

PYTHON SCRIPTING WITH VICON SHOGUN

WHAT'S INSIDE

About Python scripting with Vicon Shogun	2
Python scripting with Shogun Live	3
About Python scripting with Shogun Live	3
Install the Shogun Live Python module	4
Connect a Python client to Shogun Live	10
Python scripting with Shogun Post	14
About Python scripting with Shogun Post	14
Install the Shogun Post Python module	15
Connect a Python client to Shogun Post	19
Use Vicon ShogunPost SDK interfaces	20
Switch the command line between HSL and Python	22
Run Python scripts in Shogun Post	23
Python / HSL interaction	24
Run Python from ShogunPostCL	25

© Copyright 2020–2022 Vicon Motion Systems Limited. All rights reserved.

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 Blade™, Vicon Control™, Vicon Lock™, Vicon Lock+™, Vicon Lock Lab™, Vicon Lock Studio™, Vicon Nexus™, Vicon MX™, Vicon Pegasus™, Vicon ProCalc™, Vicon Retarget™, Vicon Shōgun™, Vicon Studio™, T-Series™, Vicon Tracker™, Vicon Vantage™, Vicon Vero™, Vicon Vertex™, 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 Python scripting with Vicon Shogun

About Python scripting with Vicon Shogun

This guide contains introductory information about using Python with Vicon Shogun:

- [Python scripting with Shogun Live \(page 3\)](#)
- [Python scripting with Shogun Post \(page 14\)](#)

Python scripting with Shogun Live

About Python scripting with Shogun Live

Python scripting with Shogun Live

About Python scripting with Shogun Live

These topics provide basic, introductory information to get you started using Python with Shogun Live:

- [Install the Shogun Live Python module \(page 4\)](#)
- [Connect a Python client to Shogun Live \(page 10\)](#)


Python scripting with Shogun Live

Install the Shogun Live Python module

Install the Shogun Live Python module

To use Python and the Shogun Live Python module, you must first install them.

Both Python 2 and Python 3 are supported. Vicon recommends that you use the latest full release of Python 3, unless your project requires you to use a specific version of Python.

 With Shogun 1.4 and later, Python 3.0 and later is fully supported. Legacy support only is provided for Python 2.7.

For information on installation, see:

- [Install Python \(page 5\)](#)
- [Install the Python module \(page 6\)](#)
- [Check that the Python module installed correctly \(page 9\)](#)

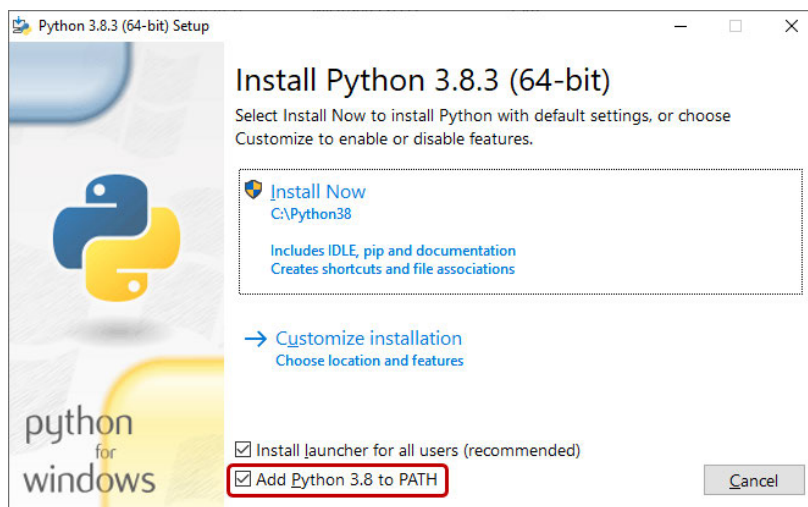
Python scripting with Shogun Live

Install the Shogun Live Python module

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 #.# to PATH is selected:



