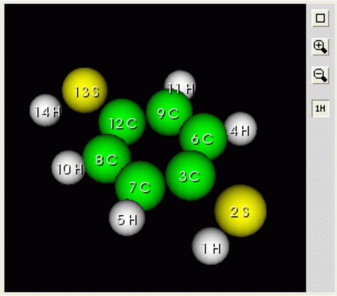
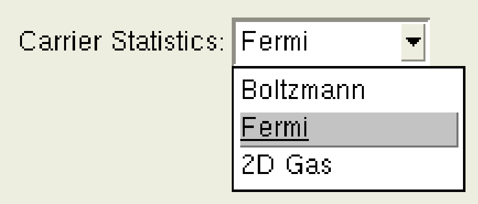
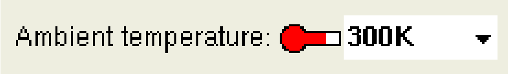
Hub Technology Group
Rosen Center for Advanced Computing

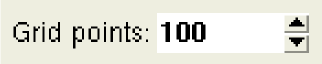
Advanced Rappture Concepts and Tips

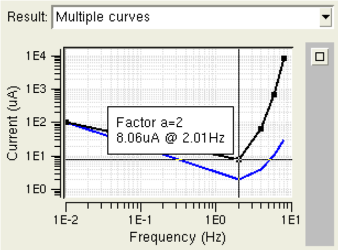



Carrier Statistics: Fermi

Boltzmann
Fermi
2D Gas

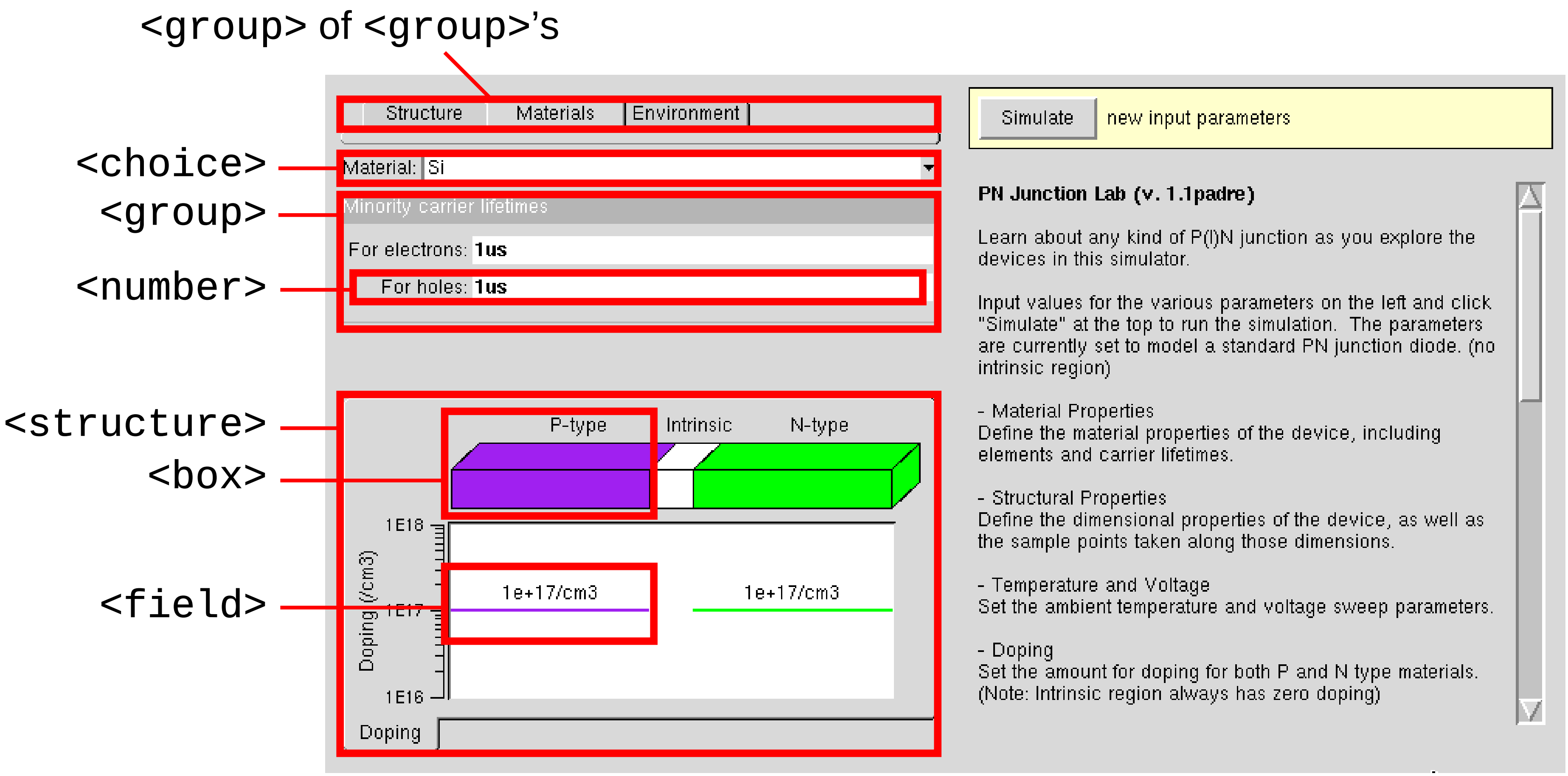

Ambient temperature: 300K


Grid points: 100


Result: Multiple curves

Michael McLennan
Software Architect
HUBzero™ Platform for Scientific Collaboration

Identify the elements



Material: Si

For electrons: 1us

For holes: 1us

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.

