

Tools for ASM and disks administration and monitoring

ASM tools

- **lsop** - lists ASM operations in progress, see example below

```
[ ~]$ lsop
```

INST_ID	GROUP_NUMBER	OPERA	STAT	POWER	ACTUAL	SOFAR
2	2	REBAL	RUN	4	4	668

- **lsdg** - lists ASM diskgroups, see example below

```
[ ~]$ lsdg
```

State	Type	Rebal	Unbal	Sector	Block	AU	Total_MB	Free_MB	Req_mir_free_MB
MOUNTED	NORMAL	N	N	512	4096	1048576	3662304	1350775	915576
MOUNTED	NORMAL	Y	N	512	4096	1048576	3956400	3352434	359672

- **listdisks.py** - lists ASM disks details, storage path, diskgroup etc

```
Usage: listdisks.py [-o column_number] [-h]
       -o             sorts by column_number (format: 3 or 3,2,...)
       -h             this help
```

- **asmiostat.py** - monitor asm IO rates (syntax just like in iostat command)

```
Usage: asmiostat.py [ <interval> [ <count> ] ]
       <interval>    interval in seconds (default 5)
       <count>       number of repetitions
```

Disk tools

- **lssdisks.py** - lists connected storages and visible disks and their multipath status, see example below

```
[ ~]$ lssdisks.py
```

The following storages are connected:

* Host interface 1:

```
Target ID 1:0:0: - WWPN: 210000D023BC3F63 - Storage: itstor311, Port: 0
Target ID 1:0:1: - WWPN: 210000D023CBE093 - Storage: itstor312, Port: 0
Target ID 1:0:2: - WWPN: 210000D023DBE041 - Storage: itstor313, Port: 0
Target ID 1:0:3: - WWPN: 210000D023EC3F90 - Storage: itstor314, Port: 0
```

* Host interface 2:

```
Target ID 2:0:0: - WWPN: 220000D023BC3F63 - Storage: itstor311, Port: 1
Target ID 2:0:1: - WWPN: 220000D023CBE093 - Storage: itstor312, Port: 1
Target ID 2:0:2: - WWPN: 220000D023DBE041 - Storage: itstor313, Port: 1
Target ID 2:0:3: - WWPN: 220000D023EC3F90 - Storage: itstor314, Port: 1
```

SCSI Id	Block DEV	MPath name	MP status	Storage	Port
[0:0:0:0]	/dev/sda	-	-	-	-
[1:0:0:0]	/dev/sdb	itstor311_CRS	OK	itstor311	0
[1:0:0:1]	/dev/sdc	itstor311_1	OK	itstor311	0
[1:0:0:2]	/dev/sdd	itstor311_2	OK	itstor311	0
[1:0:0:3]	/dev/sde	itstor311_3	OK	itstor311	0
...					

Device mapper (*multipath.conf*) auto-generation

- If all storages are connected to a node and they're listed in *~/scripts/wwns.dat* file one can use **gen_multipath.py** script to auto-generate *multipath.conf* file, see example below
 - ◆ generate new file:

```
gen_multipath.py >new_multipath
```

◇ for some DBs one need to reuse existing configuration of *multipath.conf* (excluded devices, etc.), in order to do it please call **gen_multipath.py** with **y** option:

```
gen_multipath.py y >new_multipath
```

- ◆ compare with existing one (check generated names and potential configuration differences):

```
diff -Bbc new_multipath /etc/multipath.conf
```

- ◆ overwrite current one (using *cp*, instead of *mv* preserves destination file's permissions):

```
sudo cp new_multipath /etc/multipath.conf  
rm new_multipath
```

- ◆ make multipath reload it's configuration:

```
sudo multipath
```

This topic: PSSGroup > AsmTools

Topic revision: r3 - 2009-01-22 - DawidWojcik



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback