



What's Under the Hood?

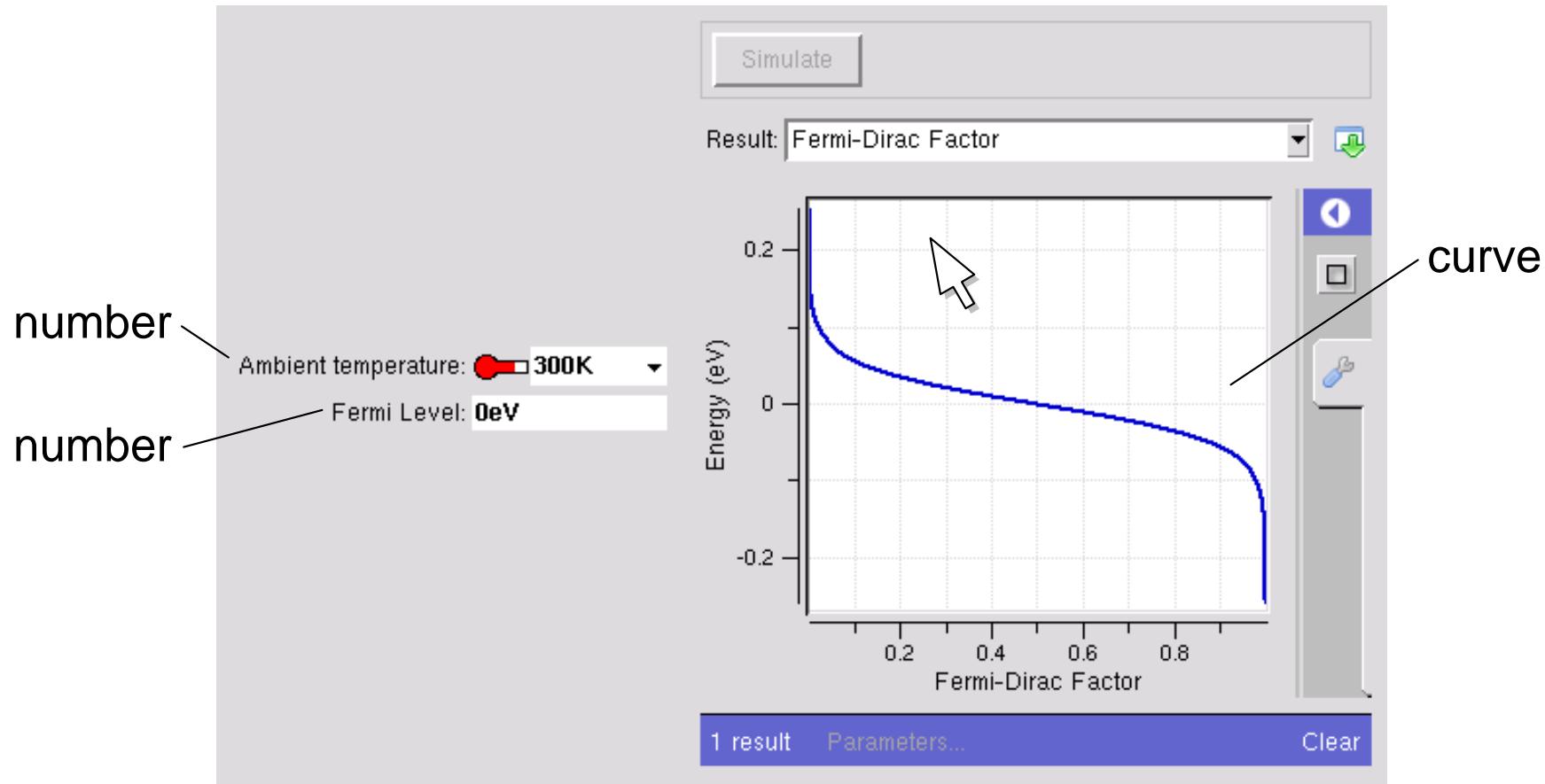
Michael McLennan

HUBzero® Platform for Scientific Collaboration

Purdue University



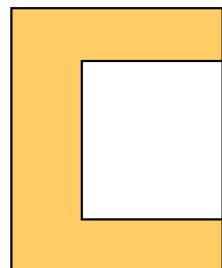
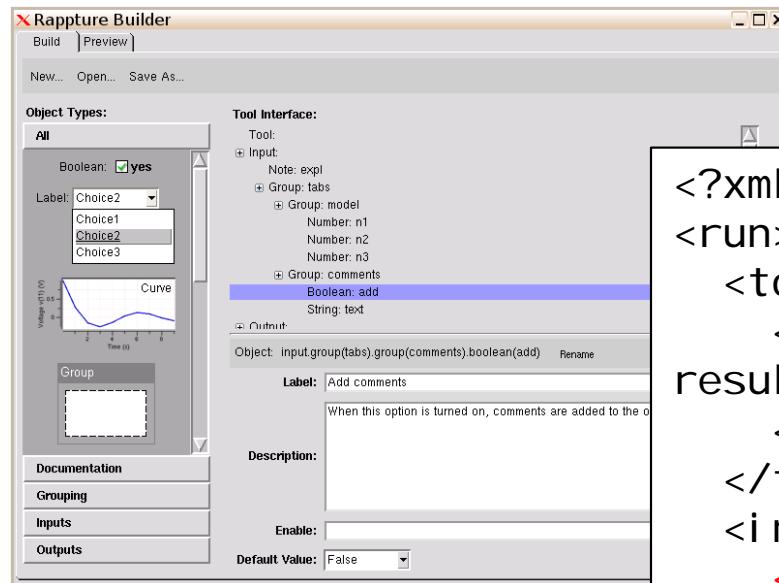
Example: Fermi function tool



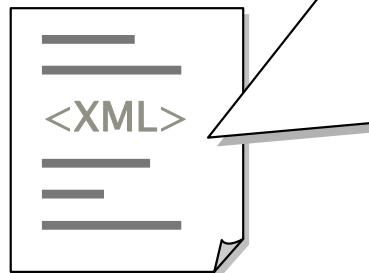
How does it work?



