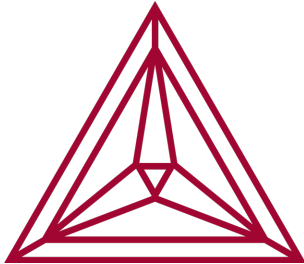


# Quick Installation Guide: TC-Python

Thermo-Calc Version 2019a



## TC-Python Quick Install Guide

---

This quick guide helps you do a TC-Python API installation.



There is an HTML help file you can open in your browser, plus a PDF of the same content, and several examples available with your Thermo-Calc installation.

Once you have installed TC-Python, go to Thermo-Calc and from the menu select **Help** → **Manuals Folder** and go to:

- **TC-Python**, then double-click index.html (or any .html file).
- **Software Development Kits (SDKs)** for the PDF.



Go to Thermo-Calc and search the online help (**Help>Online Help**).



For an installation video plus more information, go to <https://www.thermocalc.com/products-services/software/tc-python/>

This quick guide helps you to get a working TC-Python API installation.



A license is required to run TC-Python.

### *Step 1: Install a Python Distribution*



If you already have a Python distribution installation, version 3.5 or higher, skip this step.

These instructions are based on using the Anaconda platform for the Python distribution. Install version 3.5 or higher to be able to work with TC-Python, although it is recommended that you use the most recent version.

#### *Install Anaconda*

1. Navigate to the Anaconda website: <https://www.anaconda.com/download/>.
2. Click to choose your OS (operating system) and then click **Download**. Follow the instructions. It is recommended you keep all the defaults.

### *Step 2: Install Thermo-Calc and the TC-Python SDK*



TC-Python is available starting with Thermo-Calc version 2018a.

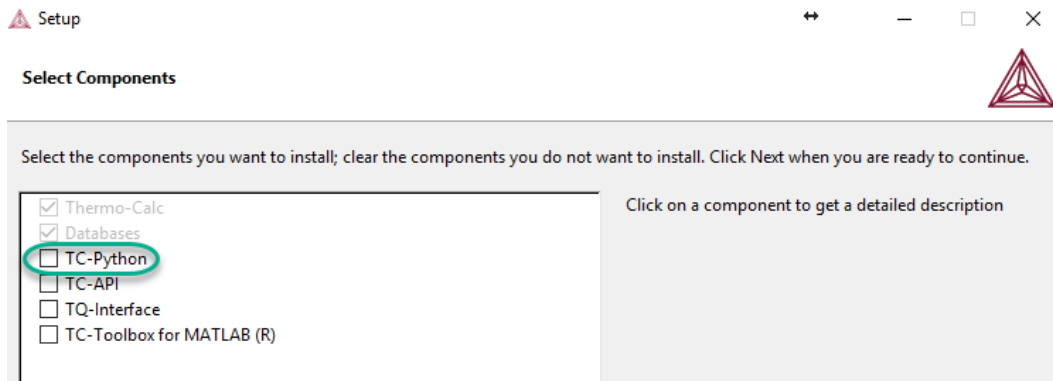
1. Install Thermo-Calc and choose a **Custom** installation.

▶ See "Custom Standalone Installation" in the *Thermo-Calc Installation Guide*.



If you have already installed Thermo-Calc, you need to find the installation file (e.g. Windows \*.exe, Mac \*.zip and Linux \*.run) to relaunch the installer and then continue with the next steps.

2. On the **Select Components** window, click to select the **TC-Python** check box.



3. On the **Install TC-Python** window, click **Next**.
4. When the installation is complete, the TC-Python folder opens and includes the \*.whl file needed for the next step. There is also an **Examples** folder with Python files you can use in the IDE to understand and work with TC-Python.



The installation location for this API is the same as for other SDKs and based on the OS. For details, see "Default Directory Locations" in the *Thermo-Calc Installation Guide*.

### Step 3: Install TC-Python

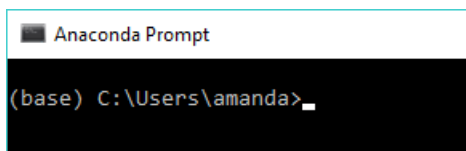
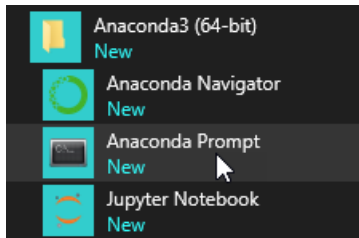
▶ [Updating to a Newer Version](#)

On Windows, it is recommended you use the Anaconda distribution prompt (i.e. Anaconda), especially if you have other Python installations.



Do not use Virtual Environments unless you have a good reason to do so. Also see ["Fixing Potential Issues with the Environment" on page 6.](#)

1. Open the command line. For example, in Anaconda on a Windows OS, go to **Start** → **Anaconda** → **Anaconda Prompt**.



