RisingTide iSCSI Target and Snapshots

User's Reference Manual

RisingTide Systems
2000 Executive Parkway
San Ramon, CA 94583

www.risingtidesystems.com

Edition 1.1
Printed Date: 19 August 2009

This Manual describes configuring and operating the RisingTide LIO iSCSI Target and Snapshots. In particular:

- iSCSI Targets: starting, stopping and backing-up their configuration
- iSCSI Storage Objects: creating and releasing different backstore types
- iSCSI Endpoints: creating, configuring, enabling and disabling
- iSCSI Network Portals: creating, enabling and disabling
- iSCSI Authentication and its configuration
- Volume Snapshots and their configuration
- Microsoft Windows XP iSCSI Initiitor and its configuration
- Microsoft Vista iSCSI Inititor and its configuration

Contents

1. RisingTide LIO iSCSI Target

iSCSI Targets

Starting, Stopping and Status

A Target node can be started and stopped anytime by running the script that is run at boot time.

Start the Target node

```
# /etc/init.d/target start
Loading target_core_mod/ConfigFS core:   [OK]
Calling ConfigFS script /etc/target/tcm_start.sh for target_core_mod:   [OK]
Calling ConfigFS script /etc/target/lio_start.sh for iscsi_target_mod:   [OK]
```

Stop the Target node

```
# /etc/init.d/target stop
Unloading LIO-Target/ConfigFS fabric:   [OK]
Unloading target_core_mod/ConfigFS core:   [OK]
```

The unload messages will reflect the iSCSI Target Endpoints and storage objects that have been configured.

Query the Target node

Example: Show the status of a running empty iSCSI Target (there are no Storage Objects defined yet):

```
# /etc/init.d/target status

[---------------------------] TCM/ConfigFS Status [-----------------------------]

[---------------------------] LIO-Target Status [-----------------------------]
```

Dumping and Backing Up Configurations

The configuration of the Target and the iSCSI objects can be dumped and/or saved. These commands display the configfs syntax that gets run at init.d/target start time to reinstate the running system.

Dump the Target configuration to stdout

Example: Dump the configuration of an initialized but empty generic Target Engine:

```
# tcm_dump --s
modprobe target_core_mod
#### ALUA Logical Unit Groups
#### ALUA Target Port Groups
```

Example: Dump the configuration of an initialized but empty Target (i.e., no Storage Objects defined):

```
# lio_dump --s
Unable to access lio_root: /sys/kernel/config/target/iscsi
```

Keep running *tcm_dump --s* and *lio_dump --s* to see the configfs syntax as you call different CLI ops.

Save the current Target configuration

Example: Establish a completed Target configuration as the new default configuration:

```
# tcm_dump --o
Are you sure you want to overwrite the default configuration? Type 'yes': yes
Making backup of LIO-Target/ConfigFS with timestamp: 2009-08-08_18:00:14.332605
Generated LIO-Target config: /etc/target/backup/lio_backup-2009-08-
08_18:00:14.332605.sh
Making backup of Target_Core_Mod/ConfigFS with timestamp: 2009-08-08_18:00:14.332605
Generated Target_Core_Mod config: /etc/target/backup/tcm_backup-2009-08-
08_18:00:14.332605.sh
Successfully updated default config /etc/target/lio_start.sh
Successfully updated default config /etc/target/tcm_start.sh
```

Back up the current Target configuration

tcm_dump --t <FileName>

Example: Dump the current generic Target Engine configuration into the file tcm_backup-2008-08-08_BASE in the directory /etc/target/backup:

```
# tcm_dump --t 2009-08-08_BASE
Making backup of Target_Core_Mod/ConfigFS with timestamp: 2009-08-08_BASE
```

lio_dump --t <FileName>

Example: Dump the current iSCSI Target configuration into the file lio_backup-2008-08-08_BASE in the directory /etc/target/backup:

```
# lio_dump --t 2009-08-08_BASE
Making backup of LIO-Target/ConfigFS with timestamp: 2009-08-08_BASE
```

The Target configuration backups are preserved in /etc/target/backup.

iSCSI Storage Objects

An iSCSI *Storage Object* is formed by configuring or creating a *Backstore* with a local name for the Storage Object. The typical Backstore types for iSCSI Targets are:

- File IO
- Block IO
- Raw SCSI disk
- RAMDISK

The Backstore is referenced by its *Virtual HBA* (host bus adapter) type and the Storage Object name.

The Target then generates a *T10 WWN Unit Serial* number for the Storage Object, by which is a world-wide unique ID for the Storage Object.

File Backstore

Using an ordinary Linux/VFS file requires a filename and size (if it is a file on a mounted filesystem), or a filename with size 0 if the file is referencing an underlying block device (e.g., /dev/sde).

Create a file for the Backstore

tcm_node --fileio fileio_<HBA>/<StorageObjectName> <PathName>/<FileName> <Size_in_Bytes#>

Example: Create a Storage Object with the name my_file0 and create a 1GB file in the ~/tmp directory as Backstore for it:

```
# tcm_node --fileio fileio_0/my_file0 /home/marcf/tmp/my_file0 1000000000
 ConfigFS HBA: fileio_0
Successfully added TCM/ConfigFS HBA: fileio_0
 ConfigFS Device Alias: my_file0
Device Params ['fd_dev_name=/home/marcf/tmp/my_file0,fd_dev_size=1000000000']
Status: DEACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 1024
        LIO FILEIO ID: 0        File: /home/marcf/tmp/my_file0  Size: 1000000000
Set T10 WWN Unit Serial for fileio_0/my_file0 to: 1201ae93-93b5-4c8b-881c-9627697162d6
Successfully created TCM/ConfigFS storage object:
/sys/kernel/config/target/core/fileio_0/my_file0
```

Upon success a T10 WWN Unit Serial will be generated for the Storage Object, e.g.: 1201ae93-93b5-4c8b-881c-9627697162d6

Display the resulting T10 WWN Unit Serial

tcm_node --wwn <HBA>/ <StorageObjectName>

Example: Display the T10 WWN for the Storage Object fileio_0/my_file0.

```
# tcm_node --wwn fileio_0/my_file0
T10 VPD Identifier Association: addressed logical unit
T10 VPD Identifier Type: NAA
T10 VPD Binary Device Identifier: 360014051201ae93d93b5d4c8bd881cd9
T10 VPD Identifier Association: addressed logical unit
T10 VPD Identifier Type: T10 Vendor ID based
T10 VPD ASCII Device Identifier: LIO-ORG
T10 VPD Unit Serial Number: 1201ae93-93b5-4c8b-881c-9627697162d6
```

Block IO Backstore

Using a block device for Backstore requires a udev path from either:

- /dev
- /dev/mapper
- /dev/*VolumeGroup*/*LogicalVolume*
- /dev/disk

tcm_node --block <HBA>/<StorageObjectName> <PathName>/<BlockDeviceName>

Allocate a block IO device for the Backstore

Example: Create a Storage Object with the name my_iblock0 and allocate the third SCSI disk in the system (/dev/sdc) as Backstore for it:

```
# tcm_node --block iblock_0/my_iblock0 /dev/sdc
 ConfigFS HBA: iblock_0
 ConfigFS Device Alias: my_iblock0
Device Params ['/dev/sdc']
Status: DEACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 128
        iBlock device: sdc
        Major: 8 Minor: 32  CLAIMED: IBLOCK
Set T10 WWN Unit Serial for iblock_0/my_iblock0 to: 5ec70fa1-4302-495a-8717-
f5af1d4ace8a
Successfully created TCM/ConfigFS storage object:
/sys/kernel/config/target/core/iblock_0/my_iblock0
```

This is how it would looks like for a physical disk, but the same applies to any block device that appears in /dev, e.g., see blow for logical volumes.

Upon success a T10 WWN Unit Serial number will be generated for the Storage Object.

Allocate a logical volume for the Backstore

Example: Create a Storage Object with the name my_lv0 and allocate the LVM logical volume lv0 in the first volume group vg0 (/dev/vg0/lv0) as Backstore for it:

```
# tcm_node --block iblock_0/my_lv0 /dev/vg0/lv0
 ConfigFS HBA: iblock_0
 ConfigFS Device Alias: my_lv0
Device Params ['/dev/vg0/lv0']
Status: DEACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 128
        iBlock device: dm-0
        Major: 253 Minor: 0  CLAIMED: IBLOCK
Set T10 WWN Unit Serial for iblock_0/my_lv0 to: e326ea5a-5ce8-4e78-afef-b429138a38f3
Successfully created TCM/ConfigFS storage object:
/sys/kernel/config/target/core/iblock_0/my_lv0
```

Upon success a T10 WWN Unit Serial number will be generated for the Storage Object.

