

# Human.xml in Barotrauma - Wiki Guide

## Introduction

The `human.xml` file in Barotrauma is a crucial configuration file that defines all attributes, behaviors, and appearances of human characters in the game. This guide provides a comprehensive breakdown of the file structure, explaining each element and attribute in detail to help modders customize human characters effectively.

## File Structure Overview

The `human.xml` file follows an XML structure and starts with a root `Character` element containing numerous child elements that define various aspects of human characters, including:

- Basic attributes and behaviors
- Health and damage system
- Inventory configuration
- Character customization options
- Sounds and animations
- Visual effects

## The `Character` Element

The root element that contains all character definitions. It has numerous attributes that define the fundamental properties of humans in the game.

## Key Attributes

Attribute	Description	Value
<code>specifiertags</code>	Enables tag-based character specification	<code>true</code> enables the tag system
<code>skincolors</code>	Available skin color options	List of hex color codes with weights
<code>haircolors</code>	Available hair color options	List of hex color codes with weights
<code>facialhaircolors</code>	Available facial hair color options	List of hex color codes with weights
<code>SpeciesName</code>	Defines the species identifier	<code>"Human"</code>
<code>Humanoid</code>	Determines if bipedal animation is used	<code>"True"</code> enables human-like movement
<code>CanInteract</code>	Ability to interact with items and environment	<code>"True"</code> enables interaction
<code>CanClimb</code>	Ability to climb ladders	<code>"True"</code> enables climbing
<code>Husk</code>	Whether character is a husk by default	<code>"False"</code> for normal humans
<code>UseHuskAppendage</code>	Whether husk visuals can be applied	<code>"True"</code> enables husk transformation visuals
<code>NeedsAir</code>	Whether character requires oxygen	<code>"True"</code> means humans will drown without air
<code>NeedsWater</code>	Whether character requires water	<code>"False"</code> means humans don't need water to survive
<code>CanSpeak</code>	Ability to communicate	<code>"True"</code> enables voice communication
<code>BloodDecal</code>	Type of blood decal to display	<code>"blood"</code> defines the blood texture
<code>CanEat</code>	Whether character can consume items	<code>"True"</code> enables eating
<code>EatingSpeed</code>	Speed of item consumption	Higher values = faster eating
<code>UsePathFinding</code>	Whether AI uses pathfinding	<code>"True"</code> enables navigation

## Important Functional Attributes

Attribute	Description	Value
<code>Noise</code>	Base noise level generated by character	Higher values = more detectable by monsters
<code>Visibility</code>	How visible the character is	Higher values = more easily spotted
<code>HideInSonar</code>	Whether hidden from sonar	<code>"False"</code> means humans appear on sonar
<code>HideInThermalGoggles</code>	Whether hidden from thermal vision	<code>"False"</code> means humans are visible on thermal
<code>DisableDistance</code>	Distance at which character is unloaded	Higher values keep characters loaded at greater distances

## Character Customization

`Heads` Element

Defines the available head sprites for character creation.

xml

Copy

```
<Heads>
  <Head tags="head1,male" sheetindex="0,0" />
  <!-- Additional head definitions... -->
</Heads>
```

Each `Head` element has:

- `tags`: Identifies the head style and gender compatibility
- `sheetindex`: Position in the sprite sheet (column, row)

### `HeadAttachments` Element

Defines wearable elements like hair, beards, and face accessories.

### Hair Configuration Example:

xml

Copy

```
<Wearable type="Hair" tags="male" replacewhenwearinghat="1">
  <sprite name="Hair 4" texture="Content/Characters/Human/Human_male_hair.png" sheetindex="3,0" />
</Wearable>
```

Key attributes:

- `type`: Type of attachment (`Hair`, `Beard`, `FaceAttachment`, etc.)
- `tags`: Gender or special tags that determine when attachment is available
- `replacewhenwearinghat`: Whether the hair is hidden when wearing headgear
- `commonness`: Relative frequency for random character generation (higher = more common)

## Health System

The `health` element defines how characters take damage, bleed, and display injuries.

xml

Copy

```
<health Vitality="100" DoesBleed="True" CrushDepth="6000" UseHealthWindow="True" BleedingReduct
  <!-- Limb definitions... -->
</health>
```

