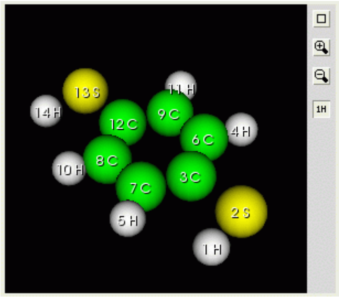
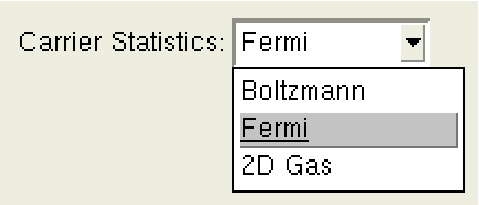
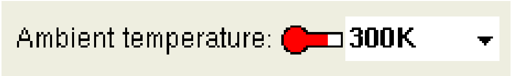
Advanced Rappture Concepts



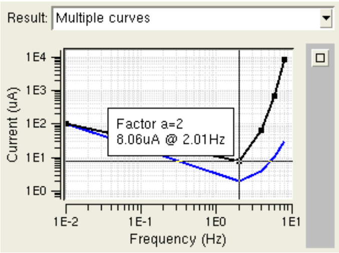


Carrier Statistics: Fermi



Ambient temperature: 300K

Grid points: 100



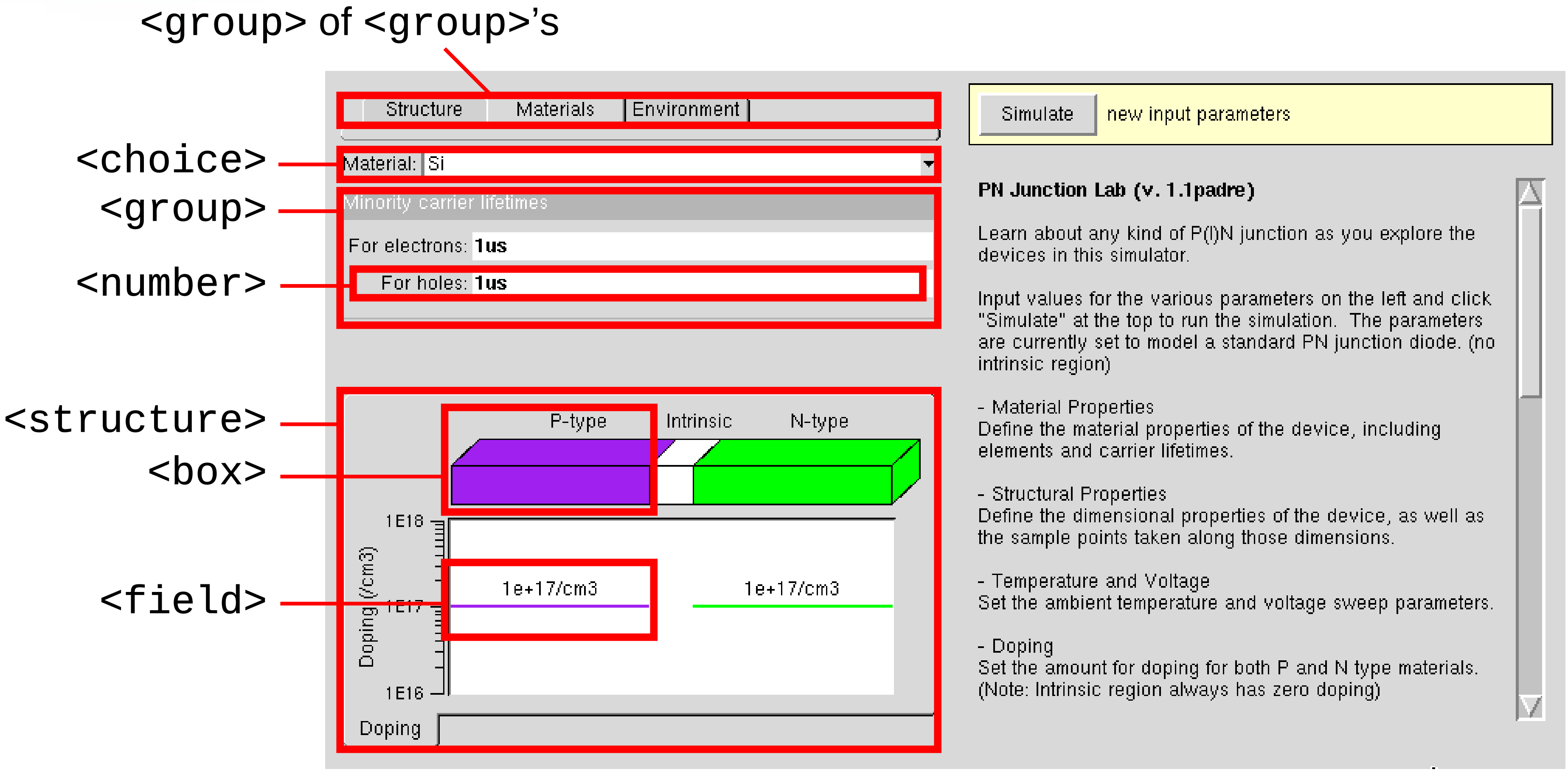
Result: Multiple curves

Michael McLennan

Software Architect

Network for Computational Nanotechnology

Identify the elements



Structure | Materials | Environment

Material: Si

Minority carrier lifetimes

For electrons: 1us

For holes: 1us

Simulate new input parameters

PN Junction Lab (v. 1.1padre)

Learn about any kind of P(I)N junction as you explore the devices in this simulator.

Input values for the various parameters on the left and click "Simulate" at the top to run the simulation. The parameters are currently set to model a standard PN junction diode. (no intrinsic region)

- Material Properties
Define the material properties of the device, including elements and carrier lifetimes.

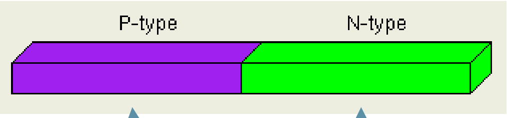
- Structural Properties
Define the dimensional properties of the device, as well as the sample points taken along those dimensions.

- Temperature and Voltage
