For Calling Thermal Desktop From Python Script By: Dean Keithly

1. install Python 2.7 from <https://www.python.org/downloads/windows/>
   1. Select the x86-64 msi installer and run it
2. Add a bunch of paths to path
   1. C:\Python27
   2. C:\Python27\Scripts
3. Install pip: This will basically help you install all the stuff without much legwork
   1. Go to <https://github.com/BurntSushi/nfldb/wiki/Python-&-pip-Windows-installation> and find the link to download get-pip.py
   2. Open a command prompt cmd.exe
   3. Cd to the folder get-pip.py is located in
   4. Type python get-pip.py
4. Install pythonnet (I am installing 2.3.0)
   1. <https://pypi.org/project/pythonnet/>
   2. From the command prompt type pip install pythonnet
5. Install ipython
   1. From command line type pip install ipython
6. Adding TdApiV1 to Your Python Path
   1. Find where TdApiV1.dll (or the current thermal desktop equivalent) is located. Mine was located in C:\Windows\Microsoft.NET\assembly\GAC\_64\TdApiV1\v4.0\_1.0.64.0\_\_7132edd86f548fdf but it is possible that it is located in the GAC\_MSIL folder (same level as GAC\_64). Note that the GAC\_64 folder should contain a folder labeled TdApiV1 and TdApiV1.dll will be in a subfolder of this
   2. Open your environment variables (in windows 10 type “edit the system environment variables” into the Cortana search bar)
   3. In the “system properties” window under the “Advanced” tab click “Environment Variables”
   4. In the “User variables for XXusernameXX” create a new variable with variable name “PYTHONPATH”. Input the folder filepath to TdApiV1.dll to the value field and hit ok
   5. Edit “PYTHONPATH” and add the directory of your python installation. Mine is “C:\Python27”

Verify Things work

1. Open a new command prompt
2. Type ipython
3. Type import sys
4. Type sys.path
   1. Verify that the filepaths you just added to PYTHONPATH are in that list of filepaths. If not, Go and fix it
5. Type import clr
   1. If this fails, go and redo your pythonnet installation
6. Type import System
   1. If this fails, there is some issue with the current version of pythonnet you are using and I can’t help you
7. Type TAV = clr.AddReference(“TdApiV1”)
   1. BOOM! Congratulations you have created TAV, a TdApiV1 object

Running Thermal Desktop with Python

When running a new case, Python will open a template .dwg file

This .dwg file will contain the default ThermalDesktop units set to SI, reference the JPL model materials, JPL optical material properties, and large list of case sets.

From blank.dwg

Thermal->preferences

Set units to all si units

Hit ok

I WILL HAVE TO MANUALLY CREATE THESE PROPERTIES FOR MY STUFF

Go to thermal->thermalphysical properties-> edit property data

\*Add all properties we will use in model for materials (can manually specify these here)

OR

Thermal-thermapphysical properties-open/create propertyDB

Specify filename and put in whatever directory you want, select it and hit ok

(looking at edit property data shows you it is input)

Thermal->optical properties->open/create property DB (same as for thermalphysical properties)

#Now create simplest model possible

Need 1 node and 1 boundary condition \*tehnically two nodes

Thermal->Fd/FEM network->Node

Position node

\*View shaded Thermal PP is nicer to see as opposed to wireframe

Edit node

Thermal-Edit-select-node

Would define a specific\_heat for the node

Hit ok

Create boundary node

Thermal->Fd/FEM network->Node to create new node

Thermal-edit

Manually Select node (there is only one node)

Select boundary radial button (means constant temperature)

Set boundary temperature (usually 4K) \*interesting that it is not 2.7K but it doesn’t really matter

Hit OK

(graphically it appears as a trapezoid)

Going to connect nodes

Thermal->FD/FEM network->Node to node conductor

Select the two nodes

\*visually creates conduction oath between them

Thermal->edit

Manually select the conductor

Assign it thermal conductance (by default is conductor)

(Can check checkbox for radiation)

Hit OK

We are going to run this

Thermal-CaseSet manager (has a default case in there)

Add

(default case set 1)

Edit

(we didn’t edit anything in caseset 1)

(normally we would pull the full trajectory into here) and define solar panel pointing angles inside of IME and use these within Thermal desktop case set manager

Can choose steady state or transient

Hit ok

Select run selected case

(we don’t care about node numbers within thermal desktop) all nodes are created with node number 1

Click automatically renumber the duplicate ids

(thermal desktop is now running)

(should see successful dialog)

To run transient

Edit case manaer

Specify transient and specify start time and end time

UNSELECT steady state otherwise it will run steady state then transient

Run selected case

(should see successful dialog)

Thermal desktop has output file

Plotting data

Click on node and click on plot output data see results in XYZ editor

To run sensitivity analysis

Define a symbol

Thermal->symbol manager

Add symbol call it “Cond”

Give it default value like 1 W/K

\*repeat for all uncertain parameters

Thermal-edit

Manually Select conductor

Specify value for conductor as “Cond” by double clicking on value and then type in the expression OR right click and add general->defined symbole name

Then hit OK

Then hit OK

In case set manager

Under symbols, specify different value than what was there by specifying an override for that specific value

The end goal is to run a case set sensitivity analysis using the API

Notes: output file should already be saved

Digression

Technicallyin the caseset manager the Radiation tasks can input a trajectory but it requires position of spacecraft relative to the suin and planet and the angle of the spacecraft relative to the sun and planet.

Eric just takes his best judgement. Minimum altitude from Venus

Joshua spent his time developing a cubesat configuration.

The importance of mechanical configuration in early stages, in later stages, how it will look and get past the review board