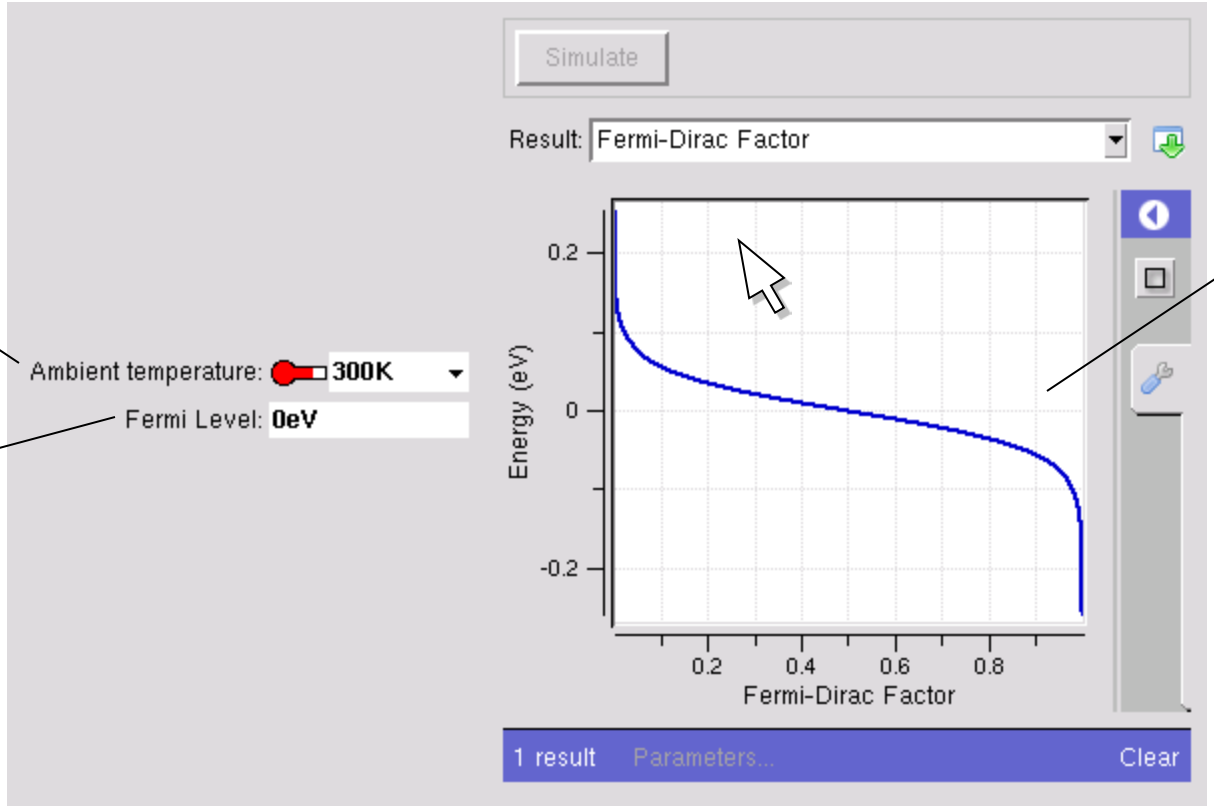
What's Under the Hood?

Michael McLennan

HUBzero® Platform for Scientific Collaboration

Purdue University

Example: Fermi function tool



number

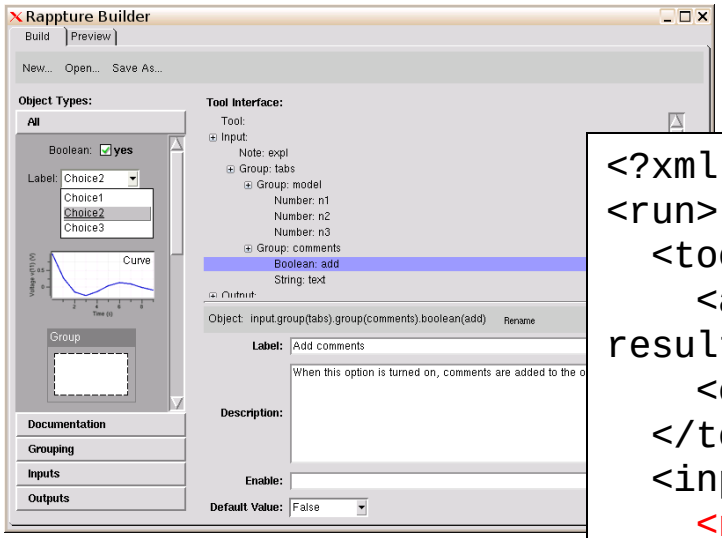
number

curve

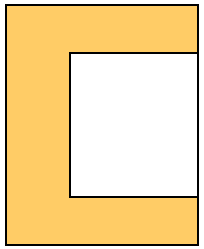
How does it work?

Tool Definition File

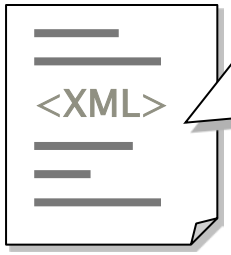
Builder




```
<?xml version="1.0"?>
<run>
  <tool>
    <about>Press Simulate to view
results.</about>
    <command>@tool/fermi @driver</command>
  </tool>
  <input>
    <number id="temperature">…</number>
    <number id="Ef">…</number>
  </input>
  <output>
    <curve id="f12">…</curve>
  </output>
</run>
```



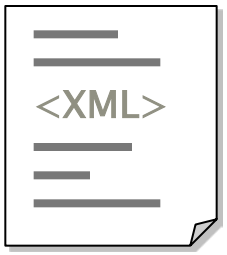
skeleton program



tool.xml

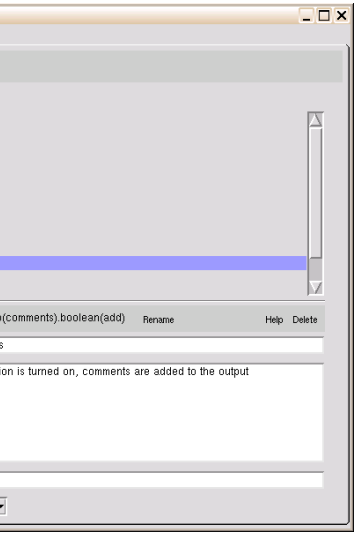
- Info about the tool, how to run it
- Input objects
- Output objects

Running produces a driver file

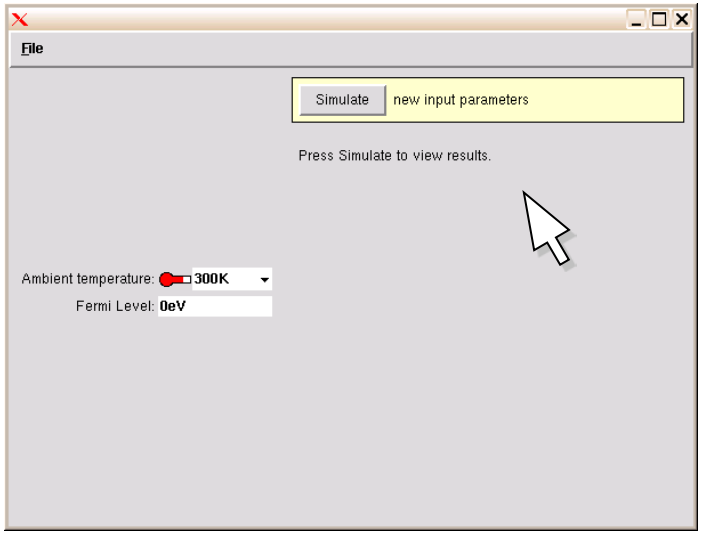


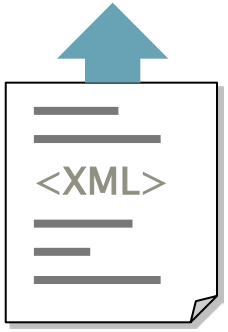
driver1827.xml

Builder



Runner





tool.xml

Running produces a driver file

```
<?xml version="1.0"?>
<run>
  …
  <input>
    <number id="temperature“>
      <units>K</units>
      <min>0K</min>
      <max>500K</max>
      <default>300K</default>

    </number>
    <number id="Ef“>
      <units>eV</units>
      <default>0eV</default>

    </number>
  </input>
  <output>
    <curve id="f12">…</curve>
  </output>
</run>
```

