

META MOD 2.0 GUIDE VERSION 1.0 (01.2017)

Content:

1. Meta Mod General Information:
 - 1.1 important new features in Meta Mod 2.0:
 - 1.2 current limitations of the mod:
 - 1.3 installation:
 - 1.4 installation of external mod scripts
2. C&C SAGE ENGINE AND META MOD LUA MODDING GUIDE:
 - 2.1 Lua general information:
 - 2.2 Rapid prototyping and testing lua code:
 - 2.3 Possibilities:
 - 2.4 Limitations:
 - 2.5 Using parameters of C&C sage engine lua functions:
 - 2.6 Info on some special Meta Mod functions
 - 2.7 map integration guide:
 - 2.8 LUA IO:
 - 2.9 creating lua mod scripts for the Meta Mod external mod script loader
 - 2.9 lua related coding on the xml side:

MetaMod General Information:

1.1 Important new features in Meta Mod 2.0:

- autostart config file: With over hundred options, this file determines your games behaviour: From gamemodes to camera and performance setup you can configure the game in any way you like it. You can change it manually by openening and editing the file with a texteditor or by using the meta command menu -> edit autoconfig
The changes apply after each new map load or map restart, so you don't have to restart the whole game again.
- external mod scripts loader: Framework that enables you to add new lua based mod scripts to the game. You simply place the script file to the MetaModIO workfolder. Then you can activate it in the meta command menu or enable an option so that it automatically loads the existing mod scripts.
- meta command menu: An ingame textual menu, that provides hundreds of options to setup and manipulate the game, or just play around. It gives you full control over a large set of script and engine mechanisms as well. Main categories are:
 - mixed commands
 - external scripts
 - gamemodes
 - edit autoconfig
 - spawn
 - templates
 - ai control & builder
 - unit actions
 - team actions
 - camera setup

- statistics
- selected unit info
- map information & manipulation
- meta shop actions
- diplomacy & more
- counter
- performance
- music, sound, movies
- other
- write lua command
- modinfo

Under "write lua command" you can write and execute lua code live in game (like a command console in other games). Although for rapid live code testing there's a much more convenient way, also described in this guide.

■ new gamemodes

- native art of defense/tower defense: This is one key feature of the mod. It uses advanced spawn mechanisms and self learning evolutionary AI components for a variative gameplay experience. The AI adapts to the players strength and tries to find a balance state, that isn't too easy but also not too hard to play against. In general the spawn rate follows a logistic growth model but the AI can use other math growth formulas as well and is also able to evolve it's personality and change it's behaviour dynamically. Random factors are an important part of the algorithms, so that you never know what's happening next. Many different spawn round types and events are included:
 - Spawn round types: StandardSpawner (mixed spawner), RandomUnitTypeSwarm, RandomUnitCategorySwarm (Tier1 to Tier4), StormriderSwarm, FanaticSquadSwarm, LowHealthAvatarSwarm, BoostedAvatarsSwarm, AlienCustomizableVehicleSwarm, HammerheadZoneRaiderSwarm (ZoneRaiders inside Hammerheads), Endboss
 - random events: UnitAmbush (spawns at certain hotspots), DefenseStructureAmbush, TeamHealthOverride
 - rare random events: RandomSuperweapon, TeamHealthOverrideExtreme

To use this gamemode you can choose ANY map you like and you don't even need to setup an enemy AI slot. The mod will automatically detect the best spawn point for the attack waves and use the creeps team as your enemy.

There are two variants to play this gamemode:

1. timed rounds: Each new attack wave occurs after a certain ammount of time, if all enemies of the current round have been eliminated. You can also configure the time between waves and the first wave delay.
2. user initiates next wave: You have to trigger the next wave by a button. With this variant money will be dynamically provided by the mods algortihms. All tiberium is permanently removed from the map. This mode is good for players who don't like time pressure.

Note: This gamemode is in an alpha version and propably needs more fine adjustment for my standards.

- random army gamemode: Each player gets a group of random units and needs to fight and survive. Last man (team) standing wins. Base building disabled. The units you get is all you will have.
- unit vs. unit: Each player gets one unit (chooseable in the autostart config). Last women (team) standing will win.
- hero general: Each player gets an additional super boosted unit (the hero general,

chooseable in the autostart config). Victory condition: Kill all enemy hero generals and let your own survive.