Set the ambient temperature and voltage sweep parameters.

- Doping
Set the amount for doping for both P and N type materials. (Note: Intrinsic region always has zero doping)

<structure>

Structure of physical system being simulated

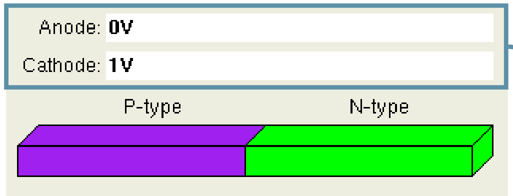


Just 1-D boxes, for now

```
<structure>
  <current>
    <components>
      <box>
        <about>
          <label>P-type</label><color>purple</color>
        </about>
        <corner>0</corner>
        <corner>0.1um</corner>
      </box>
      <box>
        <about>
          <label>N-type</label><color>green</color>
        </about>
        <corner>0.1um</corner>
        <corner>0.2um</corner>
      </box>
    </components>
  </current>
</structure>
```

<structure>

Structure of physical system being simulated



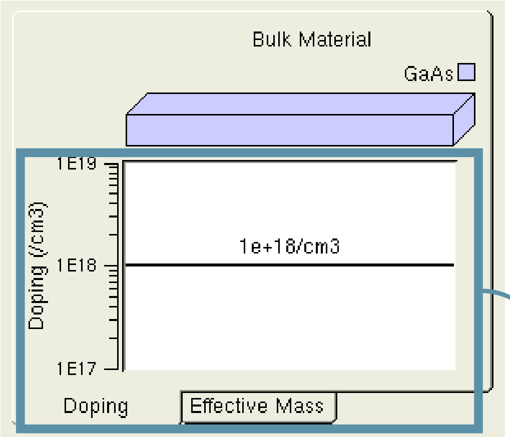
Anode: 0V
Cathode: 1V

```
<structure>
  <current>
    <parameters>
      <number>
        <about><label>Anode:</label></about>
        <units>V</units>
        <default>0V</default>
      </number>
      <number>
        <about><label>Cathode:</label></about>
        <units>V</units>
        <default>1V</default>
      </number>
    </parameters>

    <components>…</components>
  </current>
</structure>
```