Ramdisk Backstore

Create a Ramdisk for the Backstore

Ramdisk configurations require a *PageCount* value to define the size of the ramdisk. The page size is 4k bytes (4096 bytes).

tcm_node --ramdisk <HBA>/<StorageObjectName> <PageCount#>

Example: Create a Storage Object with the name my_rd0 and create a 4MB ramdisk as Backstore for it:

```
# tcm_node --ramdisk rd_dr_0/my_rd0 1024
 ConfigFS HBA: rd_dr_0
Successfully added TCM/ConfigFS HBA: rd_dr_0
 ConfigFS Device Alias: my_rd0
Device Params ['1024']
Status: DEACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 1024
        LIO RamDisk ID: 0  RamDisk Makeup: rd_direct
        PAGES/PAGE_SIZE: 1024*4096  SG_table_count: 1
```

Releasing Storage Objects

Release a Storage Object in a Backstore

tcm_node --freedev <HBA>/<StorageObjectName>

Example: Release a single file Storage Object (for file Backstores):

```
# tcm_node --freedev fileio_0/my_file0
 ConfigFS HBA: fileio_0
 ConfigFS Device Alias: my_file0
Successfully released TCM/ConfigFS storage object:
/sys/kernel/config/target/core/fileio_0/my_file0
```

Example: Release a single ramdisk Storage Object (for ramdisk Backstores):

```
# tcm_node --freedev rd_dr_0/lun_my_rd
 ConfigFS HBA: rd_dr_0
 ConfigFS Device Alias: lun_my_rd
Successfully released TCM/ConfigFS storage object:
/sys/kernel/config/target/core/rd_dr_0/lun_my_rd
```

Release an entire Backstore

Note: For the Storage Objects being released in the Backstore, there must be no active iSCSI exports, otherwise this operation will currently fail, as it doesn't yet automatically freeing all active iSCSI export.

tcm_node --delhba <HBA>

Example: Release an entire file Backstore including all of its associated Storage Objects:

```
# tcm_node --delhba fileio_0
hba_path: /sys/kernel/config/target/core/fileio_0
 ConfigFS HBA: fileio_0
 ConfigFS Device Alias: my_file0
Successfully released TCM/ConfigFS storage object:
/sys/kernel/config/target/core/fileio_0/my_file0
Successfully released TCM HBA: /sys/kernel/config/target/core/fileio_0
```

Example: Release an entire ramdisk Backstore including all of its associated Storage Objects:

```
# tcm_node --delhba rd_dr_0
hba_path: /sys/kernel/config/target/core/rd_dr_0
 ConfigFS HBA: rd_dr_0
 ConfigFS Device Alias: lun_my_rd
Successfully released TCM/ConfigFS storage object:
/sys/kernel/config/target/core/rd_dr_0/lun_my_rd
Successfully released TCM HBA: /sys/kernel/config/target/core/rd_dr_0
```

Storage Object Attributes

List Attributes

tcm_node --listdevattr <HBA>/<StorageObjectName>

Example: List the attributes of the block Storage Object my_iblock0:

```
# tcm_node --listdevattr iblock_0/my_iblock0
TCM Storage Object Attributes for /sys/kernel/config/target/core/iblock_0/my_iblock0
       task_timeout: 0
       queue_depth: 32
       hw_queue_depth: 32
       max_sectors: 128
       hw_max_sectors: 128
       block_size: 512
       hw_block_size: 512
       emulate_tas: 1
       emulate_ua_intlck_ctrl: 0
```

Most of these attributes reflect the properties of the underlying Storage Object and thus are read-only.

Set Attribute

tcm_node --setdevattr <HBA>/<StorageObjectName>

Example: Set the task timeout time of the block Storage Object my_iblock0:

```
# tcm_node --setdevattr iblock_0/my_iblock0 task_timeout 30
Successfully set TCM storage object attribute: task_timeout=30 for
/sys/kernel/config/target/core/iblock_0/my_iblock0/attrib/task_timeout
```

iSCSI Endpoints

An iSCSI *Endpoint* is formed by associating an *iSCSI Qualified Name* (IQN) with an iSCSI *Target Portal Group* (TPG).

A TPG is formed by associating a Storage Object with one or more LUNs, and identified by the Target's IQN plus a TPG tag and a LUN tag. The Target IQN is a freely user-definable alphanumeric string. For production configurations, the recommended Target IQN is the Target's T10 WWN (see above), which makes that iSCSI Endpoint world-wide uniquely referenceable. The TPG and LUN tags are simple ordinal numbers.

For simplicity, for the Target IQN in the following examples the local name "iscsi-test" is used.

Create an iSCSI Endpoint

lio_node --addlun <TargetIQN> <TPG#> <LUN#> <LUNName> <HBA>/<StorageObjectName>

Note: To make a new LUN visible for the Initiator, the Initiator needs to expressly *Rescan Disks*.

