Sentaurus

Sentaurus is an advanced, industry-standard TCAD (Technology Computer-Aided Design) software suite that allows one to design and simulate transistors. It is used in many research groups and companies.

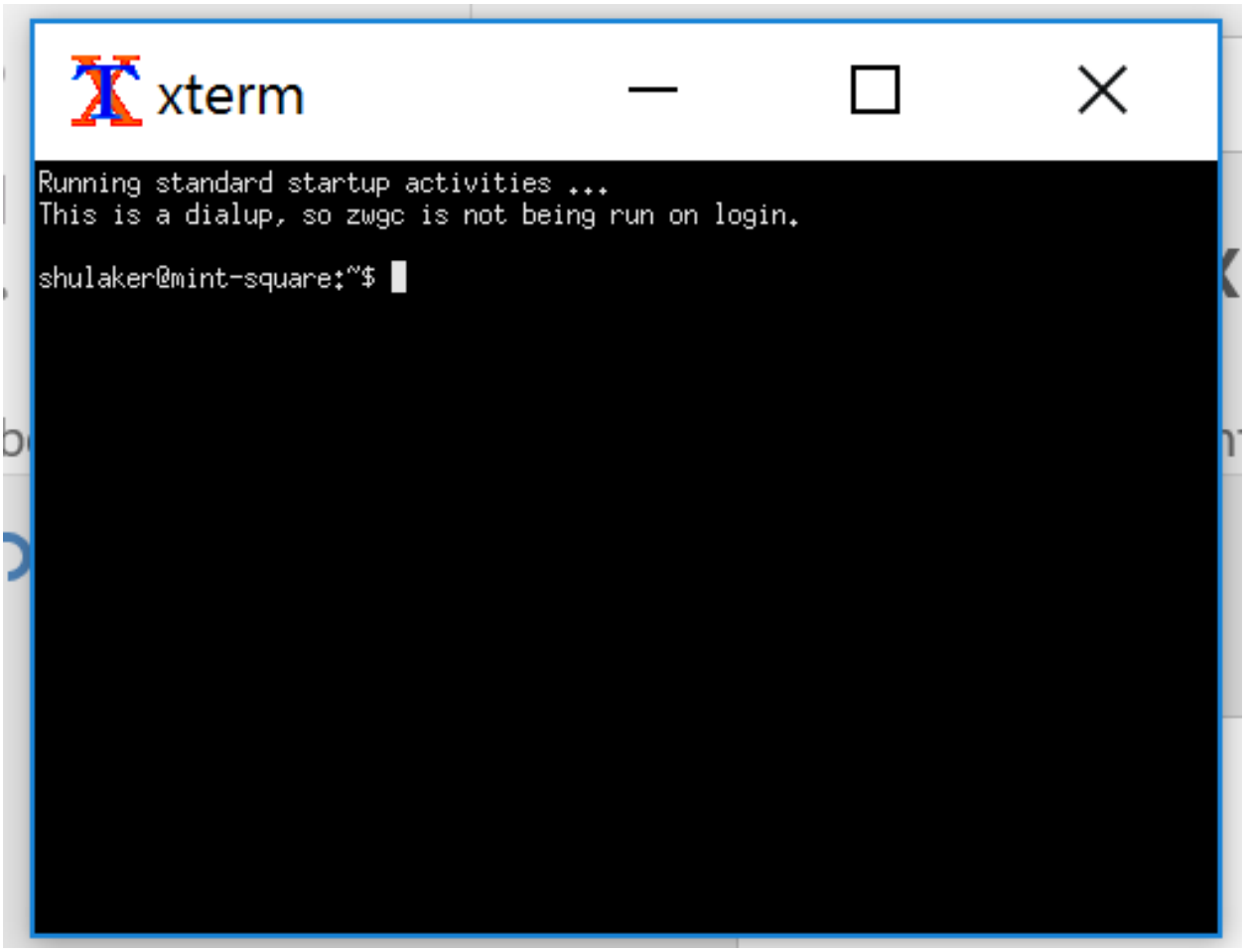
To use Sentaurus, you have to access the Athena computer cluster at MIT. Details are available here:

http://kb.mit.edu/confluence/display/istcontrib/Getting+Started+with+Athena

If you are using windows, you will want to use XWin32: https://ist.mit.edu/xwin32

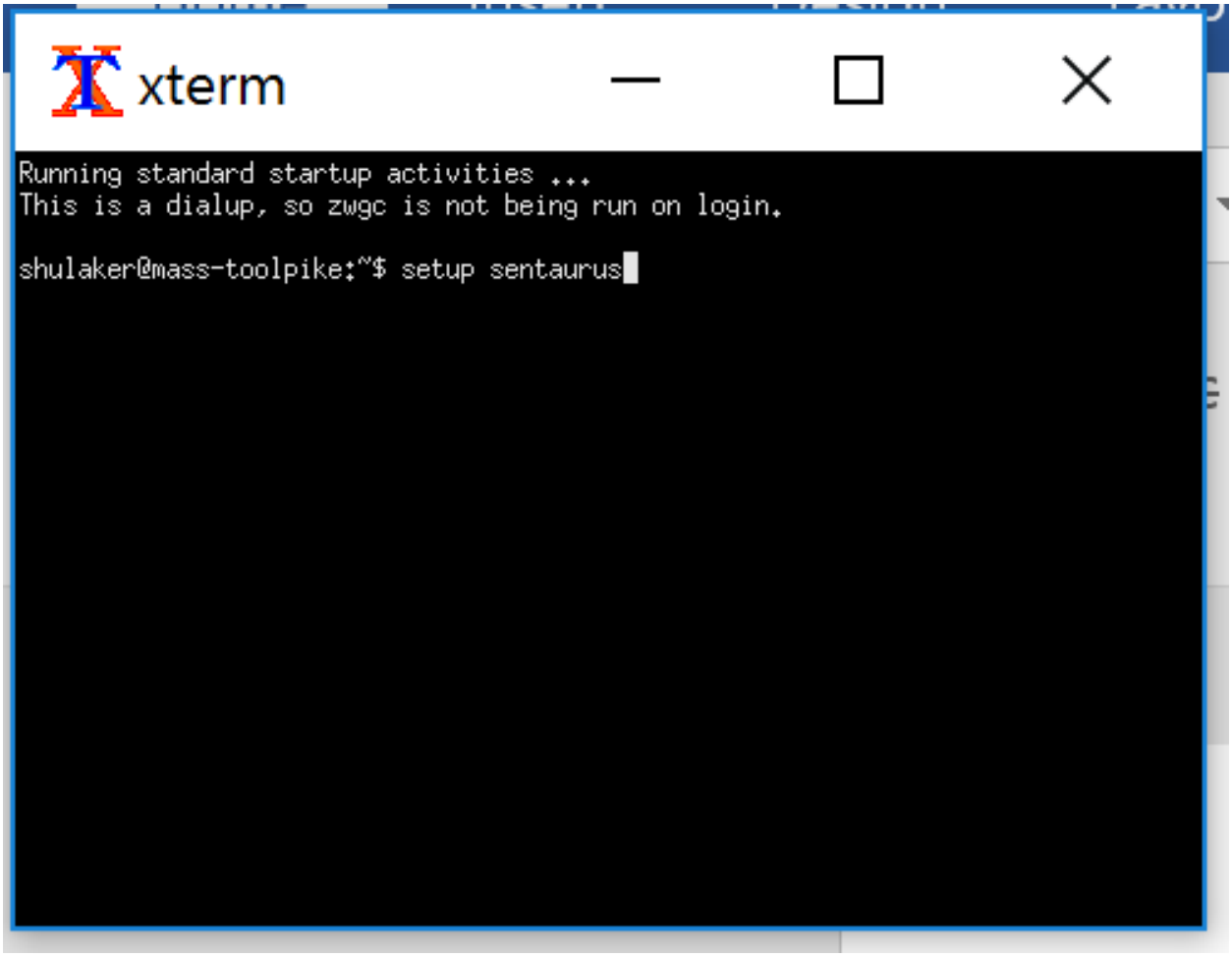
If you are using mac: http://kb.mit.edu/confluence/pages/viewpage.action?pageId=3907239

Once you log onto Athena, it will look like this:



Step1: setup sentaurus

Type in: setup sentaurus



Step2: navigate to folder

Make or navigate to a directory where you want your work to be done in.

