# Local Overrides

## How to locally override client data

Client data is data that is not processed by the server, but by the client. So any changes to the client data are resulting in some behaviour modification. These modifications may be visual, audible (sound) or movement strategy.

## Visual modifications

These include map visuals, homunculus or mercenary outfits, icons and sprites used for items and descriptive texts.

### Map modifications

Any map can be redesigned locally. To make it work the collision tiles should be left untouched, and any modifications should place obstacles where obstacles are found in the original design and where red block collition tiles are. Anything else can be modified. Note, that any NPCs are put on this map by the server. The NPCs are not part of the client data.

Here is a modified Cando map:


This design uses a tileset from TMW Brasil. It resembles a small fishing or beach village, which i think fits better to Candor than the Artis tileset. Artis is a major city, and the Artis tileset fits well to Artis, but not to Candor.

## Homunculus/Mercenary

It is possible to modify the visual (equipment) properties of any homunculus or mercenary as well as its movement strategies. You can add any race sprite and any item sprite that is available. Neither homunculi not mercenaries have a sex/gender attached. So if a homunculus looks male or female is just a metter of the race sprites chosen. It is even possible to replace the race sprite with a monster sprite, although then it would obviously have very strange results if you add item sprites that are designed for human races.

It is also possible to change its movement propertties. Both modifications are shown here:


This shows my avatar to the left, and the homunculus to the right. The avatar outfit is just standard as by the added (equipped) equipment. The homunculus here is an allrounder with new item sprites: A Desert Helmet, a Leasher Shirt, A cotton Skirt, dyed to a color similar to leather, cotton boots dyed to leather colot too, a small shield and an iron sword, and leather gloves (a recolored version of the cotton gloves). The race sprite is the original race sprite, a simple female human.

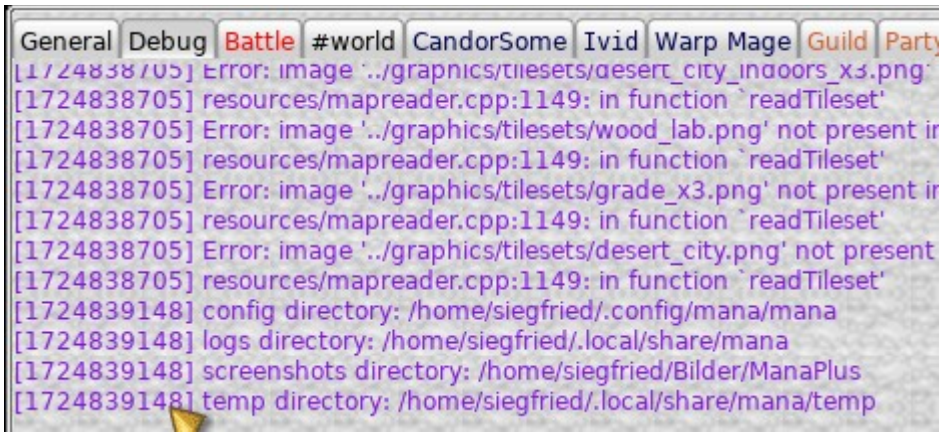
The position is also modified. Instead of in front of the avatar it is at the side of the avatar, same direction. The image shows the „stand still“ position. Attack and walk positions may be different.


This shows another avatar with a „Mage“ homunculus. The outfit here is more mage-like. Also the position here is different. The mage is a ranged homunculus. A ranged homunculus should keep some distance to the avatar. Here it is 5 in Y-direction and 1 in X-direction. The Y-direction is negative, because the homunculus is behind the avatar. A positive Y-direction would place the homunculus in front of the avatar. This placement in front of the avatar may be useful for a ranged avatar with a melee homunculus.

Mercenaries can be configured the same way. Just note that the attack range can not be configured, since this is server data. But the movement can indeed configured here in client data. So keeping the right distance to attack can indeed be configured here.

# How to?


First step is to identify the directory where the client stores its updates. The easiest way is to enter the command „/dirs“ in the debug chat tab. The response shows various directories used by the client. The directory we’re interested in is the logs directory:




So for me this directory is found at ${HOME}/.local/share/mana. There i find the manaplus.log file. Also i find a directory named „updates“ there. In that updates directory there is a subdirectory for every server the client has connected so far. In case of ML this is updates.tmw2.org. That directory then contains a number of zip files with the client data retrieved from the server. For ML there is a directory named „ml“ here. This again contains a number of zip files with client data.

So in that directory (${HOME}/.local/share/mana/updates/updates.tmw2.org/ml) create another directory named „local“. There you can put your local overrides.

To modify the homunculi, get the file named „homunculuses.xml“ either from one of the zip files or from the repository and put it into your newly created local directory. You can now edit that in whatever way you want. To activate your modifications restart the client. The „/update“ command might work too, since it retrieves all the client data and then has to re-read all this data. I have not tested this though.