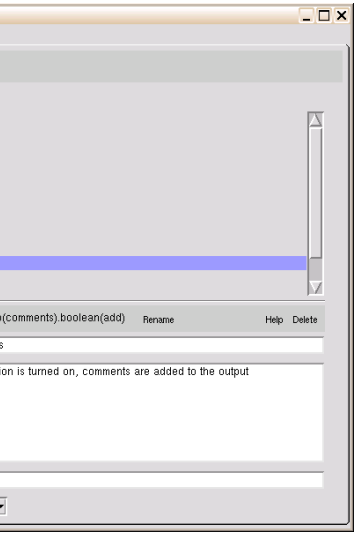
tool.xml

```
<?xml version="1.0"?>
<run>
  …
  <input>
    <number id="temperature“>
      <units>K</units>
      <min>0K</min>
      <max>500K</max>
      <default>300K</default>
      <current>77K</current>
    </number>
    <number id="Ef“>
      <units>eV</units>
      <default>0eV</default>
      <current>200meV</current>
    </number>
  </input>
  <output>
    <curve id="f12">…</curve>
  </output>
</run>
```

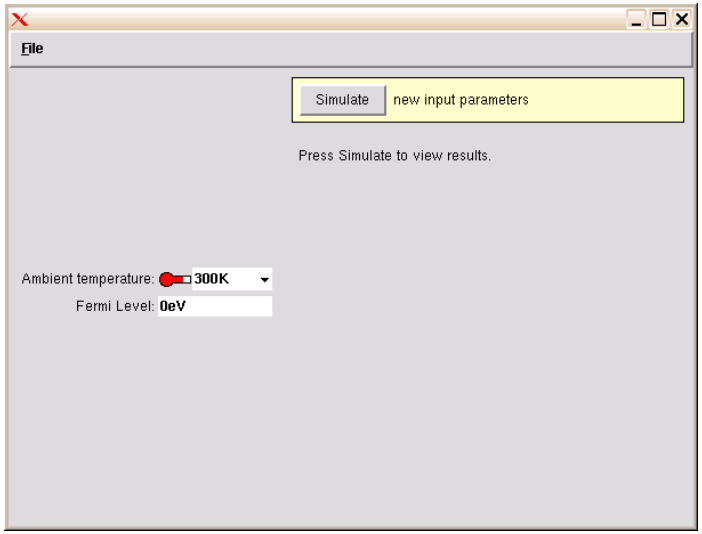
driver1827.xml

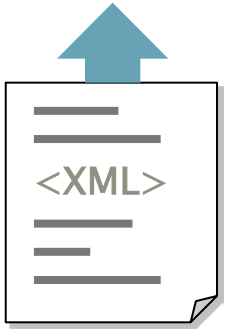
Running your program

Builder

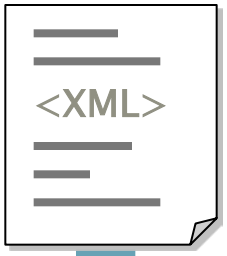


Runner

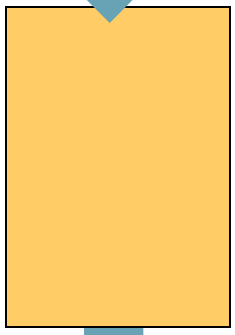




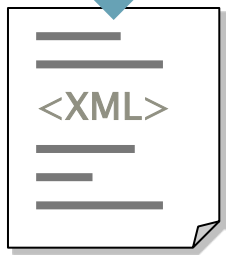
tool.xml



driver1827.xml



Your Program



run12703129102.xml

Program produces a run file

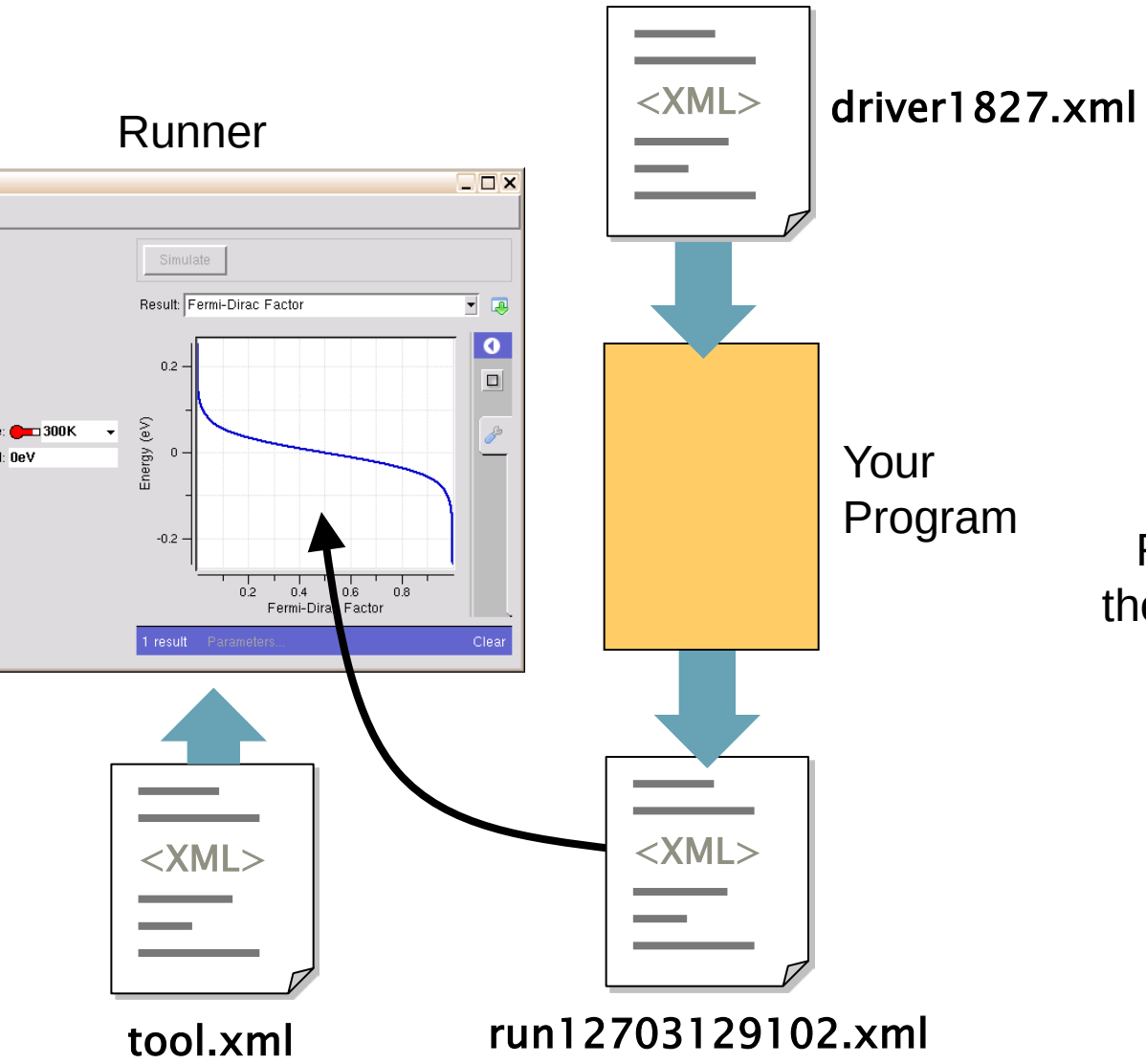