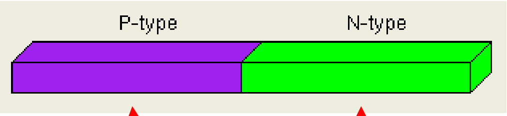
- Structural Properties
Define the dimensional properties of the device, as well as the sample points taken along those dimensions.

- 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

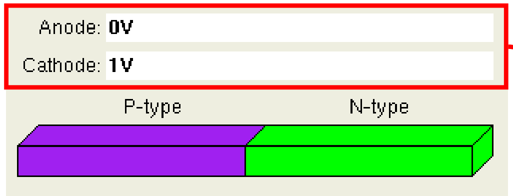


Just 1-D boxes, for now

```
<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



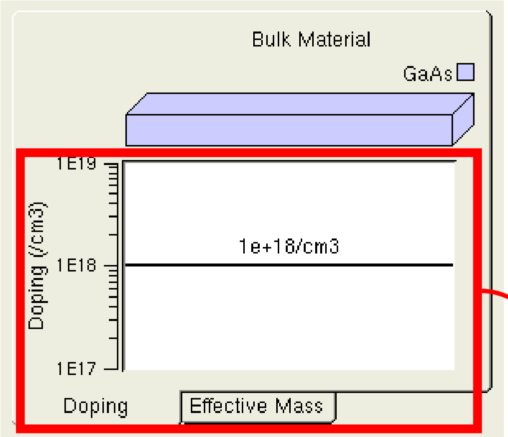
Anode: 0V
Cathode: 1V

```
<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>

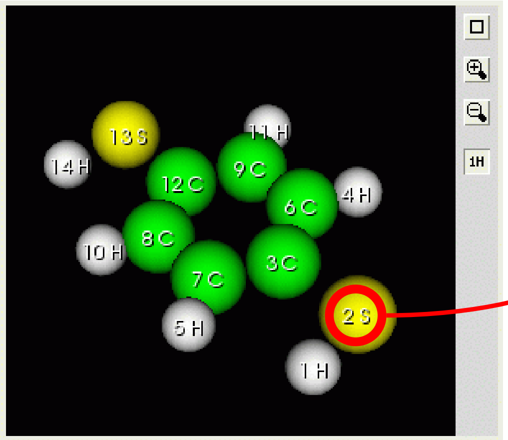
Structure of physical system being simulated



```
<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

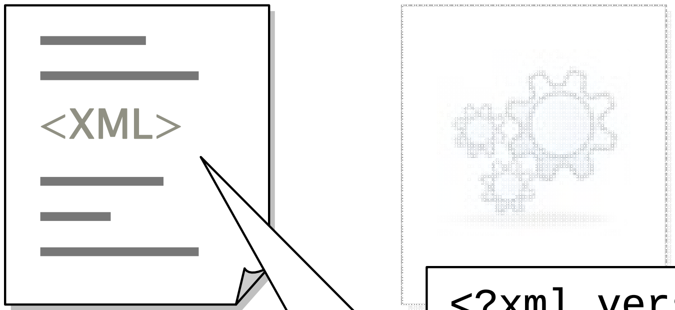


Turns atom labels on by default

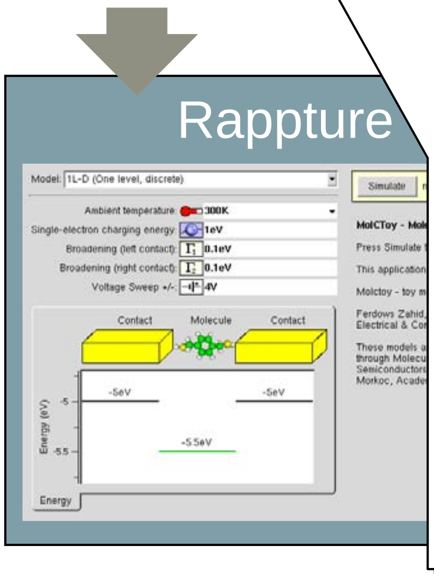
```
<structure>
  <current>
    <components>
      <molecule>
        <about><emblems>on</emblems></about>
        <formula>pdt</formula>
        <atom id="0“>
          <symbol>H</symbol>
          <xyz>-1.24935 -3.41562 0.0</xyz>
        </atom>
        <atom id="1">
          <symbol>S</symbol>
 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



tool.xml



Model: 1L-D (One level, discrete)

