ESP Clients

5/19/22 Brent Roman brent@mbari.org

Copyright MBARI 2022

ESP Client Startup

- Connects to ESPserver
  - waits for server to start if none found
  - may optionally connect to remote server
- Starts new thread for processing commands
  - names thread with after you
- Evaluates each command line
  - Ending line with '\' defers evaluation
    - ESPserver evaluates each Ruby expression
- Exit espclient with the command:

```
-> quit
```

Copyright MBARI 2022

espclient --help

Start interactive session on specified ESP server -- 4/24/22 brent@mbari.org

Options: <stdin and stdout may be redirected or piped>

```
 --remoteHost={name:port} #communicate with specified ESP hostname on TCP port
                     #:port # defaults to env var ESPcmdPort if omitted
                     #if name omtted, await connection on :port
 --server={unix pid}  #local esp server [defaults to the only one running]
 --prefix={string}    #prefix for thread name on server [espclient]
 --maxLineLen={#}     #max length of response lines [999]
                  #set maxLineLen to 0 for unlimited line length
 --time={seconds}     #seconds before reissuing prompt after idle [1.5]
                  #set repromptTime <= 0 to reprompt immediately
 --for={seconds}      #seconds to attempt connection to server [45]
 --wait             #attempt connection to server forever
 --beginWith={string} #command beginning interactive session [showlog 0]
 --all              #show all (including empty) results
 --verbose           #show stream multiplexing explicitly
 --debug            #summarize raw socket traffic on STDERR
 --help             #displays this text
```

Note that it is good practice to invoke this as:

```
   espclient yourName
```

so that your commands are tagged with yourName in the ESP server's log

Each logical line is submitted to the server for interpretation as typed.
Multiline expressions may be entered by terminating all but the last
physical line with the line continuation (e.g. \) character

Copyright MBARI 2022

Starting Detached Threads

- Detached threads
  - Have no parent threads
  - Continue running when client exits for any reason
  - Make little sense unless started from espclient
- To start a block of code in a detached thread

```
-> start(:myShortDA) {shortDA}
```

- You may substitute any code for `shortDA` above

```
-> Thread[:myShortDA].abort   #aborts assay
```

- To start a mission script within an espclient

```
-> runMission "script" #searches $ESPpath
```

  - Thread will be named "*script_mission*"

Copyright MBARI 2022

Selecting among Multiple ESP servers

- If multiple ESP servers are running
  - Cannot determine with which to connect

```
You have multiple ESP servers running.
Specify a Process ID among:
brent 8202
brent 8153
```

  - Specify one of the PIDs listed in error msg
    - For example:

```
espclient --server=8153 jones
```

  - Verify you selected the right one with

```
-> ESP::Mode   #verify correct operating mode
:simreal       #for example
```

Copyright MBARI 2022

ESP commands for clients

-> `showlog n` #display last *n* log messages

-> `hidelog` #suppress log message display

-> `clients` #list all espclients

- Enter Ruby comments to `#chat` with all

-> `ESP["name"].puts "hello"` #display 'hello'

-> `ESP["name"].interrupt` #presses control-C

-> `ESP["name"].quit` #will end client "name"'s session

- Exit Server (kills all clients!)

-> `ESP.main.exit` #This kills any running mission!

Copyright MBARI 2022

Connect to remote ESPserver

- On serverhost, start esp server with:

```
$ ESPcmdPort=9999 esp #listen on TCP port 9999
```

- On clienthost, start espclient with:

```
$ espclient -remote=serverhost:9999 joe
Session with serverhost:9999
joe@bruce:001->
```

- Much faster than logging in via ssh
- Very insecure
  - Use only on projected networks or within VPNs

Copyright MBARI 2022

Send ESPserver commands from other programs

- Pipe output of your program to espclient

```
$ echo "Trigger.now!" | espclient otto
```

  - espclient outputs only the value returned
  - logging is off by default when piping into espclient
  - to output log messages in addition to value returned:

```
$ echo "showlog 0; Trigger.now!" | espclient otto
```

- You may substitute any program for 'echo' above
- This technique has been used to remotely control ESPs

Copyright MBARI 2022