

Scripting in MoHAA: The Basics; By \$or

(MoHAA patch v1.11)

Part 1 and 2

Contents:

PART 1

- I. Introduction**
- II. Vital Knowledge**
- III. How to use Pakscape/winrar**
- IV. Opening script files**
- V. Closer look at the files**

PART 2

- VI. Let's take a closer look at a script**
- VII. First results**
 - a. Tips**
 - b. Order**

PART 1

I. Introduction

So you want to learn how to script in MoH:AA ? Well it's easy when you know it and challenging to learn it. First of all you must have enough motivation to do this. The 1 thing that scares people off from learning things like this is that they don't get results right away. Now I'm not going to get philosophical, but people with this problem are impatient. The best way is to encourage these people is to give them results first and then explain what they've done. The basics of scripting isn't as hard as they appear to be ;)

NOTE: I use scripting almost every day, even though I'm good at it, it's hard to explain what some things are for. I'll explain as hard as I can in plain English.

Parts

Part one will be about how to open certain files and how MoHAA's file system works. It's about 9 pages, but at least 7 of them are pictures. Part 2 will teach you firstly how a (map)script is put together and works. Part 2 will teach you how to make your first script!!

II. Vital Knowledge

Well first you need some tools.

Winrar/Pakscape (last one's recommended) and wordpad.

Get winrar for free here: <http://www.win-rar.com/>

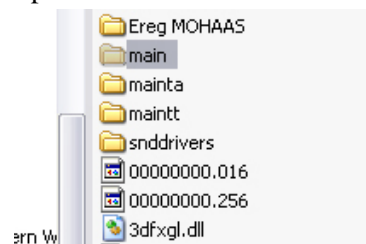
Or get pakscape here: http://medalofhonor.filefront.com/file/PakScape_PK3_Program;39910

Good thing about winrar is that the free version enables you to use ALL functions of the Full version, the catch? Well everytime you start winrar you get a popup window. That's all :D !

First, you need to know how to get to an pk3, find your MoHAA game on your hdd. Default path is:

C:\Program Files\EA Games\Medal of Honor Allied Assault\

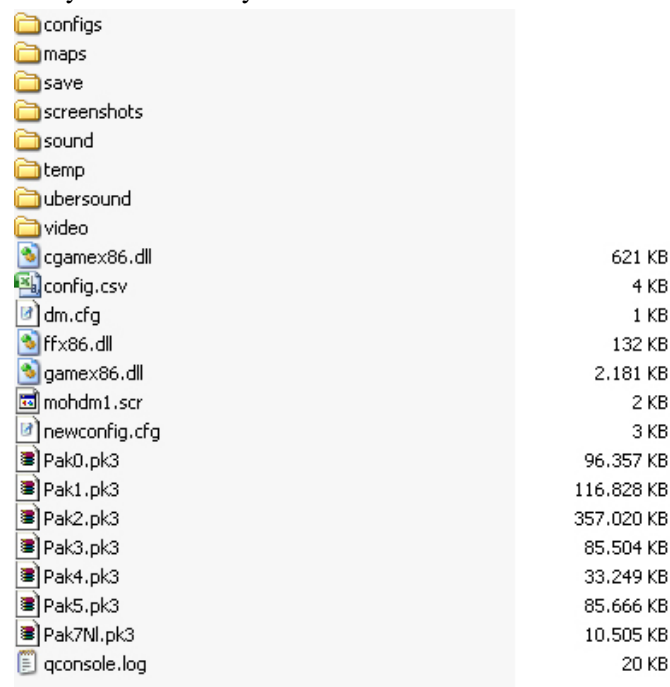
Open the main folder.



Pic1: Main directory, doubleclick to open

This is where all the maps, sounds and files are. Second you need to know how to open .pk3 but first what they are. Pk3 files are bundles of gamefiles that the game recognizes and loads. Inside are the gamefolders. Like maps and sounds. More about this later.

NOTE: The maps folder isn't created during MoHAA's install, either a custom map triggers to create it or you must do it yourself.



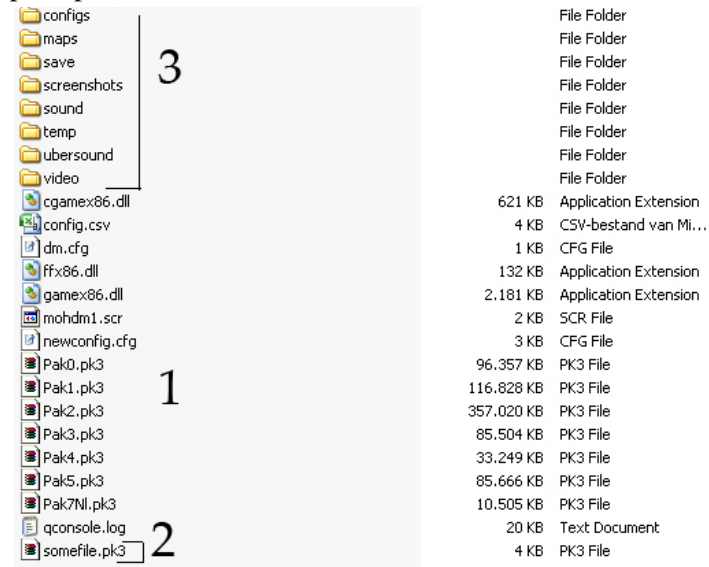
Pic2: example of my main (I cleaned it up; full of custom files lol)

When you open a pk3 file, you may notice some of these folders are also in your main folder itself.

Pk3 files their purpose is to combine are organize game files and mods/maps...etc you download.

Also bsp files (-> maps) sizes are significantly reduced when put in a pk3.

You may have noticed when downloading files, that they begins with a few 'z' characters. It's simple why, MoHAA, loads all files alphabetically. If the mod you downloaded needs to over-ride some of the gamefiles then it should be alphabetically higher than the original files (-> Pak0.pk3 up to pak6.pk3)



Pic3: Nr 1 is the alphabetical lowest file, gets loaded first and the next files over-rides any matching file in nr1. Nr2 is the alphabetical last one and over-rides any matching file loaded before it. Nr3 are folders and have the Highest priority. Meaning they over-ride any matching file loaded before.

For scripters, making a pk3 is only relevant if you're doing a map or when you're finalizing your creation. You see, the folders in the main folders, ALWAYS over-ride any pk3 file. Meaning for testing your script on a stock map, you just have to place it in the right directory in your main folder.

How to open a .pk3??

Choose between winrar or pakscape. Pakscape's most recommended, but I'll show you either way.

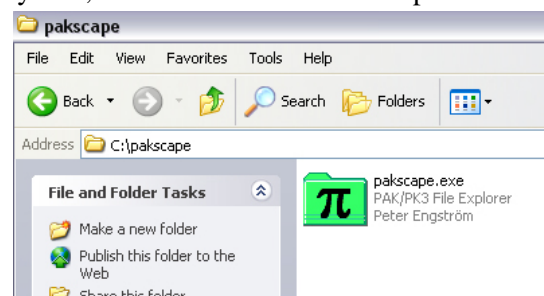
Choice 1:

Once you installed winrar (installation is easy but can be confusing: here's a tutorial

http://www.youtube.com/watch?v=Sk_4a8uBIUA One thing you have to note, here he says if there's a newer version out then download that one. I recommend not using any beta downloads, just download winrar v3.71.) It will tell you to either buy or remove winrar after 40 days. Well don't, because nothing will change, you can use all functions of the full winrar version in your trial version as long as you want, you just get that popup window when starting it.

