

# NARMD (Not Another Removable Media Disconnecter)

We are using this script to provide a slightly more user-friendly CDROM and floppy disconnect environment.

You can easily disconnect all CDROMS with

```
Get-VM | Get-CDROMDrive |  
Set-CDROMDrive -Connected $False -Confirm:$False
```

However you have little control over the above process, and might disconnect things that are actually in use. This script presents a list of VM's whose CDROM(s) or Floppie(s) are connected now, or are set to connect at startup, then prompts the operator to continue.

If the operator confirms the action, it will go ahead and disconnect all the CDROMs and Floppies and uncheck the Connect At Startup, but at least the operator knows what's happening.

If you want a log file you can invoke this as follows:

```
.\removableDrives.ps1 |  
Tee-Object C:\Logs\removableDrives.log
```

Here's a sample console session (up to the prompt)

```
VM: Server1 (PoweredOff) CD/DVD Drive 1 Connected=True ConnectAtPowerUp=True Floppy Drive 1 Connected=False  
ConnectAtPowerUp=False VM: Server2 (PoweredOff) CD/DVD Drive 1 Connected=True ConnectAtPowerUp=True Floppy  
Drive 1 Connected=True ConnectAtPowerUp=True VM: Workstation3 (PoweredOff) CD/DVD Drive 1 Connected=True  
ConnectAtPowerUp=True Floppy Drive 1 Connected=False ConnectAtPowerUp=False VM: DomainController1 (PoweredOn)  
CD/DVD Drive 1 Connected=True ConnectAtPowerUp=True Floppy Drive 1 Connected=False ConnectAtPowerUp=True Do  
you want to disconnect all removable drives, and disable the connect at startup? If you don't see any VMs everything is OK and  
you should select N here. Type Y or N:
```

The script is here for full-text searching, and is also attached.

```
"removableDrives.ps1 Version 1.0a starting" connect-viserver ($env:computername + "." + $env:userdnsdomain) |  
out-null Get-Vm | Where-Object { (((Get-CDDrive -VM $_ | Where-Object { (($_.ConnectionState.Connected -eq $True)  
-or ($_.ConnectionState.StartConnected -eq $True))) } -ne $Null) ` -or ` ((Get-FloppyDrive -VM $_ | Where-Object {  
(($_.ConnectionState.Connected -eq $True) -or ($_.ConnectionState.StartConnected -eq $True))) } -ne $Null)) } | % {  
"VM: " + $_.Name + " (" + $_.PowerState + ")" Get-CDDrive -VM $_ | ForEach-Object { " " + $_.Name + " Connected=" +  
$_ .ConnectionState.Connected + " ConnectAtPowerUp=" + $_.ConnectionState.StartConnected } Get-FloppyDrive  
-VM $_ | ForEach-Object { " " + $_.Name + " Connected=" + $_.ConnectionState.Connected + " ConnectAtPowerUp=" +  
$_ .ConnectionState.StartConnected } } if ("Y" -eq (Read-Host ` ("Do you want to disconnect all removable drives, and  
disable the connect `n" + ` " at startup? If you don't see any VMs everything is OK and you should `n" + ` " select N here.  
Type Y or N"))) { Get-Vm | Where-Object { (((Get-CDDrive -VM $_ | Where-Object { (($_.ConnectionState.Connected -eq  
$True) -or ($_.ConnectionState.StartConnected -eq $True))) } -ne $Null) ` -or ` ((Get-FloppyDrive -VM $_ | Where-Object {  
(($_.ConnectionState.Connected -eq $True) -or ($_.ConnectionState.StartConnected -eq $True))) } -ne $Null)) } | % { "VM:  
" + $_.Name + " (" + $_.PowerState + ")" Get-CDDrive -VM $_ | ForEach-Object { Set-CDDRIVE -CD $_ -StartConnected  
$False -Connected $False -Confirm:$False | % { " Changed " + $_.Name } } Get-FloppyDrive -VM $_ | ForEach-Object {  
Set-FloppyDRIVE -Floppy $_ -StartConnected $False -Connected $False -Confirm:$False | % { " Changed " + $_.Name } } }  
"removableDrives.ps1 finished."
```