

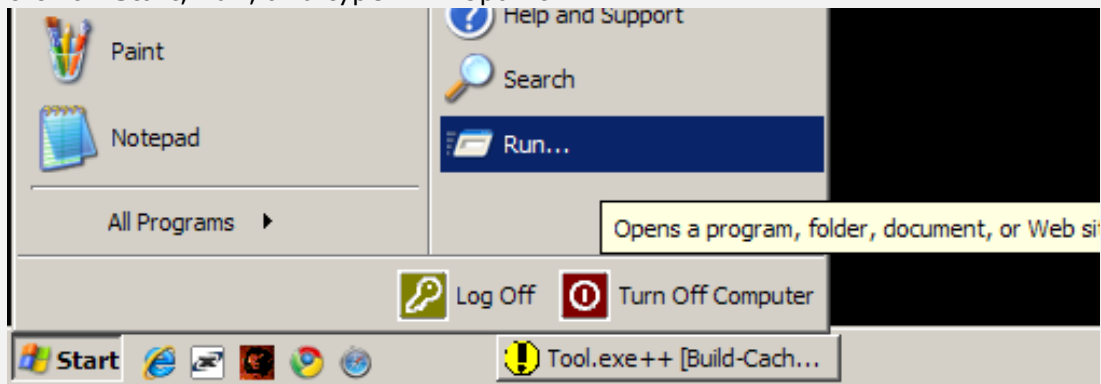
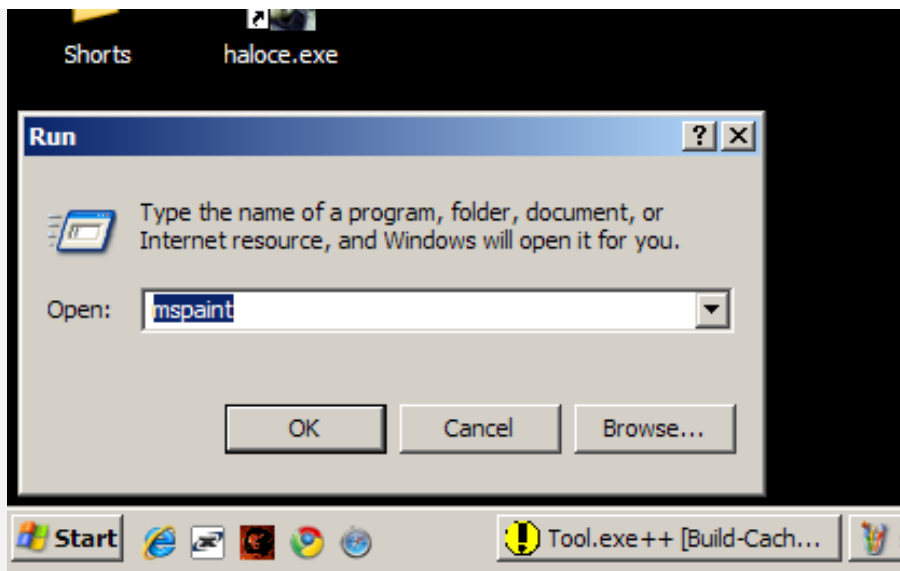


» [Forums Index](#) » [Halo CE Maps \(Bungie/Gearbox\)](#) » [Halo CE General Discussion](#) » [\[56k warning\] How to make a HUD tutorial.](#)

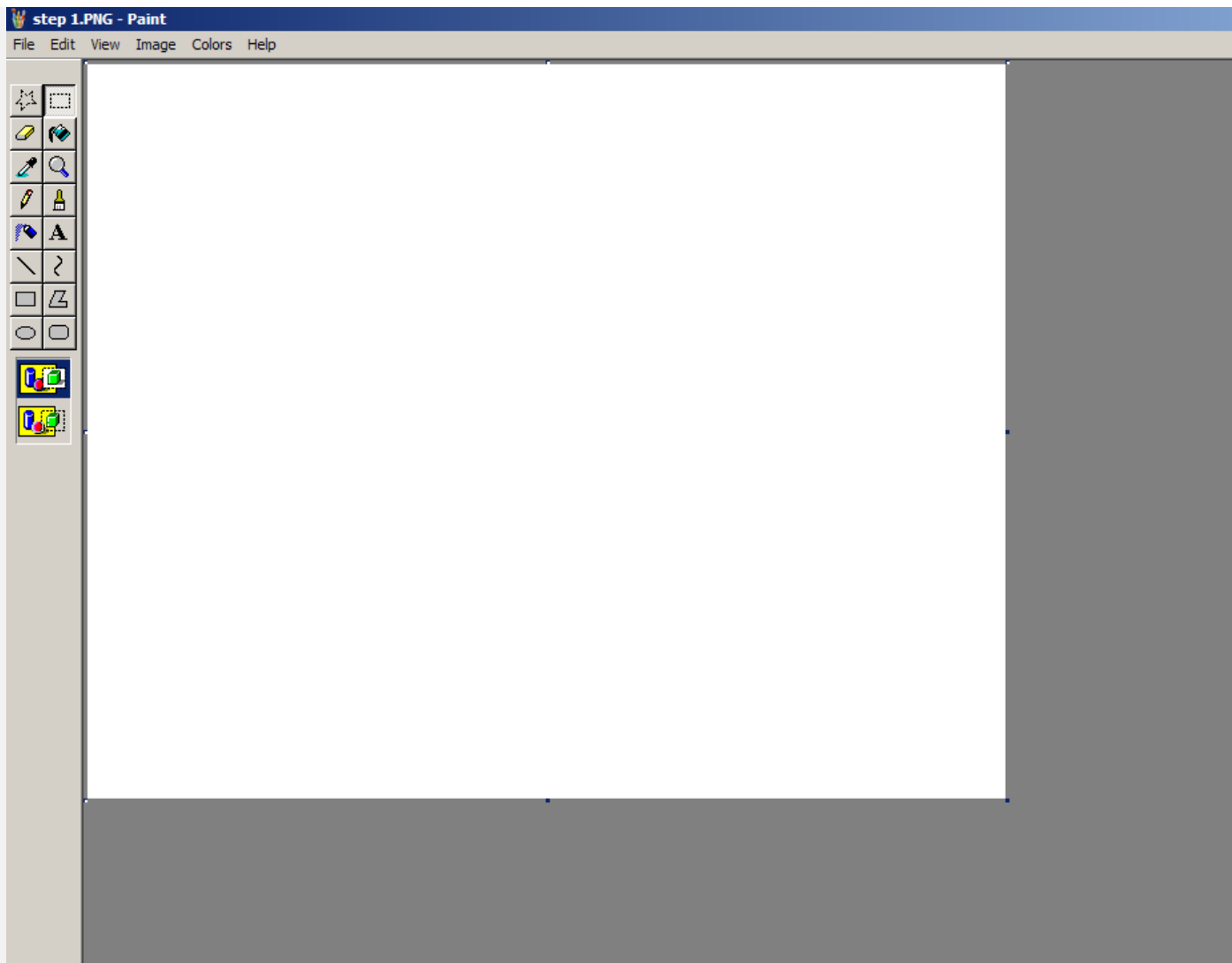
[New Topic](#)

[Post Reply](#)

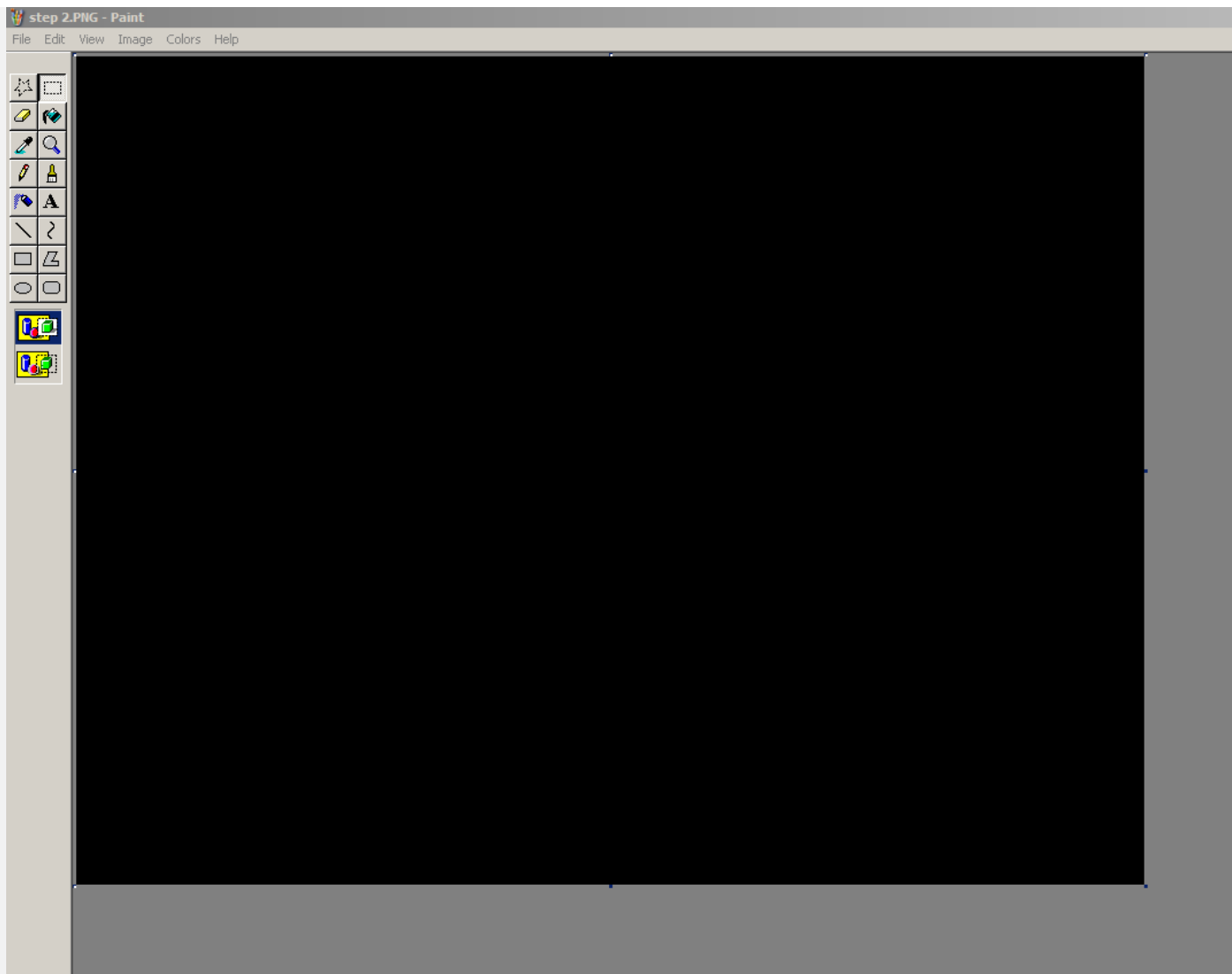
Author	Topic: [56k warning] How to make a HUD tutorial. (25 messages, Page 1 of 1)
Moderators: Dennis	
Jesse Joined: Jan 18, 2009  Don't worry, I'm here.	<p>Posted: Dec 18, 2010 04:44 PM  Msg. 1 of 25</p> <p>Hey guys, It's me Jesse, and I'm starting this thread so that I can bring you a really detailed tutorial that will cover literally everything I know about making HUDs for Halo. First off let me just say that this post is not the last and there will be many more to come. I took LOTS of pictures and I tried to cover as much as possible. This post will only cover the basics on how to make a shield meter(s) and get them in-game and working. A few posts later down, I will show you how to position them, edit sizes, and educate you on the terms that I have learned.</p> <p>First off, you can't just start using photoshop. Well you can but I myself am not pro enough to start on it so I use Paint. Yes mspaint. It really helps me when I need it</p> <p>To start your shield meter, you must first open paint. click on Start, Run, and type in "mspaint".</p> 



Click OK and you should be getting into the paint UI and get something like this:

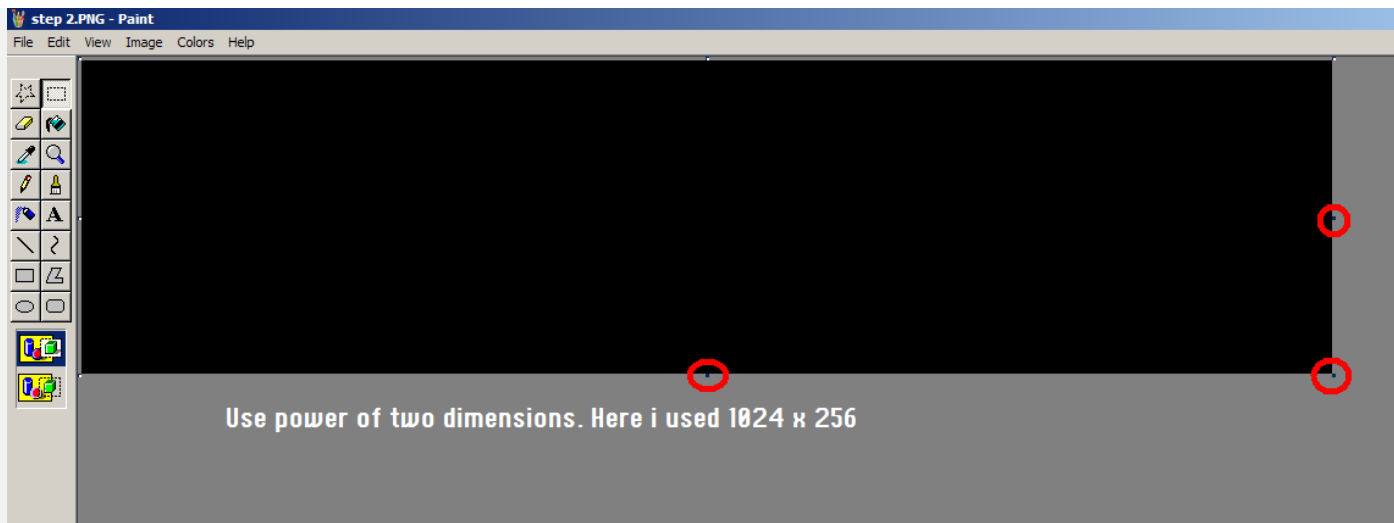


Next, make everything black like so:

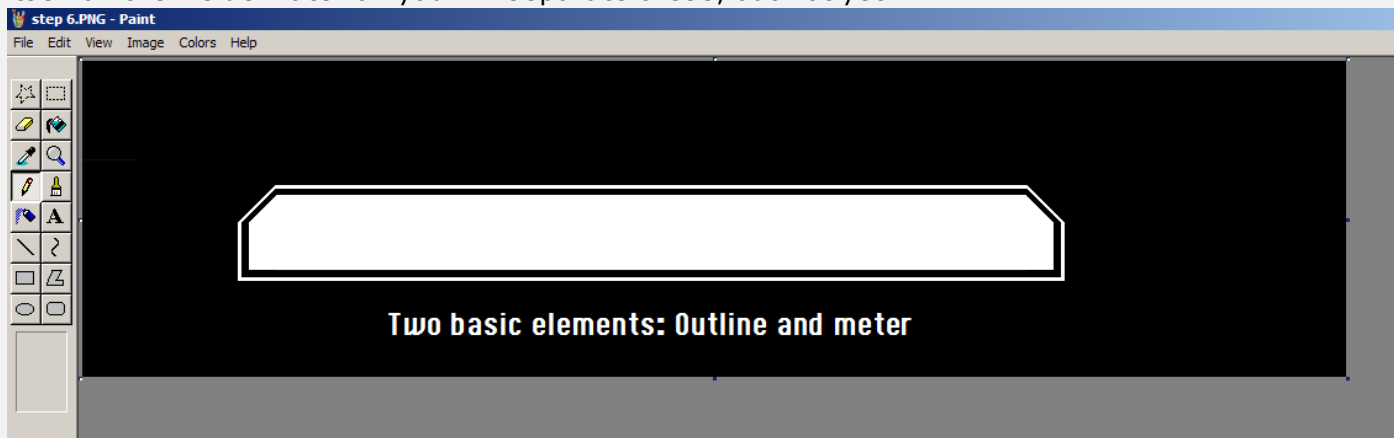


Next, make the image 1024 wide, and 256 high.

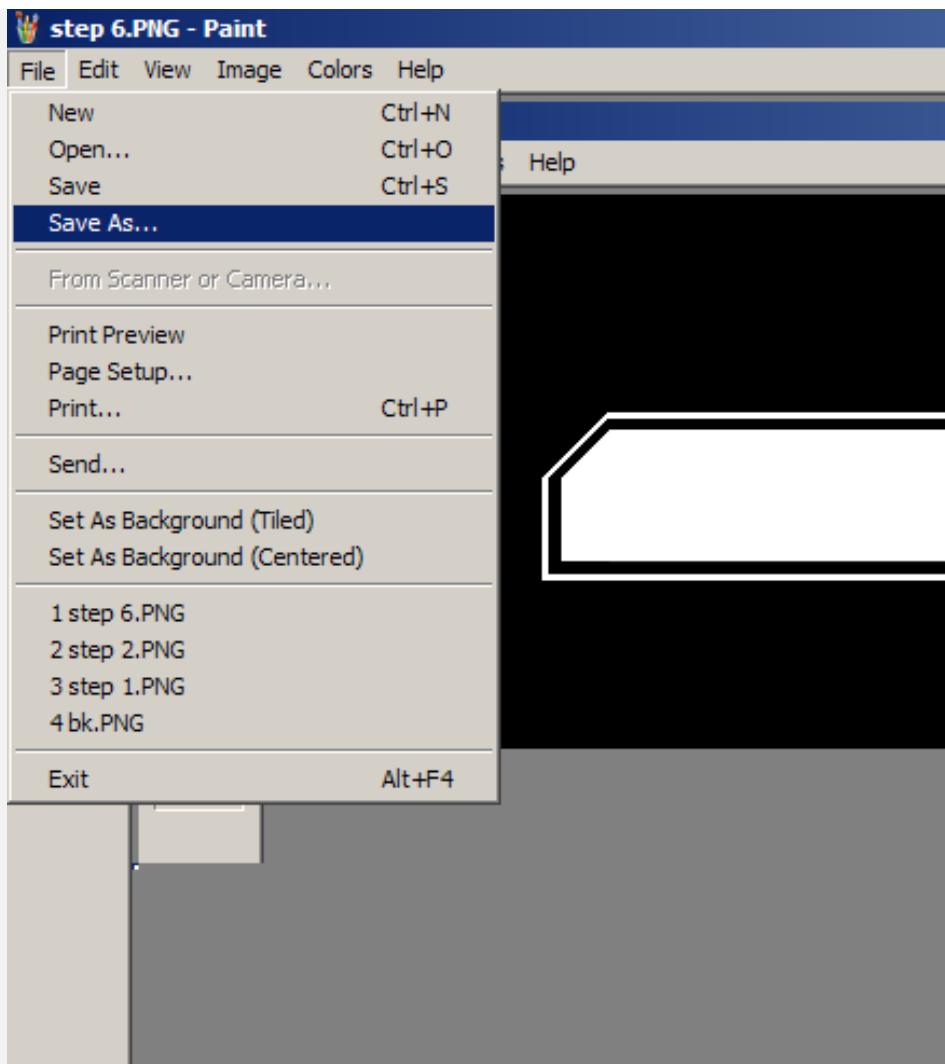
Click on either of the three blue squares to drag that edge in one of two directions (full rotation for the diagonal one.)



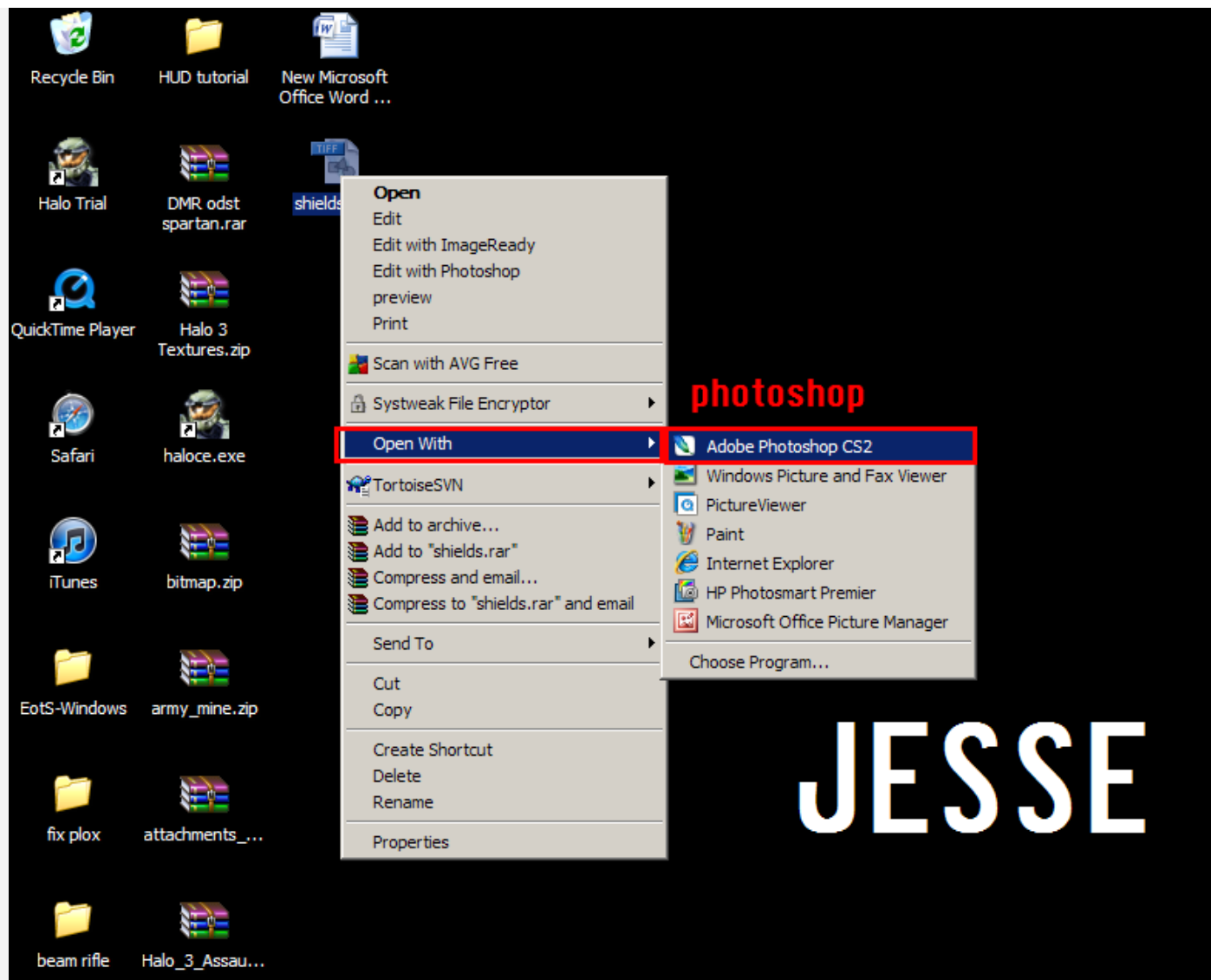
For this tutorial, I already have a basic shield meter and outline already drawn. it's really easy to make this and you can just copy it off of one of these images. Here I have the shield outline, and the meter itself on the inside. Later on you will separate these, but not yet.