Builder



skeleton
program



tool.xml

```
<?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>
```

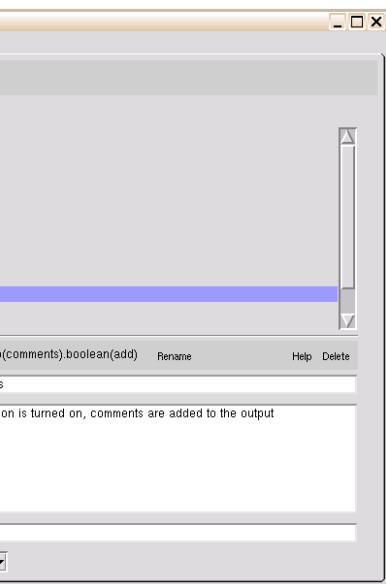
- Info about the tool, how to run it
- Input objects
- Output objects



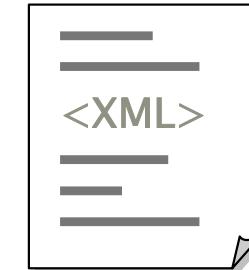
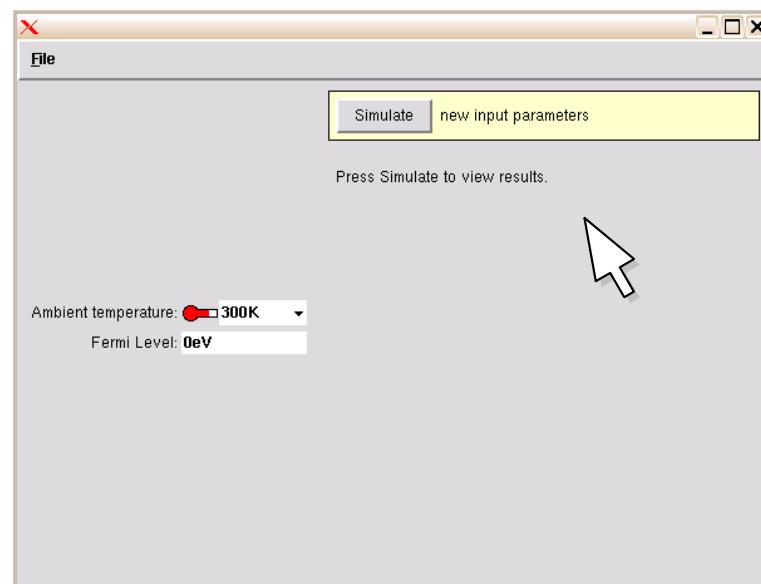
Running produces a driver file

