AJA Control Room™ Python Scripting Tutorial

AJA has implemented a Python Scripting interface for the AJA Control Room application.

Under Python script control you can:

- Name the file to be captured
- Start Capturing
- Stop Capturing

Our implementation is based upon Python v2.7. Version 3.4 will not work.

This tutorial and the example files are available as a download from:

http://www.aja.com/en/software/control-room/tutorials

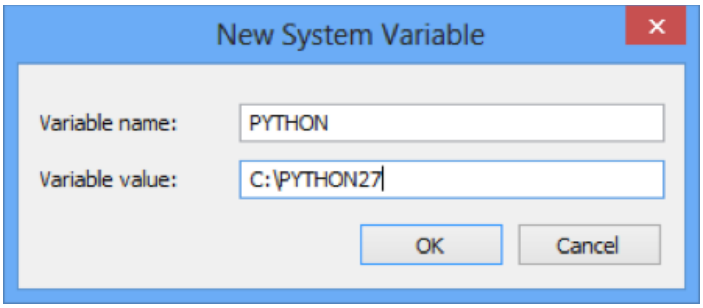
Python Installation:

Windows

The Python libraries are not installed with Windows by default. You can get the Python 2.7.6 installer from https://www.python.org/downloads/windows/.

Once this is installed, you must set an environmental variable to point to your Python installation. To add an Environmental Variable for Python:

- Right-click **My Computer**, and then click **Properties**.
- Click the **Advanced System Settings**.
- Click **Environment Variables**.
- Click one the following options, for either a user or a system variable:
  - Click **New** to add a new variable name and value.



- Click **OK**

Apple Macintosh

Python v2.7 is installed as part of the standard Mac OSX 10.8 and 10.9 system install. No additional installation is necessary.

AJA Control Room Setup for Python Scripting:

1. Copy the **python** folder to your desktop. It can be copied anywhere, but for this example we will expect it to be on your Desktop
2. Launch the **AJA Control Room** application
3. Open the AJA Control Room **Preferences** and
   a. Select the **Scripting** pane
      i. Make sure **Enable Scripting Server** is checked
      ii. The Scripting server port defaults to Port 8080. You can change this if desired but our example code expects it to be set to Port 8080. If you modify this port number you must change the appropriate value in the code to match.
   b. Select the **General** pane and
      i. Make sure **Hold onto device when app is in background** is checked
      ii. Make sure a **Capture Path** is specified
   c. Dismiss **Preferences** window
4. Select **Capture Tab** in Main window
   a. Specify desired **File Type**
   b. Specify desired **Video Codec**
   c. Specify desired **Audio Codec**
   d. Specify number of **Audio Channels**
   e. Confirm that the checkbox next to the **File Name** field is checked

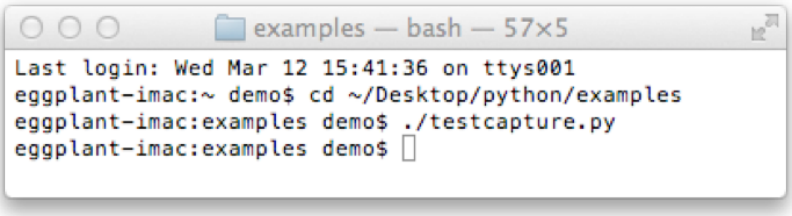
Macintosh Instructions

1. Make sure the **AJA Control Room** app is running and is set up as described above.

a. Open the **Terminal** Application

b. Type **cd ~/Desktop/python/examples**

c. Type **./testcapture.py** to run the testcapture Python script. The testcapture script tells AJA Control Room to capture a 1 second clip called "testClip". The captured clip will appear in the bin.



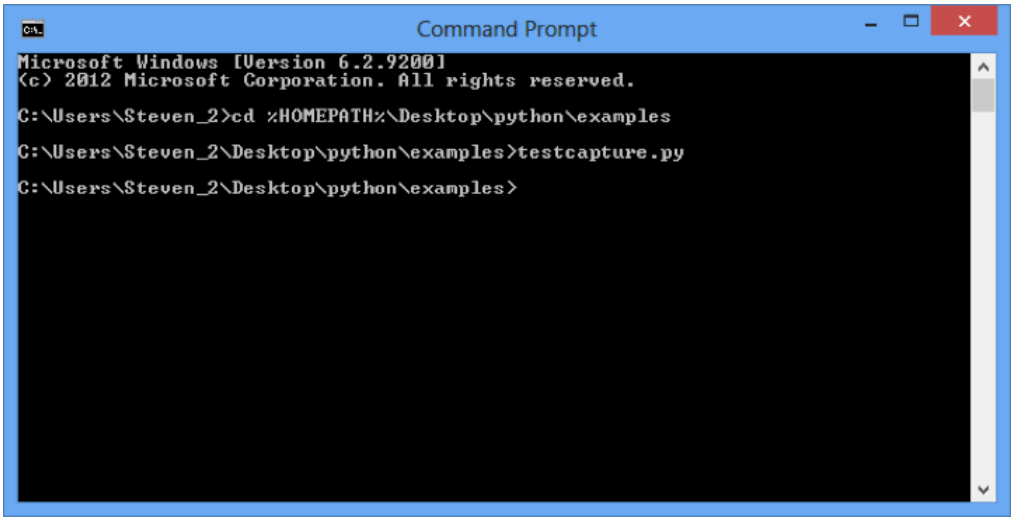
Windows Instructions

The following instructions expect the AJA Python folder to be on your Desktop. Make sure the AJA Control Room application is running

1. Open up a **Command Prompt**

2. Type **cd %HOMEPATH%\Desktop\python\examples**

3. Type **testcapture.py** to run the testcapture Python script. The testcapture script tells AJA Control Room to capture a 1 second clip called "testClip". The captured clip will appear in the bin.



Simple Modifications

Below is the code for testcapture.py

You can change the duration of the capture by changing the number (in seconds) in the parentheses in line 15 **time.sleep(1)**

The name of the file to be captured comes from the word in quotes in line 26 ("testClip" in the example below) **captureTest(client, "testClip")**

This script is meant to control the AJA Control Room app on the local computer, however you can control an AJA Control Room app on a different computer on the network by replacing **'localhost'** on line 22 with the IP address (e.g. **10.192.168.40**) of another computer on your network. If the Scripting server port in the Scripting Preferences has been changed you must match that port number in line 23. An instance of the AJA Control Room app must be running on the remote computer.

```
1  #!/usr/bin/python
2  # vim: tabstop=8 expandtab shiftwidth=4 softtabstop=4
3
4  import sys
5
6  sys.path.append("../")
7
8  from aja.controlroom.client import Client
9  import time
10
11
12 def captureTest(client,file):
13         client.capture.setFileName(file)
14         client.capture.startRecord()
15         time.sleep(1)
16         client.capture.stopRecord()
17
18 def main():
19     """
20     Used for testing
21     """
22     server = 'localhost'
23     port   = '8080'
24
25     client = Client(server,port)
26     captureTest(client,"testClip")
27
28
29
30
31 if __name__ == '__main__':
32     status = main()
33     sys.exit(status)
34
```

Example Python script "testcapture.py"