FLAME'S GUIDE TO HUDS

The idea behind making this was so people would know how to edit/move/replace specific parts of the TF2 default and custom HUDs. I will try to explain a majority of the different elements of the HUD as well as try to point out where/what every different res file and its content dictates. Some things however I am not sure of due to myself never wanting/ needing to edit. However I will try to cover the basic elements of the HUD as well as what each file and such does and how to manipulate it to your needs, thus making it easier for you to eventually do everything yourself and to your liking.

INDEX

The majority of the HUD editing process is rather simple once you understand what you are doing. Most of the headache is just the positioning of objects to the most ideal pixel as well as constantly playing a guess-and-check game moving things back and forth until they are exactly where you want them and making things not overlap. Oh, and Control+F Is your friend for navigating this document.

1. WHERE/HOW TO GET THE HUD FILES

Ok this is probably where you'll want to start. First you'll want to download a program called GCFScape. This program is free and what it does is open up the GCF files the game uses in a .rar sort of fashion, allowing you to extract whatever files you'll need to your computer.

Once the program is opened up you'll want to do File->Open. Then you need to navigate yourself to C:/ProgramFiles/Steam/Steamapps. In this file will be a majority of the GCF files for any and all of the games you have purchased through steam. In the list open up the file called "Team Fortress 2 Content." Once you're in there navigate to the /tf/ directory. When you're in there you'll want to extract two folders, tf/resource and tf/scripts and extract the two folders anywhere accessible, ideally the desktop.

On your desktop should now be the resource and scripts folder which are the two most important folders you'll need to edit your HUD. Now that those two folders are on your desktop navigate in your own computer to the following directory: C:/ProgramFiles/Steam/Steamapps/YourUserName/TeamFortress2/tf/. Once you're there put the recently extracted scripts and resource folders from the GCF into the /TeamFortress2/tf/ directory so the files should be tf/resource/ and tf/scripts/.

2. THE SETUP FOR OPTIMAL HUD EDITING

Ok so this step is not what I'd call necessary but it will definitely make your life a bit easier. First you want to right click on TF2 on your Games list and go to Properties. From there you want to go to the "General" tab and click "Set Launch Options." Once you're there add two things to the command line, –windowed and –noborder. This will make alt-

tabbing between you're .res files and the game much simpler. Also you'll want to bind a key in tf2 to hud_reloadscheme. This will bind that key to reload your HUD after you edit the .res file so you can check what you have just done and make sure it did what you wanted it to do. It's much easier than typing it into console every time and save a lot of time when it comes to checking/fixing the HUD. Being able to alt-tab quickly and check your work makes it a lot faster than using fullscreen mode and not having a key bound. This will just make it a bit quicker and not ruin your patience level.

3. EXPLANATIONS OF THE ENTITIES

Ok so when you're modifying the .res files there will be a majority of different values you can change which to some will look like a mess and be pretty confusing. The truth of it is though that they're fairly simple to read and make a lot of sense once you understand the different meanings and values. Here is a list of the majority of the entities you'll need to know. USE NOTEPAD TO OPEN THE .RES FILES.

Controlname – You'll probably want to not touch these unless you're sure you can change them and even than some combinations won't work.

Fieldname – This should say the same exact thing as the name of the entity. If it does not the game will get confused and sometimes do weird things to your HUD.

Xpos – This is the relative x position to the left side of the screen or individual HUD piece. There are a few different ways to enter x positions. C before a number makes it relative to the center ex. C0 or c-12 meaning 12 to the left of the center. There is also R which bases it off the right side such as r5 meaning 5 away from the right side.

Ypos – The same as the xpos except this dictates the vertical positioning of whatever you're trying to move. Most of the time though you'll want to stay away from the C and R with this and just stick to specific value.

Zpos – This is the "layer" it's on. You can make things have negative zpos if you want them at the bottom but it's important to change this every now and then especially if you want borders on your text or something to show up on top of something else.

Wide – This dictates the width of what you're changing. If it's a fill box it'll make it wider and taller but if it's a background it'll give you more space to move what it's containing.

Tall – The same as wide except it's the height and not the width.

***_minmode** – Whenever you see _minmode it means the value for when you have cl_hud_minmode enabled. For example xpos_minmode is the x positioning for when you have minimal HUD on and tall_minmode is the height when minimal HUD is on.

