

Manual for cmd-PC1D

Casper Eijkens

June 2019

I would like to share a simple way to operate PC1D on Windows from a Python script. The first section explains where to download the files you need, the second section explains how to run PC1D from the command prompt, the third section explains how to create PC1D-readable files and the last section contains the code to run PC1D in Python. Note, that this is a tutorial for Windows only.

1 Downloading PC1D

For this tutorial you need the command version of PC1D, which can be downloaded from PVlighthouse (link 1) by clicking on `PC1DmodandcmdPC1Dv6.2.2.zip`. In this zip-file, you find two versions of cmd-PC1D: `cmd-pc1d5.exe` and `cmd-PC1D6-2.exe`. PC1Dmod6-2 is an updated version of PC1D5 with more realistic solar cell simulations. One limitation of version 6.2 is that it does not allow batch mode operation. However, this will not form any limitation for most users. In general, `cmd-PC1D6-2.exe` is the recommended version of PC1D.

2 PC1D from command prompt

In order to run PC1D from the command prompt, open `cmd` and navigate to the folder where you saved the PC1D folder. Then, simply type:

```
cmd-PC1D6-2.exe filename.prm
```

The `.prm`-file stores all information about a PC1D solar cell. Writing this line in `cmd` is equal to loading a parameter file into PC1D and running it. In the PVlighthouse zip-file, there are already some `.prm`-files available. When you saved your folder in `C:\Users\username\PC1Dmod`, you can run `PVcell_simple.prm` as follows. The result is shown in the appendix.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\username\PC1Dmod>cmd-PC1D6-2.exe PVcell_simple.prm
```

3 Creating .prm-files

Of course, you want to test your own solar cell designs. In order to change specific parameters of the solar cell, the .prm-file first needs to be decoded. The developer of cmd-PC1D, Halvard Haug, was so kind to create a tool for converting .prm-files to ascii. This program can also be found in the zip-folder under the name `convert_prm_to_ascii.exe`. In order to execute this program, we go back to `cmd` and type:

```
convert_prm_to_ascii.exe filename.prm filename.txt
```

In our command prompt this looks as follows. If there is no file called `example.txt` in the working directory, it will be created upon execution of the program.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

~\PC1Dmod>convert_prm_to_ascii.exe PVcell_simple.prm example.txt
```

Understanding the .prm format

When opening text file `example.txt`, we get a deeper insight into PC1D. After conversion of a .prm-file to ascii, we can open the text in `Notepad` and read it. The format might seem somewhat cryptic at first sight, but once your eyes are adjusted we discern different PC1D related parameters. For instance, the following lines are copied from the textfile and belong to the front reflection window in PC1D. When both `bCoated` and `bExternal` are 0, the Reflectance “Fixed” option is selected with a value of 0.055. This part of the .prm file is linked to the reflectance window in PC1D, shown in figure 1.

```
CDevice::m_FrontRfl:
CReflectance::m_bCoated=0
CReflectance::m_bExternal=1
CReflectance::m_Filename=example_reflectance.ref
CReflectance::m_Fixed=0.055
```

As you can see, even the location of the external reflection file is specified. I suggest that you write the whole path of the file, otherwise PC1D might get confused where the file is located. So write:

```
CReflectance::m.Filename=~\PC1Dmod\examplereflectance.ref
```

The same is true for other files that PC1D uses, like the `mat`-file, the `abs`-file, the `inr`-file, et cetera. If your program does not work, this is the first place I would look. In this text file, we can change *every* PC1D parameter. I encourage you to read through the text file yourself to find the location of every parameter.

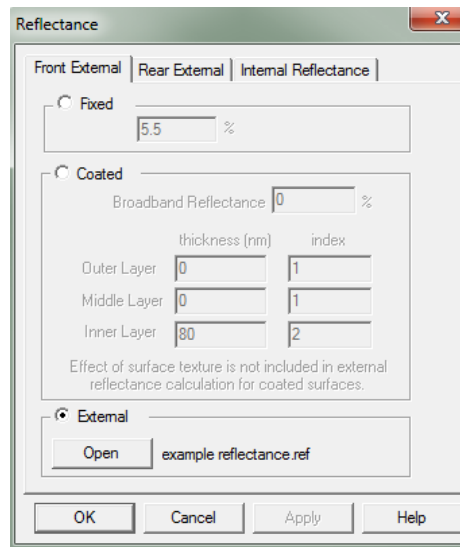


Figure 1: PC1D front reflectance window.