Ambient temperature: 300K

Single-electron charging energy: 1eV

Broadening (left contact): 0.1eV

Broadening (right contact): 0.1eV

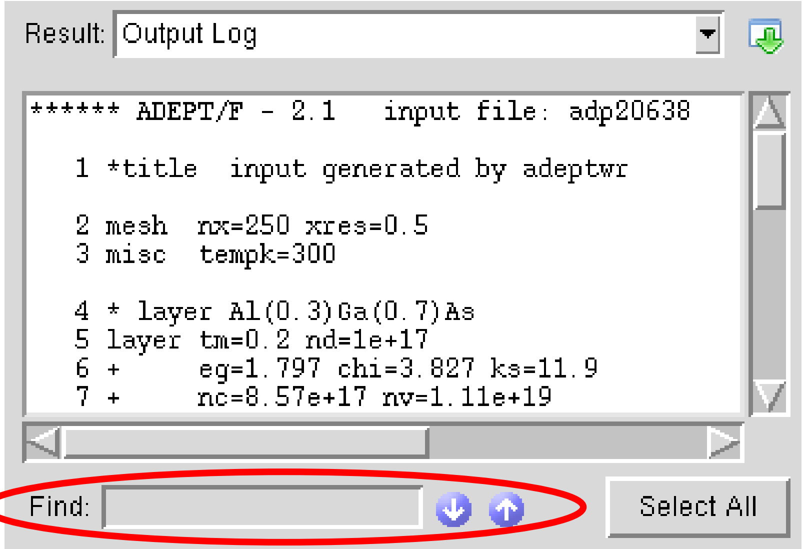
Voltage Sweep +/-: 4V

```
<?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

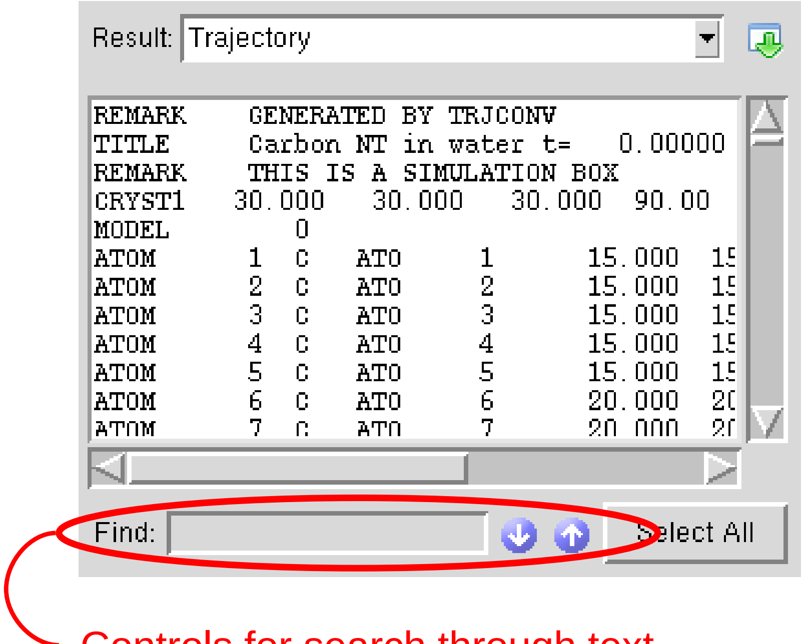
```
<output>
  <log>****** ADEPT/F - 2.1   input
file: adp20638      Sat Jul 30 19:39:36
2005 ******

   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
```

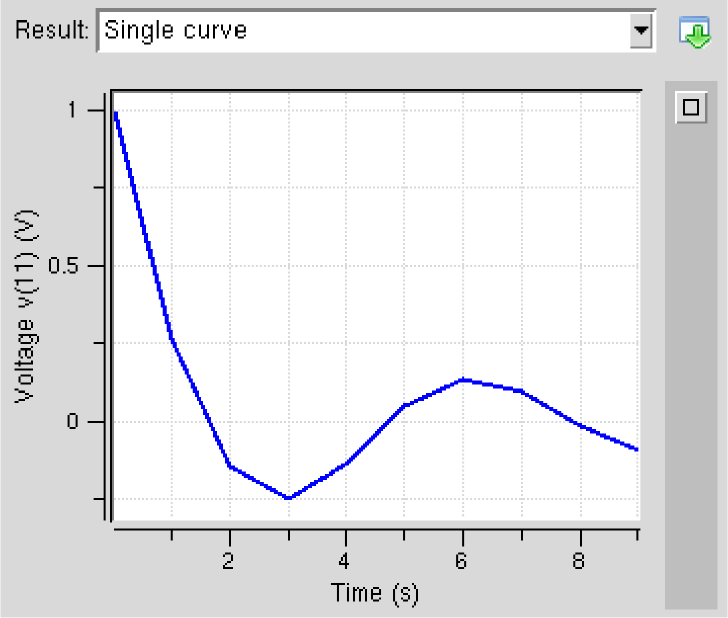
Controls for search through text