AI included in all gamemodes. Activating a certain gamemode is very easy: Use the meta command menu (special power button on the left opens it) and go to "edit autoconfig" -> gamemodes, then activate the respective entry and restart the map. Alternatively you can edit the autostart config manually -> meta command menu -> mod info -> open MetaModIO folder -> edit the AUTOSTART_CONFIG.lua with a texteditor.

Combinations of gamemodes are possible in most cases.

■ diplomacy system:

All official maps and Meta Mod maps supported. For other custom maps you can implement the Meta Mod Map integration scripts in just 1-2 minutes.(implementation described in this guide). Otherwise it's a bit buggy cause the alliance change only applies to the current available team. But all other options will work fine.

As a team commander you can directly use "request allied help" and "move attack command" and use team merge operations without waiting for confirmation from other players first. To become a team commander you can apply and then all human players need to agree via appearing accept/deny buttons. For AI players you are always considered a team commander.

The diplomacy tab menu looks like this:

Previous player	Next player	Show all relations
Request alliance	Break alliance	Send money
Apply for team commander	Request allied help	Move attack command
Temporary merge allied team	Temporary merge own team	Switch team
Take control of allied team	Take control of next allied team	

■ meta stats system with xp and rank system, all stored to savegames:

reward type		
Meta Credits (MC)	condition	ammount
	gaining XP	XP*1.75/1000
	each art of defense/ tower defense gamemode round	1
	each new highscore round for art of defense/ tower defense	1
	art of defense/ tower defense game over	14
	common victory	14

	every 3 minutes playing (any gamemode)	1
Experience Points (XP)	condition	ammount
	each art of defense/ tower defense gamemode round	1000 + dynamic value (depends on wave and enemies killed)
	each new highscore round for art of defense/ tower defense	2000
	art of defense/ tower defense game over	5000
	common victory	10000
	every 3 minutes playing (any gamemode)	1000
	meta shop item use	1200

■ meta credit shop with 26 items:

ITEM	COST MetaCredits(MC)	RANK	comment
nuke	15	10	
rift generator	15	10	
ion cannon	15	10	
super avatar	10	13	all upgrades and highest veterancy
BaseTemplate	10 + dynamic value	14	you can create your own base templates
circle defense line	12	9	
spiral defense line	12	9	
random square defense	12	8	
random anti ground defense block	7	7	
random Anti Air Defense Block	7	7	
anti Vehicle Def Start Point	-	7	
antiVehicleDefEndPoint	5 + dynamic value	7	you get a price offer first, that you can accept or deny
indestructible for 30s	9	12	
infinite energy for 30s	5	6	
slow down enemy for 30s	5	2	

infinite production speed for 30s	5	11	
build anywhere for 30s	7	8	
convert MC to Dollar	10	15	10MC=5000\$
reveal map for 10s	3	4	
get statistics	4	3	player stats, unit counts and more
guard shield emitter	35	15	
boost selected unit	3	1	
set timetravel fixpoint	1	16	
timetravel to fixpoint	44	16	
spawn 7 random units	5	4	
bet	10	0	win up to 300MC

- 14 new maps:
e.g.:
 - 2 Meta Mod adaptations: Simple Brutal Tower 1.9 by nevrose and world domination by mungovin (good for diplomacy)
 - gamemode specialized maps: MetaMod-Arena1 for unit vs. unit and random army gamemode. MetaMod-TowerDefense1, a beautiful tower defense map with lush forests and nature.
 - Engine script hack exploit map: MetaMod-RemoveZoomLimitation, run this map one time and on every other map you load the camera height zoom limitation will be gone.
- first and third person mode: New special power button that toggles between 3 view modes for the unit you have currently selected. There's also a special "psychedelic" version in the meta command menu. Command unit then via radar map or via meta command menu->unit actions->unit camera and control->navigation mode, then use the meta menu navigation buttons to move around (alpha version)
- Camera adjustment: New special power button to set the camera height (no limit) or unlock it again.
- Meta control tab menu: From here you control the meta command menu with arrow buttons. Beside that you have many additional quick control buttons, which can also be found in the meta command menu under "mixed commands".
 - AI boost toggle buttons
 - AI toggle buttons
 - quick spawner (set the unit type in the autostart config)
 - camera adjust button as well as first and third person mode button
 - toggle indestructibility
 - more...
- misc:
 - most old meta mod elements still implemented: meta tech faction, meta control bar option, secondary mission in global conquest mode disabled , more money start-options (5k up to 1 million, 5k steps) , energy shield generator, sending money to other players,...
 - For a full list of the old features look here:
<http://www.moddb.com/mods/mjjstral/downloads/meta-mod-101>
 - "no superweapons" checkbox instead of "random crates"

- and many more things...

