LOCKME V1.0.2

Introduction

LockMe provides a set of scripts and configuration files created by Tricky Trouble to enable restraints to be customised with considerable flexibility.

Most restraints come with their own proprietary system for locking and unlocking which may or may not suit your needs. These LockMe scripts can be added to restraints (with appropriate modify permissions) allowing you to personalise many aspects of how they operate, including animations, sounds, RLV restrictions and access options. You can choose to allow access to your restraints by yourself, a group, friends, public or in fact all of these. You can even choose to have a simple one click lock and unlock ability or opt for the more traditional click-for-menu system.

Some users will enjoy the possibility these scripts provide to introduce a little bit more risk into their play by allowing others the ability to access their restraints without explicitly granted permissions.

The basic provided permission system is simple, everyone may lock a device and everyone except the wearer may unlock it. Permission granting is done via linked messages, so the creator may provide other custom access types by adding their own script(s).

Please note that a basic knowledge of adding scripts to objects and editing configuration files is required, although the product does come with clear instructions and a step by step example to help you get started.

Whats in the box?

- **Primary and slave LockMe scripts**
  One of the cuffs must be selected as the master and you add the primary script in here. The slave script is added to all the other items which are part of that set.

- **Example configuration file**
  The configuration file is needed wherever you have a LockMe script. You only need to specify the options in the master item, the slave item only needs to contain the top header. This will be explained in more detail later in these instructions.

- **Readme File**
  Contains basic getting started instructions and points you to this instruction document.

- **LockGuard script v3.0.3**
  LockGuard is a fully editable script provided with this product for convenience, but this is a free separate product created by a different team.

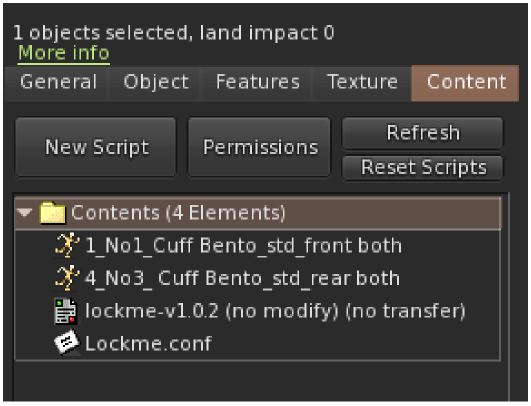
Step-by-Step

This section takes a step by step approach to applying the scripts to a set of modifiable cuffs. For this example I am using a set of the free backbone cuffs available on SL Marketplace.

In addition it utilises some Bento Handcuff animations. Although they cost $250L, they are full perm and high quality.

We will focus on the left and right wrist cuff so this is a logical set sharing the same animations and ‘rules’. Most cuff systems specify a master instance and in line with common practice, we will choose this to be the right cuff.

1. Rez the right cuff and drag in the master script the example Lockme.conf file and the animations called **1_No1_Cuff Bento_std_ front both** and **4_No3_ Cuff Bento_std_ rear both**. The content tab should look as follows:

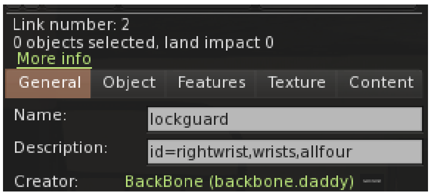


1 objects selected, land impact 0
More info

Contents (4 Elements)
1_No1_Cuff Bento_std_front both
4_No3_ Cuff Bento_std_rear both
lockme-v1.0.2 (no modify) (no transfer)
Lockme.conf

2. Select **Edit Linked** and click through the linked items until you see the script and configuration file related to LockGuard. Delete this legacy version and drag in the newer script provided as part of LockMe.

3. This linked prim needs to renamed to **lockguard** and the lockguard settings added to the description field, in this case: **id=rightwrist,wrists,allfour**.



Link number: 2
0 objects selected, land impact 0

Name: lockguard
Description: id=rightwrist,wrists,allfour
Creator: BackBone (backbone.daddy)

4. Take back the right cuff into your inventory. A new copy is created so delete the older version to avoid confusion.

5. Rez the left cuff and drag in the slave script and the example **Lockme.conf** file.

6. As with the right side, also delete and replace the **LockMaster** with the newer provided version and update the name and Description fields. The description for the left cuff will be: **id=leftwrist, wrists,allfour**

7. As with the right cuff, take back the left cuff into your inventory and delete the older version to avoid confusion.

8. Now you can wear the left and right cuff and our next step is to edit the conf file.

9. First, edit the left cuff and open the conf file and edit it to look as follows:

```
- Lockme
tag=wrists
listen-tags=wrists

- Option.LockFront
showall=1.0
display=lockguard|0

- Option.LockBack
showall=1.0
display=lockguard|0

- Option.Unlock
showall=0.0
```

