



Lua Tutorials

This page is here to help you understand ETPro Lua functions.

Before you start on this tutorial I assume you know basic Lua scripting. If not then I suggest you take a look at the online edition of the book *Programming in Lua* or the *lua-users* wiki. You can also take a look at code made by someone else, a list of ETPro mods that can be found [here](#), or our [Lua Sample Code](#).

Getting started

Loading a Script

Scripts are loaded by their file name out of the /etpro sub directory of your Enemy Territory installation, so don't forget to save them in the right folder.

if you want to load HelloWorld.lua you would have to set the server cvar lua_modules to "HelloWorld.lua".

If you want to load more mods you can add them to the same variables separated by a space like "HelloWorld.lua lol.lua pingpong.lua"

NOTE: if you modify this cvar all currently loaded lua modules will be unloaded.

After editing your script you save it and then reload it in ETPro to make your changes take effect, this can simply be done by `reset_match` server command.

Development Environment

While scripting you will see your code gets hard to read, its much easier to read with syntax highlighting. This can be done by scripting inside an lua development environment. These can be found under the Development Environments heading at [this page](#).

Tutorials

NOTE: Lines written in Bold are the actual lines we are going to write in our scripts.

Hello World

Here we will show a simple hello world example for the server console. Just follow the steps and read the comments.

1. First we need a new file so, create a new file and name it HelloWorld.lua.

Now we need an ETPro lua function that will be called automatically by ETPro when we want it. Therefore we will use `et_InitGame(levelTime,randomSeed,restart)`. That function is called once every round at the beginning. It's a good function to do some initial settings. We do not have to worry about those parameters (`levelTime,randomSeed,restart`) in this example.

2. Inside our file at line 1 we will write function `et_InitGame(levelTime,randomSeed,restart)` -- called at beginning of a gameround

Now we got a part of our script that is called at the beginning of a round so now we can add our code to it. We are going to print "hello world" in the server console. To do this we will use `et.G_Print("message")`. That function will simply print the "message" in the server console.

3. At line 2 we will write `et.G_Print("Hello World!!!\n")` -- printout our text to the console

Lastly we need to end out function at line 1 with a simple end

4. At line 3 place end --close `et_InitGame()`

Now if you did it right you should have something like this:

```
function et_InitGame(levelTime,randomSeed,restart) -- called at beginning of a gameround
    et.G_Print("Hello World!!!\n") -- printout our text to the console
end --close et_InitGame()
```

Now save and load your script and you will see Hello World!!! in the server console.

Hello Spam

Here we will show a simple Spammy hello world example for the server console.

Just follow the steps and read the comments.

1. First we need a new file so, create a new file and name it HelloSpam.lua.

Now we need an ETPro lua function that will be called automatically by ETPro many times to make it spammy. Therefore we will use `et_RunFrame(levelTime)`. That function is called every server frame. It is a good function for stuff you need to constantly check but you should not put to much code in here. We do not have to worry about those parameter (`levelTime`) in this example.

2. Inside our file at line 1 we will write function `et_RunFrame(levelTime)` --called every serverframe

Now we got a part of our script that is called at the beginning of a round so, now we can add our code to it. We are going to print "hello world" in the server console. To do this we will use `et.G_Print("message")`; that function will simply print the "message" in the server console

3. Now at line 2 we will write `et.G_Print("Hello Spam!!!\n")` -- printout our text to the console

Lastly we need to end our function at line 1 with a simple at line 3 end

4. At line 3 place end --close `et_RunFrame()`

Now if you did it right you should have something like this:

```
function et_RunFrame(levelTime) --called every serverframe
    et.G_Print("Hello Spam!!!\n") -- printout our text to the console
end --close et_RunFrame()
```

Now save and load your script and you will see Hello Spam!!! in the server console.

Hello Client

Here we will show a simple hello world example for the client console.

Just follow the steps and read the comments.

1. First we need a new file so, create a new file and name it HelloClient.lua.

Now we need a ETPro lua function that will be called automatically by ETPro when we want it. To do this we will use `et_InitGame(levelTime,randomSeed,restart)` as shown in the Hello World example.