1.2 current limitations of the mod:

- multiplayer support: There's a slight chance, it could work if all players have equal MetaModIO input files, especially the autostart config. But I cannot guarantee asynch free games currently.
- for most displays of text including the meta command menu you will see "MISSING:" on top of the text. I'm not sure if I can solve this, but I think we can live with that in exchange for all the possibilities. Also there's a little delay navigating through the meta command menu.

1.3 Installation:

Place the MetaMod_2.00 folder to the Mods folder in your C&C Kane's Wrath documents folder and use Wrath mod launcher from bibber.

1.4 Installation of external mod scripts

Place the respective script.***.lua file to your User/Documents/Command & Conquer 3 Kanes Wrath/MetaModIO folder. Note that parts of this folder path are language specific.

Next you have two options to enable it ingame:

1. Open the AUTOSTART_CONFIG.lua in the MetaModIO folder (if there is no, start a game first and let the mod create one). You will see an entry like this:

```
ModScripts["filelist"]=  
{  
  "script.ExampleScript1.lua",  
  "script.ExampleScript2.lua",  
  "script.ExampleScript3.lua"  
}
```

Now just add the filename of the new modscript inside there. Note that the last element doesn't end with a comma. To enable it (if it needs to) open the either the modscript file and set e.g.

```
ModScripts["status"]["script.ExampleScript.lua"] = 0
```

```
to ModScripts["status"]["script.ExampleScript.lua"] = 1
```

or enable the script much easier in the modscripts submenu of the meta command menu ingame.

2. Use the autoloader option modscripts submenu of the meta command menu ingame. It will create a list of all modscripts available in the MetaModIO folder. You can toggle each modscript status seperately in the list then. Note that the game will minimize and maximize quickly if the autoloader is active.

C&C SAGE ENGINE AND META MOD LUA MODDING GUIDE:

2.1 Lua general information:

C&C3 Tiberium Wars and Kane's Wrath use Lua version 4.0.1. For syntax and descriptions of all functions and their parameters use the official lua 4 reference (refman-4.0.pdf). To find syntax errors use luac4.exe, included in the meta mod package. For example a bat script can look like this:

```
@echo off
luac4.exe "C:\LUA_INPUT.lua"
pause
```

You can choose out of a variety of 933 different game related lua script commands to create even very complex game mechanics. Lua itself provides 95 native lua functions. Additionally there are dozens of new very powerfull functions that come with Meta Mod 2.0.

These important lists come with the Meta Mod package archive:

1. **ALL_SAGE_LUA_COMMANDS.txt** : This is a complete list of all lua functions/commands/variables natively available in the C&C3 sage engine as well as lua native functions,syntax and a lua event script list. The list includes many additional ExecuteAction and EvaluateCondition commands, that you cannot find in the scripts.ini file. It may also be usefull for the Lord Of The Ring game series made by EALA. How to use the correct paramters for C&C sage functions gets described in this guide. For iformation on how to use lua native functions use refman-4.0.pdf.
2. **METAMOD_LUA_REFERENCE_LIST.txt** : List of most functions, variables and tables introduced in Meta Mod 2.0 by Mjjstral.
Look into the actual source code for more details on parameter types and for use cases. In many cases function arguments are optional. You can skip optional arguments by using nil or simply let it be void. Depending on the context most arguments are strings, tables or numbers.

2.2 Rapid prototyping and testing lua code:

Open the game with start parameter -win. Open the LIVE_INPUT.lua and LIVE_OUTPUT.txt and let the texteditor stay next to the game window. Then use the "Run Lua Script" button in the meta control menu or use Force_Lua_Script_Button = yes to get a special power bar button for this. You can simply enter your code now into LIVE_INPUT.lua and also overwrite existing functions. Use the ErrorLog.txt to find errors. I also recommend Dev_Mode = yes. To output anything to the LIVE_OUTPUT.txt use the out(yourtext) and use print(yourtext) to show text ingame.

2.3 Possibilities:

- use of string parameters for action scripts (also for ObjectType)
- use of real/float parameters in action acripts
- use of dropdown list parameters (by use of index-1 number)
- use of team parameter for action scripts
- use of bool parameters (true/false) (by use of 0 and 1)
- file access and manipulation (savegames, autostart config, external modscripts)
- play custom movies, radar movies, sounds and music
- object attribute manipulation and storage in any way
- information gathering by use of evaluatecondition: statuses, kindofs, objecttypes, number...
- custom victory conditions
- counters as well as timercounters
- scripttimers for scheduling scripts

2.4 Limitations:

- use of integers for some action scripts (few exceptions)
- use of player parameter: most actions exist for team as well, so we can live with that ...
-> workarounds available in this guide
- use of coordinates for action scripts: ->workarounds available: SpawnAtPosition(...) func.

2.5 Using parameters of C&C sage engine lua functions:

all C&C sage engine lua function not ExecuteAction or EvaluateCondition commands:

These Only take object table references (except ExecuteAction commands), so use GetObj.Table(Object) to be sure. Objects without lua registration in the globals() table cannot be used with these functions.

ObjectParamterTypes:

- Table Reference:
These Objects are stored in the lua globals() table (like all lua functions and global variables) and are also of lua value type "table". They all have a distinctive index name beginning with "OBJID", so we can filter them easily. Table References for all objects which have an AILuaEventsList entry in the AIUpdate module in the object xml definition are stored in the globals table. You can force a registration if you use ForceLuaRegistration="True" in the xml object definition.
- String Reference:
All Objects can have multiple string names related to them. These references can then be used for all ExecuteAction and EvaluateCondition commands which take object names.

OtherParamterTypes:

- Team names:
Use GetTeamName(Object)
MetaMod team name variables: ClientTeamName, NeutralTeam, CreepsTeam, CivilianTeam, ObserverTeam
MetaMod team tables: GlobalTeamTable, HumanPlayerTable, AIPlayerTable, AllTeamsTable
For the actual names look into the Meta Mod source code (scripts.lua).
- Waypoints: WaypointTable, WaypointDistanceTable
- Areas: AreaTable (unfiltered)
- bool parameters (TRUE/FALSE): Use 0 or "0" for false and 1 or "1" for true.
- dropdown list parameters (by use of index-1 number) : If a dropdown menu appears for the parameter selection in world builder then you have to use the index-1 as a number or string on the lua side. most lists included in metamod as lua tables and ready to use.
- Strings: Use lua strings as parameter for all parameters which you can freely change in the textbox in world builder.
- Real or double parameters: You can use a lua number and even a string that contains the number.
- Integers: Except for relational operators there's doesn't seem to be a way to use integers sometimes. All numbers from lua are converted solely to doubles to the C side of the lua C API.
- Coordinates: Doesn't seem to work in lua.
- Player: There's no known way currently using the player parameter. Internally the player reference seems to be a struct. Actually the player reference consists of the script folder name and the player name: "WBScriptFolder/Playername", e.g. "Player_1/<This Player>". The good news is for most actions or conditions you can use similar ones with the team parameter. Another workaround to use the player parameter indirectly is to use the SendScriptHostCodeMessage method described in the map integration guide part.

ExecuteAction and EvaluateCondition:

These are the most valuable functions available. All the cool stuff can be achieved with these functions. Using these functions extensively is one of the main characteristics of Meta Mod. ExecuteAction let's you use most world builder script actions while EvaluateCondition gives you access to most world builder script conditions.

The game contains 677 ExecuteAction and 218 EvaluateCondition commands you can try to use. You can find a complete list with each and every command inside the MetaMod download archive.

