

# zEBD: NPC Appearance Customizer

## Overview

This is a [zEdit](#) patcher for customizing NPC appearance. You can use this patcher to customize the meshes and textures of NPCs. The patcher is heavily inspired by MongoMonk's Everybody's Different Redone ([LE/SE](#)) Skyproc patcher. Differences will be covered below; the overarching purpose of building this patcher was to enable 1) easy randomization of texture packs for end users and 2) fine-grained customization of NPC appearance.

## Acknowledgements

This mod is inspired by, and in fact relies upon, MongoMonk's EBD Redone mod. The keywords, spell, and magic effect written by the zEdit patcher are his. zEdit is created by mator, to whom I'm thankful for a ton of tech support and patience in guiding me through making my first patcher (and in fact my first Javascript project). Also thanks to VictorF on the zEdit Discord channel for answering a huge amount of my questions as well as to Qudix and Shybert for additional troubleshooting help.

## Permissions

The contents of the patcher's "**zEBD assets\EBD Records**" are copies of MongoMonk's records (made with permission) and may not be redistributed. Asset packs themselves (meshes, textures, etc) do not come with this mod and may not be shared without the author's permission. However, asset pack *configuration* files (including and especially those derived from my templates in "**zEBD assets\Asset Pack Settings\Blank Templates**") may be shared without any explicit permission. As for the rest of my code and resource files, please notify me by PM to Piranha91 on Reddit if you would like to modify and reupload - if you don't hear back from me within 3 weeks, you can assume the mod is abandoned and you have permission to take it over as long as you acknowledge me on your new patcher's site.

## So, how is this different from Everybody's Different Redone?

There are five main differences between zEBD and Everybody's Different Redone. For clarity, in the explanation below, "assets" refer to meshes (.nif) and textures (.dds).

- 1) **zEBD is a zEdit patcher rather than a SkyProc patcher (duh).** While SkyProc is perfectly usable, zEdit is the currently thriving ecosystem to which most patchers are migrating, and thus serves as a useful hub for several great patchers.
- 2) **zEBD promotes facile configuration sharing.** In EBD, asset packs are configured by dragging and dropping the .nif and/or .dds files into the folders where the patcher expects them to reside. Learning how to do this isn't hard, but it is a bit daunting at first glance and can be tedious, especially if you mess up. Also, once you finally do the hard work, you can't redistribute the folders to other users unless you get permission from the asset authors. In zEBD, the asset locations are defined in **config (settings) files** that are freely sharable. To install an asset pack, follow a simple set of instructions to unzip the assets into a specific folder, drop the associated AssetPackSettings.json file into "zEBD\zEBD assets\Asset Pack Settings", and you're good to go.
- 3) **zEBD enables fine-grained customization.** You can let an NPC be as specific or as random as you like. You can specify some parts while leaving others random, or specify all parts. If you don't specify anything, the NPC will be modified completely at random (subject to the rules in the settings file assigned to it). These rules can be as simple as matching Nord assets to Nords and Elven assets to Elves, or as detailed as checking the NPC's stats/combat style/etc.
- 4) **The "parts" that zEBD can recognize are extensible using distributed config files.** This means it can be extended to support new parts (such as feet, for example). Parts are configurable within the json files included in the "zEBD assets\RecordTemplates" folder.
- 5) **zEBD lacks several features from the original EBD mod:** randomizing NPC height, changing animations to reflect the NPC's gender, and randomizing NPC headparts. I intend to implement these eventually. However, if you want these currently, a stopgap solution is to run EBD (disabling the option to change NPC meshes and textures), and then run zEBD on top of that.

# Using zEBD: For End Users

Installing asset packs and running the patcher is simple. I will use the popular [Tempered Skins for Males](#) mod as an illustrative example.

## Installing zEBD

- 1) Download and install zEdit
- 2) Download zEBD and extract it to zEdit\modules
- 3) Download and install Everybody's Different Redone ([LE/SE](#)) but do not run the patcher (zEBD requires EBD's scripts to function).

## Installing an Asset Pack

- 1) Download "Tempered Skins for Males - Dressed Version" from the link above but **do not** install it using your mod manager.
- 2) Download the "Config - Tempered Skins for Males - Dressed Version" from this mod's Nexus page.  
[Config files for other asset packs can be found on this mod page, or hosted as separate mods on the Nexus and other sites.](#)
- 3) **MO2:** Copy the "zEBD - Tempered Skins for Males - Dressed" folder (provided along with the config file) into "Mod Organizer 2\modules\" and activate it in MO2.  
**Vortex:** Someone please tell me how to do this in Vortex :)
- 4) Extract the downloaded Tempered Skins for Males (the actual meshes and textures downloaded from the Nexus) into zEBD - Tempered Skins for Males - Dressed\textures\TSM
- 5) (optional but highly recommended to save disk space): in textures\TSM, using the search bar in the top right corner of the explorer window, search for "nif". Wait for it to finish, hit ctrl+a to select all, and delete.
- 6) **Tempered Skins is a special case where you do actually need to install the mod through your mod manager** to get it to work right. This is because it provides malebody\_0 and malefeet\_0 meshes which the patcher cannot apply because they're not assignable to specific NPCs. After installing the mod, you can save disk space by deleting everything in its textures\actors folder BESIDES textures\actors\character\male\maleunderwear.dds and maleunderwear\_n.dds. [For most texture packs this step is not necessary.](#)
- 7) That's it!

## Running the Patcher (and Customizing Options)

- 1) Launch zEdit
- 2) Customize the config file rules, blocked NPCs, and blocked plugins if you want (if you want to just distribute all valid texture combinations to all NPCs, simply proceed to step 5).
  - a) Click on the zEBD options: (Ctrl+Shift+P or right click on plugin list -> manage patchers). Click on “zEBD NPC Customizer”. There may be a short delay as it validates config files.
  - b) Click on Show Asset Pack Configurations
  - c) For the config file that you want to customize, click “Display subgroups”. You will be presented with a list of top-level subgroups.

The screenshot shows the 'zEBD NPC Customizer' interface. At the top, there's a 'Group Name' field with 'Tempered Skins for Males' and a 'Save' button. Below it is a 'Gender' dropdown set to 'male' and a checked checkbox for 'Display alerts from this pack'. An 'Alert to show:' field contains 'Make sure that Tempered Sk'. A 'Display subgroups' checkbox is checked. Below this is a list of subgroups, each with a 'Details' button and a 'Remove' button. The subgroups are: HD: Head Diffuse (Details), HN: Head Normals (Details), HS: Head Subsurface (Details), HSP: Head Specular (Details), HDE: Head Detail (Details), BD: Body Diffuse (Details), BN: Body Normal (Details), BS: Body Subsurface (Details), BSP: Body Specular (Details), HAD: Hands Diffuse (Details), HAN: Hands Normal (Details), HAS: Hands Subsurface (Details), and HASP: Hands Specular (Details). At the bottom left is an 'Add New Subgroup' button.

- d) Click on the subgroup whose rules you want to customize. To understand what all the customization options mean, see [Config File Structure](#). For the purpose of this tutorial, let's do the following:
  - i) Tempered Skin's “Young” diffuse skin looks like it belongs to an NPC of age ~20, whereas most Skyrim NPCs sound older. Let's lock down this texture to just young-sounding NPCs.
    - (1) Click “Details” next to “HD: Head Diffuse”
    - (2) Within the “HD” options, click “Display subgroups”.
    - (3) Repeat for “HD.H: Non-Vampire”, HD.H.D1: D1 - Younger
    - (4) Now we're within the subgroup corresponding to the “young” diffuse texture.
    - (5) To make this subgroup accessible ONLY to young-sounding NPCs, click “Add New” next to “Allowed Attributes”. In the first box, enter “VTCK” and in the second enter “MaleYoungEager [VTYP:00013AD1]”. “VTCK” corresponds to the NPC's voice type, and “MaleYoungEager” is the corresponding value for young-sounding NPCs - you can find this information in xEdit.

(6) Step 5 alone means that non-young-sounding NPCs won't get the young texture; however, young-sounding NPCs CAN still get the other diffuse textures. To force young-sounding NPCs to receive the young texture, click "Add New" next to "Force if NPC Attributes have Values". Again, enter "VTCK" in the first box and "MaleYoungEager [VTYP:00013AD1]" in the second. The final configuration should look as follows:

HD.H.D1: D1 - Younger (Details) Remove

Subgroup ID: HD.H.D1

Subgroup Description: D1 - Younger

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight: 1

Allowed Races: Add New

Disallowed Races: Add New

Allowed Attributes: Add New

VTCK MaleYoungEager [VTYP:00013AD1]

Remove

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

VTCK MaleYoungEager [VTYP:00013AD1]

Remove

Require Subgroups: Add New

HSPD1 Remove

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

textures\TSM\D - Face diffus MaleHead.dds

Remove

Add New Subgroup

ii) "Warrior" type NPCs should always look shredded, but non-warrior NPCs should occasionally look ripped as well. Body muscle definition is controlled mainly by the body normal map texture (subgroup BN).

(1) Using the "Details" and "Display subgroups" buttons, navigate to BN\BN.Y\BN.Y.C1 (C1 - Ripped).

(2) To force warrior-type NPCs to use this normal map without restricting it to only warriors, add a new "Force if NPC Attributes have Values" containing ["CNAM\FULL", "Warrior"]:

BN.Y.C1: C1 - Ripped (Details) Remove

Subgroup ID: BN.Y.C1

Subgroup Description: C1 - Ripped

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight: 1

Allowed Races: Add New

Disallowed Races: Add New

Allowed Attributes: Add New

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

CNAM\FULL Warrior

Remove

Require Subgroups: Add New

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

textures\TSM\C - Body norm malebody\_1\_msn.dds

Remove

textures\TSM\C - Body norm malebody\_1\_msn\_feet.dds

Remove

Add New Subgroup

iii) When you're finished setting conditions, scroll back up to the top and **click "Save"** next to the edited group name (Tempered Skins for Males). Otherwise your edits will be discarded.