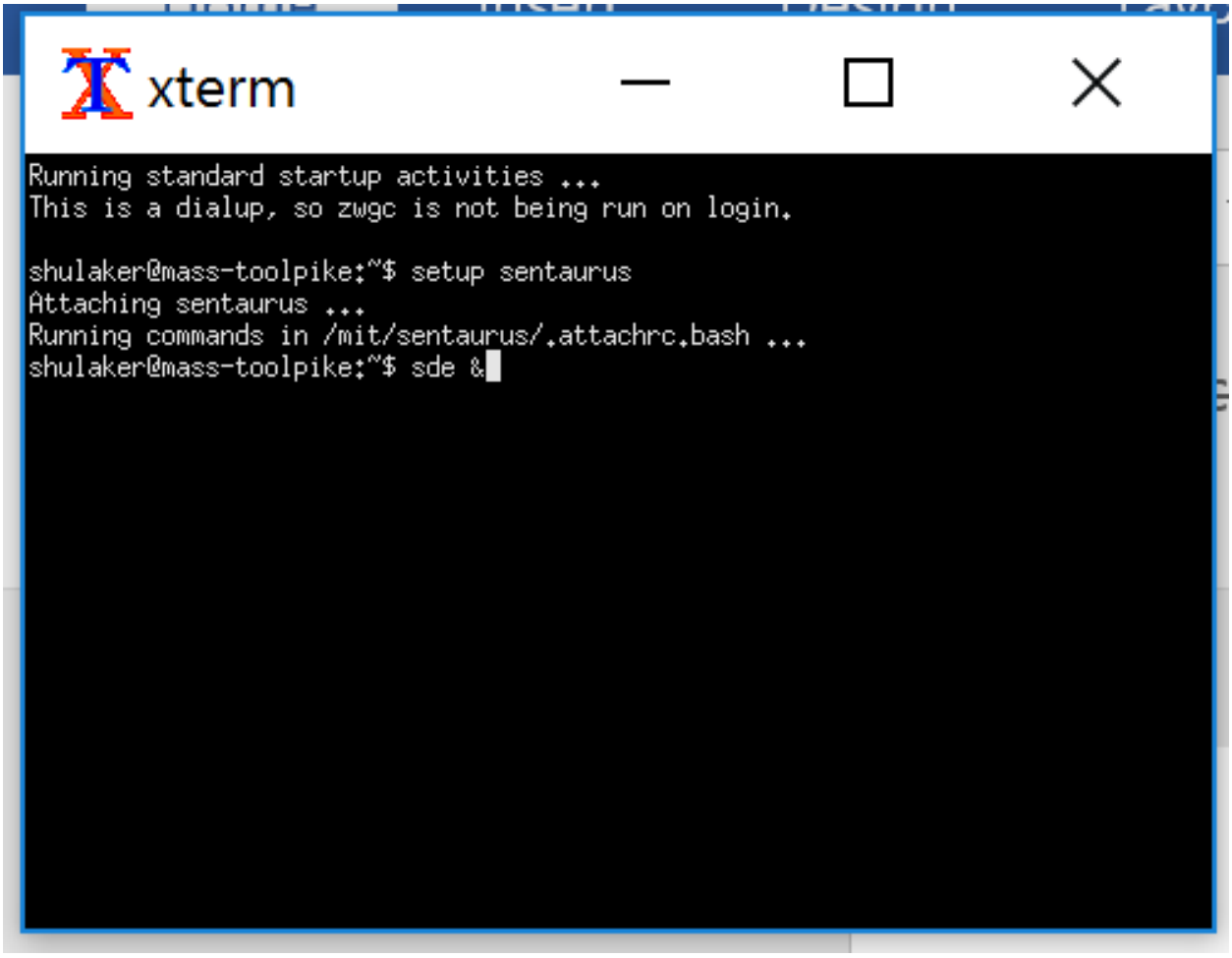
To list your folders, type in: ls

To make a new folder, type in: mkdir FOLDERNAME

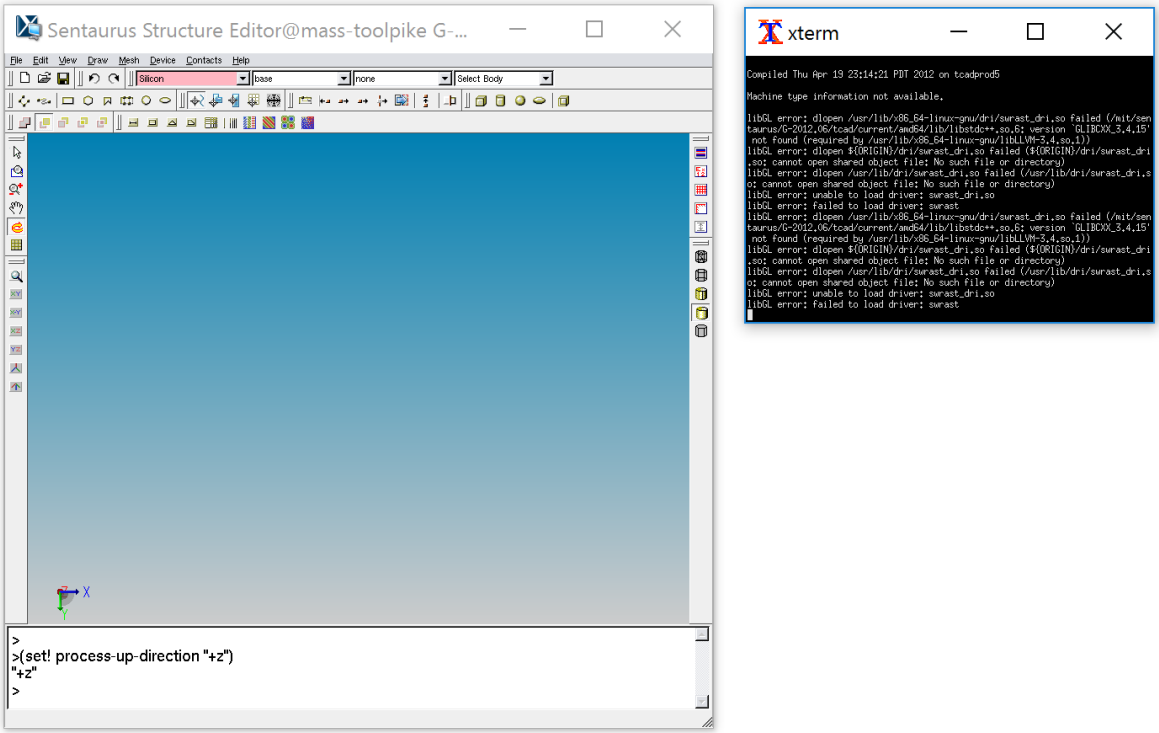
To navigate to a folder, type in: cd FOLDERNAME/

Step3: open sentaurus device editor

Sentaurus device editor is where we define what device we want to simulate. To open it, type is: sde &



The following window will open:

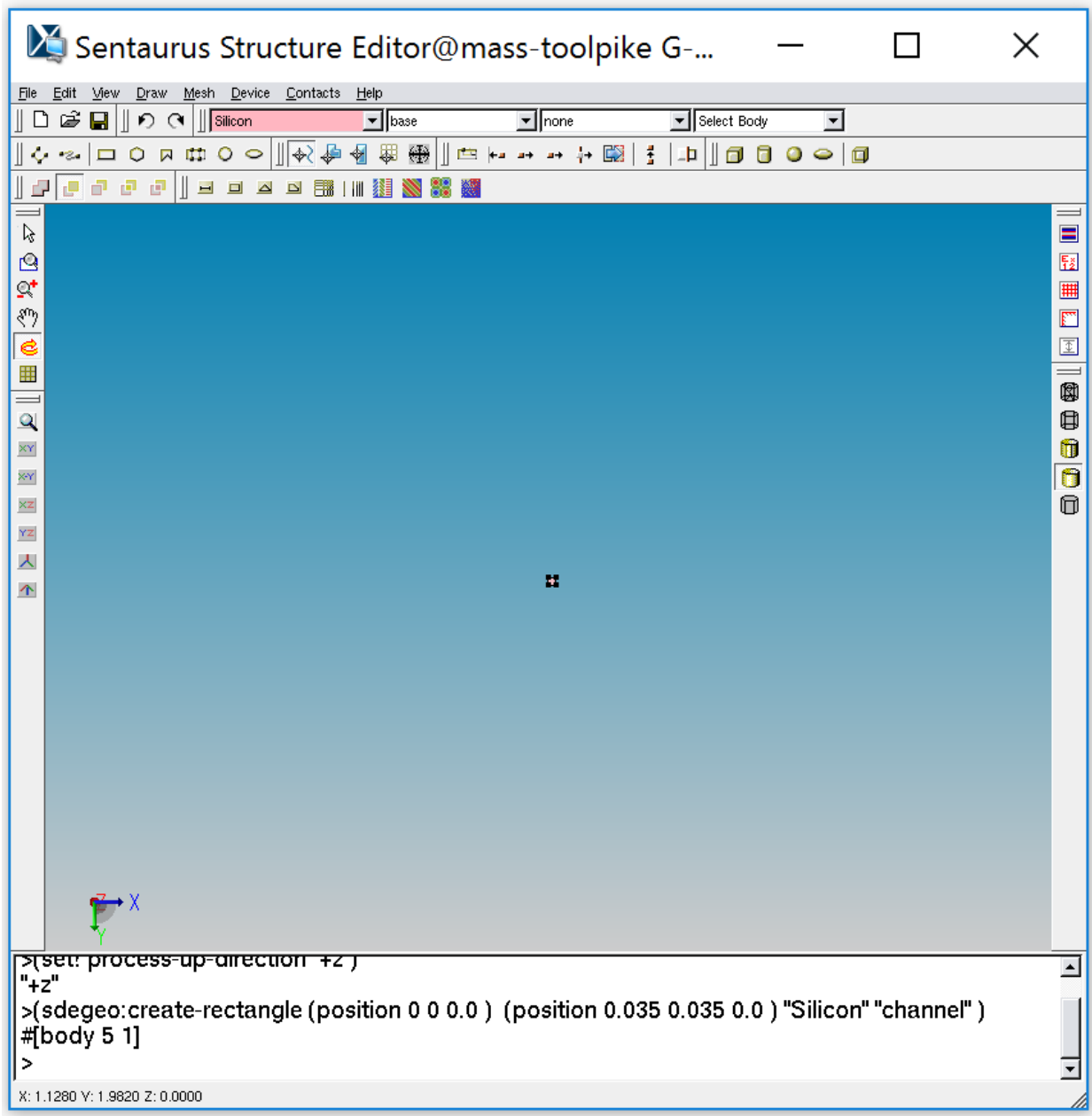


Step4: build the geometries of the device

In this step, you will define the geometry of the device.

To draw a rectangle, you type the following command in the white command prompt line at the bottom of sde:

(sdegeo:create-rectangle (position 0 0 0.0) (position 0.035 0.035 0.0) "Silicon" "channel")



The command does the following:

(sdegeo:create-rectangle (position 0 0 0.0) (position 0.035 0.035 0.0) "Silicon" "channel")

Top left (x y z)

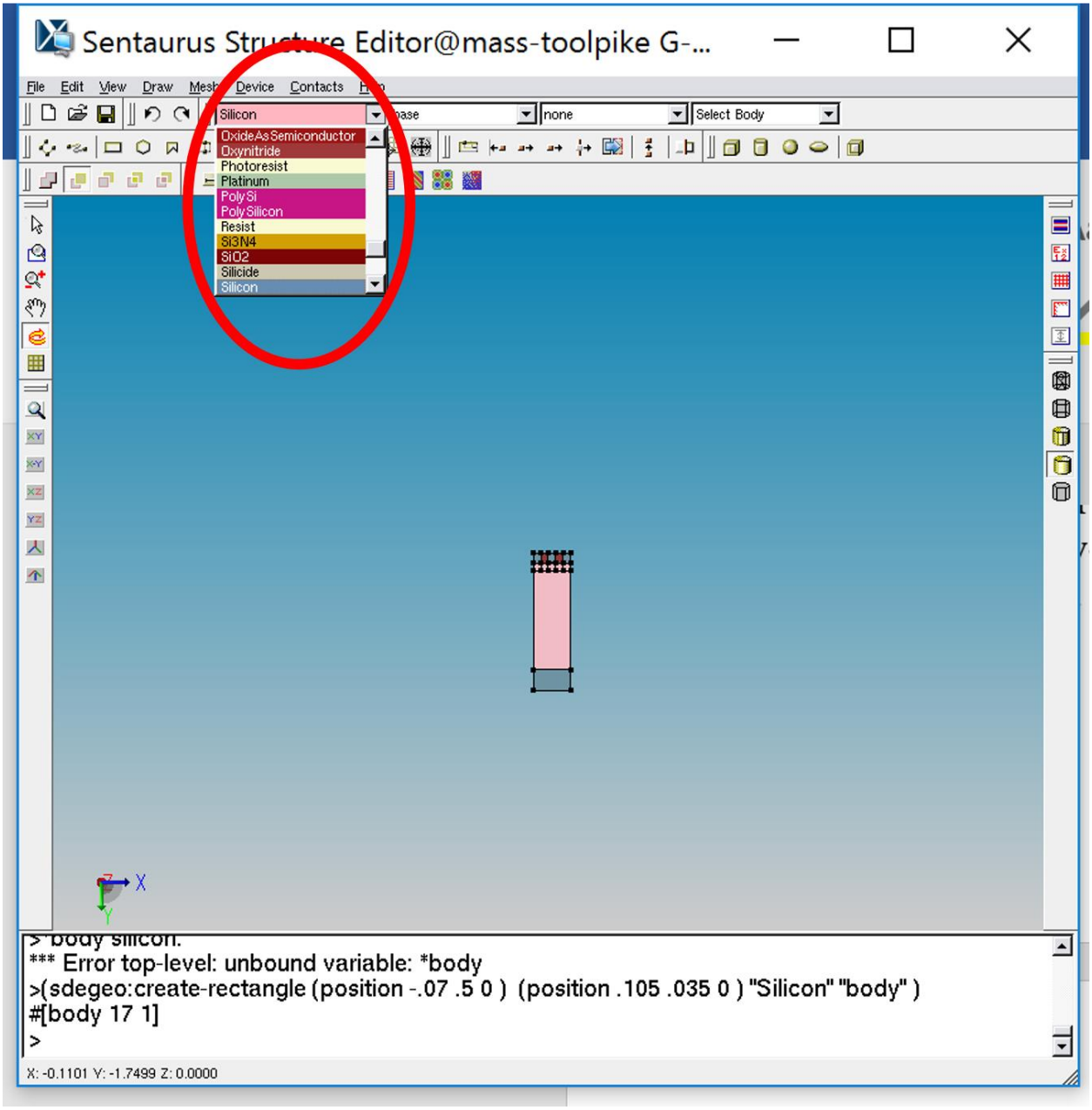
Bottom right (x y z)

