
Subject: scripts.dll 2.7 WIP update

Posted by [jonwil](#) on Sat, 15 Apr 2006 14:01:32 GMT

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

Here is a list of all the features currently added to 2.7:

Various small improvements to engine calls in engine.cpp (e.g. some engine calls now use the new Disarm_Beacon engine call to disarm beacons)

An improvement to Console_Output to enable it to be used like printf (in that you can use %s, %f etc and pass variables to it)

new engine calls:

void Send_Message_With_Obj_Color(GameObject *obj,const char *Msg); //send a messages in a given objects color

void Send_Message_With_Team_Color(int Team,const char *Msg); //Send a message in a given teams color

void Damage_Occupants(GameObject *obj,float Damage,const char *Warhead); //Damage all the occupants of a vehicle

int Get_Current_Bullets(GameObject *obj); //Get loaded bullets for an objects current gun

int Get_Current_Clip_Bullets(GameObject *obj); //Get clip/backpack bullets for an objects current gun

int Get_Current_Total_Bullets(GameObject *obj); //Get total bullets for an objects current gun

int Get_Total_Bullets(GameObject *obj,const char *weapon); //Get total bullets for a specific gun (if the object doesnt have the gun, return is zero)

int Get_Clip_Bullets(GameObject *obj,const char *weapon); //Get clip/backpack bullets bullets for a specific gun (if the object doesnt have the gun, return is zero)

int Get_Bullets(GameObject *obj,const char *weapon); //Get loaded bullets bullets for a specific gun (if the object doesnt have the gun, return is zero)

int Get_Current_Max_Bullets(GameObject *obj); //Get max loaded bullets for an objects current gun

int Get_Current_Clip_Max_Bullets(GameObject *obj); //Get max clip/backpack bullets for an objects current gun

int Get_Current_Total_Max_Bullets(GameObject *obj); //Get total bullets for an objects current gun

int Get_Max_Total_Bullets(GameObject *obj,const char *weapon); //Get max total bullets for a specific gun (if the object doesnt have the gun, return is zero)

int Get_Max_Clip_Bullets(GameObject *obj,const char *weapon); //Get max clip/backpack bullets bullets for a specific gun (if the object doesnt have the gun, return is zero)

int Get_Max_Bullets(GameObject *obj,const char *weapon); //Get max loaded bullets bullets for a specific gun (if the object doesnt have the gun, return is zero)

int Get_Position_Total_Bullets(GameObject *obj, int position); //Get total bullets for an objects gun at a specific position

int Get_Position_Bullets(GameObject *obj,int position); //Get loaded bullets for an objects gun at a specific position

int Get_Position_Clip_Bullets(GameObject *obj,int position); //Get clip/backpack bullets for an objects gun at a specific position

int Get_Position_Total_Max_Bullets(GameObject *obj, int position); //Get total bullets for an objects gun at a specific position

int Get_Position_Max_Bullets(GameObject *obj,int position); //Get loaded bullets for an objects gun at a specific position

```
int Get_Position_Clip_Max_Bullets(GameObject *obj,int position); //Get clip/backpack bullets for
an objects gun at a specific position
void Set_Current_Bullets(GameObject *obj,int bullets); //Set current loaded bullets for an object
void Set_Current_Clip_Bullets(GameObject *obj,int bullets); //Set current clip/backpack bullets for
an object
void Set_Position_Bullets(GameObject *obj,int position,int bullets); //Set position loaded bullets for
an object
void Set_Position_Clip_Bullets(GameObject *obj,int position,int bullets); //Set position
clip/backpack bullets for an object
void Set_Bullets(GameObject *obj,const char *weapon,int bullets); //Set loaded bullets for an
object
void Set_Clip_Bullets(GameObject *obj,const char *weapon,int bullets); //Set clip/backpack bullets
for an object
const char *Get_Team_Name(int Team); //Get a teams name
const char *Get_Vehicle_Name(GameObject *obj); //Get the name of a vehicle (as used for the
HUD display)
INIClass *Get_INI(char const *filename); //Open an INI file and read stuff from it (you can access
various things through the iniclass that is returned to read bool, float, int, string)
void Release_INI(INIClass *ini); //Close an INI file
void Disarm_Beacon(GameObject *obj); //Disarm a beacon
void Disarm_Beacons(int ID); //Disarm all beacons for a player
GameObject *Find_Harvester(int team); //Find this teams current harvester, if any
int GetMaxPlayerID(); //Get the maximum currently used player ID
new script JFW_Damage_Occupants_Death which damages all occupants of a vehicle when the
vehicle is killed
Versions of JFW_Character_Buy_Poke, JFW_Refill_Buy_Poke, JFW_Vehicle_Buy_Poke,
JFW_Powerup_Buy_Poke, JFW_Preset_Buy_Poke, JFW_Group_Purchase_Poke,
JFW_Powerup_Buy_Poke_Timer, JFW_Weapon_Buy_Poke, JFW_Preset_Buy_Poke_Timer &
JFW_Preset_Buy_Poke_Custom that play a sound if the poker has not got enough cash for the
purchase.
Versions of JFW_Switch_Door, JFW_Switch_Lock_Door, JFW_Toggle_Door,
JFW_Toggle_Lock_Door, JFW_Switch_Door_Team & JFW_Toggle_Door_Team that use the
object they are attached to as the object to act on (instead of needing to pass the object ID in)
new script, JFW_Vehicle_Extra which basicly creates an object at a particular bone of whatever
JFW_Vehicle_Extra is on and attaches the object to the bone. Then, when the object with
JFW_Vehicle_Extra on it is killed, the other object is destroyed too.
new console command to disarm all C4 of a player
new console command to disarm all proximity C4 of a player
new console command to disarm all beacons of a player
Bug fixes to a few places to use a more accurate player count (instead of using the number of
players on the server, it uses another value that returns the highest player ID currently in use).
Among other things, this should fix the reported PINFO bug.
A new hook that gets called anytime the "The version of bhs.dll for player x is y" message gets
printed. The hook gets passed the player ID and the version number.
A new engine call to set wireframe or not wireframe. Like the mine limit and vehicle limit, this stays
set until you set it to something else, even across maps.
Code to send the Disable_All_Collisions, Disable_Physical_Collisions and Enable_Collisions
script commands over the network.
```