- 3) To block particular NPCs or plugins from being patched, open `zEBD\NPC Configuration\BlockList.json` in your favorite text editor (notepad, notepad++, etc).
- a) The block list is already populated with four Dawnguard NPCs whose rules confound this patcher (see [Miscellaneous Notes B](#) if you're curious as to why).
  - b) To block an NPC, simply add an entry in the same format (don't forget to add a comma after the previous entry!)

```
{
  "name": "Ronthil",
  "FormID": "02003367",
  "EDID": "DLC1Ronthil",
  "rootPlugin": "Dawnguard.esm"
}
```

You can provide any of the above parameters. I recommend always providing the root plugin - otherwise if your load order changes the FormID will no longer be valid. In fact, you can provide just the last six characters of the FormID as long as you provide the root plugin. If you don't provide one or more of the above four entries, the patcher will try to match the others. For example, if you just have "name": "Sven", then any NPC named "Sven" will be blocked from patching.

- c) To block all NPCs contained within a plugin from being patched, add its name (including extension) to the "blockedPlugins" in quotes. Separate multiple plugins by a comma.

```
"blockedPlugins":
  ["Bijin Warmaidens.esp", "Bijin Wives.esp", "Bijin NPCs.esp"]
```

- d) If you see any warnings or errors related to BlockPlugins.json, run the contents of your plugins through a [JSON Parser](#) to make sure you didn't forget any commas.

- 4) To force an NPC to use one or more subgroups, open `zEBD\NPC Configuration\ForceNPCList.json` in your favorite text editor (notepad, notepad++, etc). Within the brackets, add the NPC to configure. The syntax is the same as the block list, with two extra parameters:

```
{
  "EDID": "Ralof",
  "rootPlugin": "Skyrim.esm",
  "formID": "0002BF9D",
  "forcedAssetPack": "Tempered Skins for Males",
  "forcedSubgroups": ["HD.H.D1", "BD.B2", "BN.Y.C2"]
},
```

"forcedAssetPack" is the name of the config file that you want to apply to this NPC.

"forcedSubgroups" is the list of subgroups within that config file that you want to apply. Again, if you run into problems, run the contents of your plugins through a [JSON Parser](#) to verify your formatting.

- 5) Once you're done setting conditions, click on "Build Patches" in the top left corner. Find zEBD and click "Build".





- 6) Click “Show Log” to see updates on what the patcher is doing and any warnings or errors that it may show you. **Note: if you’re using the Bijin Skins config file provided by me, you’ll see warnings for quite a few female NPCs saying “No permutations satisfied criteria for NPC...”**. This is normal; Bijin Skins doesn’t provide head detail maps for Age40 and Age50 NPCs and the vanilla detail maps cause a neck/body color mismatch (you can see this for yourself by installing Bijin Skins through your mod manager and looking at Gerda in Dragonsreach). I therefore excluded Age40/Age50 NPCs from this config file.
- 7) Go grab a coffee and come back in ~5-10 minutes depending on the size of your load order.
- 8) Once the patcher finishes, close out of zEdit and save zEBD.esp when prompted.
- 9) To see a log of the permutations generated and which NPCs they were assigned to, open zEBD\Logs\PermutationsGenerated.txt. To find the NPC that you’re interested in, just search for their name, editorID, or formID:

```
2219: HD.H.D2,HN.YE2.N,HS,HSP.D2,HDE.BDM,BD.B3,BN.Y.C1,BS.H,BSP.B3,HAD.B3,HAN.Y,HAS,HASP.B3 (gender: male) from: Tempered Skins for Males
Distribution enabled: true (to unique NPCs: true), (to non-unique NPCs: true)
Permutation probability weighting: 1
Contents: HD (Head Diffuse), HD.H (Non-vampire), HD.H.D2 (D2 - Elder), HN (Head Normals), HN.YE2 (E2 - Harsh), HN.YE2.N (Nord), HS
Allowed Races: NordRace
disallowed Races: ElderRace,ElderRaceVampire
Required Subgroups: HSP.D2,3,HSP.D2,3,BSP.B3,8,HASP.B3,12,BSP.B3,8,BD.B3,5,HASP.B3,12
Filepaths:
    tsm\d - face diffuse options\d2 - elder\textures\actors\character\male\malehead.dds
    tsm\e - face normal options\e2 - harsh\textures\actors\character\male\malehead_msn.dds
    tsm\a - main files\a0 - required\textures\actors\character\male\malehead_sk.dds
    tsm\d - face diffuse options\d2 - elder\textures\actors\character\male\malehead_s.dds
    actors\character\male\blankdetailmap.dds
    tsm\b - body diffuse options\b3 - clean hairy\textures\actors\character\male\malebody_1.dds
    tsm\b - body diffuse options\b3 - clean hairy\textures\actors\character\male\malebody_1.dds
    tsm\c - body normal options\c1 - ripped\textures\actors\character\male\malebody_1_msn.dds
    tsm\c - body normal options\c1 - ripped\textures\actors\character\male\malebody_1_msn.dds
    tsm\a - main files\a0 - required\textures\actors\character\male\malebody_1_sk.dds
    tsm\a - main files\a0 - required\textures\actors\character\male\malebody_1_sk.dds
    tsm\b - body diffuse options\b3 - clean hairy\textures\actors\character\male\malebody_1_s.dds
    tsm\b - body diffuse options\b3 - clean hairy\textures\actors\character\male\malebody_1_s.dds
    tsm\b - body diffuse options\b3 - clean hairy\textures\actors\character\male\malehands_1.dds
    tsm\a - main files\a0 - required\textures\actors\character\male\malehands_1_msn.dds
    tsm\a - main files\a0 - required\textures\actors\character\male\malehands_1_sk.dds
    tsm\b - body diffuse options\b3 - clean hairy\textures\actors\character\male\malehands_1_s.dds

Generated Records:
    MaleBody13 [ARMO]: 520009C3
    MaleTorso7 [ARMA]: 520008D4
    SkinMaleTorso7 [TXST]: 5200082F
    MaleHands5 [ARMA]: 520008D5
    SkinMaleHands5 [TXST]: 52000830
    MaleFeet7 [ARMA]: 520008D6
    SkinMaleFeet7 [TXST]: 52000831
    SkinMaleHead60 [TXST]: 52000A3F

Assigned to the following NPCs:
    [Indolent Farmer|DBRecurringContact8Ivarstead|00087B93]
    [Hadvar|Hadvar|0002BF9F]
    [Lygrleid|DLC2RRLygrleidWindhelm|040376F2]
    [Drunk|RRPatron3_3DNPC|051FB7FF]
    [Eirikur|AAEMHalvdanWfriend11|0F1711D6]
    [Halvdan|AAEMHalvdan|0F006394]
```

# Height Configuration

As of version 1.3, zEBD offers a height configuration menu that appears as follows:

## Height Configuration Menu

Save Height Configuration

☒ Change NPC Record Height

☒ Change Race Record Height

Racial Height Presets: 

Apply

RaceBretonRace

Distribution Modeuniform

Base Male Height1.000000

Male Height +/-0.000000

Base Female Height0.950000

Female Height +/-0.000000

RaceBretonRaceVam

Distribution Modeuniform

Base Male Height1.000000

Male Height +/-0.000000

Base Female Height0.950000

Female Height +/-0.000000

...

Add New Race Height Configuration

Get Heights from Load Order

Global Distribution Modeuniform 

Apply To All

Height +/- 0.020000 

Apply To All

There are two variables that control the NPC's height. There is a "height" element within NPC records, and a DATA\Male Height & DATA\Female Height element in each RACE record. It appears that the final NPC height is the product of the value in the NPC record and that in the RACE record. zEBD therefore lets you set both values. You can choose to override one or both of these settings with those configured in zEBD using the checkboxes at the top of the menu. The race record height will simply be overwritten to match the zEBD value, while the NPC record height will be randomized.

Each entry in the height config menu corresponds to the Editor ID of a given race. Below each Editor ID are boxes controlling the male and female heights of that race. You can set them manually, or load from several presets available from the *Racial Height Presets* dropdown menu (don't forget to click Apply after selecting).

Below the Base Heights are the controls for per-NPC randomization. You can set the probability distribution function (uniform vs. bell curve) and the range between which NPCs can vary. If the range is set to 0.02, the height value set in the NPC's record will be set to between 0.98 and 1.02. This will be multiplied with the base racial height. If using bell curve distribution, the range is interpreted as 3 standard deviations and any NPCs above or below the range will be clipped to those boundaries.

At the bottom you can add settings for a new race (for supporting custom races). You can also get the current race heights from your load order (but you'll need to add the race and set its Editor ID appropriately before zEBD will recognize and populate it). This can be useful if you want to examine and tweak the heights set by another mod.

Below those options are "apply to all" buttons to quickly apply the given randomization options to all of the above configurations.

At the top, there is a "Save Height Configuration" button. Make sure to save your changes after customizing.



# Using zEBD: For Asset Pack Settings Creators

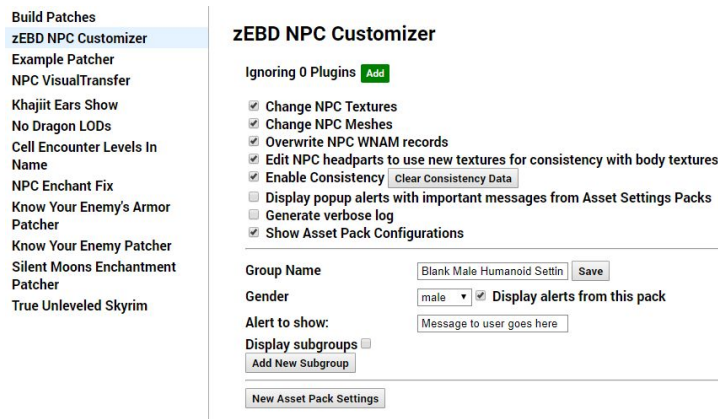
## Tutorial

So you want to put in your blood, sweat, and tears so that others can download your config file and use it with naught but a simple click of “Build Patch”? Here’s how it’s done (thanks to the blank config files provided, it’s less painful than it looks). Let’s create the configuration settings for the “Tempered Skins for Males” that we used in the “Installing an Asset Pack” tutorial.