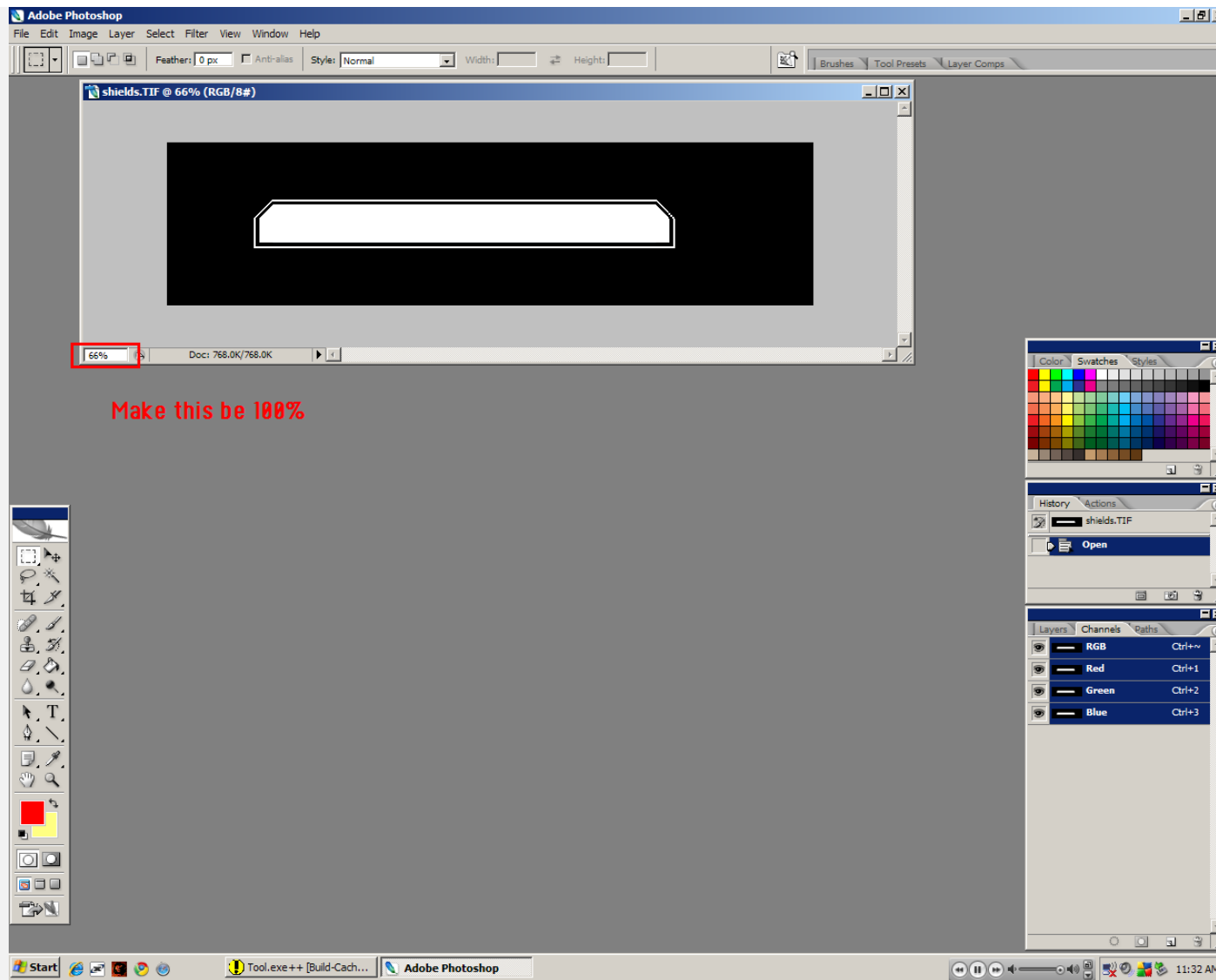
Now if you have drawn your own, or pasted mine, just save it to your desktop. Make sure to save it as a TIF file, or else tool won't be able to compile it, and TIF files don't lose quality.



next, you want to open your saved image with photoshop. Here I used photoshop CS2. Yours may be a bit different, but it's usually the same.

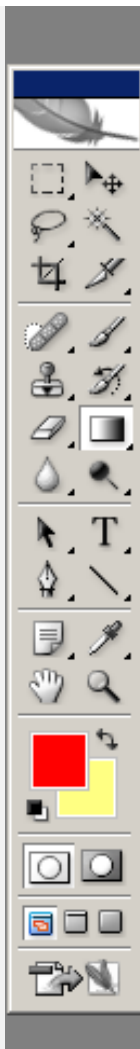


Once you open the image, your UI should look somewhat similar to this:

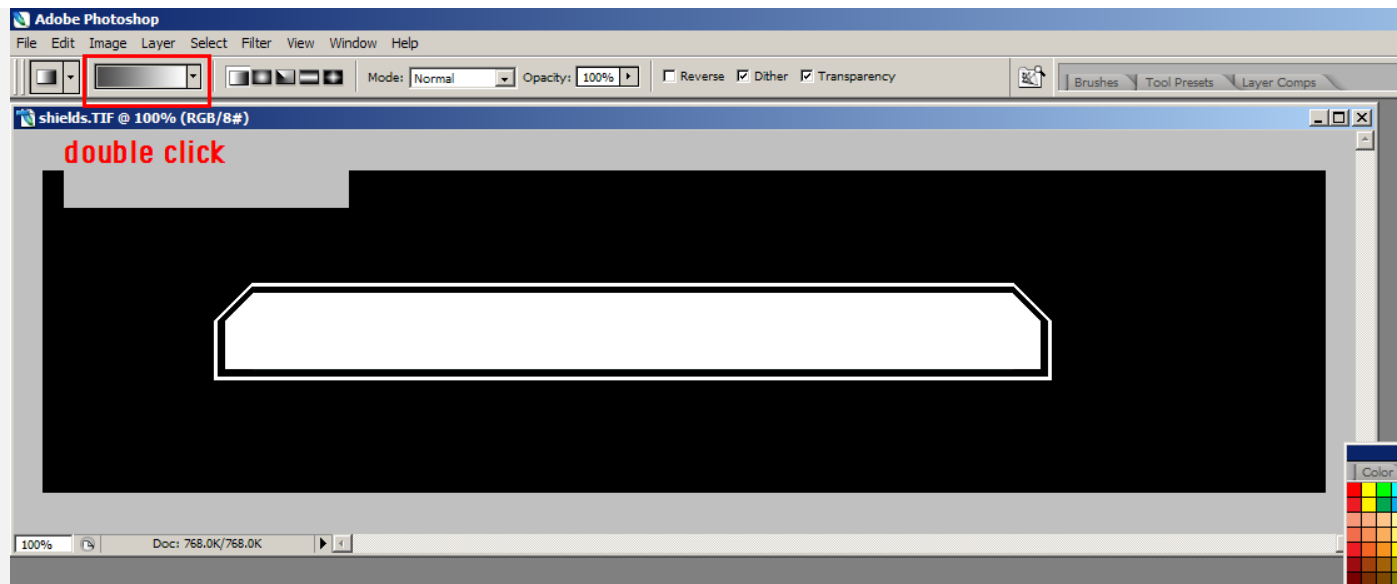


make sure to set the image size to 100% It doesn't edit the actual bitmap size, just the preview size.

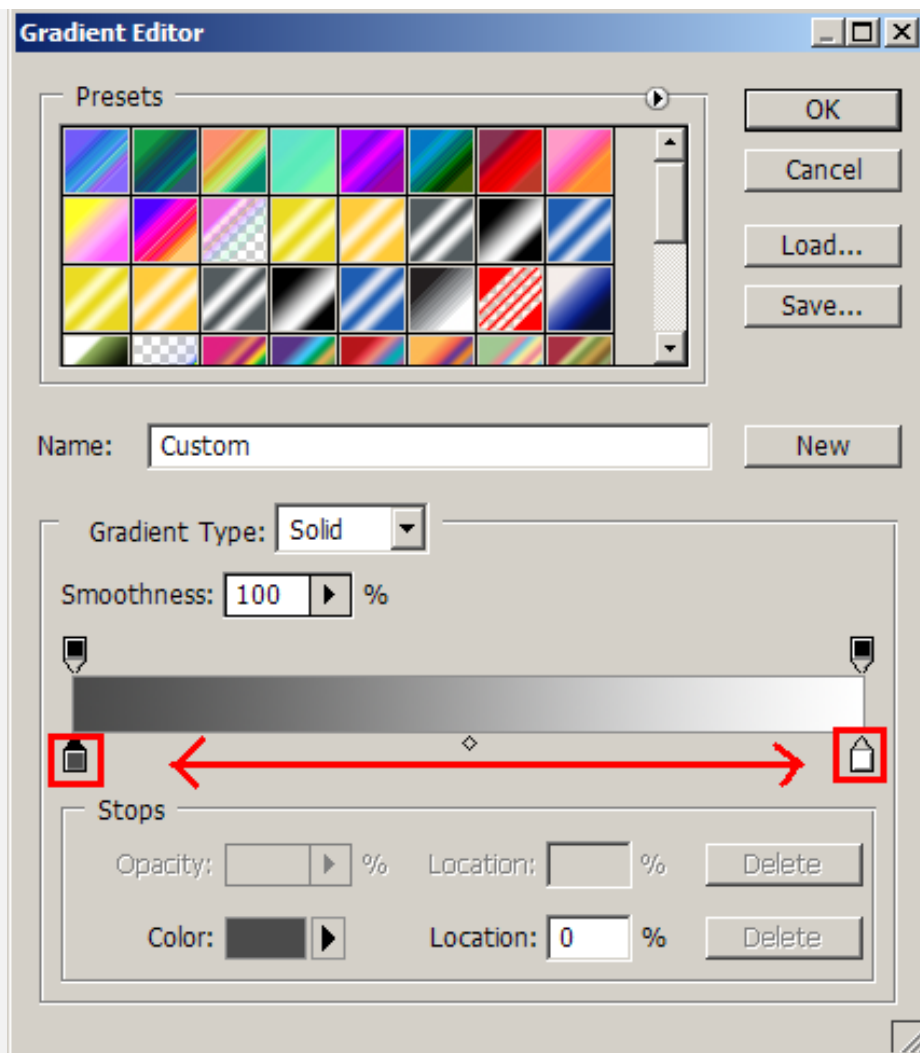
To make a gradient you select the paint bucket tool, then "drag" it sideways to the right and it will show you two subheadings. One for the bucket, and one for the gradient tool. Here we need the gradient tool



Once you select the gradient tool, this option on the top will be there.

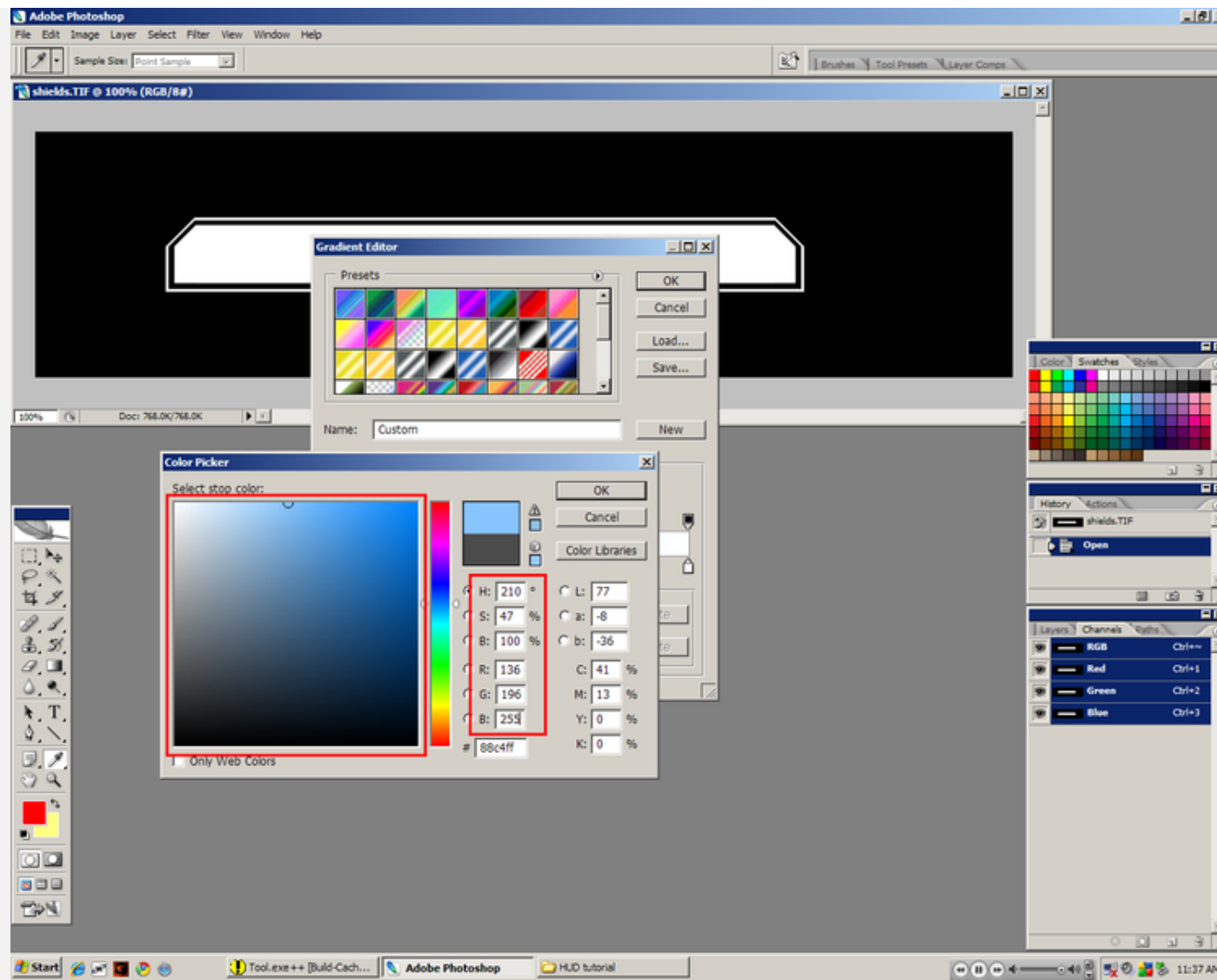


Click on it and then this menu should come up.



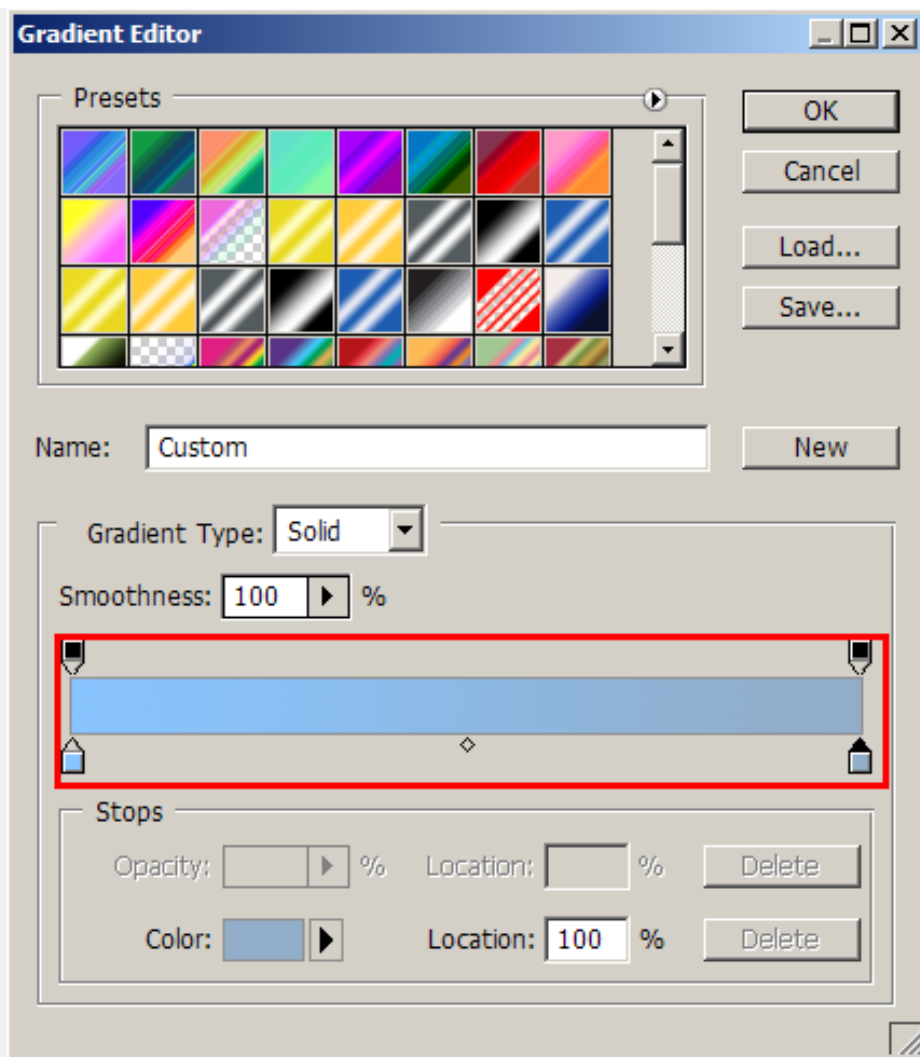
Click on either of the two color tabs to edit what colors the gradient uses.

Once you click on one, a menu such as this should appear.



Choose and edit the colors that you want the gradient to contain.

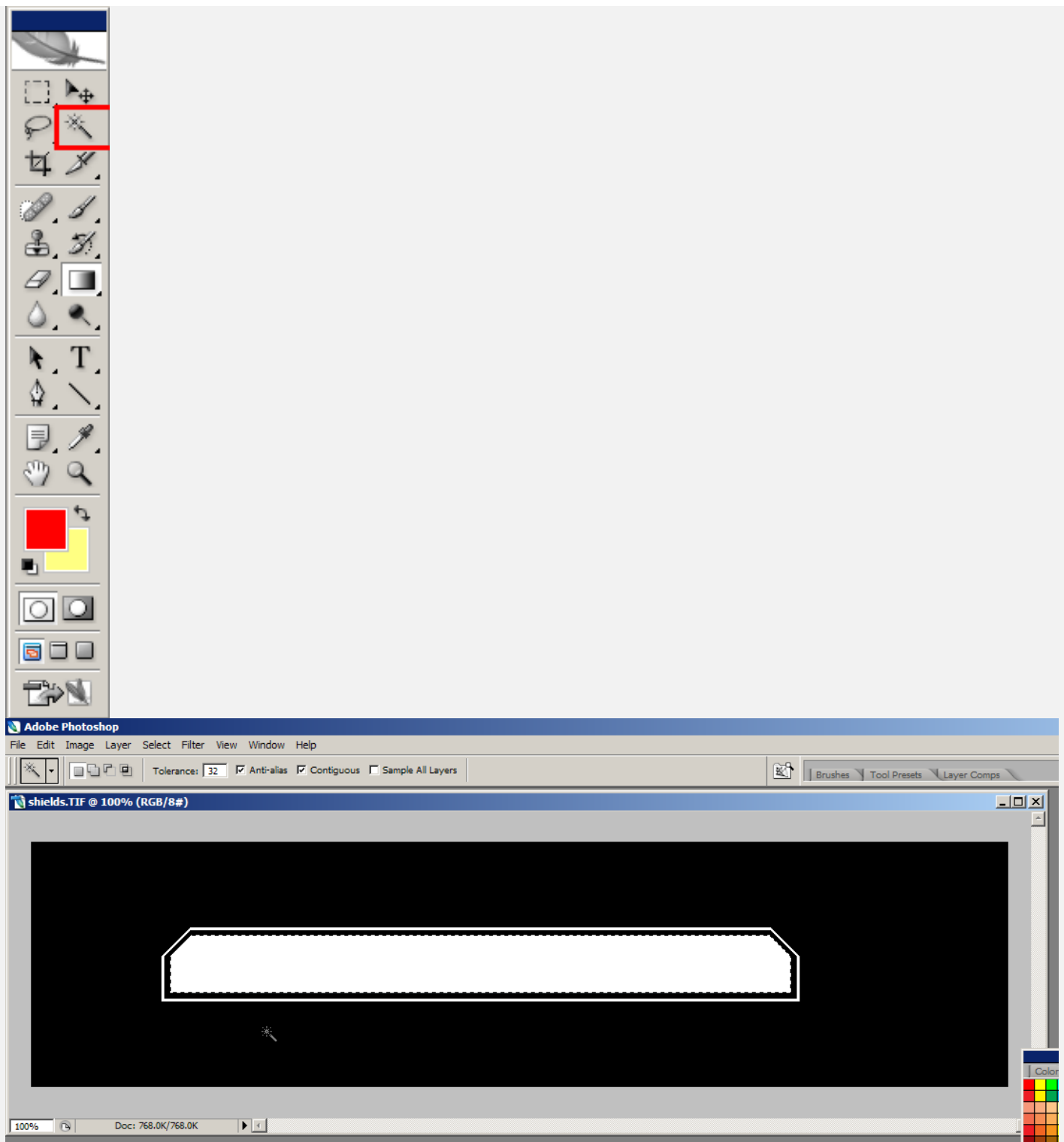
When you finish, it should look like this depending on your colors:



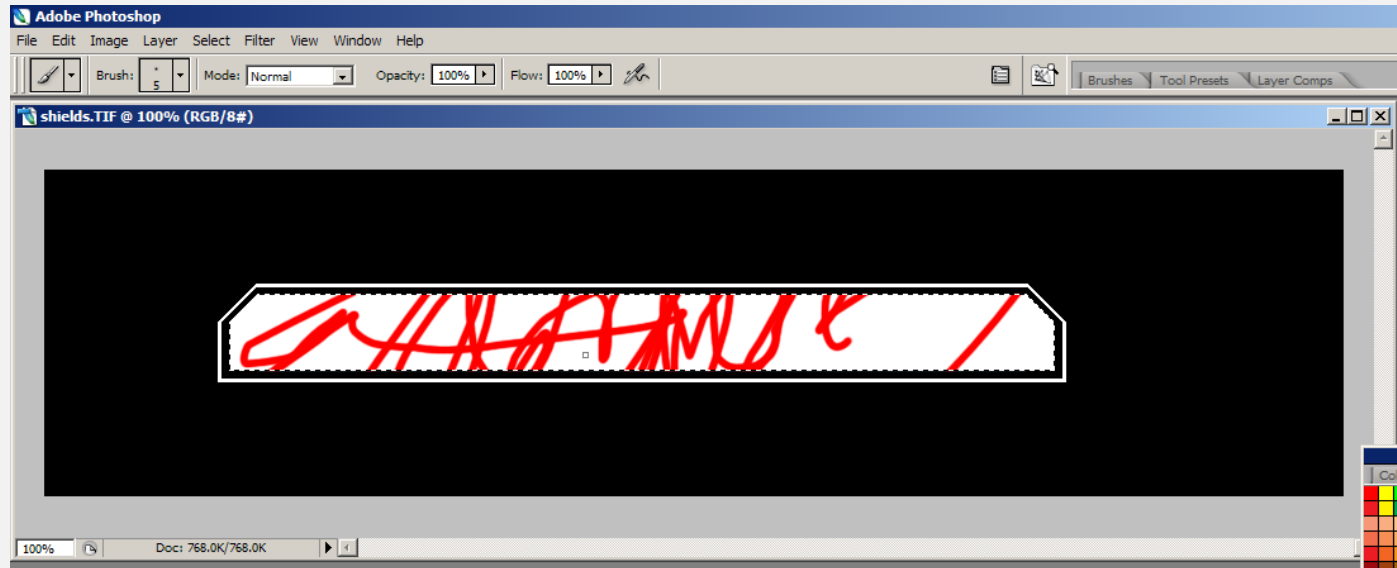
Now, head over to the panel on the left and you will see the magic wand tool



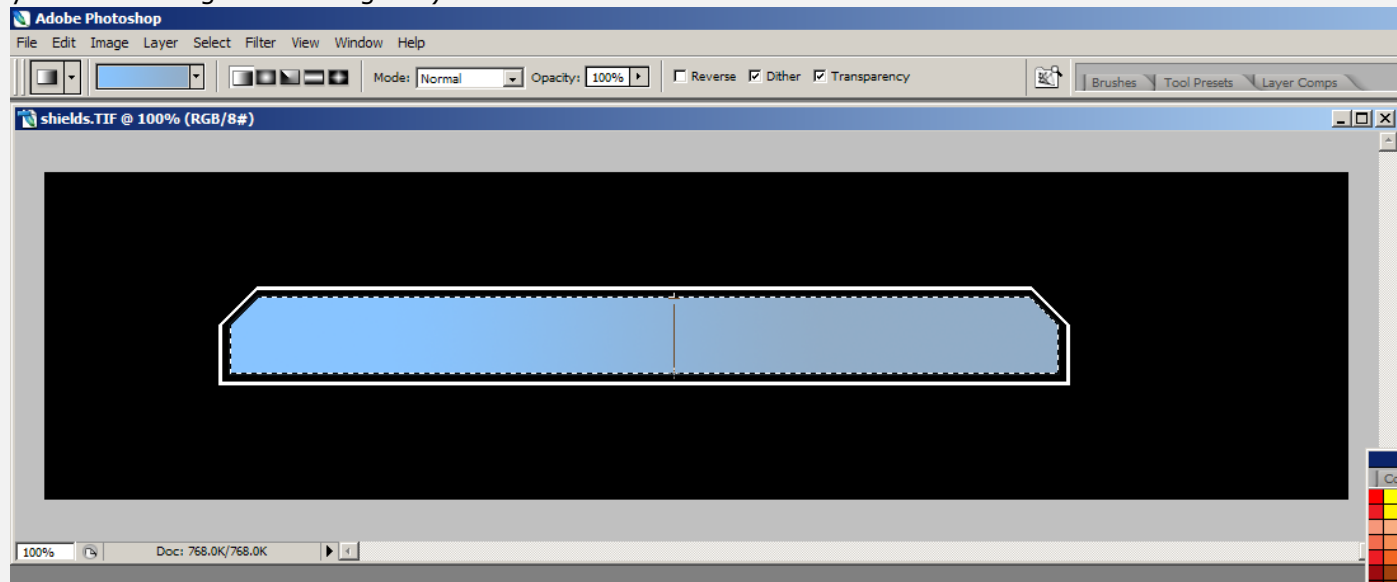
Select that one and click on the middle of the meter part of the image. This tool will attempt to guess what edges you want selected and 70 percent of the time it gets it right.