Same as before

<structure>

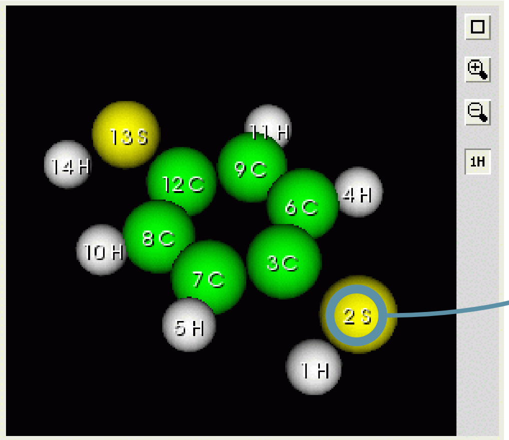
Structure of physical system being simulated



```
<structure>
   <current>
     <units>um</units>
     <parameters>
       <number id="doping">…</number>
     </parameters>
     <components>…</components>
     <fields>
       <field>
         <about>
           <label>Doping</label>
           <color>black</color>
           <scale>log</scale>
         </about>
         <units>/cm3</units>
         <component>
           <constant>doping</constant>
           <domain>box0</domain>
         </component>
       </field>
       …
```

<structure>

Structure of physical system being simulated

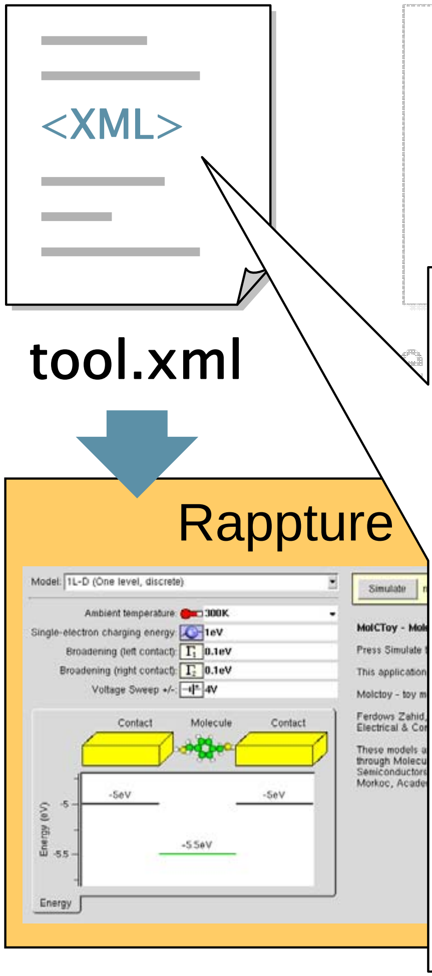


Turns atom labels on by default

```
<structure>
  <current>
    <components>
      <molecule>
        <about><emblems>on</emblems></about>
        <formula>pdt</formula>
        <atom id="0“>
          <symbol>H</symbol>
          <xyz>-1.24935 -3.41562 0.0</xyz>
        </atom>
        <atom id="1">
          <symbol>S</symbol>
          <xyz>0.08092 -3.19426 0.0</xyz>
        </atom>
        …
      </molecule>
    </components>
  </current>
</structure>
```

Focus on <output> side of tool.xml

description of tool, including inputs and outputs



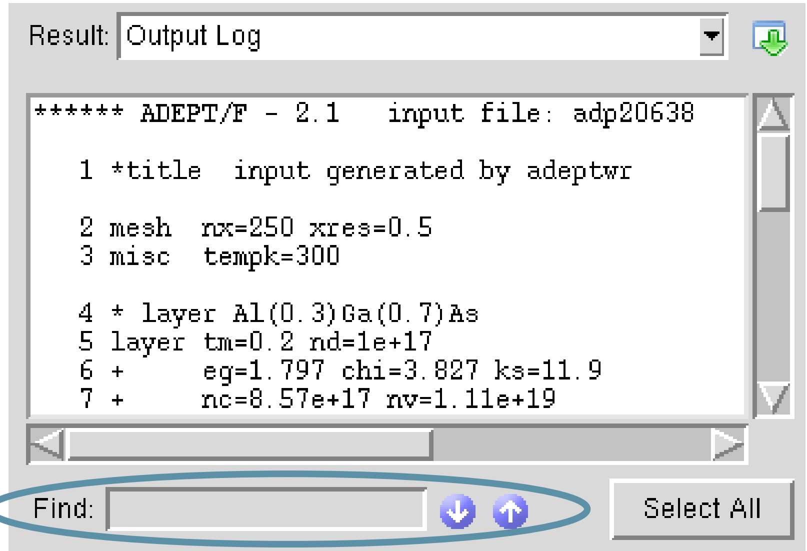
Model: 1L-D (One level, discrete)
Ambient temperature: 300K
Single-electron charging energy: 1eV
Broadening (left contact): 0.1eV
Broadening (right contact): 0.1eV
Voltage Sweep +/-: 4V

```
<?xml version="1.0"?>
<run>
  <tool>
    <about>This is my tool.</about>
    …
  </tool>
  <input>

  </input>
  <output>

  </output>
</run>
```