2. Inside our file at line 1 we will write function `et_InitGame(levelTime,randomSeed,restart)` --called at beginning of a gameround

Now we need a function to print a message to the client his console. We will use `et.trap_SendServerCommand(-1,"print \"message\n\"")` to achieve this. That function will simply print the "message" in the client's console.

The first parameter is the client slot to send to. When this is -1 ETPro will send the message to all clients. The 2nd parameter looks confusing and like the original author's spelling, it is. First you have the main command witch is "print ..". Then, at the place of the dots, comes the data to send with the print. In our case this is "message\n".

The \n will send a newline command to the console so the next message written to the console is on the next line. The \" are just quotes but since they are already inside a string you need to prefix them with a \ (commonly known as an escape character).

3. So just write this at line 2 `et.trap_SendServerCommand(-1,"print \"Hello Client\n\"")` -- printout our text to the console

Last we need to end our function at line 1 with a simple end

4. At line 3 place end --close `et_InitGame()`

Now if you did it right you should have something like this:

```
function et_InitGame(levelTime,randomSeed,restart) -- called at beginning of a gameround
    et.trap_SendServerCommand(-1,"print \"Hello Client\n\"") -- printout our text to the console
end -- close et_InitGame()
```

Now save and load your script and you will see Hello World!!! in the client's console at the begin of a round.

Server Command lol

Here we will show how to make a new server command that will simply print hihi.

Just follow the steps and read the comments.

1. First we need to create a new file for the script. Create a new file and name it ServerCmdLol.lua.

To start we need an ETPro Lua function that will be called automatically by ETPro when a certain server command has been entered.

To do this we will use `et_ConsoleCommand()`.

It is called for every server command but not for engine commands like say,broadcast,echo.

2. Inside our file at line 1 we will write function `et_ConsoleCommand()`--called when someone types a command in conosle or uses sa todo a server command

Now the first thing we need to know is what command was entered.

Commands that are entered will be split up by spaces, so `cmd1 hello all` will be split up into "cmd1", "hello" and "all".

The only exception is that `cmd2 "hello all"` will be split up in "cmd2" and "hello all" because, "hello all" is entered as one string by enclosing them in quotes.

To get a part of the command you should use `et.trap_Argv(partnumber)`

`partnumber 0` is the actual command and, `partnumber 1` is the first parameter of the command

so in our example above we would get:

```
et.trap_Argv(0) = "cmd1"
```

```
et.trap_Argv(1) = "hello"
```

```
et.trap_Argv(2) = "all"
```

3. At line 2 we will write `local cmd = et.trap_Argv(0)` --load out the first parameter of the test typed in console (the command itself)

The commands are sent the same way they were entered, so if you enter `cMd3` you will get `cMd3`.

To detect all variations, we will convert the string to lowercase using a normal Lua function.

4. To make it lower case we will write `cmd = string.lower(cmd) --convert it to lowercase so it is easier to match in our if statements at line 3`

Now we can use a simple if statement to check if the command is lol.

5. At line 4 write `if cmd == "lol" then --run our code if the command was "lol"`

If thats true then the script should print "hihi".

6. To do this, at line 5 write `et.G_Print("hihi\n") --print out "hihi"`

Now that we have handled the command in our ETPro Lua script, we do not need to send it to the normal ETPro code since that will return "Unknown Command" anyway.

We can let ETPro ignore the command by returning 1 out of our function.

7. So at line 6 we will write `return 1 --return 1 to make ETPro ignore the command so you wont get an 'invalid command' warning`

Thats all we need to do for this command so we can start closing our function

8. At line 7 we end our if statement of line 4 so we will write `end --close the if at line 4`

Now we get to the code that gets executed when the command was not lol.

We got nothing todo here so we will return 0 to make ETPro handle it.

9. To make ETPro handle all other commands, write `return 0 --if the command wasnt 'lol' then we will get here and return 0 to make ETPro handle it at line 8`

Now we are almost done just end our function.

