keywords to be reserved (meaning that you should not create your own methods with these names):

alt	Press the [Alt] key
command	An alias for the “apple” method
delete	A convenience method for pressing the [Delete] key
drag	Drag something with the mouse
drag_over	Might be used to move the mouse in a drag and drop operation
drop	Drop something you were dragging
hide	Hide the current window
maximize	Maximize the current window
minimize	Minimize the current window
rightclick	Right-click at a given x/y location
shift	Press the [Shift] key
wait_for_image	Wait for a given image to appear on screen

Note that a few of these method names come from looking at the documentation for the Abbot Robot class (<http://abbot.sourceforge.net/doc/api/abbot/tester/Robot.html>).

Our available methods (our API) will probably grow beyond these methods, but these are the ones we see will be needed in the near future.

## Demo/training videos

We currently have one video online which demonstrates the use of our AGT software:

- Using AGT to test the Google Chrome browser  
<http://www.youtube.com/watch?v=pfhLdc64cek>

(more documentation to come ...)

### Sections Needed

- More example scripts
- Writing your own methods (commands)
- Known problems