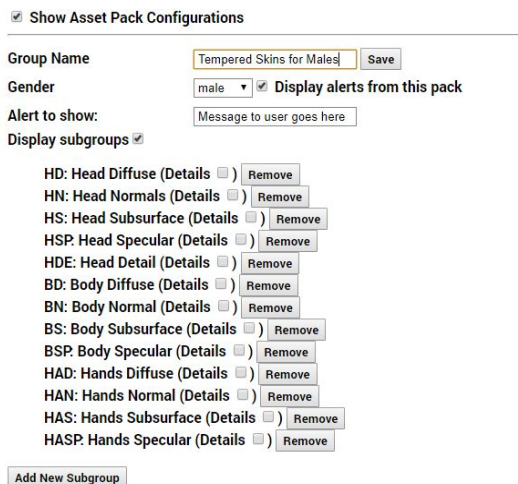
- 1) Download “Tempered Skins for Males - Dressed Version” from the link above but **do not** install it using your mod manager.
- 2) **MO2:** Copy `zEBD\Asset Folder Template` into `MO2\mods`  
**Vortex:** Someone please tell me how to do this in Vortex :)
- 3) Rename the copied “`Asset Folder Template`” to “`zEBD - Tempered Skins for Males - Dressed`”.
- 4) Within the above folder, rename “`meshes\EBD - YourMod`” to “`meshes\TSM`” and “`textures\EBD - YourMod`” to “`textures\TSM`”. “TSM” is just a convenient folder label - you can make it whatever you want. Within our config file, all mesh paths will start with “`meshes\TSM\...`” and texture paths will start with “`textures\TSM\...`”.
- 5) Set aside a copy of the new “`zEBD - Tempered Skins for Males - Dressed`”. You will include this empty set of folders along with the config file itself for distribution, and you don’t want to accidentally share any assets themselves.
- 6) Extract the downloaded Tempered Skins for Males (the actual meshes and textures downloaded from the Nexus) into `textures\TSM`. Activate `zEBD - Tempered Skins for Males - Dressed` in MO2/Vortex.
- 7) (optional but highly recommended to save disk space): in `textures\TSM`, using the search bar in the top right corner of the explorer window, search for “nif”. Wait for it to finish, hit ctrl+a to select all, and delete.
- 8) Copy “`zEBD\zEBD assets\Asset Pack Settings\Blank Templates\Blank Male Humanoid Settings.json`” into “`zEBD\zEBD assets\Asset Pack Settings\`” and rename it to “`Tempered Skins for Males.json`”.
- 9) If you have not yet read “**Config File Structure**” (below), please do so now. *Note: zEBD makes no assumptions about the structure of your subgroups. When I started making config files, I followed the folder structure of the downloaded archive. However, I quickly realized that*

*makes it almost impossible to keep track of which assets I had dealt with and which still needed to be assigned, so instead I adopted a strategy of making one top-level subgroup for each path that needs to be assigned. I found this to be a much better way to go about it, so that's what I will demonstrate in this tutorial.*

- 10) Time to edit the config file. Launch zEdit, click on the zEBD settings, and tick the “Show Asset Pack Configurations” box. **Note:** You can also create and edit the .json files directly if you feel more comfortable doing so. My strategy has been to set up the subgroups within the zEBD GUI and then fill in the details in a text editor. The problem with the GUI is that if you accidentally click outside of it in zEdit it'll disappear and discard all of your changes since the last save. For the sake of this tutorial I will just show how to use the GUI.



- 11) Change “Group Name” to “Tempered Skins for Males” and tick the “Display subgroups” box. You can see that the blank template was pre-populated with all of the top-level subgroups that you will need to assign.



- 12) If you want to exclude certain types of NPCs from being patched with any subgroup within this config file, it's good practice to put these exclusions into the first subgroup, because all top-level subgroup conditions are carried through into the final permutation. For example, Bijin Skins doesn't provide Age40 or Age50 detail maps. Without them, the patcher tries to use the vanilla detail maps, but this causes a noticeable neck seam. Therefore, if you look at my Bijin Skin settings, I excluded all “Age40” and “Age50” NPCs from the “HD” subgroup, which

effectively excludes them from the entire config file. Tempered Skins includes pretty much every male texture so this is not necessary here.

- 13) For each top-level subgroup, look through the downloaded archive for the relevant path. Let's start with Head Diffuse, which should correlate to malehead.dds and malehead\_vampire.dds. Searching for malehead.dds in textures\TSM, we see that it's provided in four locations:

textures\TSM\D - Face diffuse options\D1 - Younger\textures\actors\character\male
textures\TSM\D - Face diffuse options\D2 - Elder\textures\actors\character\male
textures\TSM\D - Face diffuse options\D3 - Hard Life\textures\actors\character\male
textures\TSM\D - Face diffuse options\D4 - Stubbled faces by Geonox\textures\actors\character\male

Meanwhile, maleheadvampire.dds appears only once, in: textures\TSM\A - Main files\A0 - Required\textures\actors\character\male. There is also a separate texture for Snow Elf and Afflicted.

- 14) Check the "Details" box next to "HD", and then check "HD"'s "Display Subgroups" box. You can see that the blank template has already prepared non-vampire and vampire subgroups for you with the appropriate conditions set.

HD: Head Diffuse (Details ☒)

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races:

Disallowed Races:

Allowed Attributes:

Disallowed Attributes:

Force if NPC Attributes have Values:

Require Subgroups:

Exclude Subgroups:

Add Keywords to NPC Record:

Asset Paths:

Display subgroups ☒

HD.H: Non-vampire (Details ☐)

HD.V: Vampire (Details ☐)

- 15) Check "Details" for "HD.H: Non-vampire". Since we are replacing this texture, click "Remove" next to the default path in "Asset Paths".

Asset Paths:

- 16) Since we have four non-vampire malehead.dds texture to assign, click the "Add New Subgroup" button four times and then check "Display subgroups" (note: "Display subgroups only appears if the subgroup has subgroups, so you won't see it until you click "Add New Subgroup").

HD.H: Non-vampire (Details ☒ )

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races:

Disallowed Races:

Allowed Attributes:

Disallowed Attributes:


Force if NPC Attributes have Values:

Require Subgroups:

Exclude Subgroups:

Add Keywords to NPC Record:

Asset Paths:

Display subgroups ☐ 

You should see four blank subgroups appear.

Display subgroups ☒

defaultId: Default Name (Details ☐ )

defaultId: Default Name (Details ☐ )

defaultId: Default Name (Details ☐ )

defaultId: Default Name (Details ☐ )

- 17) Check the “Details” box next to the first one. Change the subgroup ID and description to match the texture we’re assigning.

Display subgroups ☒

HD.H.D1: D1 - Younger (Details ☒ )

Subgroup ID:

Subgroup Description:

- 18) Now we need to add the path. Next to “Asset Paths”, click “Add New”. You will see two blank boxes appear. The first one should be filled with the filepath relative to the top-level mod folder:  
 textures\TSM\D - Face diffuse options\D1 - Younger\textures\actors\character\male\malehead.dds  
 The second box is the filename of the corresponding path in the corresponding recordTemplate (If you’re not familiar with Record Templates yet, read the [Record Template Structure](#) section). Go to zEBD\zEBD assets\RecordTemplates\ . The texture we’re assigning is for male humanoid characters, so open StandardTemplates\_Male.json in a text editor such as NotePad or NotePad++. Scrolling down to the "SkinMaleHead" record, we see that the diffuse texture is "TX00 - Difuse": "Actors\Character\Male\MaleHead.dds". Therefore, in the second box we write the filename to replace: “MaleHead.dds”

Asset Paths:

- 19) Since this texture is for humanoid non-vampire races, we could set allowedRaces to “humanoid\_NonVamp” (from the Group Definition “humanoid”; see [Group Definition Structure](#) section if you’re unfamiliar). However, notice that the parent subgroup (HD.H:

Non-vampire) already has that parameter set. Since all child subgroups will inherit their parents' restrictions, we don't need to set it again.

Also, looking at the preview pictures for this texture, it looks like it corresponds to NPCs in their early 20s. To reflect this in-game, let's add an AllowedAttribute of VTCK, MaleYoungEager [VTYP:00013AD1] to restrict this texture to NPCs that have young-sounding voices (see [Attribute Searching](#)).

Display subgroups ☒

HD.H.D1: D1 - Younger (Details ☒ Remove)

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races:

Disallowed Races:

Allowed Attributes:

Disallowed Attributes:

Force if NPC Attributes have Values:

Require Subgroups:

Exclude Subgroups:

Add Keywords to NPC Record:

Asset Paths:

Congrats! You've assigned your first texture and it only took 19 steps. **Press "Save" next to "Group Name" to save your work!**

20) Repeat for all of the other asset types within the asset pack. I won't show what I did for every one, but I will highlight some important details.

