

# Pi

- Magic Mirror
  - Installation
  - Magic Mirror Modules:
- Backup Scripts
  - Staging Configs To A USB

# Magic Mirror

<https://github.com/MichMich/MagicMirror>

Magic Mirror

# Installation

Execute the following command on your Raspberry Pi to install MagicMirror<sup>2</sup>:

```
bash -c "$(curl -sL https://raw.githubusercontent.com/MichMich/MagicMirror/master/installers/raspberry.sh)"
```

## Start MM:

```
cd ~/MagicMirror/
```

```
npm start
```

Grab the unclutter package to hide mouse on idle

```
sudo apt install unclutter
```

Magic Mirror

# Magic Mirror Modules:

Install here:

~/MagicMirror/modules/

Add to config:

~/MagicMirror/config/config.js

Default Template:

~/MagicMirror/config/config.js.sample

Current list in use:

MMM-DarkSkyForecast

MMM-EARTH

MMM-Globe

MMM-OnScreenMenu

MMM-Remote-Control

# Backup Scripts

For posting drafts, notes on backing up the pi and mirror with bash scripts

# Staging Configs To A USB

**These scripts are broken, and are no longer being worked on.**

If you just copy them and execute them without reading them, they will not work, and may have unexpected results.

dd can be used instead to backup the image of the SD card of the Pi and its filesystem.

The idea behind these scripts is to stage configuration files in an alternate directory before moving them to a primary backup directory. This enables you to test your settings on a fresh install via the stash scripts before overwriting the last working configuration stored with the backup scripts.

Setup pi-> ./stash.sh -> format pi -> ./unstash.sh -> verify settings -> ./repack

**The above workflow allows for the below to occur if something breaks...**

Setup pi-> ./stash.sh -> format pi -> ./unstash.sh -> break settings -> format pi -> ./unpack

In this way, the pi can be tested without having to reconfigure every package and setting upon every reset.

Setup the Pi to your new configuration, once your settings are working and you are ready to reformat the pi and test your backup script, run `stash.sh`

```
#!/bin/bash
##Stash.sh script for storing local MM settings in a temp directory
##Use unstash.sh to test stashed settings - Used like a hacky VCS
##Stage settings in stash, format pi, run unstash.sh
##If the settings work as expected, backup using repack.sh
#####
HISTFILE=/home/pi/.bash_history
set -o history
history > /media/pi/<STORAGEDEVICE>/Notes/stash/bash.history
apt-mark showmanual | sort -u > /media/pi/<STORAGEDEVICE>/Notes/stash/package.list
sudo cp -vur ~/MagicMirror/config/ /media/pi/<STORAGEDEVICE>/Notes/stash/MagicMirror/config/
sudo cp -vur /home/pi/MagicMirror/modules /media/pi/A03F-DF04/Notes/stash/MagicMirror
```

```
sudo cp -vu /etc/ssh/sshd_config /media/pi/<STORAGEDEVICE>/Notes/stash/  
sudo cp -vu /etc/ssh/ssh_config /media/pi/<STORAGEDEVICE>/Notes/stash/  
sudo cp -vur ~/.ssh/ /media/pi/A03F-DF04/Notes/stash/  
sudo cp -vu /home/pi/.bashrc /media/pi/<STORAGEDEVICE>/Notes/stash/  
sudo cp -vu /etc/wpa_supplicant/wpa_supplicant.conf /media/pi/<STORAGEDEVICE>/Notes/stash/  
sudo cp -vu /home/pi/*.sh /media/pi/<STORAGEDEVICE>/Notes/stash/
```

Replace `<STORAGEDEVICE>` with the name of your USB device plugged into the Pi, and feel free to alter the deeper configs as you see fit.

Reformat the Pi, run `sudo raspi-config` to enable Wi-Fi, make sure you are connected and then run `unstash.sh`

```
#!/bin/bash  
##Unstash.sh script using stashed settings for custom MM  
##Unpacks stashed settings copied by stash.sh  
##Used to prevent loss of working configs if unstash.sh fails  
##To return to last working state,format the pi and run unpack.sh  
##Use repack.sh and unpack.sh to backup working builds  
#####  
  
printf "\nDo you want to setup stashed PI settings?\n"  
read Pi  
  
if [ $Pi == 'Y' ]||[ $Pi == 'y' ]  
then  
    sudo cp -vu $DIR/wpa_supplicant.conf /etc/wpa_supplicant/  
    sudo cp -vu $DIR/ssh_config /etc/ssh/  
    sudo cp -vu $DIR/sshd_config /etc/ssh/  
    sudo cp -vur $DIR/.ssh/ $HOME  
    sudo systemctl restart ssh.service sshd.service  
    #sshd fails to restart on Debian / PI?  
fi  
  
printf "\nDo you want to install MagicMirror?\n"  
read Script  
  
if [ $Script == 'y' ] || [ $Script == 'Y' ]  
then  
    printf "\nInstalling Magic Mirror using custom unstash script....\n\n"
```

```
printf "\ncurl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -\n"
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
```

```
printf "\nsudo apt install -y nodejs\n"
sudo apt install -y nodejs
```

```
printf "\nsudo npm install pm2 -g\n"
sudo npm install pm2 -g
```

```
printf "\nsudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u $USER --
hp /home/pi\n"
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u $USER --hp $HOME
```

```
printf "\nsudo git clone https://github.com/michmich/magicmirror\n"
sudo git clone https://github.com/MichMich/MagicMirror
```

```
printf "\nnpm install\n"
cd $HOME/MagicMirror && sudo npm install
cd
```

```
printf "unstash.sh magic mirror install complete.\nMagic Mirror installed! Navigate to ~/magicmirror/ and
run the following to start manually:\nnpm start\n\n"
fi
```

```
printf "\nCopy stashed Magic Mirror configs?\n"
read Configs
```

```
if [ "$Configs" == "y" ] || [ "$Configs" == "Y" ]
then
    # I can't remember why I didnt specify a root for stash/ here
    # Stick $DIR here if needed, I think it is
    sudo cp -ruv /stash/MagicMirror/* $HOME/magicmirror/
    sudo cp -vu /mm.sh $HOME
    sudo chmod a+x mm.sh
fi
```

```
#Regardless of path, show useful help text, script finished
pm2 startup
```

```
printf "\n#####Important Instructions#####\nBe sure to also run the following to save
```



```
pm2 status:\npm2 save\nTo see pm2 command examples, run -h or pm2 examples:\npm2 -h\npm2
examples\n\nRUN COMMAND BELOW TO START MAGIC MIRROR:\n\nnpm2 start mm.sh\n"
```

