Delete Compose New

Participants

cdcooley
Beat Yesterday, 02:22 PM
Block

CrowdAura
Beat Today, 09:56 AM

Turn Notifications OFF

Folders

New 1
My Conversations 2
Drafts 1

Edit Add

Storage

Your messenger storage
0% of your quota (2500 messages)

Search Messages

Go

Disable Messenger

Inquiry about Skyui3 config.txt

CrowdAura Sent Yesterday, 09:35 AM

JoeSteerman
Member
31 posts
Favourite Game: Skyrim

Hi, I'm creating this mod for Skyui icons and such. It's achieved mostly through the modification of Skyui3'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, Forbreatum.

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

Spoiler Hide

```
;
; ------------------------------------------------------------------------------
; {Properties}
; ------------------------------------------------------------------------------
;
; SET PROPERTIES ---------------------------------------------------------------
; Set values based on keywords/attributes
; Often set a property like material with a numerical value for proper sorting
; and then materialsortbyvalue with keyword to sort by material
; ALL_CAPS like MATERIAL_IRON are constant numerical values set by skyui3.
; ALL_CAPS like ARMOR_HEAVY are keywords.
; .displayattr.keywordvalue.set = /propertyfloat/ value|float
; for setting properties based on displayattr.
; .displayattr.filter = /displayattr/ value|bool
; .displayattr.set = /propertyfloat/ value|float
;
; Set Compound Properties ------------------------------------------------------
; Set properties that are a combination of other properties, used for icon column sorts
; for changing compositions of displayattr.keywordvalue.set and .filter
; Set weight=1 and weight=2 to .displayattr.keywordvalue.set = /floatfloat/ weight|float
; will put all equal heavy armor together and then sort by body/heavy/etc vs the default of put
; together and then sorting by weight item.
;
; Properties -------------------------------------------------------------------
; IconProperties
; magicProperties
;
; .propIconGroup.propertyfloat = /subtype/ subtypeDisplay|
; .propIconGroup.filter = /formtype/ ...|
; .propIconGroup.overwrite = true
; .propIconGroup.dataMembers.filter = /formid/ ...|
; .propIconGroup.dataMembers.set = /subtype/ ...|/subtypeDisplay/ ...|
;
; .propIconGroupOther.propertyfloat = /subtype/ /subtypeDisplay/|
; .propIconGroupOther.filter = /formtype/ ...|
; .propIconGroupOther.defaultvalues = /subtype/ ...|/subtypeDisplay/ ...|
; .propIconGroupOther.keywords.SomeKeyword = /subtype/ ...|/subtypeDisplay/ ...|
;
; IconProperties = /itemGroup/, /itemGroupOther/, ...|
;
;
;
; .propMagicGroup.propertyfloat = /subtype/ /subtypeDisplay/|
; .propMagicGroup.filter = /formtype/ ...|
; .propMagicGroup.dataMembers.filter = /formid/ ...|
; .propMagicGroup.dataMembers.set = /subtype/ ...|/subtypeDisplay/ ...|
;
; .propMagicGroupOther.propertyfloat = /subtype/ /subtypeDisplay/|
; .propMagicGroupOther.filter = /formtype/ ...|
; .propMagicGroupOther.defaultvalues = /subtype/ ...|/subtypeDisplay/ ...|
; .propMagicGroupOther.keywords.SomeKeyword = /subtype/ ...|/subtypeDisplay/ ...|
;
; magicProperties = /magicGroup/, /magicGroupOther/, ...|
;
;
;
; Icon Labels ------------------------------------------------------------------
; IconLabels
; magicLabels
;
; .propIconGroup.propertyfloat = /iconLabel/, /iconColor/|
; .propIconGroup.filter = /formtype/ ...|
; .propIconGroup.defaultvalues = /iconLabel/ ...|
; .propIconGroup.dataMembers.filter = ...
; .propIconGroup.dataMembers.set = ...
;
; IconLabels = /iconGroup/, ...|
;
;
;
; .propMagicIconGroup...
;
; magicIconLabels = /magicIconGroup/, ...|
;
; Compound Properties ----------------------------------------------------------
; IconCompoundProperties
; magicCompoundProperties
;
; .compoundProps.iconCompoundGroup.propertyfloat = /mySortProperty/
; .compoundProps.iconCompoundGroup.filter = ...
; .compoundProps.iconCompoundGroup.conjunctionlist = /"type", /formtype/, ...|
; .compoundProps.iconCompoundGroup.defaultvalues = /"type", undefined, /formtype/ undef
;
; IconCompoundProperties = /iconCompoundGroup/, ...|
;
;
;
; .compoundProps.magicCompoundGroup.propertyfloat = /mySortProperty/
; .compoundProps.magicCompoundGroup.filter = ...
; .compoundProps.magicCompoundGroup.conjunctionlist = /"type", /formtype/, ...|
; .compoundProps.magicCompoundGroup.defaultvalues = /"type", undefined, /formtype/ unde
;
; magicCompoundProperties = /magicCompoundGroup/, ...|
;
```

His response was:

> **Quote**
>
> I think the one responsible for writing that was CDCooley, a pretty great scripter.