- In the blank template, the default file paths begin with "Skyrim.esm". If the mod author packaged the assets in a folder structure corresponding to Skyrim's data path (as almost all of them do), you can quickly change these without worrying about the full path by simply replacing "Skyrim.esm" with the partial directory of your asset:

Asset Paths:

IPC Appearance Merge\mods\zEBD - Tempered Skins for Males - Dressed\textures\TSM\IE - Face normal options\E1 - Smooth\textures\actors\character\darkelfmale

malehead\_msn.d

Becomes:

Asset Paths:

- For the HD.V (vampire diffuse textures), note that while the new file name is malehead\_vampire.dds, the destination filename in the Record Template remains



## “MaleHead.dds”

HD.V: Vampire (Details ☒ ) Remove

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races: Add New

Remove

Disallowed Races: Add New

Allowed Attributes: Add New

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

Require Subgroups: Add New

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

Remove

Add New Subgroup

- c) In the blank asset pack templates, the normal map subgroups HN and BN all contain the expected type-specific subgroups that are ready to be filled out:

Display subgroups ☒

HN.Y: Younger (Details ☒ ) Remove

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races: Add New

Disallowed Races: Add New

Remove

Allowed Attributes: Add New

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

Require Subgroups: Add New

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

Add New Subgroup

Display subgroups ☒

HN.Y.B: Breton (Details ☒ ) Remove

HN.Y.D: Dark Elf (Details ☒ ) Remove

HN.Y.N: Nord (Details ☒ ) Remove

HN.Y.H: High Elf (Details ☒ ) Remove

HN.Y.O: Orc (Details ☒ ) Remove

HN.Y.I: Imperial (Details ☒ ) Remove

HN.Y.R: Redguard (Details ☒ ) Remove

HN.Y.V: Vampire (Details ☒ ) Remove

HN.Y.W: Wood Elf (Details ☒ ) Remove

HN.E: Elder (Details ☒ ) Remove

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races: Add New

Remove

Disallowed Races: Add New

Allowed Attributes: Add New

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

Require Subgroups: Add New

Remove

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

Remove

Add New Subgroup

- d) The “head detail” subgroup comes pre-filled with conditions to make sure that NPCs receive age-appropriate face textures. You just need to point the paths to the correct textures to replace blankdetailmap.dds and the other detail map variants. For example, the asset path for the Age40 subgroup should point to:  
 [“textures\intervening\_path\maleheaddetail\_age40.dds”, “blankdetailmap.dds”].  
 See the image on the following page:

HDE: Head Detail (Details ☒) Remove

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races: Add New

Disallowed Races: Add New

Allowed Attributes: Add New

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

Require Subgroups: Add New

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

Add New Subgroup

Display subgroups ☒

HDE.BDM: BlankDetailMap (Details ☐) Remove

HDE.A40: Age 40 (Details ☒) Remove

Subgroup ID:

Subgroup Description:

Enabled: ☒ Distribute to Random NPCs: ☒

Allow Unique NPCs: ☒ Allow Non-Unique NPCs: ☒

Distribution Probability Weight:

Allowed Races: Add New

Disallowed Races: Add New

Allowed Attributes: Add New

FTST	on_Age40 [TXST:000F2058]	<span>Remove</span>
FTST	SkinHeadMaleDarkElfCompl	<span>Remove</span>
FTST	SkinHeadMaleHighElfCompl	<span>Remove</span>
FTST	SkinHeadMaleImperialComp	<span>Remove</span>
FTST	SkinHeadMaleOrcComplexic	<span>Remove</span>
FTST	SkinHeadMaleRedguardCon	<span>Remove</span>
FTST	SkinHeadMaleWoodElfComf	<span>Remove</span>
FTST	SkinHeadMaleNordComplex	<span>Remove</span>

Disallowed Attributes: Add New

Force if NPC Attributes have Values: Add New

FTST	SkinHeadMaleBretonComple	<span>Remove</span>
FTST	SkinHeadMaleDarkElfCompl	<span>Remove</span>
FTST	SkinHeadMaleHighElfCompl	<span>Remove</span>
FTST	SkinHeadMaleImperialComp	<span>Remove</span>
FTST	SkinHeadMaleOrcComplexic	<span>Remove</span>
FTST	SkinHeadMaleRedguardCon	<span>Remove</span>
FTST	SkinHeadMaleWoodElfComf	<span>Remove</span>
FTST	SkinHeadMaleNordComplex	<span>Remove</span>

Require Subgroups: Add New

Exclude Subgroups: Add New

Add Keywords to NPC Record: Add New

Asset Paths: Add New

Skyrim.esm\Actors\Characte	blankdetailmap.dds	<span>Remove</span>
----------------------------	--------------------	---------------------

**A note about feet:** The default feet texture paths in Skyrim.esm are the same as those of the body texture paths. However, some mods assign different textures to the feet. To make sure that feet can be uniquely assigned, the zEBD standard templates point to unique paths for them (e.g. `“actors\\character\\male\\malebody_1_feet.dds”` rather than what the vanilla records use `“actors\\character\\male\\malebody_1.dds”`). If you’re making a texture pack that contains some, but not all, feet textures (diffuse, normals, subsurface, specular), make sure that you revert the unassigned one back to its vanilla state. For example, if your texture pack provides only the diffuse feet textures, add the following path for normals: `[“Skyrim.esm\\actors\\character\\male\\malebody_1_msn.dds”, “malebody_1_msn_feet.dds”]`. Repeat analogously for the `_sk` and `_sk` feet textures.

## Debugging Your Config File

In addition to the PermutationsGenerated.txt log generated in zEBD\Logs, as of zEBD version 1.3, a developer controls file is offered to assist in debugging. This file contains several options that may be of use. zEBD\zEBD assets\MiscConfig\DevControls.json offers the following options:

- permutationBuildupLogger: Setting to “true” will generate a log of how the different permutations are built up from their constituent subgroups. Useful for config file creators - if a subgroup “drops out”, this log will show you at which stage that happens. Log is located at zEBD\Logs\PermutationBuildupLog.txt
- verboseLogger: Verbose mode technically exists, but the functionality was severely gutted in the few days before Beta 1 release so it was disabled. DevControls allows you to re enable it, but it is currently not very informative. VerboseMode may be updated in a future release but is currently not useful.

In addition, the following options can expedite testing of custom assignments by bypassing the need to generate records from config files:

- savePermutations: Setting to “true” will cause the logger to save permutations to a set of json files next time the patcher is run. These files are:
  - zEBD\NPC Configuration\GeneratedPermutations.json
  - zEBD\NPC Configuration\GeneratedRecords.json
  - zEBD\NPC Configuration\GeneratedRecordsMaxPriority.json
  - zEBD\NPC Configuration\GeneratedLinkageList.json
- loadPermutations: If set to true, the patcher will look for the above four json files and load permutations and records from those files rather than generating them *de novo* from the config files. This speeds up the patcher time (significantly if you have a lot of config files installed). However, any changes made to the config files will be ignored (since it’s loading only from the json files), so this is useful mainly for playing with the block list and forced NPC assignment list.

# Config File Structure

Each config file consists of “General” settings and “Subgroups”. Each subgroup has its own subgroup settings. In the example below I am showing the JSON file in a text editor, but all settings can be altered directly from within zEdit as well (in the zEBD configuration menu).

## *General Settings:*

```
"groupName": "Blank Male Humanoid Settings",  
  "gender": "male",  
  "displayAlerts": true,  
  "userAlert": "Message to user goes here",
```

groupName: *A name for this config file.* Should be similar to the asset pack for which you’re making the config.

gender: *either “male” or “female”.* Reflects the gender of NPCs to which this config file can be applied.

displayAlerts: *true or false.* If true, the userAlert will be displayed as a popup when the user opens zEdit’s settings menu. The user can disable this once they’ve seen the message.

Subsequently, the config file contains subgroups. Each subgroup contains variants of the assets to be assigned to the given NPC. Top level subgroups will be concatenated together to make the final permutation that can be assigned to an NPC. In addition to the file paths of the assets themselves, the subgroups contain rules about which NPCs can use the given assets. Subgroups can have their own nested subgroups to arbitrary depth (don’t worry, there will be an illustrated example below - see *Generating Permutations from Subgroups*).

```
"subgroups": [  
  {  
    "id": "HD",  
    "enabled": true,  
    "distributionEnabled": true,  
    "allowedRaces": [],  
    "disallowedRaces": [],  
    "allowedAttributes": [],  
    "disallowedAttributes": [],  
    "forceIfAttributes": [],  
    "allowUnique": true,  
    "allowNonUnique": true,  
    "name": "Head Diffuse",  
    "requiredSubgroups": [],
```

```

    "excludedSubgroups": [],
    "addKeywords": [],
    "probabilityWeighting": 1,
    "paths": [],
    "subgroups": []
}

```

id: *Every subgroup gets its own unique id.* This can be whatever you want, but try to keep it short and reasonable, such as “HD” for the subgroup responsible for providing the head diffuse texture. Subgroup ids do not need to include their parent IDs (the way I structured mine in the tutorial) but it’s helpful to keep track of inheritance. **IMPORTANT: Make sure each id in your config file is unique!**

enabled: *true or false.* If false, this subgroup will be ignored. This is for the end user to disable variants that they don’t want. If you find yourself setting subgroups in your config file to “enabled: false” as you’re making it, please consider just creating a separate config for the disabled variants. For example, Bijin Skins provides either default or black subsurface textures. The end user probably won’t want both, so rather than setting one to enabled and the other to disabled, just make two different config files (Bijin Skins - Subsurface Default and Bijin Skins - Subsurface Black). **IMPORTANT: If you set a top-level subgroup to “enabled = false”, the entire config file will be disabled.**

distributionEnabled: *true or false.* If false, the subgroup will not be assigned to NPCs at random unless the NPC is also a match for its forceIfAttributes (see below). This can be used to restrict certain assets to appear only on certain NPCs (see the “HDE” subgroup within any of my provided config files as examples). This setting is overridden by forceIfAttributes and user-specified assignments in `zEBD\NPC Configuration\ForceNPCList.json`.

allowedRaces: *Races to which the given subgroup can be applied.* This can include both EditorIDs (e.g. BretonRace) or groups of EditorIDs defined in `zEBD\zEBD assets\RestrictionDefs\*.json`. Within that folder, zEBD comes with a `GroupDefs.json` file which contains useful groups. If your subgroup requires a grouping which isn’t present in this file, make a new .json file in the same format as `GroupDefs.json` and instruct your users to drop it into the same folder. zEBD will load all group definitions within the RestrictionDefs folder. If allowedRaces is left empty, the subgroup can be applied to all races. *Nothing, including `zEBD\NPC Configuration\ForceNPCList.json`, overrides allowedRaces.*

disallowedRaces: *Races to which the given subgroup cannot be applied.* If an entry appears in both allowedRaces and disallowedRaces, it will be removed from allowedRaces at runtime (the config file will not be altered). This means that you can set allowedRaces to [“humanoid\_NonVamp”] and disallowedRaces to [“elven”] to get a subgroup that can only be applied to Nords, Bretons, Redguards, Imperials, Orcs, and Elders. *Nothing, including `zEBD\NPC Configuration\ForceNPCList.json`, overrides disallowedRaces.*



allowedAttributes: *Element values that the NPC must have for the subgroup to be applied.* This is an array of length 2, where array[0] is the element path and array[1] is the element value. For example, adding an allowedAttribute of ["VTCK", "MaleYoungEager [VTYP:00013AD1]"] will allow this subgroup to be added only if the NPC has the MaleYoungEager voice type. Attributes can be queried through arrays and layers of records (see the **Attribute Searching** section). Only one attribute from each subgroup must be satisfied by the NPC to permit the permutation to be applied (e.g. *OR logic within the subgroup and AND logic between subgroups*).