Example: Create the iSCSI Endpoint iscsi-test from a file Backstore (my_file0):

```
# lio_node –-addlun iscsi-test 1 0 lun_my_file fileio_0/my_file0
Successfully created iSCSI Target Logical Unit
```

Example: Create the iSCSI Endpoint iscsi-test from a block Backstore (my_iblock0):

```
# lio_node --addlun iscsi-test 1 1 lun_my_block iblock_0/my_iblock0
Successfully created iSCSI Target Logical Unit
```

Example: Create the iSCSI Endpoint iscsi-test from a logical volume Backstore (my_lv0):

```
# lio_node --addlun iscsi-test 1 2 lun_my_lv iblock_0/my_lv0
Successfully created iSCSI Target Logical Unit
```

Example: Create the iSCSI Endpoint iscsi-test from a ramdisk Backstore (my_rd0):

```
# lio_node --addlun iscsi-test 1 3 lun_my_rd rd_dr_0/my_rd0
Successfully created iSCSI Target Logical Unit
```

List all iSCSI Endpoints

lio_node --listendpoints

Example: List all iSCSI Endpoints:

```
# lio_node --listendpoints
\------> iscsi-test
        \-------> tpgt_1  TargetAlias: LIO Target
         TPG Status: DISABLED
         TPG Network Portals:
         TPG Logical Units:
                 \-------> lun_0/lun_my_file -> target/core/fileio_0/my_file0
                 \-------> lun_1/lun_my_block -> target/core/iblock_0/my_iblock0
                 \-------> lun_2/lun_my_lv -> target/core/iblock_0/my_lv0
                 \-------> lun_3/lun_my_rd -> target/core/rd_dr_0/my_rd0
```

Remove a single iSCSI LUN from an iSCSI Endpoint

lio_node --dellun <TargetIQN> <TPG#> <LUN#>

Example: Delete LUN 0 in the TPG 1 of the iSCSI Endpoint iscsi-test:

```
# lio_node –dellun iscsi-test 1 0
Successfully deleted iSCSI Target Logical Unit
```

Delete a TPG in an iSCSI Endpoint

lio_node --deltpg <TargetIQN> <TPG#>

Example: Delete the TPG 1 of the iSCSI Endpoint iscsi-test:

```
# lio_node --deltpg iscsi-test 1
Successfully released iSCSI Target Portal Group: iscsi-test TPGT: 1
```

Remove an entire iSCSI Endpoint

Removes an iSCSI Endpoint including all of its associated LUNs.

lio_node --deliqn <TargetIQN>

Example: Delete the entire iSCSI Endpoint iscsi-test:

```
# lio_node --deliqn iscsi-test
Successfully released network portal: 192.168.1.10:3260
Successfully deleted iSCSI Target Logical Unit
Successfully released iSCSI Target Portal Group: iscsi-test TPGT: 1
Successfully released iSCSI Target Endpoint IQN: iscsi-test
```

iSCSI Network Portals

A iSCSI *Network Portal* is formed by adding an IP address and TCP Port number to an iSCSI Endpoint. The IP address can be an IPv4 or an IPv6 address (note that the IPv6 address notation is in square brackets). The TCP port number for the iSCSI protocol is defined by the IANA as 3260.

Create an iSCSI Network Portal

lio_node --addnp <TargetIQN> <TPG#> <IPaddr>:<Port#>

Example: Create an iSCSi Network Portal from the Endpoint iscsi-test on the node 192.168.1.10:

```
# lio_node --addnp iscsi-test 1 192.168.1.10:3260
Successfully created network portal: 192.168.1.10:3260 created iscsi-test TPGT: 1
```

Delete an iSCSI Network Portal

lio_node --delnp <TargetIQN> <TPG#> <IPaddr>:<Port#>

Example: Delete the iSCSi Network Portal on the Endpoint iscsi-test on the node 192.168.1.10:

```
# lio_node --delnp iscsi-test 1 192.168.1.10:3260
Successfully released network portal: 192.168.1.10:3260 created iscsi-test TPGT: 1
```

iSCSI Endpoints – Attributes

iSCSI Initiators can now log into the Target, but they still need to authenticate themselves.

The Target can either be run with CHAP authentication disabled (i.e. no ACLs), which is not recommended for production configurations and hence called “demo mode”.

In production environments, CHAP authentication can be enabled for Initiators only, or mutual CHAP Initiator/Target authentication can be enforced.

Disabling Authentication

Authentication should not be disabled in production setups, hence it is referred to as “demo mode”. If demo mode is enabled, all Initiators can access all LUNs on the specified iSCSI Endpoint.