```
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

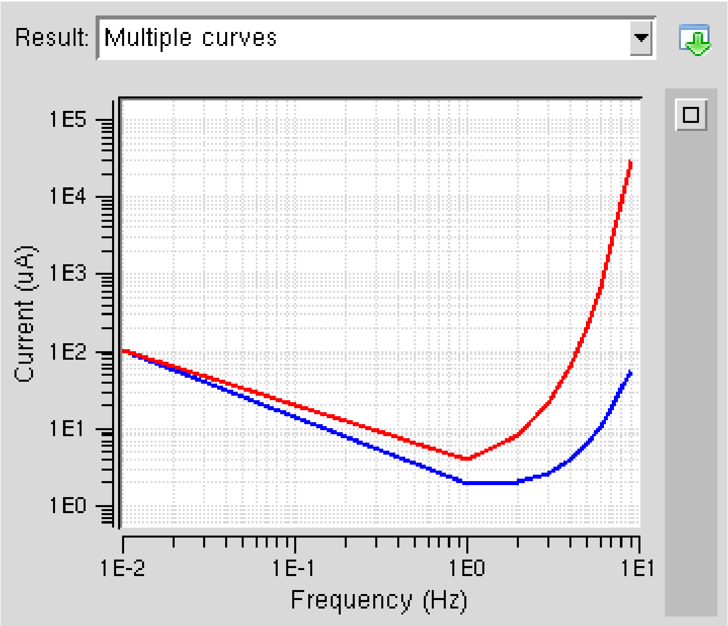


Result: Single curve

```
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



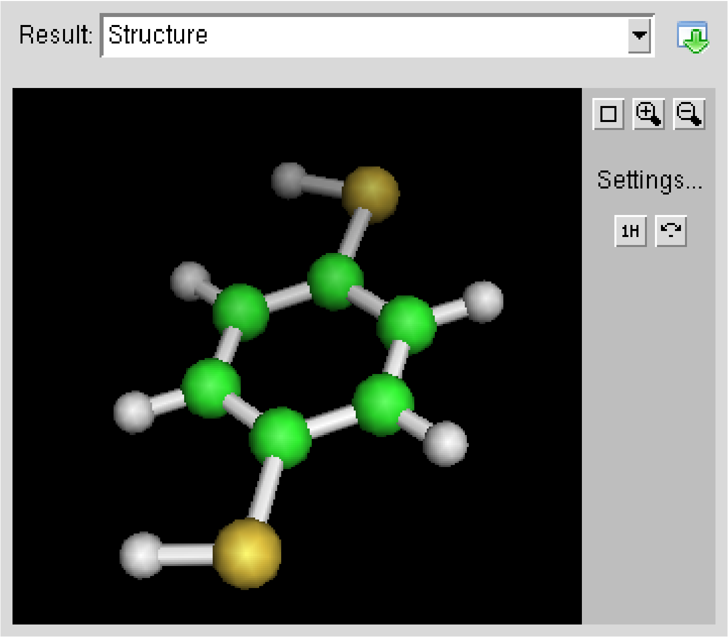
```
import Rappture
import sys
driver = Rappture.library(sys.argv[1])
…
path = ‘output.curve(m1)’
driver.put(path+’.about.group’,
   ’Multiple curves’)
driver.put(path+’.about.label’,
   ’factor a=1’)
…
path = ‘output.curve(m2)’
driver.put(path+’.about.group’,
   ’Multiple curves’)
driver.put(path+’.about.label’,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

Same group name

Different labels
for different curves

<structure>

Molecules



```
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)

```
import Rappture
[1])
’, dxdata)
yle’,
ity 1’)
’, dxdat2)
yle’,
-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>

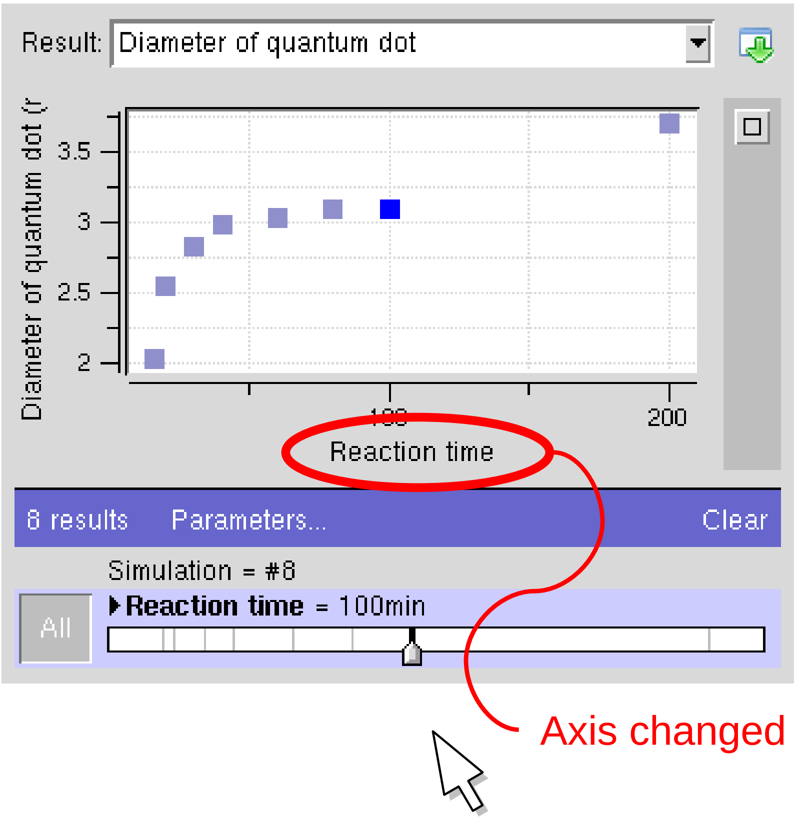
Sequence of images, curves, or fields

```
<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

Simulation = #8

Reaction time = 100min

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)
…
```