10. So at line 9 we will write `end --close function et_ConsoleCommand() at line 1`

Now if you did it right you should have something like this:

```
function et_ConsoleCommand() --called when someone types a cokmmand in conosle or uses sa todo a server command
    local cmd = et.trap_Argv(0) --load out the first parameter of the test typed in console (the command itself)
    cmd = string.lower(cmd) --convert it to lower case so its easier to match in our if statements
    if cmd == "lol" then --run our code if the command was 'lol'
        et.G_Print("hihi\n") --print out 'hihi'
        return 1 --return 1 to make etpro ignore the command so u wont get an 'invalid command' warning
    end --close the if at line 7
    return 0 --if the command wasnt 'lol' then we will get here and return 0 to make etpro handle it
end -- close et_ConsoleCommand()
```

Now save and load your script and you will be able to use lol as a server command.

Server Command Dumpuserid

Here we will show how to make a new server command that will simply preform a the dumpuser cmd on the name of the client slot we enter.

Just follow the steps and read the comments.

1. First we need to create a new file for the script. Create a new file and name it `ServerCmdDumpuserid.lua`.

To start we need an ETPro Lua function that will be called automatically by ETPro when a certain server command has been entered.

To do this we will use `et_ConsoleCommand()`.

It is called for every server command but not for engine commands like `say,broadcast,echo`.

2. Inside our file at line 1 we will write `function et_ConsoleCommand() --called when someone types a command in conosle or uses sa todo a server command`

Now the first thing we need to know is what command was entered.

Commands that are entered will be split up by spaces, so `cmd1 hello all` will be split up into "cmd1", "hello" and "all".

The only exception is that `cmd2 "hello all"` will be split up in "cmd2" and "hello all" because, "hello all" is entered as one string by enclosing them in quotes.

To get a part of the command you should use `et.trap_Argv(partnumber)`

`partnumber 0` is the actual command and, `partnumber 1` is the first parameter of the command

so in our example above we would get:

```
et.trap_Argv(0) = "cmd1"
```

```
et.trap_Argv(1) = "hello"
```

et.trap_Argv(2) = "all"

3. At line 2 we will write local cmd = et.trap_Argv(0) --load out the first parameter of the test typed in console (the command itself)

The commands are sent the same way they were entered, so if you enter cMd3 you will get cMd3.

To detect all variations, we will convert the string to lowercase using a normal Lua function.

4. To make it lower case we will write cmd = string.lower(cmd) --convert it to lowercase so it is easier to match in our if statements at line 3

Now we can use a simple if statement to check if the command is dumpuserid.

5. At line 4 write if cmd == "dumpuserid" then --run our code if the command was "dumpuserid"

If thats true then we need to know witch client id to dump the userinfo from, therefor we just load in the 2nd comand parameter whit et.trap_Argv(1).

6. To do this, at line 5 write local id = et.trap_Argv(1) --load out the first parameter the command (the userid to dump)

now we got the 2nd parameter as string, the first thing we do now is check if it isnt an empty string ("")

7. So at line 6 write if id ~= "" then --check if parameter1 isnt empty

if its not an empty string then we wil contiune but if it is we will print back an error. We will get to the error later at line 17. Now we will go on whit the next step. We will convert our string to a number this way, we not only get it as a number but, we can also get a nil if it can be coverted so, we can see if the 2nd parameter of the command is a real number and not a word.

8. so at line 7 write local idnumber = tonumber(id) --get a number out of the string

Now we will check if its a real number and not a nil. Todo this we can simply check if its 'true' since only 'nil' and 'false' are 'false' in lua. If its 'false' we will print back an error but we will handle that at line 15.