9. Next edit the right cuff and open the conf file and edit it as follows:

```
- Lockme
options=LockFront|LockBack|Unlock
quick-lock=LockBack
quick-release=LockFront,LockBack
tag=wrists
listen-tags=wrists
release-option=Unlock

- Option.LockFront
animation=1_No1_Cuff Bento_std_front both
rlv=detach=n,showinv=n,viewscript=n,edit=n,rez=n,touchfar=n,
unsharedunwear=n,sharedunwear=n
sound=8c7efd3b-a121-b6f8-c1ce-1667cc6ee990
```

```
join=leftwrist
access-type=all
showall=1.0
display=lockguard|0

- Option.LockBack
animation=4_No3_ Cuff Bento_std_rear both
rlv=detach=n,showinv=n,viewscript=n,edit=n,rez=n,touchfar=n,
unsharedunwear=n,sharedunwear=n
sound=8c7efd3b-a121-b6f8-c1ce-1667cc6ee990
join=leftwrist
access-type=all
showall=1.0
display=lockguard|0

- Option.Unlock
sound=b9a74fa4-7386-2daa-89c1-2ef62b64b076
showall=0.0
access-type=all
```

Understanding the .conf file

In this section we will look at each line of the configuration file and understand what it does and the format to use.

```
options=LockFront|LockBack|Unlock
```

This setting specifies each option that is available for the wrist cuffs separated by a pipe. In this example, we just have lockfront, lockback and unlock. So specifically this provides options to lock the arms in front , lock the arms behind and unlock. These of course can be expanded as as you wish. These options will appear in the menu which will appear if you are not using the quick lock option.

```
quick-lock=LockBack
```

This setting specifies the specific option which will be triggered if you are in the release state and click the cuffs. So in this case if the right cuff is clicked then it will trigger the Lockback option. Obviously change this setting to LockFront if you want the arms to be locked in front on quick click. If you don’t want to use quick lock then place // in front of this line so the setting is treated as a comment and ignored.

```
quick-release=LockFront,LockBack
```

The quick-release setting specifies a list of states separated by a comma that will quick release you on click. So if you are in the LockBack state, a click of the cuffs will release you.

```
release-option=Unlock
```

This setting specifies that in the case of release, **Option.Unlock** should be triggered.

```
tag=wrists
```

This is the official name that is presented to the script in the master cuff. This setting can exist in the slave cuffs but is ignored as it is not the master.

```
listen-tags=wrists
```

This is the official name that is presented to the script in the slave cuffs. This setting can exist in the master cuff but is ignored as it is not a slave.

So now we focus on the settings inside **Option.LockFront**.

```
animation=1_No1_Cuff Bento_std_front both
```

This is the animation that will be triggered if Option.LockFront is called. The animation in question needs to exist in the contents area of the master cuff and be referenced exactly.

```
rlv=detach=n,showinv=n,viewscript=n,edit=n,rez=n,touchfar=n,
unsharedunwear=n,sharedunwear=n
```

These are the RLV commands which will be applied when Options.Lock is called. For more information you should refer to the RLV API which is the ultimate reference source for the correct use of RLV commands The commands listed here apply the following restrictions:

- **detach=n**
  The wearers cuffs can not be detached.

- **showinv=n**
  The wearers inventory is disabled.

- **viewscript=n**
  None of the scripts in the cuffs can be viewed by the wearer.

- **edit=n**
  The cuffs can not be edited by the wearer.

- **rez=n**
  The wearer will not be able to rez items while locked.

- **touchfar=n**
  The wearer will be unable to touch items which are far away.

- **unsharedunwear=n**
  The wearer can not unwear items outside the #RLV folder (i.e. unshared).

- **sharedunwear=n**
  The wearer can not unwear items inside the #RLV folder (i.e. shared).

```
sound=8c7efd3b-a121-b6f8-c1ce-1667cc6ee990
```

This is the sound file that will be triggered if **Option.LockFront** or **Option.LockBack** is called. (A published sound file is referenced by it's UUID). In this case it will be the sound of the cuffs being locked.

```
join=leftwrist
```

This triggers the primary cuff (right cuff) to be joined with the left cuff using the LockGuard functionality in the separately provided script. To edit the LockGuard settings you need to edit the script itself which is fully Mod.

```
showall=1.0
```

This setting specifies that the cuffs should have zero transparency, i.e. be fully displayed. The value needs to be between 0.0 and 1.0 where 0.0 is fully transparent. Please note that this option may display linked prims that should usually remain hidden, so note the display setting below to see how to hide such items.

```
access-type=all
```

This setting specifies the access rule so in other words, specifies the rules around who is able to unlock the cuffs. Options are group, self, public and all. To use the group option you and the other person must be displaying the same active group.

```
display=lockguard|0
```

This setting is used to hide or show specific linked prims. The first parameter is the name of the linked prim and after the pipe is the value. 0 means hide and 1 means show.