

---

Subject: complete changelog for scripts.dll 2.7

Posted by [jonwil](#) on Tue, 25 Apr 2006 13:27:28 GMT

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

---

This is a complete changelog for new features in scripts.dll 2.7.

I am still hard at work finishing the last features (like the radar) but rest assured that I will push 2.7 out the door as soon I can (and I will make sure the big stuff has been tested)

Complete changelog: (items marked with a \* will require bhs.dll 2.7 on the client to function)

Bug fixes to the nick exploit fixes, the ID console command and the PINFO console command so that it will correctly check the entire nickname list.

The WOLSEND console command is now gone.

New console command WOLNAME that is used by LFDS WOL to set the WOL server name (remember that even with LFDS XWIS, the server still thinks its a GameSpy server and behaves as such)

New console command DISARM to disarm all C4 of a player

New console command DISARMP to disarm all proximity C4 of a player

New console command DISARMB to disarm all beacons of a player

New code to send the special packet that WOLSEND used to send (but to send it properly so people on all NAT connections can join)

A new hook that will trigger anytime the "The version of player x is y" message is printed. Hook is passed the player ID and version.

\*void DLLEXPORT Set\_Wireframe\_Mode(unsigned int mode); //Sets the wireframe mode. Will remain in effect untill changed by another call to Set\_Wireframe\_Mode

\*Code to send Disable\_All\_Collisions, Disable\_Physical\_Collisions and Enable\_Collisions over the network

\*void DLLEXPORT Load\_New\_HUD\_INI(GameObject \*obj,const char \*ini); //Loads a new hud ini at runtime (the only things that can be changed at runtime are texture filenames and most but not all color related tags). Note that once you load an ini file, things will stay the same until changed again.

\*void DLLEXPORT Update\_PT\_Data(); //Indicates that PT data updating has been completed and triggers a send of the PT data over the network. Bear in mind that the PT data, once changed, stays changed until its changed again to something else.

\*A fix to the Remove\_Weapon engine call such that it sends properly over the network now (and functions correctly)

\*New code for custom HUD. The following parts exist in the new HUD code:

You can define colors which will then be used later on in the code.

You can customize the health and shield numbers

You can customize the bullet count and clip count numbers

You can customize the weapon name and image

You can hide the names of enemy soldiers, with an exceptions list (e.g. spy presets go there so that spies are not hidden)

You can define a texture entry which then has one or more rectangles taken from that texture.

This is how the generic elements of the custom HUD are drawn.

You can change the health and shield bars.

You can hide the weaon box (the bakground behind the weapon numbers and image)

You can hide the info box (which covers everything thats not text on the other side to the weapon box except the radar including the health and shield bars)

You can customize the compass text (i.e. the text that says "N,S,E,W etc)

You can customize the radar. As of right now, objectives and "radar markers" wont be displayed. Only normal objects will be displayed. Also, functionality to allow for a scrolling map (like renalert showed off a while back) may or may not make it into 2.7.

You can customize the "time remaining" and "credits" text (hide it, draw new text elsewhere)

You can hide the player/team list

You can customize the health/shield icon (in normal renegadethis is the little cross next to the health number). Unlike normal renegade, you wont be able to make it flash back and forth (like when you have almost no health)

You will be able to hide the powerup icons (that is the icons and text that display when you get a powerup)

Several engine calls that disarm beacons now use the new Disarm\_Beacon engine call to correctly disarm the beacon.

Bug fixes to a couple of classes (like cGameData) in light of some new discoveries.

