

---

Subject: Lua V5 (0.5.0) Released

Posted by [jnz](#) on Sun, 16 Aug 2009 22:25:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

At long last, after countless hours. It is finally here.

First of all, for anyone who have not followed the LuaPlugin. This is a plugin for SSGM, built on scripts 3.4.4 to integrate a Lua interpreter. There are over 300 of the API functions bound to Lua and many many hooks.

If you are interested in learning Lua, I can provide some very good links.

<http://www.lua.org/pil/> - a good tutorial detailing almost everything you need to know

<http://lua-users.org/wiki/TutorialDirectory> - a good reference

Now, on to what's new.

I have upgraded the functions calls. The LuaPlugin will no longer ignore function calls with incorrect parameters. It instead throws an error.

I have upgraded the code that handles sockets.

I have also changed the way Vector3 works. It is now a table instead. The details are in the changelog.

Full changelog: (Please read this for vital changes which will affect your old Lua code.)

11 July 2009

Started changes log

Added support for new hooks

function Serial\_Hook(PlayerId, Serial)

Called when the player's client sends his serial hash (must be requested)

function Loading\_Hook(PlayerId, Loading)

Called when a player is loading a map, and then when the player is ready.

Not called when the player first joins the game

function Damage\_Hook(PlayerId, Damager, Target, Damage, Warhead)

Called when a player does damage,

works on everything except buildings

function Ping\_Hook(PlayerId, Ping)

This is called whenever the client sends a "ping" to the server

Ping is the unique number they have sent.

function Suicide\_Hook(PlayerId)

This is called when a player pressed the "suicide" button.

function Radio\_Hook(Team, PlayerId, a, Radioid, b)

This is called for radio commands players can send to their team.

The OnChat function has been changed  
function OnChat(int PlayerId, int Type, Message, Target)  
The new parameter target is for private messaging.

Some new hooks can also be blocked  
You simply add "return 0" to stop execution and block  
For example  
function OnChat(PlayerId, Type, Message, Target)  
if Muted[PlayerId] == 1 then --this player is not permitted to use chat  
return 0  
end  
end

The full list of function available to block are:  
Damage\_Hook  
OnChat  
Suicide\_Hook  
Radio\_Hook

Added a new function RequestSerial(PlayerId)  
This requests the player's client to send his serial  
When the serial is sent the Serial\_Hook is called

Updated ExamplePlugin.Lua to reflect changes to Hooks

12 July 2009  
Fixed small bug where the blockable hooks would always block.

Added support for sqlite3  
Using it couldn't be any more simple  
Define a callback function:  
function sqlite\_cb(userdata, argc, data, name)  
argc is the total amount of columns  
data contains the cell data for each column  
name contains the call name for each column  
data and name are both arrays so you can iterate through them with  
for k,v in pairs(data) do  
io.write(v .. " - " .. name[k])  
end

The callback function is called for each row of the result

To open a database file, just load the class 'sqlite'  
local db = sqlite("mydb.db")

To query the database, use db:Query("data", "callback", query)  
db:Query("userdata", "sqlite\_cb", "SELECT \* FROM mytable;")

Fixed bug preventing OnCharacterPurchase, OnVehiclePurchase and OnPowerupPurchase being called

Fixed a crash bug when Get\_Translated\_String was called with a valid preset.

Changed Get\_Translated\_String, it now needs 2 arguments.

Get\_Translated\_String(Team, PresetId)

Added Get\_Health

Added Get\_Max\_Health

Added Get\_Shield\_Strength

Added Get\_Max\_Shield\_Strength

25 July 2009

Fixed a bug where the server would crash when something other than a string was passed to a function expecting a string.

Added LongToIP, this function takes a number and returns an IP address. For example:

a1,a2,a3,a4 = LongToIP(1234567)

Added IPToLong, this function takes 4 parameters and returns a string.

Added Is\_A\_Star, this takes a gameobject and returns true or false.

27 July 2009

Made a slight change to all the Lua function bindings, they no now longer ignore calls that are not made with the correct amount of arguments.

Instead, they now throw an error.

29 July 2009

Added Enable\_Stealth(Object, Stealth)

Object is the object to effect

Stealth is either 1 or 0, 1 enables and 0 disables.

05 August 2009

Added new hook OnError(Error)

This hook is called whenever there is an unhandled error in Lua

It is not called across all LuaPlugins loaded, only for the LuaPlugin with the error.

This hook is also available in LuaScripts.

Updated example script and plugins with the new OnError call.

07 August 2009

Added Get\_Rotation()

This gets all the maps in rotation  
Returns a table

Removed SLNode and all associated classes.

Added Get\_All\_Objects  
Returns a table with all objects in-game.

Added Get\_All\_Players  
Returns a table with all players in-game.

Added Get\_All\_Vehicles  
Returns a table with all vehicles in-game.

Added Get\_Buildings  
Returns a table with all buildings in-game.

08 August 2009

Display\_Nod\_Player\_Terminal\_Player(GameObject)  
Displays a Nod purchase terminal for GameObject, must be a player

Display\_GDI\_Player\_Terminal\_Player(GameObject)  
Displays a GDI purchase terminal for GameObject, must be a player

Added Reload\_All\_Plugins()  
Reloads all plugins.

09 August 2009

Added Set\_Health(GameObject, Amount)  
Sets the object's health

Added Set\_Shield\_Strength(GameObject, Amount)  
Sets the object's shield strength

Changed the way a Vector3 works. It is now a simple Lua table.  
local pos = {X = 10, Y = 40, Z = 20}

--OR

```
local pos = {}  
pos.X = 10  
pos.Y = 40  
pos.Z = 20
```

Added events Load and Unload  
Updated example plugin to reflect this.