9. Todo this we will write at line 8 if idnumber then --check if its not a nil

Now we go on assuming we got a normal client slot id so we will try to parse his name. there are 3 ways to parse a name but i use the most simpel one. the other ways are to get it out of a configstring or userinfo. but we will use the et.gentity_get() on the 'pers.netname'

10. So at line 9 we write local idnickname = et.gentity_get(id,"pers.netname") --get the player name for player slot "id"

Now if the person entered a wrong number where there is no client connected then we get an empty string so first we now check if our string is not empty. if its empty we will print an error but we handle that at line 13

11. At line 10 write if idnickname ~= "" then --check if the username isnt empty

Now we go on. We got a username so wut we can do next is make a server command whit it

12. At line 11 write local servercommand = "dumpuser " .. idnickname --create the command to send back to the server

Oke now its ready to send to the server as a command this is done by et.trap_SendConsoleCommand()

13. So at line 12 write et.trap_SendConsoleCommand(et.EXEC_APPEND,servercommand) --send our command to the server tobe executed

Now we are done all that left now is to close the stuff up and add the error messages for when one of our if statements return false. first we add the error message for the ifstatement of line 10. we make an else whit a et.G_Print()

14. at line 13 write else et.G_Print("Player " .. id .. " is not on the server\n") --warning message if the used a wrong slot number

Now Close the if at line 10

15. at line 14 write end --close the if at line 10

Now the error message for the if at line 8

16. at line 15 write else et.G_Print("invalid player slot, useage: dumpuserid \n") --print out a warning that they have to enter player slot

Now Close the if at line 8

17. at line 16 write end --close the if at line 12

Now the error message for the if at line 6'

18. at line 17 write else et.G_Print("invalid player slot, useage: dumpuserid \n") --print out a warning that they have to enter player slot

Close the if at line 18

19. at line 18 write end --close the if at line 6

If u look carefullu u will see that all the thing we did so far are all whitin the first if statement that checks if the command is "dumpuserid". We are about to close that if statement but first we got to make sure this command does not get handled by the server since, we already handled and the server would just say "Uknown Command dumpuserid" anyway. in order to stop it from going to the server its command list we will return 1

20. So at line 19 u write return 1 --return 1 to make etpro ignore the command so u wont get an 'invalid command' warning

Now we can close the if at line '4'

21. At line 20 write end --close the if at line 8

For all other commands we want the server to decide wut todo whit it not our script there we will return 0 to just let the server handle it

22. At line 21 write return 0 --if the command wasnt 'dumpuserid' then we will get here and return 0 to make etpro handle it

Now all that left is to close the function

23. At the line 22 write end --close function et_ConsoleCommand() at line 5

Now if you did it right you should have something like this:

```
function et_ConsoleCommand()--called when someone types a command in conosle or uses sa todo a server command
  local cmd = et.trap_Argv(0) --load out the first parameter of the test typed in console (the command itself)
  cmd = string.lower(cmd) --convert it to lower case so its easier to match in our if statements
  if cmd == "dumpuserid" then --run our code if the command was 'dumpuserid'
    local id = et.trap_Argv(1) --load out the first parameter the command (the userid to dump)
    if id ~= "" then --check if parameter1 isnt empty
      local idnumber = tonumber(id) --get a number out of the string
      if idnumber then --check if its not a nil
        local idnickname = et.gentity_get(id,"pers.netname") --get the player name for player slot "id"
        if idnickname ~= "" then --check if the username isnt empty
          local servercommand = "dumpuser " .. idnickname --create the command to send back to the server
          et.trap_SendConsoleCommand(et.EXEC_APPEND,servercommand) --send our command to the server tobe executed
          else et.G_Print("Player ".. id .." is not on the server\n") --warning message if the used a wrong slot number
        end --close the if at line 14
      else et.G_Print("invalid player slot, useage: dumpuserid \n") --print out a warning that they have to enter player slot
        end --close the if at line 12
      else et.G_Print("invalid player slot, useage: dumpuserid \n") --print out a warning that they have to enter player slot
        end --close the if at line 10
    return 1 --return 1 to make etpro ignore the command so u wont get an 'invalid command' warning
  end --close the if at line 8
return 0 --if the command wasnt 'dumpuserid' then we will get here and return 0 to make etpro handle it
end --close function et_ConsoleCommand() at line 5
```

Now save and load your script and you will be able to use dumpuserid as a server command.