## Key Health Attributes

Attribute	Description	Value
Vitality	Base health points	100 is standard human health
DoesBleed	Whether character bleeds from injuries	"True" enables bleeding damage
CrushDepth	Depth at which character is instantly killed	In pressure units (higher = more resistant)
UseHealthWindow	Whether to use the health UI	"True" enables medical interface
BleedingReduction	Natural resistance to bleeding	0 means no resistance
BurnReduction	Natural resistance to burns	0.075 gives slight burn resistance
ConstantHealthRegeneration	Natural health recovery rate	0 means no passive healing

## Limb System

Each body part is defined as a `Limb` element within the `health` section:

xml

Copy

```
<Limb name="Head">
  <Sprite texture="Content/UI/Health/MedUIAtlas.png" sourcerect="1,0,126,374" highlightarea="41
  <Highlightsprite texture="Content/UI/Health/MedUIAtlas.png" sourcerect="1,515,126,374" />
  <VitalityMultiplier type="damage" multiplier="2.0" />
</Limb>
```

Key limb attributes:

- `name`: Identifies the body part
- `VitalityMultiplier`: Modifies damage or bleeding effects for that limb
  - `type`: Can be "damage" or "bleeding"
  - `multiplier`: Value that modifies the effect (e.g., 2.0 means double damage)

## Inventory System

The `inventory` element defines what items characters can carry and equip.

xml

Copy

```
<inventory arrowslot="9" Slots="Card, Headset, Head, InnerClothes, OuterClothes, LeftHand, RightHand">
```

Key inventory attributes:

- `arrowslot`: Default selected inventory slot
- `Slots`: List of slot types that define what items can be equipped where
- `AccessibleWhenAlive`: Whether inventory can be accessed while character is alive

## Sound Effects

The `sound` elements define audio played during various character states.

xml

Copy

```
<sound File="Content/Characters/Human/female_damage1.ogg" State="Damage" Range="500" Volume="1"
```

Key sound attributes:

- `File`: Path to the sound file
- `State`: When the sound plays (e.g., `"Damage"` when taking damage)
- `Range`: How far the sound travels
- `Volume`: Relative volume level
- `Tags`: Conditions for playing (e.g., `"Female"` only for female characters)

## Visual Effects

Several emitter elements control particle effects for various states:

xml

Copy

```
<damageemitter drawontop="True" Particle="gib" AngleMin="0" AngleMax="360" ScaleMin="0.25" Sca
```

Types of emitters:

- `damageemitter`: Visual effects when taking damage
- `bloodemitter`: Blood particles when bleeding
- `gibemitter`: Body parts when character is dismembered

## Advanced Character Configuration

### Gender System

The `Vars` element creates variables that can be used for character customization:

```
<Vars>
  <Var var="GENDER" tags="female,male" />
</Vars>
<MenuCategory var="GENDER" />
<Pronouns var="GENDER" />
```

This defines:

- A gender variable with two possible values
- Menu categorization based on gender
- Appropriate pronouns based on gender selection

## Modding Tips

### Adding Custom Content

To add custom heads, hair, or other visual elements:

1. Create sprite sheets following the same format as the originals
2. Reference them in the XML with the correct file paths
3. Add the appropriate elements to the XML structure

### Custom Names

Character names are defined in separate files:

- `firstnames_male.txt`: Male first names
- `firstnames_female.txt`: Female first names
- `lastnames.txt`: Last names for all characters

Add new names by editing these files, one name per line.

### Creating Content Packages

For comprehensive mods, create a content package:

```
<contentpackage name="MyHumanMod" path="Data\ContentPackages\MyHumanMod">
  <Character file="Mods\MyHumanMod\human.xml" />
  <!-- Other modified files... -->
</contentpackage>
```

This allows players to easily switch between vanilla gameplay and your mod.

## Common Modding Scenarios

### Changing Health Values

To make humans more resistant to damage:

xml

Copy

```
<health Vitality="150" BleedingReduction="0.2" BurnReduction="0.2" ConstantHealthRegeneration="0.2" />
```

### Adding Custom Hair Styles

xml

Copy

```
<Wearable type="Hair" tags="male" replacewhenwearinghat="1">  
  <sprite name="Custom Hair" texture="Mods/MyMod/custom_hair.png" sheetindex="0,0" />  
</Wearable>
```

### Modifying Inventory Capacity

To give characters more inventory slots:

xml

Copy

```
<inventory Slots="Card, Headset, Head, InnerClothes, OuterClothes, LeftHand, RightHand, Bag, Arm" />
```

## Conclusion

The `human.xml` file offers extensive customization options for modders looking to alter the appearance, abilities, and mechanics of human characters in Barotrauma. By understanding the structure and attributes detailed in this guide, you can create everything from small tweaks to comprehensive overhauls of the human character system.