
Subject: Westwood Scripts [M02]

Posted by [Troll King](#) on Thu, 28 Jun 2012 10:54:26 GMT

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

Greetings all people and happy modders of Renegade.

(If you wish to see the link right now, because you can't wait to see the scripts. Go down to the bottom of the message.)

A few weeks ago I had a thought in my mind to Decompile the original Westwood scripts.

HOWEVER: I do not claim this is an EXACT copy of the scripts. I've tried to get them as close as I possibly could. I'm still a newb at doing this. Also, I couldn't identify some Script Events (like Created/Custom etc.) and some local class variables. I left their names as Unknown**. I left some comments in the scripts, also at the Unknown events for hints to discover what event it could be.

Commands->Find_Object is often used and I rarely check what object it is looking for so I just named those GameObject *SearchObj**.

DO NOT TRUST THE ActionParamsStructs COMPLETELY BECAUSE IT IS VERY HARD FOR ME AT THIS TIME TO KNOW EXACTLY HOW IT WAS USED, IT'S ONLY A CLOSE REFLECTION HOW IT MUST HAVE LOOKED LIKE.

It is true that all the names of variables are made up by me. All their names are lost FOREVER and are most likely NOT the same as the original. Although all strings are EXACTLY the same as the original.

This release contains the following scripts:

- M02_Encyclopedia_Reveal
- M02_Data_Disk
- M02_Commando_Start
- M02_GDI_Helicopter
- M02_Nod_Jet_Waypath
- M02_Nod_Jet
- M02_Mendoza
- M02_Destroy_Vehicle
- M02_Approach_Vehicle
- M02_Player_Vehicle
- M02_Nod_Vehicle
- M02_Nod_Apache
- M02_Stationary_Vehicle
- M02_GDI_Soldier
- M02_Nod_Convoy_Truck
- M02_Destroy_Objective
- M02_Helipad
- M02_Dam_MCT
- M02_Power_Plant
- M02_Obelisk
- M02_Nod_Soldier

M02_Reset_Spawn
M02_Objective_Zone
M02_Objective_Controller

I believe these are ALL M02 scripts besides 'M02_Respawn_Controller'. I didn't decompile this one just yet because there are too many things in there I don't understand just yet.

I only ask these things from you:

Please do not claim that this release was yours (feel free to decompile them yourself, but do not use my link as yours).

Please do give me feedback about this, these scripts only have cost me alot of time. I'm currently working on the MX0 scripts aswell.

Happy Modding!

Q: So why the M02 scripts you ask?

A: It was a request from zunnie.

Q: Do all the numbers in there resemble the original numbers?

A: Most likely yes, but I cannot guarantee that.

Q: What help text number is this from the Commands->Set_HUD_Help_Text?

A: I have not a clue, nor do I know what all other numbers resemble for example from Create_Logical_Sound.

Q: What does Debug_Message?

A: Dunno.

Q: Can I ask you something about M02_Respawn_Controller?

A: Sure you can. My preference goes to IRC. I'm usually in #mpf from MultiPlayerForums and #tt from BlackIntel.

Special thanks to JonWil, for helping me out by indicating what program I should use etc.

IMPORTANT NOTE: NONE OF THE NEGATIVE FLOATS ARE CORRECT, HAVE TO LOOK INTO IT HOW TO FIX IT!

HERE COMES THE ALMIGHTY LINK TO THEM!! @ @#%%^&

<http://pastebin.com/iC0FtbdD>

File Attachments

1) [Neijwiert_M02_Scripts.h](#), downloaded 360 times

Subject: Re: Westwood Scripts [M02]
Posted by [iRANian](#) on Thu, 28 Jun 2012 19:04:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

did you check what 'Commands->' function is being called by comparing the offset being called with the offsets that are loaded up in Get_Script_Commands()? what tricks/programs did you use

Subject: Re: Westwood Scripts [M02]
Posted by [iRANian](#) on Thu, 28 Jun 2012 19:15:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

that's pretty impressive btw

Subject: Re: Westwood Scripts [M02]
Posted by [Troll King](#) on Fri, 29 Jun 2012 09:09:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Yes I did that. The actual Definition of Script_Commands is in TT.dll and Scripts2.dll is calling from that aswell. so it most likely has scripts.h too or something like that and it would call upon a link up to TT.dll. An example how Commands->Get_Position(); would look like: `*(int (__cdecl **)(char *, int))(dword_101D3700 + 64))(&v63, a2);`

where dword_101D3700 would be the memory position of the beginning of Script_Command_Class and + 64 the offset. I'm using IDA Pro V6.1 with Hex-Rays V1.5.

