**Mefy’s Gametype Addon Tutorial**

Here's a basic tutorial for creating an addon gametype to the Extended-Gametypes mod. This will work with Version 1.2.2 or later.   
  
Why would you want to do this?   
By making your gametype an addon, it will automatically work along with FT, CTF, DEM, and any other gametype addons without conflicting. Your gametype will be activated through the standard g\_extgametype cvar that many server hosts are already familiar with. And servers that run mixed-gametype rotations will be able to incorporate your new gametype into their rotation easily.   
  
You will need to choose an abbreviation for your gametype. This is what server hosts will set g\_extgametype to in order to activate your gametype. It can't be any of these since they're already used:   
  
FFA - Free-For-All   
TDM - Team-Match   
RBM - Round-Based-Match   
OBJ - Objective-Match   
TOW - Tug-of-War   
LIB - Liberation   
CTF - Capture-The-Flag   
FT - Freeze-Tag   
FTOBJ - Freeze-Tag-Objective   
DEM - Demolition   
FTCTF - Freeze-Tag-CTF   
FTDEM - Freeze-Tag-Demolition   
FTTOW - Freeze-Tag-TOW   
  
For Assassination mode I chose AS as the abbreviation.   
  
If you look in the Assassination addon pk3 you will see the following files:   
  
global/libmef/gametypes/as.scr   
global/asscripts/dm/mohdm1.scr   
global/asscripts/dm/mohdm2.scr   
global/asscripts/dm/mohdm3.scr   
global/asscripts/dm/mohdm4.scr   
global/asscripts/dm/mohdm6.scr   
global/asscripts/dm/mohdm7.scr   
  
The first file (as.scr) is the gametype script. The rest of the files are the map setup scripts. If your gametype needs to be set up specifically for each map then you will need these map setup scripts. If not then you will only need the gametype script.

The gametype script must be named global/libmef/gametypes/<abbr>.scr where <abbr> is your gametype abbreviation. To make it function as an addon you will need to create 3 threads inside of it: a gametype registration thread, gametype initialization thread, and gametype execution thread.   
  
You should be looking at the as.scr script in the Assassination addon pk3 to follow along with the explanations of these threads.   
  
**1) Gametype Registration Thread**   
The first thread is the gametype registration thread. Its purpose is to announce the presence of your gametype. It must be the very first thread in your gametype script.

|  |
| --- |
| register\_gametypes:  level.mef\_gametypes["as"] = global/libmef/gametypes/as.scr::init\_gametype  end |

This thread does one thing: set a key in the level.mef\_gametypes array. The key name should be your gametype abbreviation and the value should be the name of the initialization thread for your gametype.   
  
**2) Gametype Initialization Thread**   
The second thread is the gametype initialization thread. Its purpose is to define the name of the gametype, which base gametype mode it will be run in, define any gametype settings, and start the gametype execution thread.

|  |
| --- |
| init\_gametype:  self.gametypestring = "Assassination"  self.basegametype = 4    waitthread global/libmef/gametypes.scr::new\_setting "pistolonly" bool 1  waitthread global/libmef/gametypes.scr::add\_roundbased\_settings  thread as\_begin  end |

In this thread you need to set self.gametypestring to your gametype's name (Assassination in this case) and self.basegametype to the gametype number (what g\_gametype would be set to). For Assassination it will be 4 which means the gametype will run in Objective mode.   
  
You can also define any gametype settings here. In this case I've defined the "pistolonly" setting which is a boolean that is defaulted to true. I also call the "add\_roundbased\_settings" thread which will add the "roundlimit", "timelimit", and "fraglimit" cvar settings. Look at ft.scr, ctf.scr, and dem.scr for more examples of how to use gametype settings.   
  
Finally you need to start your gametype execution thread (using thread, not waitthread). In this case I call as\_begin.

**3) Gametype Execution Thread**   
The third thread is the gametype execution thread. It contains all of the code for your gametype. The following code is just the beginning of the execution thread for the Assassination gametype.

|  |
| --- |
| as\_begin:  if (level.as\_running)  end  else  level.as\_running = 1    level.as\_version = "0.2 (27.02.2005)"    println "[ASSASSINATION]: Initializing assassination script v" level.as\_version "..."    level.mef\_dontdefinespawns = 1  waitexec ("global/asscripts/" + level.mef\_fullmapname + ".scr")    if !(level.as\_enabled)  {  println "[ASSASSINATION]: ERROR: map " (getcvar(mapname)) " is not assassination-enabled! Terminating!"  end  } |

By default my mod will manipulate the spawn points on mohdm1, mohdm4, mohdm6, mohdm7, and mp\_bazaar\_dm. By setting level.mef\_dontdefinespawns to 1, you can prevent it from doing that.   
  
This line executes a map-specific setup script for the gametype:

|  |
| --- |
| waitexec ("global/asscripts/" + level.mef\_fullmapname + ".scr") |

If your gametype needs special setup for each map then you can do something like this. All of the map setup scripts would be under one folder (global/asscripts in this case). The level.mef\_fullmapname variable contains the name of the map (dm/mohdm1, obj/obj\_team1, lib/mp\_anzio\_lib, etc...).   
  
For Assassination mode a map setup script will redefine the spawn points for a map, set the scoreboard messages, and define which team is the VIP team. It also sets the level.as\_enabled variable to 1 to declare that the gametype has actually been set up for that map. Then we can check in as.scr whether or not the map is set up:

|  |
| --- |
| if !(level.as\_enabled)  {  println "[ASSASSINATION]: ERROR: map " (getcvar(mapname)) " is not assassination-enabled! Terminating!"  end  } |

If it's not set up then we stop immediately.   
This covers the basics you need to get your addon gametype running. If you have any questions just post them here