

VICON TRACKER PYTHON API QUICK START GUIDE

WHAT'S INSIDE

- About this guide 2
- Install the Tracker API 3
- Connect to the terminal server 9
- Example scripts 11

© Copyright 2020–2023 Vicon Motion Systems Limited. All rights reserved.
Revision 1. For use with Tracker 4.0

Vicon Motion Systems Limited reserves the right to make changes to information or specifications in this document without notice. Companies, names, and data used in examples are fictitious unless otherwise noted. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic or mechanical, by photocopying or recording, or otherwise without the prior written permission of Vicon Motion Systems Ltd.

Vicon® is a registered trademark of Oxford Metrics plc. Vicon Control™, Vicon Lock™, Vicon Lock Lab™, Vicon Lock Studio™, Vicon Tracker™, Vicon Valkyrie™, Vicon Vantage™, Vicon Vero™, Vicon Viper™, Vicon ViperX™ and Vicon Vue™ are trademarks of Oxford Metrics plc.

VESA® is a registered trademark owned by VESA (www.vesa.org/about-vesa/). Other product and company names herein may be the trademarks of their respective owners. For full and up-to-date copyright and trademark acknowledgements, visit <https://www.vicon.com/vicon/copyright-information>.

Vicon Motion Systems is an Oxford Metrics plc company.
Email: support@vicon.com Web: <http://www.vicon.com>



About this guide

About this guide

The Tracker API lets you control certain features of Tracker via a Python API. The API enables you to automate some common functions of Tracker, such as loading, playing and exporting data, activating or deactivating different objects, or triggering parts of the workflow.

This document enables you to get started with the Tracker API.

Install the Tracker API

Install the Tracker API

To use the Tracker API with Python, you must make sure that you have both installed.

The Tracker API provides support for Python 2.7 and Python 3. Vicon recommends that you use the latest full release of Python 3, unless your project requires you to use a specific version of Python.

These procedures guide you through the installation process:

- [Check Python version on page 4](#)
- [Installing Python on page 5](#)
- [Installing the Tracker Python module on page 6](#)
- [Check that the Python module installed correctly on page 8](#)

Install the Tracker API

Check Python version

If you are not sure if you have Python installed or which version of Python you are using, you can open a command prompt and run the `py` command. For example:

```
c:\>py
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

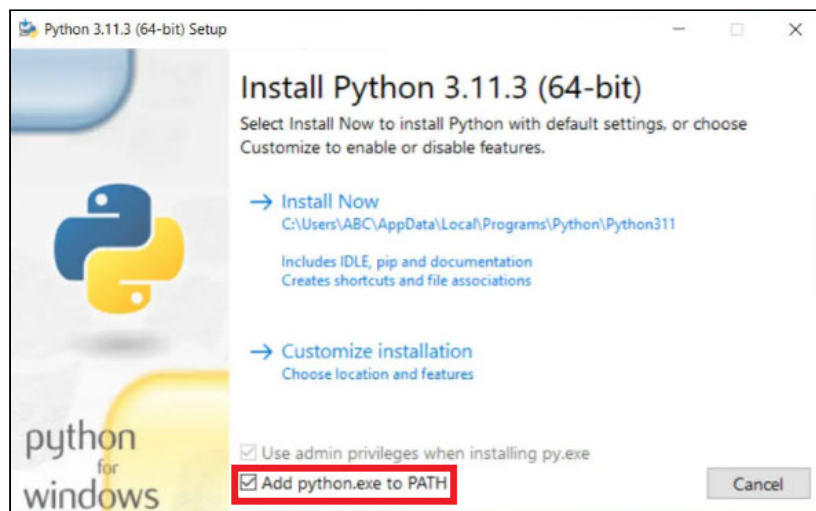
If you do not have Python installed, see [Install Python on page 5](#).

Install the Tracker API

Install Python

To install Python 2 or 3:

1. Go to <https://www.python.org/downloads/>
2. Locate the required version and install Python, ensuring that Add python.exe to PATH is selected:



In the above image, ABC is replaced with your username for the installation folder.

Install the Tracker API

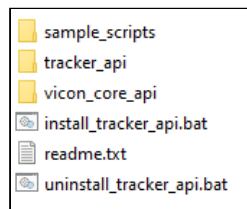
Install the Tracker Python module

To install the Tracker Python module:

1. Locate the installation files. If you installed Tracker in the default location, they are found in this folder:

`C:\Program Files\Vicon\Tracker4.x\SDK\Python`

These files are displayed:



2. Install the Tracker Python module in either of the following ways, depending on your particular installation:

- The simplest way is to run the batch file (*install_tracker_api.bat*) that is included in the Tracker installation (as shown in the above image). This usually works well if:
 - Python was installed to the PATH variable; or
 - Multiple versions of Python are installed, but you want to install the API to the latest version that you installed; or
 - Only a single version of Python is installed.

If any of these conditions apply, see [Install the python module by running the batch file on page 7](#).

- In all other cases, install the Python module by using pip. This usually applies if:
 - Multiple versions of Python are installed, but you want to install to a specific version; or
 - Multiple different versions of Python are installed and you want to install to all of them (in this case, you must install the module for each version); or
 - Only a single version of Python is installed, but you didn't install to PATH.

If any of these conditions apply, see [Install the Python module by running pip on page 7](#).

Install the Tracker API

Install the Python module by running the batch file

To do this:

1. Navigate to the Python install folder:

```
C:\Program Files\Vicon\Tracker4.x\SDK\Python
```

2. Double-click *install_tracker_api.bat*.

The installation process initializes automatically.

Install the Python module by running pip

1. Navigate to the *Scripts* folder for the Python that you want to use:

- For Python 3, the default installation folder is:

```
C:
```

```
\Users\\AppData\Local\Programs\Python\Python<version>\Scripts
```

- For Python 2.7, the default installation folder is:

```
C:\Python27\Scripts
```

2. Open a command window or powershell in that folder.

3. Run the following command to install the Vicon Core API:

```
C:
```

```
\Users\\AppData\Local\Programs\Python\Python311\Scripts> .\pip.exe install "C:\Program Files\Vicon\Tracker 4.0\SDK\Python\vicon_core_api"
```

4. Run the following command to install the Tracker API:

```
C:
```

```
\Users\\AppData\Local\Programs\Python\Python311\Scripts> .\pip.exe install "C:\Program Files\Vicon\Tracker 4.0\SDK\Python\tracker_api"
```

Install the Tracker API



Note

The above examples use a Python 3.11 installation with Tracker 4.0. Your path and commands may differ slightly.

Check that the Python module installed correctly

Check that the following modules have been installed:

- **vicon_core_api**: This is the core remote control API and includes a client for communication with the terminal server.
- **tracker_api**: Services API for accessing Tracker specific application functionality.

To test that the Tracker Python module installed correctly, try importing one of the modules in Python:

```
>>> import vicon_core_api
```

If the above process fails to recognize the module, try the following:

- Check the *site-packages* folder in the Python installation for the *tracker_api* or *vicon_core_api* folder. For Python 3.11, the location of the default installation folder is:
`C:\Users\<username>\AppData\Local\Programs\Python\Python311\Lib\site-packages`
- Check your system environment variables and ensure that the scripts folder for the Python installation you want to use is the highest in the list. For Python 3.11, the default location of the installation folder is:
`C:\Users\<username>\AppData\Local\Programs\Python\Python311\Scripts`

If either of the modules folders is missing, and you have verified the path, re-run through the installation process described in [Installing the Tracker Python module on page 6](#).

Connect to the terminal server

Connect to the terminal server

To connect to the terminal server, first import the Vicon Core API module:

```
>>> import vicon_core_api
>>> from vicon_core_api import *
```

Next, create a client. This automatically tries to connect to the specific host address on the default port (52800)

```
>>> c = Client('localhost')
```

Check that the client successfully connected to the server:

```
>>> print(c.connected)
True
```

If the response is `False`, ensure that you have an instance of Tracker running at the specified host address and your firewall is not blocking traffic on port 52800, before creating a new client.

When you have successfully connected, you can access the services provided by the Tracker terminal server.

This example uses basic object services:

```
>>> import tracker_api
>>> from tracker_api import BasicObjectServices
>>> services = BasicObjectServices(c)
```

Connect to the terminal server

When it is connected, you can call methods on the Tracker instance.

For example, to get a list of objects in the Tracking panel, use:

```
>>> result, object_list = services.basic_object_list()
>>> print(result)
Ok: the function succeeded
>>> print(object_list)
['Object1', 'Object2'...]
```

All API calls return a result code, which are described in `vicon_core_api/result.py`. One possible failure code is `Result.RPCNotConnected`, which is received if the connection to the terminal server is lost. For example:

```
>>> result, object_list = services.basic_object_list()
...
vicon_core_api.client.RPCError: RPCNotConnected: The connection to the remote
function or callback is not open
```

To display a list of all available functions and documentation:

```
>>> help( tracker_api)
```

Example scripts

Example scripts

You can find example scripts showing the use of common API functions at:

`C:\Program Files\Vicon\Tracker 4.0\SDK\Python\sample_scripts`

All the scripts have documentation and take a `--help` option that gives details of the relevant arguments.

To run a sample script, open a command window or power shell in the scripts folder above. You can do this in one of two ways:

- Open the command prompt and change your directory to the scripts folder:

```
c:\> cd C:\Program Files\Vicon\Tracker  
4.0\SDK\Python\sample_scripts
```

or

- Hold SHIFT+right-click in the scripts folder and select *Open command window here* or *Open Powershell window here*.

From here you can run the example script of your choice.

Example scripts

The following examples use the command window.

camera_calibration_wave.py

This script demonstrates how to use API functions to control the calibration process of starting and stopping the wand wave.

```
C:\Program Files\Vicon\Tracker 4.0\SDK\Python\sample_scripts> py
camera_calibration_wave.py
```

If successful, calibration controls are displayed.

```
Calibrating cameras...
Controls:  s - Start wand wave
          p - Stop wand wave
          c - Cancel wand wave
          x - Exit
-----
```

capture_control.py

This script shows how to capture live data.

```
C:\Program Files\Vicon\Tracker 4.0\SDK\Python\sample_scripts> py
capture_control.py
```

```
Capture name: 'API_Capture'
Capture controls:  s - Start capture
                  p - Stop capture
                  c - Cancel capture
                  x - Exit
-----
```

The Capture name is listed before the controls. To change the capture name, use `capture_services` and `SetCaptureName`.

For questions on using the Tracker API, contact [Vicon Support](#)¹.

¹ <mailto:support@vicon.com>