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 Like Reply

cdcooley Sent Yesterday, 11:40 AM

Excellent poster
Forum Member
7,4,7 posts
4,769 posts

Those are actually part of the original documentation included by the Skyui3 team, specifically by Schlangster.

What I learned I found by looking at the file and the Skyui3 source code. By the time I was interested in using the file Schlangster had already gone through and fixed it pretty well. Some of this you're probably already discovered, but I'll decribe 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 List.Layout there are a sets of variables that control the actual layout of the various columns of the list. These are also fairly easy to understand once you learn what them long enough but are important because they control the sorting order when you click on the column headers and also can even be used with 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's important to know 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, size, sort, 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.iconColumns.states". And "columns.iconColumns.states.0" 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, CASEINSENSITIVE 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 else from 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 opens). 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.states.label.text = 'DAMAGE'
; columns.itemNameColumn.states.entry.text = '@text'
; columns.itemNameColumn.states.sortAttributes = /equipping, favorite, text, formid/
; columns.itemNameColumn.states.sortOptions = /ASCENDING | NUMERIC, /ASCENDING | NUMERIC/
;
```

There are a number of magic string values you can use along with constants. Things like "DAMAGE" are translated into constants 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 Skyui3 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", "iteminons", "magicIcons", "itemCompoundProperties", and/or "magicCompoundProperties" to reference them. In the base file the only one used is "iteminons". 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 make labels, colors, and sorting work. The documentent filter can use any of the attributes.properties that you find in the "sortAttributes" lists from the the columns sections.