Produces the user interface *automatically!*

<log>

Standard output from simulator



```
****** ADEPT/F - 2.1   input file: adp20638

   1 *title  input generated by adeptwr

   2 mesh  nx=250 xres=0.5
   3 misc  tempk=300

   4 * layer Al(0.3)Ga(0.7)As
   5 layer tm=0.2 nd=1e+17
   6 +     eg=1.797 chi=3.827 ks=11.9
   7 +     nc=8.57e+17 nv=1.11e+19
```

Controls for search through text

Treated as unimportant
(low level) output, and
therefore listed last

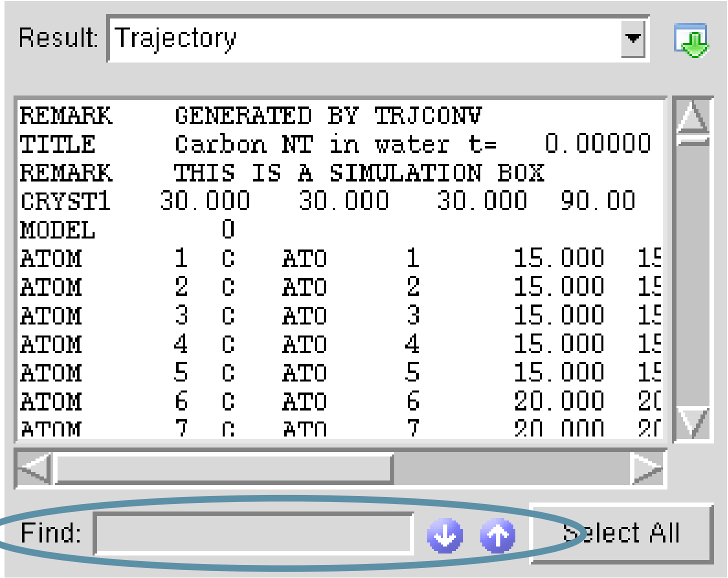
```
<output>
  <log>****** ADEPT/F - 2.1   input
file: adp20638      Sat Jul 30 19:39:36
2005 ******

   1 *title  input generated by adeptwr

   2 mesh  nx=250 xres=0.5
   3 misc  tempk=300
   …
  </log>
</output>
```

or, in Python…

```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
driver.put('output.log', stdout)
```

<string>

Other output files from simulator—including binary files



```
REMARK    GENERATED BY TRJCONV
TITLE     Carbon NT in water t=   0.00000
REMARK    THIS IS A SIMULATION BOX
CRYST1   30.000   30.000   30.000  90.00
MODEL          0
ATOM      1  C   ATO     1      15.000  15
ATOM      2  C   ATO     2      15.000  15
ATOM      3  C   ATO     3      15.000  15
ATOM      4  C   ATO     4      15.000  15
ATOM      5  C   ATO     5      15.000  15
ATOM      6  C   ATO     6      20.000  20
ATOM      7  C   ATO     7      20.000  20
```