Enable demo mode

lio_node --demomode <TargetName> <TPG#>

Example: Enable demo mode on the iSCSI TPG iscsi-test #1:

```
# lio_node --demomode iscsi-test 1
Successfully disabled Initiator ACL mode (Enabled DemoMode) on iSCSI Target Portal
Group: iscsi-test 1
```

Disable CHAP authentication

lio_node --disableauth <TargetName> <TPG#>

Example: Disable authentication on the iSCSI TPG iscsi-test #1:

```
# lio_node --disableauth iscsi-test 1
Successfully disabled iSCSI Authentication on iSCSI Target Portal Group: iscsi-test 1
```

Demo mode by default write protects disk backstores to prevent erroneous data overwrites.

Disable write protection for demo Targets

Note: The configFS tree can be "tabbed" through.

Example: Disable write protection on the iSCSI TPG iscsi-test #1:

```
# echo 0 > /sys/kernel/config/target/iscsi/iscsi-
test/tpgt_1/attrib/demo_mode_write_protect
```

Example: Verify that write protection has been disabled successfully on the iSCSI TPG iscsi-test #1:

```
# cat /sys/kernel/config/target/iscsi/iscsi-test/tpgt_1/attrib/demo_mode_write_protect
0
```

You can find more useful information in /proc/scsi_target/mib/* and /proc/iscsi_target/mib/* for the SNMP MIBs to help troubleshoot.

Enabling Authentication

First, make sure the Initiator is logged out of the Target.

Enforce ACLs

Enabling ACL mode is optional if it hasn't been explicitly disabled before (e.g., by configuring demo mode).

lio_node --enableaclmode <TargetIQN> <TPG#>

Example: Enable Initiator ACL on the iSCSI TPG iscsi-test #1:

```
# lio_node --enableaclmode iscsi-test 1
Successfully enabled Initiator ACL mode (Disabled DemoMode) on iSCSI Target Portal
Group: iscsi-test 1
```

Allow the iSCSI Initiator to login into an Endpoint

Note: The iSCSI Initiator is referenced by its InitiatorIQN.

lio_node --addnodeacl <TargetIQN> <TPG#> <InitiatorIQN>

Example: Add the Initiator node iqn.1991-05.com.microsoft:lenovo-8f5a784d to the ACL on the iSCSI TPG iscsi-test #1:

```
# lio_node --addnodeacl iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d
```

```
Successfully added iSCSI Initaitor ACL iqn.1991-05.com.microsoft:lenovo-8f5a784d for
iSCSI Target Portal Group: iscsi-test 1
```

List the currently configured node ACLs

The current configured node ACLs reflect the permitted Initiators, along with their running sessions.

lio_node --listnodeacls <TargetIQN> <TPG#>

Example: List a completed ACL configuration with qualified Initiator logins:

```
# lio_node --listnodeacls iscsi-test 1
\------> InitiatorName: iqn.1991-05.com.microsoft:lenovo-8f5a784d
         InitiatorAlias:
         LIO Session ID: 11   ISID: 0x40 00 01 37 00 00  TSIH: 11  SessionType: Normal
         Session State: TARG_SESS_STATE_LOGGED_IN
         ---------------------[iSCSI Session Values]-----------------------
           CmdSN/WR  :  CmdSN/WC  :  ExpCmdSN  :  MaxCmdSN  :     ITT    :     TTT
          0x00000010   0x00000010   0x000001e0   0x000001ef   0x000001df   0x000002bc
         ----------------------[iSCSI Connections]-------------------------
         CID: 1  Connection State: TARG_CONN_STATE_LOGGED_IN
            Address 192.168.1.65 TCP  StatSN: 0x000001e4
```

Make an iSCSI LUN accessible for the Initiator

Note: The iSCSI Initiator is referenced by its InitiatorIQN.

lio_node --addlunacl <TargetIQN> <TPG#> <InitiatorIQN> <LUN#> <MappedLUN#>

Note that the concept of MappedLUN# is that the initiator LUN ACLs can arbitrarily map any of the LUN# values to the preferred value for a particular Initiator. The MappedLUN# is the LUN# that the Initiator sees. So a mapping could be LUN#=1 and MappedLUN#=0 to make the iSCSI Target endpoint LUN=1 appear as LUN=0 on the Initiator. The mapping are typically identical.