Major change to LuaScripts.

I have now completely dropped the folder "LuaScripts".

To create a lua script you now first create a table like so:

```
myscript = {}
```

Then add your callbacks like so:

```
function myscript:Created(ID, o)
  print("CREATED")
  Start_Timer(ID, o, 10, 0)
end
```

```
function myscript:Timer_Expired(ID, o, num)
  print("TIMER_EXPIRED")
end
```

Simply register by using Register\_Script  
Register\_Script("myscript", "", myscript)

Now you're ok to attach it:

```
local o = Create_Object("Invisible_Object", {X = 0, Y = 0, Z = 0})
Attach_Script_Once(o, "myscript", "")
```

Updated Example plugin to reflect new changes.

13 August 2009

Added basic remote server uploading capabilities.

With 3 functions:

Upload

Starts an upload

Consult the example code to see what parameters it needs.

Check\_Upload

Returns 0 - 5

Consult the example code to see what the numbers mean.

Check\_Upload\_Error

Gets the system error (if any) from the last upload attempt.

Sample code:

```
local t = {}
t.Host = "ftp.myhost.co.uk"
t.Port = 21
t.Username = "joebloggs"
t.Password = "amazingpassword"
t.RemoteFile = "/foo/bar/test.txt"
t.LocalFile = "C:\\test.txt"
```

Upload(t)

```
local check = Check_Upload()
if check == 1 then
  --upload still in progress
elseif check == 2 then
  --Failed, most likly the computer doesn't support uploading
  local error = Check_Upload_Error() -- for the system error code
elseif check == 3 then
  --Failed, most likely unable to connect
  local error = Check_Upload_Error() --for the system error code
elseif check == 4 then
  --Failed, most likely the local file was not found, or the remote file could not be accessed
  local error = Check_Upload_Error() --for the system error code
elseif check == 5 then
  --Upload complete
end
```

14 August 2009

Fixed Create\_Script\_Zone

It now needs 3 arguments

It no longer uses the BoxClass

--This creates a script zone with preset "Script\_Zone\_All"

--At (1 , 2, 5)

--It is a 5x5x5 box

local Center = {X = 1, Y = 2, Z = 5}

local Extent = {X = 5, Y = 5, Z = 5}

Zone = Create\_Script\_Zone("Script\_Zone\_All", Center, Extent)

15 August 2009

Added cPlayer(PlayerID)

This builds a new table and returns it.

The table holds:

AlliesKilled

ArmHit

ArmShots

BuildingDestroyed

ClientUpdateFrequency

CreationTime

CreditGrant

CrotchHit

CrotchShots

DamageScaleFactor

Deaths

EnemiesKilled

FinalHealth  
Fps  
GameTime  
HeadHit  
HeadShots  
IpAddress  
IsActive  
IsHuman  
IsInGame  
IsWaitingForIntermission  
JoinTime  
Kills  
KillsFromVehicle  
LastDamaged  
LastDamager  
LegHit  
LegShots  
Money  
Ping  
PlayerId  
PlayerName  
PlayerType  
PowerupsCollected  
Rung  
Score  
SessionTime  
ShotsFired  
Squishes  
TorsoHit  
TorsoShots  
TotalTime  
VehiclesDestroyed  
VehicleTime

#### Added The\_Game

This builds a new table and returns it.

The table holds:

IntermissionTimeLeft  
IsAutoRestart  
Port  
IsPassworded  
MapNumber  
TimeLimit\_Minutes  
MinQualifyingTime\_Minutes  
IsTeamChangingAllowed:  
Owner  
FrameCount  
Password

TimeRemaining\_Seconds  
WinnerID  
GameStartTime  
GameDuration\_Seconds  
WinType  
UseLagReduction  
StringVersionsMatch  
SpawnWeapons  
INI  
IsQuickMatch  
RemixTeams  
RadarMode  
MVPName  
MaxWorldDistance  
MVPCount  
IsLaddered  
IntermissionTime\_Seconds  
IsFriendlyFirePermitted  
IsDedicated  
GameplayPermitted  
DoMapsLoop  
MapCycleOver  
Motd  
ModName  
GameTitle  
CFGModTime  
DriverIsAlwaysGunner  
MaxPlayers  
CanRepairBuildings  
MapName  
CurrentPlayers  
IsClanGame  
GrantWeapons  
SettingsDescription  
IP

Added Set\_cPlayer(PlayerId, table)

Using any combination of the returned keys from cPlayer,  
You can set a player's cPlayer information.

Added Set\_The\_Game(table)

Using any combination of the returned keys from The\_Game,  
You can set the game information.

16 August 2009

Added MemoryWrite(Address, Table)

Do not use this function unless you know exactly what you are doing.



This function writes code directly into Renegade's memory space

The table is the op-codes to be written at Address

Warning: This function can cause the server to crash

Added MemoryRead(Address, Amount)

Do not using this function unless you know exactly what you are doing.

This function reads code directly from Renegade's memory space

It returns a table with the op-codes read from Address

Warning: This function can cause the server to crash

Added Set\_Shield\_Type(Object, Type)

Sets the shield type of Object.

EDIT: Please see the official release topic for any critical updates.

<http://www.dcomproductions.com/forums/viewtopic.php?f=14&t=1594&p=12635>

---