Python scripting with Shogun Live

Install the Shogun Live Python module

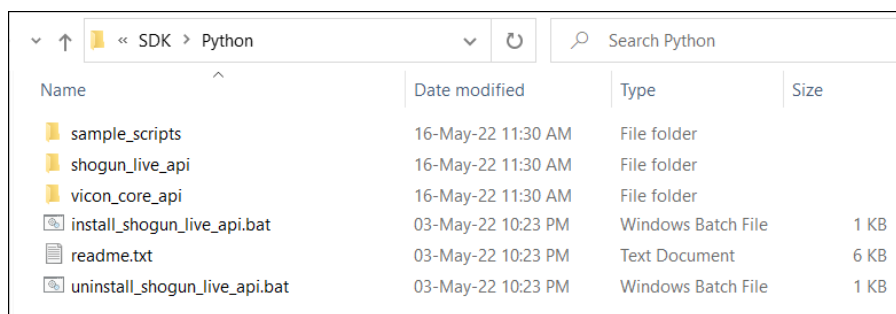
Install the Python module

To install the Python module:

1. Locate the installation files, which, if you installed Shogun Live in the default location, are found in this folder:

`C:\Program Files\Vicon\ShogunLive#\SDK\Python`

These files are displayed:



2. You can install the Shogun Live Python module in either of the following ways, depending on your particular installation:
 - The simplest way is to run the batch file (*install_shogun_live_api.bat*) that is included in the Shogun Live 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, [install the Python module by running the batch file \(page 7\)](#).

- 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, [install the Python module by running pip \(page 8\)](#).

Python scripting with Shogun Live

Install the Shogun Live Python module

Install the Python module by running the batch file

To do this, in a cmd window, run the batch file, *install_shogun_live_api.bat*.



```
Command Prompt
C:\Program Files\Vicon\Shogun Live\F00K\Python>install_shogun_live_api.bat
C:\Program Files\Vicon\Shogun Live\F00K\Python>python install "C:\Program Files\Vicon\Shogun Live\F00K\Python\vicon_core_api"
Processing C:\Program Files\Vicon\Shogun Live\F00K\Python\vicon_core_api
Requirement already satisfied: numpy in c:\users\p...venv\local\programs\python\python310\site-packages (from vicon_core_api==1.8.0) (1.1.0)
Using legacy 'setup.py install' for vicon_core_api, since package 'wheel' is not installed.
Installing collected packages: vicon_core_api
  Running setup.py install for vicon_core_api ... done
Successfully installed vicon_core_api-1.8.0

C:\Program Files\Vicon\Shogun Live\F00K\Python>python install "C:\Program Files\Vicon\Shogun Live\F00K\Python\shogun_live_api"
Processing C:\Program Files\Vicon\Shogun Live\F00K\Python\shogun_live_api
Requirement already satisfied: vicon_core_api in c:\users\p...venv\local\programs\python\python310\site-packages (from shogun_live_api==1.8.0) (1.1.0)
Requirement already satisfied: numpy in c:\users\p...venv\local\programs\python\python310\site-packages (from vicon_core_api==1.8.0) (1.1.0)
Using legacy 'setup.py install' for shogun_live_api, since package 'wheel' is not installed.
Installing collected packages: shogun_live_api
  Running setup.py install for shogun_live_api ... done
Successfully installed shogun_live_api-1.8.0

C:\Program Files\Vicon\Shogun Live\F00K\Python>
```

Python scripting with Shogun Live

Install the Shogun Live Python module

Install the Python module by running pip

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

- The default installation folder for Python 3 is:
C:\Users\\AppData\Local\Programs\Python\Python<version>\Scripts
- The default installation folder for Python 2.7 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:

```
.\pip.exe install "C:\Program Files\Vicon\ShogunLive1.8\SDK\ShogunLiveAPI\vicon_core_api"
```

```
Windows PowerShell
PS C:\Python38\Scripts> .\pip.exe install "C:\Program Files\Vicon\ShogunLive1.8\SDK\ShogunLiveAPI\vicon_core_api"
Processing c:\program files\vicon\shogunlive1.8\SDK\shogunliveapi\vicon_core_api
Requirement already satisfied: enum34 in c:\python38\lib\site-packages (from vicon-core-api==1.1.8) (1.1.10)
Installing collected packages: vicon-core-api
  Running setup.py install for vicon-core-api ... done
Successfully installed vicon-core-api-1.1.8
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
PS C:\Python38\Scripts>
```

