Creating a Danglymesh for NWN1-models in Blender (by Black Rider V. 1.0)

Danglymesh (or AuroraFlex - as it is called in (G)Max) is a mesh, which is influenced by the movement of the object (e.g. a creature model) and/or the wind settings of an area. It is able to sway/dangle back and forth.

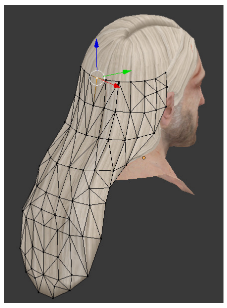
From a (G)Max-tutorial:
They can be affected by:
- part movement relative to the world (they have an inertia).
- wind dynamics, either from global wind settings or from wind events like explosions.

The general process is to assign values to the mesh properties and certain vertices of this mesh, which determine the strength of the influence. In Blender, it's accomplished via weight painting - the same way as for creating a skinmesh.

You'll need Blender and the tool "Neverblender".
https://neverwintervault.org/project/nwn1/other/tool/neverblender-27
https://neverwintervault.org/project/nwn1/other/tool/neverblender-28

This tutorial is written for Blender 2.79, but the process should be the same for 2.8 and higher. I assume. that you have some basic Blender-knowledge. Please check out all those other good Blender-tutorials on the Neverwinter Vault, e.g. here:
https://neverwintervault.org/project/nwn1/other/do-it-blender
Also all screens are made using the ***Default Layout*** of Blender.

For this tutorial, I used "hair" as an example:

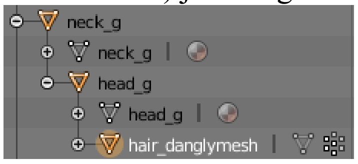


But there are a lot of other examples, where you find danglymeshes, for example on clothes, flags, …

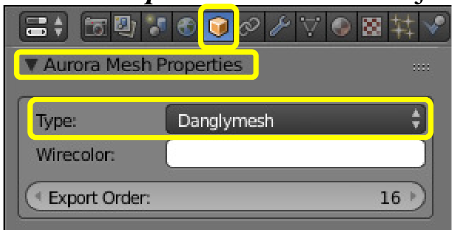
Let's start:
There are two basic steps to do, to create a ***danglymesh***: Assign values to the vertices of the danglymesh and set the danglymesh properties. The order of all the steps doesn't really matter, but I like to start (after doing some basic steps) with the vertex-work, because we do need (during this process) the defined ***Vertex Group*** for the settings in the ***Danglymesh Properties***.

Step 1: Parenting objects

Since my example dangly-hair-object is a separate mesh and is to be moved with the head (head_g), it needs to be linked/parented to the head (head = parent, hair = child). For this, (in the ***Outliner***) just drag the hair-object onto the head_g-object:



Step 2: Setting the mesh properties
Set the ***Propetries*** of the hair-object to ***Danglymesh***:



Aurora Mesh Properties

Type: Danglymesh

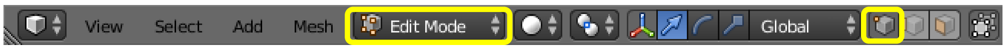
Wirecolor:

Export Order: 16

Remark: The settings for the Danglymesh Properties could be done now, but for this tutorial, it is covered further down (see step 5 on page 7).

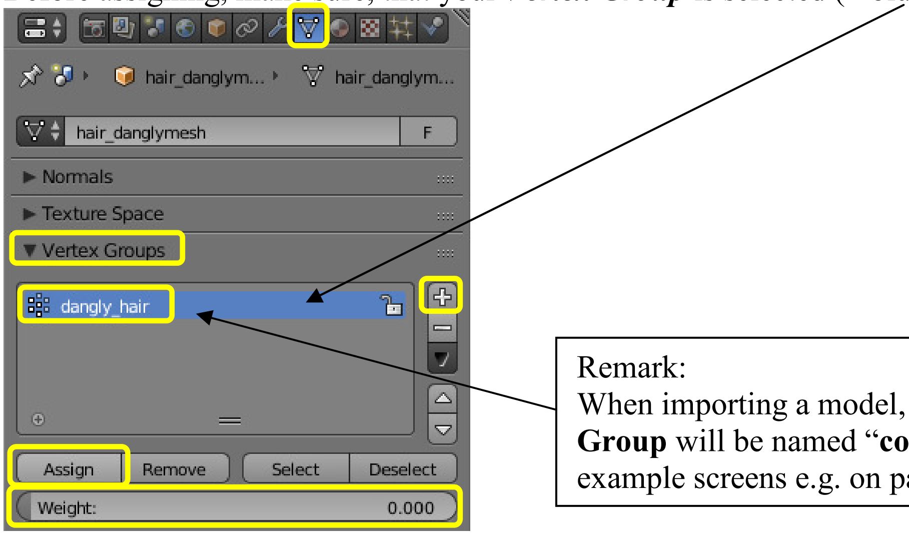
Step 3: Create a Vertex Group
Now all the vertices of this object, which shall be influenced, need to be collected in a ***Vertex Group.***
(Remark: I'm sure, if really all vertices need to be added to this group or only those, which do get a "dangle-value" - I have not tested it…)

a) Selecting the vertices
In ***Edit*** mode activate ***Vertex Select***



(Normally, all vertices should already be selected (yellowed tint). To be on the safe side, press the [A]-key to see, if they change their coloring.)