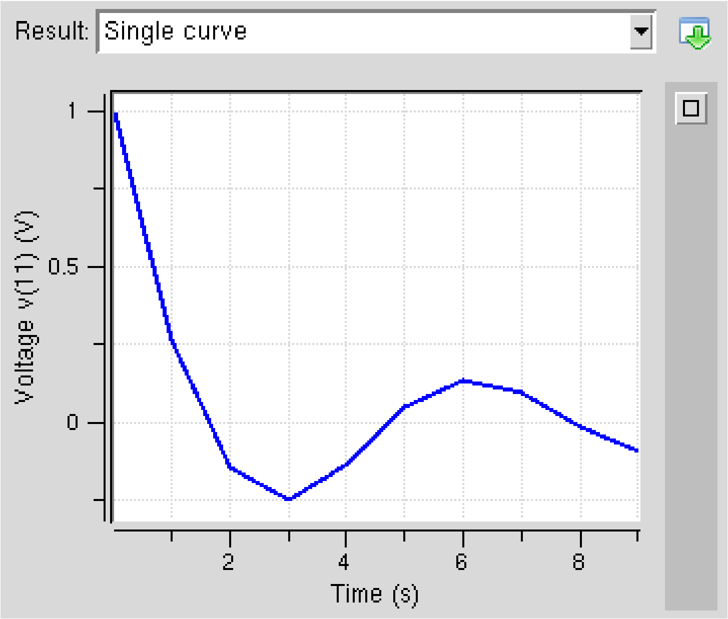
Controls for search through text

```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
path = ‘output.string(traj)’

dr

dr

dr
```

```
<output>
  <string id=“traj”>
    <about>
      <label>Trajectory Data</label>
      <description>Data in pdb
format</description>
    </about>
    <current>REMARK   GENERATED BY
TRJCONV
TITLE     Carbon NT in water t=   0.00000
…
    </current>
  </string>
</output>
```

<curve>

X-Y plots

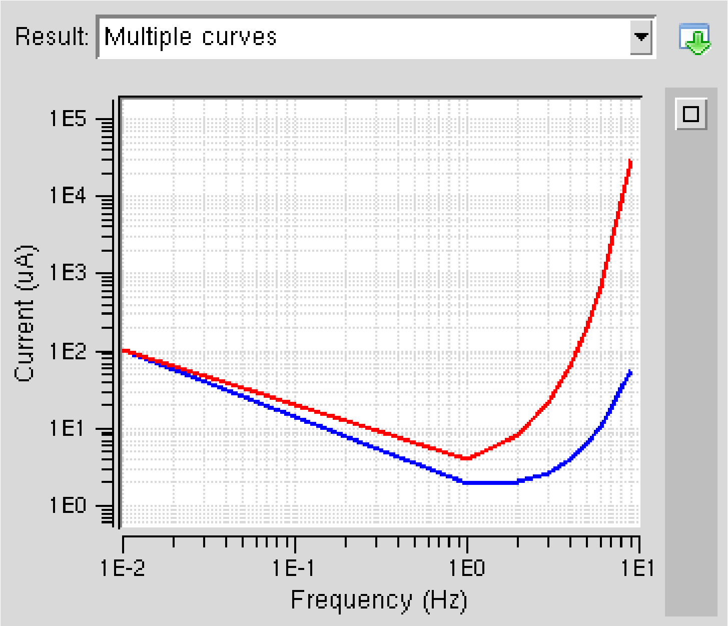


Result: Single curve

```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
path = ‘output.curve(single)’

driver.put(path+’.about.label’,
  ‘Single curve’)

driver.put(path+’.xaxis.label’, ‘Time’)
driver.put(path+’.xaxis.units’, ‘s’)
driver.put(path+’.yaxis.label’, ‘Voltage’)
driver.put(path+’.yaxis.units’, ‘V’)

data = """
0 0
1 2
3 4 """
driver.put(path+’.component.xy’, data)
```