disallowedAttributes: *Element values that the NPC may not have for the subgroup to be applied.* If the provided value at array[1] is found at path array[0] relative to the NPC being patched, the patcher will skip this subgroup and move on to the next one. Has priority over allowedAttributes.

forcelfAttributes: *Element values that will force the patcher to apply the given subgroup.* If the value at array[1] at path array[0] is found, the patcher will apply this subgroup in preference to others. This overrides distributionEnabled. For a usage example, see the "HDE" subgroup within any of my provided config files. Important behaviors:

- Only one forcelfAttribute from each subgroup must be satisfied by the NPC to permit the permutation to be applied (e.g. *OR logic within subgroups*).
- In cases where permutation A has one subgroup with forcelfAttributes satisfied by the given NPC while permutation B has two such subgroups, permutation B will win because it is more specific.
- forcelf attributes are not first-come, first-served. The list of all generated permutations is first filtered such that only permutations with forcelfAttributes that apply to the given NPC remain. The list is further filtered to maximize the forcelf specificity (see the above paragraph). From this filtered list, a permutation is drawn at random. Therefore, it is possible for a permutation to satisfy forcelf criteria and still not get selected, although this should be a rare event.

allowUnique: *true or false.* If false, NPCs with `ACBS - Configuration\Flags\Unique` (unique NPCs) will not receive this subgroup. Overrides everything besides user-specified assignments in `zEBD\NPC Configuration\ForceNPCList.json`.

allowNonUnique: *true or false.* If false, NPCs *without* `ACBS - Configuration\Flags\Unique` (non-unique NPCs) will not receive this subgroup. Overrides everything besides user-specified assignments in `zEBD\NPC Configuration\ForceNPCList.json`.

Name: *A short description of this subgroup.* For legibility purposes only.

requiredSubgroups: *Array of other subgroups (list only the id) that must be applied for this subgroup to be applied.* Useful, for example, to link "elder" face normals and body normals if present in different subgroups.

excludedSubgroups: *Array of other subgroups (list only the id) that may not be applied if this subgroup is applied.*

addKeywords: Any text string added here will be converted into a keyword object and applied to the NPC that receives the given subgroup. Potentially useful for integration with Dynamic Animation Replacer, though I haven't tried it yet.

probabilityWeighting: The relative probability of an NPC receiving this subgroup. Set to 1 by default. Note that this is multiplicative: If a permutation containing subgroupA and subgroupB is generated, and subgroupA has probabilityWeighting = 5 and subgroupB has probabilityWeighting = 10, the permutation will be  $5 \times 10 = 50$  times more likely to be assigned than another permutation where the probabilityWeighting of all subgroups is 1.

paths: array containing the asset paths for this subgroup. Paths are mapped onto record objects found in `zEBD\zEBD assets\RecordTemplates\*.json` (see [Record Template Structure](#)). Array[0] is the path of the new asset (wherever you as the config file author instruct the user to extract the assets), and Array[1] is the file name to be matched within the RecordTemplate. Paths must be prefaced with their asset type ("textures" or "meshes"). Example:

```
"paths": [
  [
    "textures\\CoverKhajiits -
Male\\khajiitmale\\Data\\Textures\\actors\\character\\khajiitmale\\khajiitmalehead_s.
dds",
    "khajiitmalehead_s.dds"
  ]
]
```

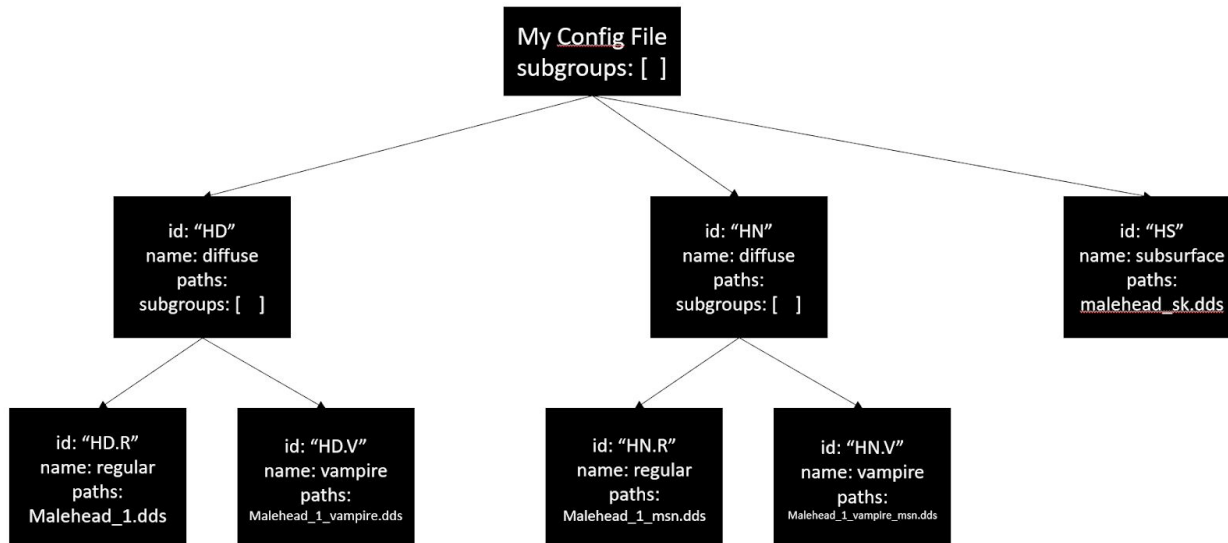
Remember, the goal here is to make it as easy as possible for the user to extract assets and run the patcher, which might mean you generate obnoxious file paths such as the one above.

Subgroups: an array of other subgroups nested within the current one (see [Generating permutations from Subgroups](#)).

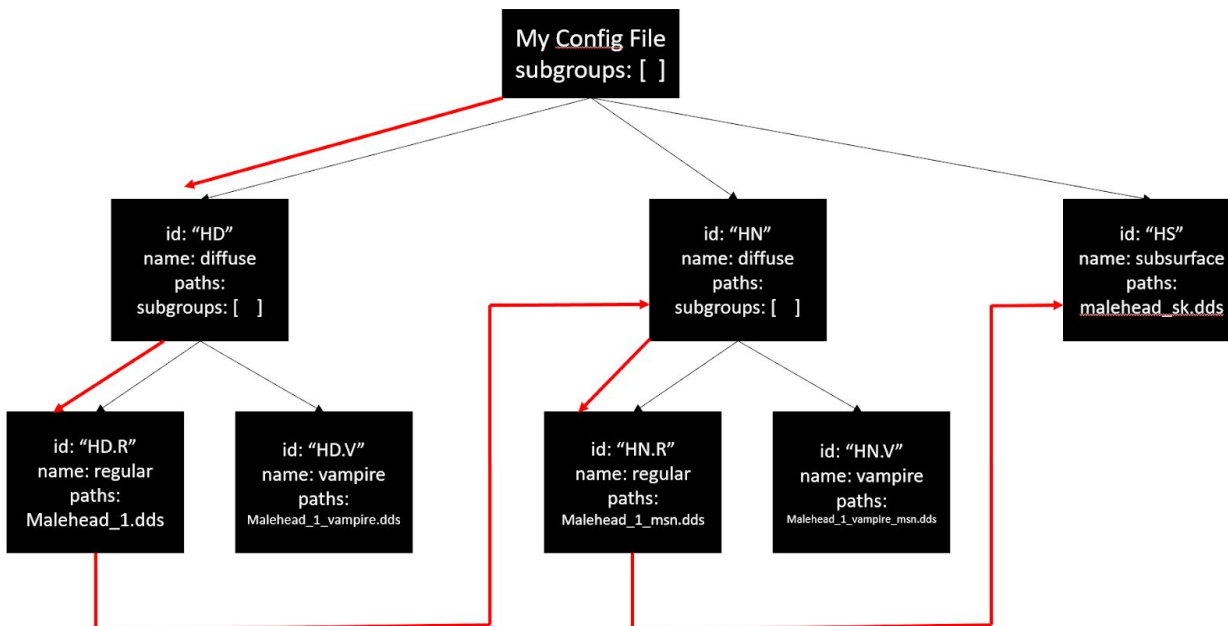
# Generating Permutations from Subgroups

To generate a permutation, the patcher traverses to the bottom of the available nested subgroups for each top-level subgroup and combines the top-level variants together. TL;DR: Top-level subgroups are combined together after choosing a single subgroup from lower-level subgroups.