4. Run the following command to install the Shogun Live API:

```
.\pip.exe install "C:\Program Files\Vicon\ShogunLive1.8\SDK\ShogunLiveAPI\shogun_live_api"
```

```
PS C:\Python38\Scripts> .\pip.exe install "C:\Program Files\Vicon\ShogunLive1.8\SDK\ShogunLiveAPI\shogun_live_api"
Processing c:\program files\vicon\shogunlive1.8\SDK\shogunliveapi\shogun_live_api
Requirement already satisfied: vicon_core_api in c:\python38\lib\site-packages (from shogun-live-api==1.4.0) (1.1.8)
Requirement already satisfied: enum34 in c:\python38\lib\site-packages (from vicon_core_api->shogun-live-api==1.4.0) (1.1.10)
Installing collected packages: shogun-live-api
  Running setup.py install for shogun-live-api ... done
Successfully installed shogun-live-api-1.4.0
WARNING: You are using pip version 19.2.3, however version 20.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
PS C:\Python38\Scripts>
```

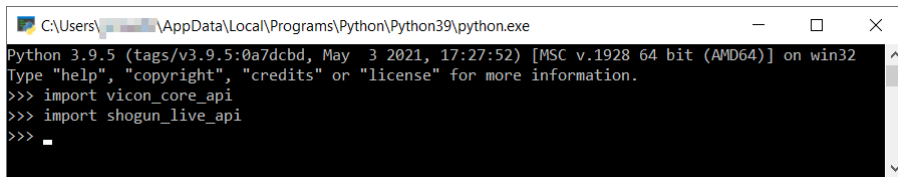

Python scripting with Shogun Live

Install the Shogun Live Python module

Check that the Python module installed correctly

To test that the Shogun Live Python module installed correctly, run Python and try the following import statements:

- `import vicon_core_api`
- `import shogun_live_api`



```
C:\Users\...AppData\Local\Programs\Python\Python39\python.exe
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import vicon_core_api
>>> import shogun_live_api
>>> _
```

If these import without any errors then the installation has been successful.

Python scripting with Shogun Live

Connect a Python client to Shogun Live

Connect a Python client to Shogun Live

For information related to connecting a Python client to Live, see:

- [Create a client and check the connection \(page 10\)](#)
- [Example: Setting the capture name \(page 11\)](#)
- [Explore using Python \(page 13\)](#)

Create a client and check the connection

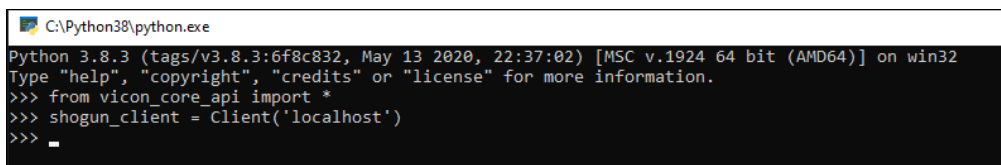
To connect a Python client to Shogun Live:

1. Start Shogun Live.
It is now ready to be interacted with via the Python API.
2. Start Python and begin by importing the Vicon Core API:

```
from vicon_core_api import *
```

3. Create a client. In this example, the object is named `shogun_client`:

```
shogun_client = Client('localhost')
```



```
C:\Python38\python.exe
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from vicon_core_api import *
>>> shogun_client = Client('localhost')
>>> _
```

Where:

- `shogun_client` is the name of the object, which can be any name you want
- `localhost` can be replaced with an IP address or hostname running Shogun Live.

4. Check that the client is connected:

```
shogun_client.connected
```

This returns True if connected, and False if not.