2. At the command line, enter the following. Make sure there are no spaces at the end of the string or in the folder name or it will not run:

```
pip install <path to the TC-Python folder>/TC_Python-
<version>-py3-none-any.whl
```



For example, on a Windows OS Standalone custom installation, when you install for all users, the path to the TC-Python folder is

```
C:\Users\Public\Documents\Thermo-Calc\2019a\SDK\TC-Python\
```



Details for Mac and Linux installations are described in "Default Directory Locations" in the *Thermo-Calc Installation Guide*. On Linux typically *pip3* is used.

3. Press <Enter>. When the process is completed, there is a confirmation that TC-Python is installed.

#### *Step 4: Install an IDE (Integrated Development Environment)*

Any editor can be used to write the Python code, but an IDE is recommended, e.g. PyCharm. These instructions are based on the use of PyCharm.

Use of an IDE will give you access to code completion, which is of great help when you use the API as it will give you the available methods on the objects you are working with.

1. Navigate to the PyCharm website: <https://www.jetbrains.com/pycharm/download>.
2. Click to choose your OS and then click **Download**. You can use the **Community** version of Pycharm.
3. Follow the instructions. It is recommended you keep all the defaults.



For Mac installations, you also need to set some environment variables as described below in [Mac OS: Setting Environment Variables](#) .

### *Step 5: Open the IDE and Run a TC-Python Example*

After you complete all the software installations and set up the pip install path, you are ready to open the IDE to start working with TC-Python.

It is recommended that you open one or more of the included examples to both check that the installation has worked and to start familiarizing yourself with the code.

#### *Open the TC-Python Project in PyCharm*



When you first open the TC-Python project and examples, it can take a few moments for the Pycharm IDE to index before some of the options are available.

1. Open PyCharm and then choose **File** → **Open**. The first time you open the project you will need to navigate to the path of the TC-Python installation as done in Step 4.



For example, on a Windows OS Standalone custom installation, when you install for all users, the path to the TC-Python folder is

```
C:\Users\Public\Documents\Thermo-Calc\2019a\SDK\TC-Python\
```



Details for Mac and Linux installations are described in "Default Directory Locations" in the *Thermo-Calc Installation Guide*.

2. Click the **Examples** folder and then click **OK**.
3. From any subfolder:
  - Double-click to open an example file to examine the code.
  - Right-click an example and choose **Run** .

## Fixing Potential Issues with the Environment

In most cases you should run TC-Python within your **global** Python 3 interpreter and not use Virtual Environments unless you have a good reason to do so. If there are problems with the interpreter settings, you can resolve them in the Settings window:

1. Go the menu **File** → **Settings**.
2. Navigate in the tree to *<Project.YourProjectName>* and choose **Project Interpreter**.
3. Click on the settings symbol close to the **Project Interpreter** drop-down menu and choose **Add**.
4. Now choose **System Interpreter** and add your existing Python 3 interpreter.
5. Select your added interpreter and confirm.



If you are not following the recommended approach and create a new project (**File** → **New Project**), you need to consider that by default the options to choose the interpreter are hidden within the **Create Project** window. Therefore, click **Project Interpreter: New Virtual Environment** and in most cases choose your *System Interpreter* instead of the default *New Virtual Environment*.



If you really need to use a Virtual Environment, it is recommended you review the hints provided in the **Best Practices** section in the full TC-Python HTML help or PDF.

## Updating to a Newer Version

When updating to a newer version of Thermo-Calc, **you always need to also install the latest version of TC-Python**. It is not sufficient to run the installer of Thermo-Calc. The procedure is generally identical to [Step 3](#):

```
pip install <path to the TC-Python folder>/TC_Python-<version>-py3-none-any.whl
```

In case of problems you may wish to uninstall the previous version of TC-Python in advance:

```
pip uninstall TC-Python
pip install <path to the TC-Python folder>/TC_Python-<version>-py3-none-any.whl
```

However, that should normally not be required. Note that on Linux usually *pip3* is used.

You can check the currently installed version of TC-Python by running:

```
pip show TC-Python
```

## Mac OS: Setting Environment Variables

In order to use TC-Python on Mac you need to set some environment variables.

```
TC19A_HOME=/Applications/Thermo-Calc-2019a.app/Contents/Resources
```

If you use a license server:

```
LSHOST=<name-of-the-license-server>
```

If you have a node-locked license:

```
LSHOST= NO-NET
```

```
LSERVRC=/Applications/Thermo-Calc-2019a.app/Contents/Resources/lserverc
```

In PyCharm, you can add environment variables in the configurations.

Select **Run** → **Edit Configurations** to open the **Run/Debug Configurations** window. Enter the environment variable(s) by clicking the button to the right of the **Environment Variables** text field. Now the environment variable(s) will be set for each new configuration by default. The same way for configuring the environment variables can be used on other operating systems as if necessary.