Choice 2:

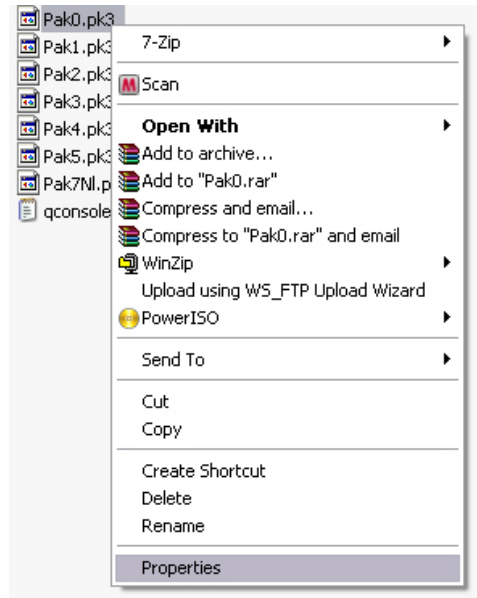
In pakscape's case, download it. Next create a folder in C:\ (or the drive that has windows) and call it pakscape. Move the pakscape program (exe file) from the place you downloaded it to, to this new folder. Pakscape's completely free, no catches. Picture 4 example:



Pic4: Pakscape install folder

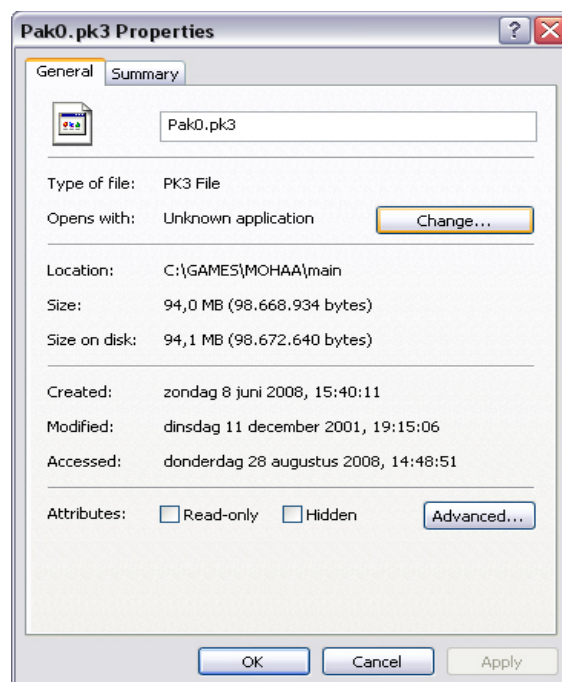
Now in both cases:

Rightclick any pk3 file. Select properties.



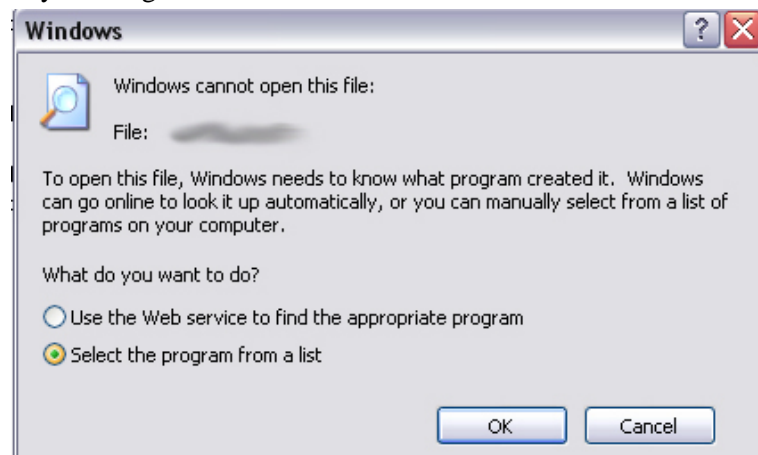
Pic5: Right Click on pak0

A new window pops up. You'll find at the top of the new window: "File type opens with:" Click the "change" button.



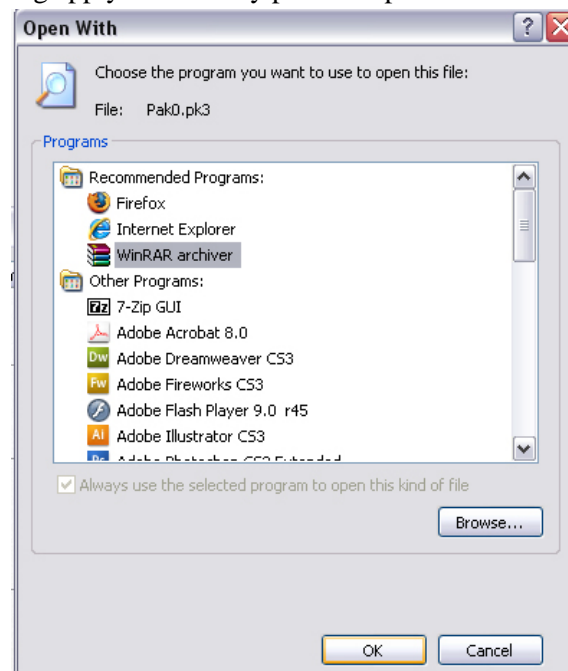
Pic 6: Properties of pak0

Windows sometimes asks you if you'd like him to search a program that can open the file on the internet or if you'd like to choose a program in a list (list of programs you have installed of course). Select the latter one by clicking the checkbox. Press Ok.



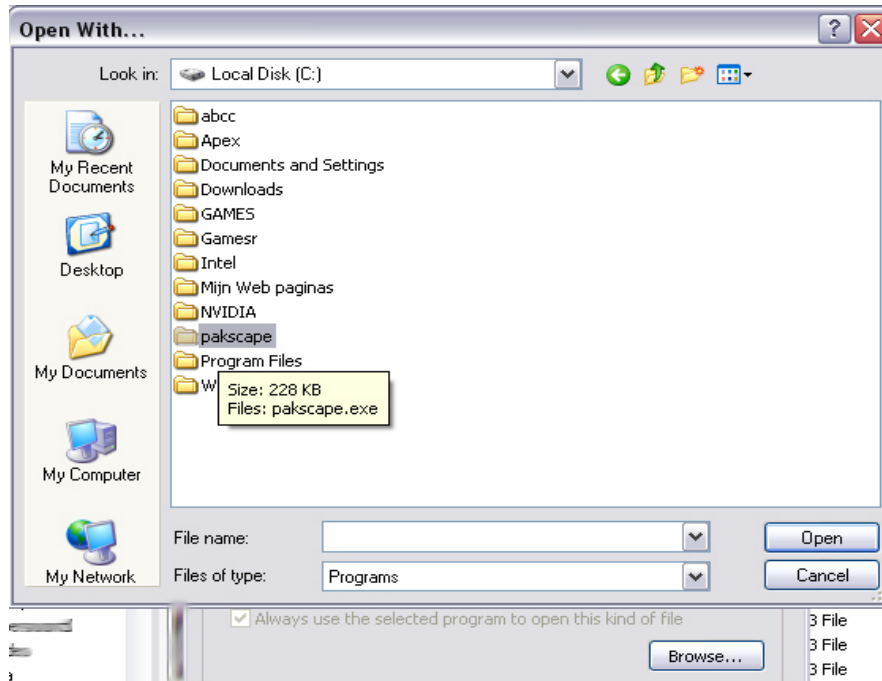
Pic7: Windows appropriate program question box

Then choose WinRAR from the new window in the list. Press 'Ok' or 'Apply' or something. Close the properties window by clicking apply. Now every pk3 file opens with winrar.

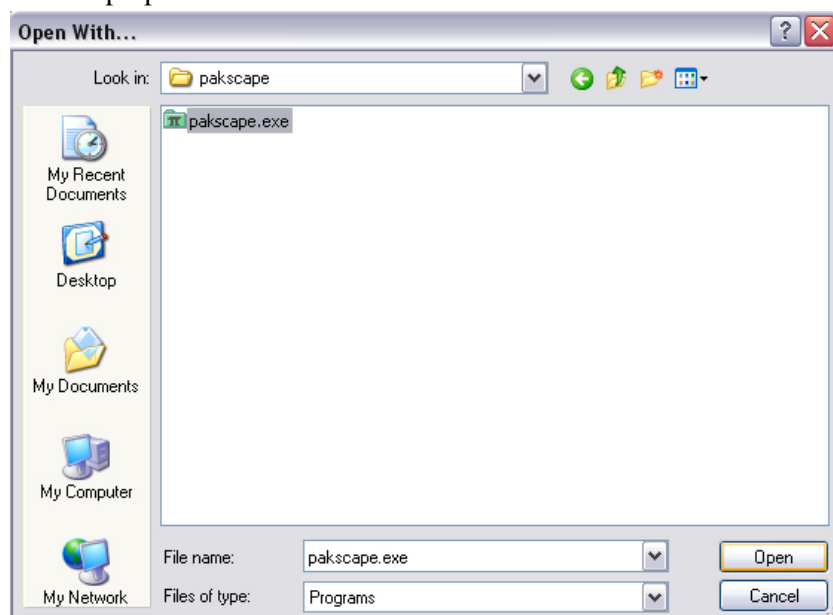


Pic8: Winrar program select

If you're using pakscape, then you should click **browse** in the same window (Like Picture 8, above!), go to the folder where you placed the pakscape.exe file.