<curve>

Multiple curves on the same plot



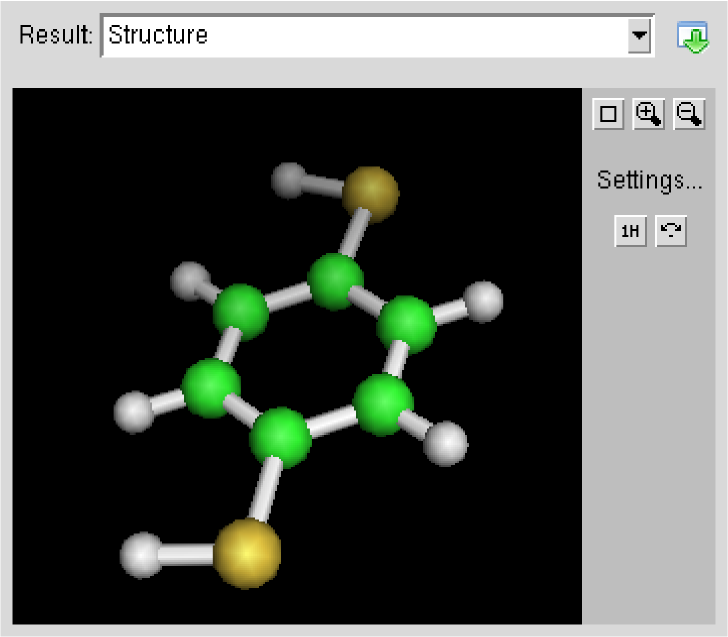
```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
path = ‘output.curve(m1)’
driver.put(path+’.about.group’,
   ’Multiple curves’)
driver.put(path+’.about.label’,
   ’factor a=1’)
…
path = ‘output.curve(m2)’
driver.put(path+’.about.group’,
   ’Multiple curves’)
driver.put(path+’.about.label’,
   ’factor a=2’)
```

Same group name

Different labels
for different curves

<structure>

Molecules



Result: Structure

```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
path = ‘output.structure(mol)’
driver.put(path+’.about.label’, ‘Structure’)

path += ‘components.molecule’
driver.put(path+’.atom(0).symbol’, ‘H’)
driver.put(path+’.atom(0).xyz’, xyz0)
driver.put(path+’.atom(1).symbol’, ‘S’)
driver.put(path+’.atom(1).xyz’, xyz1)
…
```

<field>

Scalar field defined over (x,y) or (x,y,z)

Result: 3D Wavefunctions

```
import Rappture
y[1])
x', dxdata)
tyle',
city 1')
x', dxdat2)
tyle',
-max 1')
```

```
object 1 class gridpositions counts 126 30 22
origin 0.00000000E+00 0.00000000E+00 .0  0  00E+00
delta 50.0 0.0 0.0
delta 0.0 50.0 0.0
delta 0.0 0.0 69.0476190476
object 2 class gridconnections counts 126 30 22
object 3 class array type double rank 0 items 83160 data follows
0.28865594E-02
0.28865594E-02
0.28865594E-02
…
-0.71134413E-02
-0.71134413E-02
attribute "dep" string "positions"
object "regular positions regular connections" class field
component "positions" value 1
component "connections" value 2
component "data" value 3
```