Material this part of the device is

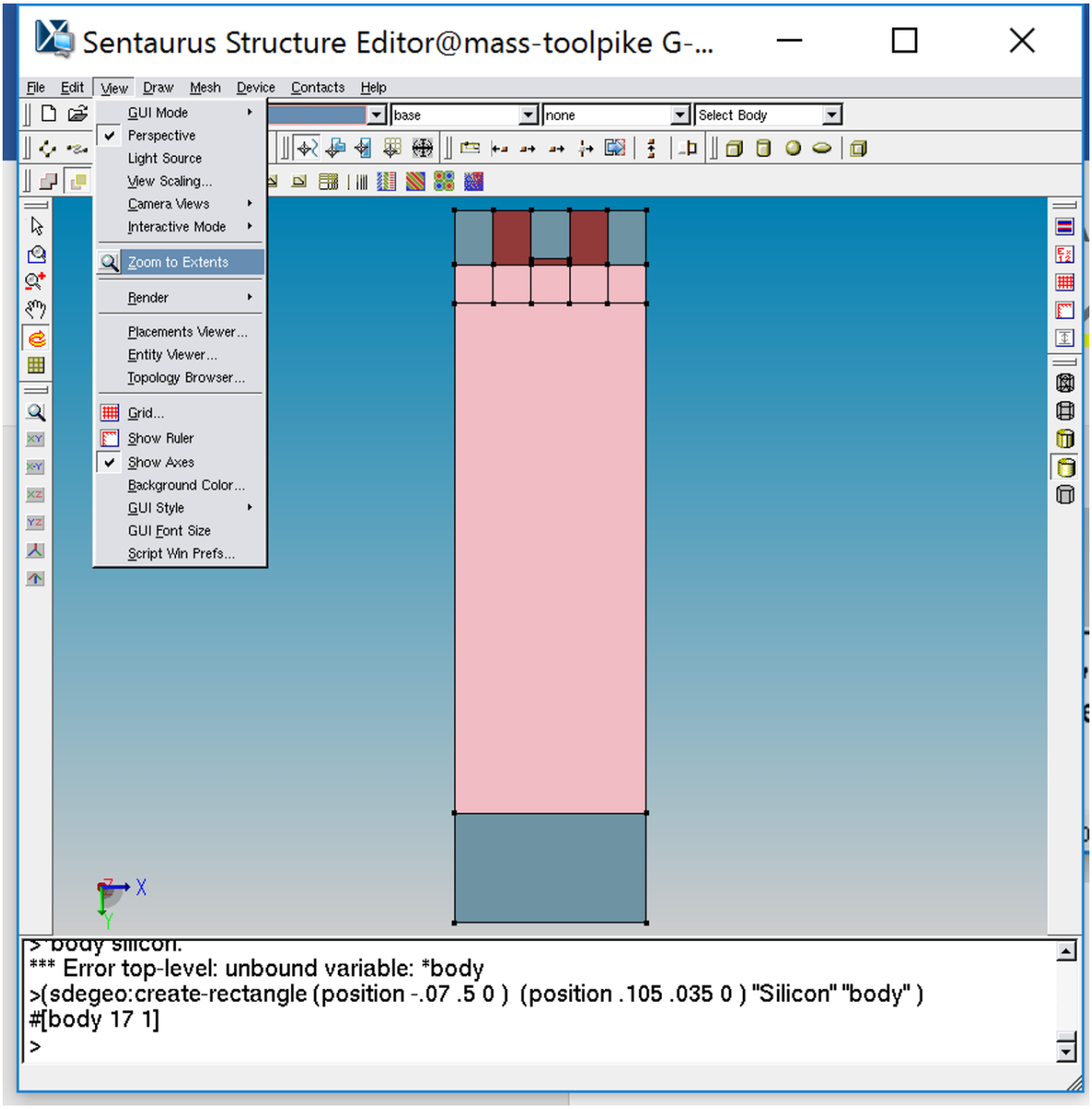
What you want to name it

For instance, the command above defines the silicon channel. To define, the gate oxide, you would write:
(sdegeo:create-rectangle (position 0 0 0) (position 0.035 -0.005 0) "SiO2" "gate_oxide")

A list of different materials you can choose is found here:



Above is also an example of an entire draw transistor. To zoom in to your design, click zoom -> zoom to extents.



The starting transistor geometry is given here:

assign geometries
*silicon channel:
(sdegeo:create-rectangle (position 0 0 0.0) (position 0.035 0.035 0.0) "Silicon" "channel")
*gate oxide:
(sdegeo:create-rectangle (position 0 0 0) (position 0.035 -0.005 0) "SiO2" "gate_oxide")
*Gate metal electrode:
(sdegeo:create-rectangle (position 0 -0.005 0) (position 0.035 -0.05 0) "Aluminum" "gate_electrode")
*spacer oxide between metal gate and metal source:
(sdegeo:create-rectangle (position 0 0 0) (position -0.035 -0.05 0) "SiO2" "spacerL")

*spacer oxide between metal gate and metal drain:
(sdegeo:create-rectangle (position 0.035 0 0) (position 0.07 -0.05 0) "SiO2" "spacerR")
*Drain metal electrode:
(sdegeo:create-rectangle (position 0.07 0 0) (position 0.105 -0.05 0.0) "Aluminum" "drain_electrode")
*Source metal electrode:
(sdegeo:create-rectangle (position -0.07 0 0) (position -0.035 -0.05 0) "Aluminum" "source_electrode")
*Body metal electrode:
(sdegeo:create-rectangle (position -0.07 0.6 0) (position 0.105 0.5 0) "Aluminum" "body_electrode")
*silicon under source metal:
(sdegeo:create-rectangle (position -.07 .035 0) (position -0.035 0 0) "Silicon" "source_n")
*silicon between silicon under source metal and silicon channel, called source extension:
(sdegeo:create-rectangle (position -.035 .035 0) (position 0 0 0) "Silicon" "source_n_ext")
*silicon between silicon under drain metal and silicon channel, called drain extension:
(sdegeo:create-rectangle (position .035 .035 0) (position .07 0 0) "Silicon" "drain_n_ext")
*silicon under drain metal:
(sdegeo:create-rectangle (position .07 .035 0) (position .105 0 0) "Silicon" "drain_n")
*body silicon:
(sdegeo:create-rectangle (position -.07 .5 0) (position .105 .035 0) "Silicon" "body")

Step5: assigning doping profiles

The command for defining the doping is as follows:

(sdedr:define-constant-profile "constant_channel_doping" "BoronActiveConcentration" 3e18)
(sdedr:define-constant-profile-region "constant_channel_doping_placement" "constant_channel_doping" "channel")

The equations above do the following:

Equation 1: (sdedr:define-constant-profile "constant_channel_doping" "BoronActiveConcentration" 3e18)

Arbitrary name defining this doping profile

Doping type

Doping value

Equation 2: (sdedr:define-constant-profile-region "constant_channel_doping_placement" "constant_channel_doping" "channel")

Arbitrary name defining this doping profile placement

Name from equation 1

Name of the rectangle that is getting this doping value, from Step4

To assign all doping values in the transistor:

assign doping
*channel doping:
(sdedr:define-constant-profile "constant_channel_doping" "BoronActiveConcentration" 3e18)
(sdedr:define-constant-profile-region "constant_channel_doping_placement" "constant_channel_doping" "channel")
*drain extension doping:
(sdedr:define-constant-profile "constant_drain_ext_doping" "PhosphorusActiveConcentration" 5e+19)
(sdedr:define-constant-profile-region "constant_drain_ext_doping_placement" "constant_drain_ext_doping" "drain_n_ext")
*drain doping:
(sdedr:define-constant-profile "constant_drain_doping" "PhosphorusActiveConcentration" 5e+19)
(sdedr:define-constant-profile-region "constant_drain_doping_placement" "constant_drain_doping" "drain_n")

```
*source extension doping:
(sdedr:define-constant-profile "constant_source_ext_doping" "PhosphorusActiveConcentration" 5e+19)
(sdedr:define-constant-profile-region "constant_source_ext_doping_placement" "constant_source_ext_doping" "source_n_ext")
*source doping:
(sdedr:define-constant-profile "constant_source_doping" "PhosphorusActiveConcentration" 5e+19)
(sdedr:define-constant-profile-region "constant_source_doping_placement" "constant_source_doping" "source_n")
*body doping:
 (sdedr:define-constant-profile "constant_body_doping" "BoronActiveConcentration" 5e+19)
(sdedr:define-constant-profile-region "constant_body_doping_placement" "constant_body_doping" "body")
```

Step6: assign the areas where you will apply the voltages:

This step defines the rectangles where you will be apply voltages, in order to measure current flowing through the transistor.

You use the following 3 lines of code to define a contact:

```
(sdegeo:define-contact-set "Gate_contact" 4  (color:rgb 1 0 0 ) "##" )
(sdegeo:set-current-contact-set "Gate_contact")
(sdegeo:set-contact-boundary-edges (list (car (find-body-id (position 0.02 -0.04 0)))) "Gate_contact")
```

The equations do the following:

Equation 1: (sdegeo:define-contact-set "Gate_contact" 4 (color:rgb 1 0 0) "##")

Arbitrary name for the contact The color you want it to be

Equation 2: (sdegeo:set-current-contact-set "Gate_contact")

don't worry about this one

Equation 3: (sdegeo:set-contact-boundary-edges (list (car (find-body-id (position 0.02 -0.04 0)))) "Gate_contact")