Visible – Whether or not you can actually see the entity. Visible 1 means it's on all the time and visible 0 means it's never on or only on when it needs to be.

Enabled – This means it works pretty much. If enabled is set to 0 you will never see the entity and it will do close to nothing. There are exceptions to the visible and enabled values but for the majority enabled 1 means it'll show up and enabled 0 means it won't.

Font – This dictates the font the entity uses. These are found in the clientscheme.res file in the tf/scripts/ folder.

FgColor – This is the color of the font you're using. These are also located in the clientscheme.res file in the tf/scripts/ folder. You can make your own fonts in the order Red Green Blue Opacity. Ie. 0 0 255 255 would give you an opaque blue color for whatever you're using it for and 0 255 0 150 would give you a semi-transparent green.

TextAlignment – Bases where the font is started from sort of like Microsoft Word. Left means the font is left aligned, Right means its right aligned, and Center means it's aligned to the center.

Image – This is the file used as the picture or background. You can change them but a lot of the time you'll just want to turn images to visible 0 or enabled 0 since a majority of them are just weird looking backgrounds.

Scaleimage and Pincorner – Not sure what they do since changing them never really did anything for me. Scale image I would imagine scales the image based on your resolution and Pincorner I have no idea what it does.

These are the majority of the entities you'll need to use. There are a few others but they are pretty self explanatory or not too necessary to know. The other ones include specific team images or dull and bright text, which seem to do nothing for me. Just stick to editing the above values and there isn't much you can't do to the thing you want to edit. Most of the best custom HUDs were done just by changing fonts and x/y positioning and putting nice

looking borders on fonts. Making the visibility to 0 on a lot of the background images also makes life a bit less cluttered.

4. CLIENTSCHEME.RES FILE

This file is located in the tf/resource/ directory. In this file (which you can open with notepad, just like every other .res file) you can add/delete fonts, and add colors. This file is going to be primarily used to check to see which fonts you can use and possibly add a custom combination of font style and font size. You can also add colors to this file if you want a special color for something and want to use it a lot of the time. Adding colors is as easy as just making a new name and putting in a value for the red green blue values. You can pretty much just copy a font, hit enter, and paste the font under it, change the name and the values and use it in the rest of your .res files when you need to. The same can be done for fonts.

To add a custom font you'll need the font (.ttf) file in your tf/resource/ directory. Once it's there you can add it to the list of fonts pretty much anywhere in the list in the middle/bottom of the clientscheme.res file. The different entities for fonts are pretty straightforward when you read through them. The only one not in all of them adds a border to the font. If you want a border on a font without a border, simply copy and paste the font underneath itself, change the name and add a value called "outline" and set it to "1." This will put an outline on the font.

One important thing to remember though is that changes to the clientscheme.res file cannot be seen/ checked until you restart TF2. For example adding a new font or color to clientscheme.res followed by calling the font in a different entity will not work when you do a hud_reloadscheme since the game loads clientscheme.res separately from the rest of the HUD dictating files.

This file is not even really needed as far as HUD editing goes but sometimes you can't get the text to fit where you want it and the only way to make it work is to create a new font yourself based off some of the existing ones. Other than that this file is really only necessary for reference to the font names as well as the color combinations.

5. HUDLAYOUT.RES FILE

This file is pretty much the backbone of the entire HUD. It is important to know what this file is for as well as when you'll need to use it. Think of this file as the team, the specific .res files as the player, and the entities in the .res file as the weapons, except for TF2 HUDs if that makes sense. Pretty much the only thing you'll want to edit in this file is the x and y pos and xpos and ypos _minmode values. When you edit an entity in here it essentially moves the entirety of the other .res files in the resource/ui/ directory.

What this means is that when you move the x and y pos in here for say HudAmmoWeapons it will move the whole background and entity for the HudAmmoWeapons.res file in the resource/ui/ directory where you'll edit the actual numbers and relative locations. You'll understand what I mean when you move something in here since it is kind of hard to understand. I will not go into the meaning of all of the different entities in this file since it could take another 5 pages but just know this file dictates things on a larger scale so if you want to move something from the bottom right of your screen to the top left and the specific tf/resource/ui .res file is unable to do it, odds are what you're looking to do is change the positioning of its entity in the hudlayout.res file.

