Table of Contents

Coupling QuantumATK with Synopsys tools

Version: 2016.3

Downloads & Links

- PDF
- ATK Reference Manual
- Basic QuantumATK Tutorial

In this tutorial you will learn how to extract band structure data from a first-principles (DFT) calculation in a file format that can be directly imported in **Synopsys** tools for **TCAD simulations**. We will use a simple silicon crystal as the example, but the methodology is general and can in principle be applied to any semiconductor material, subjected to for instance strain, or even a nanowire or 2D thin film.

This tutorial is aimed both at experienced users and those with more limited experience, who are trying this functionality for the first time. In order to keep the tutorial short and succinct, we will not explain so much, but just present the steps you need to follow. It is however a good idea to go through the Basic QuantumATK Tutorial to learn the basics of **QuantumATK** (ATK) and its graphical user interface **QuantumATK** (**QuantumATK**).

If you are familiar with **QuantumATK**, you may skip the following section - just start **QuantumATK** (in any project) and proceed to the next section to install the **addon** needed for this tutorial.

Preparations

You will of course first need to download and install the software (there is both a Linux and Windows version).

Start **QuantumATK**, typically by double-clicking the desktop icon. If this is the first time you try **QuantumATK** and you don't have a license, the software will automatically attempt to download a demo license over the internet - naturally you need to be online for this to work. For alternative methods to obtain a license, see the license to apply for an extended trial license.

Spend just a few minutes to familiarize yourself with the program.

First, a project selection dialog will appear. If this is your first encounter with **QuantumATK**, it is recommended to first open the *Example Project*, to have some data to play around with.

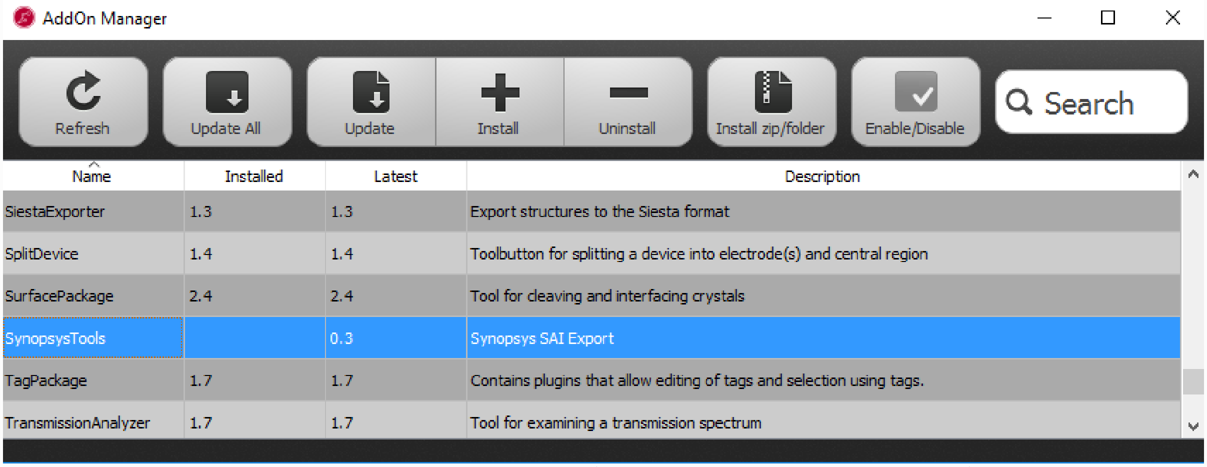
- In the left panel you see a list of files in the project. By checking the little box next to a file name, the contents of the file will be displayed on the **LabFloor**, the middle panel, provided that the file type is understood by QuantumATK (additional file types can be supported via plugins).

- By selecting an object icon on the **LabFloor**, a set of tools on the right-hand toolbar panel will become active, indicating possible ways you can interact with this object by clicking the corresponding panel button. For instance, a band structure can be opened in the Bandstructure Analyzer.
- Some objects - in particular atomic configurations - can also be dragged with the mouse to e.g. the **Viewer** to show them in 3D, or the **Builder** for manipulations of the geometry.

Installing the addon

To export band structures to the **Synopsys SAI** format, you need to install the **SynopsysTools** addon in QuantumATK.

Open the **AddOn Manager** from the **Help** menu. Scroll down to “SynopsysTools” (or type “syn” in the search box) - as you will see, this addon is not installed, but is available on the server (the version number may vary).



Name	Installed	Latest	Description
SiestaExporter	1.3	1.3	Export structures to the Siesta format
SplitDevice	1.4	1.4	Toolbutton for splitting a device into electrode(s) and central region
SurfacePackage	2.4	2.4	Tool for cleaving and interfacing crystals
SynopsysTools		0.3	Synopsys SAI Export
TagPackage	1.7	1.7	Contains plugins that allow editing of tags and selection using tags.
TransmissionAnalyzer	1.7	1.7	Tool for examining a transmission spectrum

Select the SynopsysTools package and click **Install** (alternatively, just click **Update all**) to download and install it.