This script *will take a while to run*. Once it is finished, all the packages, configurations, and settings will be restored. If the result is expected, continue to run `repack.sh`

```
#!/bin/bash
##Repack.sh script for backup of verified local settings
##This will overwrite your previous settings that will be used with unpack.sh
##Use stash.sh to store and test builds before placing config files here
#####
sudo history > /media/pi/A03F-DF04/Notes/backups/bash.history
comm -23 <(apt-mark showmanual | sort -u) <(gzip -dc /var/log/installer/initial-status.gz | sed -n 's/^Package:
//p' | sort -u) > /media/pi/A03F-DF04/Notes/backups/package.list
sudo cp ~/MagicMirror/config/config.js /media/pi/A03F-DF04/Notes/backups/MagicMirror/config/
sudo cp -r ~/MagicMirror/modules/ /media/pi/A03F-DF04/Notes/backups/MagicMirror/
sudo cp /etc/ssh/sshd_config /media/pi/A03F-DF04/Notes/backups/
sudo cp /etc/ssh/ssh_config /media/pi/A03F-DF04/Notes/backups/
sudo cp -r ~/.ssh/ /media/pi/A03F-DF04/Notes/backups/
sudo cp ~/.bashrc /media/pi/A03F-DF04/Notes/backups/
sudo cp /etc/wpa_supplicant/wpa_supplicant.config /media/pi/A03F-DF04/Notes/backups/
```

This will backup your working settings to an alternate directory, allowing you to continue to alter the settings in the stashed directory without worrying about losing the last working state of the Pi.

`unpack.sh`

```
#!/bin/bash
##Unpack.sh script for custom MM
##Unpacks custom settings copied by repack.sh
##Use stash.sh and unstash.sh to backup local for testing builds
#####

#cmd setup raspi-config?

#Copy custom .bashrc configuration
#sudo cp /media/pi/A03F-DF04/Notes/.bashrc ~/
```

#Copy current wifi and SSH / SSHD settings

```
sudo cp /media/pi/A03F-DF04/Notes/backups/wpa_supplicant.conf /etc/wpa_supplicant/
```

```
sudo cp /media/pi/A03F-DF04/Notes/backups/ssh_config /etc/ssh/
```

```
sudo cp /media/pi/A03F-DF04/Notes/backups/sshd_config /etc/ssh/
```

```
sudo cp -r /media/pi/A03F-DF04/Notes/backups/.ssh/ ~/
```

```
sudo systemctl restart ssh sshd
```

#?Why error after sshd

#Update and get used packages - Upgrade?

```
sudo apt update && cat /media/pi/A03F-DF04/Notes/backups/package.list | sudo apt install
```

#Is sudo inherited by &&?

```
#sudo apt update && comm -23 <(apt-mark showmanual | sort -u) <(gzip -dc /var/log/installer/initial-status.gz |
```

```
sed -n 's/^Package: //p' | sort -u) | sudo apt install
```

#Set up MagicMirror and modules

```
git clone https://github.com/MichMich/MagicMirror
```

```
pm2 stop MagicMirror
```

```
pm2 save
```

```
sudo ./MagicMirror/installers/raspberry.sh
```

```
sudo cp -r /media/pi/A03F-DF04/Notes/backups/MagicMirror/ ~/
```

```
sudo cp /media/pi/A03F-DF04/Notes/backups/mm.sh ~/
```

```
sudo chmod 777 mm.sh
```

```
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u pi --hp /home/pi
```

```
pm2 start mm.sh
```

```
sudo x-terminal-emulator -e sudo pm2 logs mm
```

```
sudo x-terminal-emulator -e sudo htop
```

```
sudo x-terminal-emulator -e sudo tail -f /var/log/ssh/auth.log
```

```
sudo x-terminal-emulator -e sudo pm2 show mm ; pm2 status
```

```
pm2 save
```

#Break script, print help, enter to continue setup of custom MM

#Allows time to run pm2 startup output command manually, resume unpack.sh on enter

```
printf "#####Important Instructions#####\n"
```

```
pm2 startup
```

```
printf "Be sure to also run the following to save pm2 status:\npm2 save\nTo see pm2 command examples, run -h"
```

or pm2 examples:\npm2 -h\npm2 examples\n"

#Prompt for rsync init on this pi, hook to script