# Assignment #13
Regression Testing

In this assignment, we'll try out the Rappture Regression Tester on our Spirograph tool.

> *NOTE:* The Regression Tester is still a work in progress. It doesn't yet support all of the data types in Rappture. Anything that you put into the tool.xml file by hand—the loader, meshes, fields, etc.—won't be understood by the tester. Hopefully that will get fixed soon. But instead of starting with the very latest version of the Spirograph tool, we'll start with our first version and create a simple test suite for it.

1) Start with your solution from Assignment #5, the simple Spirograph tool:
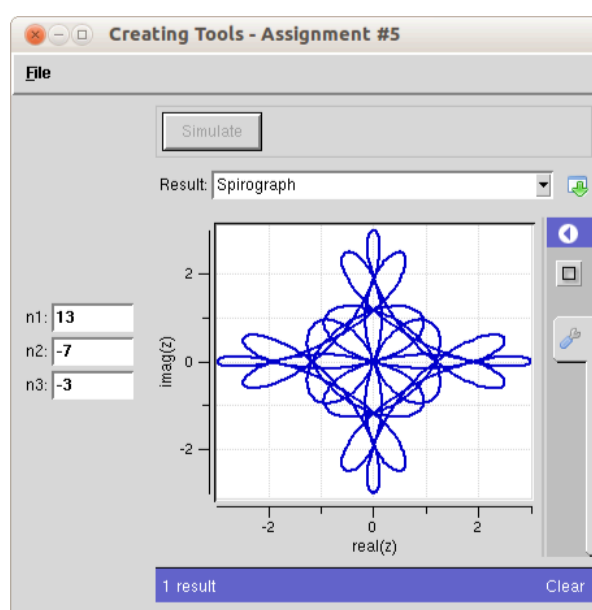
2) Run through a series of test cases to build the "golden results" for your test suite. This is easier if you clean out your "results" directory first, then run through the test cases, so the only run files in the results directory are the ones you want for the test suite:

```
cd ~/data/results/$SESSION
rm run*.xml
cd -
```

Note that the last command (`cd -`) is a handy way of getting back to the directory where you started, which is probably your tool directory.

Now, bring up the tool and run through the following cases:

- Fancy cross ....n1=13, n2=-7, n3=-3
- Flower ....n1=19, n2=-13, n3=3
- Palm branch..........n1=7, n2=-5, n3=2

3) Create a folder called "tests" in your tool directory (where the tool.xml file sits), and move the newly created run files into that directory. If you're sitting in your tool directory, you can type something like this:

```
mkdir tests
cp ~/data/results/$SESSION/run*.xml tests
```

4) Edit each of the test cases and add a `test` section at the top with a label and a description.

5) Bring up the Rappture Regression Tester and run through all of the cases:

```
rappture —tester
```

Once the tester is up, you can press *Select All* and then *Run*. It should run through all three tests and report that they all passed. After all, the tool probably hasn't changed in the past few minutes!

6) Edit one of the XML files in the tests directory to introduce a difference. For example, find the `output.component.xy` section and delete 10-15 lines of numbers, and remove the `description` field from one of the input numbers. Save your changes.

7) Select the test case that you modified and press *Run*. The tester should run just that one case, and it should report the differences that you introduced. Hover over each of the differences and press the *View* button to see more details.

8) With the corrupted test case still selected, press the *New Golden Standard* button below all of the warnings. This will overwrite the corrupted test with the latest results, so that the tests will run cleanly again.

Lessons Learned:

- The regression tester is not yet complete, but very useful for checking results from simple tools.

- When you spot a difference in a test case, it may be a "bug" (something introduced accidentally) or a "feature" (an improvement since the last version). If it's a bug, you should fix the bug and run the test suite again. If it's a feature, you can use the *New Golden Standard* button to update your test suite with the latest (correct) results.