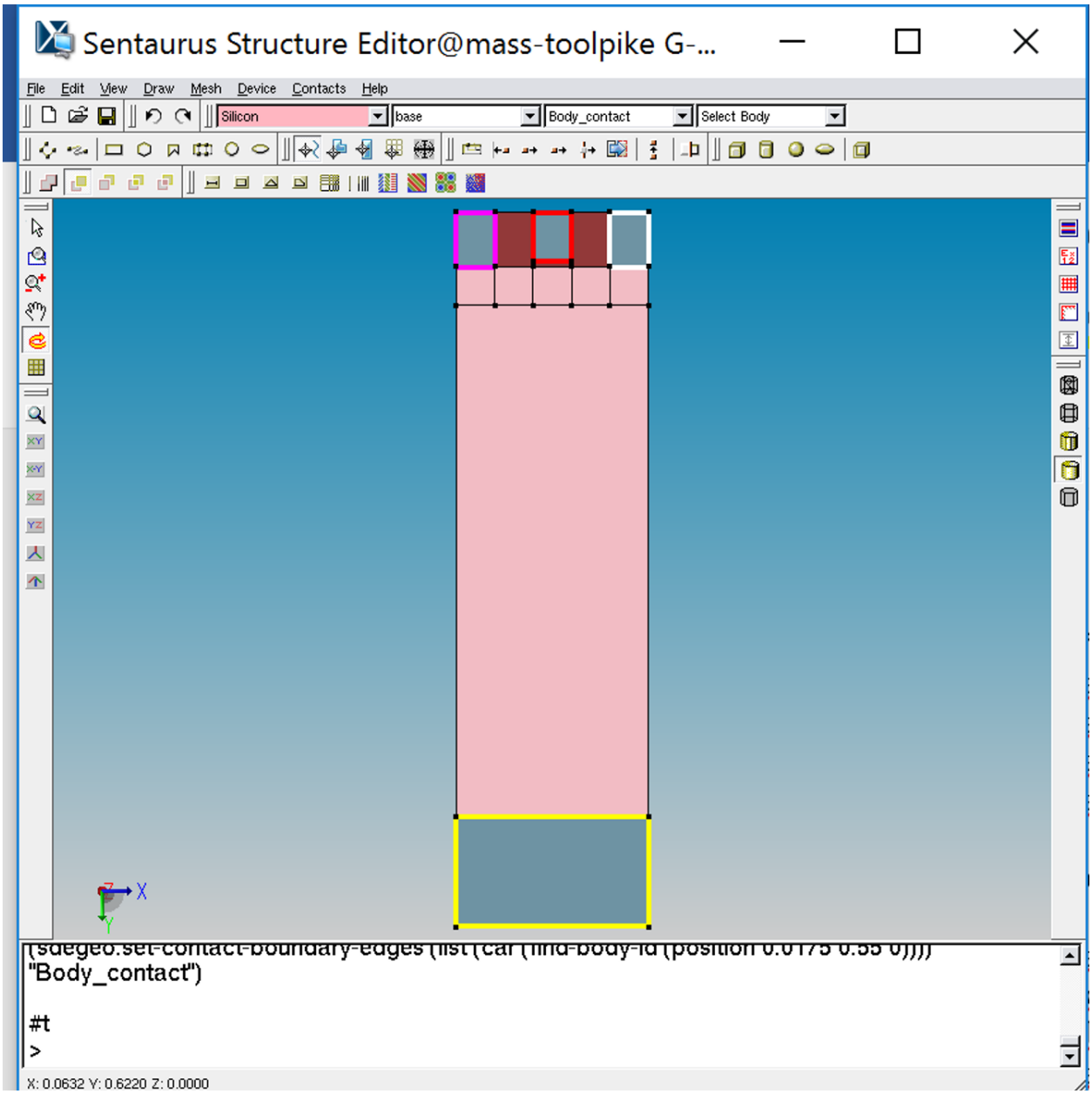
A point within the rectangle that you want the contact applied to The name you gave this contact

The lines for defining the contacts for the starting FET are:

```
(sdegeo:define-contact-set "Gate_contact" 4  (color:rgb 1 0 0 ) "##" )
(sdegeo:set-current-contact-set "Gate_contact")
(sdegeo:set-contact-boundary-edges (list (car (find-body-id (position 0.02 -0.04 0)))) "Gate_contact")
(sdegeo:define-contact-set "Source_contact" 4  (color:rgb 1 0 1 ) "##" )
(sdegeo:set-current-contact-set "Source_contact")
(sdegeo:set-contact-boundary-edges (list (car (find-body-id (position -.06 -0.04 0)))) "Source_contact")
(sdegeo:define-contact-set "Drain_contact" 4  (color:rgb 1 1 1 ) "##" )
(sdegeo:set-current-contact-set "Drain_contact")
(sdegeo:set-contact-boundary-edges (list (car (find-body-id (position 0.1 -0.04 0)))) "Drain_contact")
(sdegeo:define-contact-set "Body_contact" 4  (color:rgb 1 1 0 ) "##" )
(sdegeo:set-current-contact-set "Body_contact")
(sdegeo:set-contact-boundary-edges (list (car (find-body-id (position 0.02 0.55 0)))) "Body_contact")
```

You need to define 4 contacts for your FET: the gate, source, drain, and the bulk (this is the very bottom of the silicon wafer).

And it looks like the following (the contacts are outlined in color):



Step7: make the "mesh":

The mesh defines a grid that the software uses to solve the transistor (e.g., electric field, etc.). Improper grid sizes will cause the simulations to not converge (e.g., if the grid is spaced too close, there will be too many points and the simulation will take too long to run. Set the grid to sparse, and the simulations will not converge to a correct value).

The following lines can be copy and pasted:

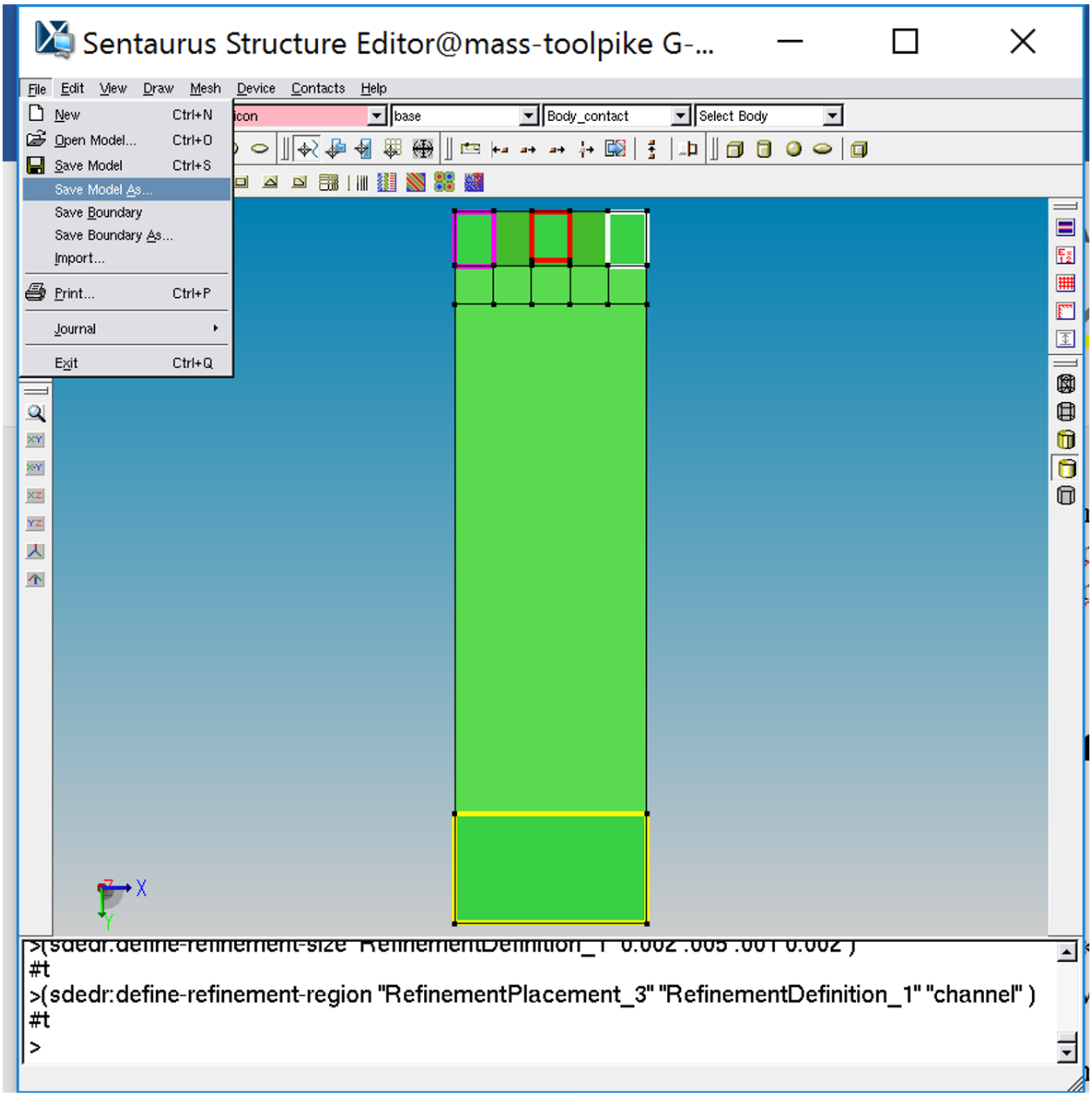
```
***make mesh***
*define and set top mesh, everything above the surface of the silicon:
(sdedr:define-refeval-window "RefWin_1" "Rectangle"  (position -0.07 0 0) (position 0.105 -0.05 0))
(sdedr:define-refinement-size "RefinementDefinition_1" 0.002 0.001 0.005 0.002 )
(sdedr:define-refinement-placement "RefinementPlacement_1" "RefinementDefinition_1" "RefWin_1" )

*define and set the bottom mesh, everything below the surface of the silicon:
(sdedr:define-refeval-window "RefWin_2" "Rectangle"  (position -0.07 .6 0) (position 0.105 0 0))
(sdedr:define-multibox-size "MultiboxDefinition_1" 0.05 0.03 .03 .0002 1 1.35 )
(sdedr:define-multibox-placement "MultiboxPlacement_1" "MultiboxDefinition_1" "RefWin_2" )

*define and set the mesh over the channel:
(sdedr:define-refinement-size "RefinementDefinition_1" 0.002 .005 .001 0.002 )
(sdedr:define-refinement-region "RefinementPlacement_3" "RefinementDefinition_1" "channel" )
```