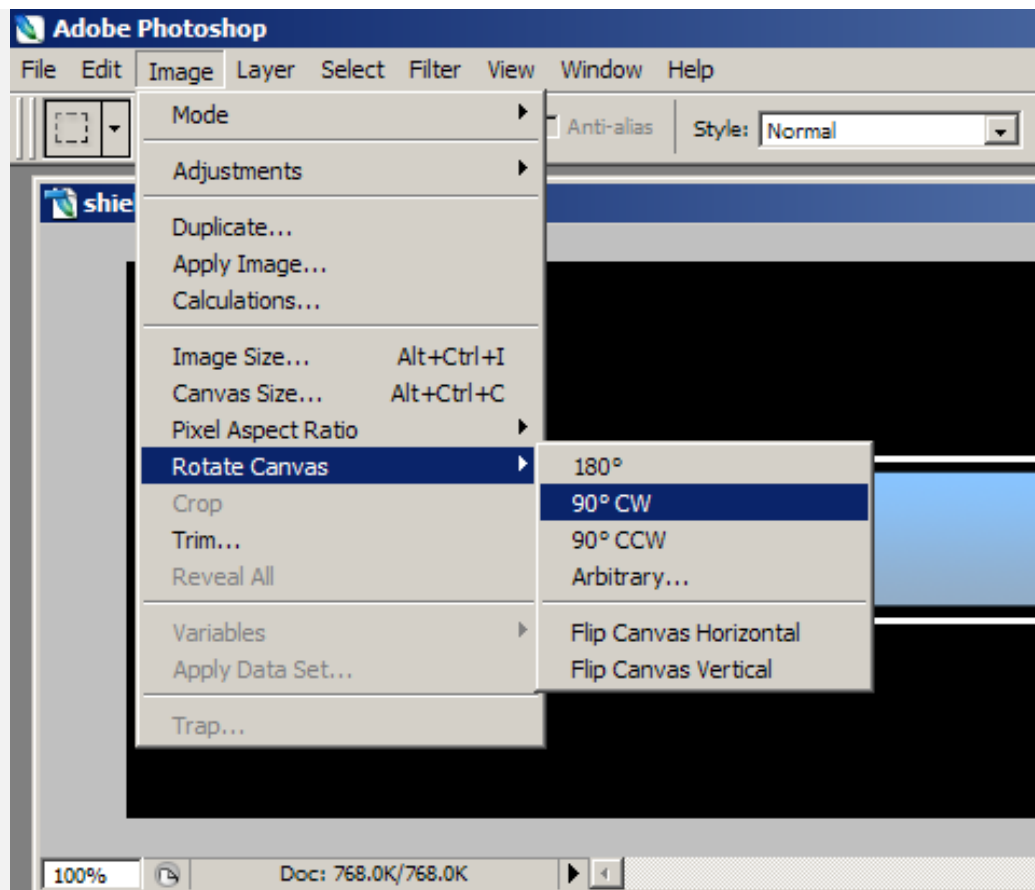
You can see that once you have made the selection, every change you make is contained within the borders.



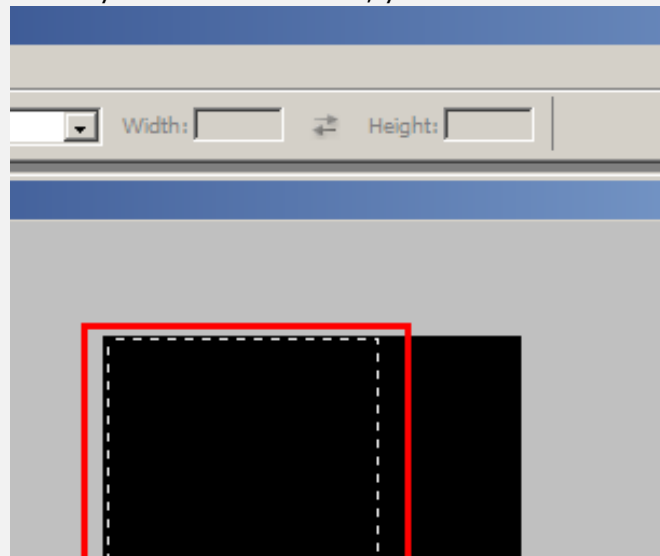
Now, select the gradient tool once more, and line it up from top down on the meter (or any direction you want the gradient to go to)

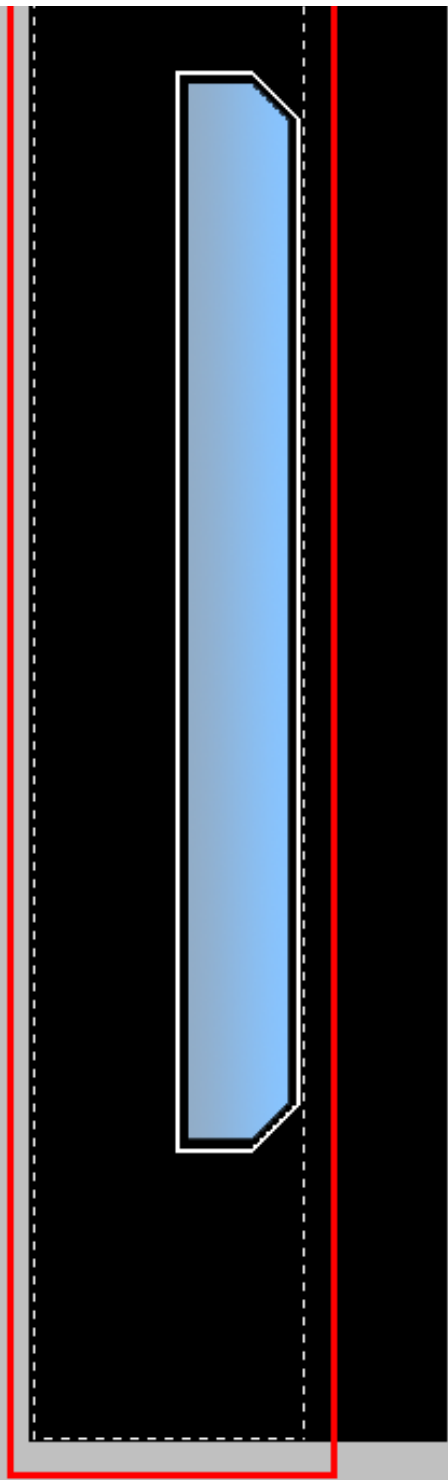


To bend the meter, rotate your canvas by 90 degrees in your preferred direction

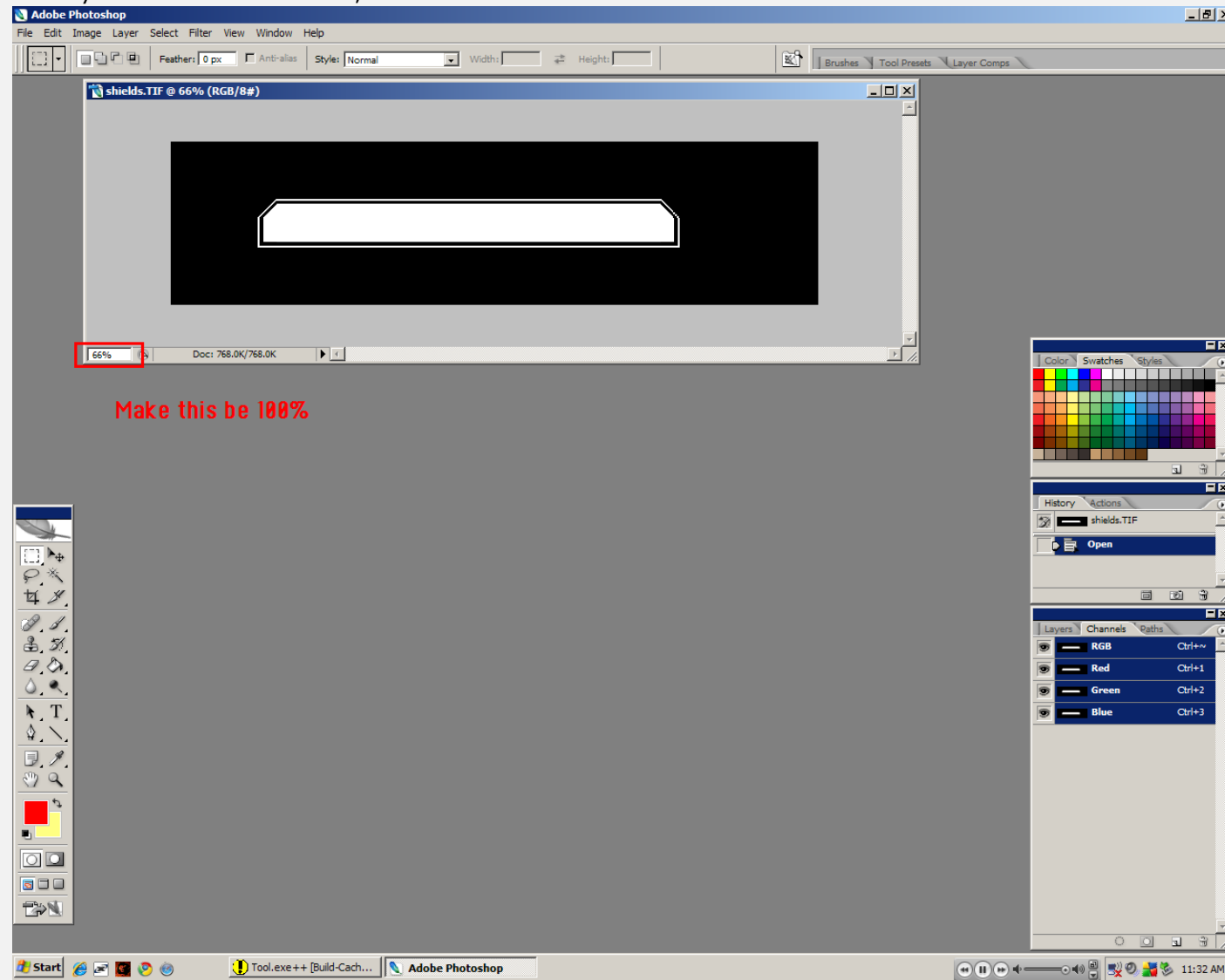


When you have done that, you select the meter such as i did.

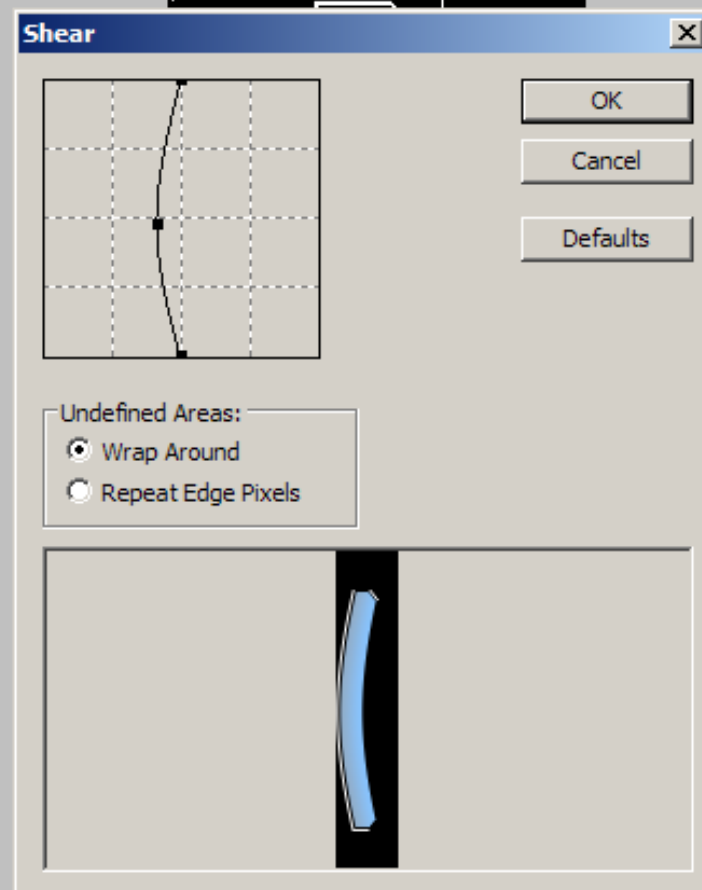




Once you selected the meter, head over to the menu bar and choose filter > Distort > Shear.

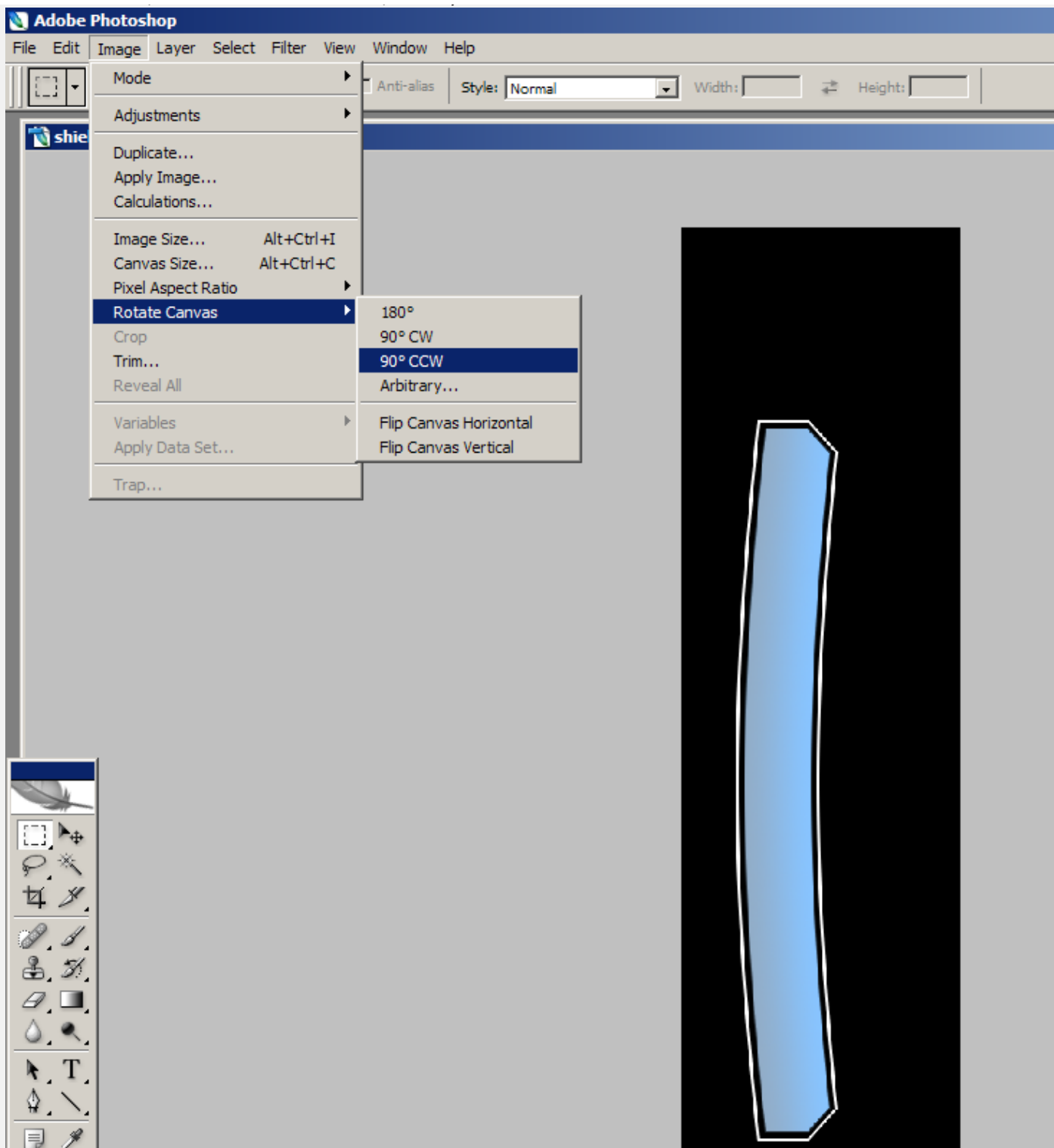


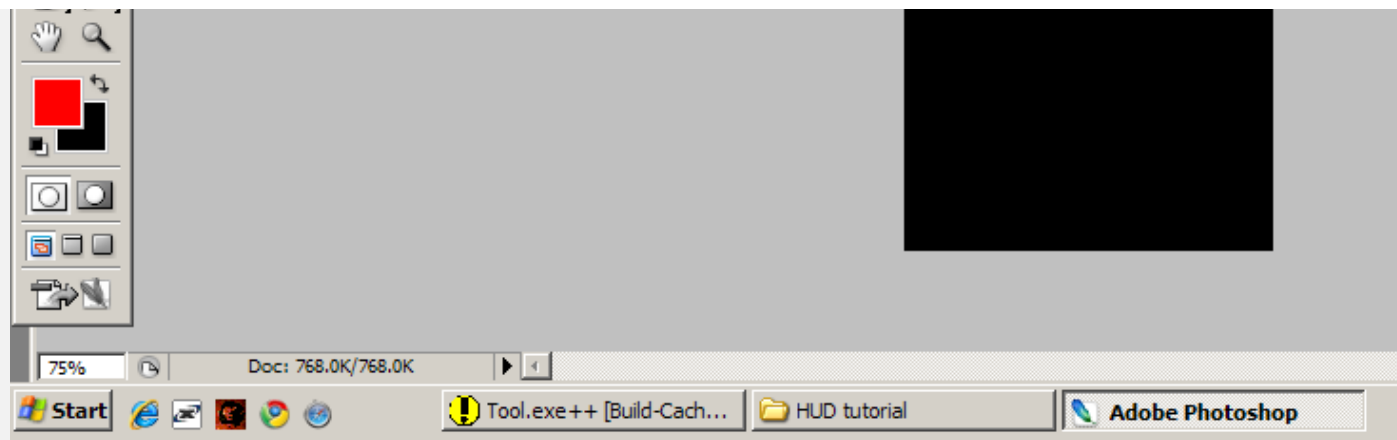
Then you should get a small menu like this.



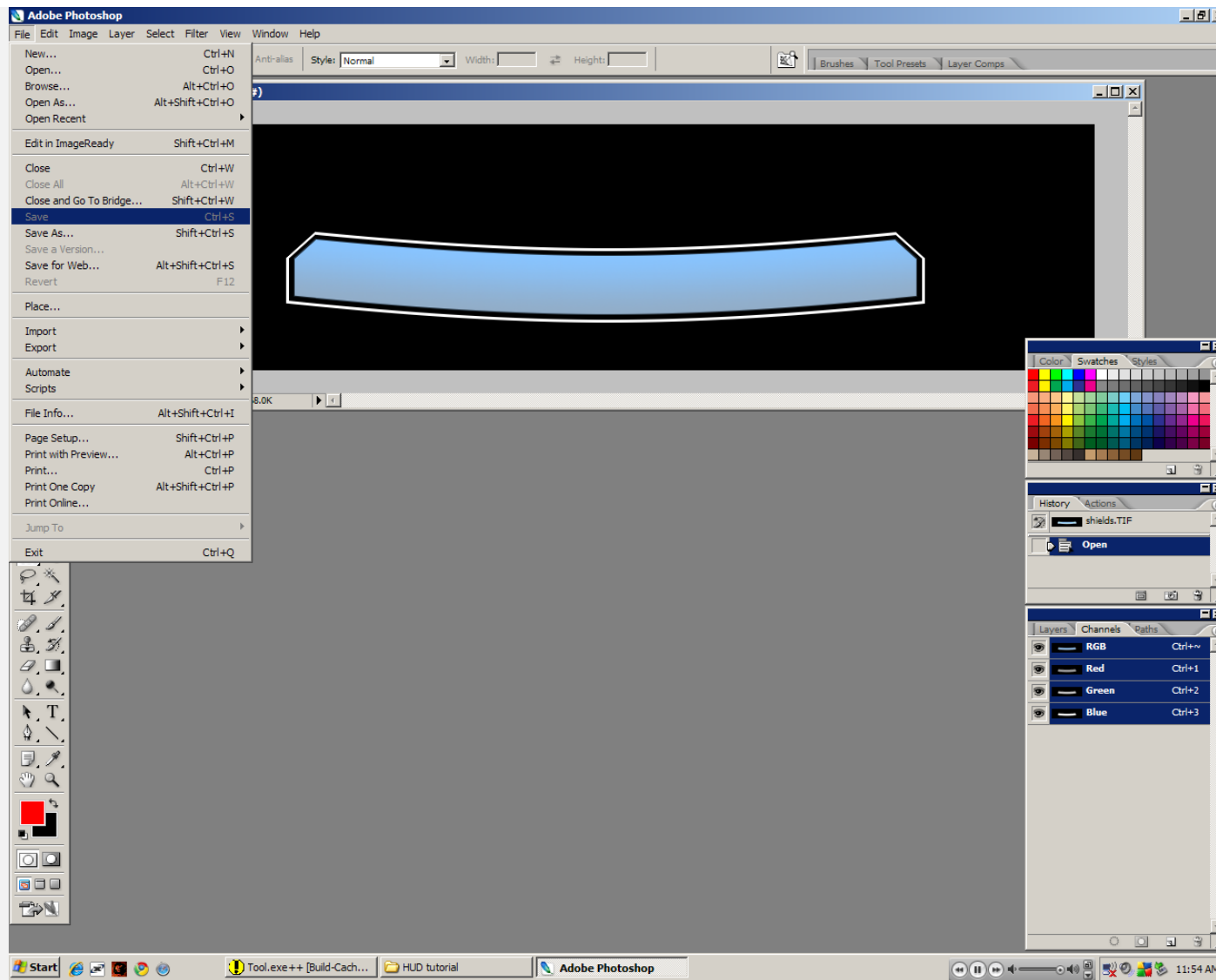
Drag the middle point (or however you wish to bend it) and make it bend how you want your meter to bend. I went for a halo 3 style with this one so I only bent it from the middle.

Once it's bent, it should look like this, then you can rotate the canvas back to how it was.