This file contains everything pretty much except the scoreboard and the spectator HUDs. This is the reason why you can't get your teams status on the normal HUD since there is no entity in this file for it. You can add some things to this file but it's typically not necessary. The target IDs, medic target, ammo, health cross, and a number of other things have their x and y position in this file making it a very big part of HUD editing. This is also the file that usually gets messed up by updates so make sure you back up your HUD as soon as you feel like you're satisfied or whenever you want just to be safe.

In the event of an update however you'll want to check your hudlayout.res next to the new one provided by the update and change the values in the new one, not replace the file. Replacing the file will make you lose some new things. For example the classless update changed the spy charge bar. So by replacing the new hudlayout with the old one, you would undoubtedly lose your spy, jarate, and sandman ball bar causing a headache and a flood of questions toward the HUD editors.

That's about as in-depth as you can get about this file. The only thing you'll really want to do in here is change x and y positions since there isn't much else you can do, except maybe make certain things wider and taller. Just know if you have a problem moving something in more specific file, the reason is probably the width/height or the x and y pos.

6.THE SCOREBOARD

The scoreboard, shown by hitting tab, is dictated by only ONE file. The file is scoreboard.res located in the tf/resource/ui/ directory. In this file you will find a bunch of different editable entities. The majority of them you'll probably not want to touch such as the one labeled "scores" but there are a few you may want to mess around with regardless.

BlueScoreBG – This is the colored background behind the word BLU on the top left of your scoreboard. Only new thing is this one is the src_corner_height/width and draw_corner_width. These dictate the roundness/thickness of the corners of the image. Not really necessary but can turn the borders into solid rectangles in other entities.

RedScoreBG – Same thing except it's the image for the background behind the word RED on the top right of your scoreboard. You can replace the images with colors if you know how. The easiest way is to copy and paste something that is just a colored box and rename the fieldname to the entity name and change the rest of the values accordingly.

MainBG – The background image behind the majority of the scoreboard. It's the gray panel behind the player list.

Blue/RedTeamLabel – Pretty much the name of the Blue team, BLU or customized

Blue/RedTeamScore – Displays the score of the Blue team

Blue/RedTeamScoreDropShadow – The shadow behind the score to make it look fancy

Classimage – This is the picture that shows up when you hit tab to tell you what class you are playing. Ie. when you're a scout and you look at the scoreboard a scout is on the bottom left. You can get rid of it if you want. Only reason I'm telling you about it.

The rest of the things in that file are pretty straightforward as to what they dictate. You can get rid of a lot of them if you want a minimalistic scoreboard but at the same time I don't see the big deal in having a large scoreboard. If you wanted a 6v6 – style scoreboard you could just change the height on the "scores" entity and get rid of the backgrounds and/or change them to something a bit smaller as well as change the fonts on the team labels and scores and such.

Just make sure when you change a font or font placement that you change the placement by the same value on its shadow or else it will look silly. The scoreboard is hardcoded into the game and there is currently no way known to make it show up when +scores is not enabled.

7. THE WEAPON AMMO

The majority of people do not like where the current weapon ammo is located or how it looks. This is one of the things which is moved on the greater scale by finding the HudAmmoWeapons value in the hudlayout.res file. You can then change the specific values in the file named HudAmmoWeapons.res in the tf/resource/ui/ directory. There are a number of important things you must know/understand to edit these values since it is not just one value to change the ammo locations for all classes/ weapons.

HUDAmmoWeaponBG – This is that sloppy looking background image in the corner of the default ammo HUD. Id set the visible to 0 on this if you want, I did. Personal preference.

AmmoInClip – How much ammo you have loaded. This is only for weapons with a clip though. For example the Heavy minigun can't be reloaded so it doesn't apply to this. Soldier rockets and scattergun shots however which have reserve ammo are dictated by this entity.

AmmoInReserve – How much ammo you have left to reload with. If you have 6 shots in your scattergun and 18 left to shoot, this will display the 18 or whatever other weapon reserve ammo.

AmmoNoClip – This is for ammo values which do not have a reserve. The Heavy's minigun, the Pyro's flamethrower, and the Pyro's flaregun are examples of this. Weapons with no clip use this entity and that's pretty much all there is to it.