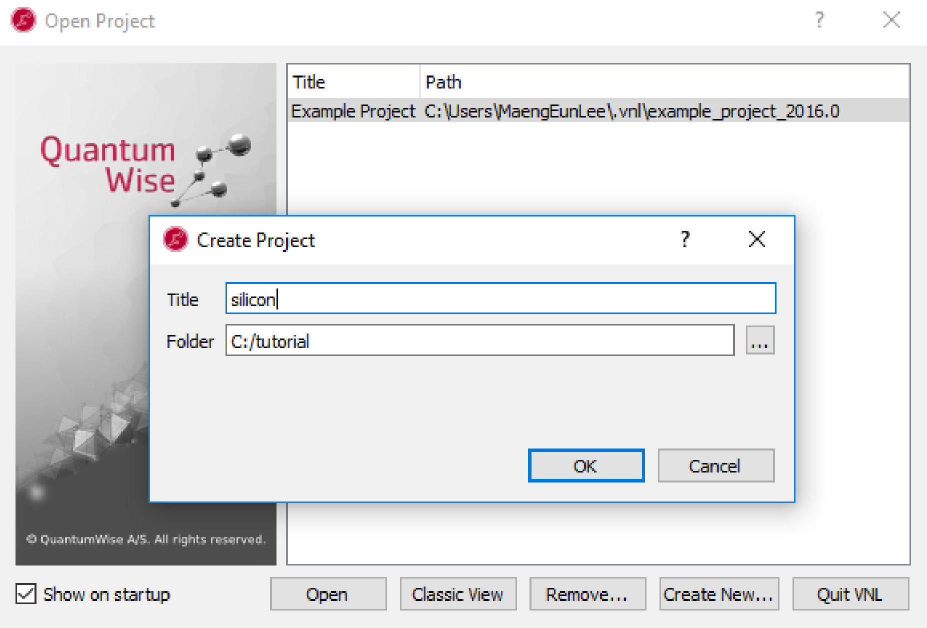
Note

You now need to restart QuantumATK for the plugin to become active.

New project

When **QuantumATK** starts *again*, make a new project for this tutorial.

1. Create a new project by clicking **Create new** in the dialog shown on startup.
2. Choose a suitable directory (that does not contain any non-Latin characters in the path).
3. Then click **Open** to start the main interface.



Open Project

Title	Path
Example Project	C:\Users\MaengEunLee\.vnl\example_project_2016.0

Create Project

Title: silicon

Folder: C:/tutorial

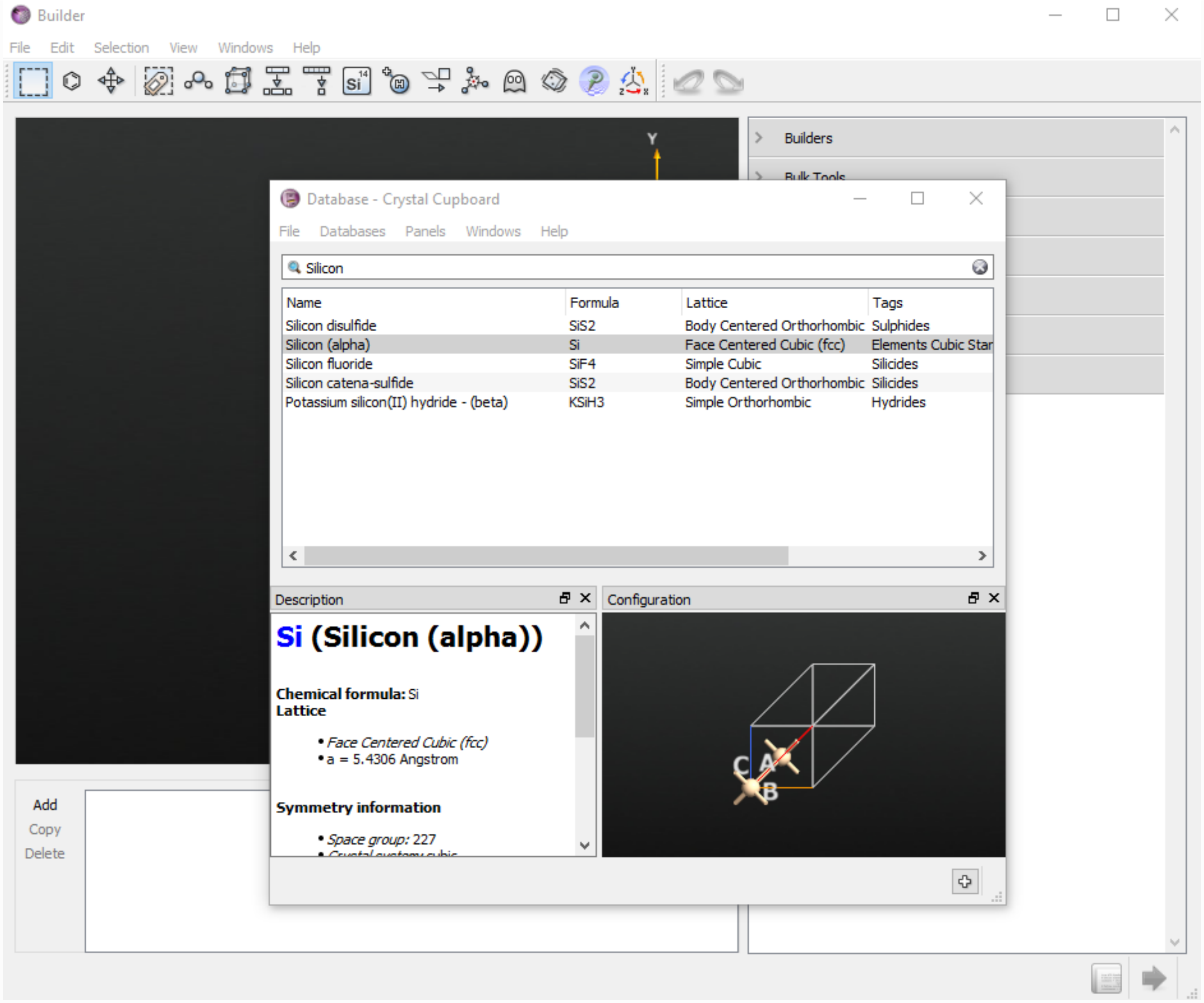
Silicon crystal

Click the **Builder** to open the atomic geometry builder.