b) Define the *Vertex Group*:
When done selecting the vertices, select the ***Data***-tab, find ***Vertex Groups***, click the '+' to add a new group, rename it (not really necessary), change ***Weight*** to **ZERO (0.0)** and click the ***Assign***-button.
Before assigning, make sure, that your ***Vertex Group*** is selected (= blue underlay)!



hair_danglym… hair_danglym…

hair_danglymesh F

Normals

Texture Space

Vertex Groups

dangly_hair

Assign Remove Select Deselect

Weight: 0.000

Remark:
When importing a model, the dangly-**Vertex Group** will be named "**constraints**" (see example screens e.g. on page 5).

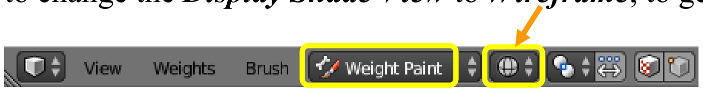
Later, this ***Vertex Group*** will be assigned to the ***Danglymesh Properties*** (see step 5 page 7).

Step 4: Weight Paint

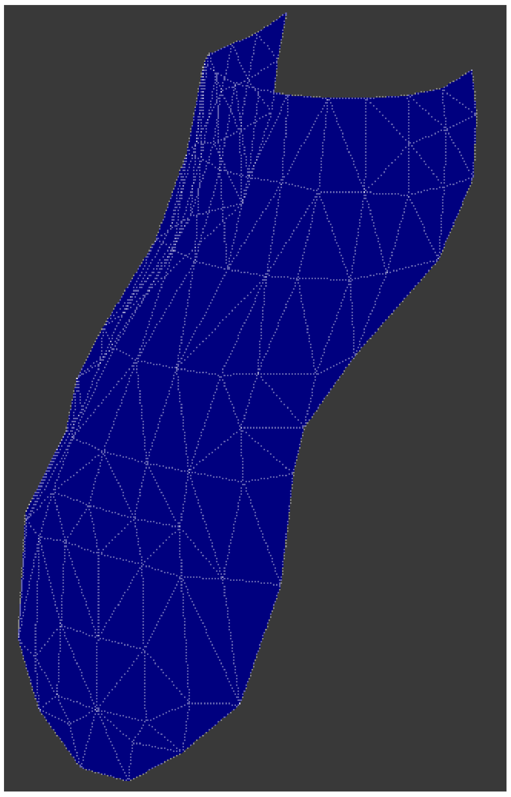
Now, it works like this: Every vertex of the mesh will get a weight value. The value will tell the NWN1-engine, how much the “dangle-influence” is.
Vertices with a weight value get colored from red (= 100% weight) down to a light blue (very low weight value). As written: A strong blue color means no weight influence (= ZERO).

a) Preparing:

Select the object, change the ***Interaction Mode*** (of the 3D-view) to ***Weight Paint***. I also like to change the ***Display Shade View*** to ***Wireframe***, to get a good control view on my doings:



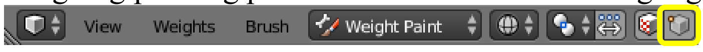
The mesh should now show a bluish coloring:



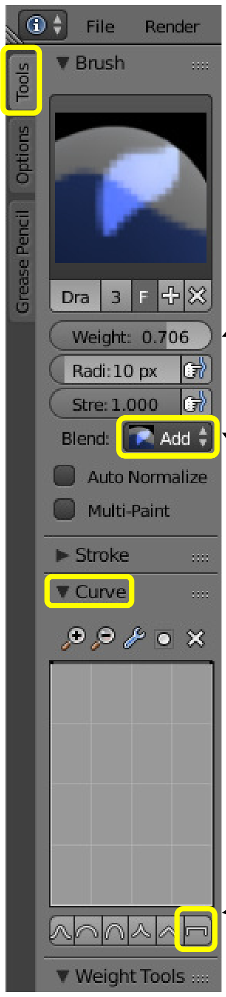
If the coloring is red, then you have not set the Weight to ZERO, when creating the ***Vertex Group*** (see step 3b on page 2). In this case, I’ll recommend, that you remove the weighting from your mesh:
Select all vertices, go to the ***Data***-tab, set the ***Weight*** of the ***Vertex Group*** to ZERO and click the ***Assign***-button (see step 3b).

b) Weight assigning/painting:

Make sure, that the ***Vertex Select Mode*** is DEACTIVATED (for the weight assigning/painting procedure = the button has a light grey background):



I like to use the following settings (left side of Blender 3D-view):



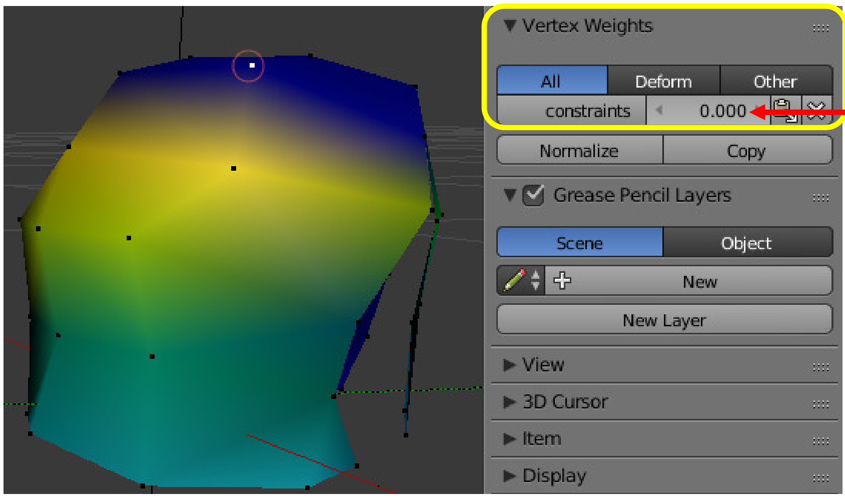
This value will change, depending on the position within the mesh and how much the dangle-value should be (see reference-screens on pages 4/5).

When clicking on a vertex/vertices ONCE, the Weight-value will be **ADD**ed to the vertex/vertices. (Accordingly when switching the Blend-type to **Subtract** (=Subt), the value will be subtracted.

This will make sure, that ALL vertices within the brush-circle will get the same amount of weight. (Other ***Curves*** lessen the influence of the weight on the vertices towards the rim of the brush-circle!)

c) Reference

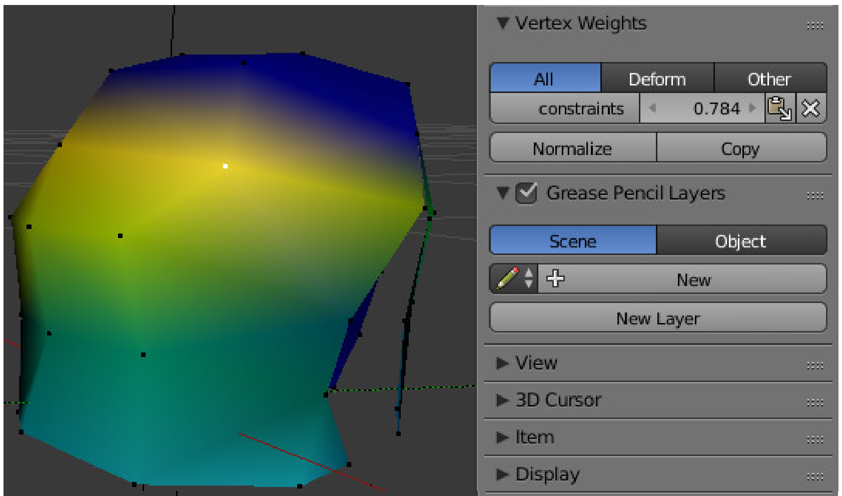
Now, it would be smart to import a model from the original game and check, what weight-values are assigned to which region of the mesh. This is my reference from: pfh0_head015



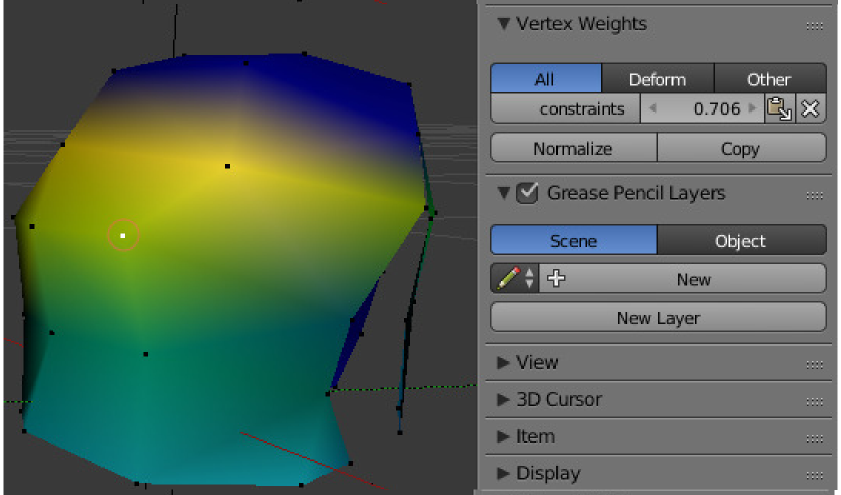
Needed value.

Remark 1 (again):
When importing a model, the dangly-**Vertex Group** will be named "**constraints**".

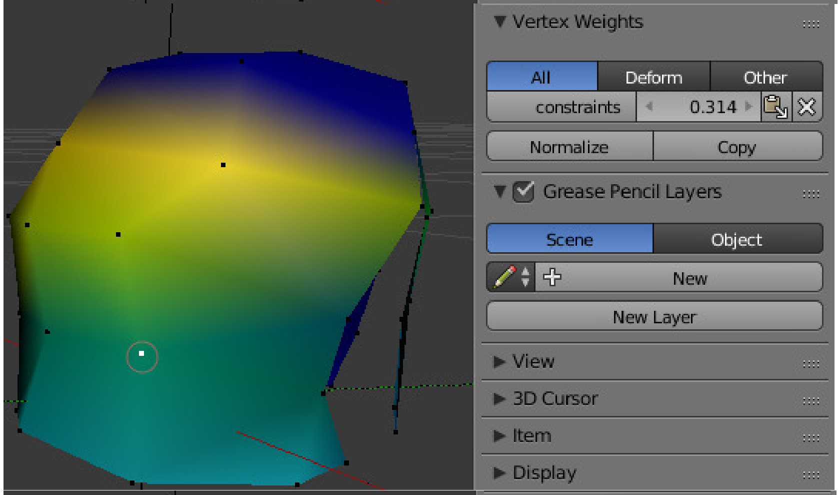
Remark 2:
To get this view, see step 4d on pages 6/7.



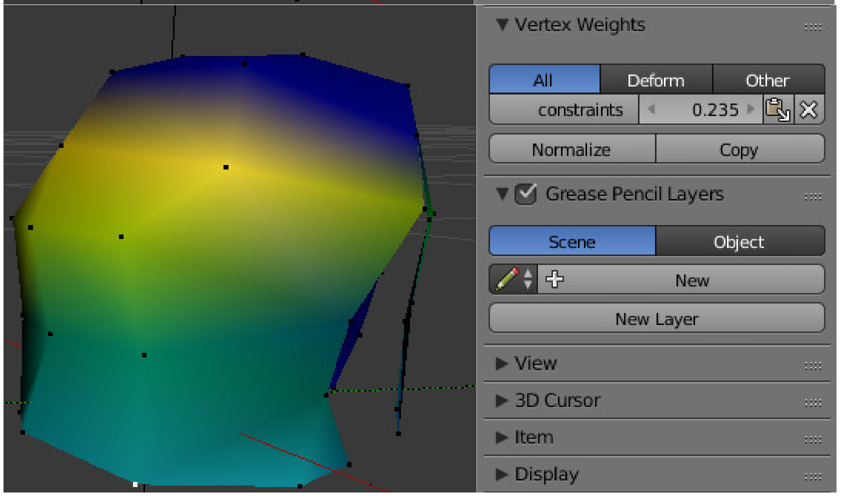
constraints 0.784



constraints 0.706



constraints 0.314



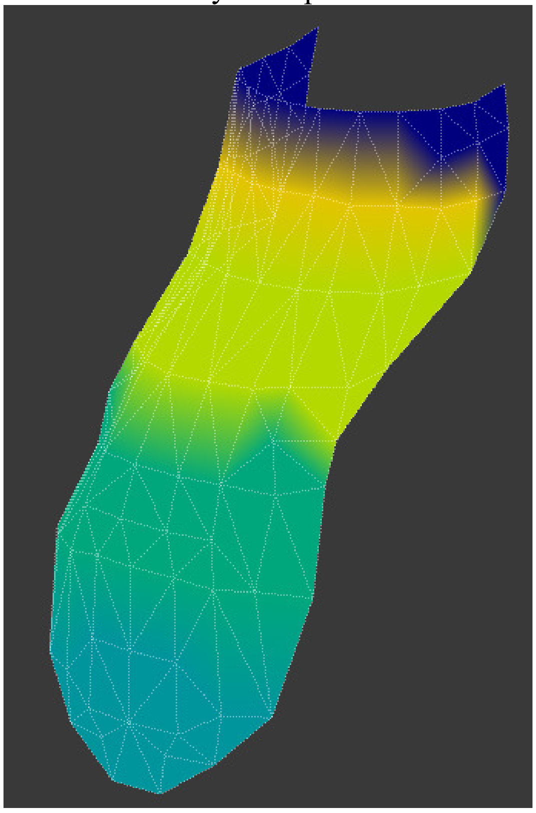
constraints 0.235

Sum up of the values (can be used as a copy base):
0.0
0.784314
0.705882
0.313726
0.235294

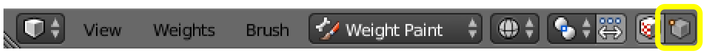
→ Now, the values don't have to be exactly like that! Something like 0.8 would pretty much behave like 0.784314! These values come from a (G)Max export. (G)Max is setting the dangle-values by applying a color-value to the vertices and they are translated into these numbers upon export.

Pick your values and set them as ***Weight***-value in the ***Brush***-panel (see step 4b on page 4). Again, make sure, that the ***Vertex Select Mode*** is DEACTIVATED (see step 4b on page 3) and LEFT-click or brush the vertices ONCE to assign the set value.

The result for my example:

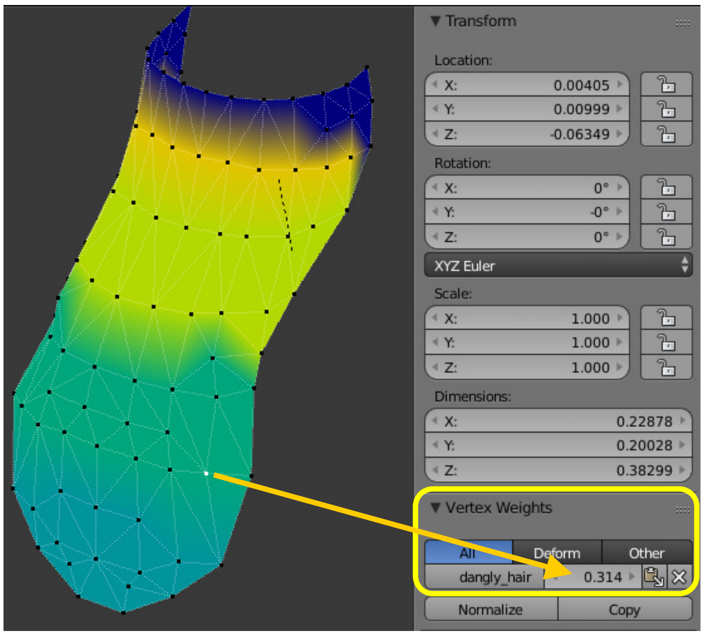


d) Fine adjust:
You might want to re-check the values of certain vertices. For this, ACTIVATE the ***Vertex Select Mode*** (= the button now has a darker grey background):



→ The vertices are now displayed within the mesh.
→ RIGHT-click a vertex to select it.

→ In the Properties-Panel ([N]-key) on the right side of the 3D-view, you find and can edit the values:



Transform

Location:
X: 0.00405
Y: 0.00999
Z: -0.06349

Rotation:
X: 0°
Y: -0°
Z: 0°

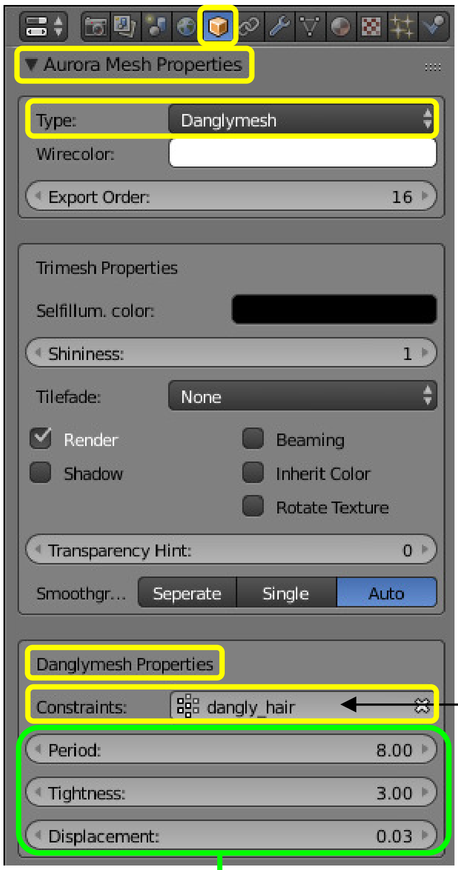
XYZ Euler

Scale:
X: 1.000
Y: 1.000
Z: 1.000

Dimensions:
X: 0.22878
Y: 0.20028
Z: 0.38299

Vertex Weights
All | Deform | Other
dangly_hair 0.314
Normalize | Copy

Step 5: Setting up the properties:
The "dangled" mesh has to be flagged as ***Danglymesh***. With your mesh selected, choose the ***Properties***-tab and find ***Aurora Mesh Properties***.
Here set ***Type*** to ***Danglymesh*** (if you have not done so before - see step 2 on page 2). Now find ***Danglymesh Properties*** and click into the field behind ***Constraints*** and choose the correct ***Vertex Group*** = your Danglymesh-Vertex Group.



Aurora Mesh Properties

Type: Danglymesh
Wirecolor:
Export Order: 16

Trimesh Properties
Selfillum. color:
Shininess: 1
Tilefade: None
Render [✓] | Beaming
Shadow | Inherit Color
Rotate Texture
Transparency Hint: 0
Smoothgr... Seperate | Single | [✓ Auto]

Danglymesh Properties
Constraints: dangly_hair
Period: 8.00
Tightness: 3.00
Displacement: 0.03

Select your Danglymesh-Vertex Group (also see step 3b on page 2).

Now the ***Danglymesh Properties Settings*** need to be set:

Taken from the **Neverblender**-PDF-Manual:

Dangle group (= Constraints)
The dangle group is a vertex group containing the weights of vertices for the danglymesh. You must select an existing vertex group. Vertex paint may be used to paint the weights. A vertex with a weight of 0.0 will not change position.

Period
Determines how quickly the vertices move and how fast they come to a rest.

Tightness
The 'tension' in the object, restricting movement.

Displacement
Determine show far the vertices are allowed to move. This is further restricted by the Dangle group.

Just for reference/comparison: Explanations form a (G)Max-tutorial:

Flex Settings
These are some sample Flex settings used in NWN, to give you an idea of the range and what sort of values are used.

Example	Sway	Strength	Flex
-	The strength of the flex that causes it to stop. Think of it as friction. A lower value means it takes longer for it to stop (see quote at bottom).	The springiness of the verticies. Higher value means it will snap and bounce a lot, while a lower one means it is more lazy and smooth.	The distance it is allowed to move (in centimeters?)
Hair	8.0	3.0	0.3
Cloth, Sleeve	6.0	3.0	0.4
Cloth, Shirt	7.0	3.0	0.35
Dangling Chainmail	5.0	0.1	1.0

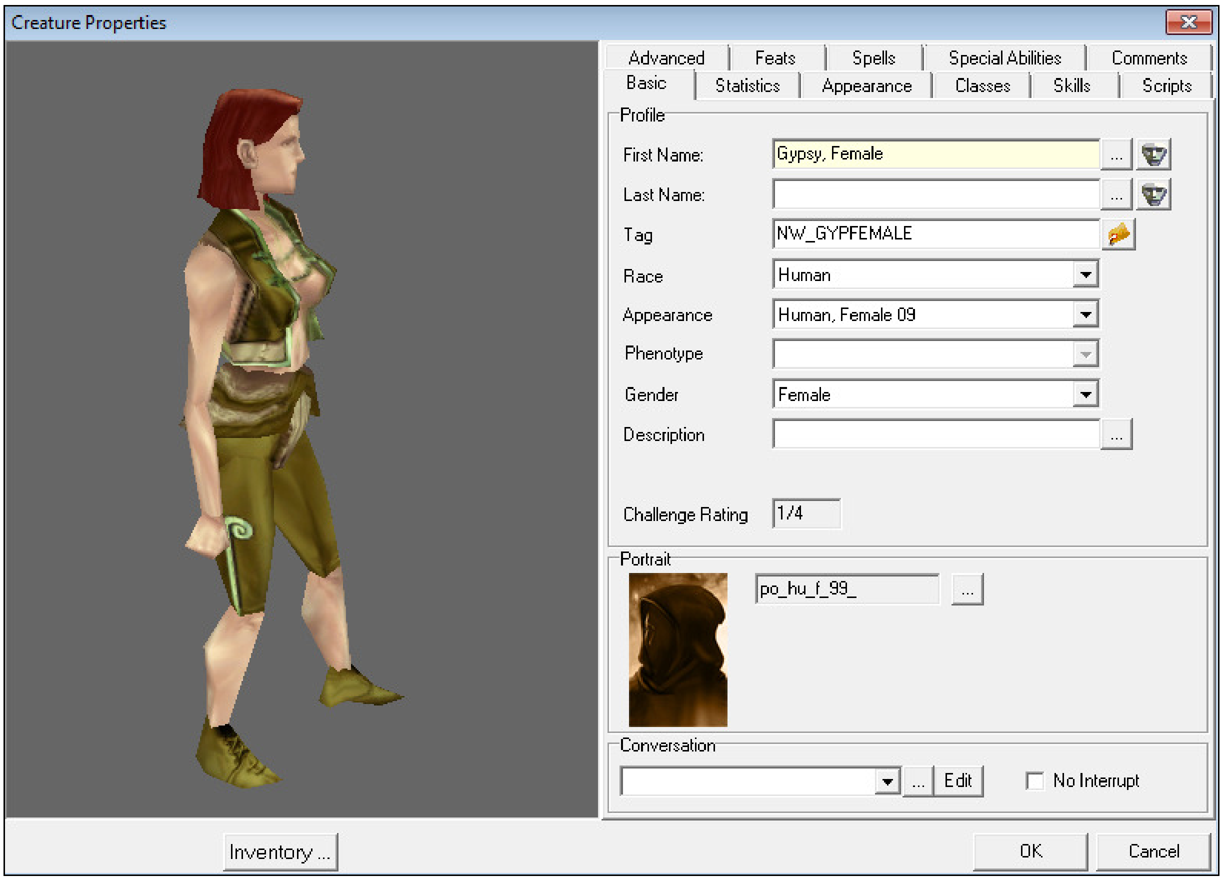
For the completeness, here is a link to a (G)Max-page:
http://harvestmoonconsortium.com/forums/viewtopic.php?f=59&t=2194

Step 6: Possible additional steps and remarks
As mentioned, if you are NOT "dangling" an existing part of a NWN1-model, you'll need to parent that danglymesh-object to a NWN1-body part (e.g. a new dangled hair-mesh to the head_g-part).

So after finding the right position for the danglymesh-object, it might be a good idea to place the origin of the child (the "dangled" mesh) at the same position as the origin of the parent.

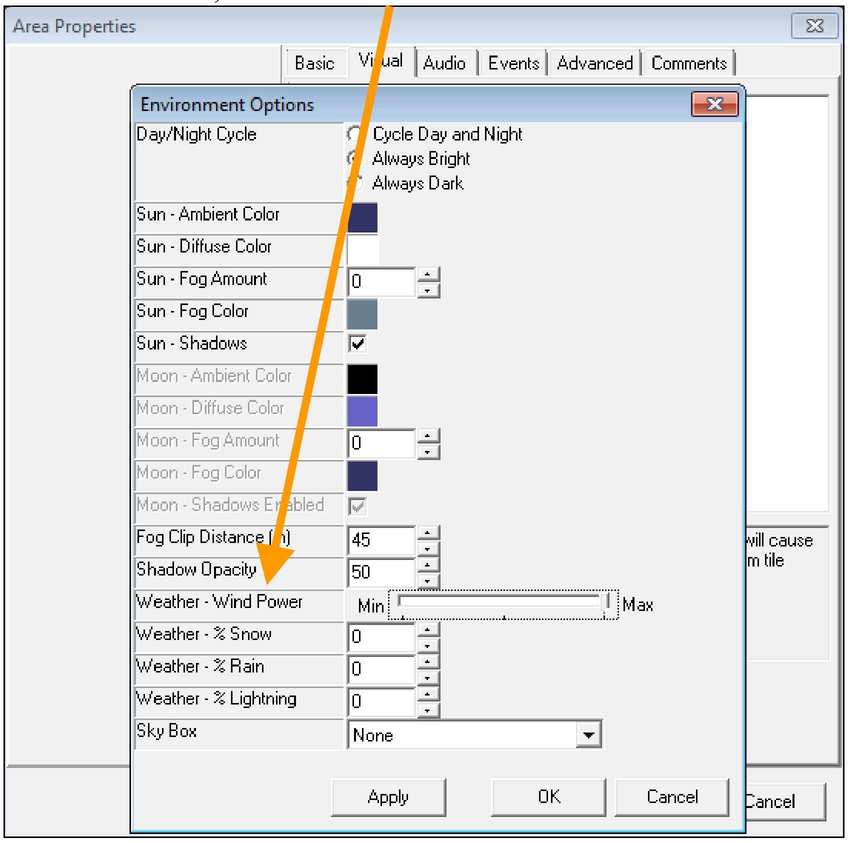
Especially, if you imported that new object from another non NWN1-model, make sure, that that the mesh doesn't hold any rotation and scale values (this might happen, when you do a e.g. obj-, fbx-, …-import)! **Do an "Apply Rotation/Scale" on the imported object - to be on the safe side!**

Step 7: Export and testing
To get the dangling look nice, some testing is required. So do an export and get you model into the toolset.
A first quick test can be done in the toolset: Open the properties of the object and, in the preview window, move the model around a bit:


Creature Properties

First Name:	Gypsy, Female
Last Name:	
Tag	NW_GYPFEMALE
Race	Human
Appearance	Human, Female 09
Phenotype	
Gender	Female
Description	
Challenge Rating	1/4
Portrait	po_hu_f_99_

For an IG-check, set the ***Wind Power*** of the area to max:



Day/Night Cycle	Cycle Day and Night [✓ Always Bright] Always Dark
Sun - Ambient Color	■
Sun - Diffuse Color	□
Sun - Fog Amount	0
Sun - Fog Color	■
Sun - Shadows	✓
Moon - Ambient Color	■
Moon - Diffuse Color	■
Moon - Fog Amount	0
Moon - Fog Color	■
Moon - Shadows Enabled	✓
Fog Clip Distance (m)	45
Shadow Opacity	50
Weather - Wind Power	Min ... Max
Weather - % Snow	0
Weather - % Rain	0
Weather - % Lightning	0
Sky Box	None

Place a "standing" version of your model into that area. If your model is a creature, put in a 2nd version, place some waypoints and let it walk. This is a pretty good way to check your dangle settings.

Happy dangling!

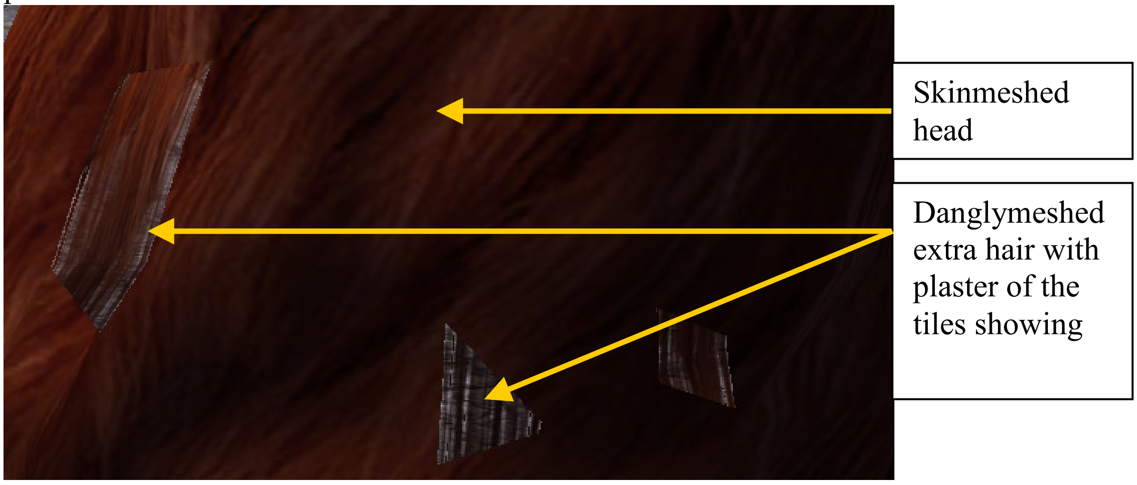
- End of the main tutorial -

Appendix

Appendix: **Combining Skinmesh and Danlymesh:**
In general: Skinmeshed parts and danglymeshes don't get along too well, because both, the skinmesh and the danglymesh, are deformed, but the vertex positions of these meshes are not dependant on each other. So, there is a chance, that a skinmeshed and a danglymeshed object may move apart (e.g. skinmeshed head and danglymeshed hair), especially, if the skinmeshed object is rigged to more than one bone! It should work OK for heads, if the head (skin-) mesh or at least the vertices, which connect to the danglymeshed object (hair), is/are weighted 100% onto the head-bone (NWN1: head_g).

Parenting:
As mentioned twice before, since a danglymesh is not a skinmesh, it needs to be parented to a NWN1-body part (Pseudo Bone). So hair needs to be parented to the head_g-object/part.

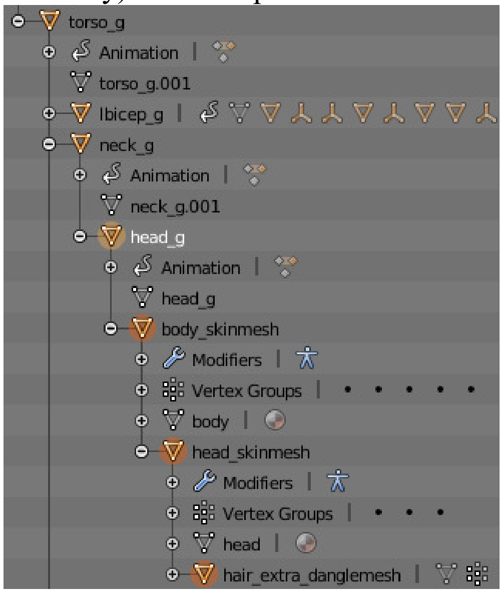
Render order:
In most cases, for a skinmesh it doesn't matter to what object it is linked. But for displaying an alpha texture it is important!
In the following example screen, I rigged a body, so the head is a skinmesh. The extra hair is a danglymesh. Now, if the extra (danglymeshed) hair is rendered BEFORE the head, the NWN1-engine will make that danglymesh a kind of "see through" object, which means, you can see the environment (e.g. grass, plaster, …) when zooming onto that extra dangly hair-part:



To get this right, you have to parent the head-skinmesh to the head_g-part and then the extra danglymeshed hair to the head-skinmesh!

There is a another problem, if - in this case - you have a skinmeshed head, which also makes use of transparent alpha areas in the texture.
Example: (skinmeshed) head hair on a separate body (skin-) mesh:

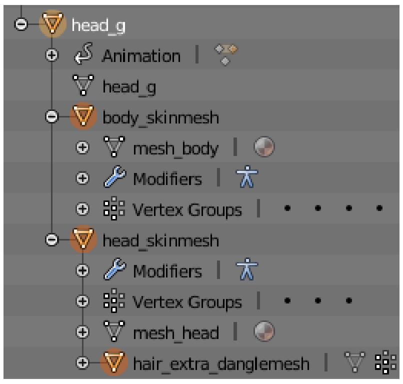
In theory, you would need to parent the body skinmesh to the head_g-part, parent the head skinmesh to the body skinmesh (so that the transparency of the back hair is rendered correctly) and then parent the extra dangle mesh hair to the head skinmesh.



```
torso_g
  Animation
  torso_g.001
  lbicep_g
  neck_g
    Animation
    neck_g.001
    head_g
      Animation
      head_g
      body_skinmesh
        Modifiers
        Vertex Groups
        body
        head_skinmesh
          Modifiers
          Vertex Groups
          head
          hair_extra_danglemesh
```

Problem is: IG, the head mesh is not rendered and the Toolset will crash. The program “CleanModels” will parent the skinmeshes back to the Aurora base. So this seems to be a combination, which will NOT work!

After some testing I found, that this way seems to be working (as long as the body mesh is rendered before the head mesh!):



```
head_g
  Animation
  head_g
  body_skinmesh
    mesh_body
    Modifiers
    Vertex Groups
  head_skinmesh
    Modifiers
    Vertex Groups
    mesh_head
    hair_extra_danglemesh
```

Here are two links to skinmeshed models with danlymeshed parts:

https://neverwintervault.org/project/nwn1/model/witcher-1-creature-conversion-project-skinmeshed-geralt-models-4-6-ravens-armor

https://neverwintervault.org/project/nwn1/model/witcher-1-creature-conversion-project-skinmeshed-corpulent-commoners

- End of the tutorial -