8. THE HEALTH METER

Lots of people enjoy editing this one. The default location of the health meter is pretty silly and the size of it is not really ideal for some people. Again this one has its overall x and y pos value stored in hudlayout.res I believe and the more intricate stuff is located in HudPlayerHealth.res in the tf/resource/ui/ directory. There are only a few things in here.

PlayerStatusHealthImage – This is the inner part of the health cross. The white part that turns red when you're lit. If you make this bigger or smaller, you need to make the rest of the things in here bigger or smaller.

PlayerStatusHealthImageBG – This is that gray outer layer of the health cross. It should be wider/taller than the normal image or else you won't be able to see it.

PlayerStatusHealthBonusImage – This is the part that blinks when you're overhealed or damaged. Make this much larger than the background just not too big or it'll look like you're in ammomod when you only have a small overheal. Just make sure you can tell when you're overhealed or have less than 50% health.

PlayerStatusHealthValue – The numerical value of how much health you have. You'll probably just want to change the font or copy and paste it to make a second one to put somewhere else or put a shadow on it but that's what this does.

9.THE CONTROL POINTS

Some people like larger control point icons, some people like small ones. I personally like them at the default size even with minimal HUD on. If the icons are too small, it's easier to miss a backcap happening or not know whether a point is being capped or not. I don't know why anyone would change the location of them but the width/height can be changed to your liking. The file for this one is ControlPointIcon.res in the tf/resource/ui/ directory.

ControlPointIcon – This is the positioning of the squares on CP maps. Not much to change except for the width and height for minmode if you want them bigger or something.

CapPlayerImage – This is the image of the little human model things that show up on the CPs when someone is capping them.

CapNumPlayers – This is the multiplier numbers next to the little human image thing to tell you how many people are capping the point ie. x2 or x3 or whatever.

10. THE PLAYER CLASS IMAGES

These are semi useless since if you need a picture to tell you what class you're playing, you're doing it wrong. HudPlayerClass.res in the tf/resource/ui/ directory is where you want to go if you want to remove the class image and the background associated with it. I used the background as a background for my health cross in my HUD which required editing the zpos in hudlayout.res but in practicality, this is the most useless of the different HUD elements.

PlayerStatusClassImage – The image of the class you are which appears in the bottom left corner of the HUD.

PlayerStatusSpyImage – The image of the class you're disguised as when you play spy which shows up next to the spy's spy class image when you're currently disguised.

PlayerStatusSpyOutlineImage – This is the border on the spy that flickers when you bump into someone or are currently decloaking.

PlayerStatusClassImageBG – This is the background color panel behind the class image. When you're on the blue team it's a blue shape and when you're on red it's a red shaped object.

11. THE MEDIC UBERCHARGE METER

A majority of the people playing medic in this day and age do not like the fact that the medic ubercharge meter is located in the bottom right of the screen and gets overlaid by incoming steam messages or notifications for updates. The file for this one is called HudMedicCharge.res and it is located in the tf/resource/ui/ directory. There are a small number of variables to change for this one. But most people just hate the positioning.

Background – This is that background image that lays behind the ubercharge meter in the bottom right. You can ditch this if you want or keep it if you have some better plans for it.

Chargelabel – This is what tells you how much uber you have. Usually looks something like UberCharge: 98% or whatever your value is currently.

Chargemeter – This is the physical meter that progressively fills up when you're healing people. Can be resized or moved wherever if you want it to be.

HealthClusterIcon – Just a little icon that shows up with a bunch of healing plus signs but only if you're not using the minimal HUD. Not really necessary but it's there if you want it.

12. THE SANDMAN/JARATE/SPY CLOAK METERS

This was recently changed with the classless update to incorporate all three bars into one. The reason for this I think was to reduce bandwidth and condense the amount of files needed to modify the HUD. This is located in HudItemEffectMeter.res in the tf/resource/ui/ folder.

There are only a few things that are in this file but it's important to know what each of them does.

ItemEffectMeterBG – This is the background image behind the meters. Not necessary but it's there for whatever you want to do with it.

ItemEffectMeterLabel – This is the wording that tells you what the meter is dictating. It'll say Ball for the Sandman or Cloak for the Spy and Jar for the Jarate.

ItemEffectMeter – This is the physical meter on the HUD that refills or empties based on what you're doing. You can make it bigger or move it or whatever you want to do.