Pic9: Pakscape Browse program select; Note the button Browse, that's the one to click. Select pakscape exe in that folder and click, open, then apply (both windows will be gone) and click apply or ok again in the properties window.



Pic10: Select pakscape program

Now all pk3 files will open with pakscape. I personally use winrar because I'm used to it, but I'll show everything (picture) with both. After explaining how to copy, paste etc in both programs, I do not need to explain that for the 2 again ;)

What you've learned

You learned the loading hierarchy of MoHAA.

You know what pk3 files are and what they are used for.

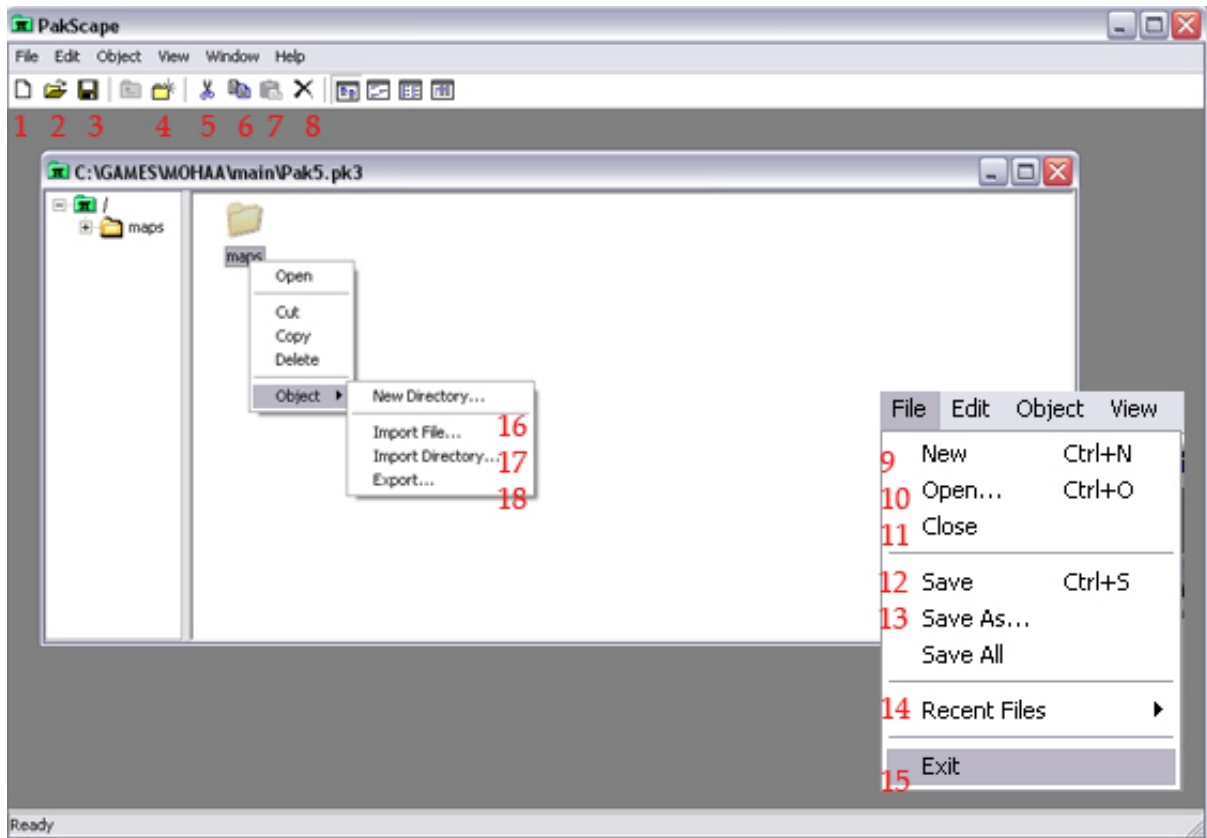
What you've did:

You installed pakscape/winrar and learned how to make pk3's automatically open with them.

III. Using pakscape or winrar.

Here I'll explain how to use winrar and pakscape.

First pakscape: Open pak5.pk3 in your main.

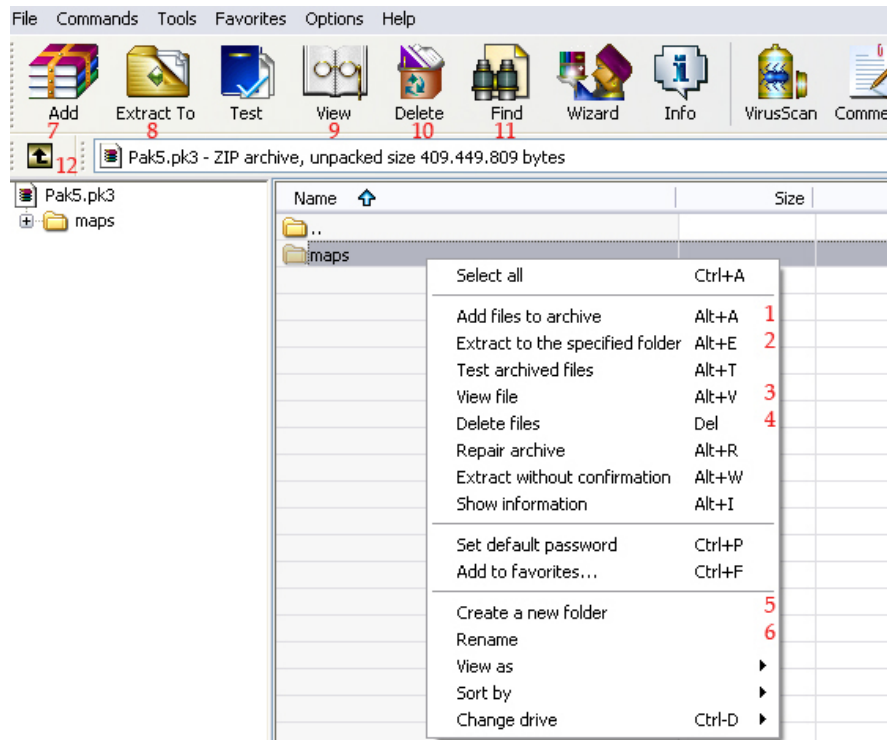


Pic11: Pakscape usage, explanation

It will look like this, note that every command has a number, now I'll tell you what it does.

1. Make a new pk3 file
2. Open an pk3 file
3. Save current work
4. Create new folder (in the folder you're working in !)
5. Cut (file)
6. Copy (file)
7. Paste (file)
8. Delete file
9. Same as nr 1
10. Same as nr 2
11. Closes current pk3 file
12. Saves current progress
13. Saves file as (another filetype? Or filename?)
14. See the recent pk3's you opened with pakscape
15. Exit pakscape
16. Import file (like a scr file)
17. Import a folder
18. Export (extract) a file or folder

Now winrar: Open pak5.pk3



Pic12: Winrar usage, explanation

Note: Winrar uses a different approach, Copying, Cutting and pasting within the pk3 isn't possible. Instead you must drag your file to copy (cannot make a copy in the same folder of the pk3 using winrar) to another directory in the pk3. Moving is also dragging to a folder on your pc or a folder in the pk3 and but it only copies, so you must delete the file you made a copy of to make it look like a move.

1. Add new files to the pk3
2. Extract selected files to a folder
3. Open/view the file in the pk3
4. Delete selected files
5. Create a new folder
6. Rename a file
7. Same as 1
8. Same as 2
9. Same as 3
10. Same as 4
11. Search a file in the pk3
12. Folder-Up

Unmentioned Similarities

Winrar and Pakscape have a lot of similarities. Both have an folder overview on your left. You can easily switch folders,drag, copy, paste etc...

In both programs you can view/edit the files by double clicking them.

In both you can drag a file outside the pk3 to extract it.

Winrar's disadvantages:

- Lack of a proper copy-cut-n'-paste system
- Lack of import feature (*importing is dragging a file into the pk3 window of winrar*)
- Lots of useless commands makes it harder to work with

Winrar's Advantages:

- Almost everybody has it, if you don't like downloading programs, then use winrar
- Search System
- Folder-Up command
- Auto, Save-system, when editing a file and closing the editing window, winrar will automatically ask you to save or not. This way you won't risk closing your program without saving.

It's up to you, but pakscape is recommended since it was exclusively made for pk3's. It's more newbie-friendly in this case.

What you've learned:

You know where the main functions are in winrar/pakscape.

You learned how to use the main functions of winrar/pakscape.