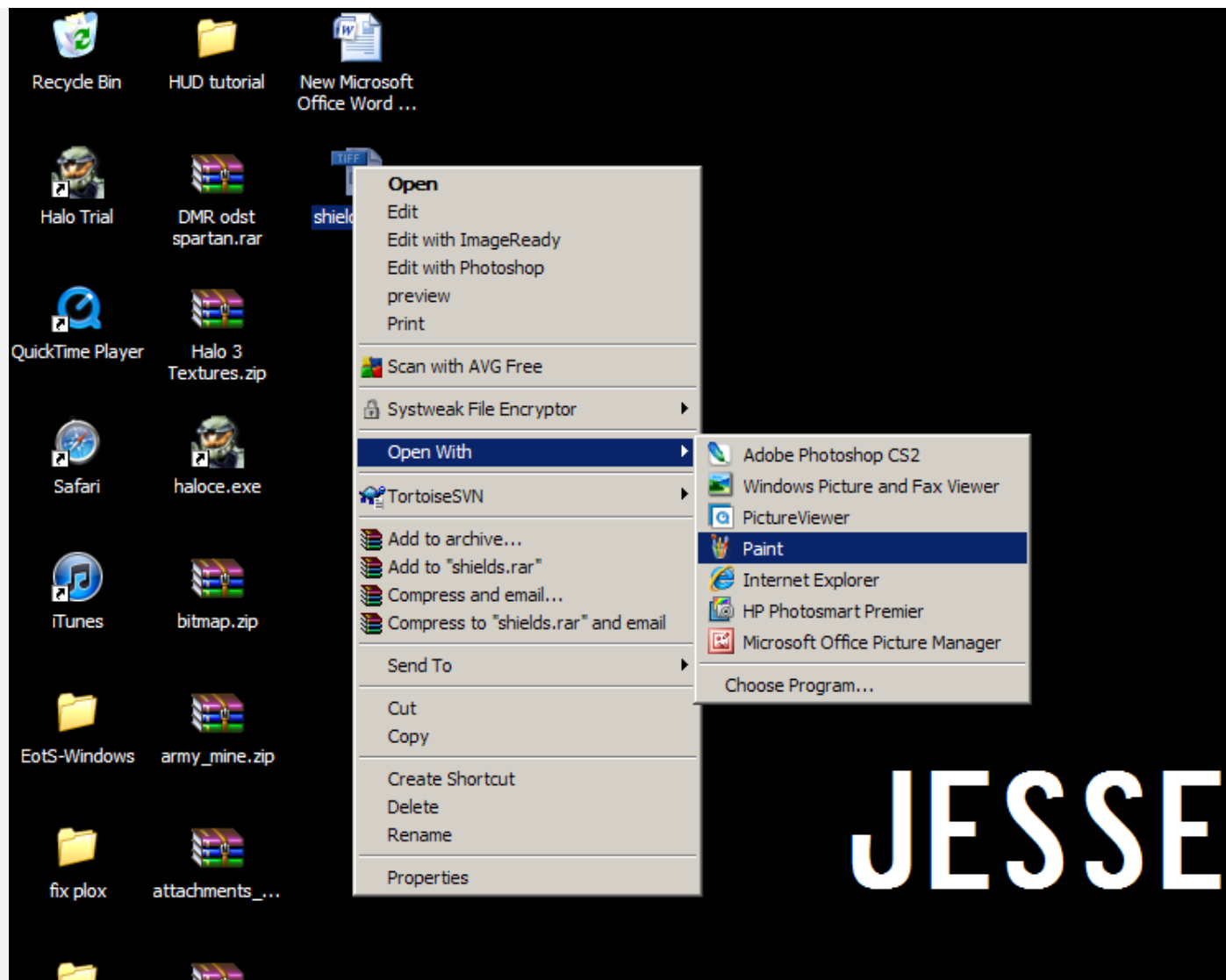




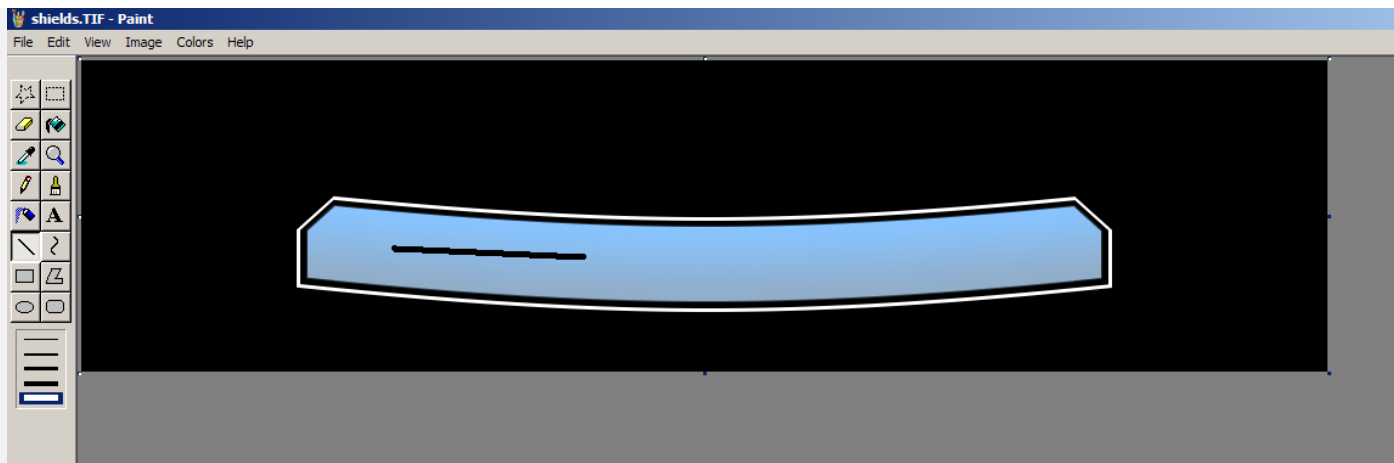
Remember to save!



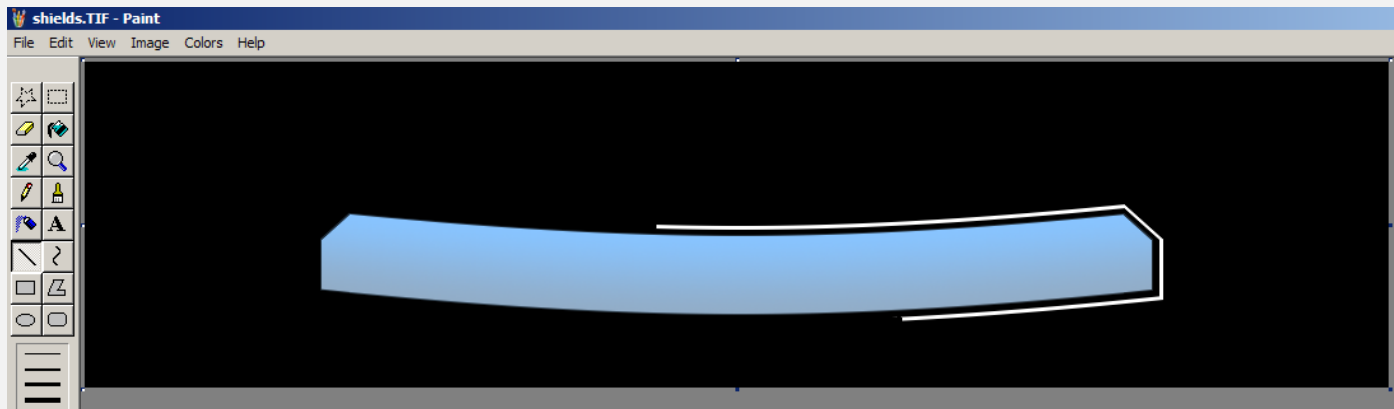
Here we bring back our good old friend paint use paint to open the file you just saved.



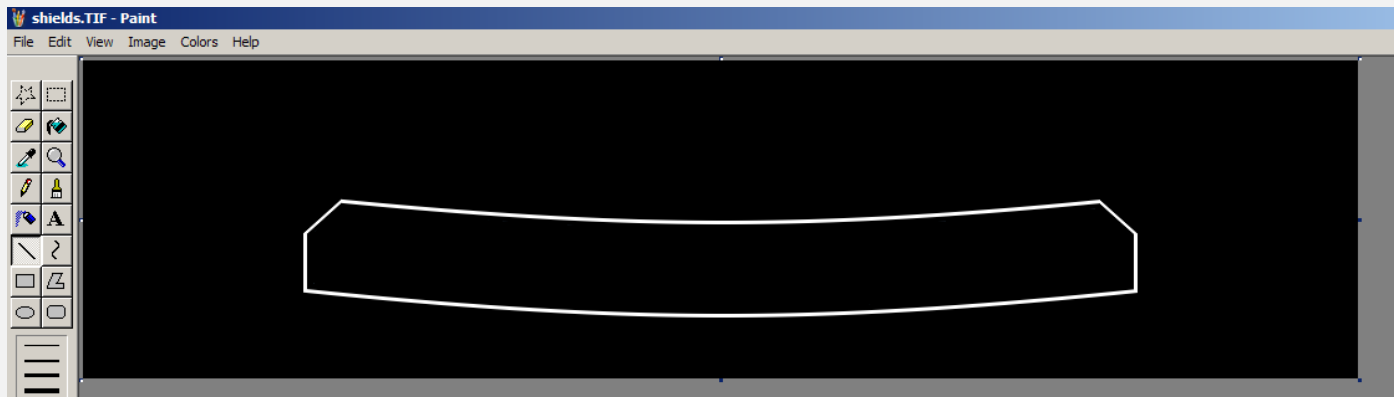
Here you're going to use a thick black line to get separate the outline from the meter.



Use the black line to draw over the outline.

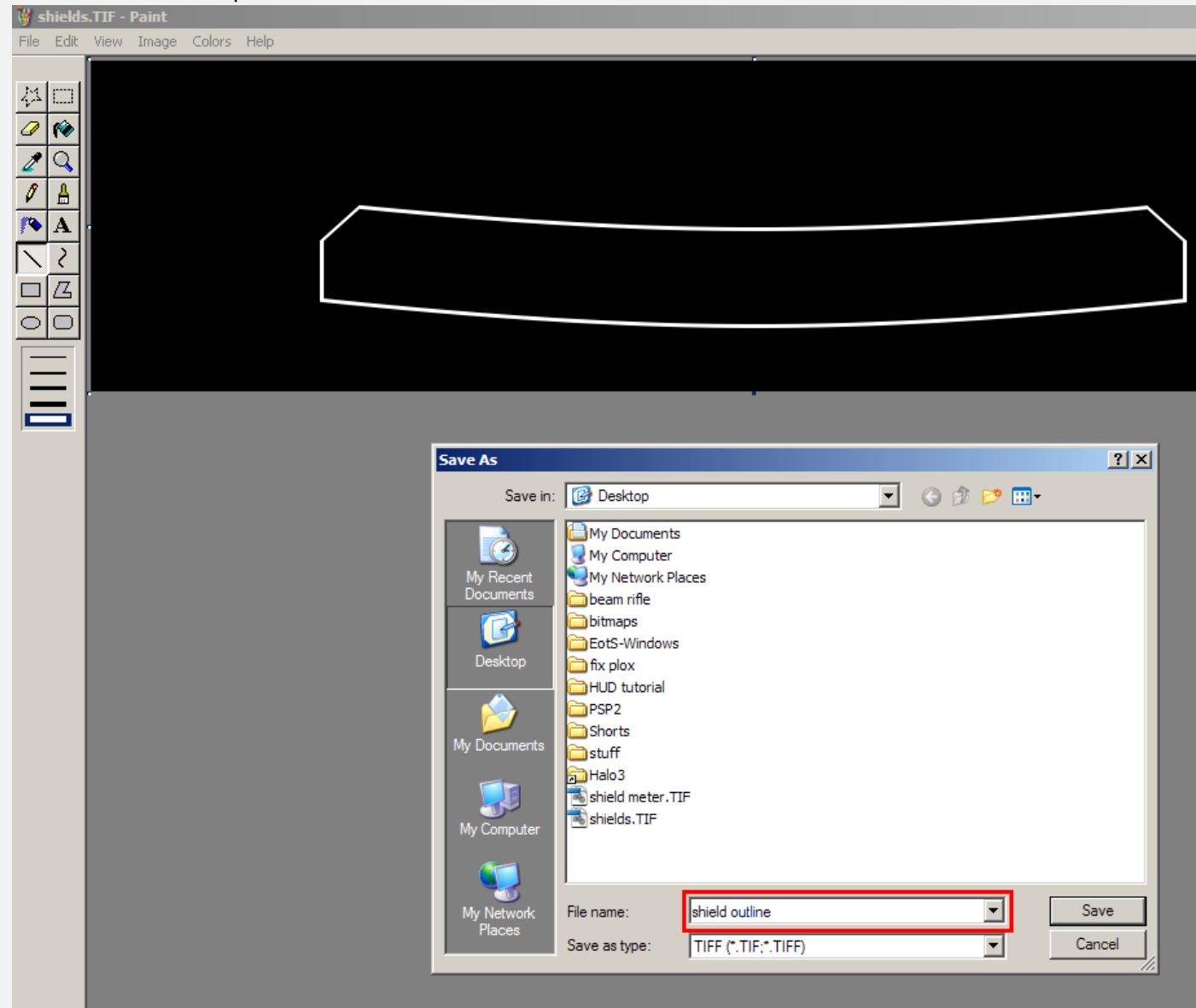


Then do the same with the meter inside.

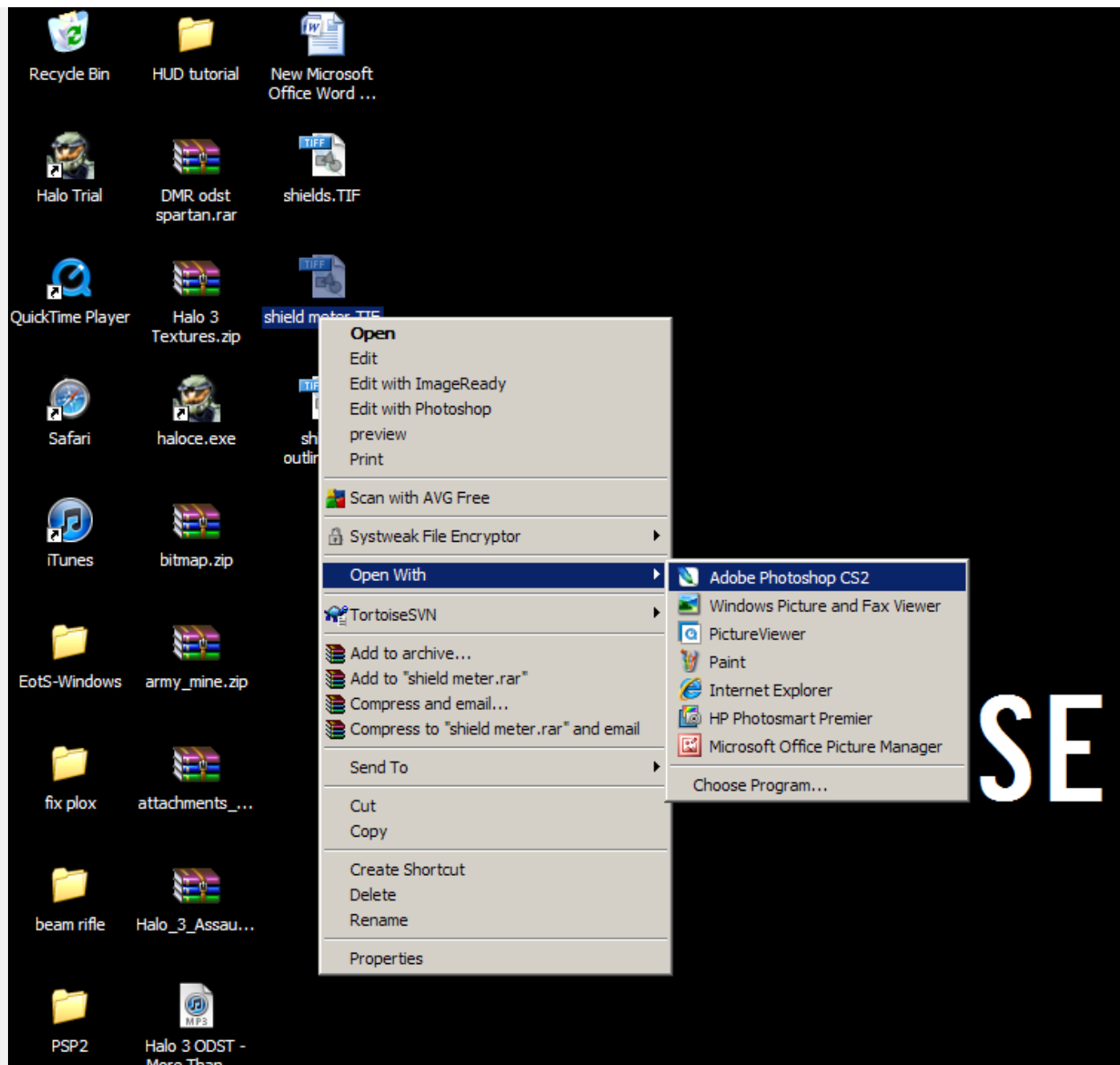


For each element, remember to SAVE AS not overwrite your old one.

Give them their respective names as such:

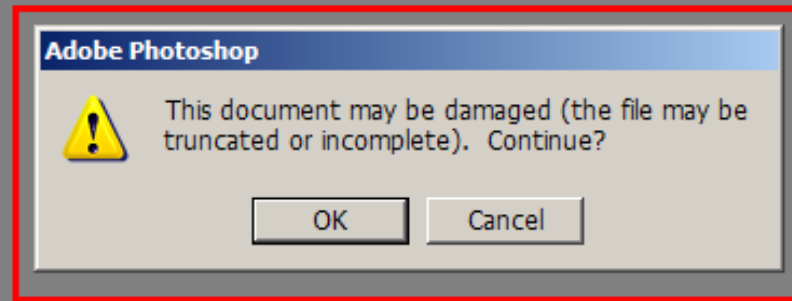


when you have them saved on your desktop, open them up with photoshop

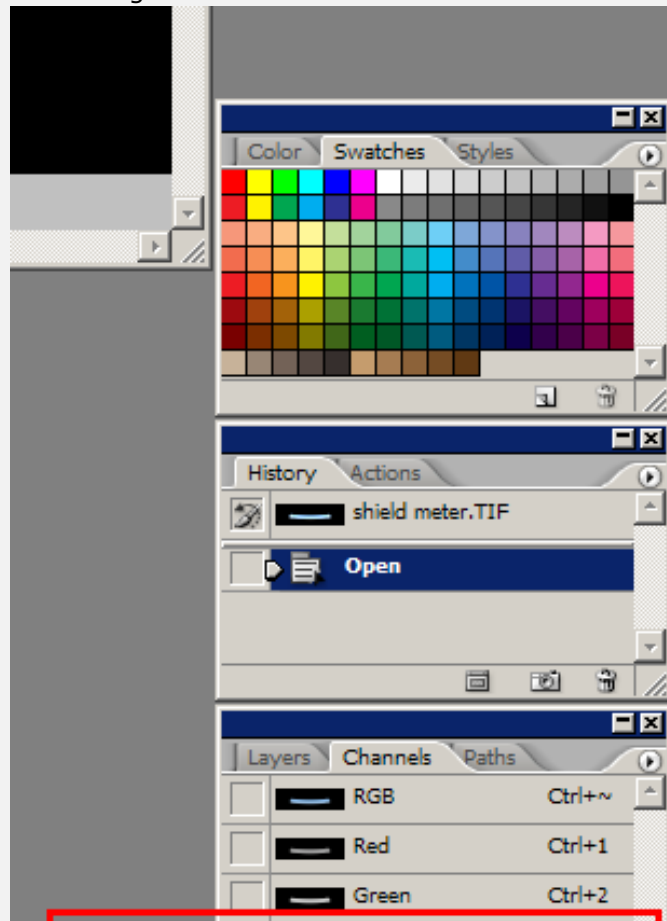


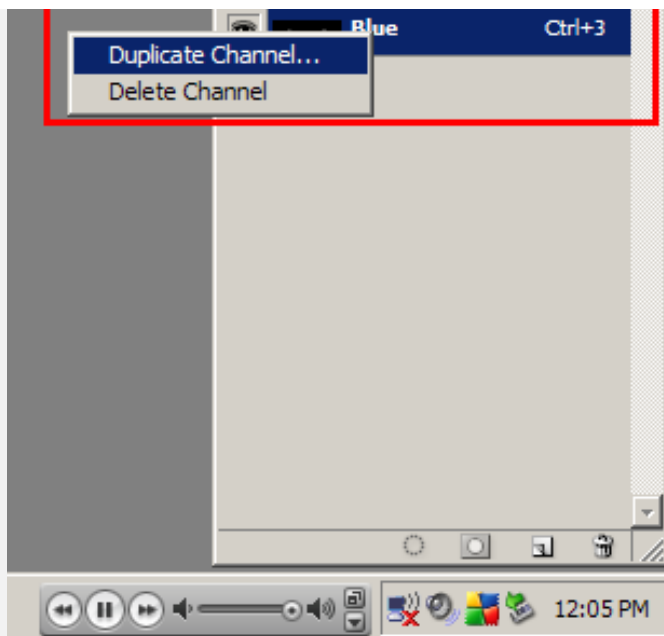
SE

If photoshop gives you this message, just ignore it. it's like tool when it complains.

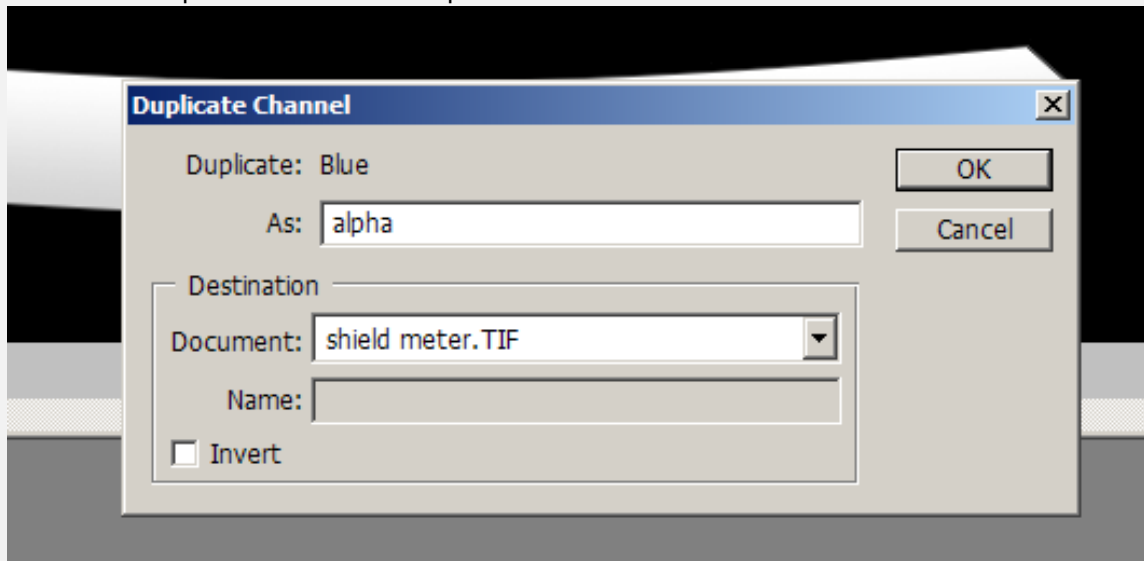


When it is open, head on over to the channels tab and click on the blue\green\red channel. whichever is the brightest or the best candidate to be in the alpha channel. make sure to duplicate it.