How to get the parameters:

1. Use the lists included in the MetaMod archive to get the first parameter (your actual command).
 2. You can also get the first parameter from the scripts.ini file. But some commands are missing here.
 3. Use the world builders script maker menu to see the order of the parameters. To get the first parameter name you can export a script, open that exported file with a texteditor and extract the raw script name, usually something written in capitals with underscores. You just have to add quotation marks for the use in lua.
- ExecuteAction:
syntax: ExecuteAction(command,parameter1,parameter2,...)
Objects: takes string as well as table references
 - EvaluateCondition:
syntax: EvaluateCondition(command,parameter1;parameter2,...)
example:
Objects: takes only unit reference in string form, use GetObj.String to make sure using the right one.
Relational Operators: Use index number of operator list minus one, or use
CompareTable["operator"]
(operator: >, >=, <, <=, ==, ~=)

2.6 Info on some special Meta Mod functions:

With over 10000 lines of lua code Meta Mod 2.00 provides many usefull functions and gives modders possibilities that even exceed the capabilities of the script editor from world builder. From the hundreds of functions, tables and variables the most important ones are maybe these:

FUNCTION	COMMENT
SetScriptTimer(time,action,loops,condition)	in 1s steps, loops=-1 for infinite, action is a function or string
SetScriptTimerFast(time,action,loops)	in 0.01s steps
SetDelayed(func)	shortcut for ScriptTimerFast with 0.01 delay
spawn(ObjectType,team,number,ref,waypoint,ActionOnObject,OnObjectType)	
SpawnAtPosition(ObjectType,team,x,y,z,ObjectRef,orientation,Waypoint,numbercount,fast,ActionOnObject)	fast: 0=normal,1=fast,2=superfast
GetObjectPosition(Object)	returns a table, accessing coordinates by table.x and table.y
GetBaseRef(team)	returns an object reference of the original base hq
push(Task)	push tasks on the TaskStack that depend

	on games simulation time, Task can be a function or a string
pop(Task)	every Task pushed should contain pop()
GetTeamName(teamobject)	
GetTeamsDescriptiveName(input)	input can be team or object
GetFactionName(input)	input can be team or object
GetStringRefListOfAllObjects(CreepsTeamToo)	
GetStringRefListOfAllObjectsForTeam(team)	
print(output, display_time)	
WriteToFile(output,file,mode,hide) or out(...)	If no file is specified, LIVE_OUTPUT.txt will be used
RandomString(l)	l=length
SetCounter(counter, value, text, no_display)	
SetTimerCounter(value, text, counter)	
delete(thisobject)	
FireWeaponOnObject(Object, Weapon)	
FireWeaponPosition(Weapon,x,y,waypoint,OnObjectType)	
flash(input)	team or object
GiveChoosePositionOption(team,action,action_with_position,waypoint)	
GiveAcceptDenyOptions(team,acceptaction,denyaction,preaction)	
GPOT	GlobalPlayerOptionTable: Use GPOT[teamname][attributename] important attributenames: "teamname", "name", "faction", "ai"
GlobalTeamTable, HumanPlayerTable, AIPlayerTable	
ClientTeamName, NeutralTeam, CivilianTeam, CreepsTeam, ObserverTeam	
Human_Players_Count, AI_Players_Count	

Note: Use the Meta Mod source code for more details and use cases!

For a full list take the METAMOD_LUA_REFERENCE_LIST.txt file.

In many cases function arguments are optional. Depending on the context most arguments are strings, tables or numbers. You can skip optional arguments by using nil.

Spawner and positional information:

SpawnAtPosition function isn't spawning delay free. Spawntime depends on game simulation! Every action that should be applied to a spawned object with this method needs to be applied delayed by use of a scripttimeraction or push(Task) or by use of the last argument "ActionOnObject". If you want to spawn a new mod object that has kindof IMMOBILE, you need to add a spawnextender entry (1. in the xml's, see Object_MetaControlDummyOCLSpawnExtender.xml, 2. In the OCLSpawnExtenderTable lua table. Otherwise the game freezes while spawning. With spawn(...) you don't have these two disadvantages but you can only spawn at waypoints, on

objecttypes and at the 0,0 coordinate.

2.7 MapIntegration Guide:

There are three types of integration:

1. MetaMod can communicate with the map if certain world builder script commands are added to the map. Only mandatory for the diplomacy alliance change options to work without problems: Import the script METAMOD_MAP_SCRIPTS_CUSTOMMAPS.scb into your map. Done.

2. To support the native art of defense/tower defense gamemode in working better you can give certain names to waypoints, paths and objects. This is not mandatory for native art of defense but can support the meta mod algorithms in using a good spawn waypoint and a waypoint path. For tower defense maps it's mandatory to give a certain name to the tower that meta mod will detect and work with.