Note

The **Builder** contains a lot of advanced tools for constructing and manipulating atomic-scale structures, but for this tutorial you will just use a predefined crystal from a database.

1. Click the button **Add** to the left of the Stash area and choose **From Database...** from the popup menu.
2. Locate the “Silicon (alpha)” structure - for instance by using the search filter - and select it.
3. Click the ✚ plus button to add the silicon crystal to the **Stash**. The database window will automatically close.
4. Now click the icon in the lower right-hand corner of the Builder, and choose **Script Generator** from the popup menu. This will transfer the geometry to the tool used for setting up the calculation model.


Builder
File Edit Selection View Windows Help
Builders
Bulk Tools
Y
Add
Copy
Delete
Database - Crystal Cupboard
File Databases Panels Windows Help
Silicon
Description
Configuration
C
A
B

Name	Formula	Lattice	Tags
Silicon disulfide	SiS2	Body Centered Orthorhombic	Sulphides
Silicon (alpha)	Si	Face Centered Cubic (fcc)	Elements Cubic Star
Silicon fluoride	SiF4	Simple Cubic	Silicides
Silicon catena-sulfide	SiS2	Body Centered Orthorhombic	Silicides
Potassium silicon(II) hydride - (beta)	KSiH3	Simple Orthorhombic	Hydrides

Si (Silicon (alpha))

Chemical formula: Si
Lattice

- *Face Centered Cubic (fcc)*
- a = 5.4306 Angstrom

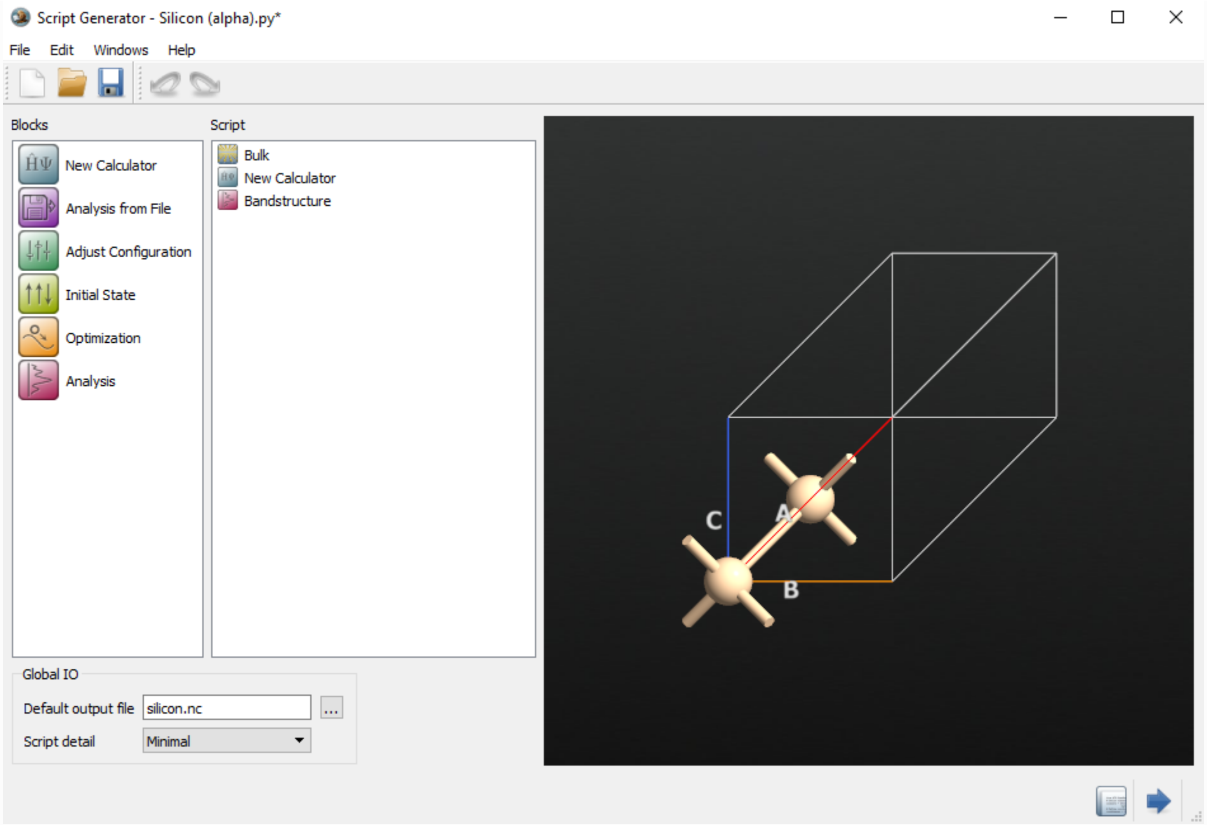
Symmetry information

- *Space group:* 227
- *Crystal system:* cubic

DFT model setup

In the **Scripter** change the following:

1. double-click **New calculator** in the “Blocks” panel;
2. double-click **Analysis** and choose **Bandstructure** from the popup menu;
3. simplify the name of the default output file to `silicon.hdf5`. This is where all results of the calculation will be stored.