```
<?xml version="1.0"?>
<run>
  …
  <input>
    <number id="temperature“>…
      <current>77K</current>
    </number>
    <number id="Ef“>…
      <current>200meV</current>
    </number>
  </input>
  <output>
    <curve id="f12“>
      <about>…




    </curve>
  </output>
</run>
```

driver1827.xml

```
<?xml version="1.0"?>
<run>
  …
  <input>
    <number id="temperature“>…
      <current>77K</current>
    </number>
    <number id="Ef“>…
      <current>200meV</current>
    </number>
  </input>
  <output>
    <curve id="f12“>
      <about>…
      <component>
        <xy>0.999955 -0.25852
0.99995 -0.255935
0.999945 -0.25335…
    </curve>
  </output>
</run>
```

run12703129102.xml

Tool Definition File

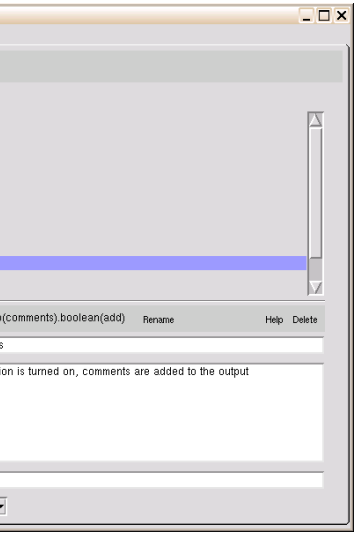


Results are moved out of the current working directory to the “results” directory

