
Subject: scripts.dll 2.1

Posted by [jonwil](#) on Mon, 17 Jan 2005 11:19:23 GMT

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

ok, some more things that will be in 2.1:

The Get_Radar_Mode engine call is gone since you can grab the radar mode from the cGameData stuff I have added.

Some of the scripts mentioned in this thread will require engine calls (not all of which will be listed since I wont know the details until I write and test the script)

Obviously the IP address fix and mine limit change fix are in 2.0.1 so they will carry over to 2.1

There are 2 new scripts similar to JFW_Attach_Script_Player_Created called JFW_Attach_Script_Building_Created and JFW_Attach_Script_Vehicle_Created that do just what they say.

One sticks the script on every building controller in the game when the building controller is created.

And the other sticks the script on every vehicle in the game when the vehicle is created.

They (like the other script attach stuff) will have settings to set which team you want them to work for (or both)

JFW_Destroy_Self_Timer which basically will, when created, start a timer.

When the timer expires, the object with JFW_Destroy_Self_Timer on it will be destroyed.

JFW_Change_Model_Health which will, when the health of what its attached to goes below a certain amount, change the model of whatever its attached to one model.

Then when it goes back over the health amount, it changes to another model and so on.

JFW_Dplbl_Vhcls_Keyboard which is exactly like the latest deployable vehicle script from KGB except that it will feature a keyboard hook to destroy instead of the current way of doing it.

JFW_Spawn_Object_Poke_Custom which will spawn an object when poked.

Then, it will disable itself until a custom is sent to it allowing another object to be spawned.

JFW_Spawn_Object_Poke_Timer is like JFW_Spawn_Object_Poke_Custom except that instead of a custom re-enabling it, it has a timer that starts when the object is spawned that re-enables it when the timer expires.

JFW_Spawn_Object_Poke is like the other 2 but it doesn't get disabled (so you can use it as many times as you like)

All 3 will have parameters to specify where the object is to be spawned.

JFW_Jetpack is a script a bit like JFW_Flying_Infantry except that there will be a keyboard hook to enable and disable the "fly" effect.

JFW_Carryable_Vehicle is a script that goes onto every vehicle you want to be carryable by a

vehicle carrying vehicle (e.g. caryall, transport etc).

Basically, it takes 2 pieces of information. The custom to listen for from the caryall vehicle. And the size of an imaginary rectangular prism around the vehicle which contains the whole vehicle within it. This is used so that when the caryall grabs the vehicle, it doesn't get too close.

JFW_Caryall is a script for something like a vehicle caryall or a transport.

The caryall has an imaginary rectangular prism around the vehicle (for example, for an Orca Caryall, it will be below the vehicle). Then, the caryall activates via a keyboard key. When activated, if no vehicle is connected to the caryall, it will search the area bounded by the invisible rectangular prism for a vehicle of the appropriate team. If none is found, nothing happens.

If one is found, settings on the caryall are combined with settings from the vehicle (which will be retrieved by sending the custom to the vehicle) to determine where relative to the caryall vehicle that the vehicle being picked up should go. It is then moved there and attached to the caryall. It will also be locked so that people can't get out of the vehicle (firing will still work though since I can't turn that off and also the vehicle will be made immobile so it can't move if I can figure out how). When you arrive at the destination, the keyboard key will be pressed again and the vehicle will be detached and (optionally) moved a certain distance in the x,y,z direction.

This script is going to be harder than I thought it would be (especially if I want to make it generic enough to handle different places of vehicle attachment and stuff), I need to handle the rotation of the vehicle in x,y,z too.