13. THE CTF HUD

This one's a doozie. When you edit the rest of the HUD try going to a CTF map just to make sure that nothing gets overlapped by the absurdly large parts of the CTF HUD. This one is controlled by HudObjectiveFlagPanel.res in the tf/resource/ui/ directory. There's not much to see here. The only thing you need to make sure you do when editing this one is to make sure you move the shadows of the numbers as well as the numbers. There are a lot of different components to this HUD element though.

Left/RightSideBG – This is just the background image for the CTF Hud. Its defaulted to the blue thing on the left and the red one on the right.

OutlineBG – This is the border on the Left/RightSideBG Image. I got rid of the two of them but if you want to resize/move it, it's this you want to edit.

Blue/RedScore – The numerical value for how many caps each team has. They are accompanied by their respective shadows.

PlayingTo – This is the small panel in the middle that says what you're playing to as far as caps. For example in a pub on turbine it may say Playing To: 3 meaning 3 caps or nothing if it's like a 24/7 server. This also has a background which is defaulted to a grey oval-esque box.

Blue/RedFlag – These are the circular compasses on the left and right which tell you the current location of the two intels. If you move one to the left make sure you move the other one to the right by the same amount or things will be off centered and look sloppy.

I'm not 100% sure as to what the rest of the things in that file do so I won't comment on them. I think they are the images that show up on your screen when you are personally carrying the intelligence briefcase. The CTF HUD in my opinion is one of the hardest to edit since it requires working with every other HUD element without overlap and can cause a headache when you think you've done something perfect but it only works for a 5CP HUD. The Payload HUD is a bit of a problem as well but I'll discuss that later.

14. THE PAYLOAD/PAYLOAD RACE HUD

This one is a bit of a headache since none of the pieces to this HUD are attached to each other. If you make the cart track smaller, you need to manually move the house icon things as well as the end of the track icons. The two files you need for the PL/PLR HUD are ObjectiveStatusEscort.res and ObjectiveStatusMultipleEscort.res in the tf/resource/ui/ folder for payload and payload race respectively. While I am well aware of the fact that most people do not enjoy playing PLR maps, many people do occasionally find themselves in a server on Badwater or Pipeline and having an overlapping HUD is kind of annoying and angering. Here are the different elements to these two files.

LevelBar – This is the track image. It can be resized, moved, deleted whatever. This is the track on the bottom of the HUD the cart moves along.

HomeCPIcon – This is the image on the left side of the track representing your base and "home." If you resize this you'll probably have to move it so it lines up with the track again. The two of them need to be moved together or things will get messed up.

EscortItemImage – This is the picture of the cart. The cart image needs to have the same width as the track I think or the cart will overshoot the track or vice versa.

SpeedBackwards – This is the arrow pointing to the left that shows up when the cart is going in reverse.

CapPlayerImage – This is the little human thing that shows up next to the multiplier telling you that someone is on the cart.

CapNumPlayers – This is the numerical value of how many people are pushing the cart. When a scouts on it, x2 three people is a x3, But this is the #.

For the payload race HUD, the majority of the stuff is still located in the ObjectiveStatusEscort.res but the PLR specific things are in ObjectiveStatusMultipleEscort.res. There are only like 3 or so things in here but they're pretty important.

Blue/RedEscortPanel – These are the blue and red tracks for the PLR HUD. The width and height of the two escort panels should be equal or it'll just look sloppy.

15. THE KILLED-BY FREEZECAM

If you play TF2 as well as I do, you see a lot of this HUD element. You can move the boxes telling you their name as well as move/remove the Nemesis icon, and even get rid of those annoying boxes telling you the weapon the guy who killed you is holding even though I'm pretty sure Valve added an option to disable it without the need for HUD edits. The file you'll want to use for this one is FreezePanel_Basic.res and FreezePanelKillerHealth.res in the tf/resource/ui/ directory. There are a few things in here you can play with.

NemesisPanelBG – The black background image behind the word nemesis and the icon when someone is dominating you.

NemesisIcon – This is the icon that shows up that says Nemesis with the two fists punching each other.

NemesisLabel – This is the name of the person who is dominating you.

FreezePanelBG – The colored background behind the ID that shows up saying Killed By: Whoever and can be moved or resized along with the rest of the items.

FreezeLabelKiller – The information of the person who killed you, more specifically just the name of the person who killed you

FreezePanelHealth - This is the icon that shows how much health the person had in the form of the health cross.