## Homunculus Outfits

You find an entry for every homunculus type available in this xml file. For race sprite take a look at the „Accurate“ homunculus. Here you find:

<sprite>monsters/tipiou.xml</sprite>

And that’s all. So the Accurate HC is simply a Piou. Note that the homunculus does not have any sex/gender assigned. Technically speaking the homunculus may look like having sex/gender but it does not have. So let’s see how the „Mage“ (the first in the file) is configured. It has ha human female sprite, so it looks like female. Let’s change that to female elven:

<sprite>races/elven-female.xml</sprite>

A naked elf without hair does not look nice. So first let’s add hair. I kept the original hair style, but modified the color:

<sprite>hairstyles/hairstyle19.xml|
#2b2b2b,3d2613,3f3f3f,4e3117,585858,774c20,858585,8f5d22,a7a7a7,ae772d,d6d6d6,dbae5c,ffffff
,f7e1ad</sprite>

The simple hair color used in the original file does not work here. To get a list of simply available colors, take a look at the itemcolors.xml in the original client data, section section list name=“hairS“. These can be used. Just copy the color from there and paste it here like in this example.

Now the equipment. A mage needs a mage robe, a mage hat and a wand:

<sprite>hairstyles/hairstyle19.xml|
#2b2b2b,3d2613,3f3f3f,4e3117,585858,774c20,858585,8f5d22,a7a7a7,ae772d,d6d6d6,dbae5c,ffffff
,f7e1ad</sprite>

<sprite>equipment/head/wizardhat.xml|#4f0a76,8010c0,d699f7</sprite>

<sprite>equipment/feet/boots-female.xml</sprite>

<sprite>equipment/chest/silkrobe-female.xml|#3c1554,6d3195,c987d1</sprite>

<sprite>equipment/weapons/weapon-staff.xml</sprite>

The color for the robe I copied from the itemcolors.xml file, section simple item colors. The color for the hat I copied from Sagratha.

The mage homunculus is a ranged one. So we need a missile. And it has a nice skill, called “Heaven Fury” at client side. We need some effect here too. Well, at least a nice effect here looks nice:

<attack id="1" missile-particle="graphics/particles/misc.firebolt.xml"
action="attack_distance"/>

<attack id="8013" missile-particle="graphics/particles/darkblueflame.particle.xml"
action="attack_magic"/>

There’s one more thing we might add here: Sound. Sound is connected to events. The most common events are hit, miss, hurt, spawn and die:

<sound event="hit">weapons/wand/wand01.ogg</sound>

<sound event="miss">weapons/wand/gluhk.ogg</sound>

<sound event="hurt">Ouch/ouch03.ogg</sound>

<sound event="spawn">special/homspawn.ogg</sound>

<sound event="die">monsters/shit.ogg</sound>

The sound effects I used here are not part of the standard client data. So I added the necessary directories and put those ogg files there. Now every time such an event happens I get an audible feedback. The sound clip should obviously be short, less than a second. Half a second is a good duration.

## Movement

The movement is controlled by various socalled “AI attributes”. Here is what I have set up for the mage homunculus:

```
<homunculus
    id="6001"
    name="Mage"
    startFollowDistance="3"
    followDistance="-5"
    warpDistance="11"
    offsetX="1"
    offsetY="-5"
    sitOffsetX="1"
    sitOffsetY="0"
    moveOffsetX="0"
    moveOffsetY="-5"
    deadOffsetX="0"
    deadOffsetY="0"
    attackOffsetX="0"
    attackOffsetY="-7"
    thinkTime="100"
    directionType="1"
    sitDirectionType="1"
    deadDirectionType="1"
    attackDirectionType="1"
>
```

The ID should obviously not be changed. The name could be changed, but you can change the name of your individual homunculus anyway.

The startFollowDistance sets the distance in tiles, when to follow the avatar. If the avatar is moving and the distance is getting greater than that the homunculus starts to follow.

The followDistance is the standard movement distance it keeps.

If the walk distance gets greater than warpDistance, then the homunculus switches to warping.

OffsetX and offsetY are the standstill relative offsets. So here we have an X offset of 1, that is 1 to the right. And we have an Y offset of -5. That is 5 tiles behind the avatar. These directions depend on the directionType attribute. Turning the direction of your avatar turns this whole coordinate system. I have set the standstill offset to some greater value here, because after finishing an attack we have a standstill situation next. So the avatar moves to that position. If that position is near the avatar, and the avatar has to fight a whole gang of monsters, the homunculus runs directly into that gang. Not very clever. Setting the distance to more than 5 results in more strange and unpredictable behaviour. So keeping a maximum of 5 seems to be useful. Of course you can experiment with that.

The sitOffseX and sitOffsetY is the offset while the avatar is sitting. Setting a close distance here is quite ok. The standard sets a position directly to the right of the avatar. I see no benefit to change that.

The moveOffset is another distance to keep during moving. I don’t know the difference to the followDistance attribute, so I keep them to the same or similar value.

The deadOffsetX and deadOffsetY are pretty useless, so I keep them as they are.

The attackOffsetX and attackOffsetY are important. For melee homunculi the standard values are quite good. For ranged we need other values. This example here uses a distance of 7 (negative, i.e. behind the avatar). The value resembles the server side attack distance.

There are various direction types. From the source I got this:

DOWN = 1,

LEFT = 2,

UP = 4,

RIGHT = 8

So this seems to be a bitmap. I have not found any reliable way of behaviour if I set any other value than these 4, f.ex. 0 or 3. So best is to choose one of these values. The directionType for the avatar is per standard 1, thus looking down. The directionType for the homunculus is per standard 4, thus looking up. That is why the standstill position of the homunculus is 2 steps in front of and looking at the avatar.

## Mercenaries

The same attributes and settings apply to the mercenaries.