Bulk
New Calculator
Bandstructure

Global IO

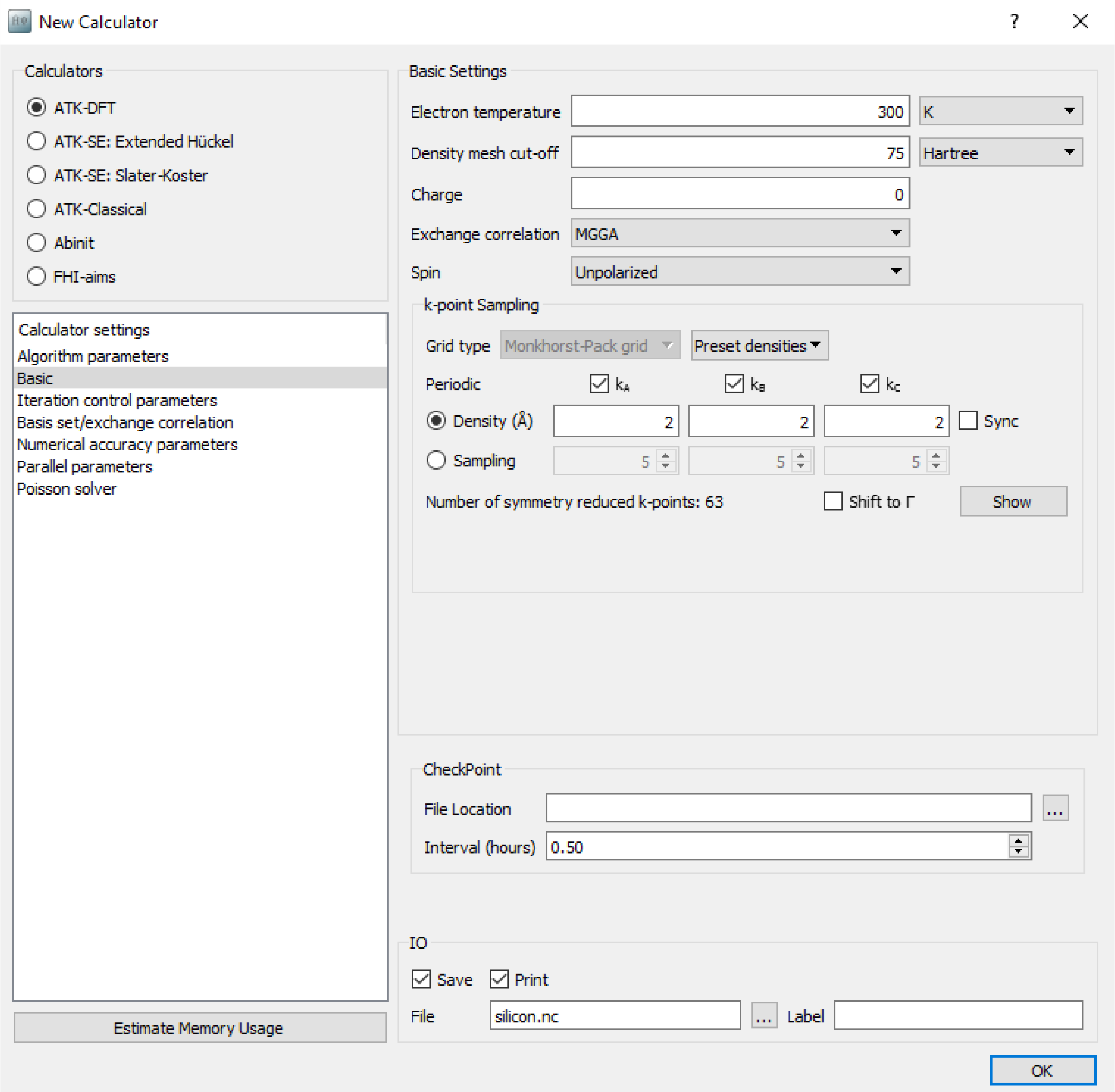
Default output file: silicon.nc

Script detail: Minimal

Double-click the **New calculator** block that was inserted in the Script panel. In this tutorial, you will only change the k-point sampling and exchange-correlation.

- Set the k-point sampling as the 2 Å density or the number of k-points to 5 in all three directions (A, B, C)
- Choose the MGGA exchange correlation

Click OK to close the window.



New Calculator

Calculators

● ATK-DFT
○ ATK-SE: Extended Hückel
○ ATK-SE: Slater-Koster
○ ATK-Classical
○ Abinit
○ FHI-aims

Calculator settings
- Algorithm parameters
  - Basic
  - Iteration control parameters
  - Basis set/exchange correlation
  - Numerical accuracy parameters
  - Parallel parameters
  - Poisson solver

Estimate Memory Usage

Basic Settings

Electron temperature: 300 K
Density mesh cut-off: 75 Hartree
Charge: 0
Exchange correlation: MGGA
Spin: Unpolarized

k-point Sampling

Grid type: Monkhorst-Pack grid | Preset densities
Periodic: ☑ k_A ☑ k_B ☑ k_C
● Density (Å): 2 2 2 □ Sync
○ Sampling: 5 5 5
Number of symmetry reduced k-points: 63 □ Shift to Γ | Show

CheckPoint

File Location:
Interval (hours): 0.50

IO

☑ Save ☑ Print
File: silicon.nc | Label:

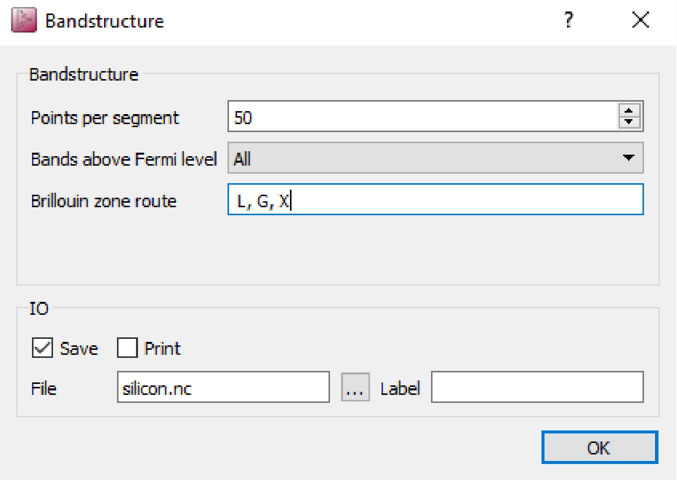
OK

Note

Parameters like the k-point sampling and a few other ones should always be systematically checked for convergence, to obtain accurate results.