Color xterm
\$ rappture

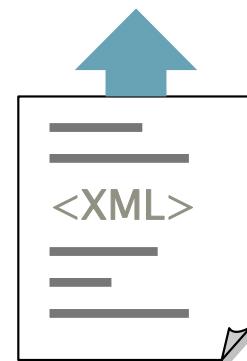
Builder



Runner



driver1827.xml



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>
```

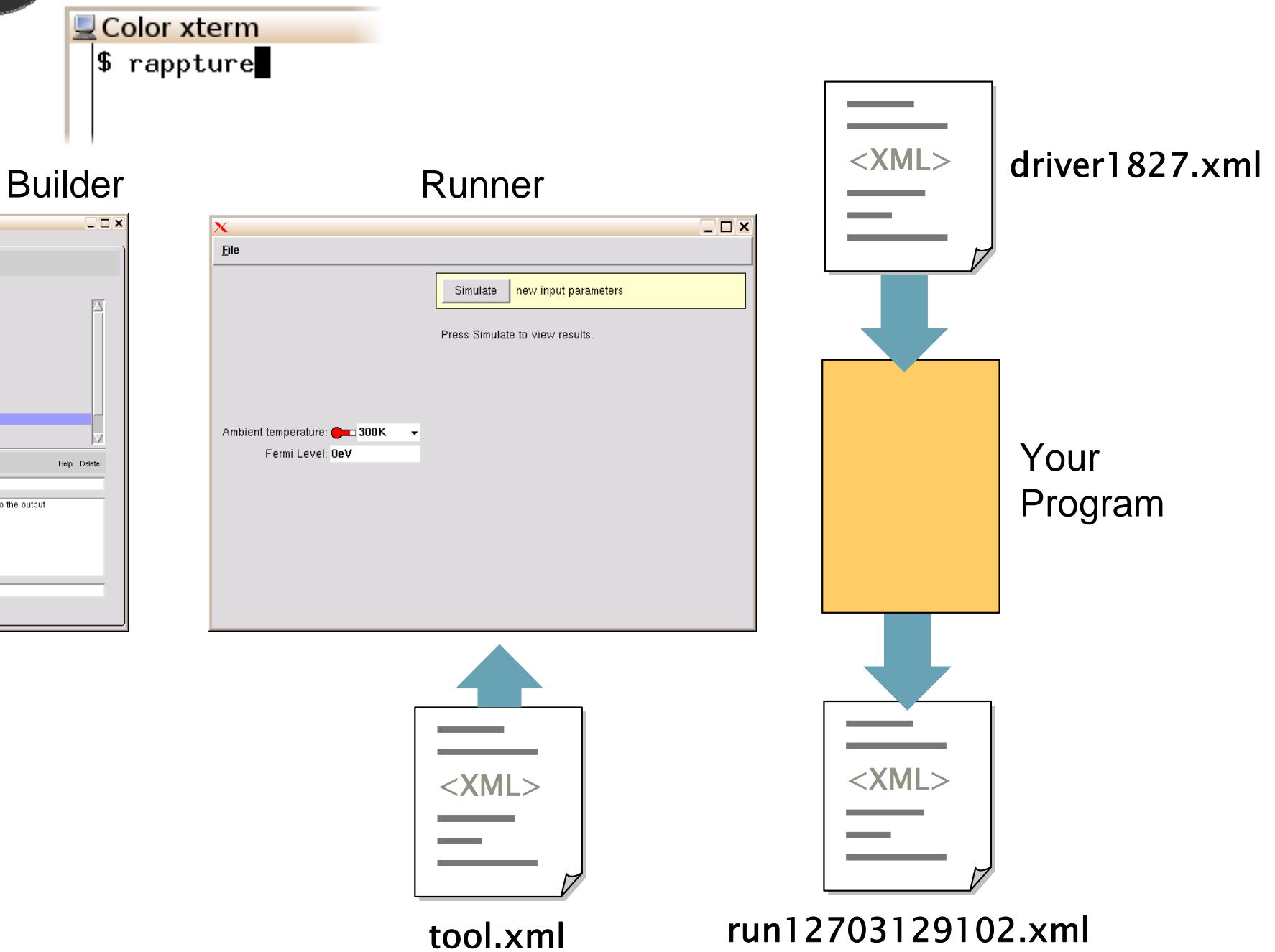
tool.xml

