

NAME

ldmtool – A tool to manage Microsoft Windows dynamic disks

SYNOPSIS

ldmtool [options]

ldmtool [options] scan [*device...*]

ldmtool [options] show diskgroup {*disk group GUID*}

ldmtool [options] show {volume | partition | disk} {*disk group GUID*} {*object name*}

ldmtool [options] {create | remove} all

ldmtool [options] {create | remove} volume {*disk group GUID*} {*volume name*}

OPTIONS

-d|--device *device*

Automatically scan *device*.

DESCRIPTION

ldmtool is a tool for managing Microsoft Windows dynamic disks, which use Microsoft's LDM metadata. It can inspect them, and also create and remove device-mapper block devices which can be mounted.

Although a filesystem can be mounted read-write and its contents modified, **ldmtool** is not able to modify the LDM metadata itself. That is, it cannot create, remove or edit dynamic disks.

It is also not able to mount RAID5 volumes which have a partition missing, although it can mount mirrored volumes with a partition missing. However, mounting a volume with a missing partition is not recommended, as **ldmtool** does not update the LDM metadata in any way. This means Windows will have no way to determine that the partitions are not synchronised when it subsequently mounted, which may result in corruption.

INVOCATION

ldmtool can be invoked either as a shell to run multiple actions, or to run a single action and return. If an action is given on the command line it will run in single action mode. If no action is given it will launch a shell.

Shell mode

When invoked as a shell, **ldmtool** will not scan any block devices by default. If any block devices are given on the command line with the **-d** option, these will be scanned. Otherwise, block devices must be scanned explicitly with the **scan** action.

Exit shell mode by sending an EOF, or with the quit or exit actions.

Single action mode

When invoked to run a single action all block devices will be scanned by default. In this case, if any block devices are specified with the **-d** option, only those block devices will be scanned.

RESULTS

ldmtool returns results as JSON-formatted data. The precise data returned is described in detail below. If a command fails it will not return any JSON data, but will instead display an error message. In shell mode, failure can be detected by the lack of a JSON-formatted result. In single action mode the caller can additionally check the exit code of the command.

ACTIONS**scan** [*device...*]

Scan all [*device*]s for LDM metadata.

Returns a list of all known disk group GUIDs. Note that it doesn't just return newly discovered disk groups.

show diskgroup {*GUID*}

Return detailed information about a single disk group.

Returns:

name
The human-readable name of the disk group

guid
The Windows-assigned GUID of the disk group

volumes
A list of the names of all volumes in the disk group

disks
A list of the name of all disks in the disk group

show volume *{disk group GUID} {volume name}*
Return detailed information about a volume.

Returns:

name
The name of the volume

guid
The Windows-assigned GUID of the volume

type
The volume type. One of: simple , spanned , striped , mirrored , raid5

size
The size of the volume in sectors

chunk-size
The chunk size, in sectors, used by striped and raid5 volumes. For other volume types it will be 0.

hint
The volume mounting hint. This value specifies how Windows expects the volume to be mounted. For a volume with an assigned drive letter, it might be 'E:'.

device
The host device-mapper device which was created for this volume if any (e.g. /dev/mapper/ldm_vol_Machine-Dg0_Volume1)

partitions
A list of the names of the partitions which constitute this volume.

show partition *{disk group GUID} {partition name}*
Return detailed information about a partition.

Returns:

name
The name of the partition

start
The offset, in sectors, of the start of the partition from the beginning of the disk

size
The size of the partition in sectors

disk
The name of the disk the partition is on

show disk *{disk group GUID} {disk name}*
Return detailed information about a disk.

Returns:

name
The human-readable name of the disk

guid

The GUID of the disk

present

Whether the disk has been discovered during scanning: true or false

If the disk is present, additionally returns:

device

The name of the host device (e.g. /dev/sda)

data-start

The start sector of the data portion of the disk

data-size

The size, in sectors, of the data portion of the disk

metadata-start

The start sector of the metadata portion of the disk

metadata-size

The size, in sectors, of the metadata portion of the disk

create {volume { disk group *GUID* } { volume name } | all}

Create a device-mapper device for either the specified volume or all volumes in all detected disk groups.

Returns a list of the device-mapper device names which were created by this action. Note that if a device already existed for a volume it will not be returned in this list.

remove {volume { disk group *GUID* } { volume name } | all}

Remove device-mapper devices for either the specified volume or all volumes in all detected disk groups.

Returns a list of the device-mapper device names which were removed by this action. Note that if no device existed for a volume it will not be returned in this list.

EXAMPLES

The following examples form a sequence from a single session of `ldmtool` running in shell mode.

