Cyan Worlds
Avatar Documentation

Table of Contents

1. AssetMan

AssetMan: Main/GlobalAvatars/

1.1. Avatar Files Used

- **Texture**: all the texture for the avatar.
- **MAX File** : all the meshes, skinning, and wiring of the avatar
- **File**: Photoshop source files for texture, macroscripts for custom 3ds Max tools, avatar documentation, and any source files for helping to create avatar content.

AssetMan: Main/GlobalAvatars/

1.2. Local Assets Directory

A copy of all the assets used within 3ds Max files should be placed in the following folder:

- C:\Assets

Getting assets manually:

- Open AssetMan and manually get the files needed

Getting assets automatically:

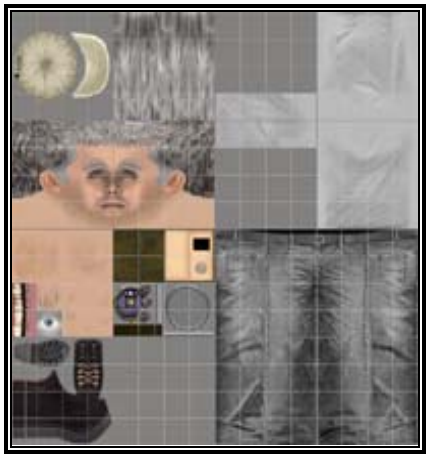
- Within 3ds Max select **File**
- Select **Open From AssetMan. . .** to download all the assets needed for that particular file

2. Texture Files

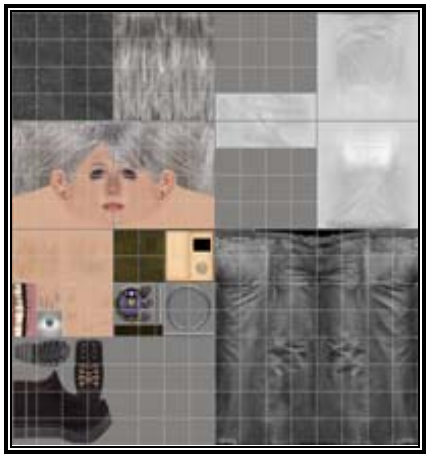
AssetMan: Main/GlobalAvatars/Texture
AssetMan: Main/GlobalAvatars/File (Source Textures)

2.1. Texture Palette

The textures are all placed on the base palette at run time, meaning the base palette swaps in new textures on the fly. Once a player decides on what to wear the base palette is saved in the current state and stored as the player's custom palette. Again, the special characters have their own custom palette that cannot change during game play.



Male Base Palette (1024 x 124)



Female Base Palette (1024 x 124)

2.2. Tile Sets

All the textures are created to fit certain tile sets. The tile set for a particular texture has a unique size and placement on the base palette.

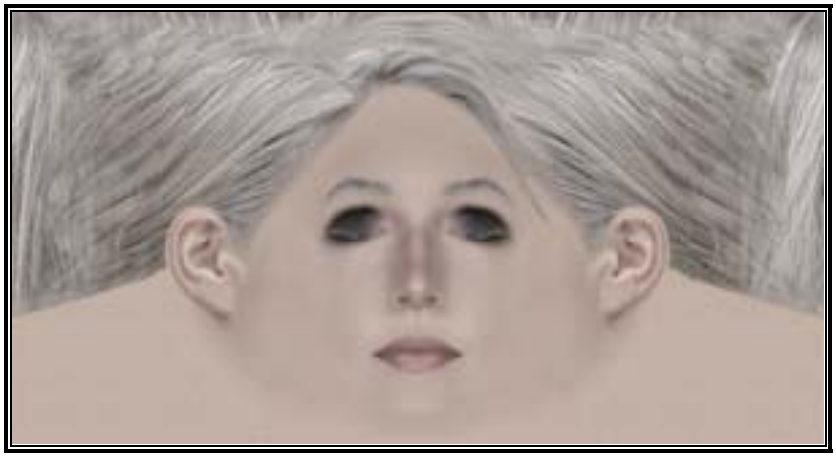
The **Head tile set** is composed of:

- **face** texture (drawn first)
- **hair** texture (drawn second)

Below is an example of how the **Head tile set** is composed.

FFace_skin.tga (size: 512 x 256, placement: 0, 256)

FHair_PonyTail01_tint1.tga (size: 512 x 256, placement: 0, 256)



Composite: face drawn first, then the hair

Tile Sets

Tile Set Name	Subgroup	Placement width, height	Size width, height
Foot		0, 768	256 x 256
Hand:			
	Finger	128, 640	128 x 128
	LOD	64, 640	64 x 64
	Palm	0, 512	256 x 128
Head:			
	Eye	64, 704	64, 64
	Extra Hair	256, 0	256 x 256
	Face	0, 256	512 x 256
	Hat	0, 0	256 x 256
Legs		512, 512	512 x 512
Torso			
	Chest	768, 0	256 x 512
	Arm	512, 192	256 x 128
Glasses		384, 640	128 x 128
KI		256, 640	128 x 128
Player Book		256, 512	256 x 128

2.3. Layout

Textures are divided into the following groups:

- Face
- Face Accessories
- Hair
- Torso
- Hands
- Legs
- Feet
- Accessories

Texture Layout

Texture Group	Texture Name	Texture Description	Notes
Face			
	*Face_skin *Face_skin1 *Face_skin2 *Face_skin3 *Face_skin4 UFaceEye_tint1	Face Base Layer Face Race Layer Face Race Layer Face Race Layer Face Age Layer Eyes	The *face* has a morphing texture with the ability to tint the skin and the eyes.
Face Accessories			
	MFace_xx_tint2 FFace_xx_tint2	Facial Hair Ear Rings	*Facial hair* is tinted globally with the hair color. *Facial jewelry* is also tintable. *Ear rings* are tintable.
Hair			
	*Hair_skin *Hair_xx_tint1 *Hair_xx_tint1 *Hair_xx_base (no alpha) *Hair_xx_tint2	Face Base Layer Hair Extra Hair Hat Hat	The *hair mesh* is tintable along with the hat.
Torso			
	*Torso_xxArm_base (no alpha) *Torso_xxArm_skin *Torso_xxArm_tint1 *Torso_xxArm_tint2 *Torso_xx_base (no alpha) *Torso_xx_skin *Torso_xx_tint1 *Torso_xx_tint2	Arm Arm Arm Arm Chest Chest Chest Chest	The *torso mesh* can have two tints with an additional skin layer available for short sleeves. Also multiple logos can be added in one or both of the tint layers. Consider making logos swappable, meaning a player will be able to pick a torso mesh, pick a torso texture, and then pick a torso logo. So each mesh item would have multiple textures, and each texture would have multiple logos.
Hands			
	UHand_xx_skin UHand_xx_skin UHand_xx_skin	Palm and Top of Hand Fingers and Thumb LOD Hand	The *hand mesh* will either be hands or gloves. The gloves will be tintalbe. The hands will be tintable through the skin color. Also the hands will have the ability to tint the fingernails.
Legs			
	*Legs_xx_base (no alpha)	Legs	The *legs mesh* will have two tint

	*Legs_xx_skin	Legs	layers with an additional skin layer available for shorts.
	*Legs_xx_tint1	Legs	
	*Legs_xx_tint2	Legs	
Feet			
	*Foot_xx_base (no alpha)	Feet	The *foot mesh* will have two tint layers with an additional skin layer available for sandals.
	*Foot_xx_skin	Feet	
	*Foot_xx_tint1	Feet	
	*Foot_xx_tint2	Feet	
Accessories			
	UAccGlasses_xx_base (alpha)	Glasses	The *glasses mesh* will have two tint layers with a transparent base layer for the lenses.
	UAccGlasses_xx_tint1	Glasses	
	UAccGlasses_xx_tint2	Glasses	
	Uacc_KI_base (no alpha)	KI	The *KI mesh* is not tintalbe.
	Uacc_PlayerBook_base (no alpha)	Player Book	The *PlayerBook mesh* is not tintable.

2.4. Texture Naming Convention

Texture Naming Convention:

- Non-Icon = Gender + Mesh Group + Texture Description + Layer
- Icon = Icon + Gender + Mesh Group + Texture Description

Icon:

- Icon_ = specifies the texture as an icon
- If it is not used than the texture is not an icon
- Icons do not have layers
- Some icons are used for more than one mesh group. In this case the mesh group is not stripped from the texture name

Gender:

F = Female
M = Male
U = Unisex - sometimes the Male and Female avatars share textures, i.e. the KI texture.

Mesh Group:

Used to describe the mesh group the texture will be used for. Below are the names for each mesh group:

Acc_
Face_
Hair_
Torso_
Hand_
Legs_
Foot_

If a mesh group is not specified then the texture can be used on multiple mesh groups. In this case the mesh group is stripped from the texture name.

Texture Description:

It is used to describe the texture. Below are examples of texture description names for each mesh group:

Mesh Group	*Texture Description*
Acc	_Glasses01, _KI, _PlayerBook
Face	_Goatee, _Eye, (stripped off)
Hair	_Curly01
Torso	_LeatherJacket01Arm, _LeatherJacket01
Hand	_Finger01, _Palm01, _LOD01,
Legs	_SailorPants01
Foot	_Boot01

If a mesh group describes the texture then a texture description is not needed. In this case the texture description is stripped from the texture name.

Layer:

1. base — Final alpha of the texture; used for the lenses on the glasses; non-tintable; used if a texture doesn't need to be tinted- i.e. KI
2. skin — Used for globally tinting the avatar skin through an alpha
3. skin1 — Race (African) morph target for the skin; blended with the skin layer based on the percentage of the morph
4. skin2 — Race (Asian) morph target for the skin; blended with the skin layer based on the percentage of the morph
5. skin3 — Race (Scandinavian) morph target for the skin; blended with the skin layer based on the percentage of the morph
6. skin4 — Age morph target for the skin; blended with the skin layer based on the percentage of the morph
7. tint1 — Used for tinting through an alpha
8. tint2 — Used for tinting through an alpha

Non-Icon Examples: Gender + Mesh Group + Texture Description + Layer

MLegs_Jean01_tint2 => M + Legs_ + Jeans01 + _tint2 =>

- M = Gender (Male)
- Legs_ = Mesh Group
- Jeans01 = Texture Description
- _tint2 = Layer

UTorso_LeatherJacket01_tint1 = > U + Torso + _LeatherJacket01 + _tint1 =>

- U = Gender (Unisex)
- Torso_ = Mesh Group
- LeatherJacket01 = Texture Description
- _tint1 = Layer

FFace_skin4 = > F + Face + _ skin4 =>

- F = Gender (Female)

- Face = Mesh Group
- (stripped off) = Texture Description
- _ skin4 = Layer

Icon Example: Icon + Gender + Mesh Group + Texture Description
Icon_FHair_PonyTail01 => Icon_ + F + Hair_ + PonyTail01 =>

- Icon_ = Icon
- F = Gender (Female)
- Hair_ = Mesh Group
- PonyTail01 = Texture Description

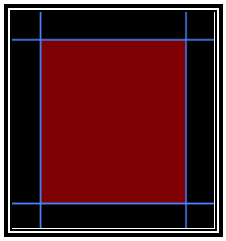
Icon Example: Icon + Gender + Texture Description
Icon_U_NoPrint => Icon_ + U + NoPrint =>

- Icon_ = Icon
- U = Gender (Unisex)
- (stripped off) = Mesh Group
- NoPrint = Texture Description

2.5. Icons

AssetMan: Main/GlobalAvatars/Files/IconGenerator

Icons are 128 x128 pixels and are then cropped at run time. There is a reason for this, just know that it is not to make life more difficult. Below is a basic icon with the red area indicating what part of the icon will be visible during game play. The IconGenerator is a Photoshop document found in AssetMan that has borders set up to match what is cropped during game play.



IconGenerator (128 x 128 with proper borders)

3. Max Files

AssetMan: Main/GlobalAvatars/MAX File

3.1. Special Characters

All the special characters divided into separate 3ds Max files with their own custom palette.

- bas.avatar.Engberg.max EngbergPalette.tga
- bas.avatar.Kodama.max Dr_Ikuro_palette.tga
- bas.avatar.RandMiller.max RandMillerPalette.tga
- bas.avatar.Sutherland.max SutherlandPalette.tga
- bas.avatar.Victor.max VictorLaxmanPalette.tga
- bas.avatar.Watson.max WatsonPalette.tga
- bas.avatar.Yeesha.max YeeshaPalette.tga

- bas.avatar.Zandi.max ZandiPalette.tga

3.2. Male and Female Files

The male and female avatars have multiple files: the **Main** file, the **SmoothBase** file, and multiple **Extra Clothes** files.

Main Files:

- bas.avatar.male.max
- bas.avatar.female.max

SmoothBase Files:

- bas.avatar.male.SmoothBase.max
- bas.avatar.female.SmoothBase.max

Extra Clothes Files:

- bas.avatar.male.01.max
- bas.avatar.female.01.max

Male and Female Files

	Main File	SmoothBase File	Extra Clothes Files
Description	Bones, Biped, Full Set of Clothes, LOD, Face Morphs, Weight Morphs, Plasma Materials	Full set of Clothes, LOD, Weight Morphs. Copied from Main File. Used for smoothing the avatar.	Extra clothes, LOD, Weight Morphs, Plasma Materials.
X-Refs	None	None	**Main File** (Bones, Biped, Low and Med LOD) **SmoothBase File** (All)

Notes:

- X-Ref in the Handle (Male or Female), High, Med, and Low dummy boxes to correctly position bones and mesh.
- X-Refed mesh items need to be skinned again in order to export correctly
- All components need to be remade

3.3. Mesh Naming Convention

Mesh Naming Convention:

- Normal = Gender + LOD + Mesh Group + Group Item + Morph Sequence
- Reference = Reference + Gender + LOD + Reference Description
- SmoothBase = SmoothBase + Gender + LOD + Mesh Group + Group Item + Morph Sequence

Reference:

Ref_ = Reference

- Defines any mesh items that are not exported
- Only used to help create other mesh items

SmoothBase:

SB_ = SmoothBase

- Defines any mesh items that are created for the SmoothBase component

Gender:

F = Female
M = Male

LOD:

H = High
M = Med
L = Low

Mesh Group:

Acc* =	Accessory items used by the avatar
• AccGlasses	Glasses
• AccKI	Player KI
• AccPlayerBook	Player Book
Hair	Head without the face; includes hats; all hair styles
Face	One morphing face
Torso	Shirts, jackets, vests. . .
*Hand = LHand, RHand	Hands, gloves. . .
Legs	Pants, shorts, cargos. . .
*Foot = LFoot, RFoot	Boots, tennis shoes, sandals. . .

Group Item:

These are numbered one through the total number of group items created for particular mesh group. For example, the Torso mesh group could have three group items:

1. Torso01 (long sleeve t-shirt)
2. Torso02 (t-shirt)
3. Torso03 (jacket)

If a group item is not specified than there is only one item in the mesh group. In this case the mesh group is stripped from the texture name.

Morph Sequence:

If the mesh item is a base mesh item and not a morph target then the morph sequence is stripped from the name. Otherwise the following is used for morph targets:

- _Mph*A
- _Mph*B

Normal Examples: Gender + LOD + Mesh Group + Group Item + Morph Sequence

MHTorso01 => M + H + Torso + 01 =>

- M = Gender (Male)
- H = LOD (High)
- Torso = Mesh Group
- 01 = Group Item
- (stripped off) = Morph Sequence

FMLegs02 = > F + M + Legs + 02

- F = Gender (Female)
- M = LOD (Med)
- Legs = Mesh Group
- 02 = Group Item
- (stripped off) = Morph Sequence

FHFace_ MphCheeksA = > F + H + Face + _MphCheeksA

- F = Gender (Female)
- H = LOD (High)
- Face = Mesh Group
- (stripped off) = Group Item
- _MphCheeksA = Morph Sequence

SmoothBase Example: SmoothBase + Gender + LOD + Mesh Group + Group Item + Morph Sequence

SB_MHTorso03 => SB_ + M + H + Torso + 03 =>

- SB_ = SmoothBase
- M = Gender (Male)
- H = LOD (High)
- Torso = Mesh Group
- 03 = Group Item
- (stripped off) = Morph Sequence

Reference Example: Reference + Gender + LOD + Reference Description

Ref_FHNeckBase = > Ref_ + F + H + Face + _MphCheeksA

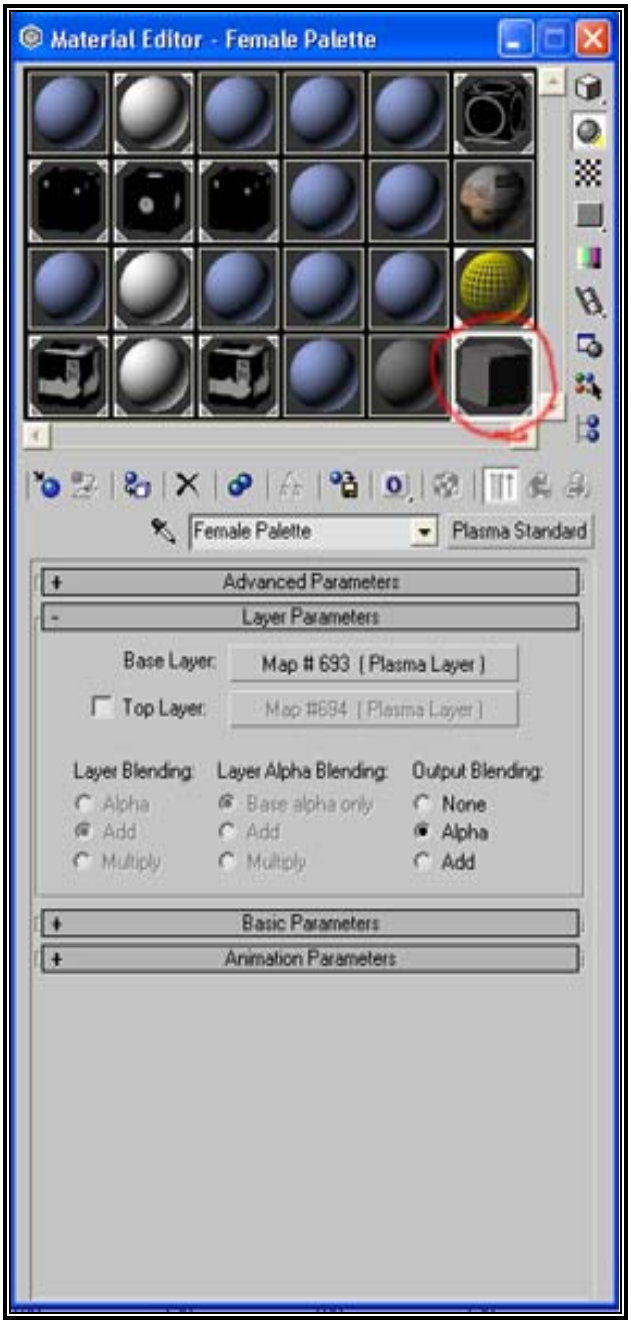
- Ref_ = Reference
- F = Gender (Female)
- H = LOD (High)
- NeckBase = Reference Description

3.4. Materials

3.4.1. Plasma Standard (Male or Female Palette)

All clothing mesh items need to have the following material:

- Plasma Standard (see below)

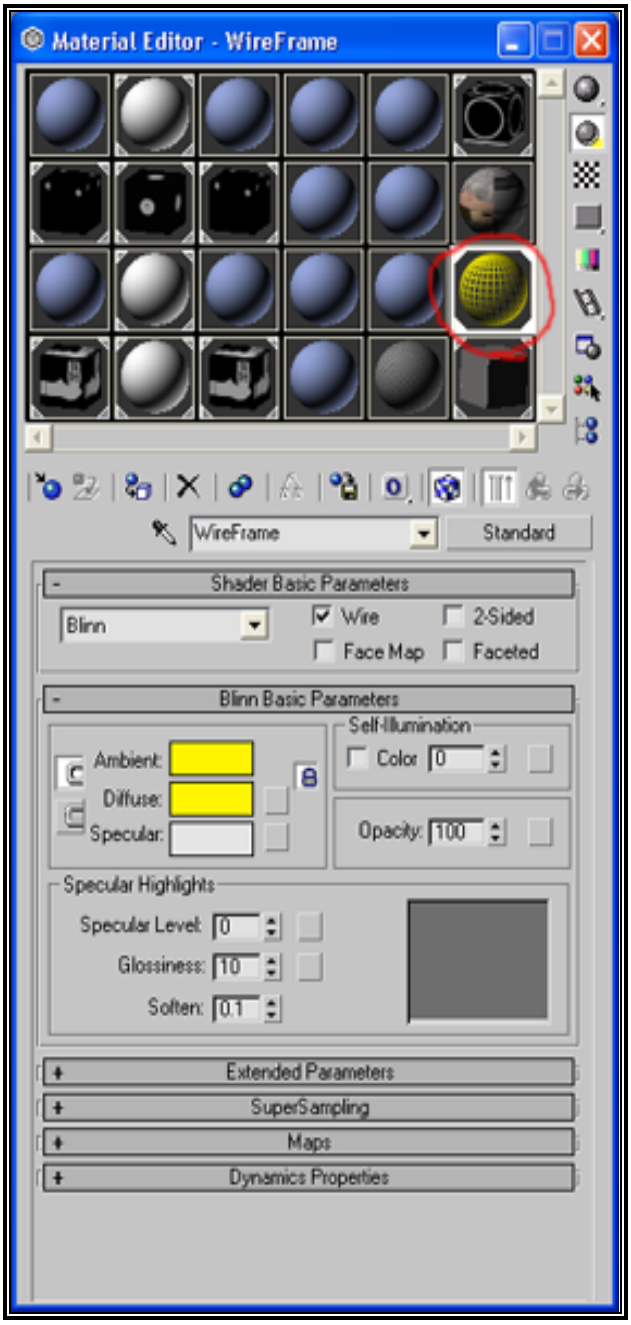


Material Editor (Plasma Standard: FemalePalette)

3.4.2. Standard (Bones, Biped, and the Collider)

All Bones, Biped, and the Collider should have the following material:

- Standard (see below)



Material Editor (Standard: WireFrame)

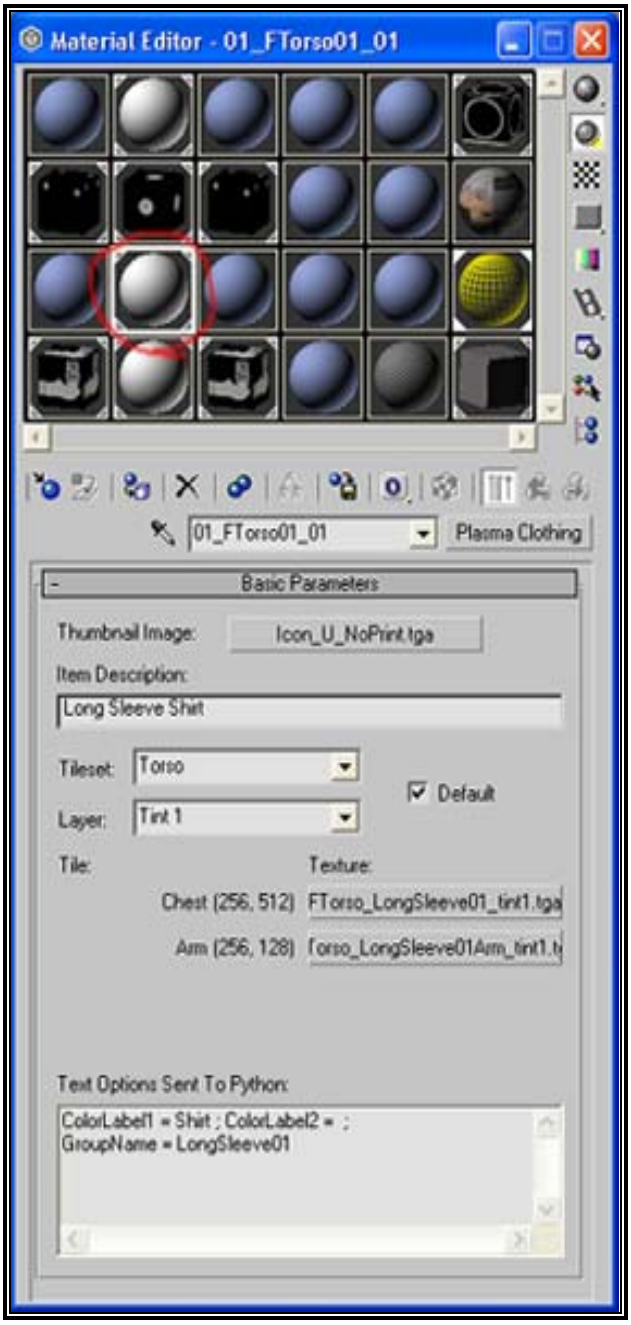
- Note: these object can have any material, but the below material helps with organization

3.4.3. Plasma Clothing (All Clothing Materials)

3.4.3.1. Overview

All Clothing Material attached to Clothing Components should be the following material:

- Plasma Clothing (see below)



Material Editor (Plasma Clothing: 01_FTorso01_01)

3.4.3.2. Material Naming Convention

Material Naming Convention:

- Non-Icon = Clothing File + Gender + Mesh Group + Group Item + Texture Number
- Icon = Icon + Texture Description

Clothing File:

These are numbered one through the total number of clothing files. The first file is the base file. For example, there could be three total clothing files:

1. 01_ (base file)

2. 02_ (extra clothes file 1)
3. 03_ (extra clothes file 2)

If a material is only found in the base file (_01) then the clothing file is not needed. In this case the clothing file is stripped from the material name.

Gender:

F = Female
M = Male

Mesh Group:

Acc* =	Accessory items used by the avatar
• AccGlasses	Glasses
• AccKI	Player KI
• AccPlayerBook	Player Book
Hair	Head without the face; includes hats; all hair styles
Face	One morphing face
Torso	Shirts, jackets, vests. . .
*Hand = LHand, RHand	Hands, gloves. . .
Legs	Pants, shorts, cargos. . .
*Foot = LFoot, RFoot	Boots, tennis shoes, sandals. . .

Group Item:

These are numbered one through the total number of group items created for particular mesh group. For example, the Torso mesh group could have three group items:

1. Torso01 (long sleeve t-shirt)
2. Torso02 (t-shirt)
3. Torso03 (jacket)

If a group item is not specified then there is only one item in the mesh group. In this case the mesh group is stripped from the texture name.

Texture Number:

These are numbered one through the total number of textures created for particular mesh item. For example, the Torso02 (i.e. t-shirt) mesh item could have three texture numbers:

1. _01 (plain t-shirt)
2. _02 (logo 1 t-shirt)
3. _03 (logo 2 t-shirt)

If a texture number is not specified then there is only one texture for the mesh item. In this case the texture number is stripped from the material name.

Texture Description (Used for Icon Materials):

It is used to describe the texture. Below are examples of texture description names for each mesh group:

Mesh Group	*Texture Description*
Acc	_Glasses01, _KI, _PlayerBook
Face	_Goatee, _Eye
Hair	_Curly01
Torso	_LeatherJacket01Arm, _LeatherJacket01
Hand	_Finger01, _Palm01, _LOD01,
Legs	_SailorPants01
Foot	_Boot01

Non-Icon Examples: Clothing File + Gender + Mesh Group + Group Item + Texture Number

01_MTorso01_02 => 01_ M + H + Torso + 01 + _02 =>

- 01_ = Clothing File
- M = Gender (Male)
- Torso = Mesh Group
- 01 = Group Item
- 01 = Group Item
- _02 = Texture Number

02_FLegs02_01= > F + M + Legs + 02

- _02 = Clothing File
- F = Gender (Female)
- Legs = Mesh Group
- 02 = Group Item
- _01 = Texture Number

FFace = > F + Face

- (stripped off) = Clothing File
- F = Gender (Female)
- Face = Mesh Group
- (stripped off) = Group Item
- (stripped off) = Texture Number

Icon Examples: Icon + Texture Description

Icon_Khakis => Icon_ + Khakis =>

- Icon_ = Icon
- Khakis = Texture Description

Icon_LeatherJacket => Icon_ + LeatherJacket =>

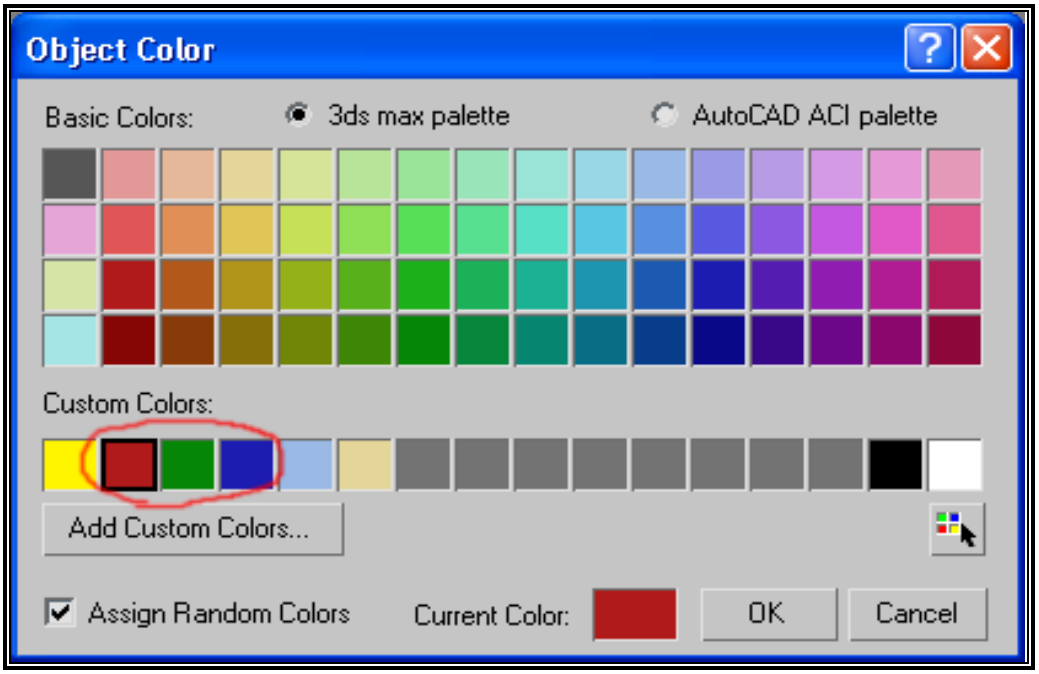
- Icon_ = Icon
- LeatherJacket = Texture Description

3.5. LOD and Polygon Count

There are three static levels of detail for the avatar:

- High (Red)
- Med (Green)
- Low (Blue)

If the mesh item is not a morph target, then the wire frame color for mesh items should be set to either red, green, or blue (see below).



LOD Wire Frame Colors (High = Red, Med = Green, Low = Blue)

The target polygon count for each level of detail should be as follows:

- High 5000 polygons
- Med 2500 polygons
- Low 312 polygons.

These are only target numbers. The number of polygons will vary based on the avatar, but should be relatively close to the target polygon count. Below is a sample polygon count for an avatar:

Sample Avatar Polygon Breakdown

------- High Poly Avatar ----------

04%	Accessories	152
06%	Hair	268
23%	Face	1016
24%	Torso	1042
30%	Both Hands	1318
09%	Legs	384
04%	Both Feet	152
100%	Current Avatar	4636
108%	Target Avatar	5000

Total Bones Used: 75

------- Medium Poly Avatar ----------

04%	Accessories	51
16%	Hair	178
28%	Face	318

20%	Torso	224
09%	Both Hands	104
16%	Legs	184
07%	Both Feet	76
100%	Current Avatar	1239
101%	Target Avatar	1250

Total Bones Used: 23

------- Low Poly Avatar ----------

04%	Accessories	12
17%	Hair	53
12%	Face	38
28%	Torso	90
10%	Both Hands	32
16%	Legs	51
13%	Both Feet	40
100%	Current Avatar	340
92%	Target Avatar	312

Total Bones Used: 17

3.6. Dummy Boxes and the Physics Hull

Dummy boxes are used to attach Plasma components that are used during export. The **Handle** (Male or Female) has the majority of components while the **High** LOD dummy box has morph components associated with it.

LOD Dummy boxes consist of:

- High
- Med
- Low

LOD linking rules:

- All high mesh items should be linked to the High LOD dummy box
- All med mesh items should be linked to the Med LOD dummy box
- All low mesh items should be linked to the Low LOD dummy box

The **Collider** is used as the simple physics shape for the avatar.

There are five dummy boxes used in the game environment for footsteps, swimming, etc.:

- Print_L Foot
- Print_R Foot
- Print_L Hand
- Print_R Hand
- Print_Trunk

The **Convergence** dummy box used as a seek object for animations.
The **BookHandle** dummy box used to help animate the Player book.

3.7. Pivot Points

The **Collider** should have the following pivot point setup:

- Position = [X = 0, Y = 0, Z = 0]
- Rotation = [X = 0, Y = 0, Z = 0]

All **dummy boxes** except the BookHandle should have the following pivot points:

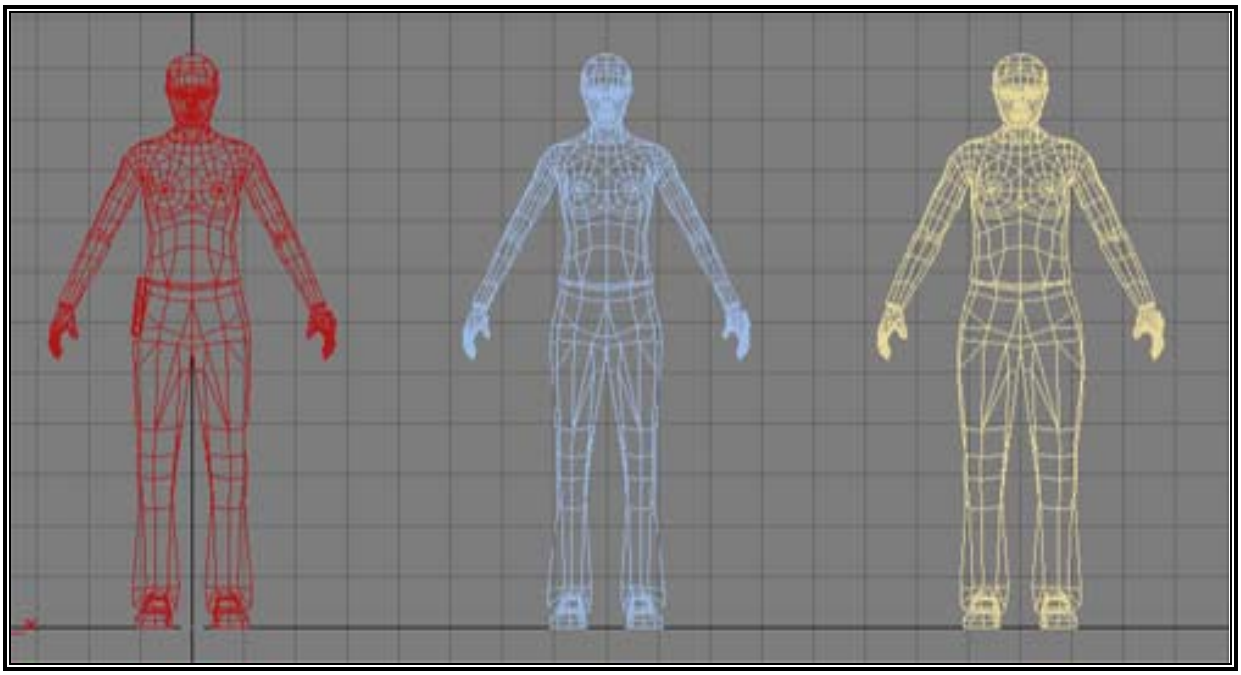
- Rotation = [X = 0, Y = 0, Z = 0]

The **BookHandle** is used to help animate the Player book, so the position and rotation of the dummy box is variable.

If a mesh item is created in the correct position to the base avatar (the Red Avatar, see below) then all mesh pivot points should be the following:

- Position = [X = 0, Y = 0, Z = 3.5]
- Rotation = [X = 0, Y = 0, Z = 0]

Mesh items can be can be moved for easier organization (i.e. the morph targets) and therefore the pivot point will change. Only the X and Z axis should be used translate mesh items within 3ds Max.
Below is an example of the organization of a max file and how the pivot points should be set.



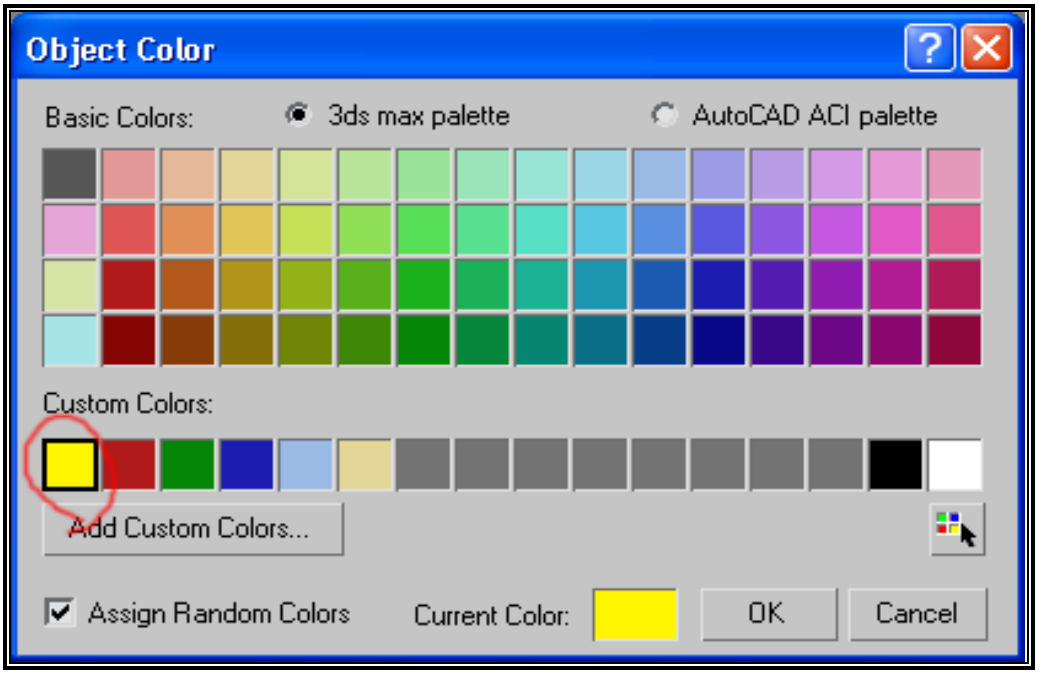
Pivot Point Position for avatar (Base Avatar = Red Avatar = [X = 0, Y = 0, Z = 3.5], Light Blue Avatar = [X = 4, Y = 0, Z = 3.5], Light Yellow Avatar = [X = 8, Y = 0, Z = 3.5])

Notice that if the Light Blue avatar and the Light Yellow avatar were translated to the X position of zero then all the pivot point would be:

- Position = [X = 0, Y = 0, Z = 3.5].

3.8. Bones

The wire frame color for bones should be set to yellow (see below).



Bone Wire Frame Color

3.8.1. Bone Naming Convention

Bones have “Bone_” followed by the name of the bone. The following are the names of all the bones:

Bone_BookHinge1
Bone_BookHinge2
Bone_Head
Bone_Jaw
Bone_LBrowInner
Bone_LBrowOuter
Bone_LCalf
Bone_LCheek
Bone_LClavicle
Bone_LEye
Bone_LEyeLid1
Bone_LEyeLid2
Bone_LFoot
Bone_LForearm
Bone_LHand
Bone_LMiddle1
Bone_LMiddle2
Bone_LMiddle3
Bone_LMouthCorner
Bone_LMouthLower
Bone_LMouthUpper
Bone_LPinky1
Bone_LPinky2
Bone_LPinky3
Bone_LPointer1
Bone_LPointer2
Bone_LPointer3

Bone_LRing1
Bone_LRing2
Bone_LRing3
Bone_LThigh
Bone_LThumb1
Bone_LThumb2
Bone_LThumb3
Bone_LToe
Bone_LUpperArm
Bone_Neck
Bone_Pelvis
Bone_RBrowInner
Bone_RBrowOuter
Bone_RCalf
Bone_RCheek
Bone_RClavicle
Bone_REye
Bone_REyeLid1
Bone_REyeLid2
Bone_RFoot
Bone_RForearm
Bone_RHand
Bone_RMiddle1
Bone_RMiddle2
Bone_RMiddle3
Bone_RMouthCorner
Bone_RMouthLower
Bone_RMouthUpper
Bone_RPinky1
Bone_RPinky2
Bone_RPinky3
Bone_RPointer1
Bone_RPointer2
Bone_RPointer3
Bone_RRing1
Bone_RRing2
Bone_RRing3
Bone_RThigh
Bone_RThumb1
Bone_RThumb2
Bone_RThumb3
Bone_RToe
Bone_RUpperArm
Bone_Root
Bone_Spine0
Bone_Spine1
Bone_Spine2

Note: “BookHandle” is not a bone because it is not skinned to anything. It is a dummy box that is the parent of “Bone_BookHinge1” and “Bone_BookHinge2”.

3.8.2. Bone LOD

There are a total of 75 bones used for the avatar. At high LOD all 75 bones are used to calculate mesh location. At medium LOD only 26 of the original 75 bones are used and only 18 of the original 75 bones are used in calculating low LOD mesh location.

High (75)	
	All Bones
Med (26)	
All Bones Except:	
	Finger Bones (30)
	Facial Bones (19)
Low (18)	
All Bones Except:	
	Finger Bones (30)
	Facial Bones (19)
	Bone_LFoot
	Bone_LHand
	Bone_LToe
	Bone_Neck
	Bone_RFoot
	Bone_RHand
	Bone_RToe
	Bone_Spine1

High, Med, and Low LOD Bones

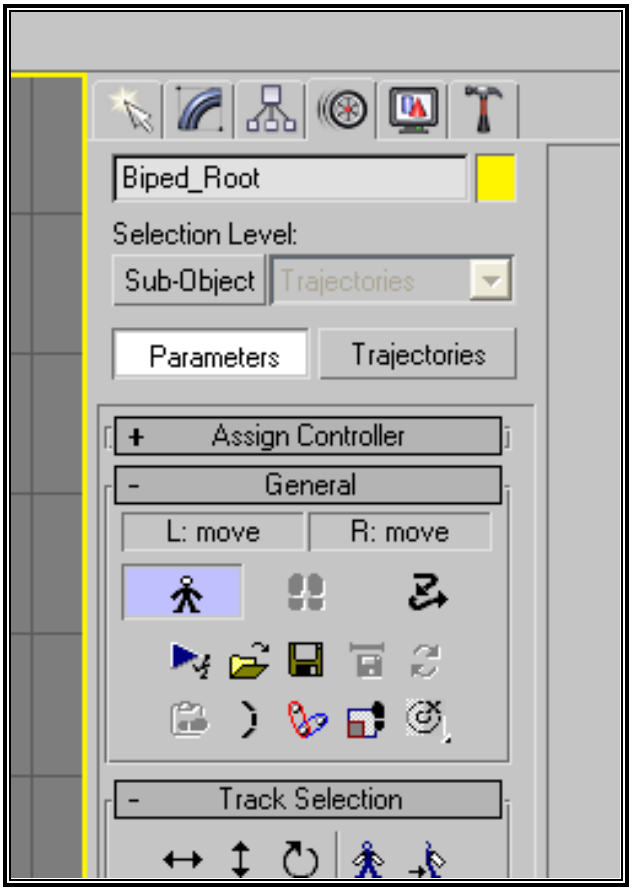
3.9. Biped

3.9.1. Overview

The biped is used to position the bones in order to check skinning. In order for this to work the biped must be linked to the bones. After checking the avatar the biped should no longer be linked to the bones and the bones should be re-linked in the correct hierarchy. There are two type of files used with biped: biped files (.bip) and figure files (.fig). Figure files load a static pose for the avatar and the biped files load an animation for the biped.

The idea is to create and skin the avatar based on the figure mode and then to test the skinning by loading biped files (animation files). Linking the biped to the bones and re-linking the bones in the correct hierarchy is made easy through the use of max scripts.

The wire frame color for the biped should be the same as the bones (yellow).



Biped Figure Mode (Biped_Root must be selected)

3.9.2. Naming Convention

Each biped object has “Biped_” followed by the name of the biped object. The following are the names for the biped objects:

Biped_Footsteps
Biped_Head
Biped_HeadNub
Biped_LCalf
Biped_LClavicle
Biped_LFoot
Biped_LForearm
Biped_LHand
Biped_LMiddle1
Biped_LMiddle2
Biped_LMiddle3
Biped_LMiddleNub
Biped_LPinky1
Biped_LPinky2
Biped_LPinky3
Biped_LPinkyNub
Biped_LPointer1
Biped_LPointer2
Biped_LPointer3
Biped_LPointerNub

Biped_LRing1
Biped_LRing2
Biped_LRing3
Biped_LRingNub
Biped_LThigh
Biped_LThumb1
Biped_LThumb2
Biped_LThumb3
Biped_LThumbNub
Biped_LToe
Biped_LToeNub
Biped_LUpperArm
Biped_Neck
Biped_Pelvis
Biped_RCalf
Biped_RClavicle
Biped_RFoot
Biped_RForearm
Biped_RHand
Biped_RMiddle1
Biped_RMiddle2
Biped_RMiddle3
Biped_RMiddleNub
Biped_RPinky1
Biped_RPinky2
Biped_RPinky3
Biped_RPinkyNub
Biped_RPointer1
Biped_RPointer2
Biped_RPointer3
Biped_RPointerNub
Biped_RRing1
Biped_RRing2
Biped_RRing3
Biped_RRingNub
Biped_RThigh
Biped_RThumb1
Biped_RThumb2
Biped_RThumb3
Biped_RThumbNub
Biped_RToe
Biped_RToeNub
Biped_RUpperArm
Biped_Root
Biped_Spine0
Biped_Spine1
Biped_Spine2

3.10. Custom Max Tools (macroscripts)

C:\3dsmax4\ui\macroscripts

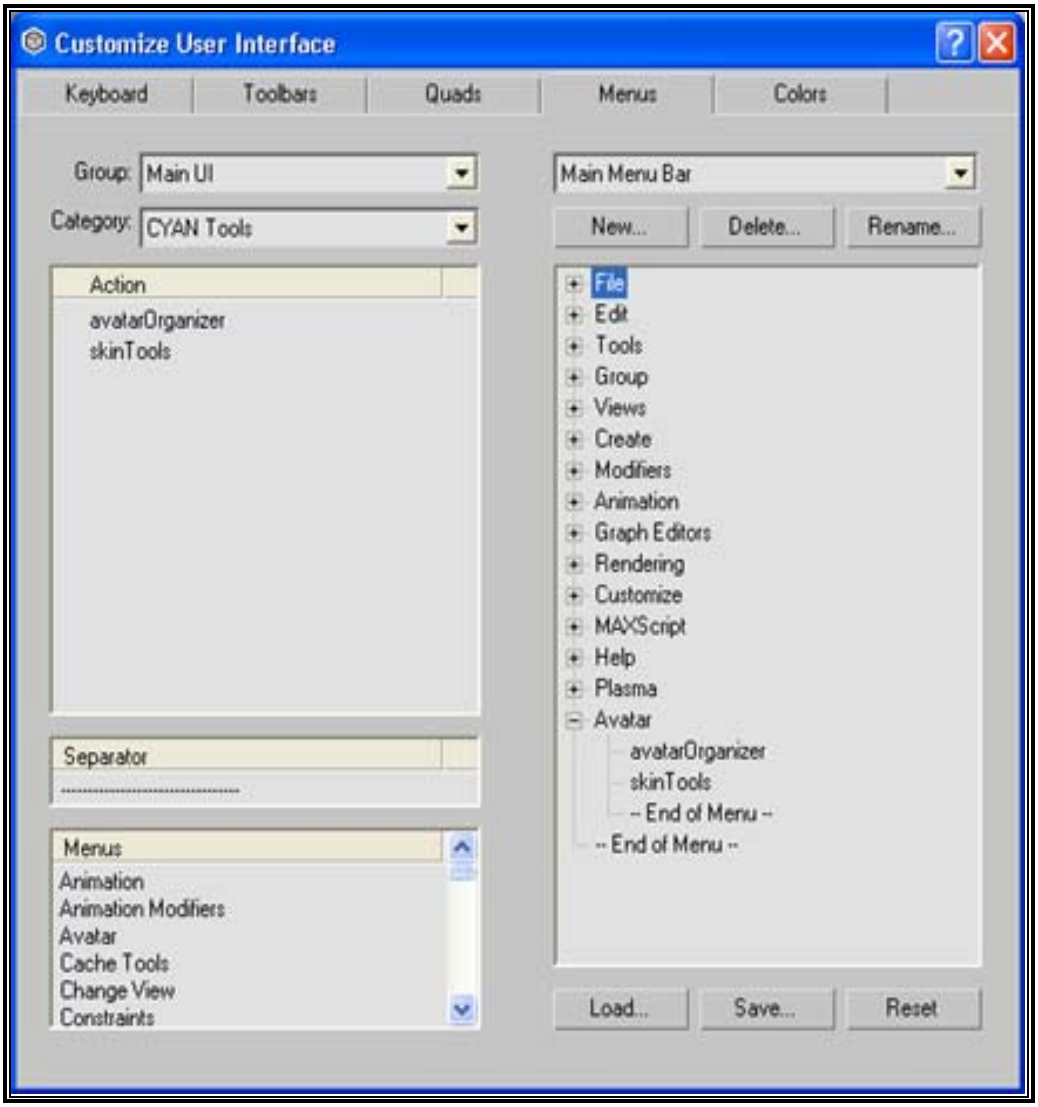
3.10.1. Adding Custom Max Tools (macroscripts)

Macroscripts are scripts written in maxscript that pug into 3ds Max and enable the user to create custom tools. Macroscripts must be placed in the following directory: C:\3dsmax4\ui\macroscripts. Once the scripts are placed in the correct location 3ds Max will be able to access the macroscripts when creating custom menus.

To create a custom menu to include custom tools (macroscripts) do the following:

- Select **Customize** file
- Choose **Customize User Interface**
- Select **Menus** tab
- Hit the **New…** button and name the new menu "Avatar" or something similar
- Make sure **Group:** is set to **Main UI**
- Make sure **Category:** is set to **CYAN Tools**
- Drag the scripts under **Action** to the menu new menu.

To create a custom menu to include custom tools go to the customize file and select "Customize User Interface". The easiest way to use the macroscripts (custom tools) is to create a custom menu called "Avatar" or something similar.



Custom User Interface

3.10.2. Using Custom Max Tools (macroscripts)

There are four custom tools that aid in avatar creation:

- Avatar Organizer
- Avatar Polygon Info
- Pivot Copier
- Skin Tools

The **Avatar Organizer** tool is used to manage the avatar files. It is similar to having selection sets, but it will dynamically update when new mesh items are added as long as the new mesh items following the correct naming convention.

The **Avatar Polygon Info** tool is used to compare an avatar clothing mesh to a target mesh of the same type. This is useful for managing polygons.

The **Pivot Copier** is used to copy the position of one mesh item to other mesh items.

Skin Tools are used to simplify the process of skinning a mesh.

Avatar Polygon Info button description:

- **Update** is used after a new mesh is selected and the polygon information needs to be updated for the newly selected mesh item

Avatar Organizer button description:

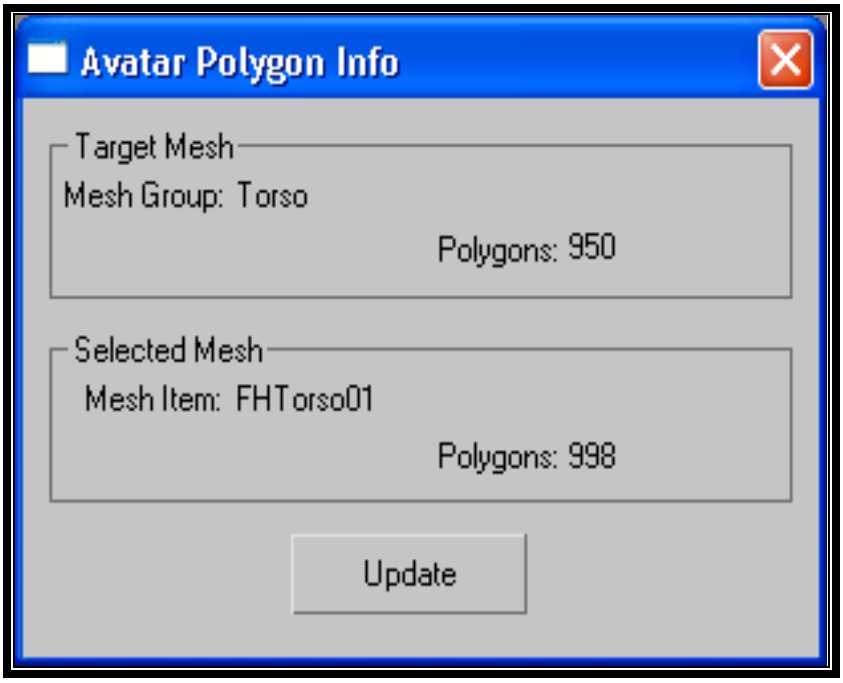
- **Mesh** is used to hide or unhide the specified mesh items.
- **Bones** is used to hide or unhide the specified bones.
- **Link to Bones** is used to re-link the bones in the correct hierarchy.
- **Biped** is used to hide or unhide the specified bones.
- **Link to Biped** is used to link the bones to the biped.
- **Select Biped** is used to select Biped_Root.
- **Dummy Boxes** is used to hide or unhide the dummy boxes.

Pivot Copier button description:

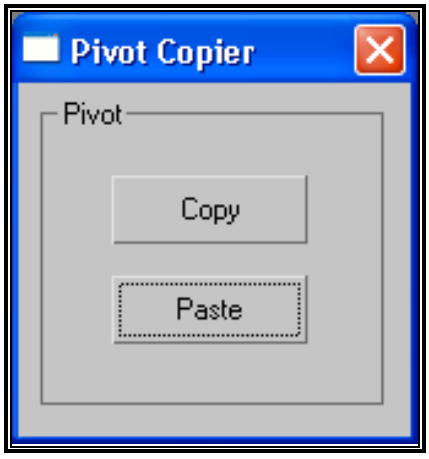
- **Copy** is used to copy the pivot of the selected mesh item
- **Paste** is used to paste a copied pivot to the selected mesh item

Skin Tools button description:

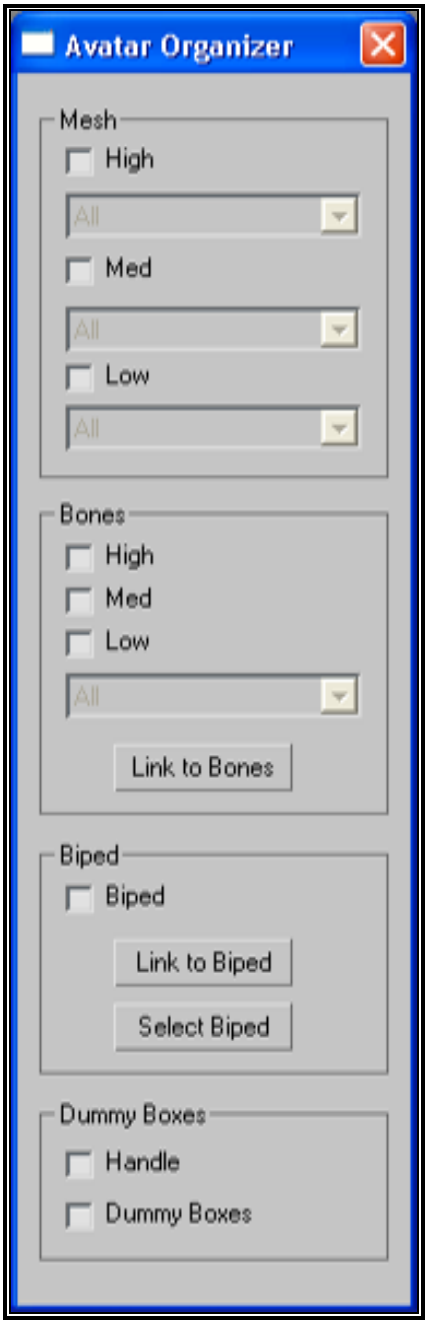
- **Save Weights Normal** is used to save the weights in the current bone order
- **Save Weight Alphabetized** is used to save the weights in alphabetical order. This is useful if the bones in the skin modifier are not in alphabetical order. If the bones are in alphabetical order than the process of skinning is simplified.
- **Replace** is used to replace the weights. The mesh must have the same number of vertices as well as the same correctly ordered bones compared to the weights file being used to replace the weights.
- **Mirror** is used to duplicate the weights of one side a symmetrical mesh to the other side.
- **Select** is used to select certain vertices within the tolerance specified. Then the selection can be set to a new global weight. This helps to simplify the weights.
- **Simplify** is used to simplify the weights by rounding all the absolute weight effects for the all the bones.
- **Zero** is used to remove any weights that have any absolute weight effect of the specified tolerance.



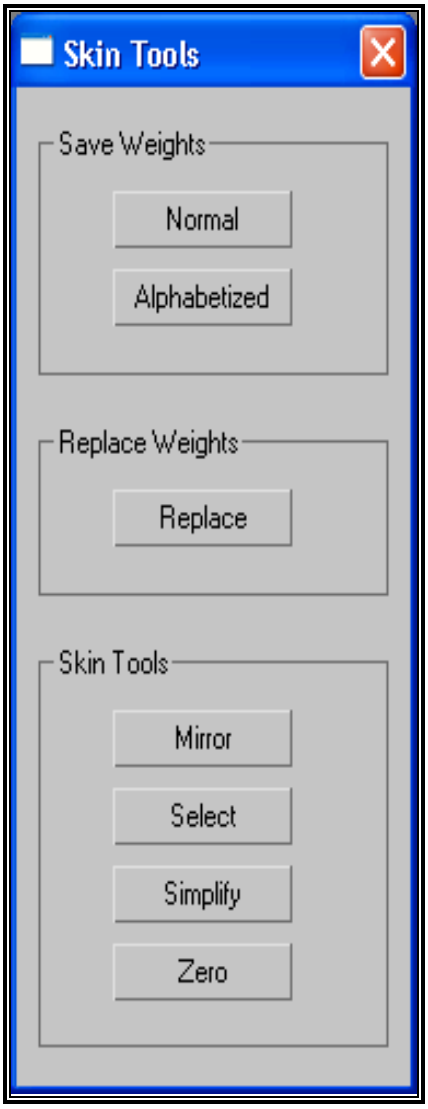
Avatar Poly Info



Pivot Copier



Avatar Organizer



Skin Tools

4. Plasma

4.1. Components

What are components?
Components are used to define rules for the avatar within Plasma.

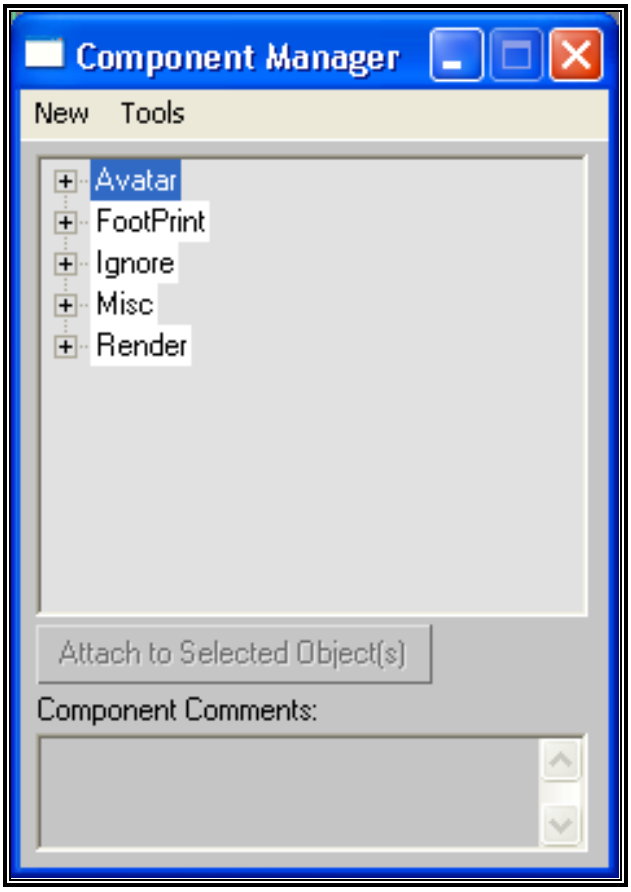
What is the Component Manger used for?
To attach components to the avatar.

Where is the Component Manager found?
Select the **Plasma** menu and then select **Component Manger**.

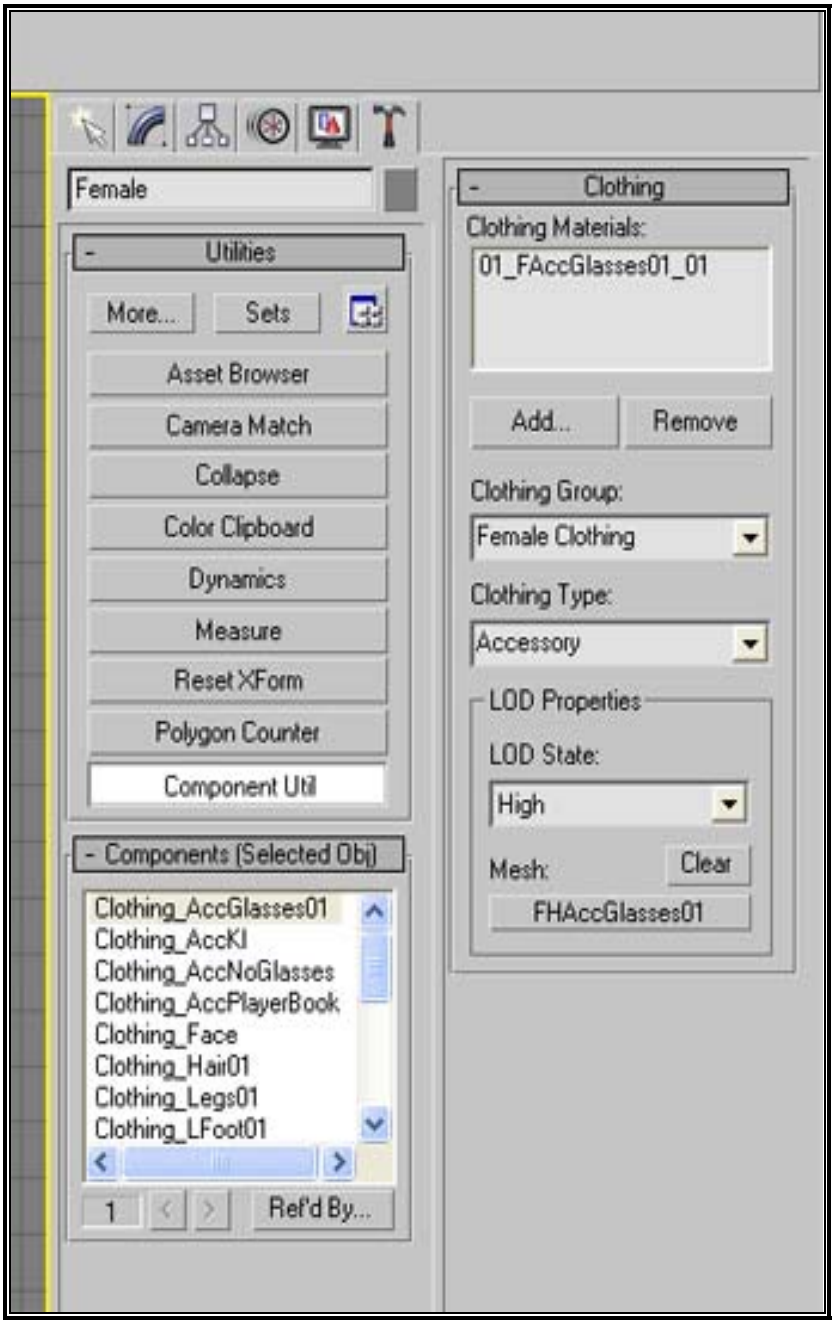
Where are the components attached?
Components are attached to dummy boxes, usually the handle (Male or Female dummy box) or the High LOD dummy box.

How are components manipulated?
Components can be manipulated using the **Component Utility**.

Where is the **Component Utility** found?
First, a dummy box with components attached to it must be selected. Second, the **Utility** panel must be selected with the **Component Utility** as the selected utility.



Component Manger



Component Utility: Female dummy box

4.2. Updating Plasma

Changes to Plasma are made on a daily basis which is why it is important to update Plasma daily. This process is easy:

- Close 3ds Max
- Remove all contents in the Plasma folder (C:\Plasma)
- Run PlasmaInstaller.exe (\\Sektor\PlasmaInstall\Latest\PlasmaInstaller.ex) to get the latest Plasma (pick the most current date)
- Get all the data from \\plasmabuild\Nightly and pick the date to match the date used with the PlasmaInstaller.exe.

- Open Main-Internal\Data and copy the **Dat** directory over the local **Dat** directory in the Plasma folder (C:\Plasma)
- Run the **plCleintSetup.exe** (C:\Plasma\plCleintSetup.exe)
- Login with user name and password to the server: **parabletest.cyanworlds.com**
- Check **Link to Age** and select **Avatar Customization**
- Hit the **Play** button to run the game once from the server, this allow everything to download
- Close the **plClient.exe**
- Run the **plCleintSetup.exe** (C:\Plasma\plCleintSetup.exe)
- Login with user name and password to the server: **Single Player**
- Select the **Clients & Datasets** tab and uncheck both **Use Data Server** boxes; this will allow local changes to not be overwritten.

4.3. Exporting to Plasma

Changes to max files need to be verified before checking them back into AssetMan. Do the following before checking a max file back into AssetMan:

- Open a 3ds Max file that needs to be updated
- Make the correct changes and **Export** the to the local Plasma directory (C:\Plasma\)
- Give the exported file a name followed by ".prd"
- If the file exports with no errors then run the **plClient.exe** to check the updated changes
- After the max file is error free check it back into AssetMan

5. Avatar Clothes

5.1. Overview

Creating new clothes for the avatar involves the following process:

Mesh:

- Create a new mesh in High, Med, and Low LOD
- Manipulate the UVP mapping to look correct for all LOD
- Skin all LOD using the right bones with the right influence
- Create any morph targets needed

Plasma Clothing Materials:

- Create new Plasma Clothing materials
- Point the materials to look up textures from AssetMan
- Make sure each material has a unique name

Avatar Components:

- All new clothes need an **Avatar Clothing** component and a **Page Info** component
- **Smoothing Base** and **Snap to Base** are used to smooth the normals and weld seam points for new clothes
- **Morph Layer** and **Morph Sequence** are used to set up morphs

5.2. Mesh Tips

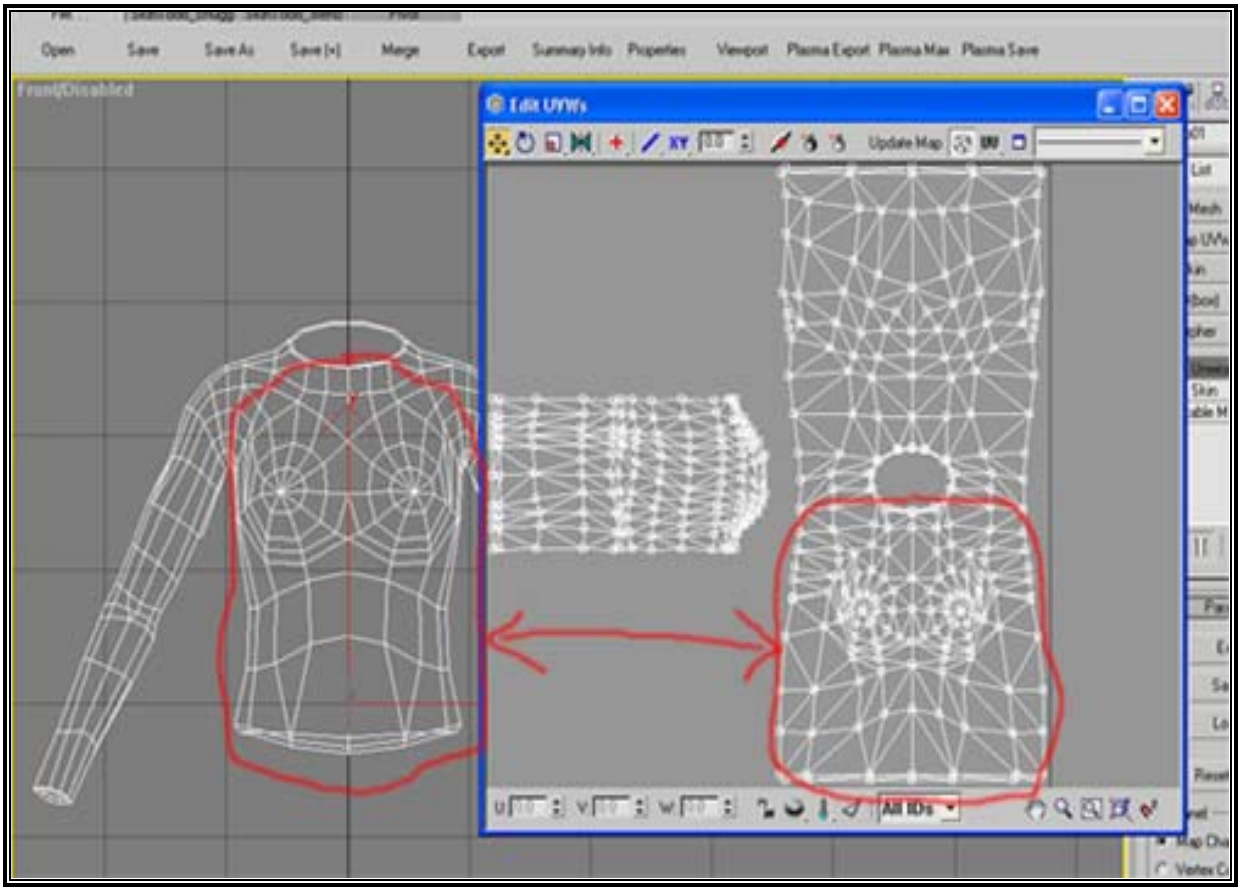
5.2.1. Creating a New Mesh:

- Use an existing mesh as a reference
- Only build one side of a symmetrical mesh:
  - Mirror a copy of the mesh
  - Flip the UVW mapping

    - Weld the new mesh to the original
- Make sure all the seams between clothing items are correct
- Make sure the geometry is correct before working on the UVW mapping
- Use the **MultiRez** modifier when creating Med LOD models for most of the work
- Look at similar existing Med LOD mesh for reference
- The Low LOD models are generic, therefore will work for most new clothes
- If new Low LOD models need to be created then use existing Low LOD models as reference
- Parent the correct LOD dummy box to any new mesh items

5.2.2. UVW Mapping:

- Look at an existing similar mesh and see what the UVW mapping looks like
- Match up the UVW to the mesh wire frame (see below)



UVW mapping should match up with the mesh wire frame

- Make sure to put seams in the most unnoticeable places

Tips for **skinning** a new mesh:

- Use the **Avatar** Organizer to make sure the correct bones are loaded for the mesh
- Use the **Skin Tools**:
    - Use the **Normal** tool to save weights in the current order or use the **Alphabetical** tool to save weights in alphabetical order
    - Use the **Replace** tool to replace the weights of a mesh
    - Only skin one side of a symmetrical mesh, then use the **Mirror** tool
    - Use the **Zero** tool to remove minimal influence from bones

- Use the **Select** tool to select and set vertices to simplify the weight
- Use the **Simplify** tool for Med and Low LOD models

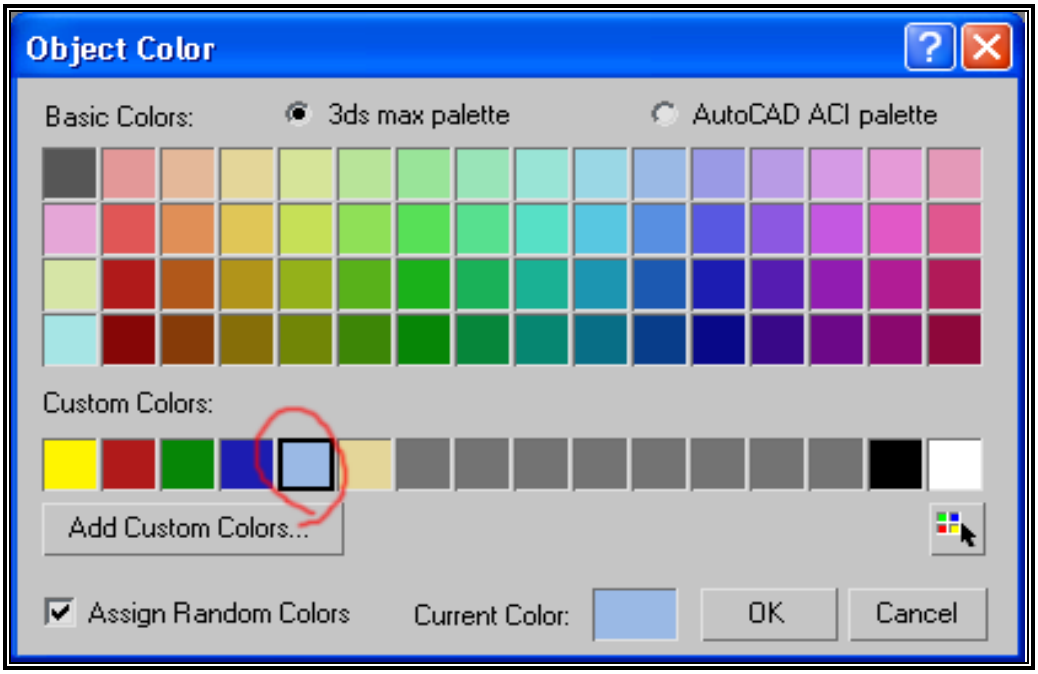
Fastest way to create a new clothing item:

- **Save** the weights of a similar mesh
- **Duplicate** the similar mesh and skin the duplicated mesh the same as the original
- **Replace** the weights with the saved weights file
- **Manipulate** vertices:
  - Added vertices will need to be re-skinned
  - If vertices are removed the UVW mapping will need to be updated
- Use the **MultiRez** modifier or duplicate an existing model to create the LOD
- Create **morph targets** by duplicating the base mesh and manipulating it
- Create new **textures** and place them in AssetMan
- Create new Plasma Clothing **materials** and have them point to AssetMan textures
- Add components:

  Avatar Clothing
  - **Page Info**
  - **SmoothingBase** and **Snap to Base**
  - **Morph Layer** and **Morph Sequence**
  - Etc.

5.2.3. Morph Targets:

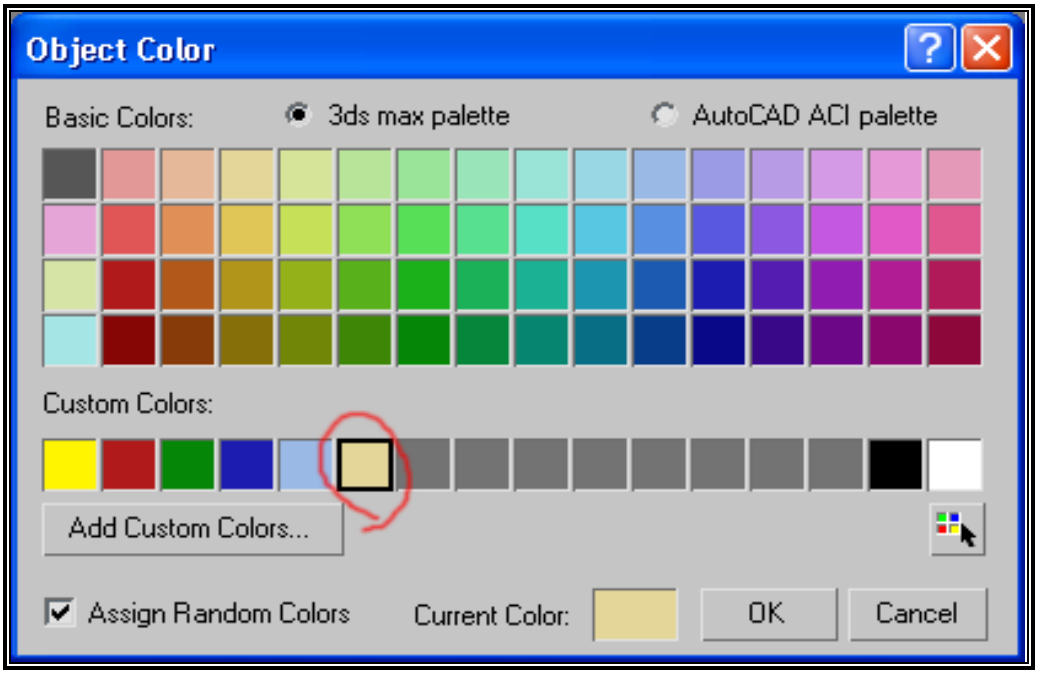
- Finish the base mesh first with the correct skinning and UVW mapping
- Duplicate the base mesh
- Reset the X-Forms on the new mesh and collapse the modifier stack
- Rename the mesh correctly: *MphA
- Set the wire frame color to light blue (see below)



***MphA wire frame color**

- Duplicate the *MphA mesh and name it correctly: *MphB

- Set the wire frame color to light yellow (see below)

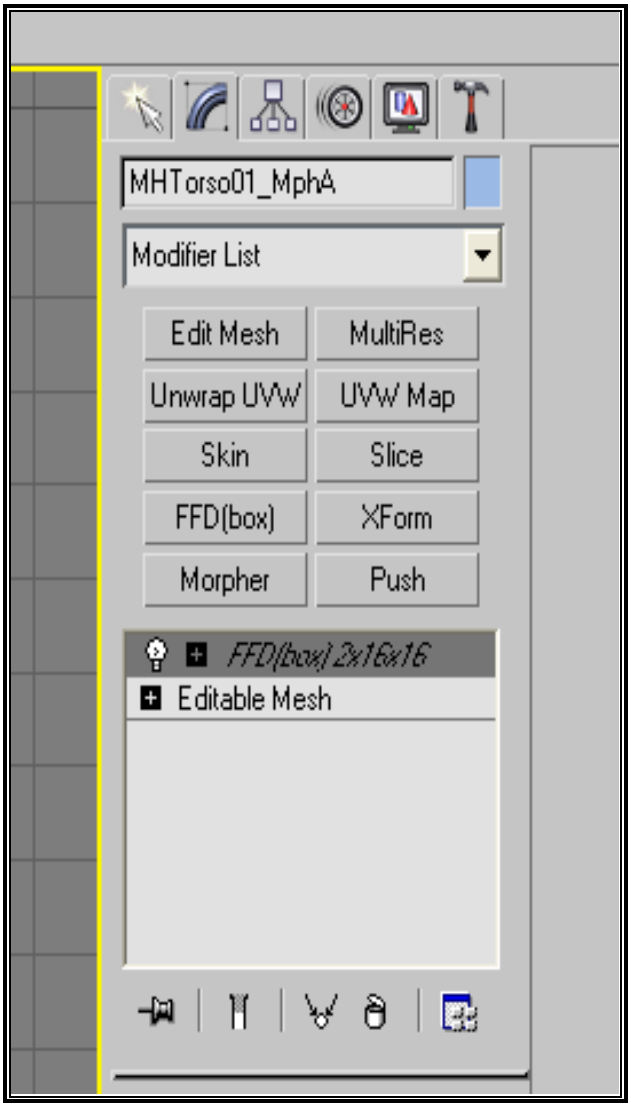


***MphB wire frame color**

- Set up should be similar to the below example of a Torso mesh

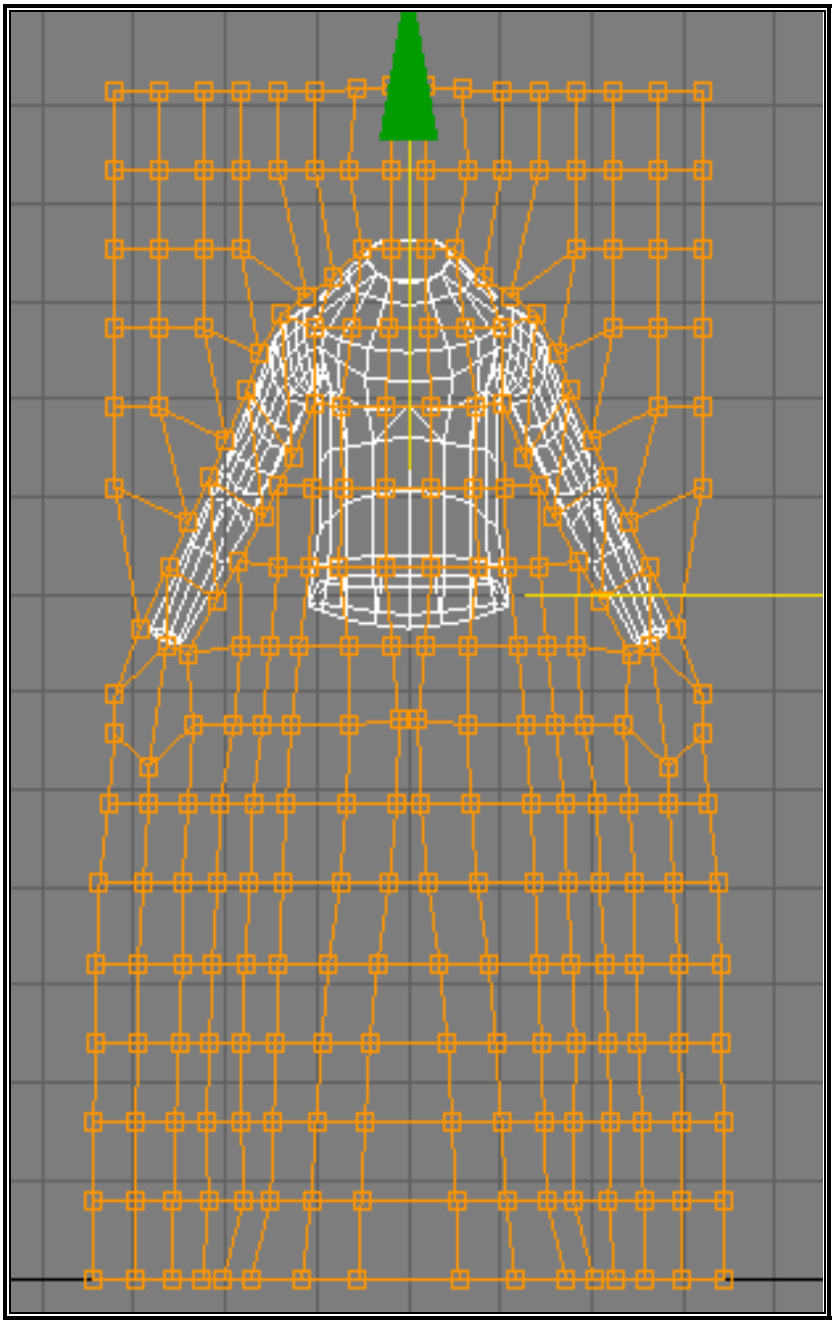
Morph Sequence example (Red = Base Mesh, Light Blue = *MphA, Light Yellow = *MphB)

- Select another *MphA clothing mesh and select the modifier tab
- Copy the FFD(box) 2x16x16 modifier (see below)



FFD(Box) 2x16x16 modifier

- Paste the modifier as an instance onto the new *MphA mesh
- The result should like the following:

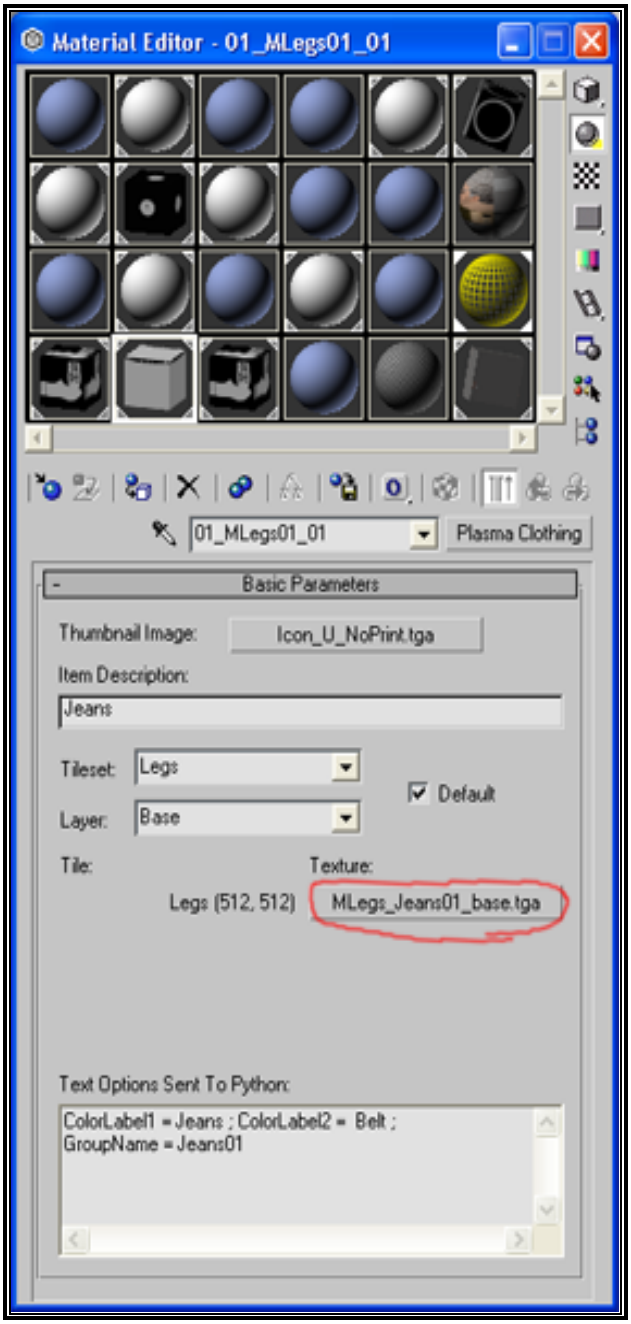


FFD(Box) 2x16x16 modifier pasted as a reference

- Select another *MphB clothing mesh and select the modifier tab
- Copy the FFD(box) 2x16x16 modifier
- Paste the modifier as an instance onto the new *MphA mesh
- As long as all the meshes share the same pivot point and all X-Forms have been reset, then the modifier should morph the mesh correctly
- Since the modifier is a reference, any changes to the modifier will be carried out to all the referenced modifiers

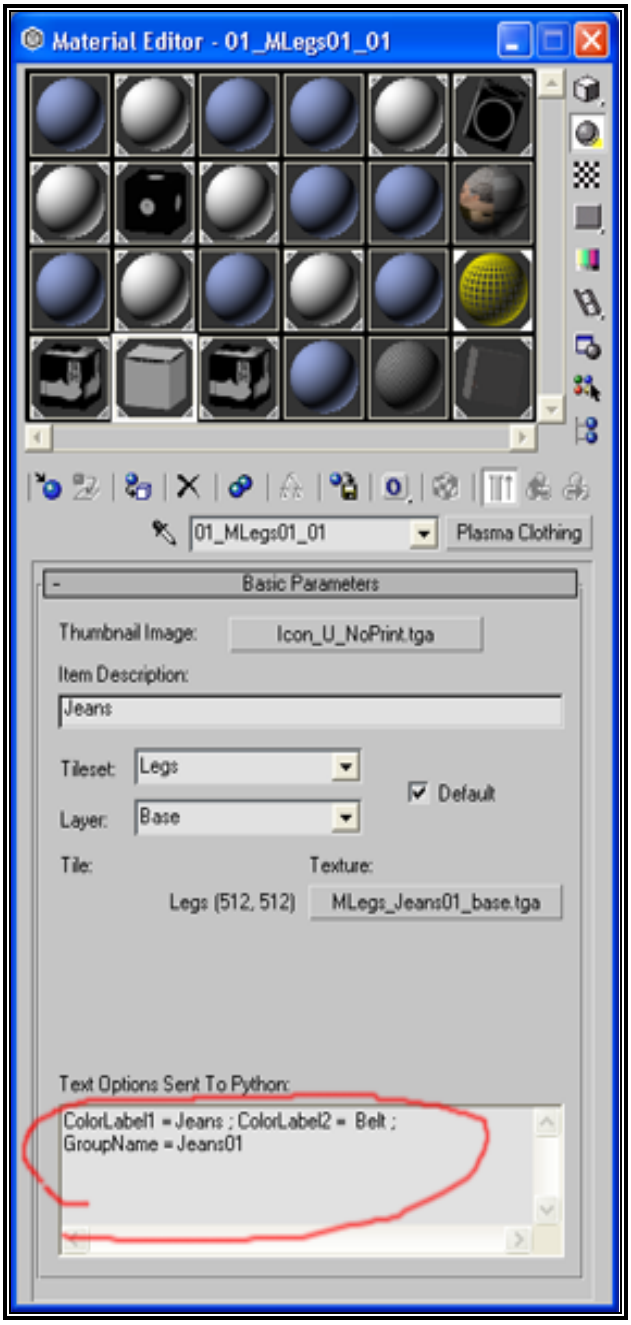
5.3. Plasma Clothing Material Tips:

- Ctrl-click the texture link buttons to remove the link
- Shift-click the texture link buttons to link to a local texture
- Click the texture link buttons to link to a texture from Asset man (see below)



Plasma Clothing: MLegs_Jeans01_base (linked to AssetMan)

- Make sure all materials have unique names
- Make sure all python code is correct (see below)



Plasma Clothing: MLegs_Jeans01_base (text options sent to python)

5.4. Avatar Component Tips:

- Make sure to update Plasma in order to use the most updated components
- Use the **Component Manager** to attach and manage components
- Select something with component attached to it and then use the **Component Utility**
- Look at the following as a reference for setting up components:
    - Similar existing mesh
    - Handle (Male or Female)
    - High LOD dummy box