Assignment #3
See What's Under the Hood

Congratulations! You've inherited a tool that almost works. Your job is to figure out how to run it, and patch it up to work properly with Rappture.

Get into the workspace, and then upload and unpack the file ex3.tgz:

```
importfile ex3.tgz
tar xvzf ex3.tgz
cd ex3
```

Try out the broken tool with Rappture:

```
rappture
```

The interface should come up okay, but when you press the *Simulate* button, it will return an error. Note that whenever a tool fails, it leaves a driver file behind.

Try running the tool by hand using the driver file. Your driver will have some unique number, but the command you type will be something like this:

```
tclsh main.tcl driver30711.xml
```

The tool is written in a language called Tcl. On the line above, you are calling `tclsh` to invoke the Tcl interpreter, passing in the name of the `main.tcl` script as the first argument, and passing your driver file as the next argument after that. It should respond by producing a Rappture "run" file, telling you the name of that file:

```
=RAPPTURE-RUN=>run1400781087196000.xml
```

If you got that far, the tool is working. The output is sitting in the run file. You can see the output by loading the run file:

```
rappture —load run1400781087196000.xml
```

This should bring up the tool showing the input numbers that you used as well as the correct output value.

Try creating your own driver file by hand. First, make a copy of the tool.xml file:

```
cp tool.xml driver.xml
```

Next, use your favorite editor to modify the new driver file. Find each line with a `<default>` parameter value and copy/paste to define a `<current>` value, like this:

```
...
<integer id="v1">
  <about>
    <label>Add this</label>
    <description>First value (v1) in the "v1 + v2" computation
performed by this tool.</description>
  </about>
  <default>0</default>
  <current>12</current>
</integer>
...
```

There are three inputs for this tool, so you should have to do this three times and define `<current>` values in three spots. If you've done everything correctly, you should be able to produce another run file:

```
tclsh main.tcl driver.xml
=RAPPTURE-RUN=>run1400781089364000.xml
rapture —load run1400781089364000.xml
```

So why doesn't the tool work? Edit the tool.xml file, and look closely at the `<command>` section. It's empty! There is no command template telling Rappture how to run your tool. If you look closely, that's exactly what the error message says. You've run the code by hand, so you know how to do it. The command section should contain something like this:

```
tclsh main.tcl driver.xml
```

Of course, it can't be exactly that. The driver files are not typically named driver.xml. They usually have a funny serial number as part of the name. Look back at the notes and find the correct substitution syntax for both the tool directory name and the driver file name. Insert the proper command string into the `<command>` ... `</command>` section, save the updated tool.xml file, and test out the tool. If you can get it to work properly when you press the *Simulate* button, you're done!

Lessons Learned:

- When a tool fails, Rappture leaves the driver file around in case you want to test things by hand.

- You can create your own driver file by making a copy of the tool.xml file, and by substituting your own `<current>` values for each input.

- Whenever the user presses *Simulate*, Rappture runs whatever command is specified in the `<command>` section of the tool.xml file. Some parts of the command string, such as the tool directory and the name of the driver file, are represented by a special syntax, and the values are substituted just before the command is executed.