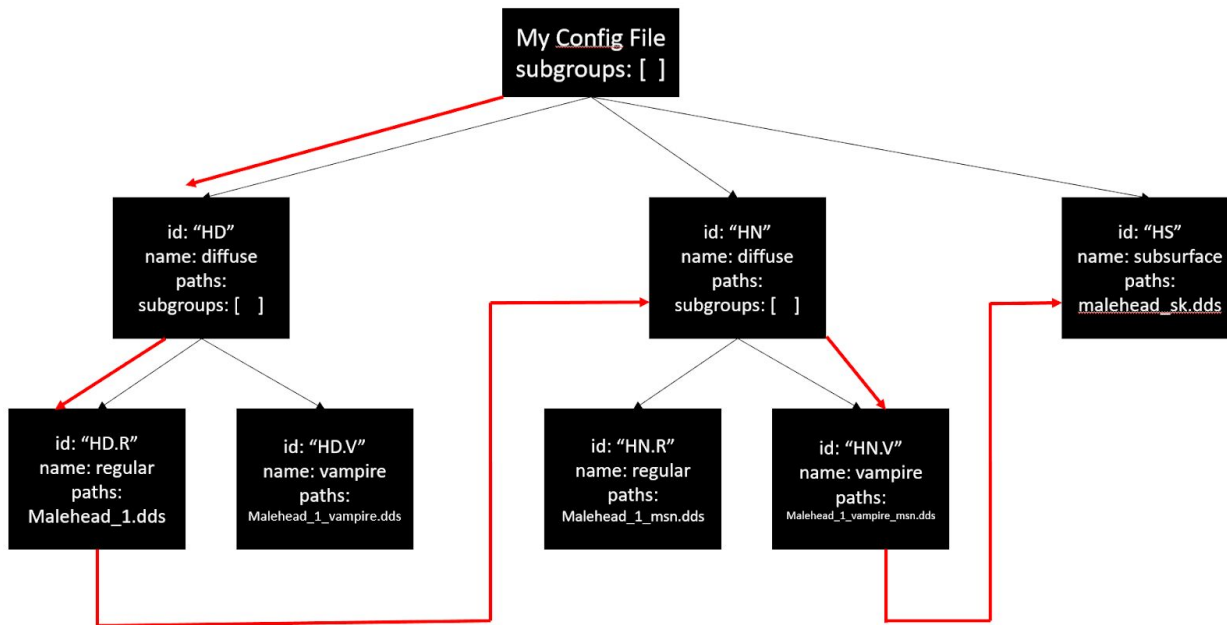
For example, say you have set up a config file as follows:



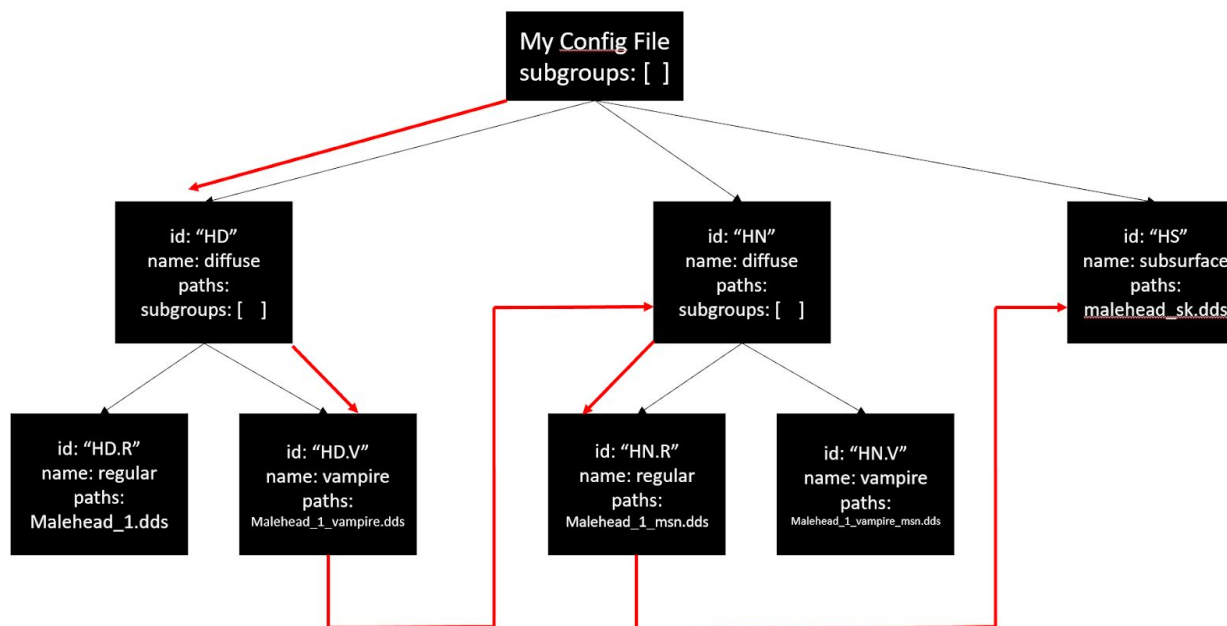
The patcher will first generate Permutation 1, containing the subgroups HD.R, HN.R, and HS. Even though “HD” and “HN” contain no paths, their associated rules (allowedAttributes, etc) will be carried forward into their child subgroups and then into the permutation itself.



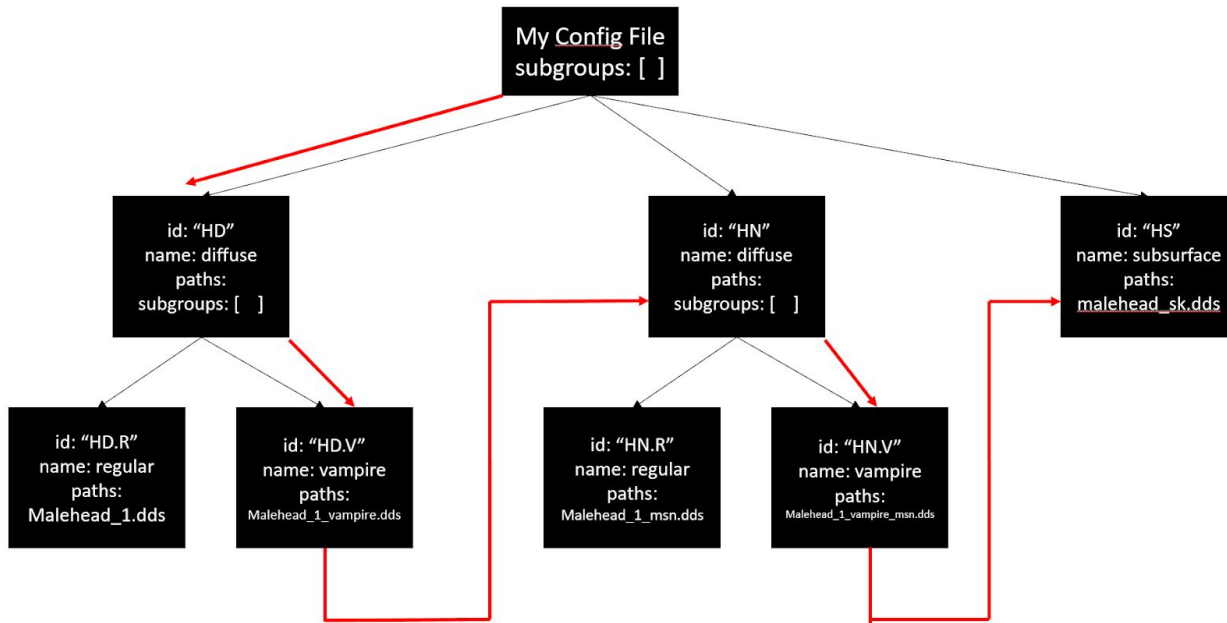
The patcher will then generate Permutation 2: HD.R, HN.V, HS:



You probably don't want regular diffuse textures being combined into a permutation with vampire normal maps. To avoid this, you would set `HD.R.requiredSubgroups` to `["HN.R"]` and `HN.R.requiredSubgroups` to `["HD.R"]`, and then link the vampire variants similarly. If this is done, the patcher will immediately discard this permutation and move on to Permutation 3: HD.V, HN.R, HS:



If the linkages are set appropriately this one should also be discarded, leaving the patcher only to generate Permutation 4: HD.V, HN.V, HS:



This is of course a simplified example with only four possible permutations, but the same algorithm is applied no matter how many subgroup variants you have. **The Permutation (combination of top-level subgroups) is what eventually gets applied to the NPC.**



## Attribute Searching

allowedAttributes, disallowedAttributes, and forcelfAttributes make use of the record search function. This function searches for a path relative to an NPC record and returns the value at that path. For example, if a patcher is assigning a subgroup to Sven, and the subgroup has allowedAttributes = `[["VTCK", "MaleYoungEager [VTYP:00013AD1]"]]`, the search function will look at Sven's VTCK record.

0001328F	Jorn	Jorn	Faction	BardSingerFaction [FACT:000CF8F9]
00013295	AtaAlan	Ataf	Rank	-1
0001347F	Sven	Sven	INAM - Death item	
0001361A	Karita	Karita	VTCK - Voice	MaleYoungEager [VTYP:00013AD1]
000136BC	Lynly	Lynly Star...	TPLT - Template	
00072663	Delacourt	Delacourt	RNAM - Race	NordRace "Nord" [RACE:00013746]
00083D99	TalsgarTheWanderer	Talsgar the ...	SPCT - Count	
03018DE0	BYOHHouse1Bard	Llewellyn t...	Actor Effects	
03019630	BYOHHouse2Bard	Sonir	Destructible	
03019631	BYOHHouse3Bard	Oriella	WNAM - Worn Armor	

Since Sven has that voice type, the subgroup will be applied. If Sven did not have that voice type, the patcher would move on to another subgroup.

The search function can also search through nested values. You could, for example, provide the following attributes: `[["AIDT\Confidence", "Brave"]]`.

0001347F	Sven	Sven	COED - Extra Data	
0001361A	Karita	Karita	AIDT - AI Data	
000136BC	Lynly	Lynly Star...	Aggression	Unaggressive
00072663	Delacourt	Delacourt	Confidence	Brave

You can also search through arrays. If you know the array index to search, you can provide it as follows: `[["Head Parts\0]", "HairMaleNord03 "HairMaleNord03" [HDPT:000514B8]"]]`.

00013295	AtaAlan	Ataf	Head Parts (sorted)	
0001347F	Sven	Sven	PNAM - Head Part	HairMaleNord03 "HairMaleNord03" [HDPT:000514B8]
0001361A	Karita	Karita	PNAM - Head Part	BrowsMaleHumanoid01 "BrowsMaleHumanoid01" [HDPT:0005...
000136BC	Lynly	Lynly Star...	PNAM - Head Part	MarksMaleHumanoid00NoScar "MarksMaleHumanoid00NoSc...
00072663	Delacourt	Delacourt	PNAM - Head Part	HumanBeard00NoBeard "HumanBeard00NoBeard" [HDPT:000...
00083D99	TalsgarTheWanderer	Talsgar the ...	PNAM - Head Part	

If you don't know which array index your value resides at, you can use an asterisk to have the function search all elements of the array: `[["Head Parts*", "MarksMaleHumanoid00NoScar "MarksMaleHumanoid00NoScar" [HDPT:0008555F]"]]`

Finally, you can search within records: `[["CNAM\DATA\Teaches", "One Handed"]]`.

