

Hintstrings not updating



NOTE: Before reading this guide, determine where your World at War root directory is located.

Non-Steam Install: C:\Program Files (x86)\Activision\Call of Duty World at War\

Steam Install: C:\Program Files (x86)\Steam\steamapps\common\Call of Duty World at War\

For the rest of this guide, it will be assumed that the word "root" refers to the appropriate installation path shown above.

Say 'Yes to all' if Windows asks you about replacing files when dragging files into your root directory.

The CoD5 engine has a limit of 48 different hintstrings on any one map at a time. If you have a lot of perk machines, doors with different prices, or custom triggers you may run into an issue where your perk hintstrings don't update when the power turns on or other weird issues with hintstrings that don't change when they are supposed to.

The Idea

Since the engine will not update any more hintstrings once the count exceeds 48, the idea is to use hud.menu to display the hintstrings instead of setHintString().

Implementing the Script

Download [trem_hintstrings.iwd v1.4](#) and drop it in **root/mods/<mapname>/**, then recompile any part of your map or mod. Launcher will auto-copy it to your **AppData/mods/<mapname>** folder next time you compile your mod.

Download [trem_hintstrings_hud.zip v1.1](#) and drop it in **root/raw/ui/**. If you have never made changes to your hud.menu then just click yes to replace the existing file - a backup of the original has already been included in the download. If you have already made changes to your hud.menu, then read [section 5](#). Now recompile your mapname_patch.ff using Launcher, as this file's csv has an include line for hud.menu already. If you have moved your hud.menu to a different FF then compile that one instead obviously.

Practical Usage

Instead of using setHintString on your triggers, you need to add

```
#include maps\trem_hintstrings;
```

to the top of any file you want to use the custom _setHintString() function in. I do not suggest using cross-file calls such as maps\ugxm_util::_setHintString() because WaW has a limit on how many times you can make these calls.

Once you have added the above line, you simply use _setHintString() exactly as you would use setHintString on a trigger:

```
trig _setHintString("This is a hintstring");  
trig setCursorHint("HINT_NOICON");
```

I also suggest implementing this to the following stock scripts for maximum effectiveness:

```
_zombiemode_blockers_new.gsc  
_zombiemode_weapons.gsc
```

Simply open each of those files, add the #include line to the top of the files, then do a Find and Replace for "setHintString" to "_setHintString". Also do a find and replace for

```
&" ZOMBIE_FLAMES_UNAVAILABLE"
```

and replace it with

```
"The power must be activated first!"
```

You can also apply it to `_zombiemode_perks.gsc`, but you will have to replace all localized strings with regular strings first. If you are using the UGX Modtools Patch, you can download these patched versions of those files and simply drop them into your mod's maps folder:
<Link Coming Soon>

Notes

- You can call `_setHintString()` on the same trigger as many times as you want - the new thread will kill the old one before any string conflicts occur.
- This fix **does not work with localized strings**. Use regular strings instead.
- If you want to display a cursorhint on your trigger, you should use `setHintString()` instead of this one because the engine will display the cursor hint properly next to the hintstring.

Merging hud changes with existing hud.menu

If you already made changes to your `hud.menu` and do not want to overwrite the file, simply open your `hud.menu` and scroll down a small amount until you find this:

```
// Cursor hints
menuDef
{
    name "Cursorhints"
    rect 0 70 40 40 HORIZONTAL_ALIGN_CENTER VERTICAL_ALIGN_CENTER
    fullScreen 0
    visible when ( CHAPLIN_CHEAT_OFF && ( dvarString( mapname ) != "credits" ) );
    itemDef
    {
        name "chRect"
        rect 0 0 40 40 // 64x64
        textscale TEXTSIZE_SMALL
        textstyle ITEM_TEXTSTYLE_SHADOWED
        ownerdraw CG_CURSORHINT
        visible 1
        decoration
    }
}
```