You know why to choose winrar and why to choose pakscape.

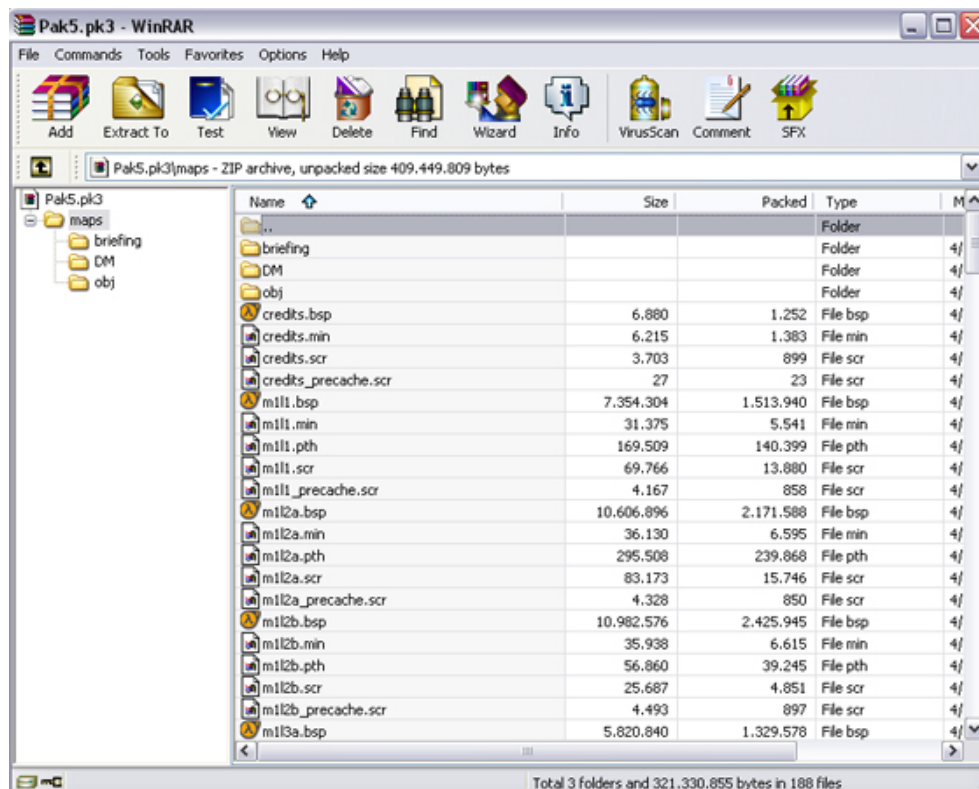
You know the advantages and disadvantages of winrar.

IV. How to open script files.

Okay, now open pak5.pk3 ;) Just double click on it.

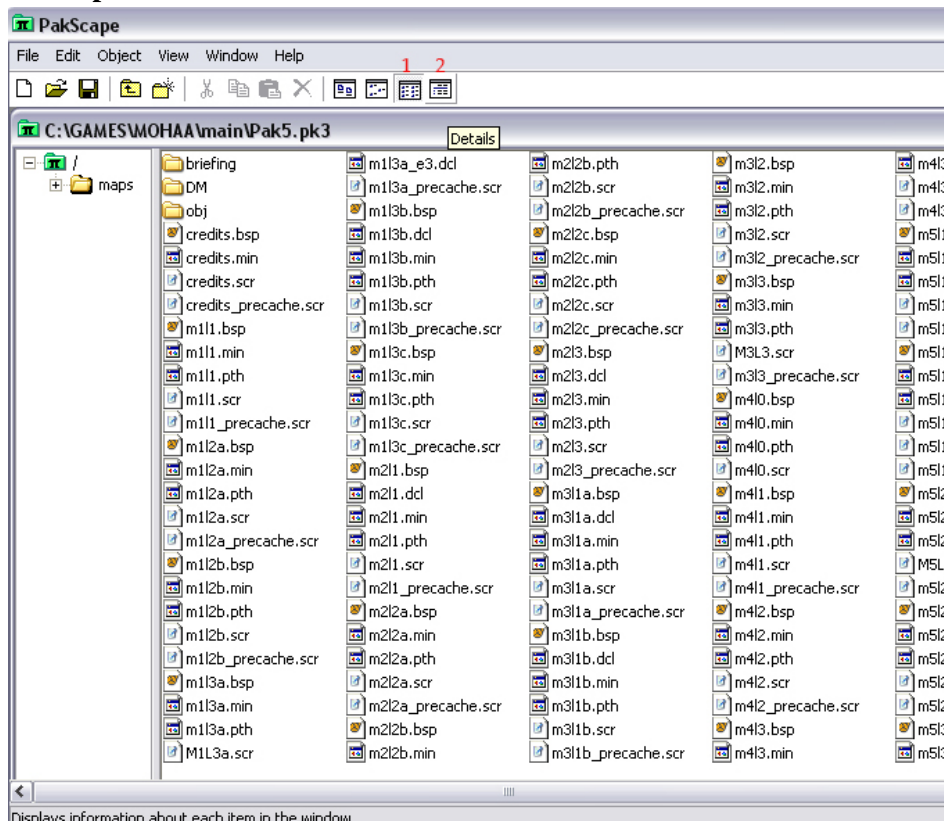
This pak contains all singleplayer maps and all multiplayer maps with their precaches, scripts...

Winrar's view:



Pic12: Winrar's view of the maps folder in the pak5.pk3

Pakscape's view:



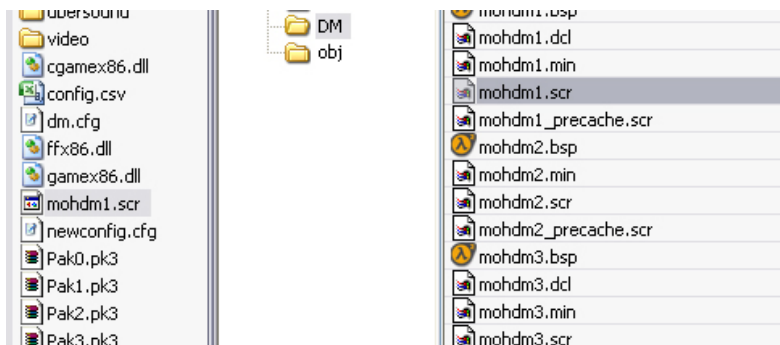
Pic13: Pakscape's view and viewusage of the maps folder in the pak5.pk3

1. Click this to get a list like in the picture above
2. Click that to get a detailed list like in picture 24 (on page 15)

NOTE: Do **NOT** edit files in any of the pak files, **EVER!!** These are all the original game files and any edit could do some serious damage. Always over-ride original files by making pk3's or using the folders of the main.

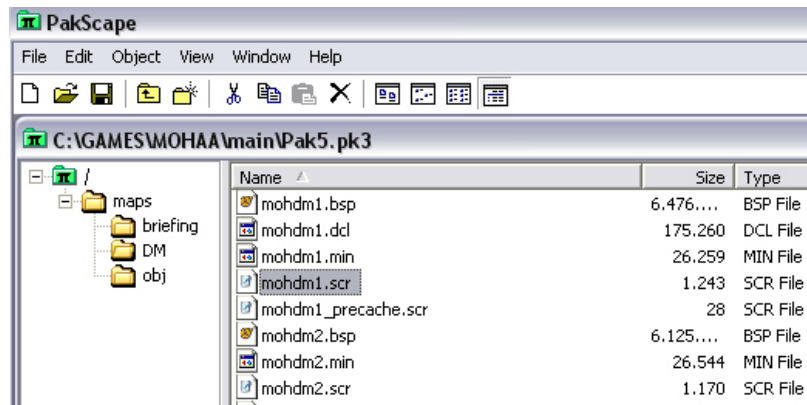
Go to maps/dm inside the pk3 file. Now you'll see all files of all MOHAA's stock dm maps. The .bsp file is the map itself. The .scr file is the script. And the precache.scr file caches map specific models and other files. Select the mohdm1.scr file and drag it to your main folder. Dragging doesn't move the file from the pk3, it copies it, the only way to move a file from a pk3 is to remove it (after copying).