After you have changed your desired hyperparameters, you can save the text file and convert it back to a `.prm`-file as follows.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

~\PC1Dmod>convert_ascii_to_prm.exe example.txt PVcell.simple.prm
```

4 Changing the text file automatically.

Now you might wonder why I explained this cumbersome method of running PC1D. This method is very useful because it enables us to operate PC1D from a program script. There are multiple ways of achieving this, but the meta code looks like:

1. Calculate your desired PC1D-parameters from a script
2. Write a text file according to the PC1D format
3. Convert the text file to a .prm file
4. Run cmd-PC1D with this .prm file
5. Collect the output from cmd

There are different ways to achieve step 2. I wrote a Python dictionary with a key for every section. After all values are included into the text strings I join them and write them into a text file, see the code in the Appendix. If you have trouble achieving this, do not hesitate to contact me! To operate step 3, 4 and 5 from a Python script, please see the code in the Appendix. I want to thank Mattias Klaus Juhl for helping me with this part. I learned a lot from his Python wrapper for PC1D (Link 2). This is basically a stripped-down version of that. Also I would like to thank Yoann and Harsh for testing.

5 Links

1. <https://www2.pvlighthouse.com.au/resources/PC1D/PC1Dmod6/PC1Dmod6.aspx>
2. https://github.com/MK8J/PC1D_wrapper

6 Appendix

6.1 Execute output

The output of running `PV_cell_simple.prm`:

Base Voltage	Base Current	Base Power
-0.011271	-0.0394364	0.000444515
-0.011271	-0.0394364	0.000444515
0.0047278	-0.0394364	-0.000186449
0.0207274	-0.0394364	-0.000817412
⋮	⋮	⋮

6.2 Python code

```
import os
from io import BytesIO
import subprocess

# Join path of current file with the folder "files" containing the
# executable programs
dir_path = os.path.join(os.path.dirname(os.getcwd()), "files")
# Assign names to the executable programs that you need
PC1D_exe = os.path.join(dir_path, "cmd-PC1D6-2.exe")
asc2prm = os.path.join(dir_path, "convert_ascii_to_prm.exe")
# Create the PC1D text file from a dictionary
ascii = "".join("{0}".format(v) for v in str_dic.values())
# Choose a folder where you want to save the text file.
# IMPORTANT! Save the file locally and not on a dropbox/ onedrive
# folder since this will crash upon syncing.
textfile = "~\\Desktop\\dataSim\\ascii.txt"
# Choose a folder where you want to save the prm file
prmfile = "~\\Desktop\\dataSim\\out.prm"
# Write the text file
with open(textfile, "w") as f:
    f.write(ascii)
# Execute conversion file
txt2prm = subprocess.run([asc2prm, textfile, prmfile],
    stdout=subprocess.PIPE)
# Execute cmd-PC1D using the new .prm-file
pc1d = subprocess.Popen([PC1D_exe, prmfile], stdout=subprocess.PIPE)
# Take the output of the of the command prompt and turn it into a
# Python-readable file
(output, err) = pc1d.communicate()
data_str = BytesIO(output)
# Kill PC1D
pc1d.kill()
# Extract Pmpp, Vmpp, Impp
if data_str is not None:
    try:
        data = np.genfromtxt(data_str, delimiter="\t", names=True)
        index = np.argmax(-data["Base_Power"])
        Pmpp = -data["Base_Power"][index]
        Vmpp = -data["Base_Voltage"][index]
        Impp = -data["Base_Current"][index]

    except:
        print(BytesIO(output).read())
        data = None
        Pmpp= np.nan
        Vmpp= np.nan
        Impp= np.nan
```