Understanding the client

Previous Up Next

All communication with the *ecflow_server* is done with *ecflow_client*

For any communication with the server, the client needs to know the machine
where the server is running and the port on the server. There can be multiple
servers running on the same machine, each with unique port numbers.

This tutorial will show examples of using the client via the shell and in a Python script.

Client Shell Interface

For a full list of available commands type:

```
ecflow_client --help
```

The *ecflow_client* uses the following method of determining the **host** and **port:**

- Default host and port is **localhost**:**3141**
- These defaults are overridden by ECF_HOST and ECF_PORT environment variables
- This can be further overridden by using –port and –host options
  and can be used for any of shell level command shown with –help option.
  For example to ping a server on the command line we can use:

  ```
  ecflow_client --ping --host=<host> --port=<port>
  ```

Client Python Interface

The functionality provided by *ecflow_client* is also available via the python *Client Server API*.
The python interface uses the same algorithm for determining the host and port,
and allows the host and port to be set explicitly. See class ecflow.Client
This is shown by the following python example:

How to ping ecflow server in python

```
import ecflow
try:
    # When no arguments specified uses ECF_HOST and/or ECF_PORT,
    # otherwise defaults to localhost:3141
    ci = ecflow.Client() # inherit from shell variables
    ci.ping()

    # Explicitly set host and port using the same client
    # For alternative argument list see ecflow.Client.set_host_port()
    ci.set_host_port("machineX:4141") # actually set the host and port (change to your host and port)
    ci.ping()

    # Create a new client, Explicitly setting host and port.
    # For alternative argument list see ecflow.Client
    ci = ecflow.Client("oetzi:3444")  # another server
    ci.ping()

    # Ping inlined
    ecflow.Client("polonius:4266").ping()

except RuntimeError as e:
    print("ping failed: ", str(e))
```

What to do

If your *ecflow_server* was started with **ecflow_start.sh** and you want to use the shell
interface, then set ECF_HOST and ECF_PORT environment variables.

It should be noted that, if the server was started with **ecflow_start.sh** script
then the default “localhost:3141” will be incorrect.
e.g. in KSH

```
export ECF_HOST=<HOST> # as given when setting up ecflow server
export ECF_PORT=<PORT> # as given when setting up ecflow server
```

netstat can be used to determine the port number If the server was started on your local machine,

```
 netstat -lnptu | grep ecflow_server
```

Previous Up Next