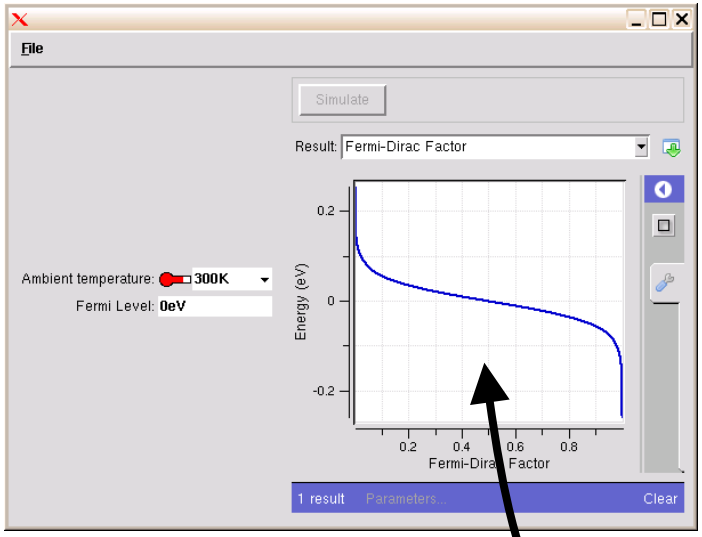
`~/data/results/$SESSION`

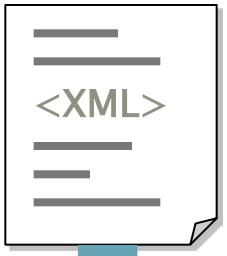
All Together

Builder

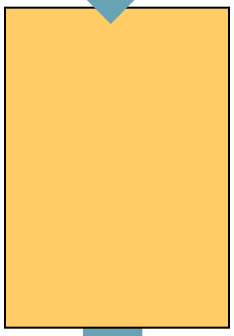


Runner

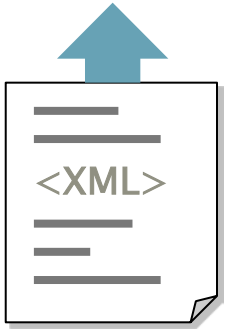




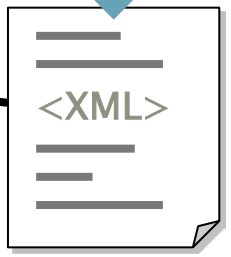
driver1827.xml



Your Program

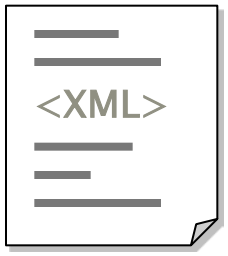


tool.xml



run12703129102.xml

Run file is a complete record of the run



run12703129102.xml

	Tool Defn	Input Vals	Output Vals
tool.xml	☑		
driver.xml	☑	☑	
run.xml	☑	☑	☑

Color xterm

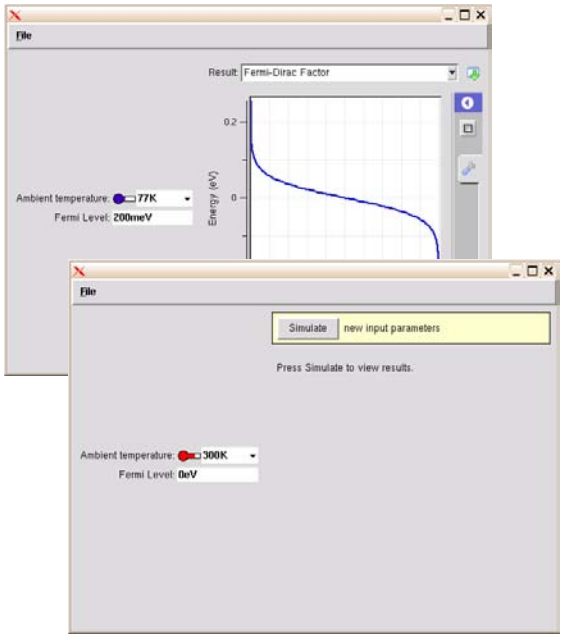
```
$ rappture -load run1301535652532513.xml
```