Describe everything

What are these two parameters?

Model parameters

Tight Binding Energy: 3eV

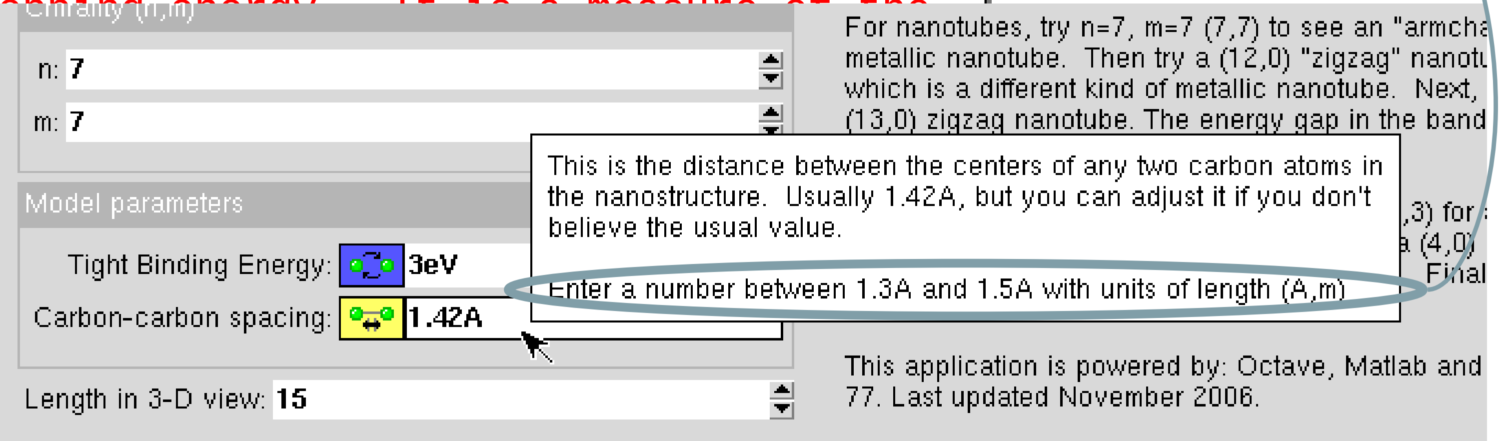
Carbon-carbon spacing: 1.42A

Good descriptions say:

- What the parameter means
- Typical value or range
- What happens at 0 or min
- What happens at ∞ or max

```
<number id="TightBindingEnergy">
  <about>
    <label>Tight Binding Energy</label>
    <description>This is the tight binding overlap
integral, or h
overlap of orb
nanotube.  Typ
references on
details.</desc
    ...
```

Rappture generates this part automatically



Chirality (n,m)

n: 7

m: 7

For nanotubes, try n=7, m=7 (7,7) to see an "armcha
metallic nanotube. Then try a (12,0) "zigzag" nanotu
which is a different kind of metallic nanotube. Next,
(13,0) zigzag nanotube. The energy gap in the band
,3) for
a (4,0)
Final

Model parameters

Tight Binding Energy: 3eV

Carbon-carbon spacing: 1.42A

This is the distance between the centers of any two carbon atoms in the nanostructure. Usually 1.42A, but you can adjust it if you don't believe the usual value.

Enter a number between 1.3A and 1.5A with units of length (A,m)