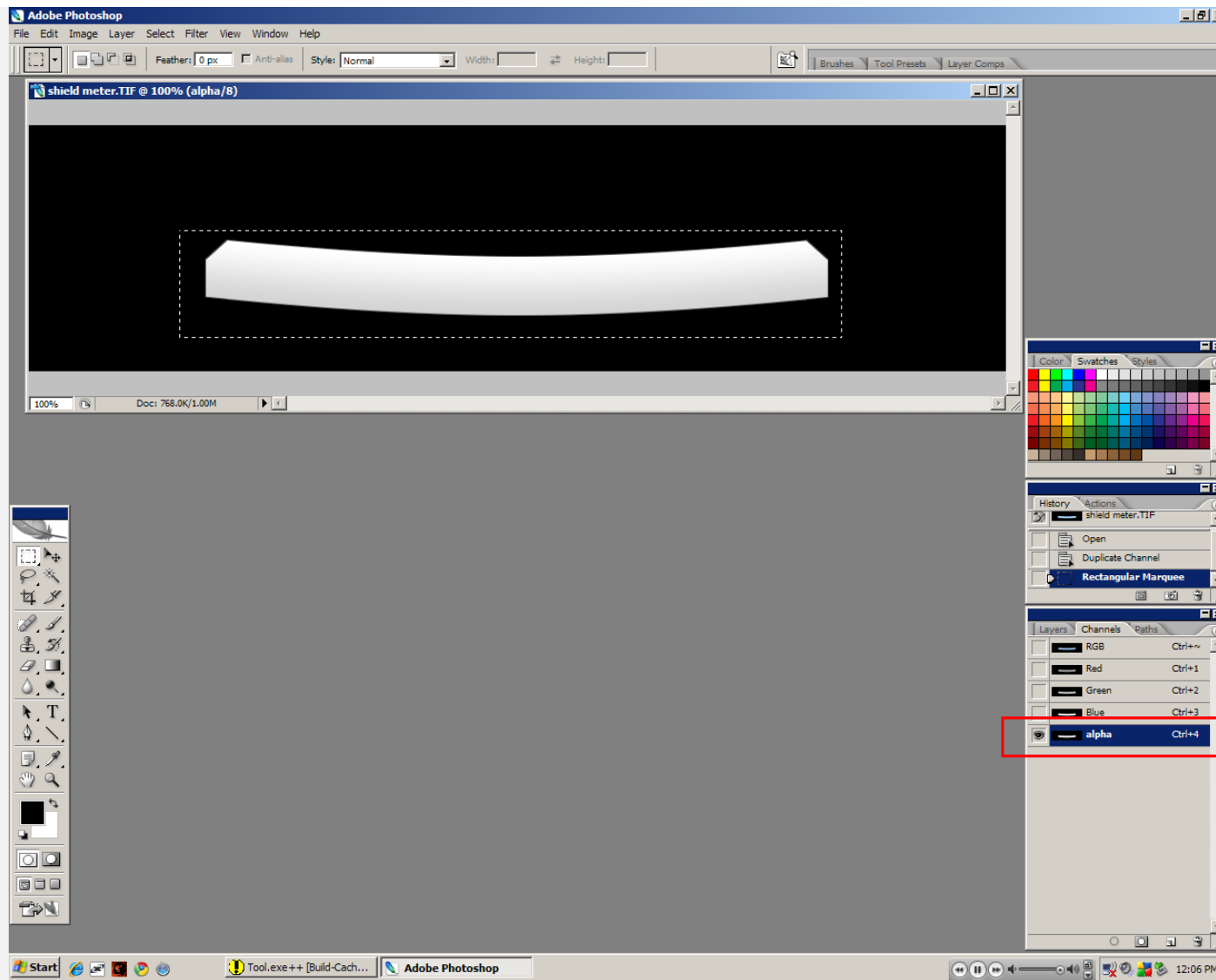




Name the duplicated channel "alpha"



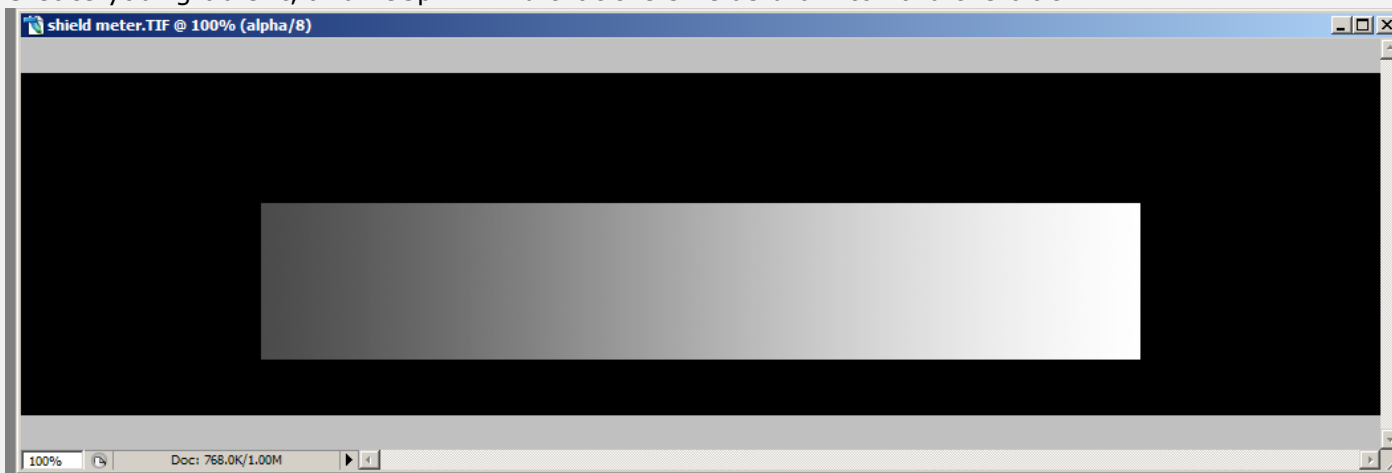
Now you see the alpha channel here, which is a copy of your selected channel



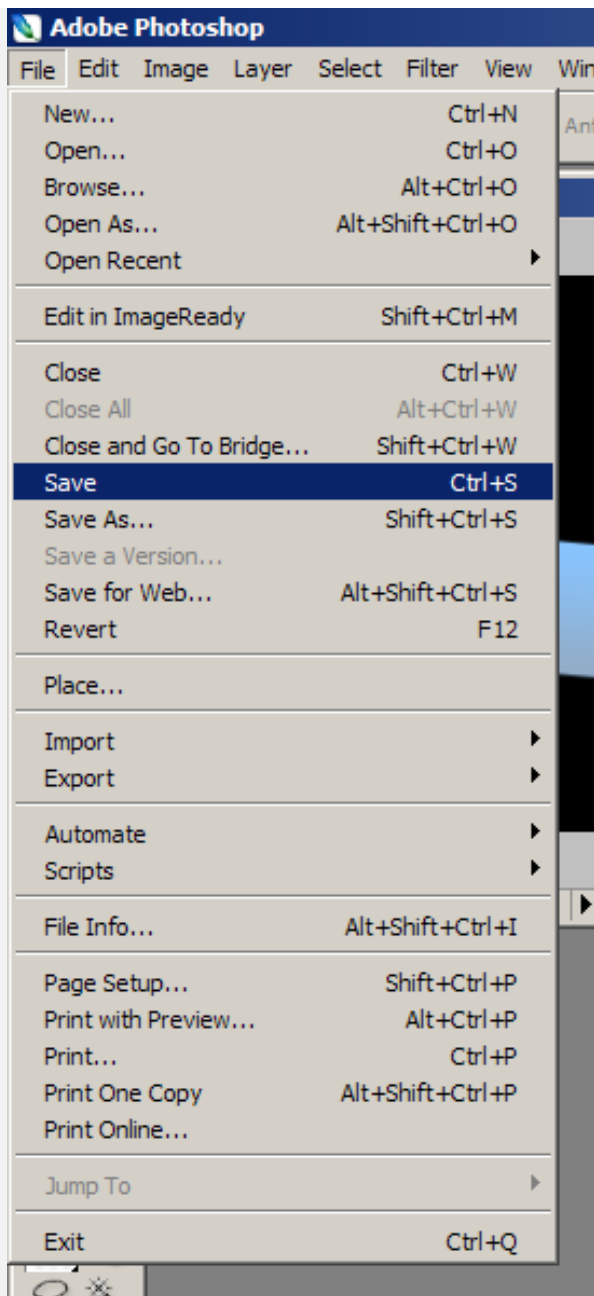
Now select the area around the shield meter, make sure to leave at least 3 pixels of space between the meter and the borders.



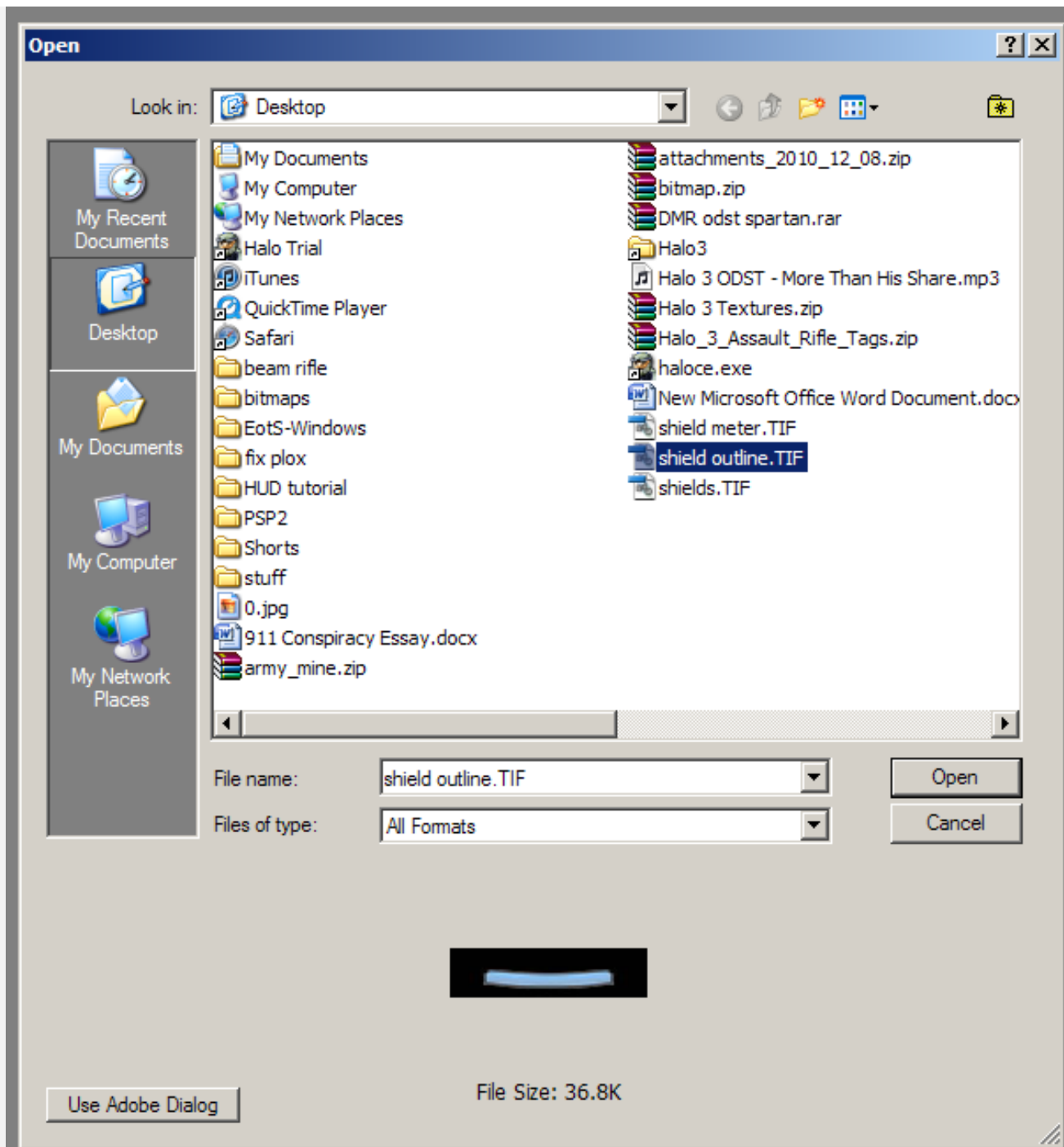
Create your gradient, and keep in mind that the shields drain toward the black.



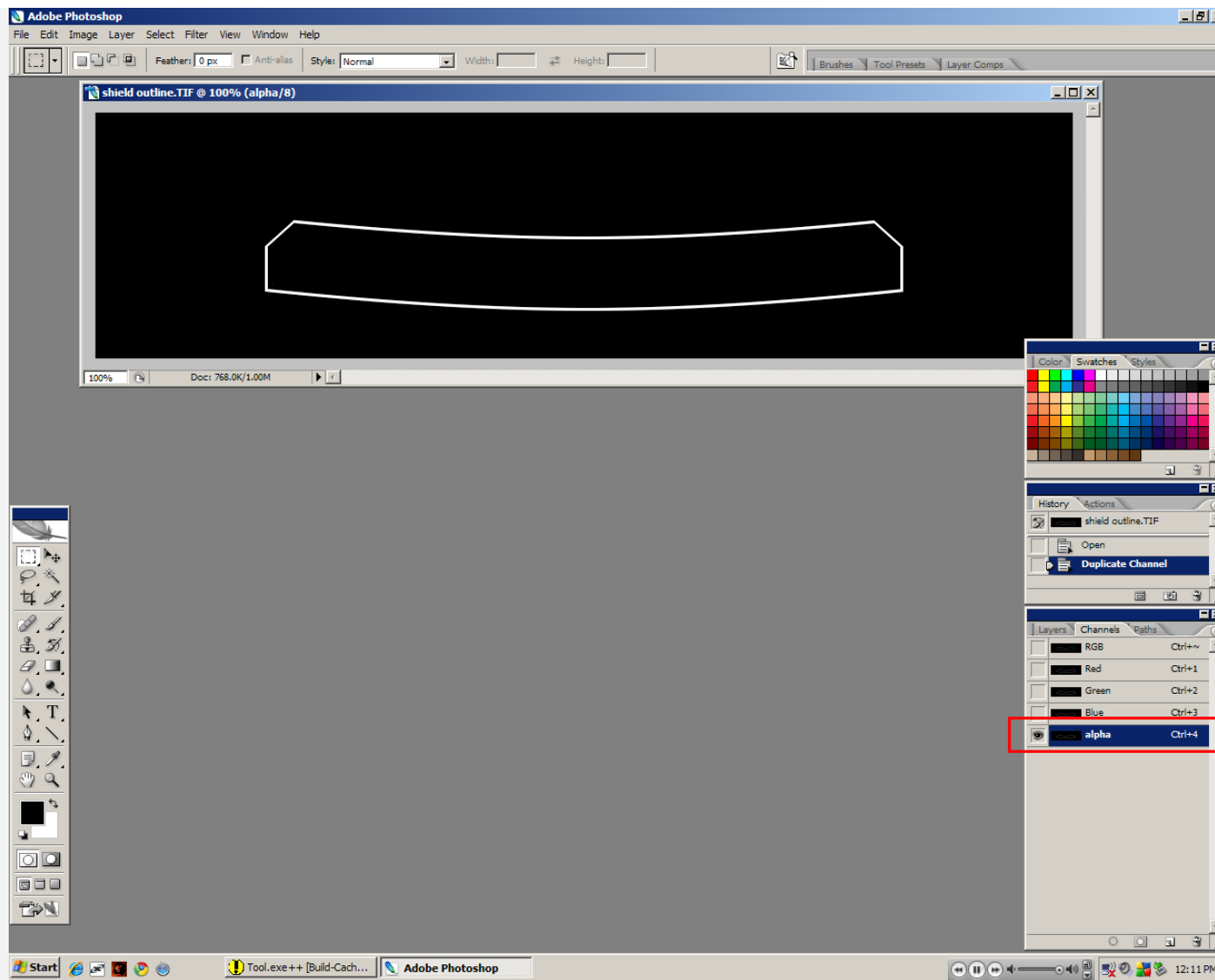
Now save your image (this one was the meter)



Now open the shield outline image



select the channel you like, duplicate it and name it "alpha" and that's it for the outline, no gradient needed.



To compile the bitmaps, you need to have your data folder which is located at

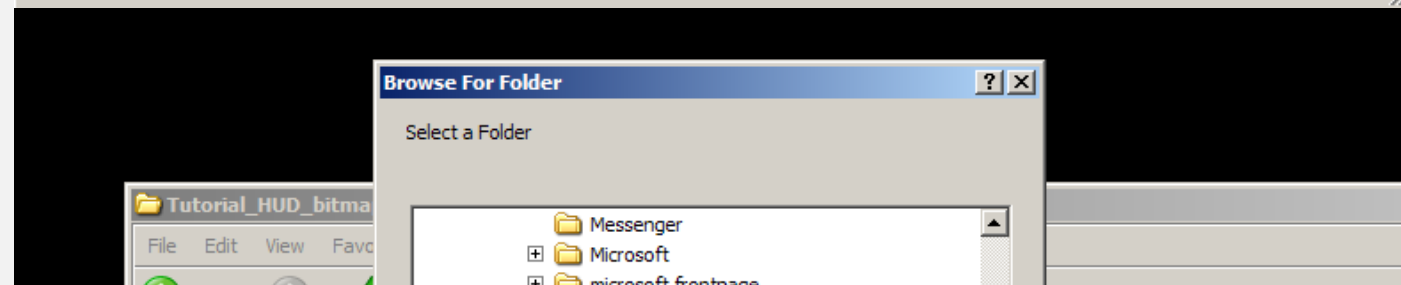
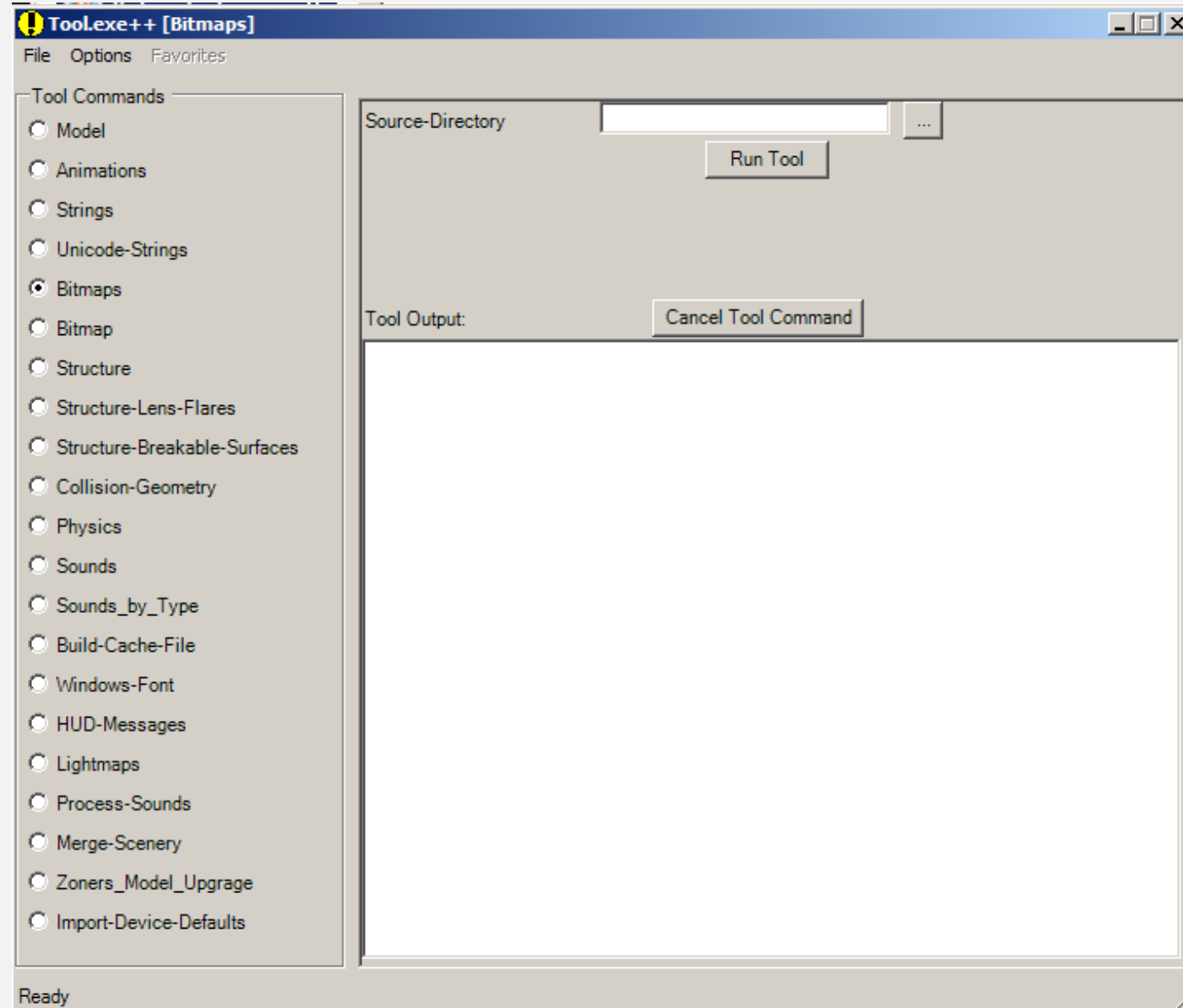
C:\Program Files\Microsoft Games\Halo Custom Edition\data

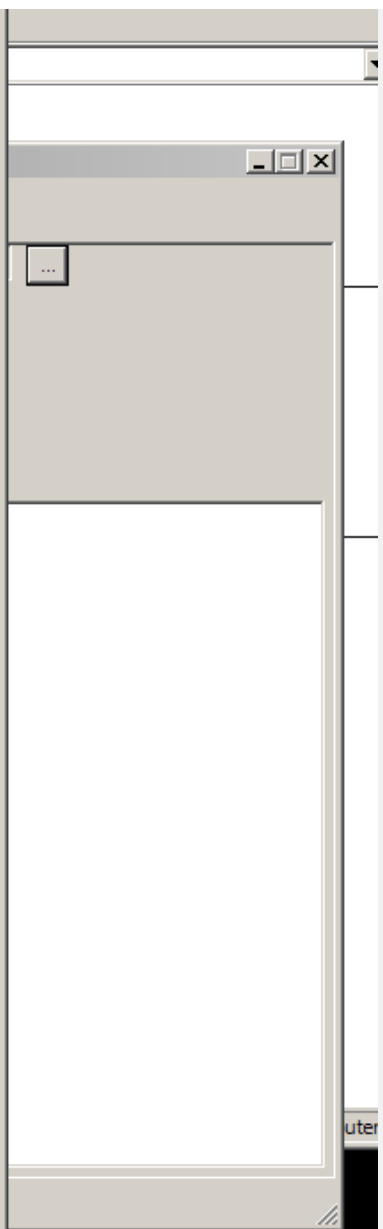
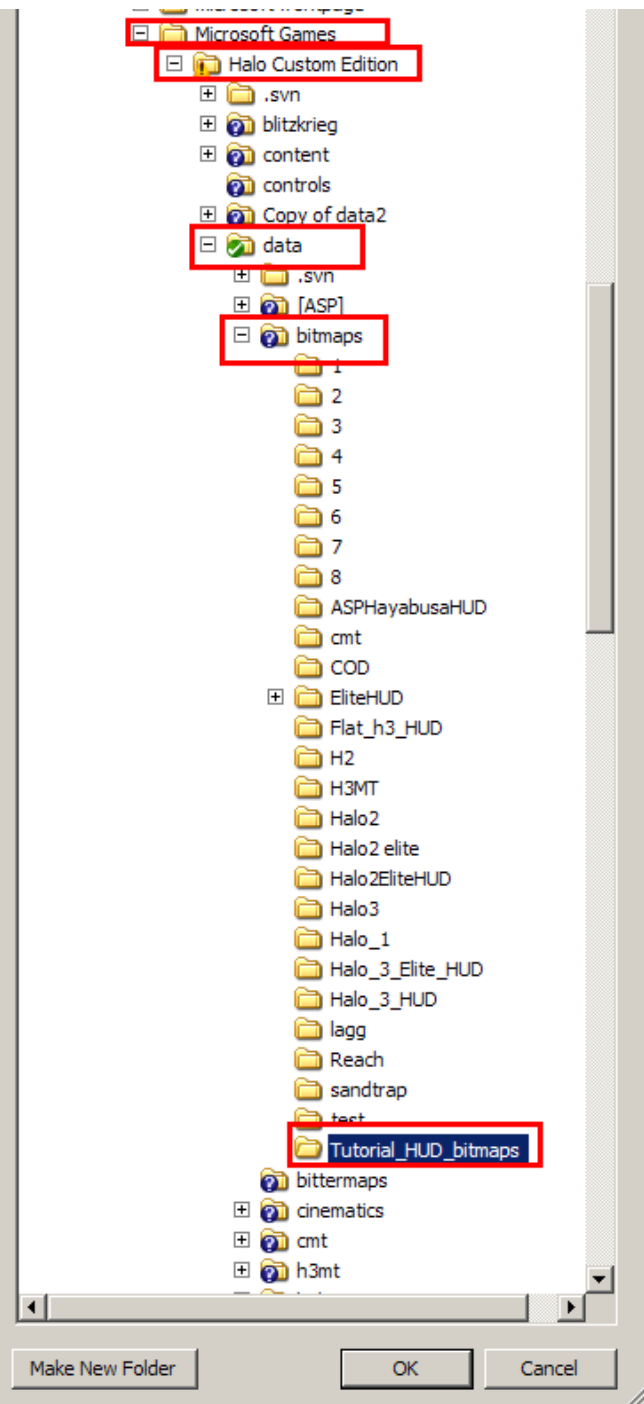
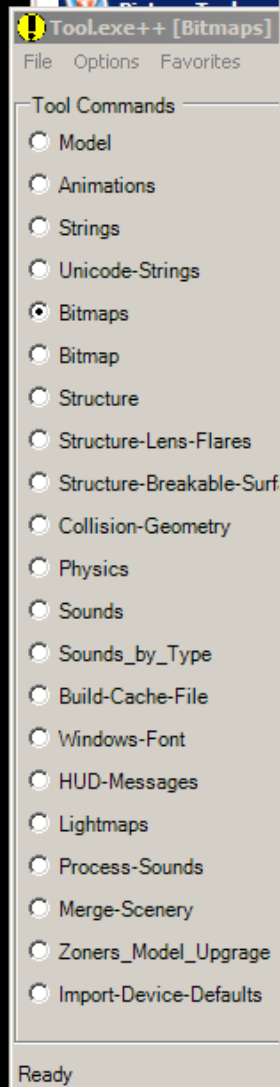
You need to create a folder without any spaces such as this