Replace that section with this:

```

// Cursor hints
menuDef
{
    name "Cursorhints"
    rect 0 70 40 40 HORIZONTAL_ALIGN_CENTER VERTICAL_ALIGN_CENTER
    fullScreen 0
    visible when ( CHAPLIN_CHEAT_OFF && ( dvarString( mapname ) != "credits" ) );
    itemDef
    {
        name "chRect"
        rect 0 0 40 40 // 64x64
        textscale TEXTSIZE_SMALL
        textstyle ITEM_TEXTSTYLE_SHADOWED
        ownerdraw CG_CURSORHINT
        visible 1
        decoration
    }
    itemDef
    {
        name "trem_hintstring"
        rect 0 0 0 0
        textscale TEXTSIZE_SMALL
        textstyle ITEM_TEXTSTYLE_SHADOWED
        textalign ITEM_ALIGN_CENTER
        exp text (dvarString("trem_hintstring_left") + KeyBinding("+activate") + dvarString
("trem_hintstring_right"))
        visible when(dvarInt("trem_hintstring_vis") == 1 && dvarString("trem_hintstring_left") != "" &&
dvarString("trem_hintstring_left") != " " && dvarString("trem_hintstring_right") != "" && dvarString
("trem_hintstring_right") != " ")
        decoration
    }
    itemDef
    {
        name "trem_hintstring_nobind"
        rect 0 0 0 0
        textscale TEXTSIZE_SMALL
        textstyle ITEM_TEXTSTYLE_SHADOWED
        textalign ITEM_ALIGN_CENTER
        exp text dvarString("trem_hintstring_left")
        visible when(dvarInt("trem_hintstring_vis") == 1 && dvarString("trem_hintstring_left") != "" &&
dvarString("trem_hintstring_left") != " " && (dvarString("trem_hintstring_right") == "" || dvarString
("trem_hintstring_right") == " "))
        decoration
    }
}
}

```

Source Code

```

//|-----|
|-----|
//| Name      : trem_hintstrings.gsc
//| Info      : Fixes hintstrings that don't update.
//| Site      : www.ugx-mods.com
//| Author    : [UGX] treminaor
//| Notes     : v1.3
//|-----|
|-----|

#include maps\_utility;

_setHintString(string) //#bestengine2014
{
    self thread setHintString_fixed_thread(string); //so that i don't have to write "thread" before all of
my trig calls.
}

```

```

setHintString_fixed_thread(string)
{
    self notify("new_thread");
    self endon("new_thread");
    if(!isDefined(string)) return;
    tokens = strTok(string, " ");
    end = false;
    leftstring = "";
    rightstring = "";
    for(i=0;i<tokens.size;i++)
    {
        if(tokens[i] == "&&1" || tokens[i] == "F" || tokens[i] == "[{+activate}]" || tokens[i] ==
"[Use]") //use key will be added by the menufile
        {
            tokens[i] = "";
            end = true;
        }
        if(end) rightstring = rightstring + tokens[i] + " ";
        else leftstring = leftstring + tokens[i] + " ";
    }
    players = getPlayers();
    while(isDefined(self))
    {
        for(k=0;k<players.size;k++)
            if(players[k] islookingatent(self) && (distance(players[k].origin, self.origin) < 75)
|| players[k] isTouching(self))
            {
                players[k].leftstring = leftstring;
                players[k].rightstring = rightstring;
                players[k] thread setHintString_show(self, leftstring, rightstring);
            }
        wait 0.1;
    }
}
setHintString_show(trig, leftstring, rightstring)
{
    if(isDefined(self.lookingatrig) && self.lookingatrig == trig) return; //no need to keep calling the
thread if we're still standing at the same trig
    while(isDefined(trig) && isDefined(self.leftstring) && self.leftstring == leftstring && isDefined(self.
rightstring) && self.rightstring == rightstring && (self islookingatent(trig) && (distance(self.origin, trig.
origin) < 75) || self isTouching(trig)))
    {
        self.lookingatrig = trig;
        self setClientDvar("trem_hintstring_left", string(leftstring));
        self setClientDvar("trem_hintstring_right", string(rightstring));
        self setClientDvar("trem_hintstring_vis", 1);
        wait 0.001;
    }
    self.lookingatrig = undefined;
    self setClientDvar("trem_hintstring_left", " ");
    self setClientDvar("trem_hintstring_right", " ");
    self setClientDvar("trem_hintstring_vis", 0);
}
isLookingAtEnt(ent)
{
    normalvec = vectorNormalize(ent.origin-self geteye());
    veccomp = vectorNormalize((ent.origin-(0,0,256))-self geteye());
    insidedot = vectordot(normalvec,veccomp);

    anglevec = angletoforward(self getplayerangles());
    vectordot = vectordot(anglevec,normalvec);
    if(vectordot > insidedot)
        return true;
    return false;
}

```