

GroundAura

Sign Out

Search...

Members

The Nexus Forums

Messenger

My Conversations

Inquiry about SkyUI config.txt

Participants

cdcooley

Read: Yesterday, 04:14 PM

Block

GroundAura

Read: Today, 10:05 AM

Turn Notifications OFF

Folders

New

My Conversations

Drafts

Edit

Add

Storage

Your messenger storage

0% of your quota (2500 messages)

Search Messages

Go

Disable Messenger

Inquiry about SkyUI config.txt

GroundAura

Sent 02 June 2022 - 09:35 AM

Journeyman

Members

46 posts

Favourite Game: Skyrim

Hi, I'm creating [this mod](#) for SkyUI icons and such. It's achieved mostly through the modification of SkyUI's config.txt file. Unfortunately I have found there is very little information out there about how to modify config.txt. I've learned a lot from Immersive Jewelry's edits to config.txt and have been in contact with it's author, Forteverum.

I asked Forteverum about some commented out text included in his mod, and where the information may have originated. Specifically this:

Spoiler

Show

His response was:

Quote

I think the one responsible for writing that was CDcooley, a pretty great scripster.

Is that correct? And if so, do you think you might be able to share some of your wisdom with me about how to modify config.txt?

Thank you for your time.

Report

cdcooley

Resident poster

4,790 posts

Premium Member

Sent 02 June 2022 - 11:40 AM

Those are actually part of the original documentation included by the SkyUI team, specifically by Schlangster.

What I learned I found by looking at the file and the SkyUI source code. By the time I was interested in using the file Schlangster had already forgotten how it worked. Some of this you're probably already discovered, but I'll describe what I know just in case you haven't.

The first few sections of that file are easy to understand (Input, SearchBox, etc.).

Starting with ListLayout there are a sets of variables that control the actual layout of the various columns of data. Those are also fairly easy to understand once you stare at them long enough but are important because they control the sorting order when you click on the column headers and can even let you add new columns (but be careful with that because there are variations on the main menus that create custom columns).

So as an example, columns.iconColumn controls the appearance of the column that shows icons. It shows icons specifically because its type attribute is set to ITEM_ICON (which is a magic constant like all other uppercase values). There are basic properties like indent, icon.size, border, etc.

The number of states determines how many sorting options you get. The icon column only has two, so there are definitions for the two sorting orders starting with "columns.iconColumn.state1" and "columns.iconColumn.state2". The label for icons is a blank because there's no visible label at the top of that column. arrowDown specifies which direction the little arrow on that label should display while that sorting method is in effect.

The sortAttributes value is a list of values used to determine the order. Looking through those shows just how many different things you can access about the specific items. The one called "text" is the displayed name of the object.

For each of those there's a matching entry in the sortOptions list that needs a order (ASSENDING/DESENDING) and a type of sort (NUMERIC/CASESENSITIVE text, etc.). The items are sorted using the first attribute, then where that attribute is equal sorted on the second, etc.

My default sort for items is to put all equipped items last, with favorites just above that, then sort everything within those groups by name but use formid as a fallback (so that even if two items have the same name, their order doesn't randomly change every time the menu refreshes). Obviously putting equipped and favorites last is a personal choice, but frankly I've already got a favorites menu to see those and the items I'm currently wearing are usually the least interesting to me and the second sort option pulls them to the top if I do want to see them.

columns.itemNameColumn.state1.label.text = <'\$NAME'
columns.itemNameColumn.state1.entry.text = <@text
columns.itemNameColumn.state1.sortAttributes = <@equipstate, @favorite, @text, @formid
columns.itemNameColumn.state1.sortOptions = <{<ASCENDING | NUMERIC>, <ASCENDING | NUMERIC>, <ASCENDING | CASESENSITIVE>, <ASCENDING | NUMERIC>}

There are a number of magic string values you can use along with constants. Things like '\$NAME' are translated string placeholders which will be converted into the correct word in an appropriate language. Things like @text are variable values come from the specific items themselves.

After the columns you get the "views" which are the collections of columns that display on each grouping (Weapons, Armor, Potions, etc.) And within that section, the final group are the "layouts" for the various menus.

Note that when SkyUI finally rebuilt the crafting menus they didn't honor all of the options from the config file. In particular you can't use any of the properties section options below on the crafting menu, it's all hard-coded!

The Properties section is very powerful. It lets you create rules to override almost all attributes and properties of items as well as changing icons, icon colors, and labels.

The "props" variable is important because it holds the actual rules, but nothing happens until you set one of the magic variables "itemProperties", "magicProperties", "itemIcons", "magicIcons", "itemCompoundProperties", and/or "magicCompoundProperties" to reference them. In the base file the only one used is "itemIcons". But you can clearly use it for setting labels and other properties as well a icons.