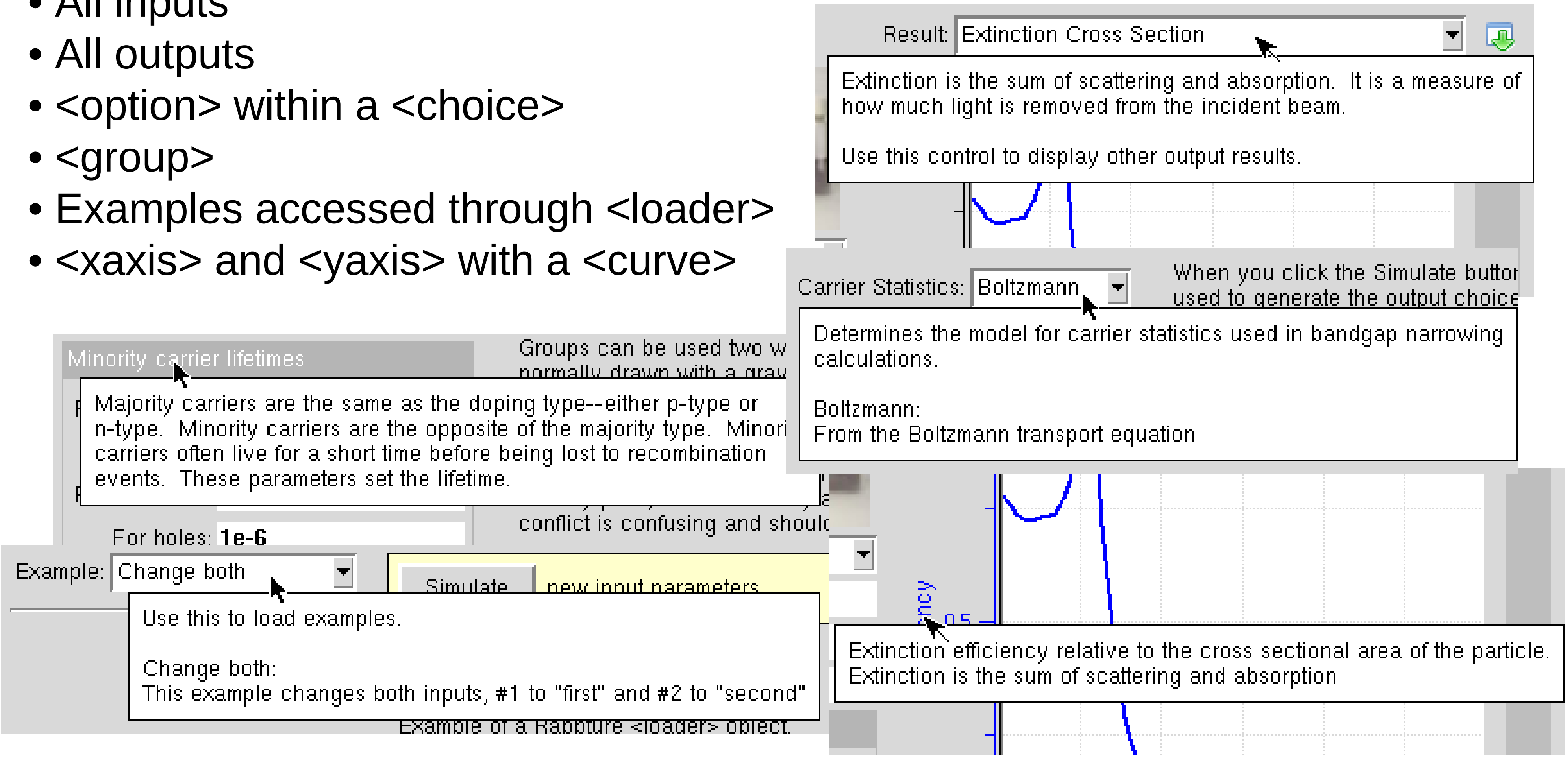
Length in 3-D view: 15

This application is powered by: Octave, Matlab and
77. Last updated November 2006.

Describe EVERYTHING

You can add `<description>`’s to…

- All inputs
- All outputs
- <option> within a <choice>
- <group>
- Examples accessed through <loader>
- <xaxis> and <yaxis> with a <curve>



Result: Extinction Cross Section

Extinction is the sum of scattering and absorption. It is a measure of how much light is removed from the incident beam.

Use this control to display other output results.

Carrier Statistics: Boltzmann

When you click the Simulate buttor used to generate the output choice

Determines the model for carrier statistics used in bandgap narrowing calculations.

Boltzmann:
From the Boltzmann transport equation

Minority carrier lifetimes

Groups can be used two w normally drawn with a gray

Majority carriers are the same as the doping type--either p-type or n-type. Minority carriers are the opposite of the majority type. Minori carriers often live for a short time before being lost to recombination events. These parameters set the lifetime.