Winrar's view:



Pic14: Winrar's view of the dm folder, selecting mohdm1.scr of pak5.pk3

Pakscape's view:

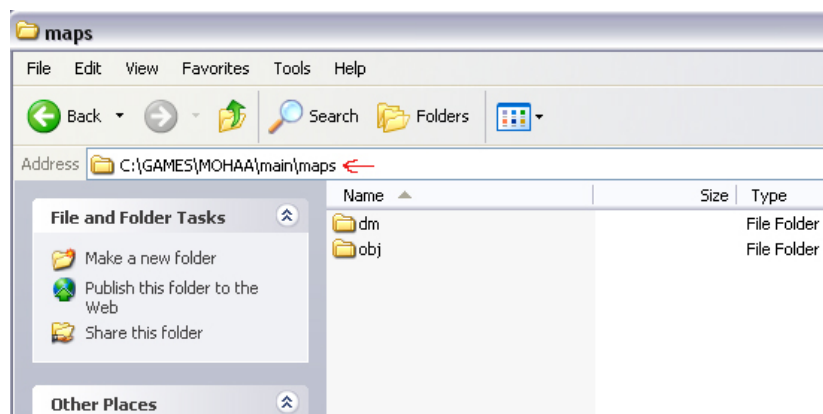


Pic15: Pakscape's view of the dm folder, selecting mohdm1.scr of pak5.pk3

Select the mohdm1.scr file again in your main folder and move(drag) it to main/maps/dm. Here it will over-ride any version of mohdm1.scr



NOTE: This map is not stock created, it's either created when a custom map triggers it or you must create one yourself. I suggest if you don't have it, to create a new folder, name it maps. Then inside your new folder create 2 new folders called "dm" and "obj" Like so:



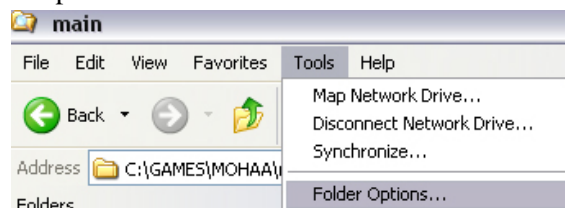
Pic16: How the maps folder with the 2 subfolders should look like

But how to open it?

It's a lot trickier than making a pk3 open with winrar/pakscape. But I'll guide you through it. It's something most scripter don't know. Instead they use other programs to open these .scr files. But I'll make you do it the easy and efficient way. Why can't you do it like with the pk3? Well because the file type .scr means something else than .scr to MoHAA, .scr to windows are screensaver files and for the user's security you cannot open it unless applying it professionally. To mohaa it means SCRipt.

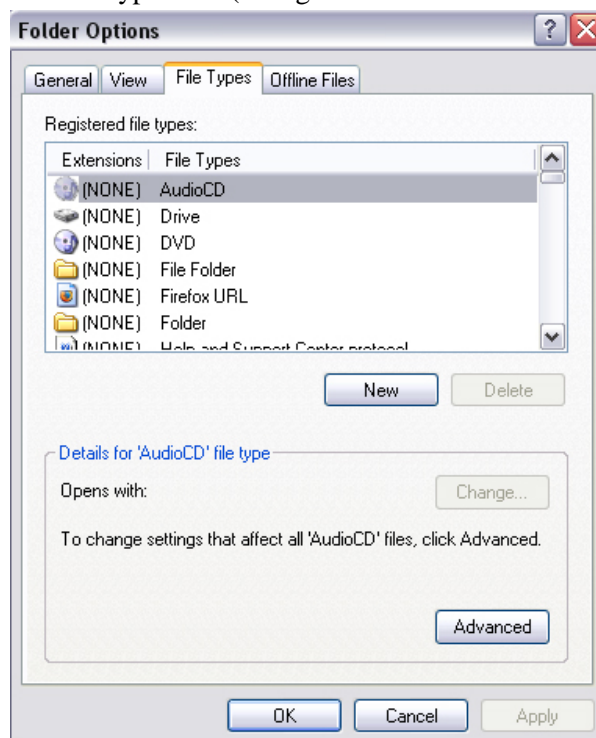
Open a folder, doesn't matter which one, just a folder.

Go to tools and choose folder options:



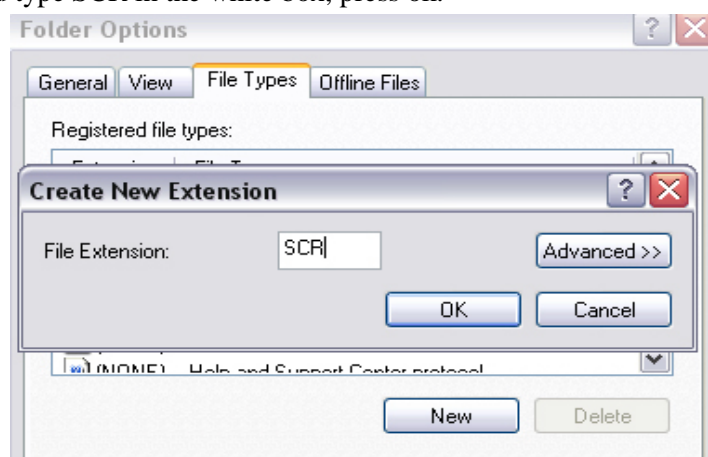
Pic17: Folder options

In the new window go to the File Types tab. (It might take a few seconds for the list to be loaded.)



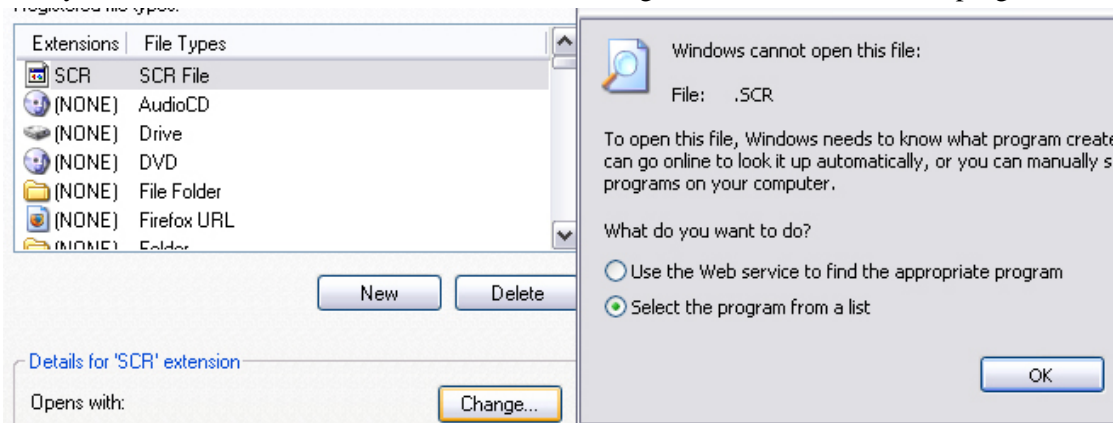
Pic18: File Type tab

Then select New and type SCR in the white box, press ok.



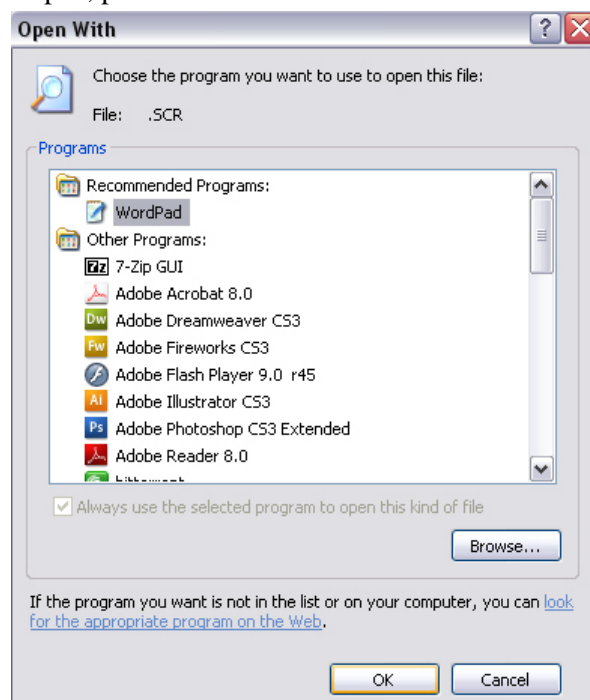
Pic19: Winrar's view of the dm folder, selecting mohdm1.scr of pak5.pk3

Now you'll see SCR in the list, select it and click change below, check to choose program from a list.



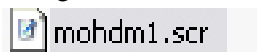
Pic20: Select the SCR file type in the list, click change, select lowest check and press ok.

Then In that list choose wordpad, press ok.