```
<?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





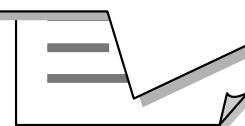
Program produces a run file

```
<?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>...
      ...
    </about>
    <component>
      <xy>0. 999955 -0. 25852
      0. 99995 -0. 255935
      0. 999945 -0. 25335...
    </component>
  </curve>
</output>
</run>
```



driver1827.xml

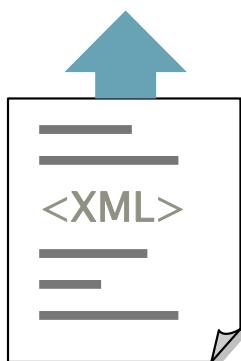
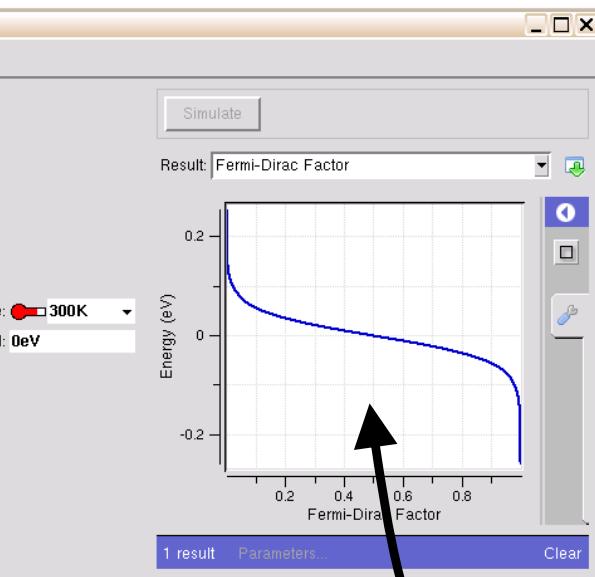
```
<?xml version="1.0"?>
<run>
...
<input>
  <number id="temperature">...
    <current>77K</current>
  </number>
  <number id="Ef">...
    <current>200meV</current>
  </number>
</input>
<output>
  <curve id="f12">
    <about>...
      ...
    </about>
    <component>
      <xy>0. 999955 -0. 25852
      0. 99995 -0. 255935
      0. 999945 -0. 25335...
    </component>
  </curve>
</output>
</run>
```