Example: Add the file backstore to the Initiator ACL (identically mapped):

```
# lio_node --addlunacl iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d 0 0
Successfully added iSCSI Initiator Mapped LUN: 0 ACL iqn.1991-05.com.microsoft:lenovo-
8f5a784d for iSCSI Target Portal Group: iscsi-test 1
```

Example: Add the block backstore to the Initiator ACL (identically mapped):

```
# lio_node --addlunacl iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d 1 1
Successfully added iSCSI Initiator Mapped LUN: 1 ACL iqn.1991-05.com.microsoft:lenovo-
8f5a784d for iSCSI Target Portal Group: iscsi-test 1
```

Example: Add the ramdisk backstore to the Initiator ACL (identically mapped):

```
# lio_node --addlunacl iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d 2 2
Successfully added iSCSI Initiator Mapped LUN: 2 ACL iqn.1991-05.com.microsoft:lenovo-
8f5a784d for iSCSI Target Portal Group: iscsi-test 1
```

List the currently configured LUN ACLs

lio_node --listlunacls <TargetIQN> <TPG#>

Example: List all LUN ACLs on the iSCSI Endpoint iscsi-test #1 (grouped by iSCSI Initiator):

```
# lio_node --listlunacls iscsi-test 1
\------> InitiatorName ACL: iqn.1991-05.com.microsoft:lenovo-8f5a784d
         Logical Unit ACLs:
         \-------> lun_0 -> target/iscsi/iscsi-test/tpgt_1/lun/lun_0
                   \-------> Write Protect for lun_0: DISABLED
         \-------> lun_1 -> target/iscsi/iscsi-test/tpgt_1/lun/lun_1
                   \-------> Write Protect for lun_1: DISABLED
         \-------> lun_2 -> target/iscsi/iscsi-test/tpgt_1/lun/lun_2
                   \-------> Write Protect for lun_2: DISABLED
```

Note that the iSCSI TCQ depth is different from the TCQ depth of the storage object. The iSCSI TCQ depth is enforced by the Target to determine how many outstanding I/Os a given Initiator can send to all of the LUNs for a given initiator on a given target endpoint. Once an I/O has been cleared by the iSCSI TCQ, it still is enforced by the underlying storage object TCQ, which is shared for the storage objects across all of the fabric module (iSCSI) exports.

Configuring CHAP Initiator Authentication

First, make sure the Initiator is logged out of the Target. Enable node ACLs to be enforced, add the Initiator to the node ACL list, and make each LUN accessible for the Initiator, as described above.

Enable CHAP Initiator authentication

lio_node --setchapauth <TargetIQN> <TPG#><InitiatorIQN> <LoginName> <Password>

Example: Enable CHAP Initiator authentication on the iSCSI Endpoint iscsi-test #1:

```
# lio_node --setchapauth iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d
iqn.1991-05.com.microsoft:lenovo-8f5a784d mytargetsecret
Successfully set CHAP authentication for iSCSI Initaitor ACL iqn.1991-
05.com.microsoft:lenovo-8f5a784d for iSCSI Target Portal Group: iscsi-test 1
```

Note that the Microsoft iSCSI Initiator uses its Initiator IQN as a login name, and it has a minimal password length requirement of 96 bits.

Show the CHAP authentication configuration

lio_node --showchapauth <TargetIQN> <TPG#><InitiatorIQN>

Example: Display the CHAP authentication configuration of the iSCSI Endpoint iscsi-test #1:

```
# lio_node --showchapauth iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d
password_mutual:
userid_mutual:
authenticate_target: 0
password: mytargetsecret
userid: iqn.1991-05.com.microsoft:lenovo-8f5a784d
```

Then relogin the Initiator into the Target.

Configuring CHAP Mutual Authentication

First, make sure the Initiator is logged out of the Target. Enable node ACLs to be enforced, add the Initiator to the node ACL list, and make each LUN accessible for the Initiator, as described above.

Enable CHAP Mutual authentication

lio_node --setchapmutualauth <TargetIQN> <TPG#><InitiatorIQN> <LoginName> <Password>

Example: Enable CHAP Mutual authentication on the iSCSI Endpoint iscsi-test #1:

```
# lio_node --setchapmutualauth iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d
iqn.1991-05.com.microsoft:lenovo-8f5a784d myinitiatorsecret
Successfully set mutual CHAP authentication for iSCSI Initaitor ACL iqn.1991-
05.com.microsoft:lenovo-8f5a784d for iSCSI Target Portal Group: iscsi-test 1
```

Show the CHAP authentication configuration

lio_node --showchapauth <TargetIQN> <TPG#><InitiatorIQN>

Example: Display the CHAP authentication configuration of the iSCSI Endpoint iscsi-test #1:

```
# lio_node --showchapauth iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d
password_mutual: myinitiatorsecret
userid_mutual: iqn.1991-05.com.microsoft:lenovo-8f5a784d
authenticate_target: 1
password: mytargetsecret
userid: iqn.1991-05.com.microsoft:lenovo-8f5a784d
```

Then relogin the Initiator into the Target.

Optional Write Protection

LUNs are created writable by default. They are protected by automatically demoting them to read-only when the Target is configured in Demo Mode. Also, the read-only attribute can be explicitly toggled.