And thanks

Subject: Re: Westwood Scripts [M02]
Posted by [iRANian](#) on Fri, 29 Jun 2012 12:22:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

ah yeah that's how i'd do it, was kinda hoping you figured out how to automate it so you don't have to look up the offsets every time a ScriptsCommands function get called.

Subject: Re: Westwood Scripts [M02]
Posted by [Troll King](#) on Fri, 29 Jun 2012 15:05:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

I made this list for it, if anybody needs it.

```
dword_101D3700 + 8 = void Debug_Message(char *format, ...);  
dword_101D3700 + 12 = void Action_Reset(GameObject *obj, float priority);  
dword_101D3700 + 16 = void Action_Goto(GameObject *obj, const ActionParamsStruct
```

```

&params);
dword_101D3700 + 20 = void Action_Attack(GameObject *obj, const ActionParamsStruct
&params);
dword_101D3700 + 24 = void Action_Play_Animation(GameObject *obj, const
ActionParamsStruct &params);
dword_101D3700 + 28 = void Action_Enter_Exit(GameObject (GameObject *obj, const
ActionParamsStruct &params);
dword_101D3700 + 32 = void Action_Face_Location(GameObject *obj, const ActionParamsStruct
&params)
dword_101D3700 + 36 = void Action_Dock(GameObject *obj, const ActionParamsStruct
&params);
dword_101D3700 + 40 = void Action_Follow_Input(GameObject *obj, const ActionParamsStruct
&params);
dword_101D3700 + 44 = void Modify_Action(GameObject *obj, int ActionID, const
ActionParamsStruct &params, bool modify_move, bool modify_attack);
dword_101D3700 + 48 = int Get_ActionID(GameObject *obj);
dword_101D3700 + 52 = bool Get_Action_Params(GameObject *obj, ActionParamsStruct
&params);
dword_101D3700 + 56 = bool Is_Performing_Pathfind_Action(GameObject *obj);
dword_101D3700 + 60 = void Set_Position(GameObject *obj, const Vector3 &position);
dword_101D3700 + 64 = Vector3 Get_Position(GameObject *obj);
dword_101D3700 + 68 = Vector3 Get_Bone_Position(GameObject *obj, const char *bone_name);
dword_101D3700 + 72 = float Get_Facing(GameObject *obj);
dword_101D3700 + 76 = void Set_Facing(GameObject *obj, float degrees);
dword_101D3700 + 80 = void Disable_All_Collisions(GameObject *obj);
dword_101D3700 + 84 = void Disable_Physical_Collisions(GameObject *obj);
dword_101D3700 + 88 = void Enable_Collisions(GameObject *obj);
dword_101D3700 + 92 = void Destroy_Object(GameObject *obj);
dword_101D3700 + 96 = GameObject *Find_Object(int obj_id);
dword_101D3700 + 100 = GameObject *Create_Object(const char *type_name, const Vector3
&position);
dword_101D3700 + 104 = GameObject *Create_Object_At_Bone(GameObject *host_obj, const
char *new_obj_type_name, const char *bone_name);
dword_101D3700 + 108 = int Get_ID(GameObject *obj);
dword_101D3700 + 112 = int Get_Preset_ID(GameObject *obj);
dword_101D3700 + 116 = const char *Get_Preset_Name(GameObject *obj);
dword_101D3700 + 120 = void Attach_Script(GameObject *object, const char *scriptName, const
char *scriptParams);
dword_101D3700 + 124 = void Add_To_Dirty_Cull_List(GameObject *object);
dword_101D3700 + 128 = void Start_Timer(GameObject *obj, ScriptClass *script, float duration,
int timer_id);
dword_101D3700 + 132 = void Trigger_Weapon(GameObject *obj, bool trigger, const Vector3
&target, bool primary);
dword_101D3700 + 136 = void Select_Weapon(GameObject *obj, const char *weapon_name);
dword_101D3700 + 140 = void Send_Custom_Event(GameObject *from, GameObject *to, int
type, int param, float delay);
dword_101D3700 + 144 = void Send_Damaged_Event(GameObject *obj, GameObject
*damager);