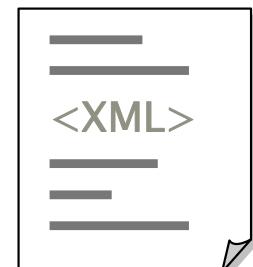
run12703129102.xml



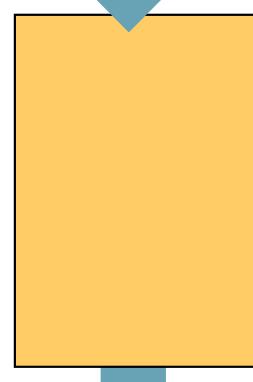
Runner



tool.xml



driver1827.xml



run12703129102.xml

Your
Program

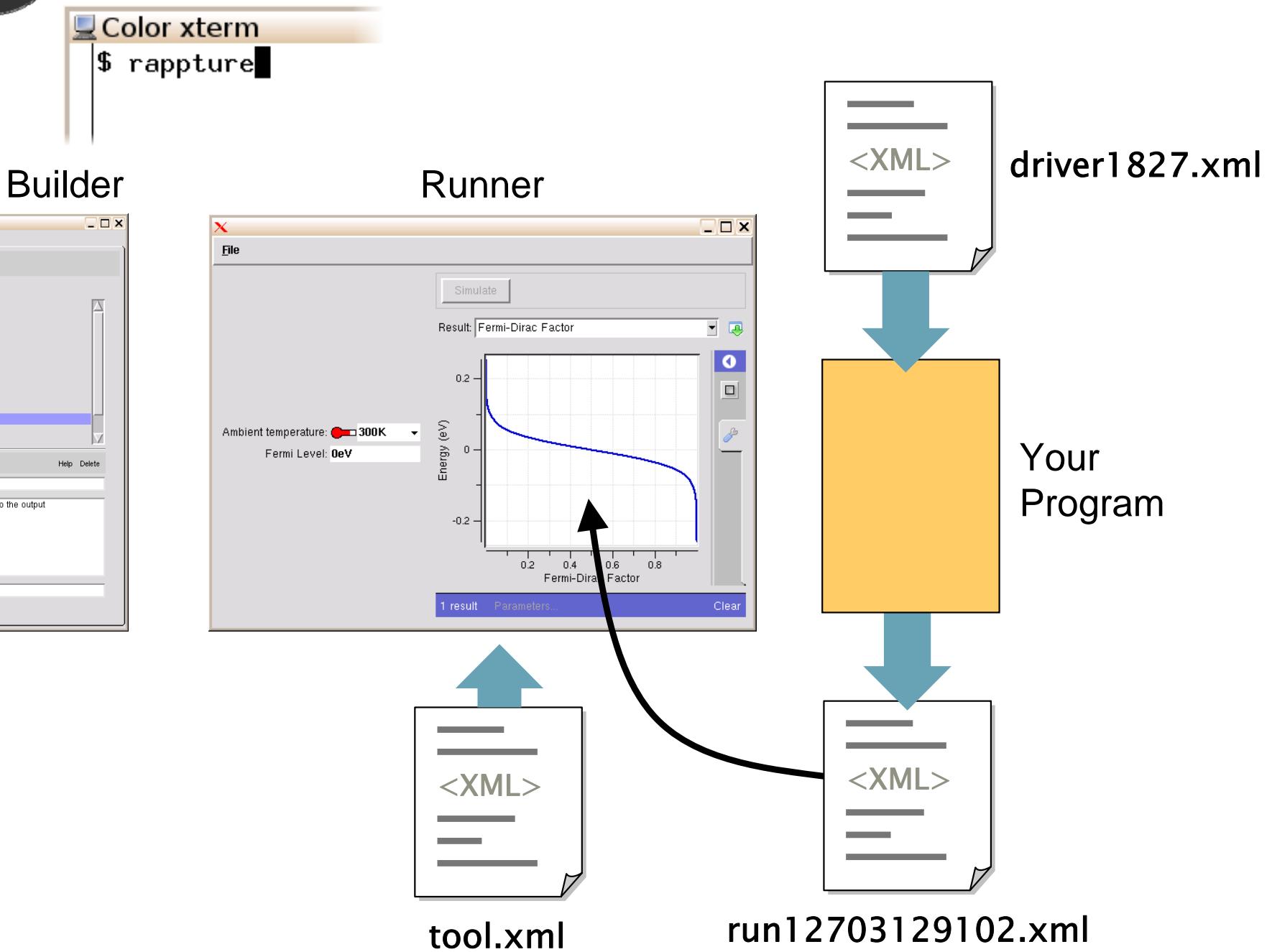
Results are moved out of
the current working directory
to the “results” directory