By using this you can create art of defense or tower defense maps that can use the advanced spawn mechanisms of meta mods native art of defense gamemode. That means you don't have to care about spawning scripts anymore. If you like to change the metamod spawning behaviour you can overwrite the respective functions with an external modscript or by adding it to the autostart.lua file. That means you can distribute your map with an additional art of defense/ tower defense mod script, that overwrites the meta mod function in general or adds code that will only be used for a certain map.

Use the following name conventions for meta mod map integration:

for art of defense and tower defense:

1. create waypoint:

name: METAMOD_NAOD_SPAWN_WAYPOINT (forces gamemode)

alternative name: METAMOD_NAOD_SPAWN_WAYPOINT_OPTIONAL (gamemode activates only if user enabled it)

2. create an optional waypoint path:

path name: METAMOD_NAOD_WAYPOINT_PATH

first waypoint name of path: METAMOD_NAOD_WAYPOINT_PATH_START

for tower defense:

tower name: METAMOD_TOWERDEFENSE_OBJECT

MetaMod will automatically detect the waypoints/tower and start its native art of defense/tower defense gamemode.

you can also use your own spawn scripts if condition "metamod_modstatus_counter" is not equal to 0, so if player has no Meta Mod.

3. LUA "client-script host" communication bridge:

A special case of MetaMod map integration applies to all official maps in kane's wrath. These maps inherit a general script for the global conquest mode. But you can use this to insert your own scripts all in one move. Simply edit the scripts of this map:

libraries/lib_mg_tacticalplay_skirmish/lib_mg_tacticalplay_skirmish.map

Notice that counters in these general scripts are not shared with the script instance used in the lua environment so we need to communicate via SendScriptHostCodeMessage(code,counter) function available in Meta Mod, which is basically converting counter values from the client side via unit experience of a hidden dummy unit to a counter value on the "script host" side.

2.8 LUA IO:

Meta Mod uses a workfolder named MetaModIO which usually sits here:

C:\Users\"USERNAME"\Documents\Command & Conquer 3 Kane's Wrath\MetaModIO

or here:

C:\Users\"USERNAME"\AppData\Roaming\Command & Conquer 3 Kane's Wrath\MetaModIO

and in rare cases here:

...\C&CKane'sWrath GameFolder"\RetailExe\1.2\...

You can also try to find/open the folder via meta command menu -> mod info -> open MetaModIO folder. Game will minimize and open the respective folder.

These files play a role in Meta Mod:

- **1.** "AUTOSTART_CONFIG.lua": This is the main file that gets loaded always. You can insert any lua code in here and it will be executed while the map starts. You can also overwrite Meta Mods lua functions, variables and tables, although a mod script is recommended in most cases.
- **2.** "METAMOD_SAVEDATA": All meta stats, such as rank, xp, metacredits and highscores are stored here. Note that this file is encrypted.
- **3.** "METAMOD_SAVEDATA_BACKUP": As the name indicates a backup, made each time the first save gets generated after a new map reload.
- **4.** "LIVE_INPUT.lua": This is especially very very usefull for modders. You can write lua code in here and then execute it live in game if you use the lua script button in the meta tab menu. Or use Force_Lua_Script_Button = yes in the autostart config for a seperate sepcial power bar button. This mod wouldn't even be created without this method.
- **5.** "LIVE_OUTPUT.txt": This is output that gets generated while using WriteToFile(...) function or it's shortcut out("text"). Also most options in the meta command menu under "other" will redirect their output into this file. You can output whole object lists other usefull stuff for debugging and testing.
- **6.** "templates.txt": This is the file that stores the templates you can create under "templates" in the meta command menu. Each template is a function that holds at least the information about the object types and their (relative-) positions.
- **7.** "ErrorLog.txt": Very usefull for debugging to trace semantic and even syntax errors, especialy for code you run live ingame via "LIVE_INPUT.lua".
- **8.** "PreLoad.lua": LUA code that gets executed before any other Meta Mod function. Unused, just for debugging purpose.
- **9.** mod script files: All files used for the external mod script loader framework. See next chapter for

creating lua mod scripts for the Meta Mod external mod script loader:

All lua sripts beginning with "script." and ending with ".lua" that reside in the MetaModIO folder-> e.g. script.ExampleScript.lua, can get detected and used by Meta Mod. To handle everything relaed to that open the "external scripts" submenu in the meta command menu. There's also an option to create an "ExampleScript", that periodically prints "hello world" to the screen. Use this script as base to create your first own scripts.