Next, double-click the **Bandstructure** block and set the route in the Brillouin zone to L,G,X, and the number of points to 50, to get a detailed picture of the most important part of the band structure.

Click OK to close the window.



Bandstructure

Bandstructure

Points per segment: 50

Bands above Fermi level: All

Brillouin zone route: L, G, X

IO

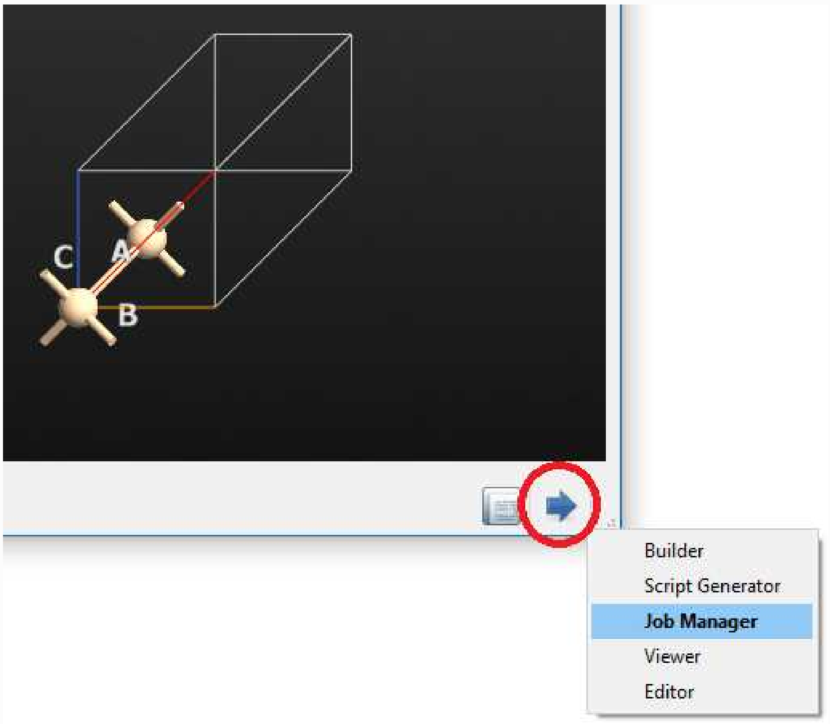
☑ Save ☐ Print

File: silicon.nc

Label:

Click the ↘ icon in the lower right-hand corner of the **Scripter**, and choose **Job Manager**.

A file dialog will pop up, prompting you to save the input file. Choose a suitable filename, such as `silicon.py` (the file is a python script, so always use the .py extension). Make sure the file is saved in the current project directory.

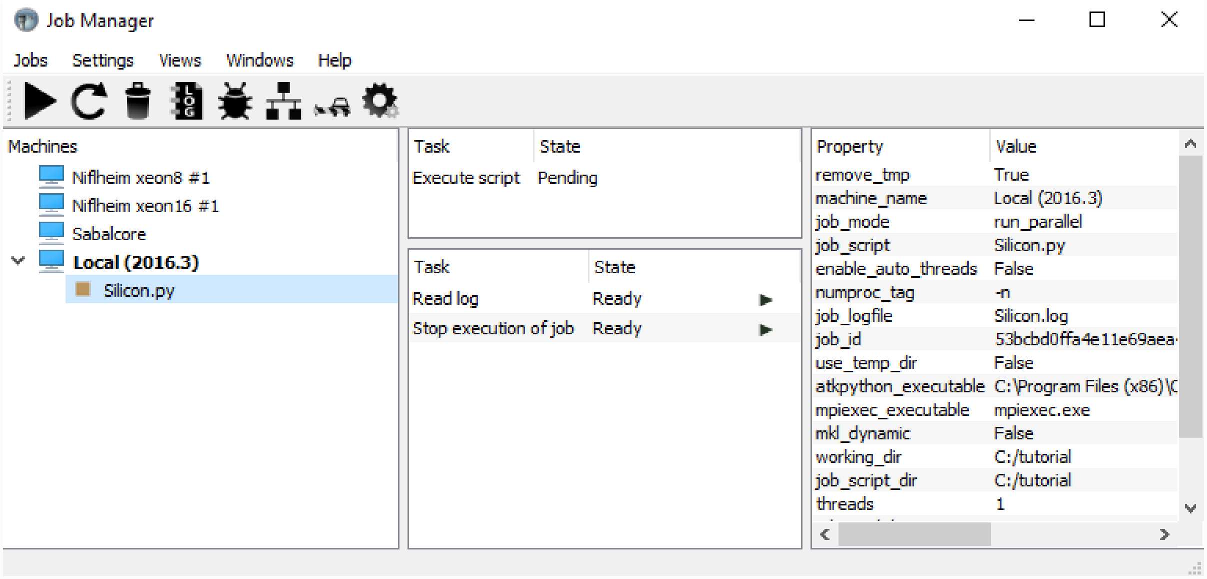


Running the calculation

After you have saved the script, the **Job Manager** opens up, and asks where you would like to run the script. Choose “Local” to run the calculation on your own machine, and click OK.

> **Note**
>
> A small job like this can be run on any computer, even a simple laptop, but for more complex simulations, the **Job Manager** offers the possibility to submit jobs to remote clusters and servers.

To run the job, click the ▶ start button on the toolbar.