reload old results

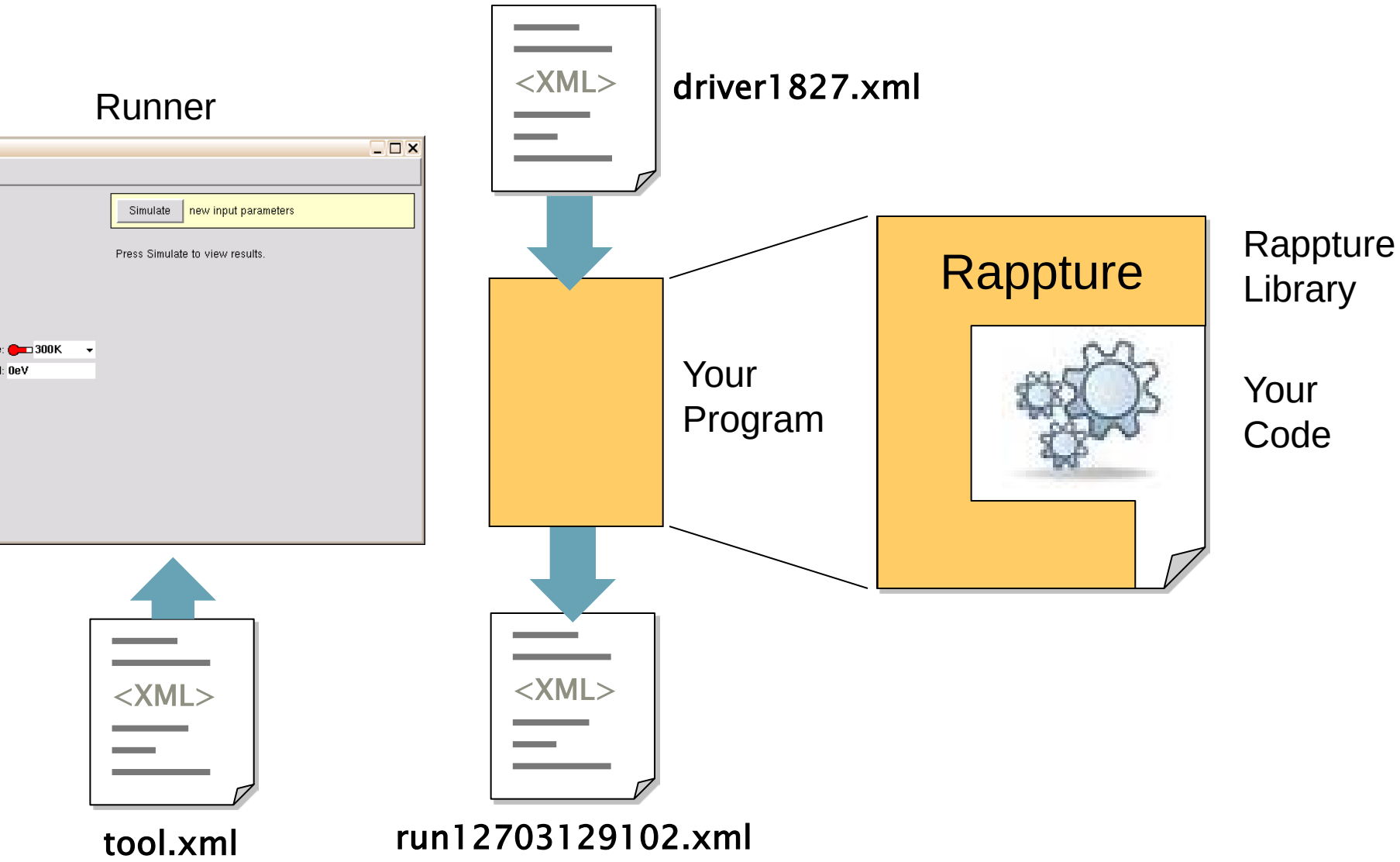
Color xterm

```
$ rappture -tool run1301535652532513.xml
```

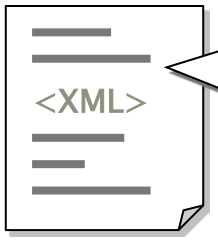
rerun an old tool



How does your program get invoked?



