

# Python ESL

## About

The Python [ESL](#) module allows for native interaction with FreeSWITCH over the event socket interface.

It allows for sending commands, receiving output and sending and receiving events and IVR interaction from the FreeSWITCH server.

The Python ESL module is an auto-generated swig python module with a binary component to it.

You **CANNOT** just copy the PM file and use that, you must properly compile the module and get the `_ESL.so` file generated also (which must be kept in the same directory as the `ESL.py`).

**NOTE:** any time a change to `libesl` occurs you will need to re-make and install the python ESL module or else things may break or work un-expectedly.

## Recommended Installation

In the FreeSWITCH source directory change to `libs/esl` and run:

```
make pymod
make pymod-install
```

This should install the ESL module into your python site-packages folder. If for some reason you want to manually install it or keep it locally you still must run the `make pymod` command to compile it but then can copy the `libs/esl/_ESL.so` and `libs/esl/ESL.py` to a folder of your choosing.

See the "Pre-requisites" section at [the ESL page](#) for the libraries needed.

## Install from PyPi

You can alternatively install a specific well tested version of this module from the [Python Package Index](#).

The `libesl` code is directly from FreeSWITCH 1.4.18 and is packaged using [setuptools](#) native SWIG support.

It can be downloaded using `pip`:

```
pip install python-ESL
```

## Usage

The python module provides two classes: `ESLconnection` and `ESLevent`. They are documented generally (non-python specific) at the [ESL page](#), in the "ESL Object" section.

You will want to first include the module:

```
import ESL
```

You can then establish a connection using:

```
con = ESL.ESLconnection("127.0.0.1", "8021", "ClueCon")
```

## Examples

```
#!/usr/bin/env python

'''
events.py - subscribe to all events and print them to stdout
'''
import ESL

con = ESL.ESLconnection('localhost', '8021', 'ClueCon')

if con.connected():
    con.events('plain', 'all')
    while 1:
        e = con.recvEvent()
        if e:
            print e.serialize()
```

```
#!/usr/bin/env python

'''
server.py
'''
import SocketServer
import ESL

class ESLRequestHandler(SocketServer.BaseRequestHandler):
    def setup(self):
        print self.client_address, 'connected!'

        fd = self.request.fileno()
        print fd

        con = ESL.ESLconnection(fd)
        print 'Connected: ', con.connected()
        if con.connected():

            info = con.getInfo()

            uuid = info.getHeader("unique-id")
            print uuid
            con.execute("answer", "", uuid)
            con.execute("playback", "/ram/swimp.raw", uuid);

# server host is a tuple ('host', port)
server = SocketServer.ThreadingTCPServer(('', 8040), ESLRequestHandler)
server.serve_forever()
```

```
#!/usr/bin/env python

'''
single_command.py - execute a single command over ESL
'''
from optparse import OptionParser
import sys

import ESL

def main(argv):
    parser = OptionParser()
    parser.add_option('-a', '--auth', dest='auth', default='ClueCon',
                    help='ESL password')
    parser.add_option('-s', '--server', dest='server', default='127.0.0.1',
                    help='FreeSWITCH server IP address')
    parser.add_option('-p', '--port', dest='port', default='8021',
                    help='FreeSWITCH server event socket port')
    parser.add_option('-c', '--command', dest='command', default='status',
                    help='command to run, surround multi-word commands in "s')

    (options, args) = parser.parse_args()

    con = ESL.ESLconnection(options.server, options.port, options.auth)

    if not con.connected():
        print 'Not Connected'
        sys.exit(2)

    # Run command
    e = con.api(options.command)
    if e:
        print e.getBody()

if __name__ == '__main__':
    main(sys.argv[1:])
```