Enable write protection for a LUN

lio_node --enablelunwp <TargetIQN> <TPG#> <InitiatorIQN> <LUN#> <MappedLUN#>

Example: Enable write protection on LUN #2 on iSCSi Endpoint iscsi-test #1:

```
# lio_node --enablelunwp iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d 2 2
Successfully enabled WRITE PROTECT for Mapped LUN: 2 for iqn.1991-
05.com.microsoft:lenovo-8f5a784d on iSCSI Target Portal Group: iscsi-test 1
```

Disable write protection for a LUN

lio_node --disablelunwp <TargetIQN> <TPG#> <InitiatorIQN> <LUN#> <MappedLUN#>

Example: Disable write protection for LUN #2 on iSCSi Endpoint iscsi-test #1:

```
# lio_node --disablelunwp iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d 2 2
Successfully disabled WRITE PROTECT for Mapped LUN: 2 for iqn.1991-
05.com.microsoft:lenovo-8f5a784d on iSCSI Target Portal Group: iscsi-test 1
```

Other Attributes

Set the TCQ depth for an Initiator

Note that the Initiator specific TCQ depth is kept with the other Initiator attributes in the ACL.

lio_node --setnodetcq <TargetIQN> <TPG#> <InitiatorIQN> 18

Example: Set the TCQ (Tagged Command Queue) depth for the iSCSI Endpoint iscsci-test #1 to 32:

```
# lio_node --shownodetcq iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d 32
Successfully set TCQ: 32 on iSCSI Target Portal Group: iscsi-test 1
```

Show the TCQ depth for an Initiator

lio_node --shownodetcq <TargetIQN> <TPG#> <InitiatorIQN>

Example: List the TCQ (Tagged Command Queue) depth for the iSCSI Endpoint iscsci-test #1:

```
# lio_node --shownodetcq iscsi-test 1 iqn.1991-05.com.microsoft:lenovo-8f5a784d
16
```

Set the ALUA Target Port Group Name for a LUN

lio_node --settgptgp <TargetIQN> <TPG#> <LUN#> <TargetPortGroupName>

Example: Set the ALUA Target Port Group name for LUN #0 on the iSCSI Endpoint iscsci-test #1.

```
# lio_node --settgptgp iscsi-test 1 0 default_tg_pt_gp
sh: line 0: echo: write error: No such device
Unable to set ALUA Target Port Group: default_tg_pt_gp for LUN: 0 on iscsi-test 1
```

Show the ALUA Target Port Group Name for a LUN

lio_node --showtgptgp <TargetIQN> <TPG#> <LUN#>

Example: Show the ALUA Target Port Group name for LUN #0 on the iSCSI Endpoint iscsci-test #1:

```
# lio_node --showtgptgp iscsi-test 1 0
TG Port Alias: default_tg_pt_gp
TG Port Group ID: 0
```

iSCSI Endpoints – Enabling and Disabling

Allow logins from iSCSI Initiators into the iSCSI Endpoint

Example: Enable Initiator logins into the iSCSI Endpoint iscsi-test #1:

lio_node --enabletpg <TargetIQN> <TPG#>

```
# lio_node --enabletpg iscsi-test 1
Successfully enabled iSCSI Target Portal Group: iscsi-test 1
```

Example: List all iSCSI Endpoints on a Target:

```
# lio_node --listendpoints
\------> iscsi-test
        \-------> tpgt_1  TargetAlias: LIO Target
         TPG Status: ENABLED
         TPG Network Portals:
                 \-------> 192.168.1.10:3260
         TPG Logical Units:
                 \-------> lun_2/lun_my_rd -> target/core/rd_dr_0/my_rd
                 \-------> lun_1/lun_my_block -> target/core/iblock_0/my_iblock0
                 \-------> lun_0/lun_my_file -> target/core/fileio_0/my_file0
```

Disable an iSCSI Endpoint

lio_node --disabletpg <TargetIQN> <TPG#>

Example: Disable Initiator logins into the iSCSI Endpoint iscsi-test #1:

```
# lio_node --disabletpg iscsi-test 1
Successfully disabled iSCSI Target Portal Group: iscsi-test 1
```