nx ny nz

xgrid

ygrid

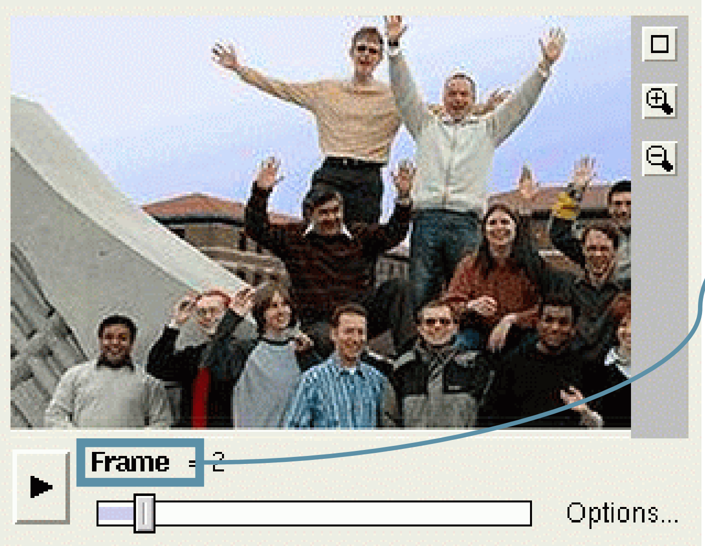
zgrid

Scalar values:
z-index varies fastest,
then y-index,
then x-index

total number of
data points:
nx * ny * nz

<sequence>

Sequence of images, curves, or fields

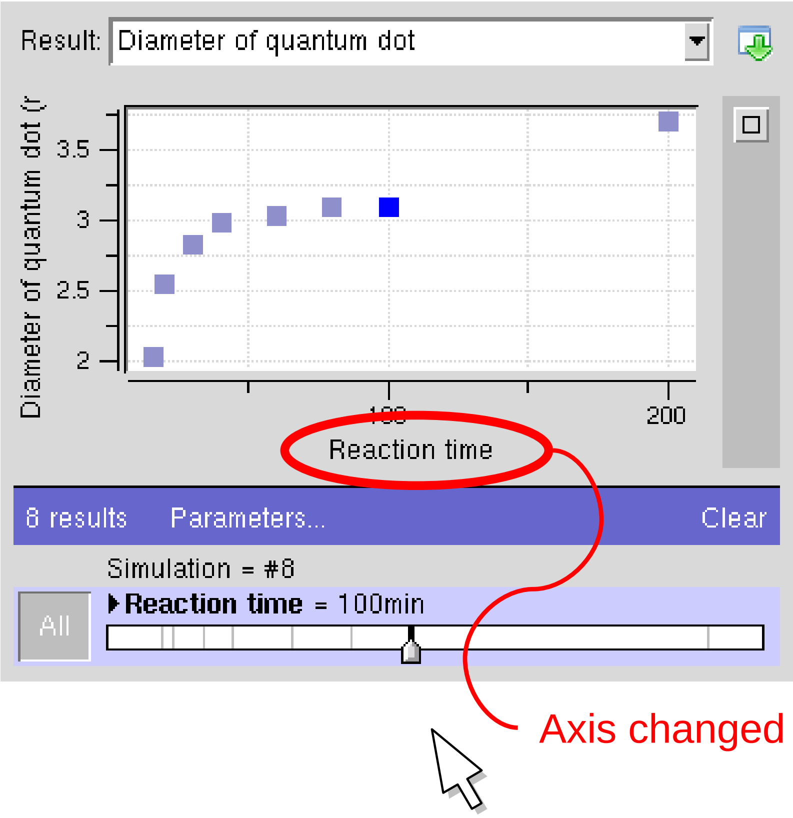


```
<sequence id="movie">
  <about>
    <label>Animated sequence</label>
  </about>
  <index><label>Frame</label></index>

  <element id="0">
    <index>1</index>
    <image>
      <current>/9j/4AAQSkZJRgAA…</current>
    </image>
  </element>

  <element id="1">
    <index>2</index>
    <image>
      <current>/9j/4ARgASkZJQQR…</current>
    </image>
  </element>
  …
</sequence>
```