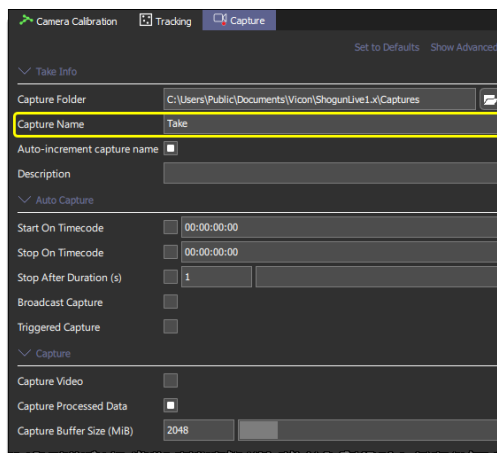
Python scripting with Shogun Live

Connect a Python client to Shogun Live

```
C:\Python38\python.exe
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from vicon_core_api import *
>>> shogun_client = Client('localhost')
>>> shogun_client.connected
True
>>> _
```

Example: Setting the capture name

In Shogun Live, towards the top of the Capture panel, the default capture name is Take.



This example shows how you can change the capture name using the Python API.

1. Import the relevant class for interacting with capture:

```
from shogun_live_api import CaptureServices
```

2. Create an object for the capture services. In this example, the object is named `capture`, but you can name it as required.

It references the object name used for the shogun client that was created:

```
capture = CaptureServices(shogun_client)
```

```
C:\Python38\python.exe
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from vicon_core_api import *
>>> shogun_client = Client('localhost')
>>> shogun_client.connected
True
>>> from shogun_live_api import CaptureServices
>>> capture = CaptureServices(shogun_client)
>>> _
```

Python scripting with Shogun Live

Connect a Python client to Shogun Live

- Use the `set_capture_name` method to set the capture name to 'MyTake001':

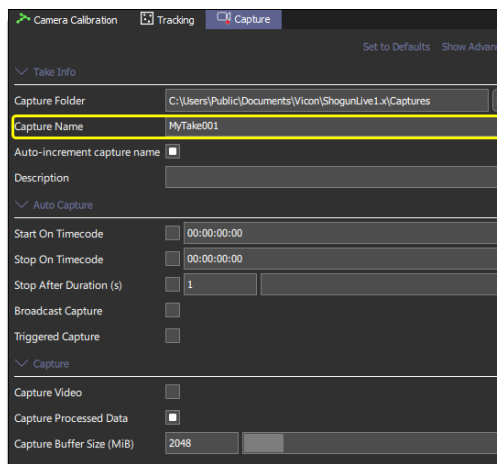
```
result = capture.set_capture_name('MyTake001')
```

```
C:\Python38\python.exe
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from vicon_core_api import *
>>> shogun_client = Client('localhost')
>>> shogun_client.connected
True
>>> from shogun_live_api import CaptureServices
>>> capture = CaptureServices(shogun_client)
>>> result = capture.set_capture_name('MyTake001')
>>>
```

- Check the return value to make sure that this succeeded:

```
C:\Python38\python.exe
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from vicon_core_api import *
>>> shogun_client = Client('localhost')
>>> shogun_client.connected
True
>>> from shogun_live_api import CaptureServices
>>> capture = CaptureServices(shogun_client)
>>> result = capture.set_capture_name('MyTake001')
>>> result
Result.Ok
>>>
```

In Shogun Live the capture name has changed:



Python scripting with Shogun Live

Connect a Python client to Shogun Live

For further information, see:

- The readme file:
C:\Program Files\Vicon\ShogunLive1.8\SDK\Python\readme.txt
- Sample scripts:
C:\Program Files\Vicon\ShogunLive1.8\SDK\Python\sample_scripts

Explore using Python

Run the Python `dir()` command on objects created, to show the methods that can be called:

For example, `dir(shogun_client)` or `dir(capture)`

```
>>> dir(shogun_client)
['_class', '_delattr', '_dict', '_dir', '_doc', '_enter', '_eq', '_exit', '_format', '_ge', '_getattr', '_gt', '_hash', '_init', '_init_subclass', '_le', '_lt', '_module', '_ne', '_new', '_reduce', '_reduce_ex', '_repr', '_setattr', '_sizeof', '_str', '_subclasshook', '_weakref', '_connect', '_locked_send_command', '_message_generator', '_read_loop', '_read_message', '_read_server_version', '_add_callback', '_add_schema_callback', '_callback_id_generator', '_callback_map', '_check_schemas', '_check_schemas_schema', '_client_failed_callback', '_condition', '_connect_timeout_seconds', '_connected', '_decoder', '_message_id_generator', '_pending_messages', '_remove_callback', '_send_command', '_send_json_command', '_send_raw_command', '_send_timeout_seconds', '_server_endpoint', '_server_version', '_socket', '_socket_timeout_seconds', '_stop', '_thread', '_version']
>>> dir(capture)
['_EState', '_class', '_delattr', '_dict', '_dir', '_doc', '_eq', '_format', '_ge', '_getattr', '_gt', '_hash', '_init', '_init_subclass', '_le', '_lt', '_module', '_ne', '_new', '_reduce', '_reduce_ex', '_repr', '_setattr', '_sizeof', '_str', '_subclasshook', '_weakref', '_registered_type_name', '_add_auto_capture_options_changed_callback', '_add_capture_options_changed_callback', '_add_latest_capture_changed_callback', '_add_take_info_changed_callback', '_call', '_cancel_capture', '_capture_description', '_capture_folder', '_capture_name', '_capture_notes', '_capture_processed_data_enabled', '_capture_video_enabled', '_client', '_duration_limit_in_seconds', '_latest_capture_errors', '_latest_capture_file_paths', '_latest_capture_name', '_latest_capture_state', '_latest_capture_timecode', '_limit_capture_duration_enabled', '_remove_callback', '_set_capture_description', '_set_capture_folder', '_set_capture_name', '_set_capture_notes', '_set_capture_processed_data_enabled', '_set_capture_video_enabled', '_set_duration_limit_in_seconds', '_set_limit_capture_duration_enabled', '_set_start_on_timecode_enabled', '_set_start_timecode', '_set_stop_on_timecode_enabled', '_set_stop_timecode', '_start_capture', '_start_on_timecode_enabled', '_start_timecode', '_stop_capture', '_stop_on_timecode_enabled', '_stop_timecode', '_unsupported']
```

Run `help()` on specific methods or classes.

For example, `help(capture.set_capture_name)`

```
>>> help(capture.set_capture_name)
Help on method set_capture_name in module shogun_live_api.interfaces.capture_services:

set_capture_name(name) method of shogun_live_api.interfaces.capture_services.CaptureServices instance
    Set the name for the next capture.

    Args:
        name < string >: Name suitable for use in file names.

    Return:
        return < Result >: Ok - On success.
        InvalidArgument - If the name includes invalid characters.

>>>
```

Python scripting with Shogun Post

About Python scripting with Shogun Post

Python scripting with Shogun Post

About Python scripting with Shogun Post

For introductory information about various aspects of using Python with Shogun Post, see these topics:

- [Install the Shogun Post Python module \(page 15\)](#)
- [Connect a Python client to Shogun Post \(page 19\)](#)
- [Use Vicon ShogunPost SDK interfaces \(page 20\)](#)
- [Switch the command line between HSL and Python \(page 22\)](#)
- [Run Python scripts in Shogun Post \(page 23\)](#)
- [Python / HSL interaction \(page 24\)](#)
- [Run Python from ShogunPostCL \(page 25\)](#)


Python scripting with Shogun Post

Install the Shogun Post Python module

Install the Shogun Post Python module

To use Python and the Shogun Post Python modules, you must first install them.

Both Python 2 and Python 3 are supported. Vicon recommends that you use the latest full release of Python 3, unless your project requires you to use a specific version of Python.

 With Shogun 1.4 and later, Python 3.0 and later is fully supported. Legacy support only is provided for Python 2.7.