~/data/results/\$SESSION

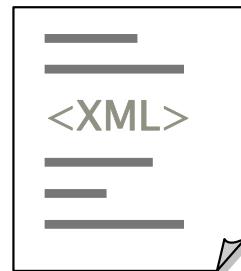


All Together





Run file is a complete record of the run

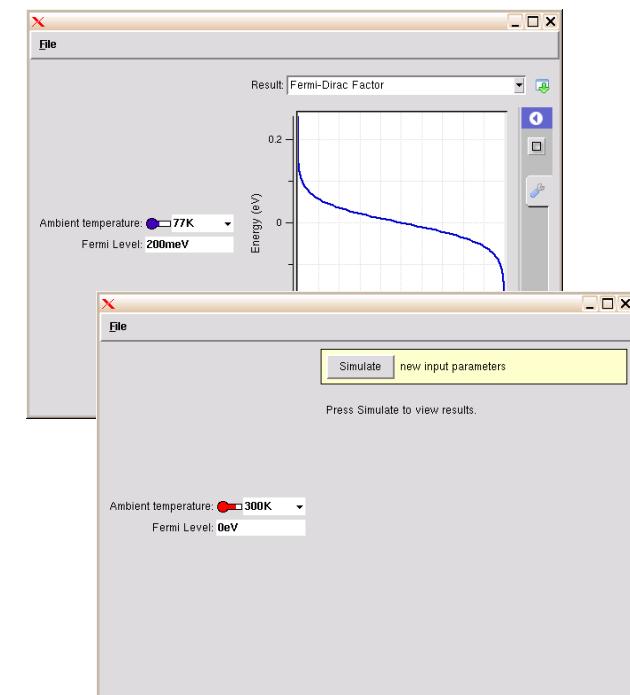


run12703129102.xml

	Tool Defn	Input Vals	Output Vals
tool.xml	<input checked="" type="checkbox"/>		
driver.xml	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
run.xml	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Color xterm
\$ rappture -load run1301535652532513.xml
reload old results

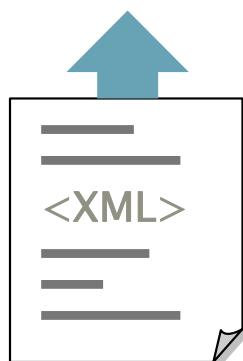
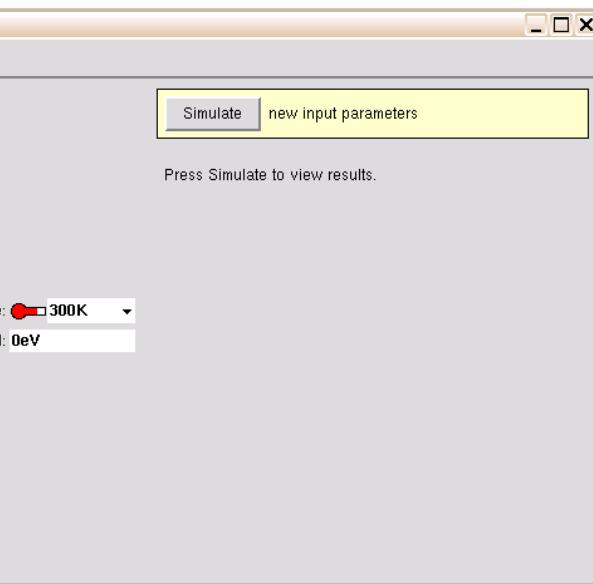
Color xterm
\$ rappture -tool run1301535652532513.xml
rerun an old tool



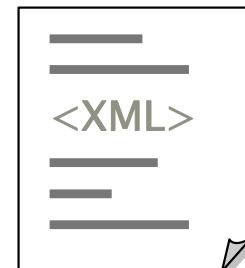


How does your program get invoked?