0001326D	Miner	Miner	DATA - DATA	
0001326E	Lumberjack	Lumberjack	Unknown	00 00 FF FF
00013270	VendorTailor	Tailor	Teaches	One Handed
00013276	Priest	Priest	Maximum training level	100

Note that the function can be a bit finicky with capitalization and abbreviations. If you set an attribute condition that doesn't seem to get applied, try varying the above (e.g., replace "DATA" with "DATA - DATA").

## Record Template Structure

Record templates are how zEBD converts subgroup information into records within zEBD.esp. The record templates are JSON objects that contain all of the information to be written into the record. zEBD will replace the template's file paths with those supplied by a permutation of subgroups. Any remaining paths will be left as they are defined in the template. These templates make the patcher extensible; any "part" can be added as a record template and made into variants using subgroups. Record Templates are found in `zEBD\zEBD assets\RecordTemplates`. Let's use MaleTorso in `StandardTemplates_Male` as an example:

```
{
  "zEBDUniqueID": "MaleTorso",
  "zEBDPosition": "MODL",
  "zEBDSignature": "ARMA",
  "zEBDAddValidRaces": [
    "Additional Races"
  ],
  "zEBDSupportedRaces": "humanoid",
  "EDID": "MaleTorso",
  "BOD2 - Biped Body Template": {
    "First Person Flags": [
      "32 - Body",
      "34 - Forearms",
      "35 - Amulet",
      "36 - Ring",
      "38 - Calves"
    ],
    "Armor Type": "Clothing"
  },
  "RNAM - Race": "DefaultRace \"Default Race\" [RACE:00000019]",
  "DNAM - Data": {
    "Male Priority": 0,
    "Female Priority": 0,
    "Weight slider - Male": [
      "Enabled"
    ],
    "Weight slider - Female": [
      "Enabled"
    ],
    "Unknown": "02 00",
    "Detection Sound Value": 0,
```

```
        "Unknown": 17,  
        "Weapon Adjust": 0  
    },  
    "Male world model": {  
        "MOD2 - Model FileName": "Actors\\Character\\Character  
Assets\\MaleBody_1.NIF"  
    },  
    "Female world model": {  
        "MOD3 - Model FileName": "Actors\\Character\\Character  
Assets\\FemaleBody_1.NIF"  
    },  
    "Male 1st Person": {  
        "MOD4 - Model FileName": "Actors\\Character\\Character  
Assets\\1stPersonMaleBody_1.NIF"  
    },  
    "Female 1st Person": {  
        "MOD5 - Model FileName": "Actors\\Character\\Character  
Assets\\1stPersonFemaleBody_1.nif"  
    },  
    "NAM0 - Male Skin Texture": {  
        "zEBDUniqueID": "SkinMaleTorso",  
        "EDID": "SkinMaleTorso",  
        "zEBDPosition": "NAM0 - Male Skin Texture",  
        "zEBDSignature": "TXST",  
        "OBND - Object Bounds": {  
            "X1": 0,  
            "Y1": 0,  
            "Z1": 0,  
            "X2": 0,  
            "Y2": 0,  
            "Z2": 0  
        },  
        "Textures (RGB/A)": {  
            "TX00 - Difuse": "Actors\\Character\\Male\\MaleBody_1.dds",  
            "TX01 - Normal/Gloss": "Actors\\Character\\Male\\MaleBody_1_msn.dds",  
            "TX03 - Glow/Detail Map": "Actors\\Character\\Male\\MaleBody_1_sk.dds",  
            "TX07 - Backlight Mask/Specular":  
"Actors\\Character\\Male\\MaleBody_1_S.dds"  
        },  
        "DNAM - Flags": [  
            "Facegen Textures",
```

```

        "Has Model Space Normal Map"
    ]
},
"NAM1 - Female Skin Texture": "SkinBodyFemale_1 [TXST:0003EDE7]",
"NAM2 - Male Skin Texture Swap List": "SkinMaleHumanBody [FLST:0007BBB1]",
"NAM3 - Female Skin Texture Swap List": "SkinFemaleHumanBody
[FLST:0007BBB2]"
}

```

zEBDuniqueID: Tells the patcher that this is a record template. Can be whatever you want; I just set it to the EDID for convenience.

zEBDposition: Tells the patcher where the record corresponding to this template belongs within the record that receives it. This really only matters for top-level recordTemplates since they are assigned directly to an NPC and the patcher needs to know where to put them. However, I left it in subrecords for the sake of consistency.

zEBDsignature: The type of record this is to be. For example, “ARMA” for an armor addon. This matches what you should see in xEdit.

zEBDaddValidRaces: An array containing record elements to which the patcher will add patchable races. This is necessary, for example, for ARMA records - if an ARMA record doesn’t have the NPC’s race within its array of “Additional Races”, it won’t show up in the game.

zEBDsupportedRaces: The races to which this record template can be applied. Just like in config files, this can be EditorIDs or one of the groupings contained in a [Group Definition](#). If you add a custom record template, make sure that it doesn’t overlap in scope (e.g. contain the same zEBDsupportedRaces) as an existing record template. Note that while all of the default zEBDsupportedRaces are strings, this can also be an array of strings.

Everything else within the record template will be written as-is to zEBD.esp. Note that if a record contains sub-templates, they are defined as objects with their own zEBDuniqueIDs (see above: )

```

"NAME0 - Male Skin Texture": {

```

## Group Definition Structure

zEBD comes with convenient group definitions for races that can be used to assign subgroups and/or record templates. These are found in `zEBD\zEBD assets\RestrictionDefs`. If you create a config file that requires custom group definitions, create a new json file containing the new definition in the appropriate format and tell users to drop it in this folder. The default group definitions are found in `GroupDefs.json`. The definition object is very simple: it contains the group's name and an array of all EditorIDs that are assigned to it:

```
{
  "name": "humanoidYoungNonVampire",
  "entries": ["NordRace", "BretonRace", "DarkElfRace", "HighElfRace",
"ImperialRace", "OrcRace", "RedguardRace", "WoodElfRace"]
},
```

Any config file or record template that references “humanoidYoungNonVampire” will have that term replaced with the above entries during patcher execution.



## Important Considerations

- A) **Try to keep the number of options in the setting file as low as possible.** If the asset pack comes with textures in multiple resolutions, make separate config files for each resolution. Same with other options that the user probably wouldn't want to mix, such as default vs. black subsurface maps. The number of possible permutations scales geometrically and the more permutations there are, the more RAM is eaten by the patcher.
- B) If you're creating a config file for an asset pack that has a lot of options, please make **liberal** use of the "requiredSubgroups" and "excludedSubgroups" fields. For example, in the demo config file for Tempered Skins for Males, the number of top-level subgroup variants (excluding beast races) is:

Head diffuse	Head normal	Head sub-surface	Head specular	Head detail	Body diffuse	Body normal	Body subsurface	Body specular	Hands diffuse	Hands normal	Hands subsurface	Hands specular
7	34	1	4	6	10	4	1	8	9	2	1	8

That's 263,208,960 possible permutations, each of which has conditions that need to be stored. That will easily cause an out of memory exception. By adding the appropriate requiredSubgroups and excludedSubgroups, the count is trimmed down to < 10,000.

## Miscellaneous Notes

- A) **When assigning normal maps**, don't forget that DA13AfflictedRace uses Breton head normals and SnowElfRace uses High Elf head normals. These are set in the blank templates for your convenience. If you're making a new config file from the blank templates and your asset pack doesn't contain resources for Afflicted and Snow Elf, don't forget to remove them from the HN section.
- B) **Dawnguard vampires are a mess.** The blank template files distributed with zEBD should take care of the following complexity, but for the sake of explanation:

Almost all male XRaceVampire NPCs, where X is any humanoid race, use the SkinHeadMaleDarkElfVampire head texture. But don't be fooled; this has nothing to do with dark elves. If you look at the texture paths, they point at actors\character\male\maleheadvampire(\_msn).dds.

Females use SkinHeadFemaleVampire, SkinHeadFemaleOrcVampire, or SkinHeadFemaleDarkElfVampire. SkinHeadFemaleDarkElfVampire uses FemaleHeadVampire.dds diffuse but DarkElfFemale\FemaleHead\_msn.dds normals. Likewise, the orc uses Orc normals. SkinHeadFemaleVampire uses femaleheadvampire\_msn normals.

Elder Race Vampires use SkinHeadMaleOld/SkinHeadFemaleOld. As far as I can tell there is only a single Elder Vampire in all of the vanilla records (dunBloodletThroneVampireBoss [NPC\_:0005B830]). In Dawnguard he gets assigned the SkinHeadMaleOld texture. Therefore, ElderRaceVampires are eligible for the same face textures as Elders.

The default texture assignments (DFTM) set in the XRaceVampire records in Dawnguard.esm are as SkinHeadMaleDarkElfVampire for males and SkinHeadFemaleVampire for females, so it's safe to set those as default. For females, check for assigned SkinHeadFemaleOrcVampire and SkinHeadFemaleDarkElfVampire FTST records using ForcelfAttributes (this is done in the blank templates using subgroups `HN.Y.V.DE` and `HN.Y.V.O`).

You get into even more confusion with some unique Dawnguard vampires. For example, Hert is a NordRaceVampire with standard SkinHeadMaleNord textures. It's been so long since I played Dawnguard that I don't remember if that's supposed to be a plot point. Vampires using non-vampire textures:

Hert: SkinHeadMaleNord

Vingalmo: SkinMaleHighElf

Orthjolf: SkinHeadMaleNordComplexion\_Age50

Ronthil: SkinMaleHeadWoodElf

I added the above four NPCs to the default blocklist to prevent them from receiving vampire textures.