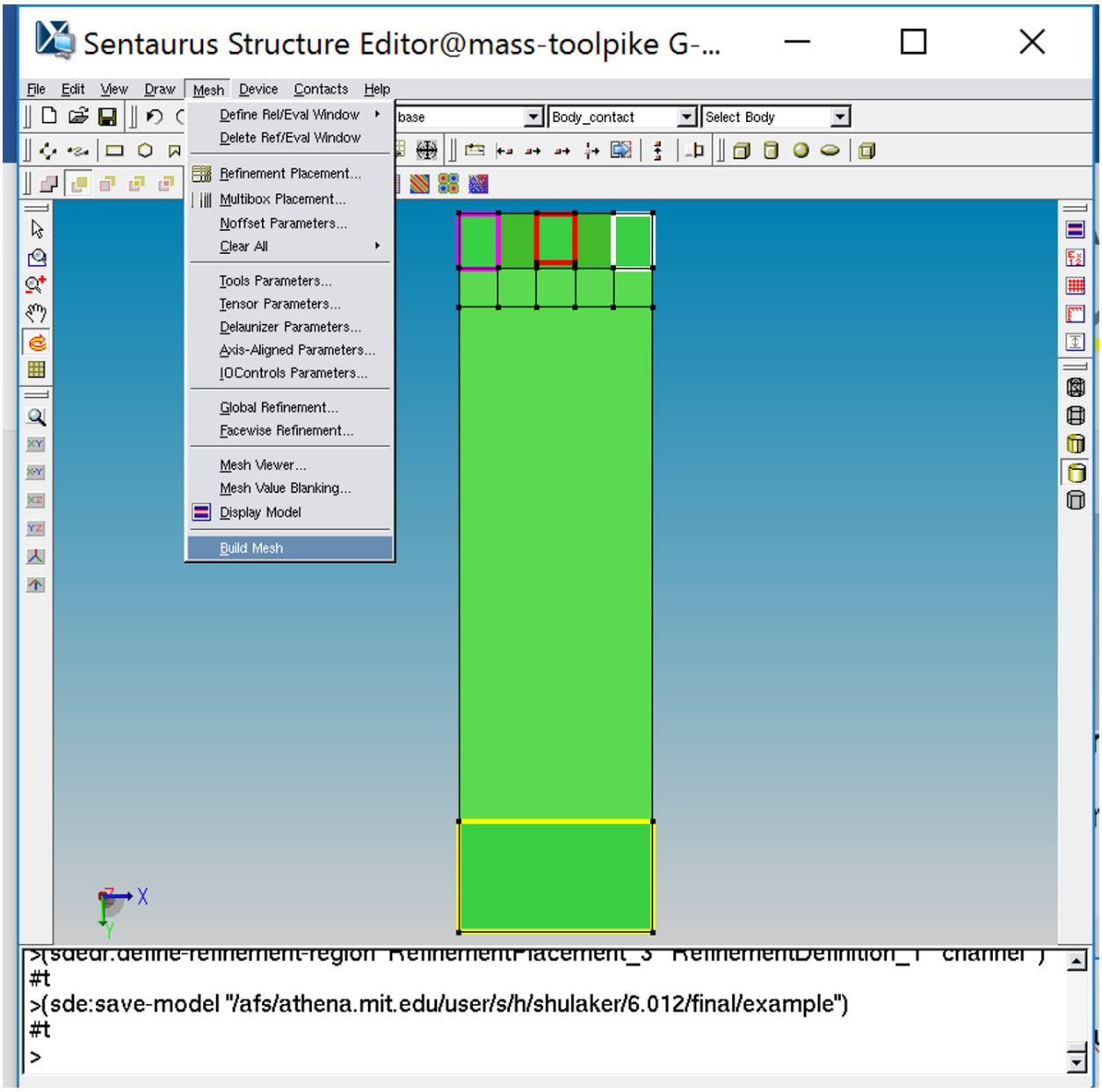
Step8: save the file

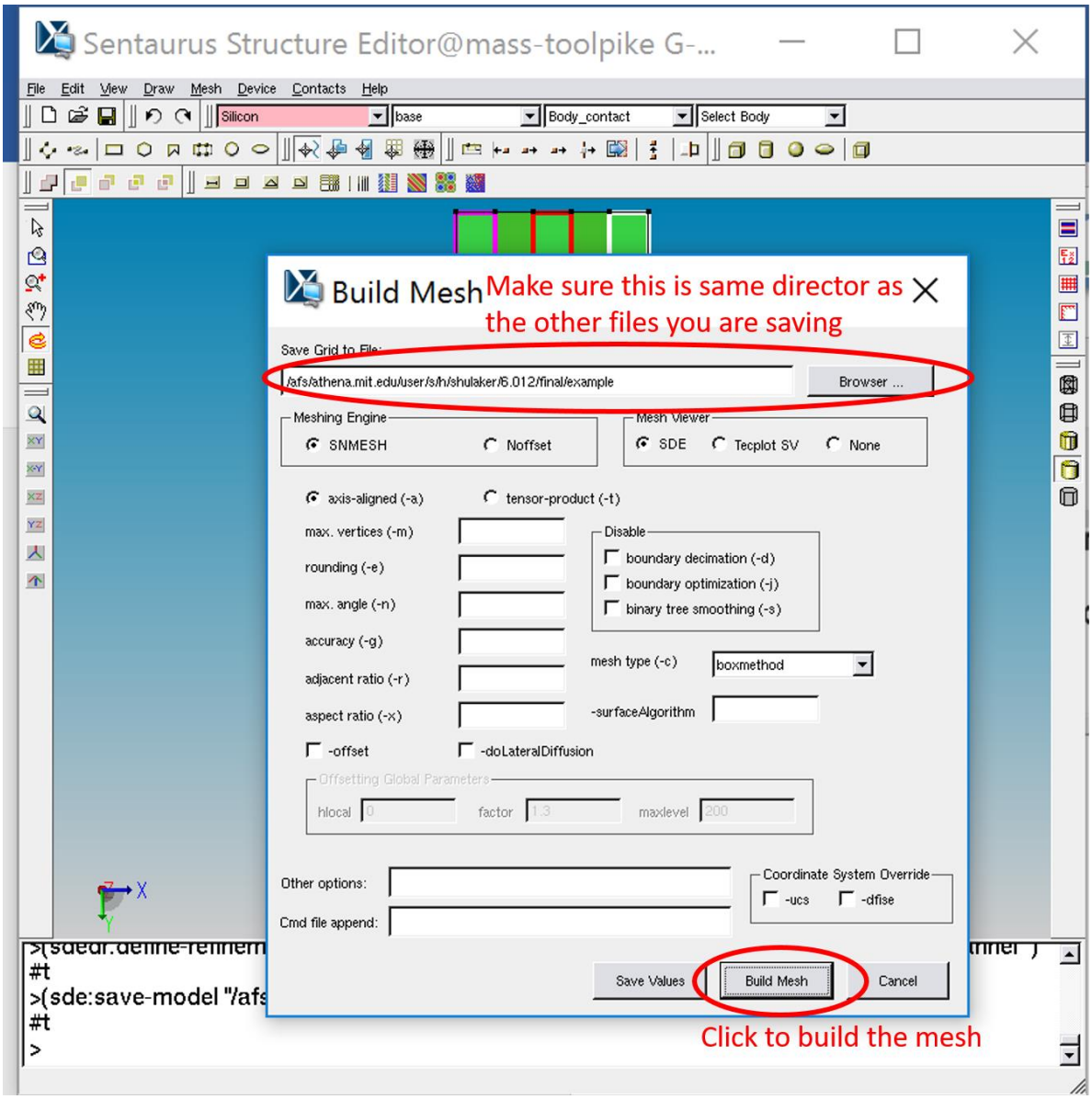
(1) Save the file. Don’t worry about an error if a message pops up. Remember what you name this.