Pic21: Programlist, select WordPad.

Then this will be applied to the SCR file type. Press Close to exit and all scr files should now be able to be opened with wordpad by double clicking them.



Pic22: That's how a .scr must look like when you're done

DONE!!

What you've did:

Now you are able to open scr files and edit them right away

NOTE: Now you'll be able to open and edit scr files in a pk3 itself using winrar or pakscape.

What you've learned:

You know scr files contain MoHAA scripts.

You've learned that editing files in the original gamefiles (the paks) can do serious damage and is strongest UNRECOMMENDED.

V. Closer Look at the files

Let's take a quick closer look at pk3 files. You must also learn scripting from the original script of the game itself. You must also know what pak contains what.

Pak0 = All scripts, all models, all TIK files, all HUD files, all fonts, localization etc (all small important files)

Pak1 = All HUD textures, All menu textures

Pak2 = All other textures

Pak3 = Most sounds

Pak4 = Dialogue Sounds

Pak5 = All maps

There are still a few folders in the MoHAA main folder:

Configs: Config files (keybindings and cvars)

Maps: Empty folder, which will be filled with automatically made .min (sometimes .pth) files of custom maps who's owners didn't put those files in pk3.

Note: I've said this earlier but the maps folder isn't made at install but gets created when a custom map triggers it.

Save: Save files and screenshots

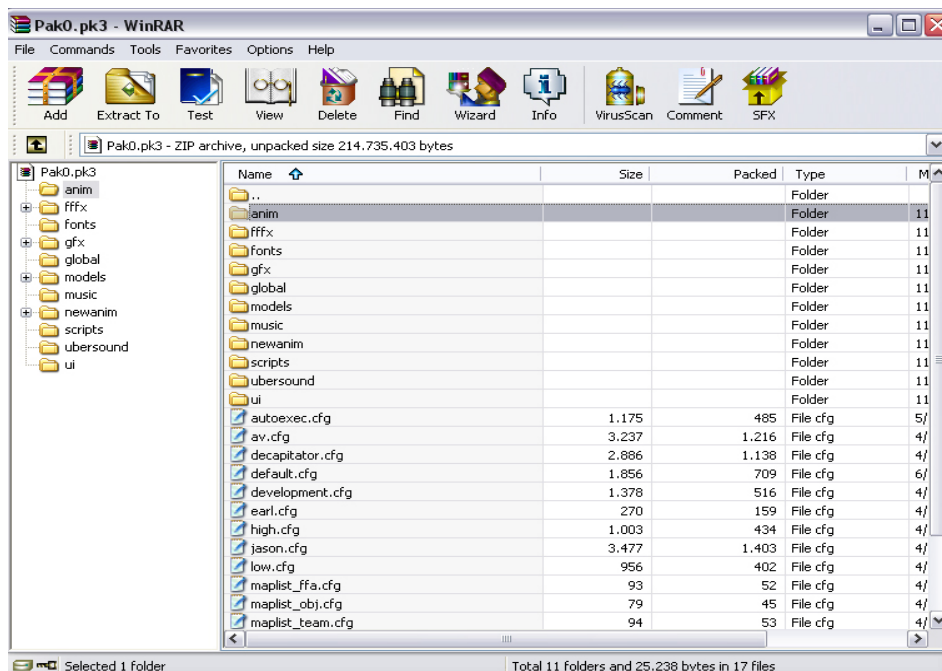
Screenshots: screenshots

Sounds: More amb sounds, vehicle sounds, music etc...

Video: The intro video files

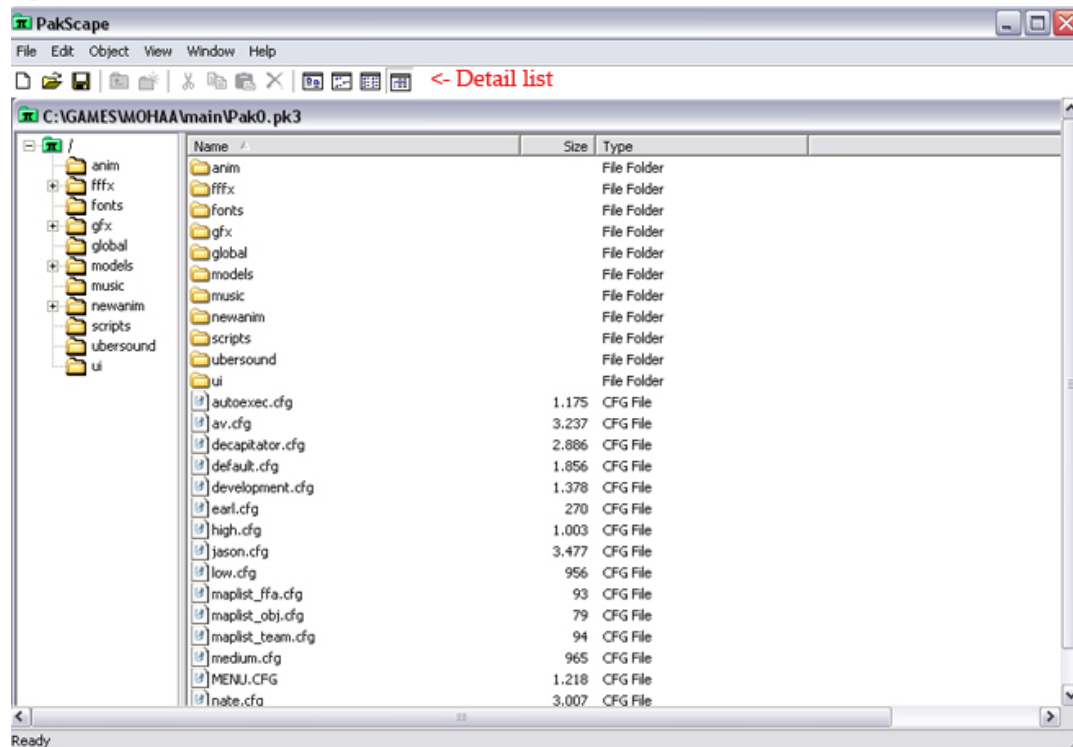
Scripts and pk3 files. Scripts are used for 1 purpose to add, improve and experiment things in maps. A script is always made for a map. There are other scripts for AI etc, these are exceptions, but are indirect also made for maps. You must understand the importance of all files before thinkin of scripting. Open Pak0.

Winrar's View:



Pic23: Winrar's view of pak0.pk3

Pakscape's View



Pic24: Pakscape's view of pak0.pk3 with Detail switched on

Go to the global folder. Inside you will see a vast collection of scripts. Most of these scripts are loaded by default. If you make side-scripts (scripts used in maps but are executed externally, like most of them in the global folder) you must always execute them in the script so the game will know you've made them for that map.

Now let's start with actual scripts!!

What you've did:

You took a closer look at the files in your main folder.

What you've learned:

You learned what pak takes care of what, and what each pak and folder contains.

PART 2

VI. Let's take a look at a script!!

Remember having to drag the mohdm1 script out of the pak and into the main/maps/dm folder?
Time to use it! Open that very script in your main/maps/dm folder.

Let's examine the script of mohdm1, better known as Southern France.

Why? Because you need to know how things are put together before messing with them ;)

However first I'll teach you 1 vital thing you need to know for the rest of your life, playing MoHAA. In the scripting world, 2 / (= //) comments everything behind it on the same line out. In other words, It makes everything behind it on 1 line, invisible to the game. Why? The game reckons everything in a script is script and will consider normal English a script error and will be unable to parse it, which means the script won't get loaded.

Why am I telling you this? Because you need to know how to place comments in your own script, for other people but mainly for yourself. Believe me, this'll come in handy later. Second you can disable parts of your script you think are causing trouble, if you delete them instead of disabling them you might come to the conclusion they weren't causing trouble at all, even worse, they are absolutely necessary.

To comment out bigger pieces of text or script, you can use this:

Place /* above the text/script you want to comment out.

Place */ underneath the text/script

The game will ignore everything between these 2 commands.