Task	State
Execute script	Pending

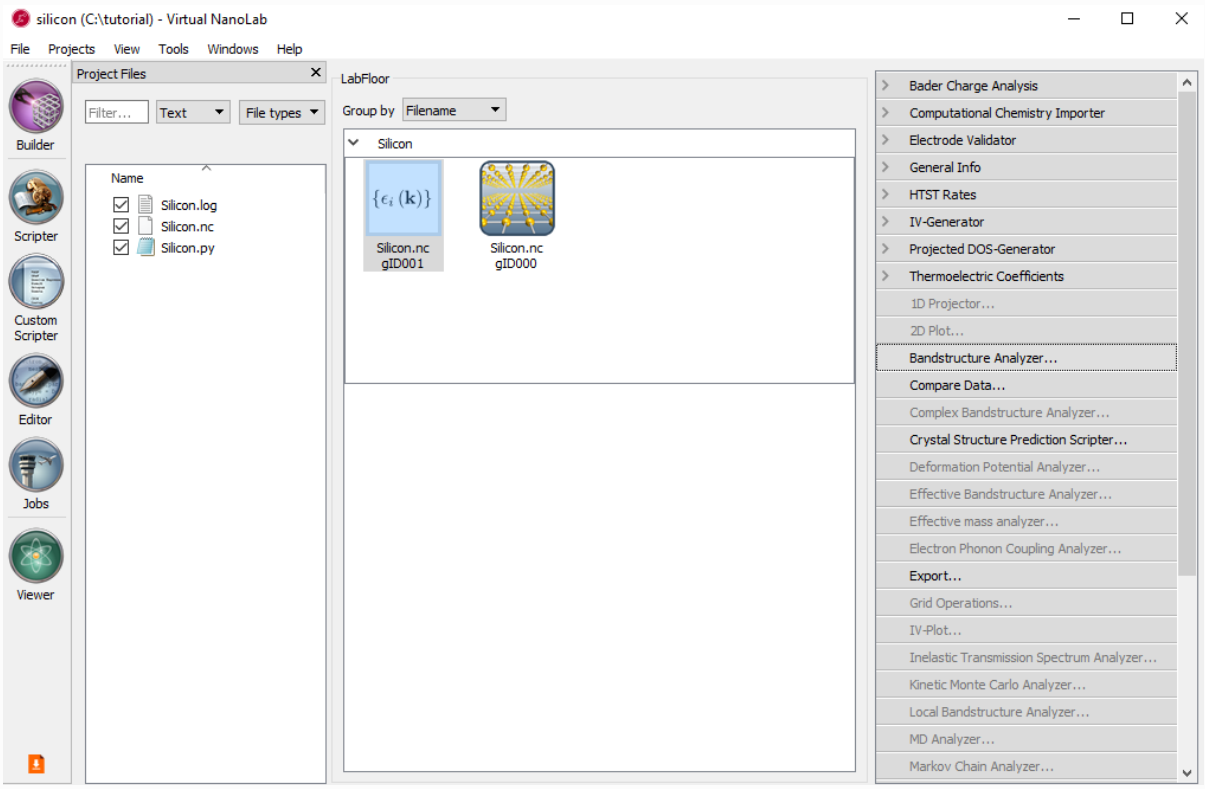
Task	State
Read log	Ready
Stop execution of job	Ready

Property	Value
remove_tmp	True
machine_name	Local (2016.3)
job_mode	run_parallel
job_script	Silicon.py
enable_auto_threads	False
numproc_tag	-n
job_logfile	Silicon.log
job_id	53bcbd0ffa4e11e69aea
use_temp_dir	False
atkpython_executable	C:\Program Files (x86)\C
mpiexec_executable	mpiexec.exe
mkl_dynamic	False
working_dir	C:/tutorial
job_script_dir	C:/tutorial
threads	1

Visualizing the band structure

The job should take less than a minute or so to complete. When it's done, you will note that a new file (`silicon.hdf5`) has appeared in the project (along with a log file for the calculation), and its contents are already visible on the LabFloor in the main QuantumATK window: a band structure and a bulk configuration.

1. Select the *band structure* object;
2. Click **Bandstructure Analyzer** on the right-hand side panel of analysis tools.