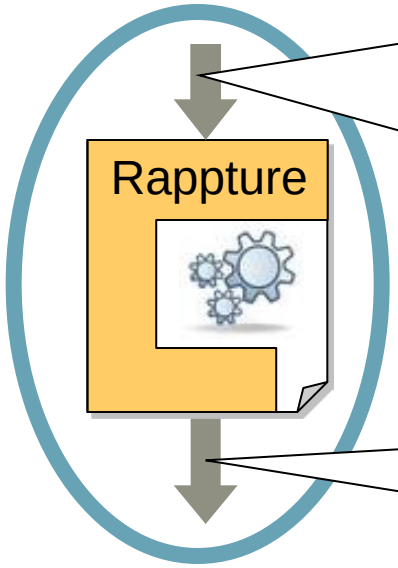
How your program gets invoked



tool.xml

```
<?xml version="1.0"?>
<run>
  <tool>
    <about>Press Simulate to view results.</about>
    <command>python @tool/fermi.py @driver</command>
  </tool>
  <input>
    …
```

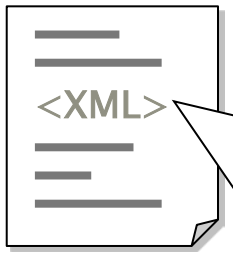
```
python /apps/yourtool/current/fermi.py driver327.xml
```




```
import Rappture
import sys
from math import *

io = Rappture.library(sys.argv[1])

Tstr = io.get('input.number(temperature).current')
Efstr = io.get('input.number(Ef).current')
…
```

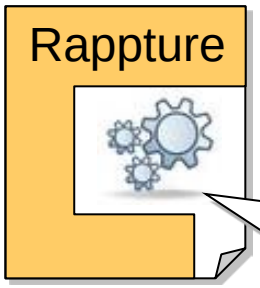
```
io.put('output.curve(f12).component.xy', xydata)

Rappture.result(io)
sys.exit()
```

Object names in tool.xml and in your program



tool.xml

```
<?xml version="1.0"?>
<run>
  …
  <input>                                  | input
    <number id="temperature">              |   .number(temperature)
        <current>300K</current>            |     .current
    </number>
    …
  </input>

  <output>                                 | output
    <curve id="f12">                       |   .curve(f12)
      <component>                          |     .component
        <xy>2.102 6.454                    |       .xy
        …
```



Your Program

```
…
Tstr = io.get('input.number(temperature).current')
Efstr = io.get('input.number(Ef).current')

io.put('output.curve(f12).component.xy', xydata)

...
```

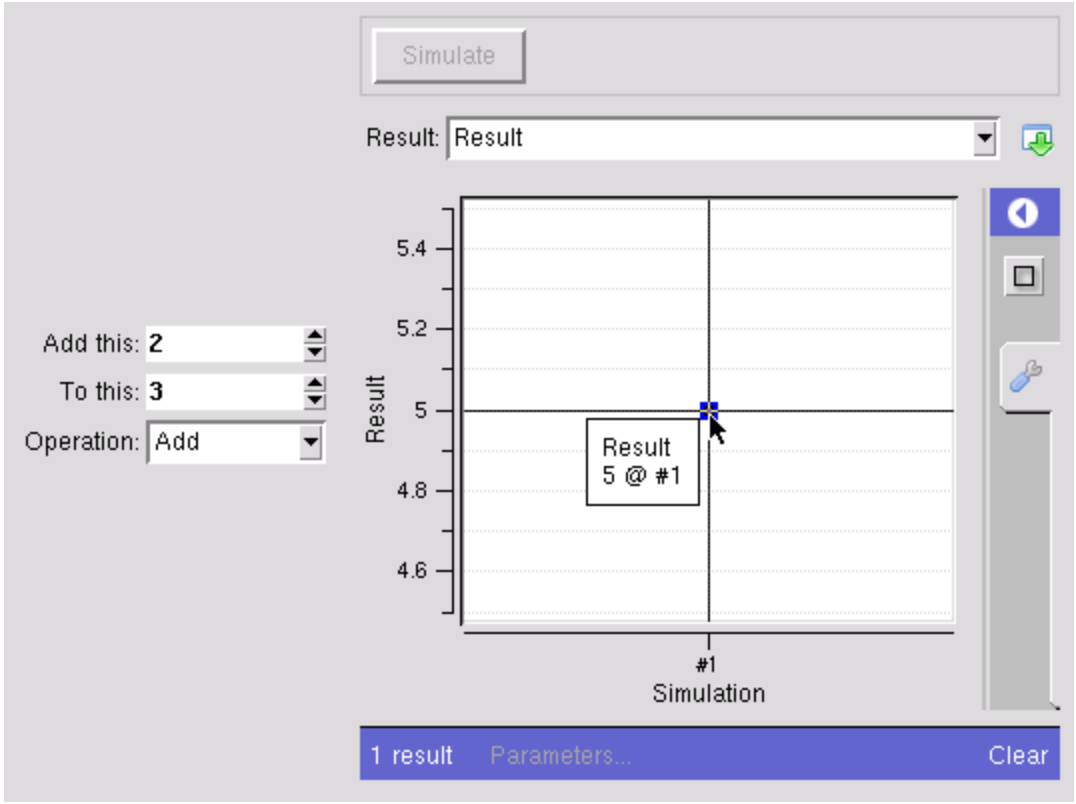
Assignment #2: Patch program into tool.xml