As for using the rulesets themselves, the best I can offer are the rules I use in my own custom file to redefine icons, colors, and subtype. The datamember1.filter can use any of the attributes/properties that you find in the 'sortAttributes' lists from the columns sections.

```
props.clothingIcon.propertiesToSet = <'iconLabel', 'iconColor', 'subTypeDisplay'  
props.clothingIcon.filter = <'formType': Form.TYPE_ARMOR  
props.clothingIcon.keywords.isTorcLighting = <'iconLabel': 'misc_torc' : some wearable lanterns  
props.clothingIcon.keywords.clothingPouch = <'iconLabel': 'clothing_pouch' : 'iconColor': 0x666666  
props.clothingIcon.keywords.CDC_PortableContainer = <'iconLabel': 'clothing_pouch', 'iconColor': 0x666666  
props.clothingIcon.keywords.clothingRing = <'iconLabel': 'armor_ring', 'subTypeDisplay': 'Ring'  
props.clothingIcon.keywords.clothingAmulet = <'iconLabel': 'armor_amulet', 'subTypeDisplay': 'Amulet'  
props.clothingIcon.keywords.clothingCirclet = <'iconLabel': 'armor_circlet', 'subTypeDisplay': 'Circlet'  
props.clothingIcon.keywords.clothingBracelet = <'iconLabel': 'armor_circlet', 'subTypeDisplay': 'Bracelet'  
  
; recolor the misc items and in a few cases change the icons  
props.otherIcon.propertiesToSet = <'iconLabel', 'iconColor'  
props.otherIcon.filter = <'formType': Form.TYPE_MISC  
props.otherIcon.keywords.CDC_PortableContainer = <'iconLabel': 'clothing_pouch', 'iconColor': 0x888888  
props.otherIcon.dataMember1.filter = <'subTypeDisplay': 'Jockpick'  
props.otherIcon.dataMember1.set = <'iconLabel': 'misc_Jockpick', 'iconColor': 0xB8BAFF  
props.otherIcon.dataMember2.filter = <'subTypeDisplay': 'Hide'  
props.otherIcon.dataMember2.set = <'iconColor': 0x65533  
props.otherIcon.dataMember3.filter = <'subTypeDisplay': 'Leather'  
props.otherIcon.dataMember3.set = <'iconColor': 0x65533  
props.otherIcon.dataMember4.filter = <'subTypeDisplay': 'Strips'  
props.otherIcon.dataMember4.set = <'iconColor': 0x65533  
props.otherIcon.dataMember5.filter = <'subTypeDisplay': 'Gem'  
props.otherIcon.dataMember5.set = <'iconColor': 0xFFFFF  
props.otherIcon.dataMember6.filter = <'subTypeDisplay': 'Gold'  
props.otherIcon.dataMember6.set = <'iconColor': 0xFFFF0  
; make all of the tools use a pickaxe icon  
props.otherIcon.keywords.VendorItemTool = <'iconLabel': 'weapon_pickaxe', 'iconColor': 0xB8BAFF  
props.otherIcon.keywords.VendorItemClutter = <'iconColor': 0xC0CCCC  
  
; custom weapon icons colors mostly  
props.weaponIcon.propertiesToSet = <'iconLabel', 'iconColor'  
props.weaponIcon.filter = <'formType': Form.TYPE_WEAPON  
props.weaponIcon.dataMember1.filter = <'formType': Form.TYPE_WEAPON  
props.weaponIcon.dataMember1.set = <'iconColor': 0xFF6666  
; make this stand out as special  
props.weaponIcon.keywords.DC2NordicPickaxe = <'iconLabel': 'weapon_pickaxe', 'iconColor': 0xB88888  
  
; ammo colors just because ammo needs its own "filter"  
props.ammoColor.propertiesToSet = <'iconColor'  
props.ammoColor.filter = <'formType': Form.TYPE_AMMO  
props.ammoColor.dataMember1.filter = <'formType': Form.TYPE_AMMO  
props.ammoColor.dataMember1.set = <'iconColor': 0xA66666  
  
; I like spell tomes to have a slightly different color  
props.bookColor.propertiesToSet = <'iconColor'  
props.bookColor.filter = <'formType': Form.TYPE_BOOK  
props.bookColor.dataMember1.filter = <'subTypeDisplay': 'Spell Tome'  
props.bookColor.dataMember1.set = <'iconColor': 0xDCCCFF  
  
; major change to colors for armor clothing and jewelry  
props.armorColor.propertiesToSet = <'iconColor'  
props.armorColor.filter = <'formType': Form.TYPE_ARMOR  
props.armorColor.dataMember1.filter = <'weightClass': Armor.WEIGHT_LIGHT  
props.armorColor.dataMember1.set = <'iconColor': 0x49955  
props.armorColor.dataMember2.filter = <'weightClass': Armor.WEIGHT_HEAVY  
props.armorColor.dataMember2.set = <'iconColor': 0x33633  
props.armorColor.dataMember3.filter = <'weightClass': Armor.WEIGHT_CLOTHING  
props.armorColor.dataMember3.set = <'iconColor': 0x55665  
props.armorColor.dataMember4.filter = <'weightClass': Armor.WEIGHT_JEWELRY  
props.armorColor.dataMember4.set = <'iconColor': 0xA6AAAA  
  
; tone down the colors for potions  
props.potionColor.propertiesToSet = <'iconColor'  
props.potionColor.filter = <'formType': Form.TYPE_POTION  
props.potionColor.dataMember1.filter = <'subType': Item.POTION_HEALTH  
props.potionColor.dataMember1.set = <'iconColor': 0xFF99AA  
props.potionColor.dataMember2.filter = <'subType': Item.POTION_HEALTHRATE  
props.potionColor.dataMember2.set = <'iconColor': 0xFF99AA  
props.potionColor.dataMember3.filter = <'subType': Item.POTION_HEALTHRATEMULT  
props.potionColor.dataMember3.set = <'iconColor': 0xFF99AA  
props.potionColor.dataMember4.filter = <'subType': Item.POTION_MAGICCKA  
props.potionColor.dataMember4.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember5.filter = <'subType': Item.POTION_MAGICCKARATE  
props.potionColor.dataMember5.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember6.filter = <'subType': Item.POTION_MAGICCKARATEMULT  
props.potionColor.dataMember6.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember7.filter = <'subType': Item.POTION_STAMINA  
props.potionColor.dataMember7.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember8.filter = <'subType': Item.POTION_STAMINARATE  
props.potionColor.dataMember8.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember9.filter = <'subType': Item.POTION_POISON  
props.potionColor.dataMember9.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember10.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember10.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember11.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember11.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember12.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember12.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember13.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember13.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember14.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember14.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember15.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember15.set = <'iconColor': 0xB8BAFF  
props.potionColor.dataMember16.filter = <'subType': Item.POTION_FIBERESIST  
props.potionColor.dataMember16.set = <'iconColor': 0xB8BAFF  
  
; make sure edible ingredients look like food  
props.edibleTag.propertiesToSet = <'iconColor', 'subTypeDisplay'  
props.edibleTag.filter = <'formType': Form.TYPE_INGREDIENT  
props.edibleTag.keywords.VendorItemFood = <'iconColor': 0x77777, 'subTypeDisplay': 'Food'  
props.edibleTag.keywords.VendorItemFoodRaw = <'iconColor': 0x99999, 'subTypeDisplay': 'Food'  
  
itemIcons = @potionColor, @ammoColor, @bookColor, @armorColor, @clothingIcon, @weaponIcon, @otherIcon, @edibleTag
```