ItemPanel – This is what you would have wanted to turn off if you wanted to get rid of the annoying So-and-so is carrying the: Whatever weapon. Valve added an option to turn it off so now this doesn't really need to be edited.

The Health value for those who killed you can be manipulated to show numerical values if you want but that needs to be done in FreezePanelKillerHealth.res. I'll get into how to edit

the health crosses to show a numerical value later but the majority of the aesthetic changes are done in FreezePanel_Basic.res.

16. THE MEDIC "COME HEAL ME" ICONS

Ok so these are those speech bubble looking things with health crosses inside them which show up when you are playing medic and someone calls for medic. I personally like to make these bubble icons smaller but some may want to make them larger based on personal preference. The file for this is MedicCallerPanel.res and it is also in the tf/resource/ui/ directory. There's not much in here to edit but there are a few crucial elements.

CallerBG – This is the physical background speech bubble. You can get rid of this. I've seen a few medics make theirs nonexistent and some make them bigger so it's harder to miss someone calling for you.

CallerBurning – This is the icon that shows up in the speech bubble for when a player is on fire. If you resize this one you might want to resize the next one as well.

CallerHealth – This is the icon that pops up when someone is calling in general. It becomes a red plus sign when someone is low on health and a white one if someone doesn't really need it too much.

After writing this section I'm beginning to wonder if it's possible to make it so you can see numerical health values on the people you're healing. Could be pretty advantageous for medics to keep track of who's overhealed without the use of particle edits but I'm not positive as to whether or not this is possible.

17. THE PUB SPECTATOR HUD

This is the HUD that shows up when you're in a pub and choose team Spectator. It has those gray bars on the top and bottom with a lot of unnecessary info if you're not using the minimal HUD and just a top gray bar if you are using the minimal HUD. Not much to see here except the respawn timers, the target IDs (not dictated in this HUD file) and the gray bars that you might want to get rid of. The file for this is Spectator.res in the tf/resource/ui/ directory and it's pretty self explanatory.

TopBar – This is the gray bar that is on the top of the spectator HUD that a lot of people seem to dislike. The trick with this one is that setting visible to 0 doesn't work. You need to make the tall value 0 to get rid of this for some reason but that's how to do it

BottomBar – This is the gray bar that is on the bottom of the screen when you have cl_hud_minmode set to 0. It contains the hints and stuff as well as map name but turning the minimal HUD on gets rid of this bar and all of its contents.

ReinforcementsLabel – This is the respawn timer. You can change the font, move it, make it bigger, whatever you want. This is what shows up when you're dead saying Respawn In:

The rest of the things in here are only visible when minimal hud is disabled. In the event you do use the default large HUD you can move the rest of the items in this HUD to your liking in that bottom gray bar. The itempanel part of this HUD is also the box that shows you what weapon the person is carrying and can be gotten rid of if you want, but there is an advanced multiplayer option to turn these off in the TF2 Options menu.

18. THE TOURNAMENT SPECTATOR HUD

This was probably the best part of any recent update. It was designed for a 6v6 style play and rearranging/playing with this HUD is pretty simple. There is SpectatorTournament.res which dictates the positioning of everything and SpectatorTournamentGUIHealth.res to change the health cross and maybe add a numerical value to the Tournament HUD player health icons. The two files are also in tf/resource/ui/ and have a pretty large range of modifications available between the two of them.

SpecGui – This is the most important one. The base values are where the 1st person's "player box" shows up, the deltas are the spacing between the individual boxes which should be a few values larger than your wide and tall values for x and y respectively.

PlayerName – The name of the player defaulted to the bottom of the individual boxes

Classimage – The picture that shows up on the left of the boxes to tell you what class the player is currently playing.

HealthIcon – The plus sign that gives you a relative feel for how much health the person has which is defaulted to the right side of the box.

RespawnTime – The countdown timer for dead players to show much time left until they spawn. Defaulted to the bottom right underneath the HealthIcon.

ChargeAmount – This is the percentage that shows up on medics to tell you what percent their uber is at.

SpecIndex – The "slot" the player is assigned to make switching to them from spectator simpler. Example: hitting a key bound to slot1 will make you spectate the player with the number 1 in their box.