Download the following tool:

```
% wget https://nanohub.org/tools/bootcamp/raw-attachment/wiki/WikiStart/ex2.tgz
```

Unpack and run:

```
% tar xvzf ex2.tgz
% cd ex2
% rappture
```

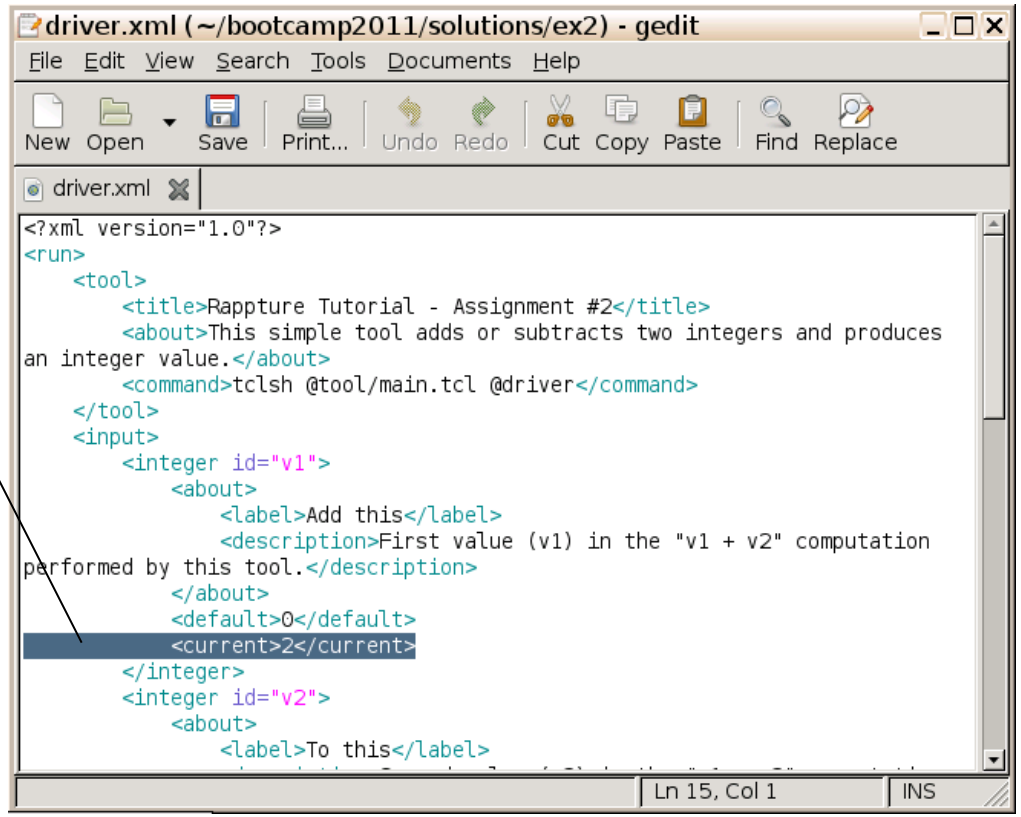


Assignment #2: Patch program into tool.xml

Create a “driver” file by hand:

```
% cp tool.xml driver.xml
% gedit driver.xml
```

Copy/paste each `<default>` value
And fill in a `<current>` value



Run it by hand:

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

Now, edit `<command>` in the `tool.xml` to make the tool run properly