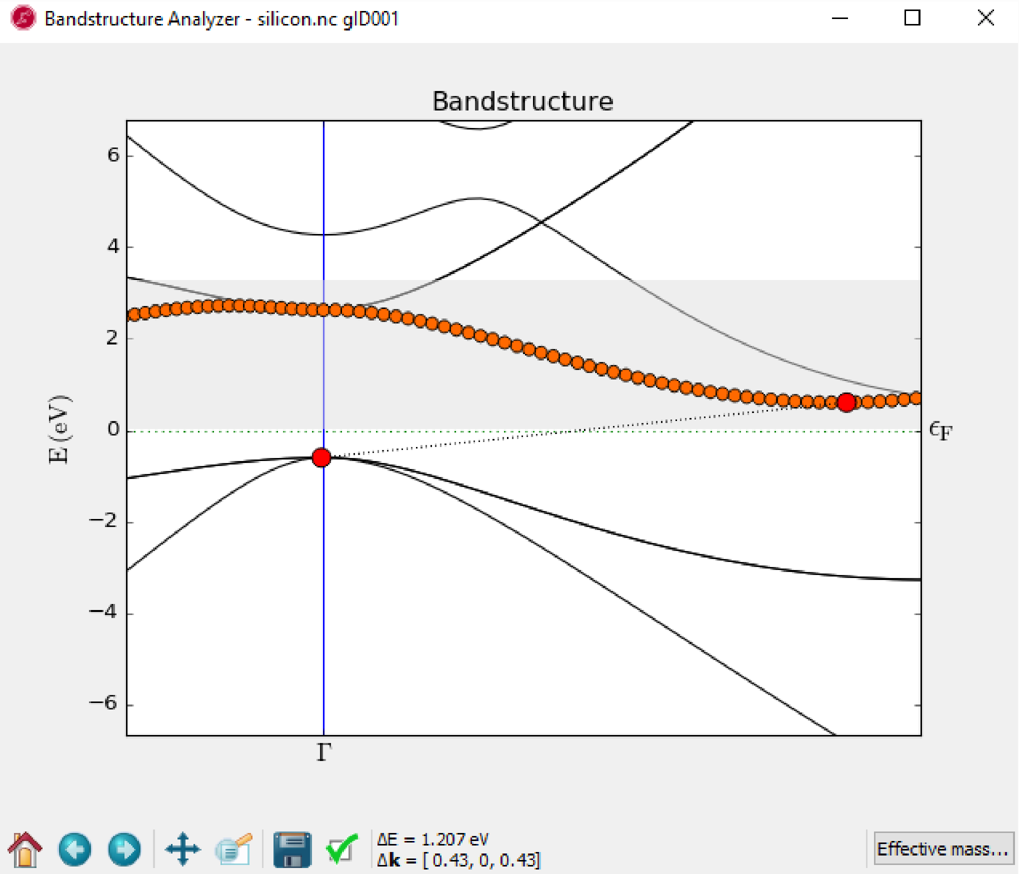

silicon (C:\tutorial) - Virtual NanoLab
File Projects View Tools Windows Help
Builder
Scripter
Custom Scripter
Editor
Jobs
Viewer
Project Files
Filter...
Text
File types
Name
☑ Silicon.log
☑ Silicon.nc
☑ Silicon.py
LabFloor
Group by Filename
Silicon
{ϵi (k)}
Silicon.nc gID001
Silicon.nc gID000
Bader Charge Analysis
Computational Chemistry Importer
Electrode Validator
General Info
HTST Rates
IV-Generator
Projected DOS-Generator
Thermoelectric Coefficients
1D Projector...
2D Plot...
Bandstructure Analyzer...
Compare Data...
Complex Bandstructure Analyzer...
Crystal Structure Prediction Scripter...
Deformation Potential Analyzer...
Effective Bandstructure Analyzer...
Effective mass analyzer...
Electron Phonon Coupling Analyzer...
Export...
Grid Operations...
IV-Plot...
Inelastic Transmission Spectrum Analyzer...
Kinetic Monte Carlo Analyzer...
Local Bandstructure Analyzer...
MD Analyzer...
Markov Chain Analyzer...

The band structure plot will appear, and you can verify the band gap to be 1.2 eV.

To measure the band gap,

1. Click the **zoom** buttom, and draw a rectangle with the mouse to zoom in around the Fermi level. Then click the zoom button again, to exit zoom mode.

2. Click a point in the band structure and hold down the mouse button, while moving the mouse to another point, and the difference in energy will be displayed in the toolbar.



ΔE = 1.207 eV
Δ**k** = [0.43, 0, 0.43]

Note

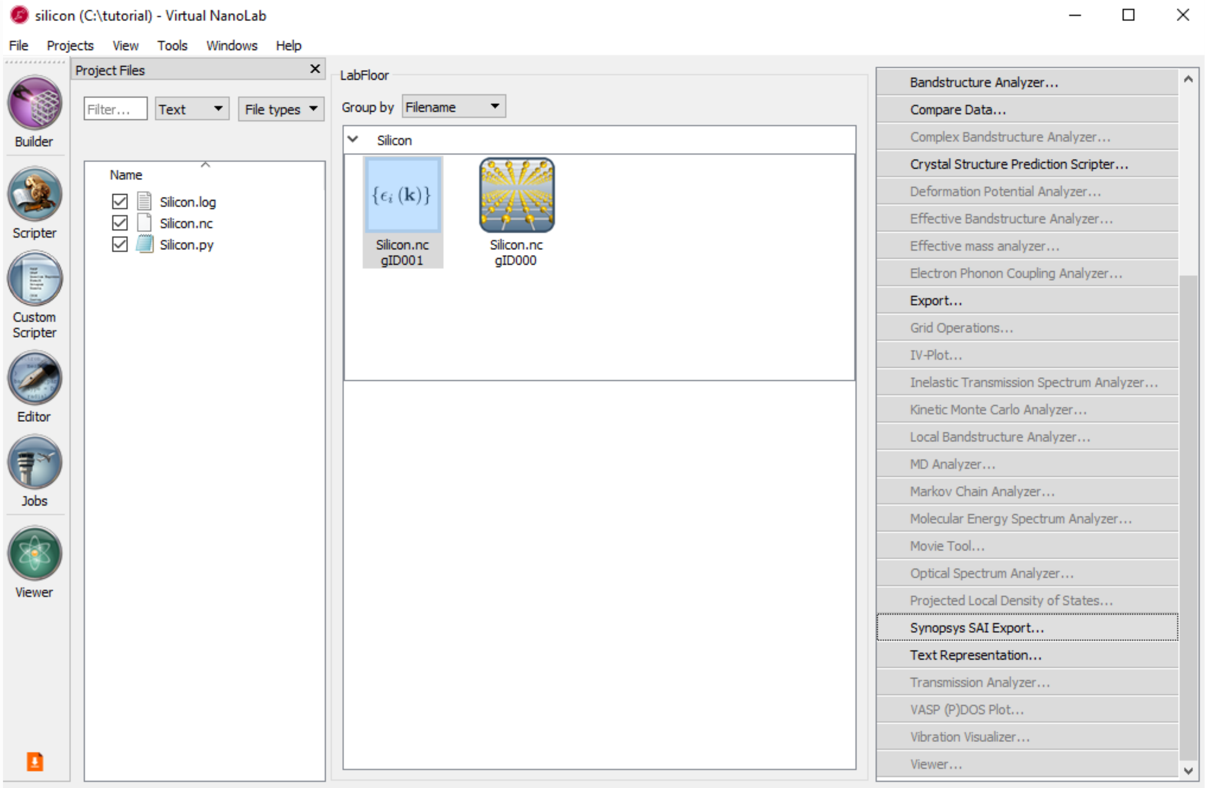
It is well-known that “standard” DFT fails to give good band gaps. In this tutorial you have, however, found that by using the TB09 meta-GGA (MGGA) exchange-correlation functional, the gap comes out close to the experimental value for silicon (and the same is true for many other semiconductors). In fact, the TB09-MGGA functional you used actually contains a tunable parameter which can be used to fit the gap very precisely.