□

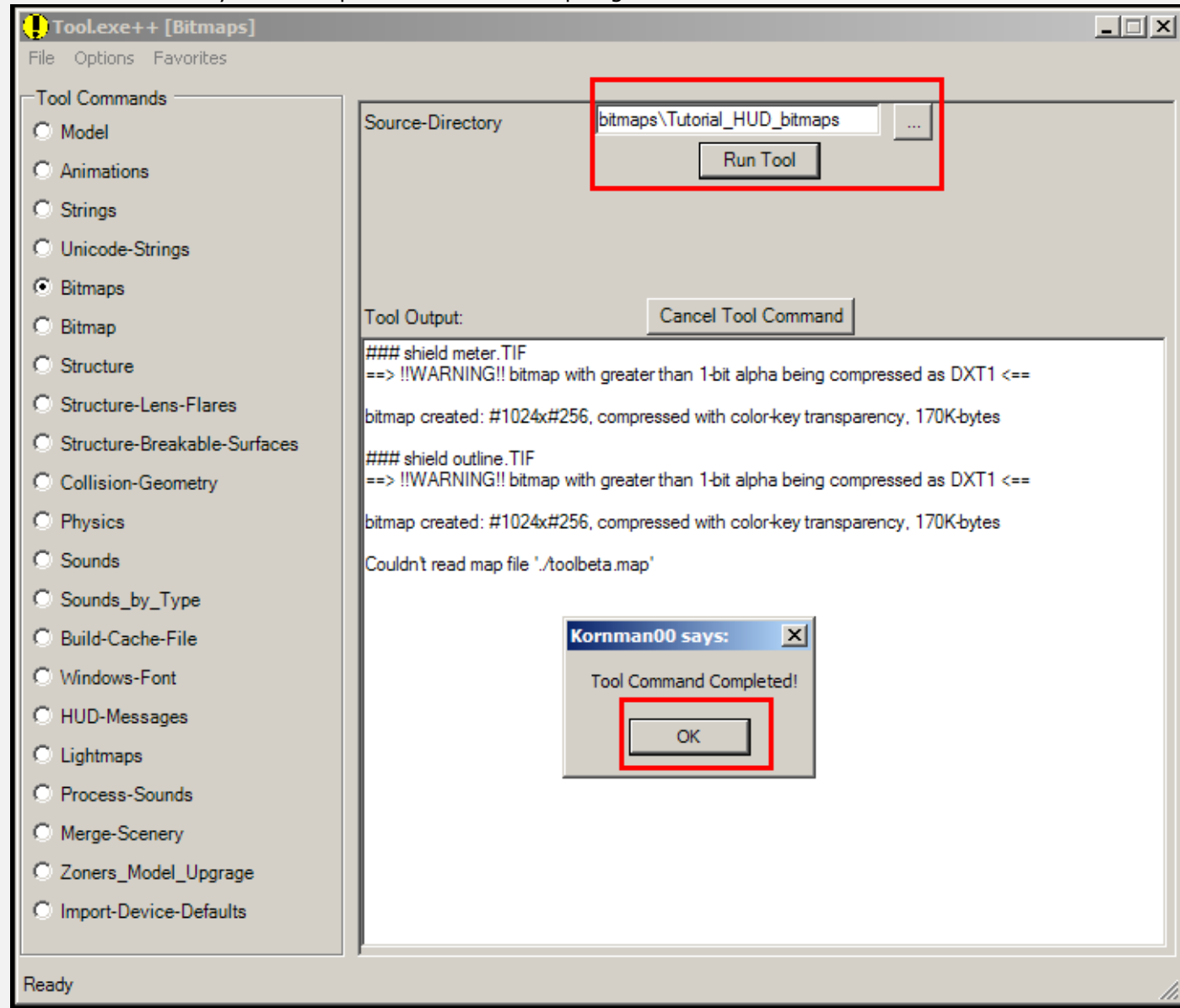
Once you create that folder, you need to run tool (in my case its Tool++) and compile the bitmaps.

For tool++, just click on the bitmaps dot and select your folder.



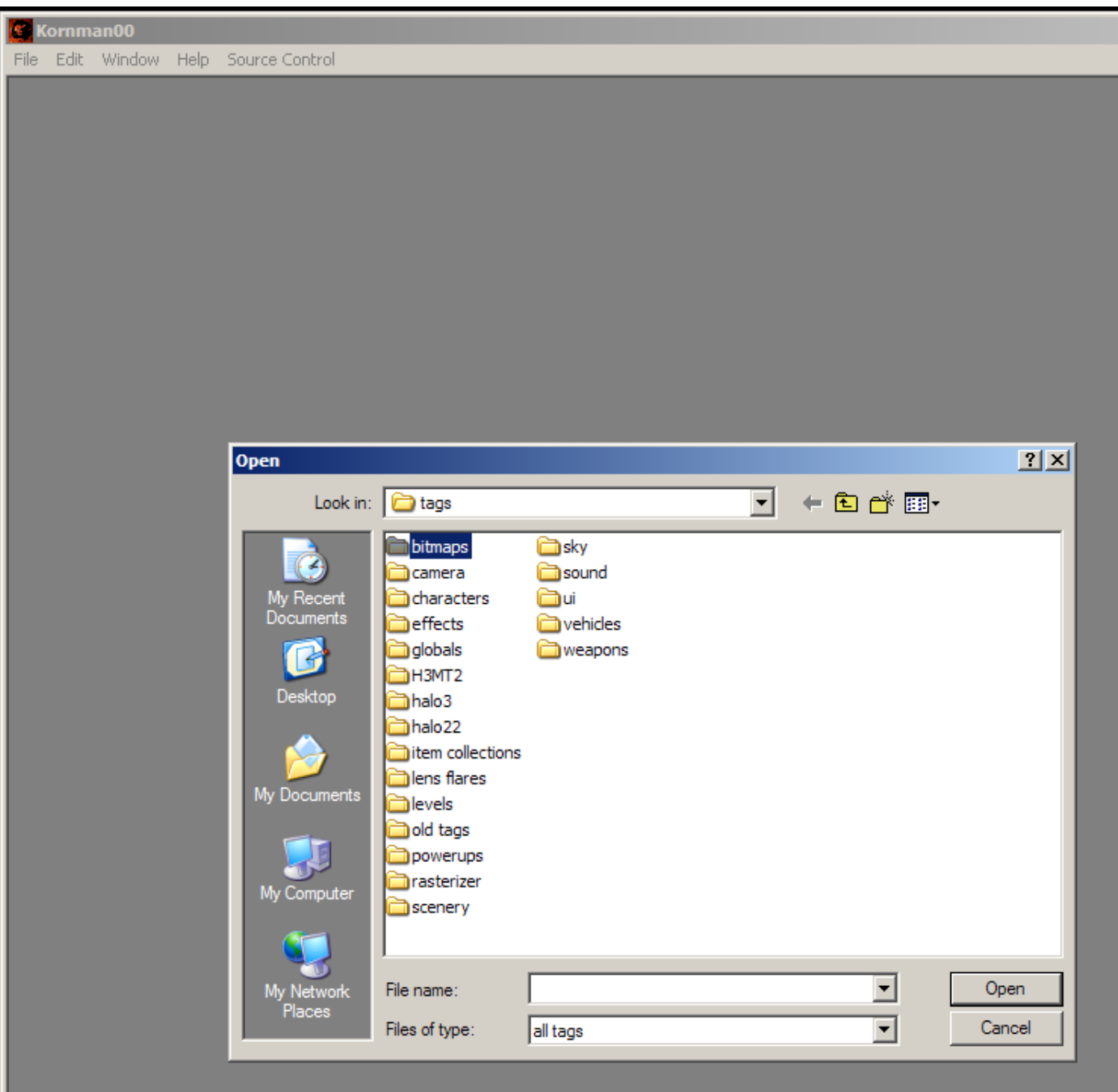


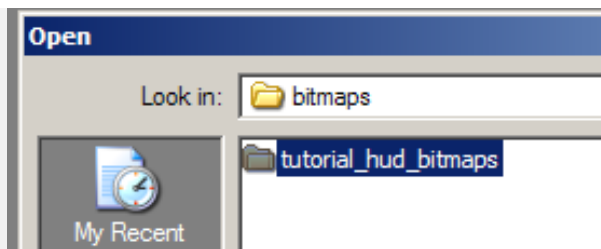
Hit "run tool" and your bitmaps should start compiling



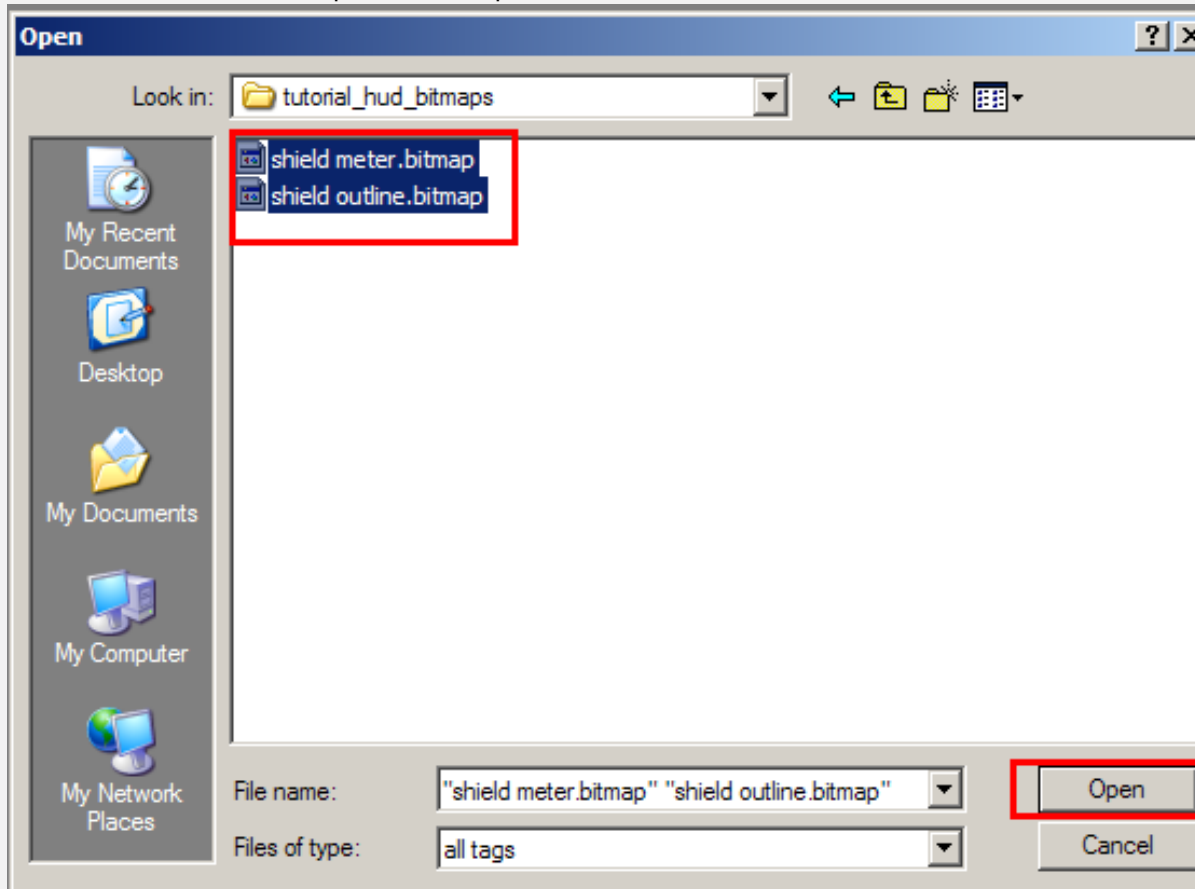
Click OK once it is done.

Once tool compiles the bitmaps, they will be sent to your tags folder. The directory is in reference to the data folders.

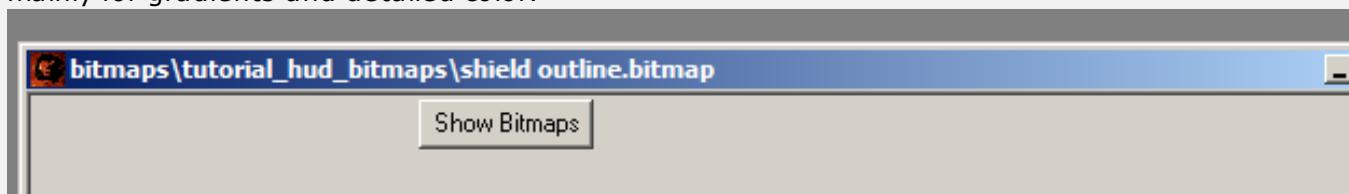




Select both of the bitmaps and hit open



For both images, select 32 bit color. You can use 16 bit color for the outline if you like, 32 bit color is mainly for gradients and detailed color.



type

Type controls bitmap 'geometry'. All dimensions must be a power of two except for SPRITES and INTERFACE BITMAPS:

- * 2D TEXTURES: Ordinary, 2D textures will be generated.
- * 3D TEXTURES: Volume textures will be generated from each sequence of 2D texture 'slices'.
- * CUBE MAPS: Cube maps will be generated from each consecutive set of six 2D textures in each sequence, all faces of a cube map must be square and the same size.
- * SPRITES: Sprite texture pages will be generated.
- * INTERFACE BITMAPS: Similar to 2D TEXTURES, but without mipmaps and without the power of two restriction.

type

2D textures

format

Format controls how pixels will be stored internally:

- * COMPRESSED WITH COLOR-KEY TRANSPARENCY: DXT1 compression, uses 4 bits per pixel. 4x4 blocks of pixels are reduced to 2 colors and interpolated, alpha channel uses color-key transparency instead of alpha from the plate (all zero-alpha pixels also have zero-color).
- * COMPRESSED WITH EXPLICIT ALPHA: DXT2/3 compression, uses 8 bits per pixel. Same as DXT1 without the color key transparency, alpha channel uses alpha from plate quantized down to 4 bits per pixel.
- * COMPRESSED WITH INTERPOLATED ALPHA: DXT4/5 compression, uses 8 bits per pixel. Same as DXT2/3, except alpha is smoother. Better for smooth alpha gradients, worse for noisy alpha.
- * 16-BIT COLOR: Uses 16 bits per pixel. Depending on the alpha channel, bitmaps are quantized to either r5g6b5 (no alpha), a1r5g5b5 (1-bit alpha), or a4r4g4b4 (>1-bit alpha).
- * 32-BIT COLOR: Uses 32 bits per pixel. Very high quality, can have alpha at no added cost. This format takes up the most memory, however. Bitmap formats are x8r8g8b8 and a8r8g8b.
- * MONOCHROME: Uses either 8 or 16 bits per pixel. Bitmap formats are a8 (alpha), y8 (intensity), ay8 (combined alpha-intensity) and a8y8 (separate alpha-intensity).

Note: Height maps (a.k.a. bump maps) should use 32-bit color; this is internally converted to a palettized format which takes less memory.

format

compressed with color-key transparency

compressed with color-key transparency
compressed with explicit alpha
compressed with interpolated alpha
16-bit color
32-bit color
monochrome

usage

Usage controls how mipmaps are

- * ALPHA BLEND: Pixels with zero alpha are ignored in mipmaps, to prevent bleeding the transparent color.
- * DEFAULT: Downsampling works normally, as in Photoshop.
- * HEIGHT MAP: The bitmap (normally grayscale) is a height map which gets converted to a bump map. Uses <bump height> below. Alpha is passed through unmodified.
- * DETAIL MAP: Mipmap color fades to gray, controlled by <detail fade factor> below. Alpha fades to white.
- * LIGHT MAP: Generates no mipmaps. Do not use!

* VECTOR MAP: Used mostly for special effects; pixels are treated as XYZ vectors and normalized after downsampling. Alpha is passed through unmodified.

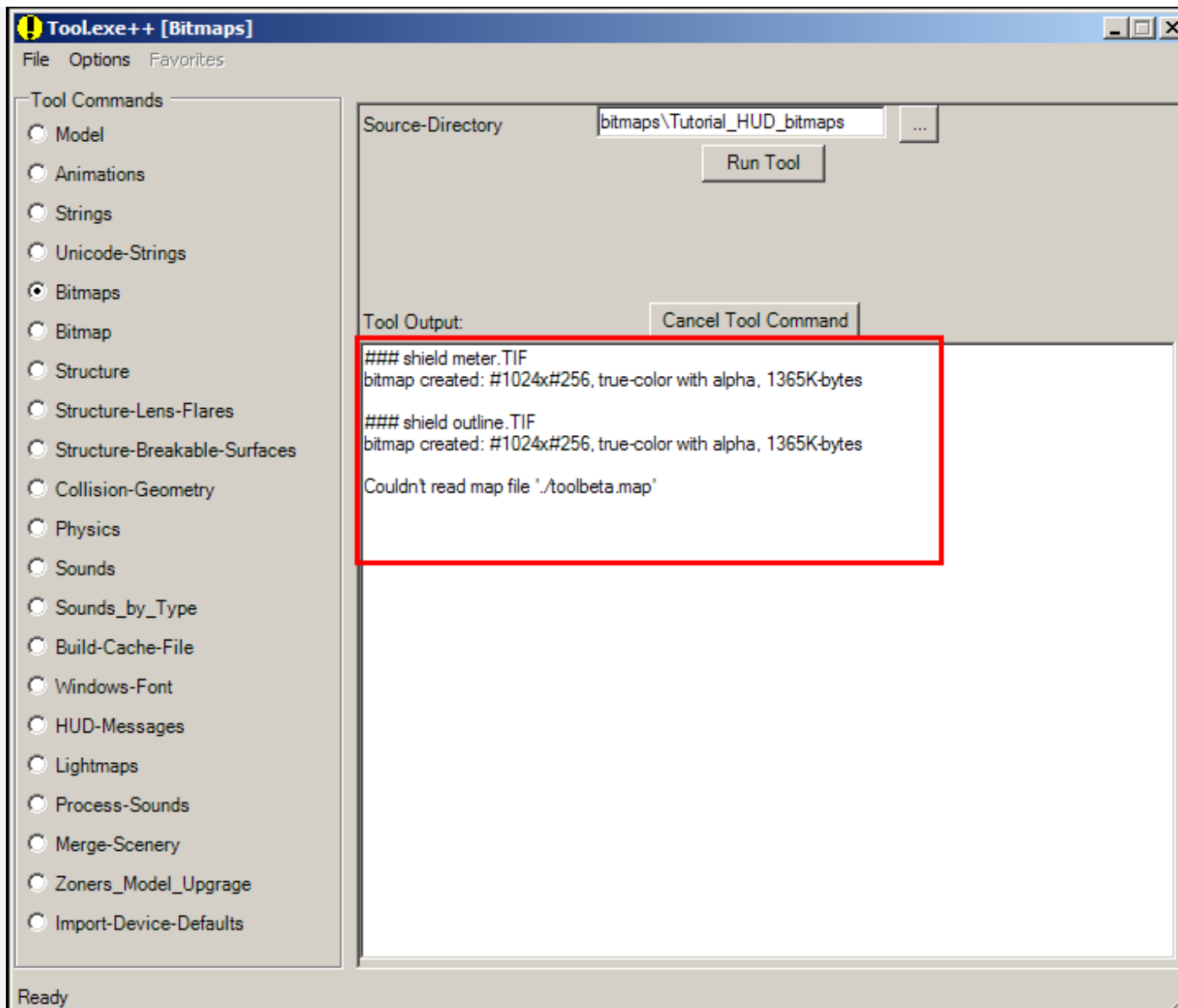
usage

default

flags

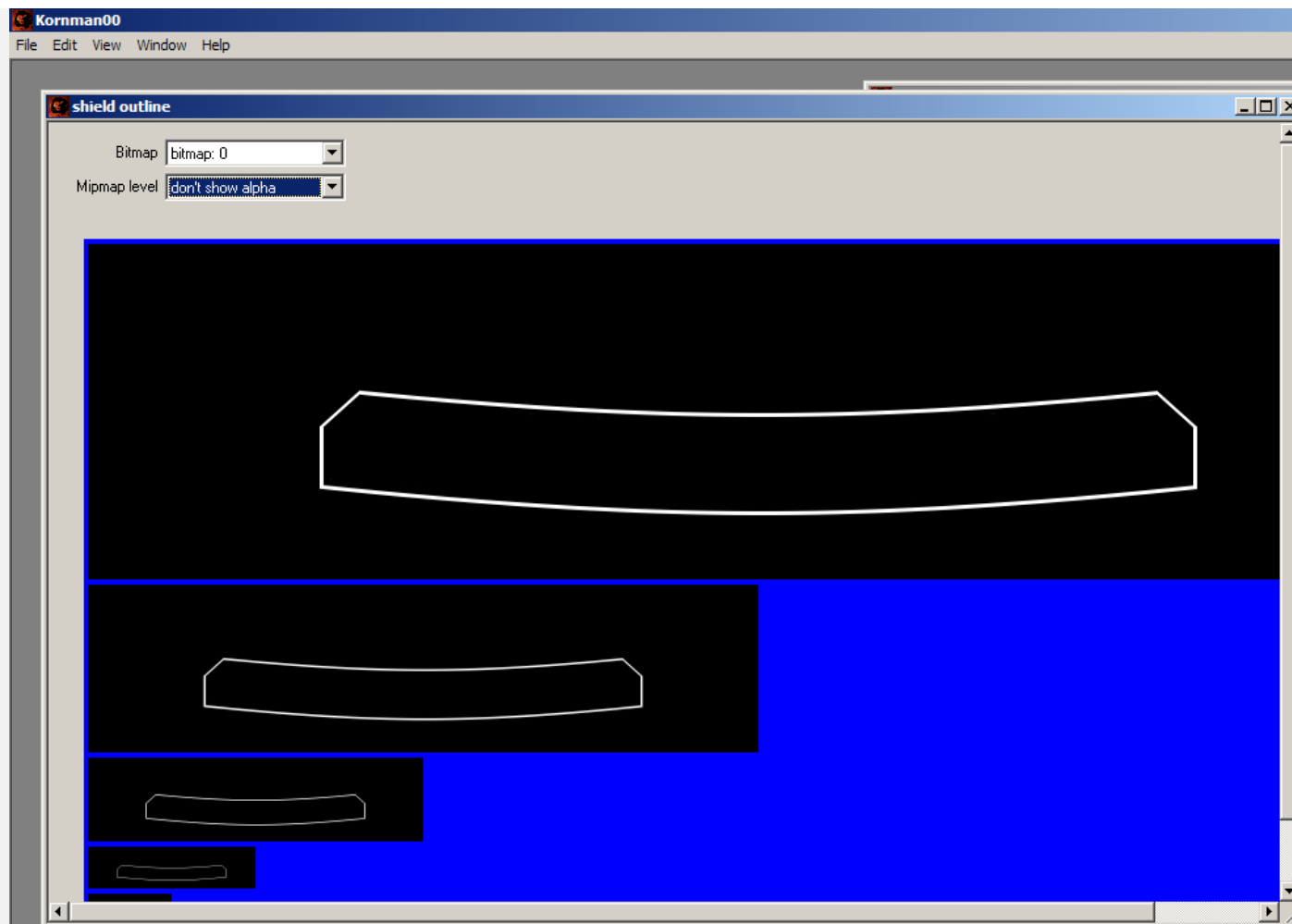
- ☐ enable diffusion dithering
- ☐ disable height map compression
- ☐ uniform sprite sequences
- ☐ filter sprite bug fix