```

dword_101D3700 + 148 = float Get_Random(float min, float max);
dword_101D3700 + 152 = int Get_Random_Int(int min, int max);
dword_101D3700 + 156 = GameObject *Find_Random_Simple_Object(const char *preset_name);
dword_101D3700 + 160 = void Set_Model(GameObject *obj, const char *model_name);
dword_101D3700 + 164 = void Set_Animation(GameObject *obj, const char *anim_name, bool looping, const char *sub_obj_name, float start_frame, float end_frame, bool is_blended);
dword_101D3700 + 168 = void Set_Animation_Frame(GameObject *obj, const char *anim_name, int frame);
dword_101D3700 + 172 = int Create_Sound(const char *sound_preset_name, const Vector3 &position, GameObject *creator);
dword_101D3700 + 176 = int Create_2D_Sound(const char *sound_preset_name);
dword_101D3700 + 180 = int Create_2D_WAV_Sound(const char *wav_filename);
dword_101D3700 + 184 = int Create_3D_WAV_Sound_At_Bone(const char *wav_filename, GameObject *obj, const char *bone_name);
dword_101D3700 + 188 = int Create_3D_Sound_At_Bone(const char *sound_preset_name, GameObject *obj, const char *bone_name);
dword_101D3700 + 192 = int Create_Logical_Sound(GameObject *creator, int type, const Vector3 &position, float radius);
dword_101D3700 + 196 = void Start_Sound(int sound_id);
dword_101D3700 + 200 = void Stop_Sound(int sound_id, bool destroy_sound);
dword_101D3700 + 204 = void Monitor_Sound(GameObject *game_obj, int sound_id);
dword_101D3700 + 208 = void Set_Background_Music(const char *wav_filename);
dword_101D3700 + 212 = void Fade_Background_Music(const char *wav_filename, int fade_out_time, int fade_in_time);
dword_101D3700 + 216 = void Stop_Background_Music();
dword_101D3700 + 220 = float Get_Health(GameObject *obj);
dword_101D3700 + 224 = float Get_Max_Health(GameObject *obj);
dword_101D3700 + 228 = void Set_Health(GameObject *obj, float health);
dword_101D3700 + 232 = float Get_Shield_Strength(GameObject *obj);
dword_101D3700 + 236 = float Get_Max_Shield_Strength(GameObject *obj);
dword_101D3700 + 240 = void Set_Shield_Strength(GameObject *obj, float strength);
dword_101D3700 + 244 = void Set_Shield_Type(GameObject *obj, const char *name);
dword_101D3700 + 248 = int Get_Player_Type(GameObject *obj);
dword_101D3700 + 252 = void Set_Player_Type(GameObject *obj, int type);
dword_101D3700 + 256 = float Get_Distance(const Vector3 &p1, const Vector3 &p2);
dword_101D3700 + 260 = void Set_Camera_Host(GameObject *obj);
dword_101D3700 + 264 = void Force_Camera_Look(const Vector3 *target);
dword_101D3700 + 268 = GameObject *Get_The_Star();
dword_101D3700 + 272 = GameObject *Get_A_Star(const Vector3 &pos);
dword_101D3700 + 276 = GameObject *Find_Closest_Soldier(const Vector3 &pos, float min_dist, float max_dist, bool only_human);
dword_101D3700 + 280 = bool Is_A_Star(GameObject *obj);
dword_101D3700 + 284 = void Control_Enable(GameObject *obj, bool enable);
dword_101D3700 + 288 = const char *Get_Damage_Bone_Name();
dword_101D3700 + 292 = bool Get_Damage_Bone_Direction();
dword_101D3700 + 296 = bool Is_Object_Visible(GameObject *looker, GameObject *obj);
dword_101D3700 + 300 = void Enable_Enemy_Seen(GameObject *obj, bool enable);