```
; props.clothingIcon.propertyfloat = /iconLabel/, /iconColor/, /subtypeDisplay/|
; .propIconGroup.filter = /formtype/ ...|
; .propIconGroup.dataMembers.filter = /formid/, /equipping/, /slot_torch/ | some wearable
; .propIconGroup.dataMembers.set = /iconLabel/ ...|/clothing_pouch/ ...|/iconColor/ ...|
; .propIconGroup.keywords.CBE_PortableContainer = /iconLabel/ ...|/clothing_pouch/ ...|icon
; .propIconGroup.keywords.ClothingRing = /iconLabel/ ...|/ring/ ...|/subtypeDisplay/ ...|
; .propIconGroup.keywords.ClothingNecklace = /iconLabel/ ...|/necklace/ ...|/subtypeDisplay/
; .propIconGroup.keywords.ClothingCirclet = /iconLabel/ ...|/armor_circlet/ ...|/subtypeDis
; .propIconGroup.keywords.ClothingBracelet = /iconLabel/ ...|/armor_circlet/ ...|/subtypeDis
;
; .propIconGroupOther.filter = /formtype/ ...|
; .propIconGroupOther.defaultvalues = /iconLabel/ ...|/iconColor/ ...|
;
; .propIconGroup.dataMembers.filter = /formtype/ ...|
; .propIconGroup.dataMembers.set = /iconLabel/ ...|/iconColor/ ...|
;
; .propIconGroup.keywords.VendorItemPotion = /iconLabel/ ...|/weapon_pickaxe/, /iconColor/ Red
; .propIconGroup.keywords.VendorItemCurrency = /iconColor/ Red|Gold|
;
; .propIconGroup.dataMembers.set = /iconColor/ ...|
; ; make all of the tools use a pickaxe icon
; .propIconGroup.keywords.VendorItemTool = /iconLabel/ ...|/weapon_pickaxe/, /iconColor/ Red
; .propIconGroup.keywords.VendorItemCutter = /iconColor/ Red|Gold|
;
; ; custom weapon icons colors mostly
; .propWeaponIconGroup.propertyfloat = /iconLabel/ ...|/iconColor/ ...|
; .propWeaponIconGroup.filter = /formtype/ ...|
; .propWeaponIconGroup.keywords.CBE_PortableCraftingStation = /iconLabel/ ...|/iconColor/
; .propWeaponIconGroup.dataMembers.set = /iconLabel/ ...|/iconColor/ ...|
; ; make this icons not as special
; .propWeaponIconGroup.keywords.DLC1WeaponPickaxe = /iconLabel/ ...|/weapon_pickaxe/ ...|/iconColor/
;
; ; some colors just because some needs its own "filter"
; .propWeaponIconGroupOtherLabel = /iconColor/ ...|
; .propWeaponIconGroupOtherLabel.filter = /formtype/ ...|
; .propWeaponIconGroupOtherLabel.keywords.CBE_PortableCraftingStation = /iconLabel/ ...|/iconColor/ ...|
;
; ; I like spell tomes to have a slightly different color
; .propBookIconGroupOtherLabel = /iconColor/ ...|
; .propBookIconGroupOtherLabel.filter = /formtype/ BOOK|
; .propBookIconGroupOtherLabel.filter = /subtypeDisplay/ "Spell Tome"|
; .propBookIconGroupOtherLabel.dataMembers.set = /iconColor/ ...|
;
; ; major change to colors for armor clothing and jewelry
; .propArmorIconGroupOtherLabel = /iconColor/ ...|
; .propArmorIconGroupOtherLabel.filter = /formtype/ Form_TYPE_ARMOR|
; .propArmorIconGroupOtherLabel.filter = /weightclass/ Armor_WEIGHT_LIGHT|
; .propArmorIconGroupOtherLabel.set = /iconColor/ Red#00FF00|
; .propArmorIconGroupOtherLabel.keywords.ArmorJewelry = /iconColor/ Red#ff00ff|
; .propArmorIconGroupOtherLabel.dataMembers.set = /iconColor/ Blu|Gold|Yel
; .propArmorIconGroupOtherLabel.filter = /weightclass/ Armor_WEIGHT_CLOTHING|
; .propArmorIconGroupOtherLabel.dataMembers.set = /iconColor/ Red|Gold|
; .propArmorIconGroupOtherLabel.filter = /weightclass/ Armor_WEIGHT_HEAVY|
; .propArmorIconGroupOtherLabel.dataMembers.set = /iconColor/ Blu#0000ff|
;
; ; tone down the colors for potions
; .propPotionIcon.propertyfloat = /iconLabel/ ...|
; .propPotionIcon.filter = /formtype/ Form_TYPE_POTION|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_HEALTH|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF0000|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_HEALTHRATE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF0000|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_HEALTHRATEALT|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF0000|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_MAGICKA|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#0000FF|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_MAGICKARATE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#0000FF|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_MAGICKARATEALT|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#0000FF|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_STAMINA|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF00ff|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_STAMINARATE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF00ff|
; .propPotionIcon.dataMembers.set = /iconLabel/ Item_POTION_STAMINARATEALT|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF00ff|
; .propPotionIcon.dataMembers.filter = /iconLabel/ Item_POTION_POISON|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF4a0|
; .propPotionIcon.dataMembers.filter = /subtypeDisplay/ Item_POTION_FIRDAMAGE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF4a0|
; .propPotionIcon.dataMembers.filter = /subtypeDisplay/ Item_POTION_ELECTRICDAMAGE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF4a0|
; .propPotionIcon.dataMembers.filter = /subtypeDisplay/ Item_POTION_FROSTDAMAGE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF4a0|
; .propPotionIcon.dataMembers.filter = /subtypeDisplay/ Item_POTION_RESTORE|
; .propPotionIcon.dataMembers.set = /iconColor/ Red#FF7f|
; .propPotionIcon.dataMembers.filter = /subtypeDisplay/ Item_POTION_FUDGE|
; .propPotionIcon.dataMembers.set = /iconColor/ Blu#A4A4A|
; .propPotionIcon.dataMembers.filter = /subtypeDisplay/ Item_POTION_DRINK|
; .propPotionIcon.dataMembers.set = /iconColor/ Blu#A4A4A|
;
; ; make sure edible ingredients look like food
; .propIconGroup.keywords.IngredientFood = /subtypeDisplay/|
; .propIconGroup.filter = /formtype/ Form_TYPE_INGREDIENT|
; .propIconGroup.keywords.VendorItemFood = /iconColor/ Red#FFFF00, /subtypeDisplay/ "Dri
; .propIconGroup.keywords.VendorItemFoodRaw = /iconColor/ Red#00000, /subtypeDisplay/ "Dr
;
; iconIcons = /propIconGroup/, /propIconGroupOther/, /propMagicGroup/, /propMagicGroupOt
;
```

And finally the Skyui3 source can be helpful in understanding how defaults are set.

https://github.com/s...rs/Data/Scripts/Source/User/Inventory/ItemCardBuilder.psc

https://github.com/s...rs/Data/Scripts/Source/User/Inventory/ItemCardFormatter.psc

https://github.com/s...rs/Data/Scripts/Source/User/Inventory/ItemCardFormatterBase.psc

Vilja is "Jon" a companion. But Dan is Jon of the saving a phoenix spouting fire and swooping down on a bumbleFed in "Jon" a food ... - SmartSquared

I didn't really set out to make a follower. I set out to make an Imps. ... - SmartSquared

Report Like Reply

CrowdAura Sent Yesterday, 12:00 PM

JoeSteerman
Member
31 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**
>
> These are actually part of the original documentation included by the Skyui3 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 Like Reply

cdcooley Sent Yesterday, 02:22 PM

Excellent poster
Forum Member
7,4,7 posts
4,769 posts

No, when I said documentation I just meant comments left in the file. While there are a few more comments in the mod's scripts 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.

Vilja is "Jon" a companion. But Dan is Jon of the saving a phoenix spouting fire and swooping down on a bumbleFed in "Jon" a food ... - SmartSquared

I didn't really set out to make a follower. I set out to make an Imps. ... - SmartSquared

Report Like Reply

CrowdAura Sent Today, 02:12 AM

JoeSteerman
Member
31 posts
Favourite Game: Skyrim

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

Report Like Reply

Archive Conversation

Fast Reply

Post Use Full Editor