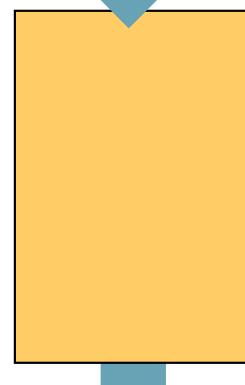
Runner



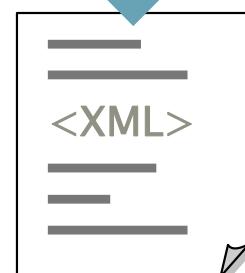
tool.xml



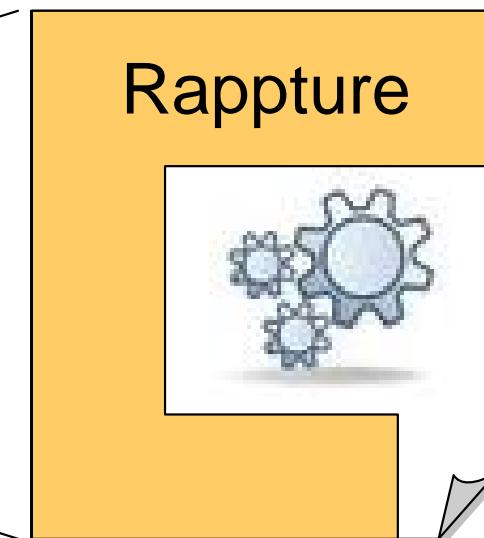
driver1827.xml



Your
Program



run12703129102.xml

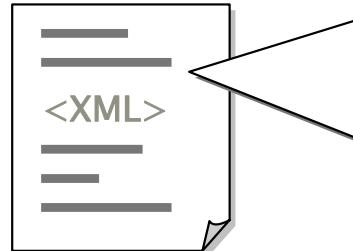


Rappture
Library

Your
Code



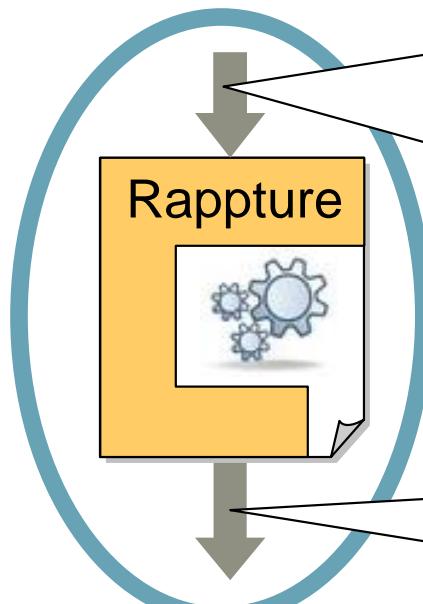
How your program gets invoked



tool.xml

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

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

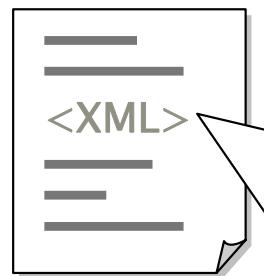


```
import Rappture
import sys
from math import *
io = Rappture.Library(sys.argv[1])

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



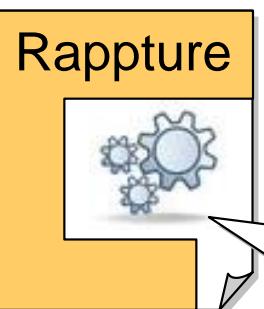
tool.xml

```
<?xml version="1.0"?>
<run>
...
<i nput>
  <number id="temperature">
    <current>300K</current>
  </number>
...
</i nput>

<output>
  <curve id="f12">
    <component>
      <xy>2. 102 6. 454
      ...
    </component>
  </curve>
</output>
```

i nput
. number(temperature)
. current

output
. curve(f12)
. component
. xy



Your Program

```
...
Tstr = i o.get(' i nput. number(temperature). current' )
Efstr = i o.get(' i nput. number(Ef). current' )