Note that Serana, Harkon, and Valerica are NordRace, not NordRaceVampire

- C) On a much simpler (but still confusing) note, **Wood Elf** males use face normals found in Actors\Character\WoodElfMale\MaleHead\_msn.dds, but females use Actors\Character\HighElfFemale\FemaleHead\_msn.dds. Why? Ask Todd.
- D) When setting up config files for beast races, make sure to handle “old” characters. Note that paths that are untouched by any subgroup will default to those in the RecordTemplate. This is a problem for “old” characters because the detail maps in the RecordTemplates all point to “blankdetailmap.dds”, so the old characters will have young detail maps. If your texture pack contains “blankdetailmap.dds” and its “old” detail map counterpart, your config file will need to handle them anyway so you won’t forget. However, if your texture pack does not contain the detail maps, you’ll need to remember to set them up using the default Skyrim paths. Below is how to do that for Argonians. See my CoverKhajiits config file for how to handle it for Khajiits.

```
{
  "id": "HDE.Arg",
  "enabled": true,
  "distributionEnabled": true,
  "allowedRaces": [
    "argonian"
  ],
  "disallowedRaces": [],
  "allowedAttributes": [],
  "disallowedAttributes": [],
  "forceIfAttributes": [],
  "allowUnique": true,
  "allowNonUnique": true,
  "name": "Argonian",
  "requiredSubgroups": [],
  "excludedSubgroups": [],
  "addKeywords": [],
  "probabilityWeighting": 1,
  "paths": [],
  "subgroups": [
    {
      "id": "HDE.Arg.Y",
      "enabled": true,
      "distributionEnabled": true,
```

```
"allowedRaces": [],
"disallowedRaces": [],
"allowedAttributes": [],
"disallowedAttributes": [
  [
    "FTST",
    "SkinHeadMaleArgonianOld [TXST:000B3047]"
  ]
],
"forceIfAttributes": [],
"allowUnique": true,
"allowNonUnique": true,
"name": "Young",
"requiredSubgroups": [],
"excludedSubgroups": [],
"addKeywords": [],
"probabilityWeighting": 1,
"paths": [
  [
    "Skyrim.esm\\Actors\\Character\\ArgonianMale\\BlankDetailmap.dds",
    "BlankDetailmap.dds"
  ]
],
"subgroups": [],
"$hashKey": "object:5021"
},
{
  "id": "HDE.Arg.O",
  "enabled": true,
  "distributionEnabled": true,
  "allowedRaces": [],
  "disallowedRaces": [],
  "allowedAttributes": [
    [
      "FTST",
      "SkinHeadMaleArgonianOld [TXST:000B3047]"
    ]
  ],
  "disallowedAttributes": [],
  "forceIfAttributes": [
    [
```

```
        "FTST",
        "SkinHeadMaleArgonianOld [TXST:000B3047]"
    ]
],
"allowUnique": true,
"allowNonUnique": true,
"name": "Old",
"requiredSubgroups": [],
"excludedSubgroups": [],
"addKeywords": [],
"probabilityWeighting": 1,
"paths": [
    [
        "Skyrim.esm\\Actors\\Character\\ArgonianMale\\Actors\\Character\\ArgonianMale\\ArgonianMaleHeadOld.dds",
        "BlankDetailmap.dds"
    ]
],
"subgroups": [],
"$hashKey": "object:5033"
}
],
"$hashKey": "object:4966"
}
```

# FAQ (Anticipated)

Q) I'm getting an error about failing to parse JSON. What do I do?

A) Your JSON file is in the wrong format. The warning should tell you which JSON file it is. Run it through an online parser such as [this one](#) and track down where the formatting error is. This should only happen if you manually edit one of the configuration files.

Q) I'm getting an error about ng-repeats when pulling up the GUI. What does this mean?

A) Did you edit an Asset Pack Settings file outside of the GUI? If so, did you copy any subgroups? If so, within the duplicated subgroup delete the "\$\$hashKey" that's automatically generated by zEdit when exporting to JSON. Duplicate hash keys are not allowed.

Q) I added a new asset pack but my NPCs aren't using it after patching.

A) If consistency is enabled, you need to clear it. Otherwise, if your previous asset packs are still installed the patcher will assign them right back to those same NPCs, because that's what consistency does.

Q) I'm getting an out of memory error during patching. What should I do?

A) I've only ever encountered this using developer mode, but it's possible in standard zEdit if your config file is very complex and doesn't have enough subgroup restrictions (see [Important Consideration B](#)). Try identifying subgroups that can be linked together using RequireSubgroups/ExcludeSubgroups. That said, for large config files the patcher can use 1 to 2 GB of RAM, so keep that in mind.

Q) My NPCs are completely white/purple!

A) That's the secret "Ghost" texture pack! ... kidding, did you remember to check (enable) the zEBD - Mod folder in the left panel of MO2? zEBD warns you if any of the file paths expected by the config file don't exist, so this shouldn't happen unless you ignore the warning.



# Known Issues

- 1) After saving an Asset Pack Settings file or viewing a popup alert from a settings file, you can no longer edit any settings or save again until you minimize zEdit or alt+tab out of zEdit and bring it back up again (thanks Mator for finding the fix).

# Change Log

2020 05 02 Beta 1:

- Initial Release

2020 05 02 Beta 2:

- Permutations are assigned to NPCs during **filter()** rather than during **patch()** to avoid creating ITMs in the case where no valid permutations can be generated for the given NPC

2020 05 08 Beta 3

- Reduced a significant source of unnecessary memory consumption to better support config files with many variants.
- "RequiredSubgroups" and "ExcludedSubgroups" can now contain a parent subgroup. Previously they had a "silent" requirement of being bottom-level subgroups.
- Permutation log now correctly displays allowed/disallowed attributes
- New General Setting: *Link NPCs by Name/Race/Gender*. If checked, NPCs that appear as multiple records with unique FormIDs (e.g. several Ulfrics, Ciceros, and a couple Dawnguard characters, as well as a few NPCs such as Sven and Nazeem in Interesting NPCs) will get assigned the same permutation.
- Fixed bug where permutation log showed Forcelf attributes as DisallowedAttributes (attributes themselves were correct; this was purely a display error).
- RecordTemplates: (fe)malebody\_1\_msn\_feet.dds changed to (fe)malebody\_1\_feet\_msn.dds to comply with the naming convention for other texture types. **All config files updated to comply with this change.**
- Fixed GUI bug where changes to "disallowedRaces" were not being saved.

2020 05 13 Beta 4

- Fixed function that was broken in Beta 3 - gives users a popup alert if the config file validator encounters an error upon parsing a subgroup.
- Fixed an error string that's sent to the error log if the parser encounters a duplicate subgroup name. Error log now correctly tells the user that this is the problem.
- Deleted src folder: not needed (duplicate code from index.js)
- "RequireSubgroups" now has implicit OR functionality - if multiple subgroups belonging to the same top-level subgroup are entered, then any of them will satisfy the RequireSubgroups criteria. **This is backwards compatible with config files from Beta 3.**
- "RequireSubgroups" now bidirectionally checks parent subgroup (previously the added subgroup's parent subgroups were checked, but the sub-permutation's constituent parent subgroups were not).
- Updated blank example config files (humanoid male and female) to reflect the new record templates introduced in Beta 3 (switch from "msn\_feet" to "feet\_msn").
- Added zEBD assets\MiscConfig\DevControls.json to enable permutation buildup logging and verbose mode
- forcelf permutations are checked for validity for the given NPC before being prioritized for random assignment. This slightly slows down the patching process but prevents a lot of spurious error messages.
- Cleaned up the warning generated if a config file references files that don't exist.

#### 2020 05 15 Version 1.0

- Code unchanged since Beta 4.
- Config files updated with Director's Cuts

#### 2020 05 18 Version 1.1

- Fixed bug where permutations with incompatible RequiredSubgroups would use the requiredSubgroup from the first top-level subgroup rather than being discarded
- Fixed config files for Tempered Skins for Males (vanilla and SOS Light) - Snow Elves were getting High Elf textures (thanks to **afboak** for pointing out the error!)
- Fixed record templates to remove unnecessary texture swap lists (thanks to **afboak** again for pointing out the error!)

#### 2020 05 21 Version 1.2

- Minor bug fix (line 84 of permutationGenerator.js had a typo in a variable name)
- Minor bug fix (patcher would fail when a race defined in GroupDefs.json was not present in the user's load order in some circumstances)
- Fixed path for `ArgonianFemaleHead_s.dds` in Argonian Female record template

#### 2020 06 02 Version 1.3

- **Feature update for end users:** Added NPC Height Configuration
- **Feature update for config file developers:** Added the ability to bypass creation of permutations and records from JSON. These options are in zEBD Assets\MiscConfig\DevControls.json.
- Bugfix: When two subgroups's requiredSubgroups were compared, if one contained the child of the other, they were deemed incompatible rather than being trimmed down to just the child subgroup. This is now fixed. This is a rare case that specifically affected the Mature Skin UNP HD (Alternate Complexions) Younger NPCs Director's Cut config file (but not the CBBE version).
- Fixed config files for Demoniac UNP and CBBE. Sweaty v2 textures were dropping out because of faulty "requiredSubgroups". My bad.
- Validation for required and excluded subgroups now tells the user if those subgroups don't exist.
- There is also now a "validate" button next to "Show Asset Pack Configurations". This lets you validate changes to your config files as you're making them (they are also still validated upon saving).
- Fixed a mistake in Blank Female Humanoid Settings.json (not used for patcher generation; only a blank template for users to make their own config file) that prevented female vampires from receiving their correct normal maps. Thanks again to **afboak** for the heads up!
- Under the hood update - separates the code to get NPC information from xelib into its own separate function (no longer in the choosePermutation function). Prevents a bit of code redundancy, and more importantly facilitates non-permutation-related functions such as height assignment.
- Under the hood update - got rid of lib\NPCAssignmentClass.js because it was a very cumbersome way to do what I was trying to do.
- UI update to the NPC config menu: Config Files and subgroups are now boxed. This should hopefully make editing a bit less confusing for configs with multiple nested subgroups.
- Huge UI update courtesy of **a lot** of help from **Mator**, both to make the cool layout for the height menu and also to get rid of some of my crappy UI code to eliminate the menu lag that would happen when loading multiple config files. Thanks Mator!