Just like a normal output image

<number> and <integer>

Just a number or integer, but compare across many runs



Result: Diameter of quantum dot

8 results Parameters... Clear

Simulation = #8

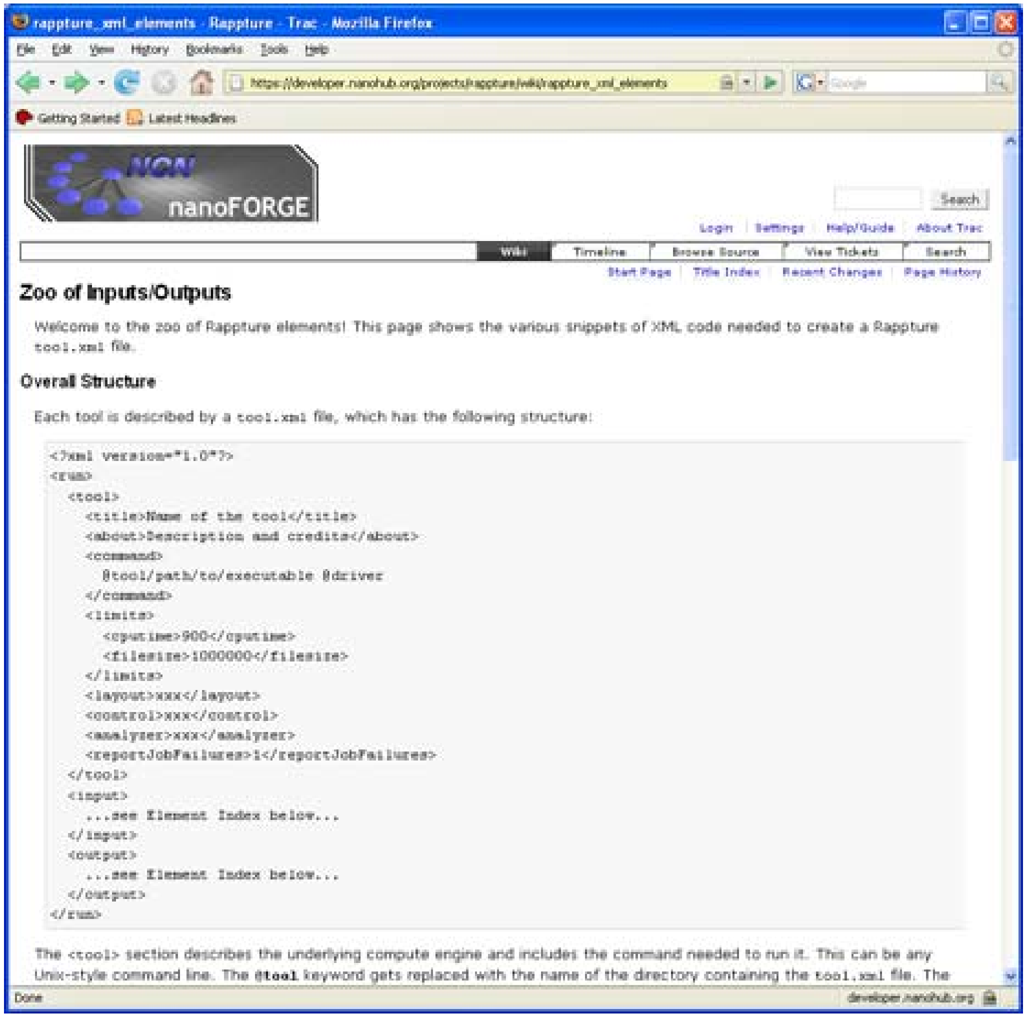
▸ **Reaction time** = 100min

All

Axis changed

```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
path = ‘output.number(d)’

driver.put(path+’.about.label’,
   ‘Diameter of quantum dot’)

driver.put(path+’.units’, ‘nm’)
driver.put(path+’.current’, d)
…
```

Tour the zoo


rappture_xml_elements - Rappture - Trac - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://developer.nanohub.org/projects/rappture/wiki/rappture_xml_elements
Getting Started
Latest Headlines
NCN nanoFORGE
Search
Login | Settings | Help/Guide | About Trac
Wiki
Timeline
Browse Source
View Tickets
Search
Start Page | Title Index | Recent Changes | Page History
Done
developer.nanohub.org

Zoo of Inputs/Outputs

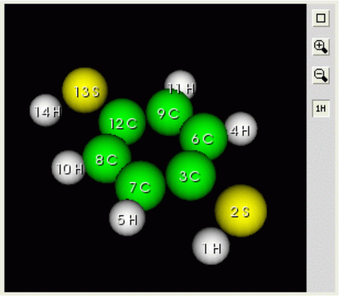
Welcome to the zoo of Rappture elements! This page shows the various snippets of XML code needed to create a Rappture `tool.xml` file.

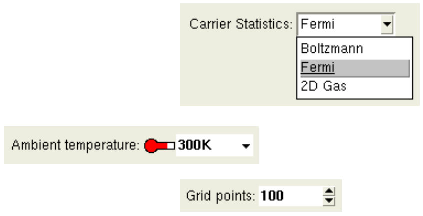
Overall Structure

Each tool is described by a `tool.xml` file, which has the following structure:

```
<?xml version="1.0"?>
<run>
  <tool>
    <title>Name of the tool</title>
    <about>Description and credits</about>
    <command>
      @tool/path/to/executable @driver
    </command>
    <limits>
      <cputime>900</cputime>
      <filesize>1000000</filesize>
    </limits>
    <layout>xxx</layout>
    <control>xxx</control>
    <analyzer>xxx</analyzer>
    <reportJobFailures>1</reportJobFailures>
  </tool>
  <input>
    ...see Element Index below...
  </input>
  <output>
    ...see Element Index below...
  </output>
</run>
```

The `<tool>` section describes the underlying compute engine and includes the command needed to run it. This can be any Unix-style command line. The **`@tool`** keyword gets replaced with the name of the directory containing the `tool.xml` file. The





Carrier Statistics: Fermi

- Boltzmann
- Fermi
- 2D Gas

Ambient temperature: 300K

Grid points: 100

Zoo of Examples

- Complete catalog of data objects online
- See screen shots
- Copy xml code