dword_101D3700 + 304 = void Set_Display_Color(unsigned char red, unsigned char green, unsigned char blue);
dword_101D3700 + 308 = void Display_Text(int string_id);
dword_101D3700 + 312 = void Display_Float(float value, const char *format);
dword_101D3700 + 316 = void Display_Int(int value, const char *format);
dword_101D3700 + 320 = void Save_Data(ScriptSave &saver, int id, int size, void *data);
dword_101D3700 + 324 = void Save_Pointer(ScriptSave &saver, int id, void *pointer);
dword_101D3700 + 328 = bool Load_Begin(ScriptLoader &loader, int *id);
dword_101D3700 + 332 = void Load_Data(ScriptLoader &loader, int size, void *data);
dword_101D3700 + 336 = void Load_Pointer(ScriptLoader &loader, void **pointer);
dword_101D3700 + 340 = void Load_End(ScriptLoader &loader);
dword_101D3700 + 344 = void Begin_Chunk(ScriptSaver &saver, unsigned int chunkID);
dword_101D3700 + 348 = void End_Chunk(ScriptSaver &saver);
dword_101D3700 + 352 = bool Open_Chunk(ScriptLoader &loader, unsigned int *chunkID);
dword_101D3700 + 356 = void Close_Chunk(ScriptLoader &loader);
dword_101D3700 + 360 = void Clear_Radar_Markers();
dword_101D3700 + 364 = void Clear_Radar_Marker(int id);
dword_101D3700 + 368 = void Add_Radar_Marker(int id, const Vector3 &position, int shape_type, int color_type);
dword_101D3700 + 372 = void Set_Obj_Radar_Blip_Shape(GameObject *obj, int shape_type);
dword_101D3700 + 376 = void Set_Obj_Radar_Blip_Color(GameObject *obj, int color_type);
dword_101D3700 + 380 = void Enable_Radar(bool enable);
dword_101D3700 + 384 = void Clear_Map_Cell(int cell_x, int cell_y);
dword_101D3700 + 388 = void Clear_Map_Cell_By_Pos(const Vector3 &world_space_pos);
dword_101D3700 + 392 = void Clear_Map_Cell_By_Pixel_Pos(int pixel_pos_x, int pixel_pos_y);
dword_101D3700 + 396 = void Clear_Map_Region_By_Pos(const Vector3 &world_space_pos, int pixel_radius);
dword_101D3700 + 400 = void Reveal_Map();
dword_101D3700 + 404 = void Shroud_Map();
dword_101D3700 + 408 = void Show_Player_Map_Marker(bool onoff);
dword_101D3700 + 412 = float Get_Safe_Flight_Height(float x_pos, float y_pos);
dword_101D3700 + 416 = void Create_Explosion(const char *explosion_def_name, const Vector3 &pos, GameObject *creator);
dword_101D3700 + 420 = void Create_Explosion_At_Bone(const char *explosion_def_name, Gameobject *object, const char *bone, GameObject *creator);
dword_101D3700 + 424 = void Enable_HUD(bool enable);
dword_101D3700 + 428 = void Mission_Complete(bool success);
dword_101D3700 + 432 = void Give_PowerUp(GameObject *obj, const char *preset_name, bool display_on_hud);
dword_101D3700 + 436 = void Innate_Disable(GameObject *object);
dword_101D3700 + 440 = void Innate_Enable(GameObject *object);
dword_101D3700 + 444 = bool Innate_Soldier_Enable_Enemy_Seen(GameObject *obj, bool state);
dword_101D3700 + 448 = bool Innate_Soldier_Enable_Gunshot_Heard(GameObject *obj, bool state);
dword_101D3700 + 452 = bool Innate_Soldier_Enable_Footsteps_Heard(GameObject *obj, bool state);
dword_101D3700 + 456 = bool Innate_Soldier_Enable_Bullet_Heard(GameObject *obj, bool

```

state);
dword_101D3700 + 460 = bool Innate_Soldier_Enable_Actions(GameObject *obj, bool state);
dword_101D3700 + 464 = void Set_Innate_Soldier_Home_Location(GameObject *obj, const
Vector3 &home_pos, float home_radius);
dword_101D3700 + 468 = void Set_Innate_Aggressiveness(GameObject *obj, float
aggressiveness);
dword_101D3700 + 472 = void Set_Innate_Take_Cover_Probability(GameObject *obj, float
probability);
dword_101D3700 + 476 = void Set_Innate_Is_Stationary(GameObject *obj, bool stationary);
dword_101D3700 + 480 = void Innate_Force_State_Bullet_Heard(GameObject *obj, const
Vector3 &pos);
dword_101D3700 + 484 = void Innate_Force_State_Footsteps_Heard(GameObject *obj, const
Vector3 &pos);
dword_101D3700 + 488 = void Innate_Force_State_Gunshots_Heard(GameObject *obj, const
Vector3 &pos);
dword_101D3700 + 492 = void Innate_Force_State_Enemy_Seen(GameObject *obj, GameObject
*enemy);
dword_101D3700 + 496 = void Static_Anim_Phys_Goto_Frame(int obj_id, float frame, const char
*anim_name);
dword_101D3700 + 500 = void Static_Anim_Phys_Goto_Last_Frame(int obj_id, const char
*anim_name);
dword_101D3700 + 504 = unsigned int Get_Sync_Time();
dword_101D3700 + 508 = void Add_Objective(int id, int type, int status, int short_description_id,
char *description_sound_filename, int long_description_id);
dword_101D3700 + 512 = void Remove_Objective(int id);
dword_101D3700 + 516 = void Set_Objective_Status(int id, int status);
dword_101D3700 + 520 = void Change_Objective_Type(int id, int type);
dword_101D3700 + 524 = void Set_Objective_Radar_Blip(int id, const Vector3 &position);
dword_101D3700 + 528 = void Set_Objective_Radar_Blip_Object(int id, ScriptableGameObj
*unit);
dword_101D3700 + 532 = void Set_Objective_HUD_Info(int id, float priority, const char
*texture_name, int message_id);
dword_101D3700 + 536 = void Set_Objective_HUD_Info_Position(int id, float priority, const char
*texture_name, int message_id);
dword_101D3700 + 540 = void Shake_Camera(const Vector3 &pos, float radius, float intensity,
float duration);
dword_101D3700 + 544 = void Enable_Spawner(int id, bool enable);
dword_101D3700 + 548 = GameObject *Trigger_Spawner(int id);
dword_101D3700 + 552 = void Enable_Engine(GameObject *object, bool onoff);
dword_101D3700 + 556 = int Get_Difficulty_Level();
dword_101D3700 + 560 = void Grant_Key(GameObject *object, int key, bool grant);
dword_101D3700 + 564 = bool Has_Key(GameObject *object, int key, bool grant);
dword_101D3700 + 568 = void Enable_Hibernation(GameObject *object, bool enable);
dword_101D3700 + 572 = void Attach_To_Object_Bone(GameObject *object, GameObject
*host_object, const char *bone_name);
dword_101D3700 + 576 = int Create_Conversation(const char *conversation_name, int priority,
float max_dist, bool is_interruptable);
dword_101D3700 + 580 = void Join_Conversation(GameObject *object, int