For information on installation, see:

- [Install Python \(page 16\)](#)
- [Install the Shogun Post Python modules \(page 17\)](#)
- [Check that the Python modules installed correctly \(page 18\)](#)

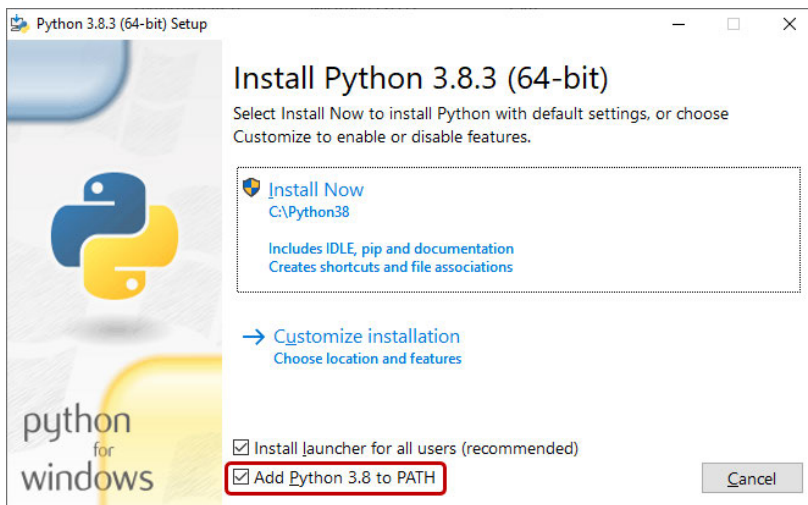
Python scripting with Shogun Post

Install the Shogun Post Python module

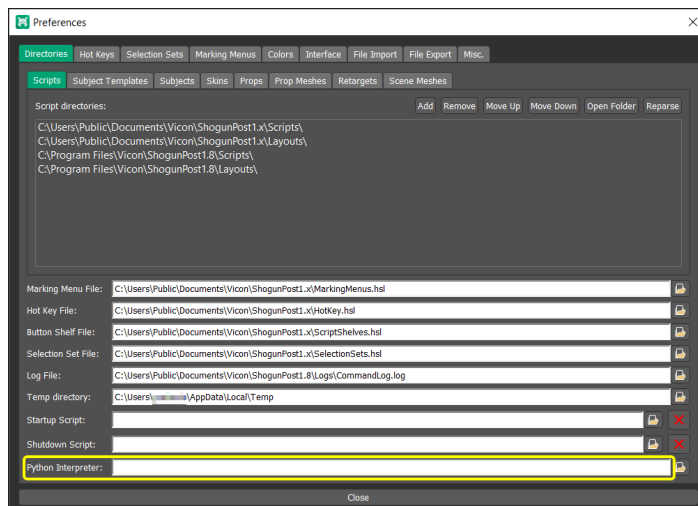
Install Python

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 #.# to PATH is selected:



To use a Python interpreter with Shogun Post, you must specify it in the Shogun Post Preferences dialog box, at the bottom of the Directories tab.