Creating your own scripts: First create and test your scripts with help of the "LIVE_INPUT.lua". To quickly find syntax errors use the luac4.exe method described above.

You can also overwrtie any existing function or variable already insde Meta Mod.

I recommend you distribute your scripts here <http://forums.cncnz.com/forum/59-tiberium-wars-kanes-wrath-maps-modding/> or at moddb.

I can also include your scripts in future meta mod releases if you send them to me.

2.9 lua related coding on the xml side:

You have several ways of triggering lua events (scriptevents.xml) that trigger lua functions (scripts.lua):

1. AILuaEventsList (AIUpdateModuleData): This is the standard way of giving an object the ability to trigger a lua function depending on one or more lua events described in the scriptevents.xml

Example:

```
<AI>
  <AIUpdate
    id="ModuleTag_AI"
    AutoAcquireEnemiesWhenIdle="NO"
    AILuaEventsList="ExampleEvent"/>
</AI>
```

The ExampleEvent in the scriptevents.xml can look like this:

```
<EventList Name="ExampleEvent" Inherit="BaseScriptFunctions">
  <EventHandler EventName="OnCreated" ScriptFunctionName="ExampleEvent1_Function" DebugSingleStep="false"/>
  <EventHandler EventName="OnDestroyed" ScriptFunctionName="ExampleEvent2_Function" DebugSingleStep="false"/>
</EventList>
```

You can even create custom events with a model condition or object status:

```
<ModelConditionEvent Name="Selected">
  <Conditions>+SELECTED</Conditions>
</ModelConditionEvent>
```

or

```
<ObjectStatusEvent Name="UnderConstruction">
  <Conditions>+UNDER_CONSTRUCTION</Conditions>
</ObjectStatusEvent>
```

2. LuaEventNugget: The object that gets hit by the weapon calls the lua event specified in the lua event nugget:

Example:

```
<WeaponTemplate
  id="ExampleWeapon"
  Name="ExampleWeapon"
  RadiusDamageAffects="ALLIES ENEMIES NEUTRALS NOT_SIMILAR" >
  <Nuggets>
    <LuaEventNugget
      EventName="ExampleLuaEvent"
      Radius="20.0"
      SendToEnemies="true"
      SendToAllies="true"
      SendToNeutral="true">
    </LuaEventNugget>
  </Nuggets>
</WeaponTemplate>
```

3. Via LuaEvent in DrawModuleData: Triggered when animation is processed. The lua event will then be "OnGenericEvent" and the second argument of the lua function that gets called is a string containing what is written for Data. In this case "ExampleAnimActive". For more examples look into the BFMEII ini sources. Example:

```
<Draws>
  <ScriptedModelDraw
    id="ModuleTag_Draw">
    ...
    <AnimationState
      ...
      <Script>
        AnyLuaCodeHere=true
      </Script>
      <LuaEvent
        Frame="0"
        Data="ExampleAnimActive"
        OnStateEnter="true"
        OnStateLeave="false">
      </LuaEvent>
    </AnimationState>
  </ScriptedModelDraw>
</Draws>
```

4. Meta Mods ways:

- run lua code manually via the "run lua script" button
- run lua code with help of the lua command receiver in the meta command menu (submenu "write lua command")
- use the scripttimer functions
- insert your code into the PeriodicMasterChecker function
- insert your code into the AUTOSTART_CONFIG.lua file

Beside these activation methods there are two more lua related code puzzles:

- ForceLuaRegistration: Set ForceLuaRegistration="True" if you want the object to be available in the lua globals table, if it has no AILuaEventsList code segment.
- DelayedLuaEventUpdate behaviour module, used once in BFMEII:

```
<DelayedLuaEventUpdate
  id="ModuleTag_LuaEventUpdate"
/>
```

I hope you enjoy my mod, and if your a modder I hope you create fullfilling new mods and mod scripts based on this mod and this guide. For discussion I recommend using the CNCNZ forums: <http://forums.cncnz.com/forum/59-tiberium-wars-kanes-wrath-maps-modding/>

Contact: metaidea7@gmail.com

Best wishes

Mjjstral aka. MetaIdea (Maximilian F.)