TopBar and BottomBar are the same as they are in the Spectator.res file and Reinforcementlabel does the same thing as the other one as well which is tell you the respawn timer for yourself. The numerical health values are dictated by SpectatorTournamentGUIHealth.res which follows the same layout as the other health icon files but I'll get more in-depth in a different section.

19. THE ENGINEER'S METAL VALUE

This one only has its own section because Valve gave it its own res file for some reason. The name of the file is not easy to find since it really doesn't give away any hints as to what it controls. The name of the file is HudAccountPanel.res and it is in the tf/resource/ui/ folder.

AccountBG – This is the background image for the metal icon and value.

MetalIcon – This is the wrench icon that shows up in the background image which is next to the amount of metal you have as engi. You can resize this and move this to use it as a background as a lot of different HUDs do.

AccountValue – This is the numerical value of how much metal you currently have. The most you'd probably want to do with this is maybe edit the font or give it a shadow or something.

Still not completely sure as to why the name for the Metal Value file is called Account Panel but it's understandable if you had trouble finding the right file to adjust the way your metal icon looked. The rest of the engineer files are located in the files for the build menus and the icons in the top left but I see no real reason to touch these. The files are there as well though if you do want to edit the engineer HUD for whatever reason. Some people don't like the background colors or images and some people want different locations of the elements to make seeing entrances and exits a bit simpler. I see no need for modifying the Engineer HUD.

20. THE TARGET IDS

This file is usually the one that makes or breaks a good HUD. The default Target ID files are kind of bulky and in the middle of the screen making it an eyesore when you're in a fight and see a spy or something causing a majority of your screen to be covered up. There are three parts to this HUD and the x/y positioning of them is done in hudlayout.res. The different elements are MainTargetID, SpecatatorID, and SecondaryID.

CMainTargetID – Who you are looking at in game, or even a building

CSpecatatorTargetID – Who you're looking at when you're in spectator mode

CSecondaryTargetID – Who you are healing.

These entities are located in the hudlayout.res file and are good to know about for the positioning of them but the physical ID aesthetics are controlled in the file called TargetID.res which is located in the tf/resource/ui/ directory. There are a lot of variables and you need to make sure not to edit too many of them or change some important defaulted values.

TargetIDBG – This is the background image that shows up behind the name and the ubercharge and health meter of the person who you are looking at.

TargetIDBG_Spec_Blue/Red – This is the background image that is displayed when you look at someone on the red team or blue team when you are in spectator mode.

TargetNameLabel – This is the name of the person you are looking at/spectating or the default information for a sentry such as who built it.

TargetDataLabel – This is the secondary information of the person you are looking at or speccing. This is usually only there when spectating a medic since it tells you the ubercharge percentage or the upgrade progress on an engineer's building.

SpectatorGUIHealth – This is the health cross and dictates when it starts to blink and the positioning of it with respect to the rest of the Target ID.

There is a way to make it so you can see numerical health values for everything. This includes the opposite team when you are playing spy, the player/building you are looking at, the person you are healing and the person you are spectating. The way to do this is fairly simple and I will discuss it in the next section. Do not play with the visible values for the target ID things because it will cause a headache, especially when you start messing around with the background images. You can add things to this HUD but I would suggest not playing around with the enabled/visibility values.

21. EVERYONE ELSE'S HEALTH CROSSES

There are a fairly large number of .res files you can edit to get the different values to show up on the different elements of your HUD. You can make numerical values on your Target IDs, your spy disguise panel, your Specator Tournament HUD and on the freezecam shot of the person who killed you (very useful). There are a number of files which do this for the different pieces and all of them are located in the tf/resource/ui/ folder.

FreezePanelKillerHealth.res – This one dictates the health cross and value for the freezecam shot of the person who killed you

SpectatorTournamentGUIHealth.res – This one dictates the health cross and value for the tournament spectator HUD crosses in the player boxes allowing you to see everyone's health at all times in spectator.

SpectatorGUIHealth.res – This one is for the TargetIDs, the spy disguised as panel, and the engineer's buildings. This one is the most important if you're a medic.