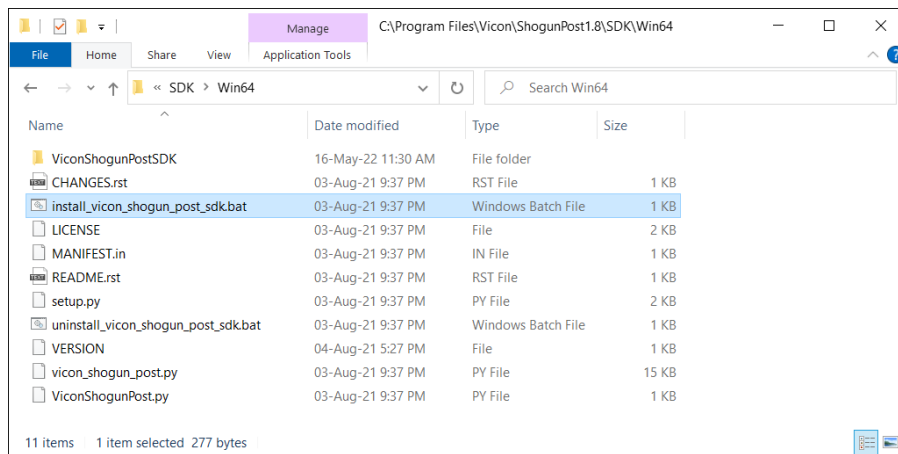
Python scripting with Shogun Post

Install the Shogun Post Python module

Install the Shogun Post Python modules

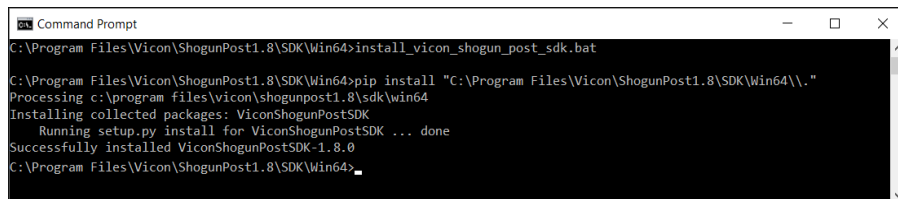
If you set the interpreter, you must also install the Shogun Post Python modules, using the installation utility provided in the following locations:

- For 64-bit installations of the Python interpreter:
`C:\Program Files\Vicon\ShogunPost#.#\SDK\Win64\install_vicon_shogun_post_sdk.bat`



- For 32-bit installations of the Python interpreter:
`C:\Program Files\Vicon\ShogunPost#.#\SDK\Win32\install_vicon_shogun_post_sdk.bat`

These scripts install the modules using the default pip tool.



Note

The same module supports Python 2 and Python 3.

Shogun Post Python modules installed

Shogun Post installs two Python modules:

- **ShogunPostSDK.** This folder contains the main Shogun Post SDK, and is the module that is most commonly used.
- **ViconShogunPost.** This module contains the Shogun Post implementation of a common Python SDK, shared by several Vicon products.

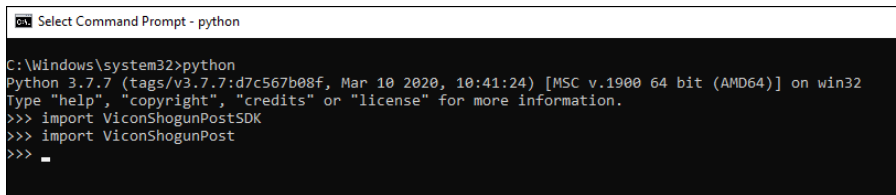
Python scripting with Shogun Post

Install the Shogun Post Python module

Check that the Python modules installed correctly

To test that the Shogun Post Python modules installed correctly, run Python and try the following import statement:

- `import ViconShogunPost`
- `import ViconShogunPostSDK`



```
Select Command Prompt - python
C:\Windows\system32>python
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import ViconShogunPostSDK
>>> import ViconShogunPost
>>> _
```

If this imports without any errors then the installation has been successful.

Python scripting with Shogun Post

Connect a Python client to Shogun Post

Connect a Python client to Shogun Post

When a Shogun Post Python client object is created it automatically tries to connect to a local instance of Shogun Post over TCP/IP via port 803.

```
import ViconShogunPost
shogun = ViconShogunPost.ViconShogunPost()
```

To direct the client to a different instance of Shogun Post, specify the appropriate parameters, or re-direct after creation using the Connect() method.

```
shogun = ViconShogunPost.ViconShogunPost( "192.168.0.0", 804 )
```

You can change the port used by Shogun Post for communication in the **Preferences** dialog box, on the **Misc** tab. The default behavior of Shogun Post is to communicate via port 803, though if multiple instances of Shogun Post are opened on the same machine they will automatically choose different port numbers (searching incrementally upwards). You can specify a fixed port in the **Preferences** dialog box, or from the application command line.

```
ShogunPost.exe -ControlStreamPort 804
```

```
ShogunPostCL.exe -controlStreamPort 804
```



Tip

If you disconnect your Ethernet cable and disable wifi, when you enter a Python command, the following error may be displayed:

```
Host Application is not connected, unable to retrieve
command list
```

This is because Python connects to Shogun Post over TCP/IP and if you are working entirely offline, Python and Shogun Post cannot connect. To solve this issue, install the Microsoft Loopback Adapter. For instructions on how to do this, see *Adding the MS Loopback Adapter on Windows 7*, on <http://blogs.msdn.com>.