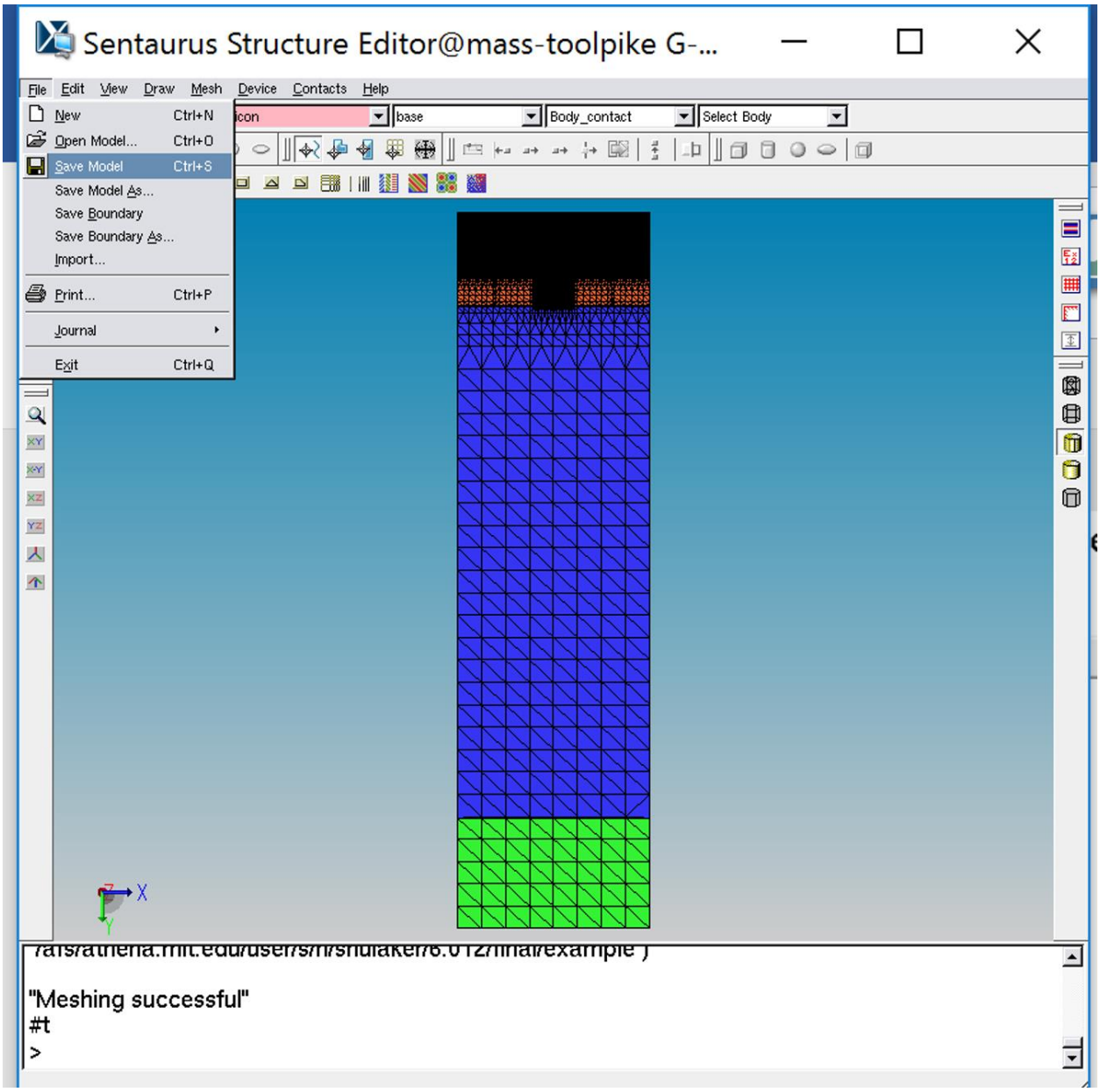
(2) Have sentaurus physically build the mesh you defined previously. This can take a while, be patient:

Mesh ➔ build mesh


Sentaurus Structure Editor@mass-toolpike G-...
File Edit View Draw Mesh Device Contacts Help
Define Ref/Eval Window
Delete Ref/Eval Window
Refinement Placement...
Multibox Placement...
Noffset Parameters...
Clear All
Tools Parameters...
Tensor Parameters...
Delaunizer Parameters...
Axis-Aligned Parameters...
IOControls Parameters...
Global Refinement...
Facewise Refinement...
Mesh Viewer...
Mesh Value Blanking...
Display Model
Build Mesh
base
Body_contact
Select Body
#t
>(sde:save-model "/afs/athena.mit.edu/user/s/h/shulaker/6.012/final/example")
#t
>



(3) Save the final files:



Above is what the final structure should look like after meshing.

Step9: update the file which sets up the simulation

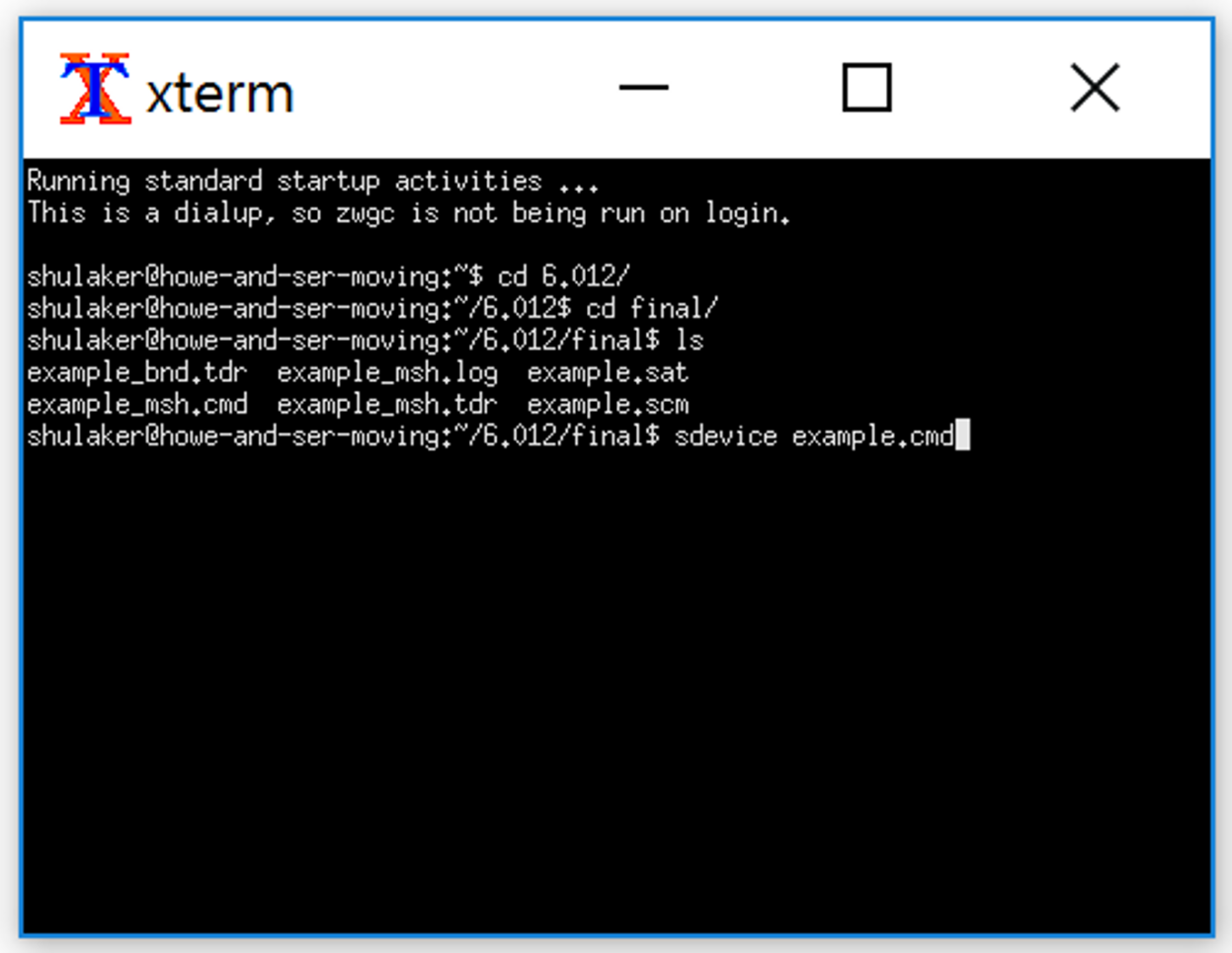
The file below is given as a text file, and is posted online. It is called “simulation_control.txt.” To execute it, you have to rename it as “simulation_control.cmd.” It defines the sweep ranges for the voltages, what physics the model should include/ ignore, etc. The file is given below. Make sure you do the following:

- Ensure you make the FILENAME match whatever you saved your transistor model under.
- Set the drain voltage and maximum gate voltage to the supply voltage you have chosen.
- Save the file in the same folder as all of your other files

```
File {
* The File section defines the input and output files of the simulation
* Input Files
Grid = "FILENAME_msh.tdr"   *fix the FILENAME
Current = "FILENAME_def.plt"
Plot = "FILENAME_def.tdr"
Output = "FILENAME_def.log"
}
Electrode {
{ Name="Drain_contact" Voltage=1.8 } * CHANGE THIS TO SET VDD
{ Name="Source_contact" Voltage=0.0 }
{ Name="Gate_contact" Voltage=0.0 }
{ Name="Body_contact" Voltage=0.0 }
}
Physics {
Mobility( DopingDep HighFieldSat Enormal )
EffectiveIntrinsicDensity ( OldSlotBoom )
}
Plot {
eDensity hDensity eCurrent hCurrent
Potential SpaceCharge ElectricField
eMobility hMobility eVelocity hVelocity
Doping DonorConcentration AcceptorConcentration
}
Math {
Extrapolate
RelErrControl *on by default
Iterations=50
Notdamped=100
}
Solve {
Coupled(Iterations=100){ Poisson }
Coupled{ Poisson Electron Hole }
*-Bias Cathode to target bias
Quasistationary(
InitialStep=0.001 Increment=1.1
MinStep=1e-5 MaxStep=0.05
Goal{ Name="Gate_contact" Voltage=1.8 } *CHANGE THIS TO SET VDD
){ Coupled{ Poisson Electron Hole }}
}
```

Step10: run the simulation

In the command prompt, type in: sdevice simulation_control.cmd

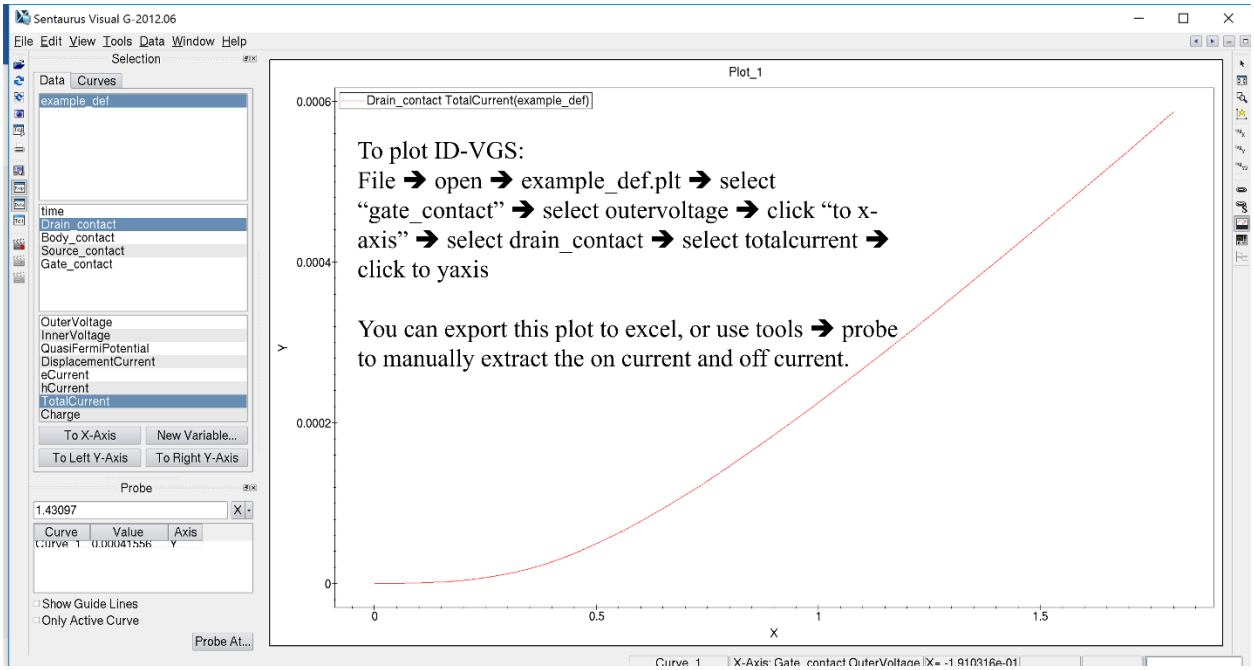


You will see the simulation running, and it will display any convergence errors, etc. in the command prompt.

Step11: extract key device metrics from generated IV curves

In command prompt, type in “svisual” to open the software to plot the IV curves.

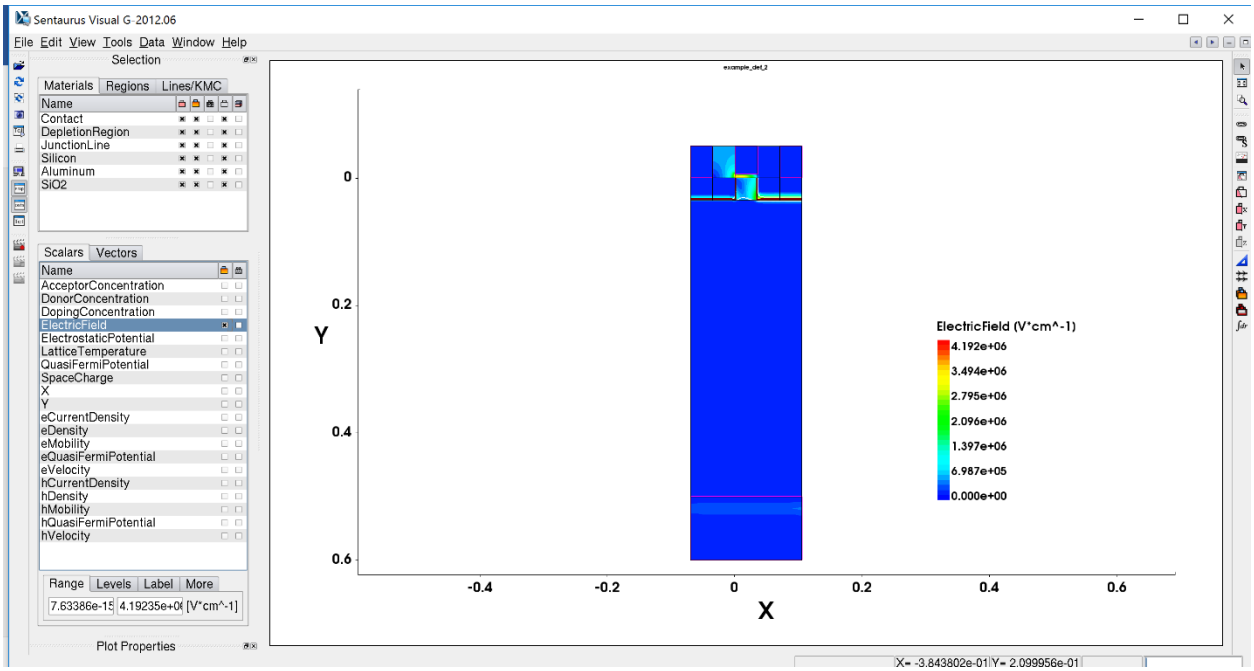
In svisual, you can do the following:



To view different aspects of the device (space charge, electric field, voltage distributions across the channel, etc.):

File ➔ open ➔ example_def.tde ➔ simple click on the parameter you want to view. Here, I clicked on electricfield, and you can see the electric field near the drain end.

Importantly: in your simulation_command.cmd file, you set up the I_D-V_{GS} sweep. This plot is only for the very final point in the sweep. Thus, this plot is when V_{DS}=V_{DD}, and V_{GS}=V_{DD}. If I wanted to look at what the device looks like when it is off, I would set V_{DS} to V_{DD}, and the maximum V_{GS} in the sweep not to V_{DS}, but to something like 0.05 V.


Sentaurus Visual G-2012.06
File Edit View Tools Data Window Help
Selection
Materials Regions Lines/KMC
Name
Contact
DepletionRegion
JunctionLine
Silicon
Aluminum
SiO2
Scalars Vectors
Name
AcceptorConcentration
DonorConcentration
DopingConcentration
ElectricField
ElectrostaticPotential
LatticeTemperature
QuasiFermiPotential
SpaceCharge
X
Y
eCurrentDensity
eDensity
eMobility
eQuasiFermiPotential
eVelocity
hCurrentDensity
hDensity
hMobility
hQuasiFermiPotential
hVelocity
Range Levels Label More
7.63386e-15
4.19235e+06
[V*cm^-1]
Plot Properties
Y
X
ElectricField (V*cm^-1)
4.192e+06
3.494e+06
2.795e+06
2.096e+06
1.397e+06
6.987e+05
0.000e+00
X= -3.843802e-01 Y= 2.099956e-01