Exporting the band structure to SAI format

The ultimate goal of this tutorial is to export the band structure in a format that can be imported in Synopsys tools. To do this, ensure that the band structure object is still selected, then locate the **Synopsys SAI Export…** button on the right-hand side toolbar.

Click it - then provide a filename and click Save. You have now exported an `.sai` file which can be imported in Synopsys tools, for further use in TCAD simulations.

It is also possible to select multiple band structure objects on the **LabFloor** and export them all with a single click on the **Synopsys SAI Export…** button. In this case, the filename of the .sai file will be automatically determined by the lab floor item title.


silicon (C:\tutorial) - Virtual NanoLab
File
Projects
View
Tools
Windows
Help
Builder
Scripter
Custom Scripter
Editor
Jobs
Viewer
Project Files
Filter...
Text
File types
Name
☑ Silicon.log
☑ Silicon.nc
☑ Silicon.py
LabFloor
Group by
Filename
Silicon
Silicon.nc gID001
Silicon.nc gID000
Bandstructure Analyzer...
Compare Data...
Complex Bandstructure Analyzer...
Crystal Structure Prediction Scripter...
Deformation Potential Analyzer...
Effective Bandstructure Analyzer...
Effective mass analyzer...
Electron Phonon Coupling Analyzer...
Export...
Grid Operations...
IV-Plot...
Inelastic Transmission Spectrum Analyzer...
Kinetic Monte Carlo Analyzer...
Local Bandstructure Analyzer...
MD Analyzer...
Markov Chain Analyzer...
Molecular Energy Spectrum Analyzer...
Movie Tool...
Optical Spectrum Analyzer...
Projected Local Density of States...
Synopsys SAI Export...
Text Representation...
Transmission Analyzer...
VASP (P)DOS Plot...
Vibration Visualizer...
Viewer...

Note

Specifically, each SAI file will be saved in the same directory as the HDF5 file containing the corresponding band structure, with the extension switched from “nc” to “sai” and with the unique object ID (seen in the label for each lab floor item - it is gID001 for the band structure created in this tutorial) appended to the filename to handle the case where there are several band structures in the same HDF5 file.

Batch processing and command line usage

A major advantage of QuantumATK is that in addition to the graphical user interface, you can also perform tasks via Python scripting.

First of all, to convert a single bandstructure, saved in an HDF5 file, to the SAI format, you just need 3 lines of Python code (assuming the addon has been installed, of course):

```
from AddOns.SynopsysTools.saiExporter import saiExporter
import sys
saiExporter(sys.argv[1], sys.argv[2])
```

Download this script atd_input_qw.py and save it in the folder containing the HDF5 file. Then run, as also described in the Synopsys *Ab-Initio to Device User Guide*:

atkpython atd_input_qw.py `$NC_FILE` `$SAI`

where `$NC_FILE` denotes the HDF5 (.hdf5) output file containing the BandStructure object, and `$SAI` denotes the desired output `.sai` file.

However, the scripting interface also makes it possible to do batch processing. For instance, assume you have a set of HDF5 files in a particular directory, and each one contains only one band structure. Then, open the **Editor** in QuantumATK and insert the following code:

```
from AddOns.SynopsysTools.saiExporter import saiExporter
import glob, sys
for nc_file in glob.glob("*.hdf5"):
    bs = nlread(nc_file, Bandstructure)[0]
    sai_file = nc_file.replace(".hdf5",".sai")
    print(nc_file, ">", sai_file)
    saiExporter(bs, sai_file)
```

Save the script in the same folder where the HDF5 files are (for instance with a name like `many_to_sai.py`), and run it from the command line (in the directory where the files are) as

atkpython many_to_sai.py

This will produce `silicon.sai` from `silicon.hdf5`, and so on. Of course, if you want to select only a subset of HDF5 files, you just need to modify `glob.glob` (”*.hdf5*”) to reflect your desired selection.

If you have HDF5 files with several band structures, use a modified script:

```
from AddOns.SynopsysTools.saiExporter import saiExporter
import glob, sys
for nc_file in glob.glob("*.hdf5"):
    all_bs = nlread(nc_file, Bandstructure)
    for i, bs in enumerate(all_bs):
        sai_file = nc_file.replace(".hdf5","_%i.sai" % i)
        print(nc_file, i, ">", sai_file)
        saiExporter(bs, sai_file)
```

This will make one SAI file for each band structure, and the files will be named after the HDF5 file (just like before) but with _*1, _2, etc* added.

As an alternative to specifying the HDF5 files inside the script, replace the line

```
for nc_file in glob.glob("*.hdf5"):
```

with

```
for nc_file in sys.argv[1:]:
```

and run it as

atkpython many_to_sai_2.py silicon*.hdf5

The possibilities to customize tasks like this with scripts in QuantumATK are endless.

Previous

Next