A new engine call to load an ini file at runtime that will replace certain stuff read from hud.ini. Like the engine call to replace the HUD_MAIN texture, this stays in force until changed again. Note that this will still work even if there is stuff in the ini file that the command doesn't read, the other stuff will be ignored. Obviously, the client needs the ini file and all the textures (just like with hud.ini)

New tags for hud.ini, as follows: (details will be forthcoming once I actually figure out how to document all this stuff in a way that someone other than myself can understand)

Firstly, you can define entries for colors (with a red, green and blue) which can be referenced elsewhere in the file.

Secondly, you can completely customize the number that shows your current health and the one that shows your current shield

Also, you can completely customize the numbers that show your current bullets (both of them)

Also, you can customize both the name and image for the current weapon (which includes things like the steering wheel/gun/seat icons for inside a vehicle)

You can set a setting to make the names (and only the names) of enemy soldiers invisible. Also, there is an exception list so that certain presets will remain visible (e.g. for spies)

Also, you can have an "unlimited" (limited only by system resources/the engine) number of "textures". Each has a texture filename and a number of rectangles associated with it. Each rectangle has a color, a location and a UV setting associated with it (the UV setting defines a rectangle on the texture, if the texture is 128x128 and the UV is 0,0,64,64 then that selects the top left corner. The UV is also used to specify the size of the rectangle.

You can customize the health bar and shield bar including textures, UV offsets, colors and many other things (you can do both renegade health bar style bars and renegade shield strength style bars)

The health/shield bars and health/shield numbers take settings to say "if player has x%, use this color"

Also, all the new hud elements can be disabled completely if you don't want them (e.g. shield bar if your mod has no shield strength settings anymore)

Plus, you can disable the radar, compass, weapon box (i.e. the circle thing that goes behind the weapon image, weapon name and bullet counts) and info box (i.e. the thing that covers every part of the main hud except the radar and the text items)

Note that some of this stuff (all the HUD stuff for example, also the wireframe mode and other things) will need bhs.dll on the client.

Here is what I still have on the 2.7 to-do list, not all of it will make it into 2.7 though:

Definitely in 2.7:

Fix the bug in Set_Death_Points/Set_Damage_Points so that they actually work

Further changes to the WOL stuff that blazer needs for LFDS WOL (expect the WOLSEND console command to go away in 2.7)

Fix the bug where the obelisk glow doesn't seem to work right in some cases

A script you can put on a map to either set wireframe mode or not set wireframe mode

A script to play one sound if the player has key x and another sound if they don't.

Engine calls to get the vehicle owner (i.e. the purchaser) and the "lock time" (i.e. how long before it will become unowned)

Note that entering a vehicle resets the owner as does the expiration of the lock time (defaults to 26 seconds when you buy the vehicle)

An engine call to give a specific weapon (not a powerup, the weapon preset)
An engine call to get the weapon preset that a particular PowerupGameObj will grant.
Check if the code to change bullets, add weapon or remove weapon has to go over the network in order to work 100% and if so, make it happen.
Disable options for other HUD elements (e.g. team/player info, credits/time remaining info)
Custom code to add a little icon for the health and one for the shield (like the little plus next to the health number in normal renegade). This will support the same color changeability as the other health/shield options plus it will optionally allow you to set a flashing icon (like renegade does when you have low health)

Probably going to be in 2.7:

Full customizability for the compass
Customizability for the radar (how much is unknown at this time but I want to implement the ability to use a texture of the current map like the ones in SP as a radar background and then it will scroll depending on where you are on the map. It would always stay pointing "north" though (i.e. top of texture = top of screen)
Changing the time remaining and credits
Investigate/fix the reported blurry screenshot bug when you take a PNG screenshot in windowed mode.
Support for the side buttons on my Microsoft USB Optical Intellimouse.
Code to read various PT related items (specifically, being able to read data from the presets listed under "Purchase Settings" and "Team Purchase Settings" in the preset tree in leveledit)

Might make 2.7, unclear at this point:

Various scripts that use the weapon related engine calls.
Ability to change the team/player info at the top of the screen. (i.e. move it and stuff)
Look into the reported turret lag issue in RenAlert (which is where the location that the graphics engine draws the turret and any bullets etc at and the actual location are different)
Find out if Action_Complete is or isn't being called properly and if its not, try and make it work.
Engine calls to set the owner and lock time of a vehicle (this will need bhs.dll on the client)
Code to change the PT data at runtime (same stuff as the read code). This will need bhs.dll on the client to work.

Naturally, all of the stuff in this post is subject to change (e.g. if testing reveals something isn't going to work, I will remove it or change it)