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



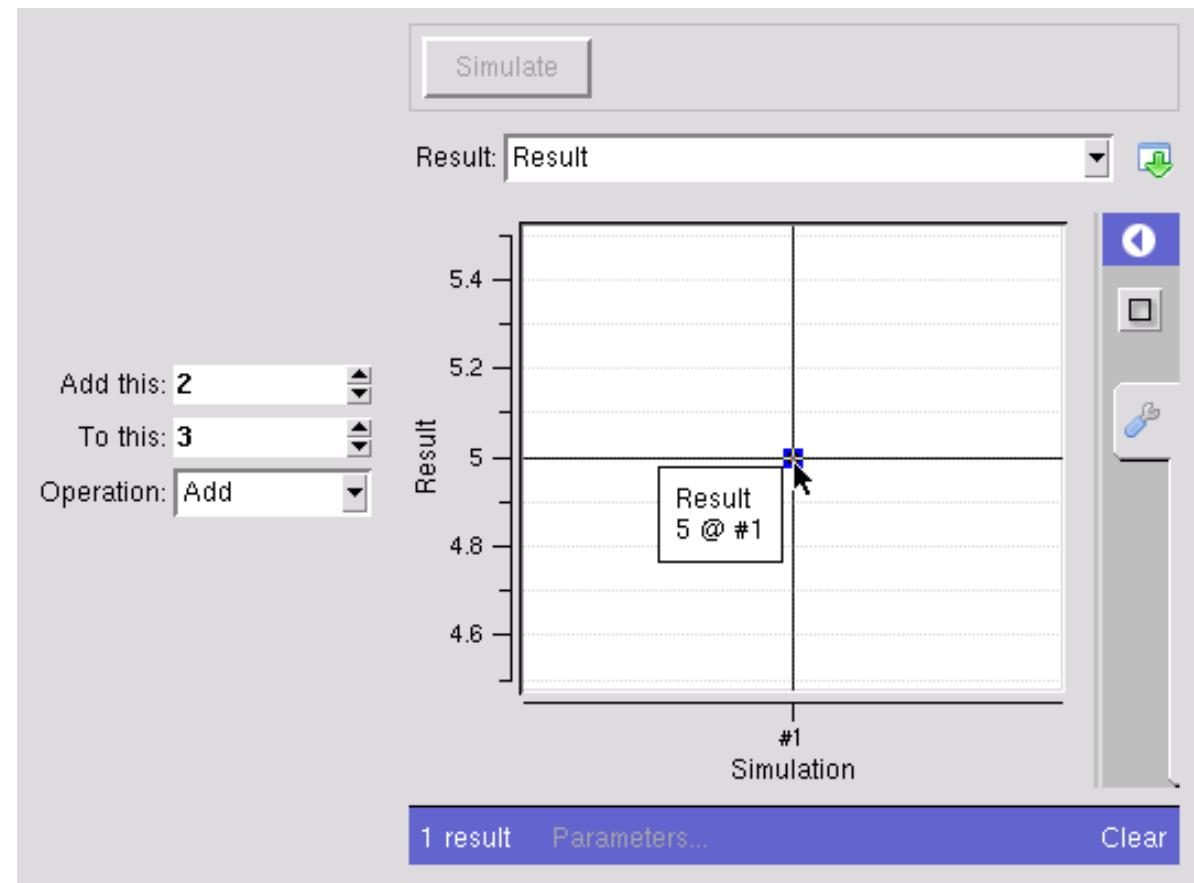
Assignment #2: Patch program into tool.xml

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

```
driver.xml (~/bootcamp2011/solutions/ex2) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
driver.xml
<?xml version="1.0"?>
<run>
  <tool>
    <title>Rappture Tutorial - Assignment #2</title>
    <about>This simple tool adds or subtracts two integers and produces
an integer value.</about>
    <command>tclsh @tool/main.tcl @driver</command>
  </tool>
  <input>
    <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>2</current>
    </integer>
    <integer id="v2">
      <about>
        <label>To this</label>
        ...
      </about>
    </integer>
  </input>
</run>
```

Run it by hand:

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

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