And finally the SkyUI source can be helpful in understanding how defaults are set.

<https://github.com/Aurora-Mod/DataSetter.as>

<https://github.com/Aurora-Mod/IconSetter.as>

Vila is just a companion. But that is sort of like saying a phoenix spouting fire and swooping down on a battlefield is just a bird. — [VoxClemant](#)

I didn't really set out to make a follower, I set out to make an Inigo. — [SmartBluecat](#)

Report

GroundAura

Sent 02 June 2022 - 12:30 PM

Journeyman

Members

46 posts

Favourite Game: Skyrim

Thanks so much for all this. You've confirmed a lot I've figured out, and filled in a lot of holes I'd had in my understanding.

Currently I have one question. When you said,

Quote

Those are actually part of the original documentation included by the SkyUI team, specifically by Schlangster.

does that mean that documentation is available to the public, on Schlangster's github or something? Or where might I find the rest of that documentation?

Thanks again, you've been a major help.

Report

cdcooley

Resident poster

4,790 posts

Premium Member

Sent 02 June 2022 - 02:22 PM

No, when I said documentation I just meant comments left in the file. While there are a few more comments embedded here and there in the source code on github, there really isn't anything truly useful when it comes to this topic. Those two source files I linked and a file that defines a bunch of the constants are about it.

Vila is just a companion. But that is sort of like saying a phoenix spouting fire and swooping down on a battlefield is just a bird. — [VoxClemant](#)

I didn't really set out to make a follower, I set out to make an Inigo. — [SmartBluecat](#)

Report

GroundAura

Sent Yesterday, 03:18 AM

Journeyman

Members

46 posts

Favourite Game: Skyrim

Alright. Sorry to keep bothering you. I just like to have as much information as possible.

Fast Reply

Post

Use Full Editor