```

```

active_conversation_id, bool allow_mode, bool allow_head_turn, bool allow_facing);
dword_101D3700 + 584 = void Join_Conversation_Facing(GameObject *object, int
active_conversation_id, int obj_id_to_face);
dword_101D3700 + 588 = void Start_Conversation(int active_conversation_id, int ActionID);
dword_101D3700 + 592 = void Monitor_Conversation(GameObject *object, int
active_conversation_id);
dword_101D3700 + 596 = void Start_Random_Conversation(GameObject *object);
dword_101D3700 + 600 = void Stop_Conversation(int active_conversation_id);
dword_101D3700 + 604 = void Stop_All_Conversations();
dword_101D3700 + 608 = void Lock_Soldier_Facing(GameObject *object, GameObject
*object_to_face, bool turn_body);
dword_101D3700 + 612 = void Unlock_Soldier_Facing(GameObject *object);
dword_101D3700 + 616 = void Apply_Damage(GameObject *object, float amount, const char
*warhead_name, GameObject *damager);
dword_101D3700 + 620 = void Set_Loiters_Allowed(GameObject *object, bool allowed);
dword_101D3700 + 624 = void Set_Is_Visible(GameObject *object, bool visible);
dword_101D3700 + 628 = void Set_Is_Rendered(GameObject *object, bool rendered);
dword_101D3700 + 632 = float Get_Points(GameObject *object);
dword_101D3700 + 636 = void Give_Points(GameObject *object, float points, bool entire_team)
dword_101D3700 + 640 = float Get_Money(GameObject *object);
dword_101D3700 + 644 = void Give_Money(GameObject *object, float money, bool entire_team);
dword_101D3700 + 648 = bool Get_Building_Power(GameObject *object);
dword_101D3700 + 652 = void Set_Building_Power(GameObject *object, bool onoff);
dword_101D3700 + 656 = void Play_Building_Announcement(GameObject *object, int text_id);
dword_101D3700 + 660 = GameObject *Find_Nearest_Building_To_Pos(const Vector3 &position,
const char *mesh_prefix);
dword_101D3700 + 664 = GameObject *Find_Nearest_Building(GameObject *object, const char
*mesh_prefix);
dword_101D3700 + 668 = int Team_Member_In_Zone(GameObject *object, int player_type);
dword_101D3700 + 672 = void Set_Clouds(float cloudcover, float cloudgloominess, float
ramptime)
dword_101D3700 + 676 = void Set_Lightning(float intensity, float startdistance, float enddistance,
float heading, float distribution, float ramptime);
dword_101D3700 + 680 = void Set_War_Blitz(float intensity, float startdistance, float enddistance,
float heading, float distribution, float ramptime);
dword_101D3700 + 684 = void Set_Wind(float heading, float speed, float variability, float
ramptime);
dword_101D3700 + 688 = void Set_Rain(float density, float ramptime, bool prime);
dword_101D3700 + 692 = void Set_Snow(float density, float ramptime, bool prime);
dword_101D3700 + 696 = void Set_Ash(float density, float ramptime, bool prime);
dword_101D3700 + 700 = void Set_Fog_Enable(bool enabled);
dword_101D3700 + 704 = void Set_Fog_Range(float startdistance, float enddistance, float
ramptime);
dword_101D3700 + 708 = void Enable_Stealth(GameObject *object, bool onoff);
dword_101D3700 + 712 = void Cinematic_Sniper_Control(bool enabled, float zoom);
dword_101D3700 + 716 = int Text_File_Open(const char *filename);
dword_101D3700 + 720 = bool Text_File_Get_String(int handle, char *buffer, int size);
dword_101D3700 + 724 = void Text_File_Close(int handle);

