



Receiving Python API Errors

Take these actions if any of the following Python API errors appear:

Symptom	Solution	Example
The Python cli API throws a <code>NameError</code> .	Import the cli module into the global namespace.	<pre>>>> cli('show clock') Traceback (most recent call last): File "<stdin>", line 1, in <module> NameError: name 'cli' is not defined >>> from cli import * >>> cli('show clock') '20:23:33.967 UTC Fri Nov 01 2013\n'</pre>
The Python clid API throws a <code>structured_output_not_supported_error</code> .	Use the cli or clip API. The clid API works only with commands that support structured data output.	<pre>>>> clid('show clock') Traceback (most recent call last): File "<stdin>", line 1, in <module> File "/isan/python/scripts/cli.py", line 45, in clid raise structured_output_not_supported_error(cmd) errors.structured_output_not_supported_error: 'show clock'</pre>

Symptom	Solution	Example
<p>The cli API and cisco objects throw a Permission denied error.</p>	<p>Make sure your login ID has sufficient permissions to access the command or resource. If necessary, ask your network administrator for additional permissions.</p>	<pre> >>> from cli import * >>> cli('clear counters') Traceback (most recent call last): File "<stdin>", line 1, in <module> File "/isan/python/scripts/cli.py", line 20, in cli raise cmd_exec_error(msg) errors.cmd_exec_error: '% Permission denied for the role\n\nCmd exec error.\n' >>> from cisco.interface import * >>> i=Interface('Ethernet3/2') Traceback (most recent call last): File "<stdin>", line 1, in <module> File "/isan/python/scripts/cisco/interface.py", line 75, in __new__ cls._Interfaces[name].config(True) File "/isan/python/scripts/cisco/interface.py", line 91, in config s, o = nxcli('show runn interface %s' % self.name) File "/isan/python/scripts/cisco/nxcli.py", line 46, in nxcli raise SyntaxError, 'Error status %d\n%s' % (status, output) SyntaxError: Error status 30 % Permission denied for the role Cmd exec error. >>> import os >>> os.system('whoami') test </pre>

Symptom	Solution	Example
The urllib2 or socket connection is not processed.	Make sure you are using the correct virtual routing context. If not, switch to the correct one.	<pre> >>> import urllib2 >>> u=urllib2('http://172.23.40.211:8000/welcome.html') Traceback (most recent call last): File "<stdin>", line 1, in <module> TypeError: 'module' object is not callable >>> u=urllib2.urlopen('http://172.23.40.211:8000/welcome.html') Traceback (most recent call last): File "<stdin>", line 1, in <module> File "/isan/python/python2.7/urllib2.py", line 127, in urlopen return _opener.open(url, data, timeout) File "/isan/python/python2.7/urllib2.py", line 404, in open response = self._open(req, data) File "/isan/python/python2.7/urllib2.py", line 422, in _open '_open', req) File "/isan/python/python2.7/urllib2.py", line 382, in _call_chain result = func(*args) File "/isan/python/python2.7/urllib2.py", line 1214, in http_open return self.do_open(httplib.HTTPConnection, req) File "/isan/python/python2.7/urllib2.py", line 1184, in do_open raise URLError(err) urllib2.URLError: <urlopen error [Errno 113] No route to host> >>> from cisco.vrf import * >>> VRF.get_vrf_name_by_id(get_global_vrf()) 'default' </pre>