Save the bitmaps and run tool again

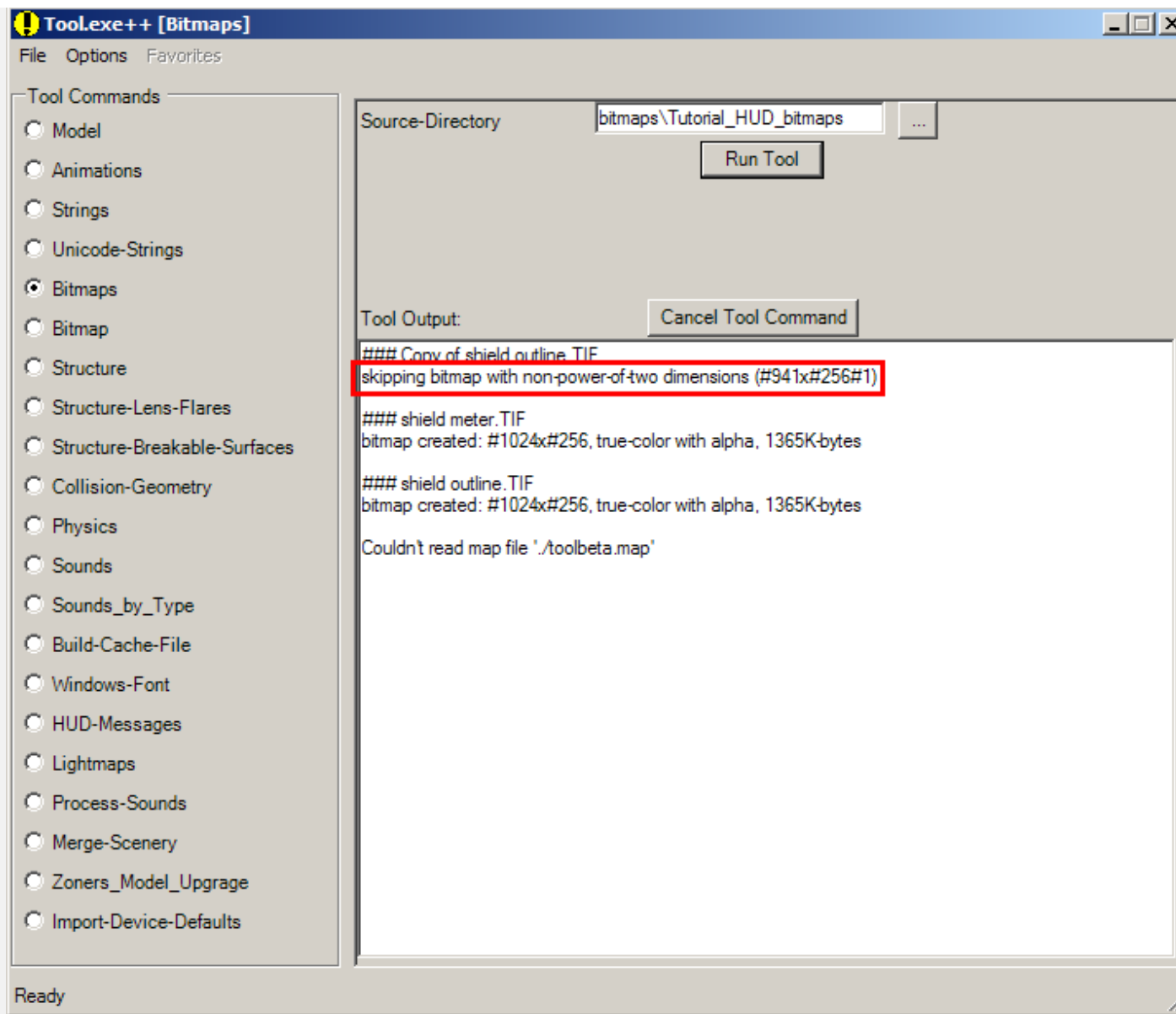


This will apply the changes that you made (to 32 bit)

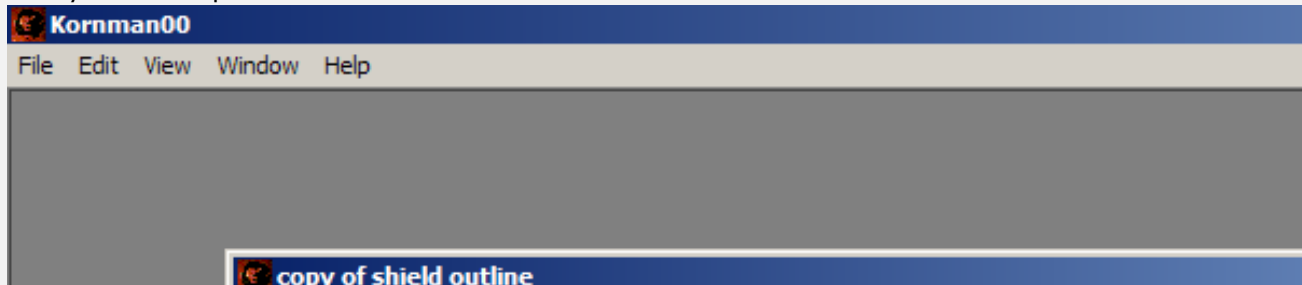
This is what the images should look like after (RGB- alpha)

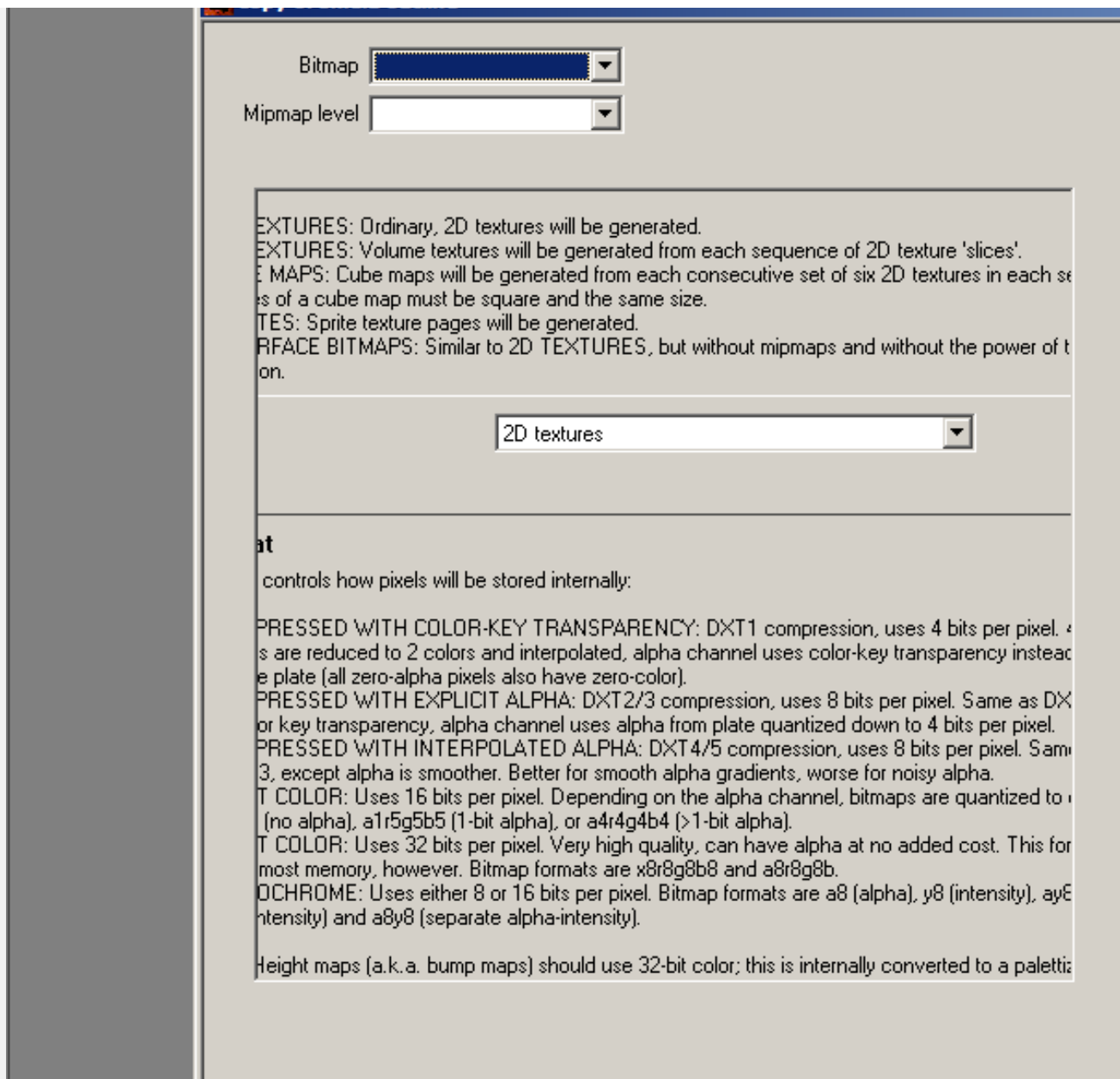


If you F-d up, tool will give you this error.

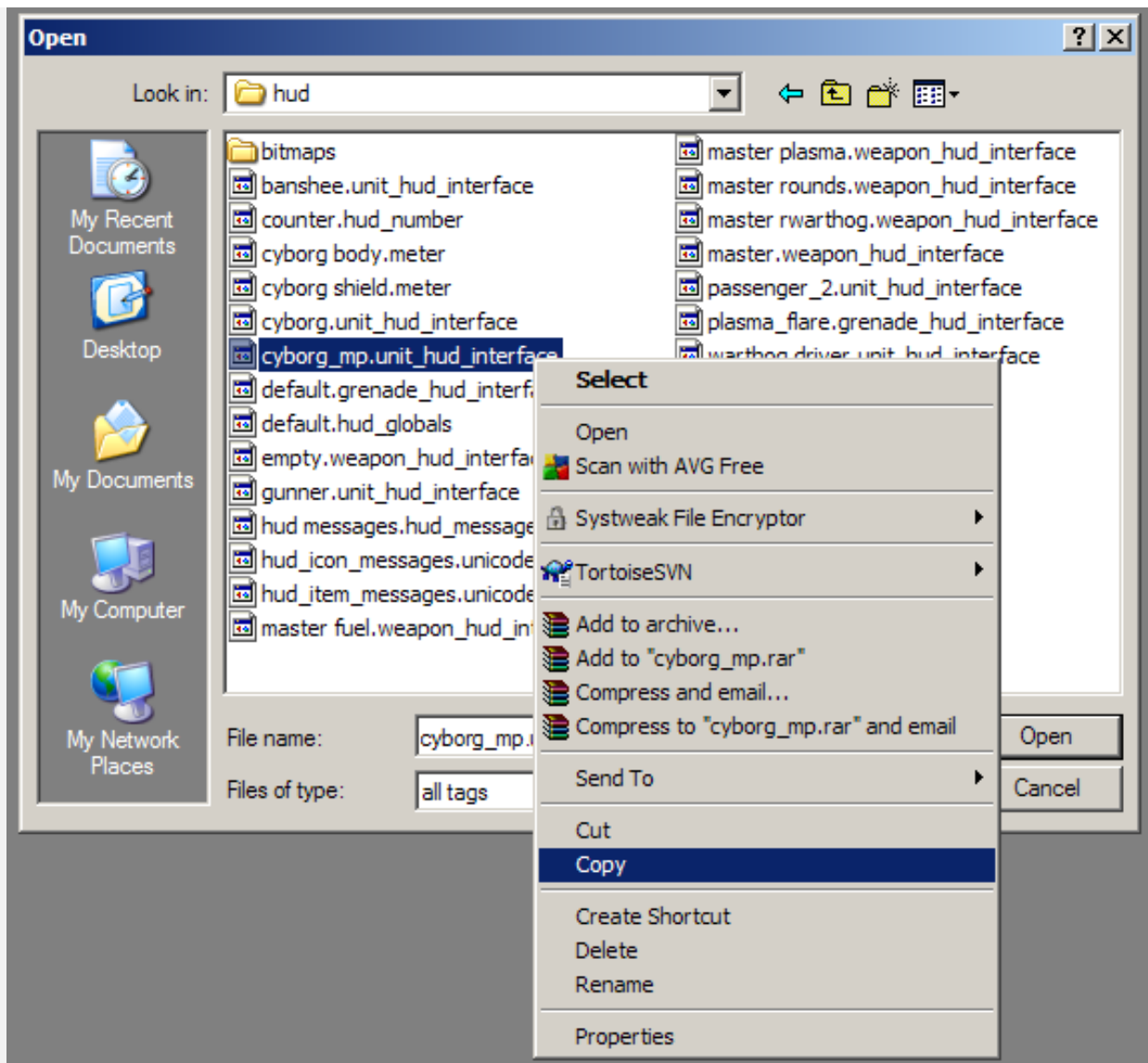


And your bitmap will look like this:

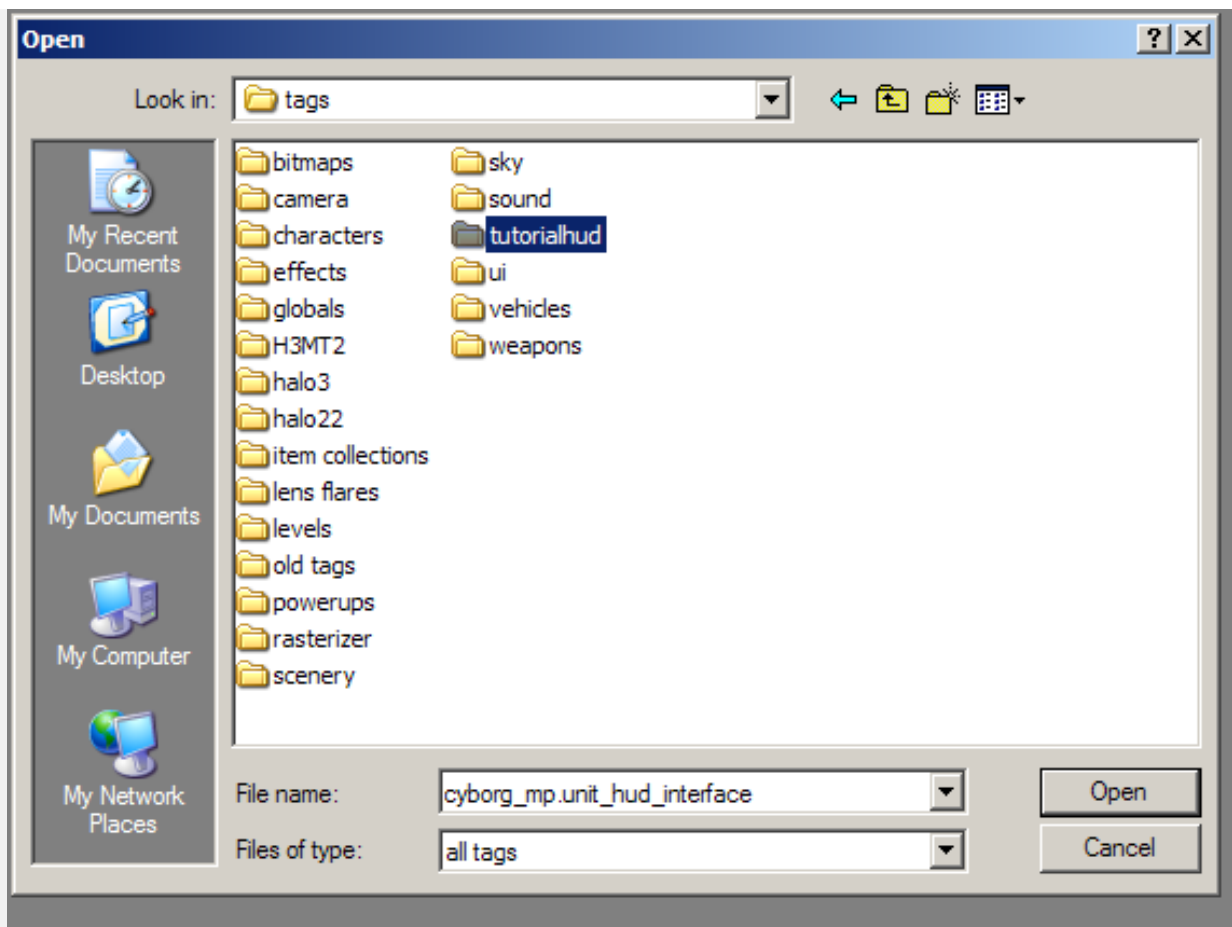




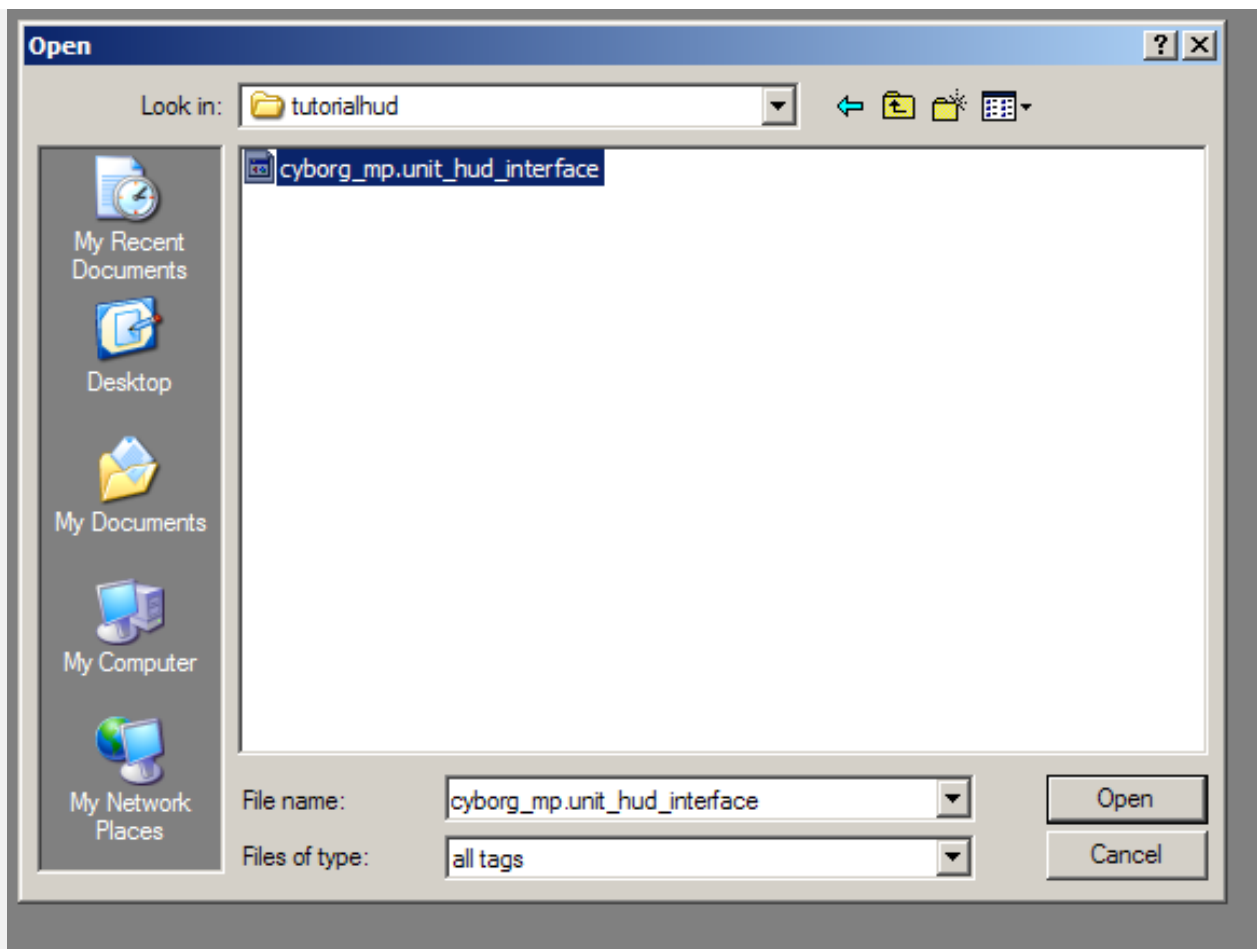
Now to really get the hud ingame
go to your tags\ui\hud folder and copy the cyborg_mp.unit_hud_interface folder.



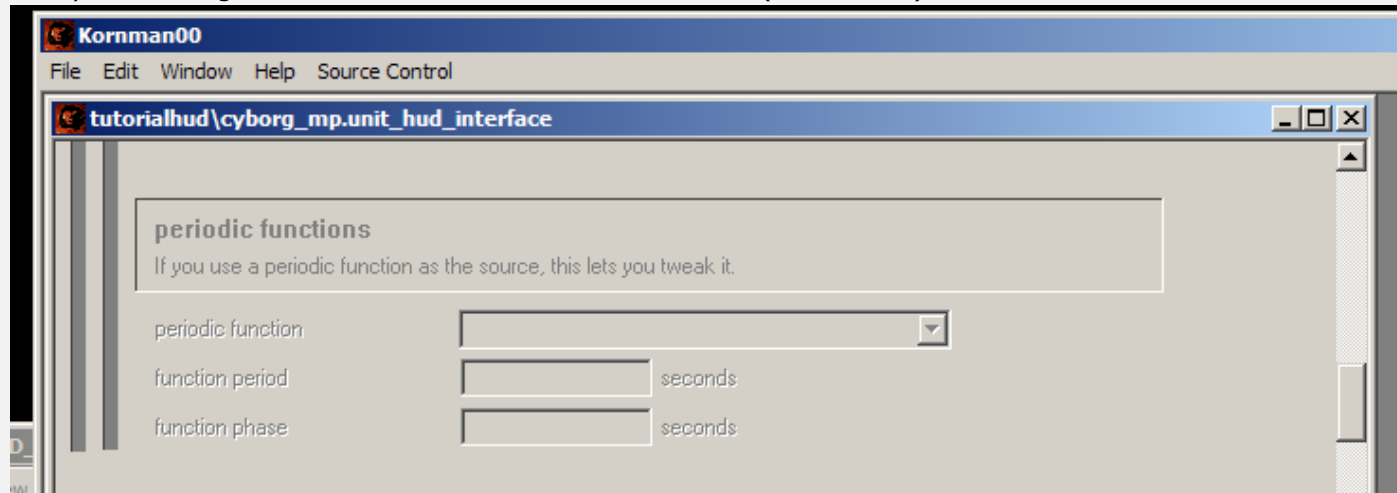
create a new folder, or paste in the same folder you copied it from, but either way, make a new one.