```


dword_101D3700 + 728 = void Enable_Vehicle_Transitions(GameObject *object, bool enable);
dword_101D3700 + 732 = void Display_GDI_Player_Terminal();
dword_101D3700 + 736 = void Display_NOD_Player_Terminal();
dword_101D3700 + 740 = void Display_Mutant_Player_Terminal();
dword_101D3700 + 744 = bool Reveal_Encyclopedia_Character(int object_id);
dword_101D3700 + 748 = bool Reveal_Encyclopedia_Weapon(int object_id);
dword_101D3700 + 752 = bool Reveal_Encyclopedia_Vehicle(int object_id);
dword_101D3700 + 756 = bool Reveal_Encyclopedia_Building(int object_id);
dword_101D3700 + 760 = void Display_Encyclopedia_Event_UI();
dword_101D3700 + 764 = void Scale_AI_Awareness(float sight_scale, float hearing_scale);
dword_101D3700 + 768 = void Enable_Cinematic_Freeze(GameObject *object, bool enable);
dword_101D3700 + 772 = void Expire_Powerup(GameObject *object);
dword_101D3700 + 776 = void Set_HUD_Help_Text(int string_id, const Vector3 color);
dword_101D3700 + 780 = void Enable_HUD_Pokable_Indicator(GameObject *object, bool enable);
dword_101D3700 + 784 = void Enable_Innate_Conversations(GameObject *object, bool enable);
dword_101D3700 + 788 = void Display_Health_Bar(GameObject *object, bool display);
dword_101D3700 + 792 = void Enable_Shadow(GameObject *object, bool enable);
dword_101D3700 + 808 = void Set_Screen_Fade_Color(float r, float g, float b, float seconds);
dword_101D3700 + 812 = void Set_Screen_Fade_Opacity(float opacity, float seconds);

Subject: Re: Westwood Scripts [M02]
Posted by [iRANian](#) on Fri, 29 Jun 2012 15:05:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

If you open the linux fds it'll show you more of the symbols

Subject: Re: Westwood Scripts [M02]
Posted by [Troll King](#) on Fri, 06 Jul 2012 12:37:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

FILE ADDED