conflict is confusing and shoulc

For holes: 1e-6

Example: Change both

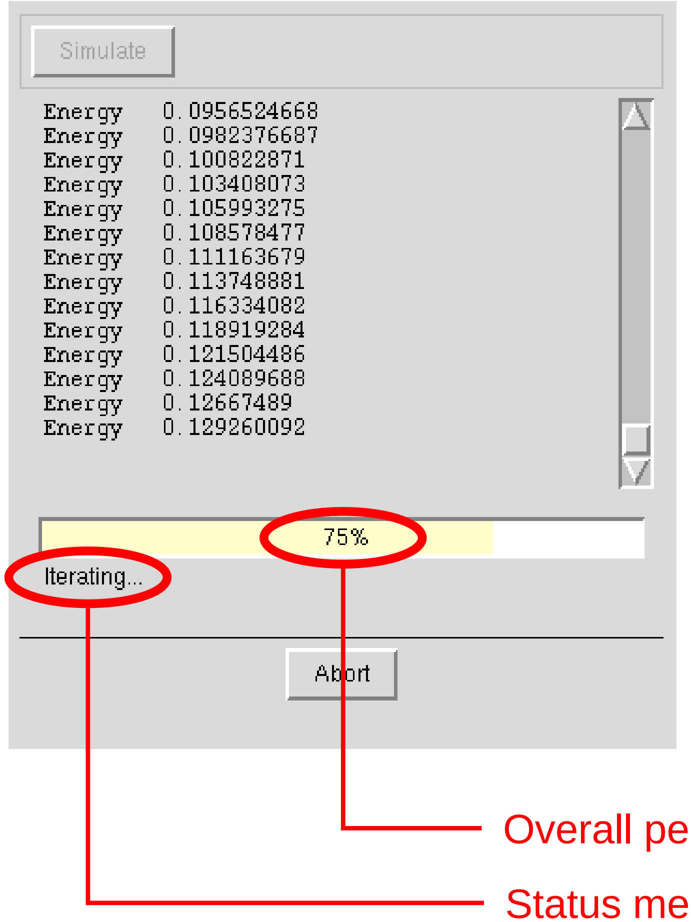
Use this to load examples.

Change both:
This example changes both inputs, #1 to "first" and #2 to "second"

Example of a Rappture <loader> object.

Extinction efficiency relative to the cross sectional area of the particle.
Extinction is the sum of scattering and absorption

Give feedback during simulation



```
Energy    0.0956524668
Energy    0.0982376687
Energy    0.100822871
Energy    0.103408073
Energy    0.105993275
Energy    0.108578477
Energy    0.111163679
Energy    0.113748881
Energy    0.116334082
Energy    0.118919284
Energy    0.121504486
Energy    0.124089688
Energy    0.12667489
Energy    0.129260092
```

Overall percentage complete

Status message

in Fortran:

```
call rp_utils_progress (75, “Iterating...”)
```

in C:

```
rpUtilsProgress(75, “Iterating...”);
```

in Matlab:

```
rpUtilsProgress(75, ‘Iterating...’)
```

in Python:

```
Rappture.Utils.progress(75, ‘Iterating...’)
```

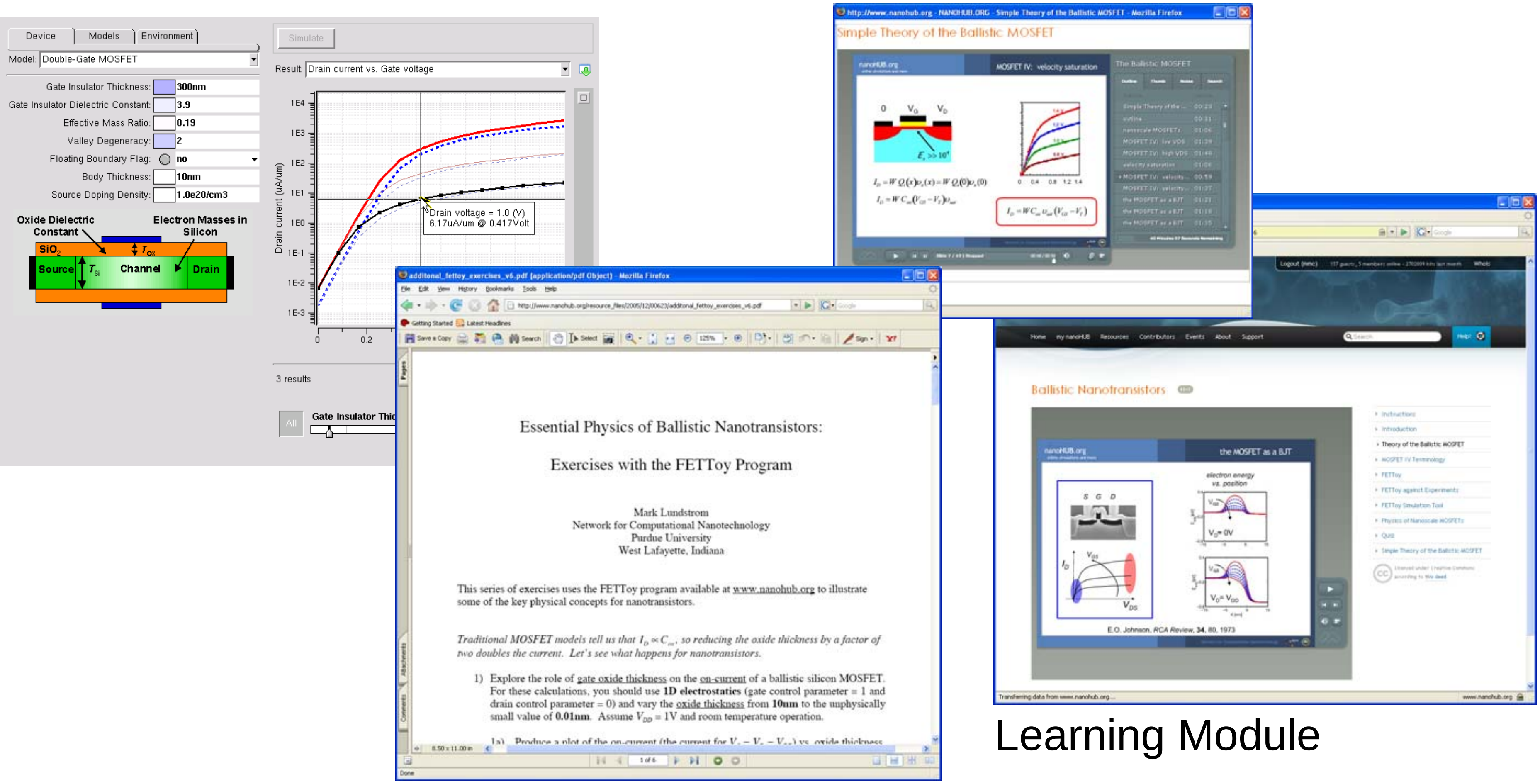
in Perl:

```
Rappture::Utils::progress(75, “Iterating...”)
```

Provide supporting materials

FETToy simulator

Seminar



Model: Double-Gate MOSFET
Gate Insulator Thickness: 300nm
Gate Insulator Dielectric Constant: 3.9
Effective Mass Ratio: 0.19
Valley Degeneracy: 2
Floating Boundary Flag: no
Body Thickness: 10nm
Source Doping Density: 1.0e20/cm3

Result: Drain current vs. Gate voltage

Drain voltage = 1.0 (V)
6.17uA/um @ 0.417Volt

3 results

Essential Physics of Ballistic Nanotransistors:

Exercises with the FETToy Program

Mark Lundstrom
Network for Computational Nanotechnology
Purdue University
West Lafayette, Indiana

This series of exercises uses the FETToy program available at www.nanohub.org to illustrate some of the key physical concepts for nanotransistors.

Traditional MOSFET models tell us that $I_D \propto C_{ox}$, so reducing the oxide thickness by a factor of two doubles the current. Let's see what happens for nanotransistors.

1) Explore the role of gate oxide thickness on the on-current of a ballistic silicon MOSFET. For these calculations, you should use **1D electrostatics** (gate control parameter = 1 and drain control parameter = 0) and vary the oxide thickness from **10nm** to the unphysically small value of **0.01nm**. Assume V_{DD} = 1V and room temperature operation.

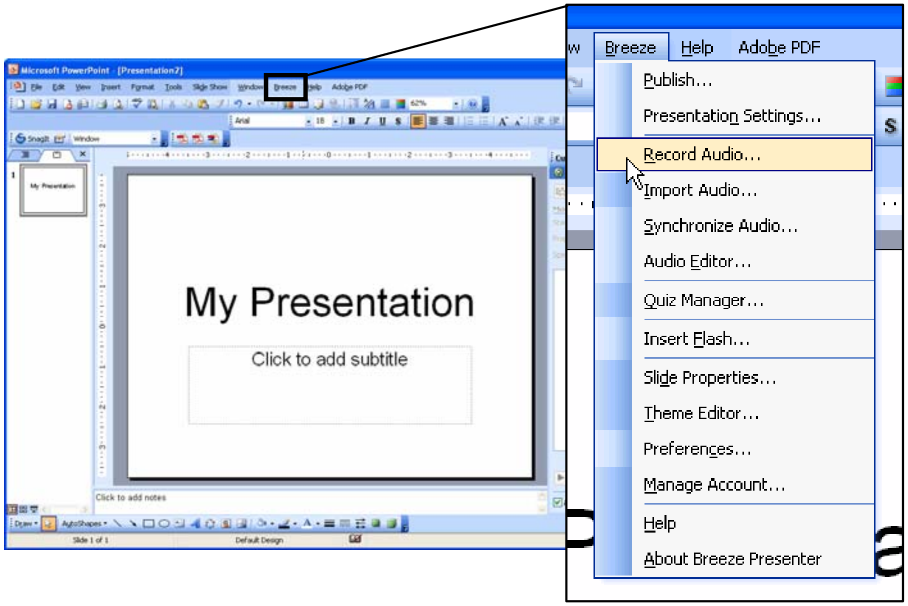
Learning Module

Homework Assignment

Create your own seminars

Easy as 1, 2, 3…

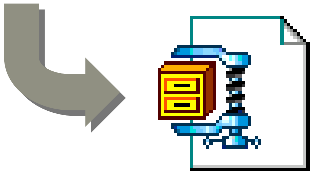
1. Start with your PowerPoint presentation
2. Download our Breeze plug-in and add your voice
3. Have Breeze “publish” the result as a zip file




Record Audio

Slide: 1 of 5 00:00.0 / 00:00.0
Press RECORD to begin recording audio

☐ Record/Play this slide only
☐ View script


Your
Seminar