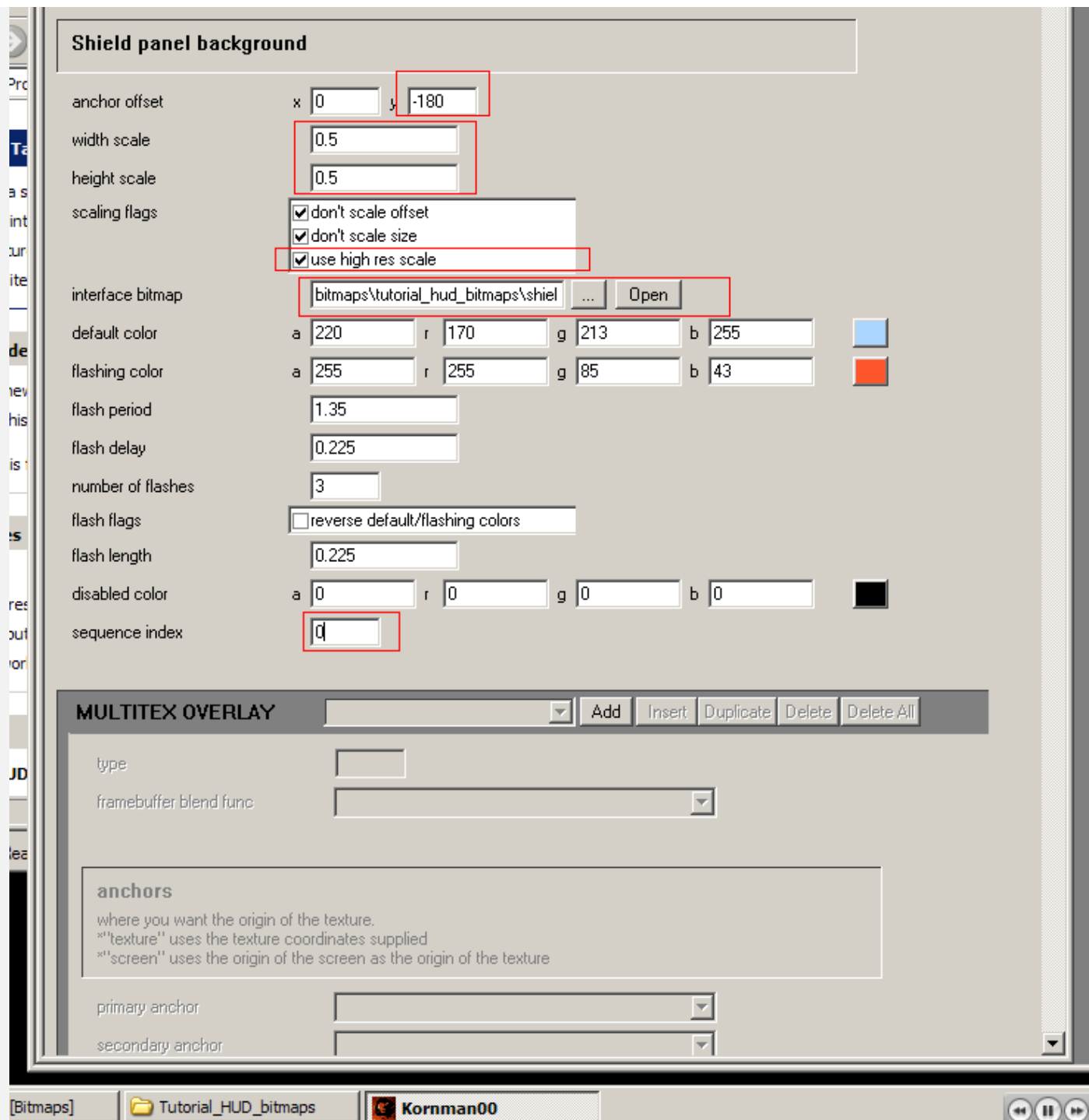


Pate :)



Edit your settings to look like this for the shield outline (background)





Anchor offset - controls the positioning of the bitmap in relation to the center.

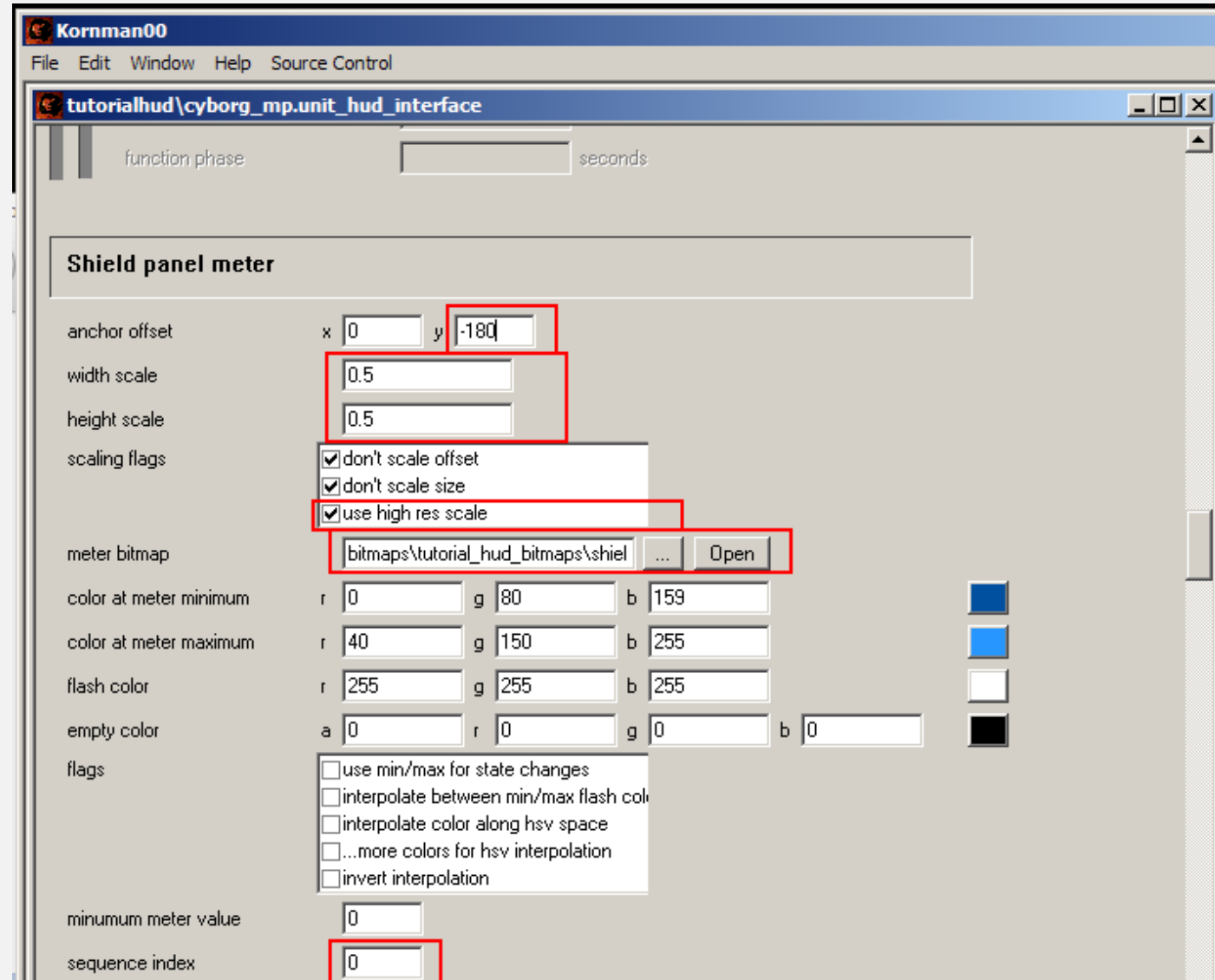
Width scale - controls the width

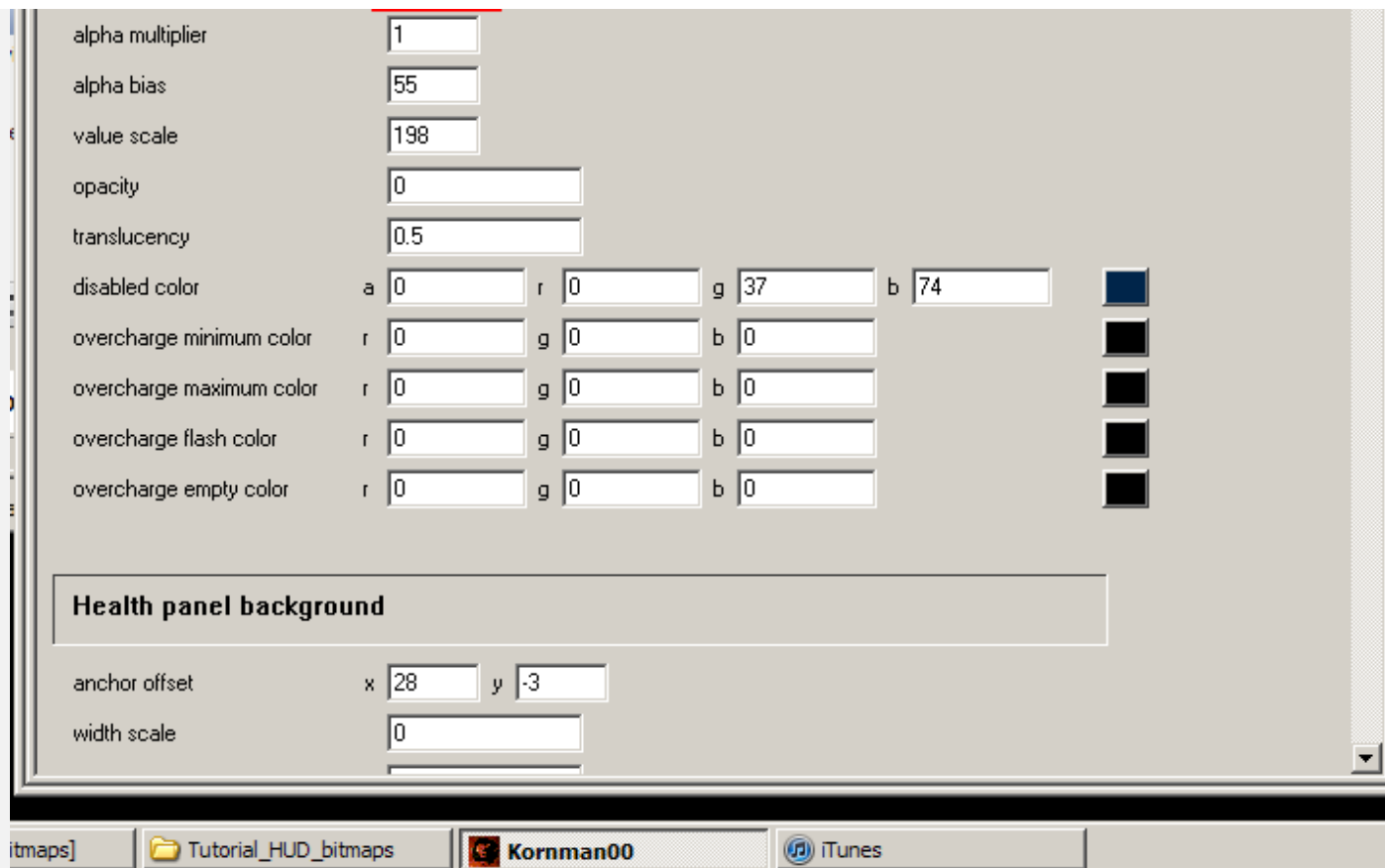
Height scale - controls the height

High res scale flag - basically makes all bitmaps show up at half size ingame

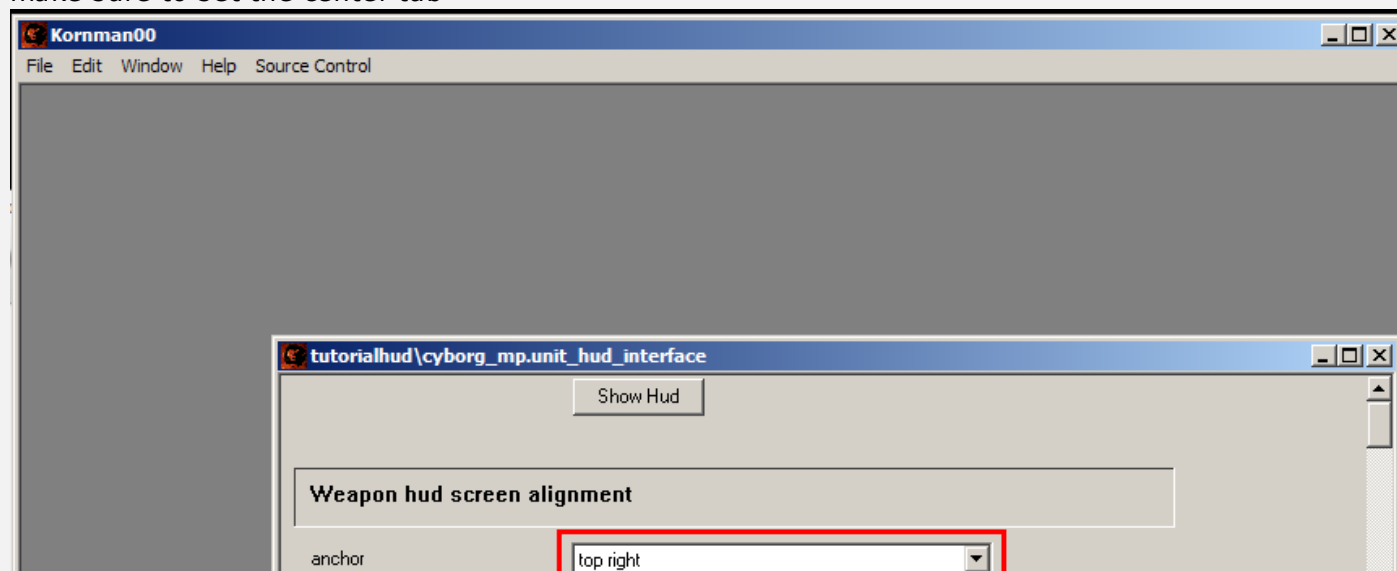
Sequence Index - controls which image is used in the bitmap (useful for sprites)

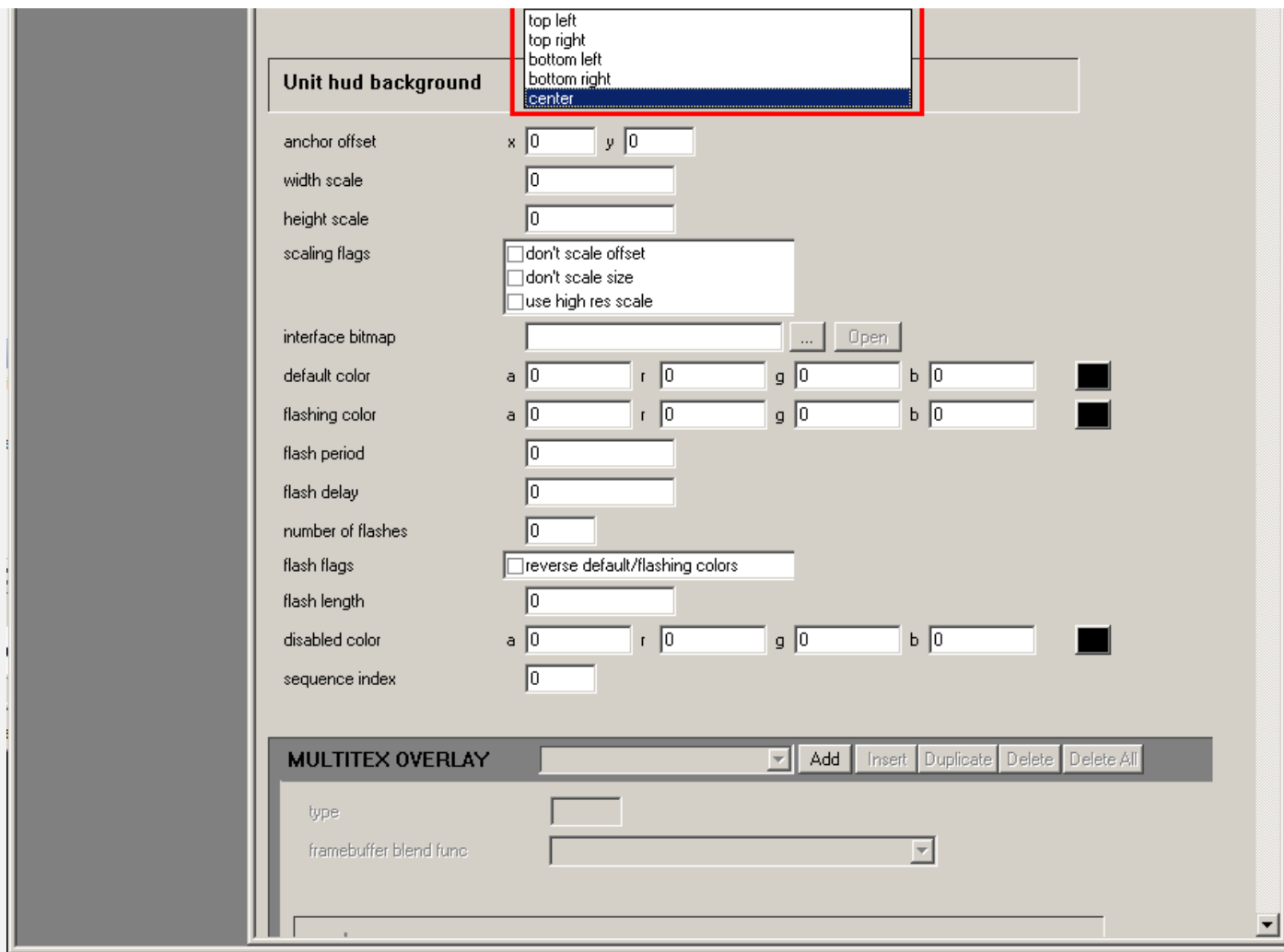
Do this for the shield meter aswell



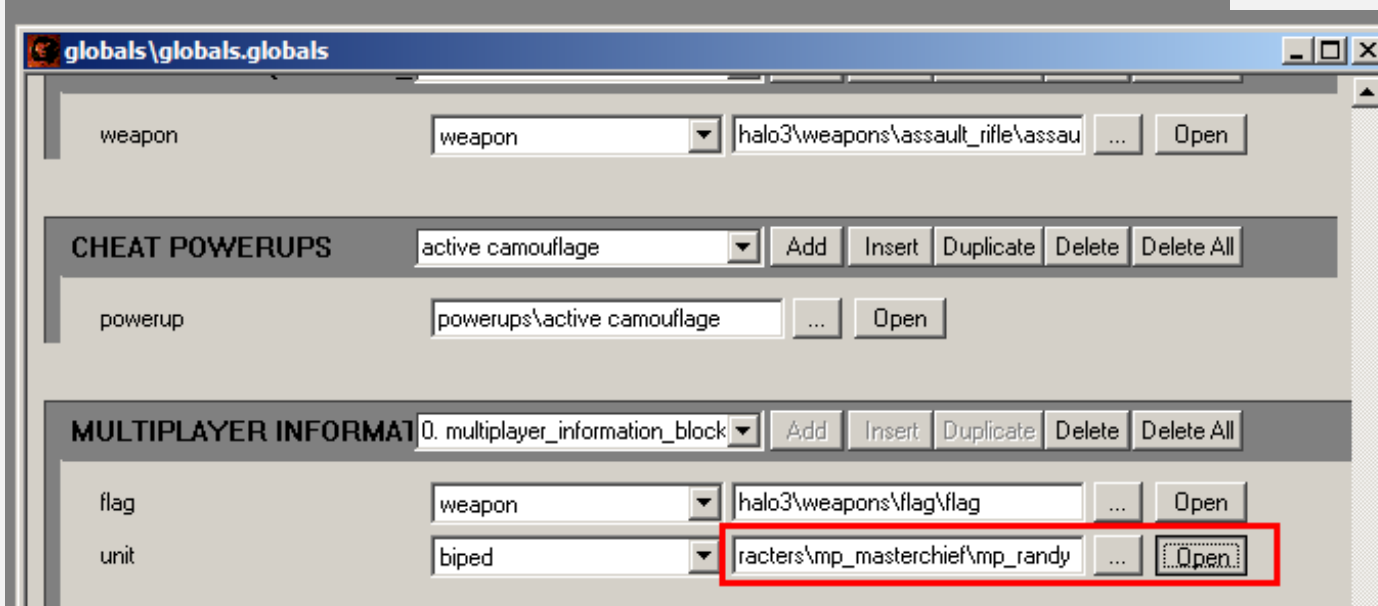
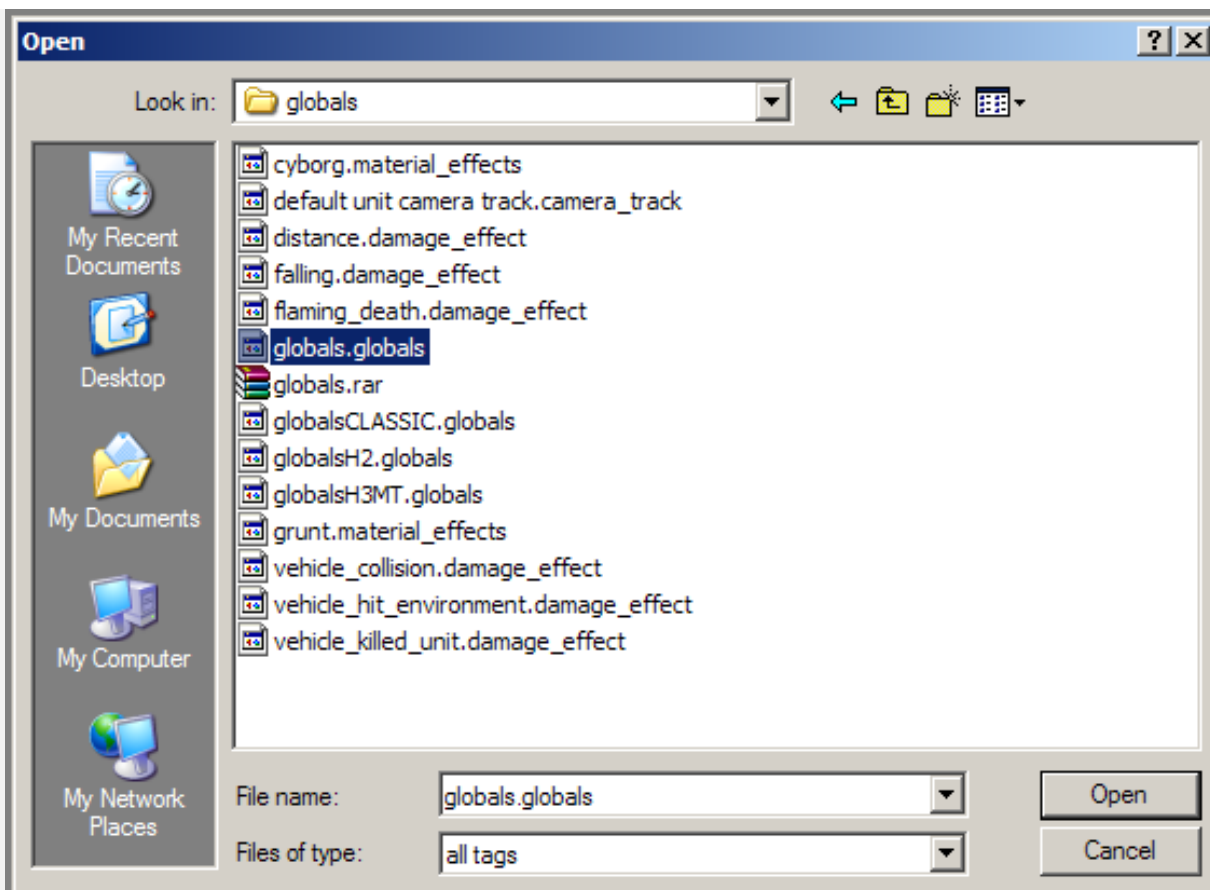


make sure to set the center tab





Now to get your multiplayer biped to use the HUD, go to your globals and open the multiplayer character referenced there.



VEHICLES

mp_warthog

Add

Insert

Duplicate

Delete

Delete All

vehicle

halo3\vehicles\warthog\mp_warth

...

Open

hill shader

shader_transparent_chica

scenery\hilltop\shaders\hilltop

...

Open

flag shader

shader_transparent_chica

weapons\flag\shaders\flag_blue

...

Open

ball

weapon

halo3\weapons\ball\ball

...

Open

SOUNDS

play_ball

Add

Insert

Duplicate

Delete

Delete All

sound

halo3\sounds\announcer\play_ball

...

Open

PLAYER INFORMATION

0. player_information_block

Add

Insert

Duplicate

Delete

Delete All

unit

biped

halo3\characters\sp_masterchief\s

...

Open

walking speed

0.512

world units per second

"double speed" multiplier

1.25

[1.0,+inf]

run forward

2.25

world units per second

run backward

2

world units per second

in the biped file scroll down to new unit hud interfaces and reference your new unit interface folder you created.

halo3\characters\mp_masterchief\mp_randy.biped

aiming acceleration maximum

0

degrees per second squared

casual aiming modifier

0

[0,1]

looking velocity maximum

0

degrees per second

looking acceleration maximum

0

degrees per second squared

AI vehicle radius

0

AI danger radius

0

melee damage

halo3\characters\mp_masterchief\

...

Open

motion sensor blip size

NEW HUD INTERFACES

unit hud interface

DIALOGUE VARIANTS

variant number

dialogue

grenade velocity world units per second

grenade type

grenade count

POWERED SEATS

driver powerup time seconds

driver powerdown time seconds

WEAPONS

weapon

Compile a multi player map and enjoy :)