Scan all loop devices for LDM metadata:

```
ldm> scan /dev/loop[0-9]*
[
  "03c0c4fc-8b6f-402b-9431-4be2e5823b1c",
  "06495a84-fbfd-11e1-8cf9-52540061f5db"
]
```

Two disk groups were detected. The list contains their GUIDs.

Show detailed information about one of the disk groups:

```
ldm> show diskgroup 06495a84-fbfd-11e1-8cf9-52540061f5db
{
  "name" : "WIN-ERRDJSBDAVF-Dg0",
  "guid" : "06495a84-fbfd-11e1-8cf9-52540061f5db",
  "volumes" : [
    "Volume4",
    "Volume1",
    "Volume2",
    "Volume3",
```

```
    "Volume5"
  ],
  "disks" : [
    "Disk1",
    "Disk2",
    "Disk3",
    "Disk4",
    "Disk5",
    "Disk6",
    "Disk7",
    "Disk8",
    "Disk9"
  ]
}
```

Show detailed information about one of the volumes in the disk group:

```
ldm> show volume 06495a84-fbfd-11e1-8cf9-52540061f5db Volume1
{
  "name" : "Volume1",
  "guid" : "c08309e9-1d30-43e5-8dc3-fcec8be76fcc",
  "type" : "spanned",
  "size" : 129024,
  "chunk-size" : 0,
  "hint" : "E:",
  "device" : "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume1",
  "partitions" : [
    "Disk1-01",
    "Disk2-01"
  ]
}
```

Show detailed information about one of the partitions in the volume:

```
ldm> show partition 06495a84-fbfd-11e1-8cf9-52540061f5db Disk1-01
{
  "name" : "Disk1-01",
  "start" : 65,
  "size" : 96256,
  "disk" : "Disk1"
}
```

Show detailed information about the disk containing the partition:

```
ldm> show disk 06495a84-fbfd-11e1-8cf9-52540061f5db Disk1
{
  "name" : "Disk1",
```

```

"guid" : "06495a85-fbfd-11e1-8cf9-52540061f5db",
"present" : true,
"device" : "/dev/loop15",
"data-start" : 63,
"data-size" : 100289,
"metadata-start" : 100352,
"metadata-size" : 2048
}

```

Create a device-mapper device for a volume:

```

ldm> create volume 06495a84-fbfd-11e1-8cf9-52540061f5db Volume1
[
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume1"
]

```

A new device-mapper device has been created, called `ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume1`. Depending on how `udev` is configured, it will probably now be available as `/dev/mapper/ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume1`.

Note that returned name of the created device-mapper device is unmangled and thus can contains not-whitelisted characters. User-space device-mapper tools (like `dmsetup`) and `libdevmapper` library accept unmangled names (if properly configured) but take extra care composing path to a created device yourself. Consider using "device" field of the **show volume** command output.

Create device-mapper devices for all volumes in both disk groups:

```

ldm> create all
[
  "ldm_vol_Red-nzv8x6obywgDg0_Volume2",
  "ldm_vol_Red-nzv8x6obywgDg0_Volume1",
  "ldm_vol_Red-nzv8x6obywgDg0_Volume4",
  "ldm_vol_Red-nzv8x6obywgDg0_Stripel",
  "ldm_vol_Red-nzv8x6obywgDg0_Raid1",
  "ldm_vol_Red-nzv8x6obywgDg0_Volume3",
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume4",
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume2",
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume3",
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume5"
]

```

Note that `ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume1` is not in the list of devices which were created as it already existed.

Remove the device-mapper device for a single volume:

```

ldm> remove volume 06495a84-fbfd-11e1-8cf9-52540061f5db Volume2
[
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume2"
]

```

```
]
```

```
ldm> remove all
```

```
[  
  "ldm_vol_Red-nzv8x6obywgDg0_Volume2",  
  "ldm_vol_Red-nzv8x6obywgDg0_Volume1",  
  "ldm_vol_Red-nzv8x6obywgDg0_Volume4",  
  "ldm_vol_Red-nzv8x6obywgDg0_Stripe1",  
  "ldm_vol_Red-nzv8x6obywgDg0_Raid1",  
  "ldm_vol_Red-nzv8x6obywgDg0_Volume3",  
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume4",  
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume1",  
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume3",  
  "ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume5"  
]
```

Note that ldm_vol_WIN-ERRDJSBDAVF-Dg0_Volume2 is not in the list of devices which were removed, as it was removed previously.

AUTHOR

Matthew Booth <mbooth@redhat.com>