Let's start, in black is what you'll see. Comments in red are made by me, explaining what each thing does.

```
// SOUTHERN FRANCE      <- an immediate example of commenting out
// ARCHITECTURE: SENN    I can type without // now because he already did it
// SCRIPTING: POWZER     however // only disables everything behind it on //
                        //the same line, I used // because this is the next
```

```
main:
```

```
/*
^ when your script involves a map or is exec'd in a mapscript, you must
always use main. Main tells the game that everything until the first "end"
is part of 1 main script, anything outside it, which isn't activated in it,
gets ignored but will be read!! So if that contains text or mistakes, your
script will not load unless ignored.
*/
```

```
// set scoreboard messages
setcvar "g_obj_alliedtext1" "Southern France"
setcvar "g_obj_alliedtext2" ""
setcvar "g_obj_alliedtext3" ""
setcvar "g_obj_axistext1" ""
setcvar "g_obj_axistext2" ""
setcvar "g_obj_axistext3" ""
```

```
/* ^ the above sets the text for in the scoreboard when you press tab. Note
that the strange command setcvar is applied. Cvars adds something(between
```



```

the second set of " ") to a predefined command line interface (ie the
console). You can make cvars yourself but that's for later.
*/

setcvar "g_scoreboardpic" "mohdml" //sets scoreboard picture

    // call additional stuff for playing this map round based is needed
    if(level.roundbased)           //English: If the game is RB, then
        thread roundbasedthread //...threads/activate the RB thread
//as stated above it activates a required script for RB if this map is
//played in RB

    level waitTill prespawn

//the things before this line is executed before you enter the map.
//everything between prespawn and spawn is executed the second you enter
//the map (ie as spectator, before you spawn)

    /*** Precache Dm Stuff
    exec global/DMprecache.scr
//this executes all needed caching of models etc...

    level.script = maps/dm/mohdml.scr
    exec global/ambient.scr mohdml

//first line here above, tells the game that this script is indeed the
//script for the map. Second line executes the amb sounds for mohdml.

    //exec global/ambient.scr mohdm2
//the creator probably wanted to use mohdm2's amb sound here but changed
//his mind.

    //$world farplane 5000
    //$world farplane_color (.333 .333 .329)
/*
These are used to set the fog in the map, there's no fog in mohdml, because
these 2 are commented out. Creator changed his mind. The first line uses
$world. This is a predefined targetname for "current map" and the command
farplane's value is how far the player can see. The higher the better, the
lower the worse. In this case it's 5000 which is pretty foggy depending on
the color. That brings us to the second line. It gives color to the
farplane (ie fog). Color uses 3 values. RGB, Red Green Blue.
*/

    level waittill spawn
//now level waittill spawn starts, here you can thread or set anything
//that gets activated as soon as the first player spawns in a match.
End //this ends the main thread.

//After the end are most threads which are activated above. In this scripts
//case, nothing gets activated but the RB thread.

//-----
//Creator uses these separation lines to organize his script, one of the
//most important things you need to do when finished scripting or even
//during is organizing everything, it'll be a lot clearer for you to
//understand.

roundbasedthread:           //a thread is a group of commands/script
//which always has a name and it defines where it starts by stating the
//name (in this case roundbasedthread) and adding a :
```

```
// Can specify different scoreboard messages for round based games here.
//since this was activated before level waittill prespawn and spawn,
//it's obvious you have to state it again.
level waitTill prespawn

level waittill spawn

// set the parameters for this round based match
level.dmrespawning = 0 // 1 or 0
//the above tells the game if the players can respawn in a match or not,
//default is 1 (meaning yes they can) that's why you didn't see this
//earlier. 0 means no (simple, isn't it?)
level.dmroundlimit = 5 // round time limit in minutes
//^English: if a roundstarts, then that round lasts 5 minutes.
level.clockside = kills // set to axis, allies, kills, or draw
/*
when working with rounds, you can set a clockside, meaning how are you
going to determine a round winner. In this case kills, meaning the team
with the most kills that round wins. If set to a team (axis/allies)then
that team wins when time runs out(used only in objective meaning if time
runs out and the other team didn't complete the objectives to win in time
then the defending team wins.
*/
level waittill roundstart
//when does a round start? When there are more than 1 player in the server
//who have spawned. Everything threaded here, only get activated when the
//round has started.
End //ends the "conditional RB thread" meaning that it's activated on a
//condition in the main thread.
```

That wasn't so hard was it? This is all the basic stuff you need to know before scripting. Some of these things you won't even use.

Now, you've got a basic knowledge to start learning how to script. Note that we examined an DM map's script, an objective map has a slightly different script and has some actual scripting in it as well.

You've learned:

How a (map)script is put together.

What every basic command and statement does.

How a (map) script works.

VII. First Results!

Note that MoHAA uses a very exact scripting language, it's very mathematical, logical and exact, any mistake can cause errors and even cause scripts to be unloaded. MoHAA won't solve any straight forward mistakes, it'll simple give you a vague explanation through the console what's causing trouble and if you're lucky a reason why in such English, no one can understand it. But...

Time to start scripting with immediate results! Now ask yourself, you've started learning scripting for a reason? To do what? To spawn something ;)

Okay we'll do that, we'll spawn something!!

Take your mohdm1.scr script again, you've dragged earlier to your main/maps/dm folder.

But what do you want to spawn? A model, a script _origin or...a trigger?

A trigger let's do that!! You got it ;) **Place this underneath the ENTIRE SCRIPT.** When threading from the main: / level waittill prespawn / level waittill spawn (unless in special cases)

```
local.trigger = spawn trigger_multiple "targetname" "trigger1"
```

You're wondering: what in heaven's name is all that???

I'll tell you, let's dissect it:

Local.

This tells you're using a thread to do something, everything behind the . can be defined by you because it must have a name, name it anything you want but make sure it does not have the same name as another local. Command unless you want it this way by referring to it.

So that explains the local bit. But what about the = Why is there a = mark? Simple logic ;)

If local.trigger refers to something a thread executes, then what does it executes?

it executes the following, simply stated by =

The spawn command is something you'll use a lot, it just spawns something. What? Well that's the next bit. It spawns a trigger but what kind of trigger? trigger_multiple so it can be triggered multiple times.

Then you'll tell the game the next bit between the "" is the targetname of this trigger by stating "targetname". The next name between the next set of "" is the targetname of it.

Let's continue...

You've spawned something, but the game doesn't know when to execute it or if it should execute it.

That's why you need to name this entire triggerscript by a thread.

trigger:

```
local.trigger = spawn trigger_multiple "targetname" "trigger1"
```

Please note, every thread must end as well. When you're done in a thread, always end it by simple using stating end. Now you spawned something. But...where is it?? Now you must enter the game to note down the coordinates. Comment both lines out:

```
//trigger:
```

```
//local.trigger = spawn trigger_multiple "targetname" "trigger1"
```

And save your file, then go in game and start the mohdm1 map in LAN.

LAN is this, a local game:



Start mohdm1 in any gametype. Once you're in the map, you must check if the console is enable or not. Go back to main menu, go to options, and advanced(the box on top). Make sure console is checked, like this:

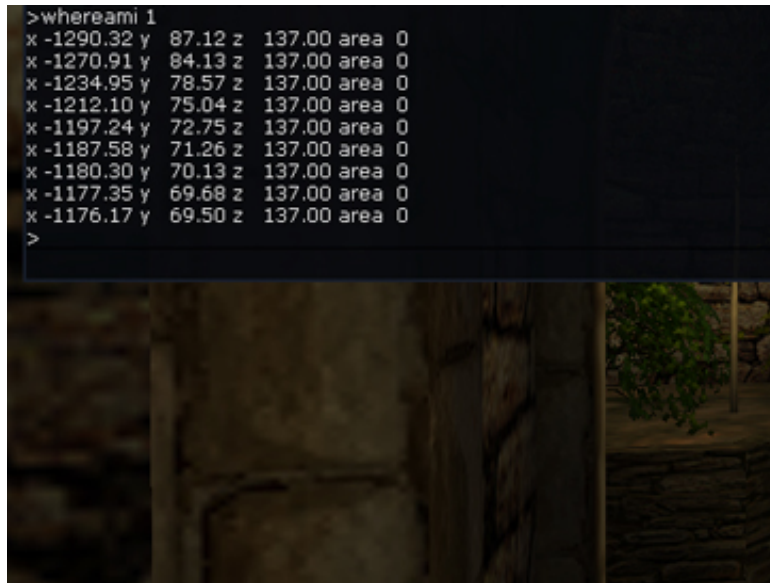


Apply the settings and go back into the game. Press the 2 key to pop up the console, if it doesn't then use the ~ key. The console should pop up, now you need the coordinates. Coordinates exist out of 3 values. Coordinates are game units, 16 units is 1 foot.

Pop up the console, and type

whereami 1

and press enter. Turn of the console and move a bit in spectator, open your console again. And tada, the game records every coordinate of every movement!



Like I said before, coordinates exists out of 3 values (unless you're dealing with 2D coordinates)

First coordinate value is X, then Y and last Z. X is length, Y is width and Z is depth/height.

Move to the place you want the trigger to be, please make sure it's on the ground. Note down the coordinates, the decimals aren't very important, neither is the area.

For now, I'll my example coordinates.

The example trigger is located here:



Let's get back to the script, first of all, uncomment your 2 lines.

Then add this:

trigger:

```
local.trigger = spawn trigger_multiple "targetname" "trigger1"  
$trigger1.origin = ( 469 430 83 )
```

Where did this: **\$trigger1.origin** come from, you might be wondering? Well that's your targetname put to use, you can continue using local.trigger, but targetnames are made to make you understand what a script is for just by reading the name.

When using a targetname always add \$ in front of it. For safety always make sure you're value is 1 space away from your (or = or whatever. Because if a negative value touches a command (like = or + or (etc..) like (-469) then the game will error and unload the script, this error has been made in.

A last question you might have is why origin is pasted to the targetname or if you use local.trigger. It's simple by pasting it to it using . it tells the game that what you're doing now is about the local.trigger itself and not what it does etc... You'll see other cases where this is used, just do as I do. It's hard to explain all cases.

Next, let's determine how big your trigger will be:

trigger:

```
local.trigger = spawn trigger_multiple "targetname" "trigger1"  
$trigger1.origin = ( 469 430 83 )  
$trigger1.setsize ( -40 -40 -40 ) ( 40 40 40 )
```

Why does it have 2 coordinate boxes? Well it determines the maximum size in length (X), width (Y) and depth (Z) and the minimum, don't use too big setsize unless intended. The first box is for the minimum size and the second for the maximum. So the size of this trigger is a 20 unit circle. Because the distance between the max and min are 20 units.

Not to determine what the trigger does when stepped on or through.

Therefor we must make a new thread and make the trigger target that thread when it's triggered.

You can check in game to see if your setsize is big enough or the way you want it. Pop up the console and use sv_showbboxes 4. Very Handy! You see a box, within its boundaries the trigger can be activated. Example:



Continuing...

```
trigger:
    local.trigger = spawn trigger_multiple "targetname" "trigger1"
    $trigger1.origin = ( 469 430 83 )
    $trigger1 setsize ( -40 -40 -40 ) ( 40 40 40 )
    $trigger1 setthread action
```

The triggers sets a thread (ie action) when it is triggered. Now the trigger is spawned and ready for use. We must end this thread since we don't need to do anything here.

```
trigger:
    local.trigger = spawn trigger_multiple "targetname" "trigger1"
    $trigger1.origin = ( 469 430 83 )
    $trigger1 setsize ( -40 -40 -40 ) ( 40 40 40 )
    $trigger1 setthread action
```

End

Now you've got a new thread!

```
action:
```

But what do you want to happen if someone triggers it? Kill him? Okay lol.

```
action:
local.player = parm.other
local.player kill
end
```

This is a sloppy thread, why? I'll tell you in a minute but first. You're wondering again, what does `local.player = parm.other` do. Well it defines the player, hard to explain. Only thing you must remember is when a setthread or someone threads to a thread where a player undergoes a certain action, you must define him, using this way. If you need to define the player with a simple thread after level waittill spawn. Then you need to use a much more difficult definition, used in special cases. But that's a bit to advanced at the moment. Second command is fairly simple, it kills the player.

I said it this thread is sloppy. Well I'll clean it up now.

```
action:
    self nottriggerable
    iprintln "triggered"
    local.player = parm.other          //defines player
    local.player kill                  //kills player
    self triggerable
end
```

Now it's cleaned up, you can add comments to help yourself and others who read your scripts.

I added `self nottriggerable`

Who is self and what is nottriggerable? Self is a predefined object. It gets defined at the creation of threads. In Automatically started threads (the threads which are automatically activated in each map are to load all things needed to play normally. They aren't found in the mapscript.)

Self = NULL (meaning nothing) and in animation scripts, it's the character itself.

Self is threads is equal to self in other threads of the group. (eg: a thread defines self, and threads another thread that threads on his turn another thread. Self will remain the same thing like in the first)

Self in object threads, such as `$trigger1 setthread action or`
`local.player thread something`

So as stated in the last case self refers to the trigger. So `self nottriggerable` means when the trigger is triggered it cannot be triggered anymore until `self triggerable`. In other words, when the trigger is in use, no one can trigger it until it's done.

```
iprintln "triggered"
```

The print is for testing purposes, if the print doesn't show, you've got a problem in your spawn thread, if it does show but the action isn't doing what it's supposed to, then there's the error. If all works well comment it out ;)

Okay, you're done. Once you get the hang of this, learning other stuff about scripting in part 3 and 4 will be a lot easier, get through this and you got it ;)

First, you need to make the game know what to activate when. Let's take spawn, I doubt spectators can trigger it.

Simple place `thread trigger` between `level waittill` spawn and the first end

No need to thread action because the trigger takes care of it ☺

Save your script!

Test time ;)

Start mohdm1 and go to the place of which I shown you a picture earlier or if you used your own coordinate, to that spot then, if you die and see triggered underneath the compass. Pat yourself on the back! It works, hehe and my body even slid a bit further:



What you've learned:

You learned how to make a trigger, using the proper scripting.

You learned how to place the trigger on a map.

You've learned how a trigger works.

You've learned how to make a trigger do something when triggered.

You know what the self object is.

You know how and when to define the player.

You've learned how to test your script (iprint)

You've learned how to make a trigger unusable when it's being used.

You've set a big step forward in scripting.

a. Tips

1.

When encountering an error **the console is your best friend**, but he does it vaguely but still clear enough for you to try and pinpoint the error. The console only displays a limited section of errors, gamemessages, loadmessages etc. If your error isn't in there, exit your map and pop up the console and type:

logfile 1

press enter. Now enter your map, wait till the error occurs. Exit the game. Go to your main and open the qconsole.log file (it should open automatically with notepad) It contains a full log of the console of all events of your last match. Remember to empty the qconsole.log when done.

2.

As I said before, when seeking coordinates type (when in a map):

whereami 1

In the console and press enter. When you've found the right spot, pop up the console and write down the last stated coordinate, that's the one you need!

3.

To view your trigger's effect, type(when in a map):

sv_showbbboxes 4

into the console and press enter. Go find your trigger and you'll see the box where your trigger has effect, in other words the setsize.

4.

Make comments. For yourself, for others. When looking back on an unfinished script you don't have to wonder, what's this for, why did I make this etc...

5.

Organize, clean and neaten your script up. Check B here below for more on this. I'm not telling this to waste your time, you'll be much more efficient in working in your script when it got a good overview.

NOTE: Thanks to windows the “ “ are no good if you copy them from this document to your script, delete them and add them again in your script, save it. If you don't the game will error and stop loading of your script. Word uses 2 types of “ and wordpad and notepad 1, that's the one the game recognizes

b. Order!

I'd like to mention it is important to write a script in a proper way. Like in my example

```
local.trigger = Is NOT local.trigger= !!
```

When I present examples, please note these things as well, because they are very important, use my syntaxes and my way of writing scripts/commands.. or you'll make mistakes!!

For every bit you do, use another line, like you'll see in the examples.

Organize your scripts using tab, example:

```
Trigger:
    //something
End
```

You'll find out this will be very useful once we're starting if, while, switch and for statements.

Separate your parts of the script using these things:

```
script
//-----
Script
```

Even separate it and give some info

```
script
//-----trigger ends-----
//-----
//-----trigger action starts-----
Script
```

These things are strongly recommended!!!

It is also important to separate projects from other files and organize it thoroughly. Make a new folder in your main called project or temp or something. And divide it in a few bits, When working scripts or maps, divide them. Make maps called "Mapping projects", "Scripting projects" Within these make another divide, "Your Project" and "Tools and Resources". The first one contains your project, the second contains, scripts you can use to learn how to make your script (better) and tools(like this tutorial). Also important, after testing it in the maps/dm (or obj) folder to take it out of there again!

More about this in the next parts of the tutorial.

ENJOY!!

**Your first made your first script
and
your first step into the scripting world ;)
Go to part 3 and 4 (available soon)**

Need any help or questions?

**Register at my forum, and use this topic to post any problems and
questions:**

<http://z4.invisionfree.com/MoHAAirborne/index.php?showtopic=18>

For urgent Contact:

MOHAALennySoldier@hotmail.com