Python scripting with Shogun Post

Use Vicon ShogunPost SDK interfaces

Use Vicon ShogunPost SDK interfaces

Two kinds of classes are defined in the *ViconShogunPostSDK* module. Classes like *Scene* or *Offline* define interfaces to distinct areas of Shogun Post's functionality. While you can create an instance of one of these interfaces, they can be more easily accessed directly from the Python client.

Example scripts showing how to run operations in Post

To import libraries:

```
import ViconShogunPostSDK
import ViconShogunPost
```

To connect to a local copy of Shogun Post:

```
shogun = ViconShogunPost.ViconShogunPost()
```

To load a file:

```
name="path and name of file to be loaded using /"
shogun.LoadFile( name )
```

To reconstruct a loaded file:

```
offline = ViconShogunPostSDK.Offline()
offline.Reconstruct()
```

To change the color of a marker:

```
scene = ViconShogunPostSDK.Scene()
objects = ViconShogunPostSDK.ObjectList( scene )
ltoe = objects['LTOE']
ltoe.Color(231,101,0)
ltoe.Color = (100,200,0)
```

Python scripting with Shogun Post

Use Vicon ShogunPost SDK interfaces

```
Command Prompt - python
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>python
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import ViconShogunPostSDK
>>> import ViconShogunPost
>>>
>>> shogun = ViconShogunPost.ViconShogunPost()
>>>
>>> name="D:/Database/Shogun Database/2020_06_12_Workflow_Testing/Handactions2.x2d"
>>> shogun.LoadFile( name )
>>>
>>> offline = ViconShogunPostSDK.Offline()
>>> offline.Reconstruct()
>>>
>>> scene = ViconShogunPostSDK.Scene()
>>> objects = ViconShogunPostSDK.ObjectList( scene )
>>> ltoe = objects['LTOE']
>>> ltoe.Color = (100,200,0)
```

Python scripting with Shogun Post

Switch the command line between HSL and Python

Switch the command line between HSL and Python

You can switch the command line in Shogun Post between HSL and Python.



The Python command line is essentially a mini script editor, not a direct interface to a Python interpreter, so variables will not persist from one call to the next. However, you don't need to create a client object when using the Shogun Post command line: default clients for the two Python modules are always available: `shogunPost` for the `ViconShogunPost` module, and `sdk` for the `ViconShogunPostSDK` module:

- `shogunPost.Command()` This is the `ViconShogunPost` module, which is the simple `Vicon SDK`.
- `sdk.Command()` This is the `ViconShogunPostSDK` module, which is the expanded `Shogun Post-specific SDK`.

Python scripting with Shogun Post

Run Python scripts in Shogun Post

Run Python scripts in Shogun Post

You can use Python scripts from anywhere in Shogun Post that currently accepts HSL scripts:

- General menu > Preferences dialog box > Directories tab.
- As a stage in a pipeline
- Attached to a shelf button
Note that when you add Python to a shelf button, a default shogunPost client is always available for use, as with the Shogun Post command line.
- Python command line (see [Switch the command line between HSL and Python \(page 22\)](#))

Python scripts have an identifying icon in the Script Viewer. You can assign them to hot keys in the same way as HSL commands.

Python scripting with Shogun Post

Python / HSL interaction

Python / HSL interaction

A command in HSL runs a Python command string, which works in the same way as the Shogun Post command line.

```
python "shogunPost.Command()";
```

You can call a Python script from HSL in the same way as a native one:

```
MyPythonScript;
```

It can also be given arguments:

```
MyPythonScript( "Arg1", "Arg2");
```

Note

Returning values back from Python to HSL is not supported. You also cannot step into a Python script in the debugger.

You can call HSL from Python, using the following HSL command.

```
Result = shogunPost.HSL('hslString;')
```

The command returns a Python string containing the result of the HSL operation.

Python scripting with Shogun Post

Run Python from ShogunPostCL

Run Python from ShogunPostCL

You can switch the ShogunPostCL command line between HSL and Python using the command `setLanguage`.

From HSL, the command is:

```
setLanguage "python";
```

From Python, the command is:

```
shogunPost.SetLanguage( "hsl" )
```