These files are the newest fixes for the more current HUDs. They all pretty much say the same things in terms of content and numerical values can be added to the bottom of all three of them if you want. All you need to do is take the following piece of code and put it inside the file toward the bottom or top wherever you want.

```
"PlayerStatusHealthValue2"
{
    "ControlName"    "Label"
    "fieldName"      "PlayerStatusHealthValue2"
    "xpos"           "9"
    "ypos"           "17"
    "zpos"           "6"
    "wide"           "32"
    "tall"           "10"
    "visible"        "1"
```

```
        "enabled"           "1"
        "labeltext"         "%Health%"
        "textAlignment"     "center"
        "font"              "hudfontsmallest"
    }
```

The positioning may not work for all three of them or any of them to be honest. That's just the one I happen to use for my HUD. You can change any and all of the values except the labeltext, the fieldname, and the controlname unless you do it right. I'd stay away from it though. There are also a bunch of different elements in those three files but they do not need an explanation. They follow the same rules as the Player Health Cross from HUDPlayerHealth.res so refer to that section if you want to edit the sizing of the different health crosses, their backgrounds, and their overheal effects.

22. THE WIN PANEL

This is pretty much the same style as the scoreboard, except it's much simpler but harder to edit. The reason it is hard to edit is because you can't get this element to display itself on your screen until a team wins a round or the round stalemates. It is controlled by the file winpanel.res in the tf/resource/ui/ folder. There is another half of this file for arena mode called arenavspanel.res but it's not really necessary to edit unless you play arena mode a lot.

Blue/RedScoreBG – These are the two colored background images behind the team name and the score on the top of the panel.

Red/BlueTeamLabel – This is the name of the team which is either BLU/RED or whatever crazy team name you assigned yourself in tournament mode.

Red/BlueTeamScore – This is the numerical value for your team's score at the end of the round.

The rest of the pieces of this HUD follow the same rules as the scoreboard. The only reason I added this to the list of explained entities is because a lot of players seem to like changing the way the win panel looks. This is the file to do it in, and I'd suggest using the console command to end the round multiple times to get it to display after you reload it.

23. EVERYTHING ELSE YOU NEED TO KNOW

There are a few things I didn't go into depth about in the other 19 pages. Damn. I will try to go over a few of the other files briefly just in case you wanted to know some more not necessary HUD elements. Some do require editing for perfectionist purposes but aren't too vital.

To add shadows – Simply copy and paste an element underneath itself adding a 2 to its name and its fieldname. Then change the zpos on the original to 1 more than it currently is, and change the ypos/xpos to 2 more than they were on the copied version. Changing the fgcolor to black for the copied version is also usually necessary.

The spy disguised as element – This has its own file for the small panel telling you who you're disguised as and what weapon you're carrying. The name of this file is DisguiseStatusPanel.res and can help you fix overlaps when you're playing spy.

Always change the map to CTF or PLR – Also change class to heavy/pyro as well as demoman/scout to make sure nothing hits anything else on accident. Engineer pistol is also a main cause for overlapping due to its 200 clip ammo.

If visible 0 doesn't remove something – This happens on a few of the different pieces of the HUD. One fix is to change the width/height to 0, or give the x and y pos a value so high that it doesn't show up on the screen at all.

Make sure you don't have any typos – If one part of the HUD has a typo in the wrong spot, it could cause the game to crash. This happens a lot of times if you try something you're not sure of or if you accidentally hit v and not control+v.

The Demoman Stickies and Bow Charge – These have their own files. They are HUDDemomanPipes and HUDBowCharge if I remember correctly. The positioning is done in hudlayout.res if you want to move them to different parts of your screen.

I think that's all there is you'll need to know about editing the HUD. If I missed something feel free to point it out or let me know so I can fix/update it. 20 pages on HUD editing is not what I'd call interesting but I hope I was able to answer most if not all of the questions as to how they work or what the different elements do/mean.

Cool Essay Nerd.

SPECIAL THANKS

Voltw - For answering all of my questions when I had no idea what on earth I was doing.

MØre - for helping me out as well when I needed help fixing a HUD element or didn't know if something would work or how it worked.

Valve - for making this game and allowing the ability to customize a majority of its elements

The Noid - His HUD made me want to start editing my own.

HoppyPotty - For being Canadian enough to help me find problems with the HUD for resolutions that weren't 16:10 aspect ratio

Seanbud,Mesr, and pMb - for carrying me through ESEA and CEVO

Jestr - For Healing Me

The GotFrag Community - For -fragging all of my posts

And anybody else who gave me HUD feedback and the 1000+ people currently using it.

THANKS A LOT, I HOPE THIS HELPS!

-Flame