Change to Console\_Output so that it can be properly used like printf (so you can do Console\_Output("abc is %d\n",abc) for example)

```
*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 then
use the members of INIClass to read from the ini file.
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
const char *Get_Powerup_Weapon(const char *Powerup); //Get the weapon name that a powerup
will grant if collected
GameObject *Get_Vehicle_Owner(GameObject *obj); //Gets the owner (if any) of a vehicle
TeamPurchaseSettingsDefClass *Get_Team_Purchase_Definition(unsigned long team); //Gets
the TeamPurchaseSettingsDefClass for a team (which you can then read from)
PurchaseSettingsDefClass *Get_Purchase_Definition(unsigned long team, unsigned long type);
//Gets the PurchaseSettingsDefClass for a given team and type (which you can then read from)
const char *Get_Definition_Name(unsigned long id); //Convert a definition/preset ID into a name
unsigned long Get_Definition_ID(const char *name); //Convert a definition/preset name into an ID
unsigned int Get_Team_Cost(const char *preset,unsigned int team); //Get the cost of a preset for
a given team. Returns zero if the preset is not found in any of the purchase terminal data or if it is
one of the free units.
unsigned int Get_Cost(const char *preset); //Get the cost of a preset. Returns zero if the preset is
not found in any of the purchase terminal data or if it is one of the free units.
*void Set_Enlisted(unsigned int team, unsigned int position, unsigned int presetid, unsigned int
stringid, const char *texture); //Change an enlisted soldier for a team
*void Set_Beacon(unsigned int team, unsigned int cost, unsigned int presetid, unsigned int
stringid, const char *texture); //Change the beacon for a team
*void Set_Refill(unsigned int team, unsigned int stringid, const char *texture); //Change the refill

```

for a team

\*void Set\_Preset(unsigned int team, unsigned int type, unsigned int position, unsigned int cost, unsigned int presetid, unsigned int stringid, const char \*texture); //Change a preset given a type and team

\*void Set\_Alternate(unsigned int team, unsigned int type, unsigned int position, unsigned int altpos, unsigned int presetid, const char \*texture); //Change an alternate given a type and a team  
new script, JFW\_Damage\_Occupants\_Death which damages all occupants when the vehicle this script is on dies.

\*new script, JFW\_HUD\_INI which goes on a soldier and loads a new hud.ini (see above for details of that)

\*new script, JFW\_Wireframe\_Mode which goes on a map and sets the wireframe mode for this map

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\_Lock\_Sound which, when poked, plays one sound if the poker doesnt have <key> and a different sound if they do.

new script, JFW\_Vehicle\_Extra which creates an object and attaches it to a given bone of another object. When the object is destroyed, so is the attachment.

new script JFW\_Vehicle\_Extra\_2 which is like JFW\_Vehicle\_Extra but also passes damage done to the host to the attachment. Also, it attaches JFW\_Vehicle\_Extra\_Attach to the object.

new script JFW\_Vehicle\_Extra\_Attach which is used to pass damage done to the attached object to the main object. In order for this to work, both the objects need to have identical settings for armour, skin, health, shield etc.

new script, JFW\_Advanced\_Guard\_Tower\_2. This does the same things as M00\_Advanced\_Guard\_Tower but allows you to customize the locations of the missile and guns relative to the building controller. Unlike JFW\_Advanced\_Guard\_Tower, this uses M00\_Advanced\_Guard\_Tower\_Missile and M00\_Advanced\_Guard\_Tower\_Gun which should prevent any bugs in JFW\_Advanced\_Guard\_Tower\_Missile and JFW\_Advanced\_Guard\_Tower\_Gun from being a problem.

New scripts from WD, MDB\_Weapon\_Scope\_Global, MDB\_Base\_Defense\_Popup\_No\_VTOL, MDB\_Base\_Defense\_Popup\_VTOL\_Only, MDB\_Base\_Defense\_Popup, MDB\_Base\_Defense, MDB\_Base\_Defense\_No\_VTOL, MDB\_Base\_Defense\_VTOL\_Only and MDB\_Send\_Custom\_Enemy\_Seen.

The defence scripts are going to be used for various defences in Reborn (MDB\_Base\_Defense\_Popup\_VTOL\_Only would be perfect for a proper TD-style popup SAM site using the model that is in buildings.zip)

Some new #defines in scripts.h for the values you can pass to Set\_Obj\_Radar\_Blip\_Shape and Set\_Obj\_Radar\_Blip\_Color

This is the complete list of features that will be in 2.7. No scripts exist yet that use the new PT functionality or the new bullet/weapon stuff but I am sure that people (myself included) will write some now that its there to be used

---