Displaying Configuration Data

Generic Target Engine

List all Storage Objects

Example: List all Storage Objects on a Target:

```
# tcm_node --listhbas
\------> rd_dr_0
        HBA Index: 4 plugin: rd_dr version: v1.0.0
        \-------> lun_my_rd
        Status: ACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 1024
        LIO RamDisk ID: 1  RamDisk Makeup: rd_direct
        PAGES/PAGE_SIZE: 1024*4096  SG_table_count: 1
        udev_path: N/A

\------> iblock_0
        HBA Index: 1 plugin: iblock version: v1.0.0
        \-------> my_iblock0
        Status: ACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 128
        iBlock device: sdc
        Major: 8 Minor: 32  CLAIMED: IBLOCK
        udev_path: /dev/sdc

\------> fileio_0
        HBA Index: 0 plugin: fileio version: v1.0.0
        \-------> my_file0
        Status: ACTIVATED  Execute/Left/Max Queue Depth: 0/32/32  SectorSize: 512
MaxSectors: 1024
        LIO FILEIO ID: 0        File: /home/marcf/tmp/my_file0  Size: 1000000000
        udev_path: N/A
```

List all ALUA LUN groups

Example: List all ALUA LUN groups on a Target:

```
# tcm_node --listlugps
\------> default_lu_gp  LUN Group ID: 0
         fileio_0/my_file0
         iblock_0/my_iblock0
         rd_dr_0/lun_my_rd
```

For pSCSI storage objects don't show up in this list, because LIO hasn't implemented any of the control emulation on the CDB pass through yet.

List all ALUA TPGs

Example: List all ALUA TPGs groups on a Target:

```
# tcm_node --listtgptgps
\------> default_tg_pt_gp  Target Port Group ID: 0
         iSCSI/iscsi-test/tpgt_1/lun_0
         iSCSI/iscsi-test/tpgt_1/lun_1
```

List Persistent Reservations

tcm_node --pr <HBA>/<StorageObjectName>

Example: List all PRs on the Storage Object my_iblock_0:

```
# tcm_node --pr iblock_0/my_iblock0
No SPC-3 Reservation holder
No SPC-3 Reservation holder
0x00000000
No SPC-3 Reservation holder
SPC-3 PR Registrations:
None
No SPC-3 Reservation holder
SPC3_PERSISTENT_RESERVATIONS
```

Show the udev Path for a Storage Object

tcm_node --udevpath <HBA>/<StorageObjectName>

Example: Display the udev path information of the Storage Object my_iblock0:

```
# tcm_node --udevpath iblock_0/my_iblock0
/dev/sdc
```

Show the T10 WWN info of a Storage Object

Note: The T10 WWN is automatically created and can be modified only if there are no active exports in the Target. In contrast, the IQN is user selectable, and so a user-defined, simpler name can be chosen.

tcm_node --wwn <HBA> <StorageObjectName>

Example: Display the T10 WWN of the Storage Object my_iblock0:

```
# tcm_node --wwn iblock_0/my_iblock0
T10 VPD Identifier Association: addressed logical unit
T10 VPD Identifier Type: NAA
T10 VPD Binary Device Identifier: 360014055ec70fa1d4302d495ad8717df
T10 VPD Identifier Association: addressed logical unit
T10 VPD Identifier Type: T10 Vendor ID based
T10 VPD ASCII Device Identifier: LIO-ORG
T10 VPD Unit Serial Number: 5ec70fa1-4302-495a-8717-f5af1d4ace8a
```

iSCSI Transport Module

List the Target Names

Example: List the Target IQNs on the Target:

```
# lio_node –-listtargetnames
iscsi-test
```

List the Network Portals of a Target

Example: List the Network Portals on the iSCSI Endpoint iscsi-test #1:

```
lio_node --listnps iscsi-test 1
192.168.1.10:3260
```

List all Endpoints

Example: List the iSCSI Endpoints on the Target:

```
# lio_node --listendpoints
\------> iscsi-test
        \-------> tpgt_1  TargetAlias: LIO Target
         TPG Status: ENABLED
         TPG Network Portals:
                 \-------> 192.168.1.10:3260
         TPG Logical Units:
                 \-------> lun_2/lun_my_rd -> target/core/rd_dr_0/my_rd
                 \-------> lun_1/lun_my_block -> target/core/iblock_0/my_iblock0
                 \-------> lun_0/lun_my_file -> target/core/fileio_0/my_file0
```

The configFS Tree – Viewing and Traversing

Example: Install the *tree* utility:

```
# yum install tree
```

Then display the respective branch to get an idea of which parameters the target exposes.

Example: Display the entire Target including all of its branches:

```
# tree /sys/kernel/config/target/
```

All of the interesting parameters are in /sys/kernel/config/target and the read-only MIBs in /proc/[iscsi,scsi]_target/mib/.

Undocumented CLI Operations

- tcm_node --addlungp
- tcm_node --addtgptgp
- tcm_node --creatdev
- tcm_node --dellungp
- tcm_node --deltgptgp
- tcm_node --establishdev
- tcm_node --setlugp
- tcm_node --setludevpath

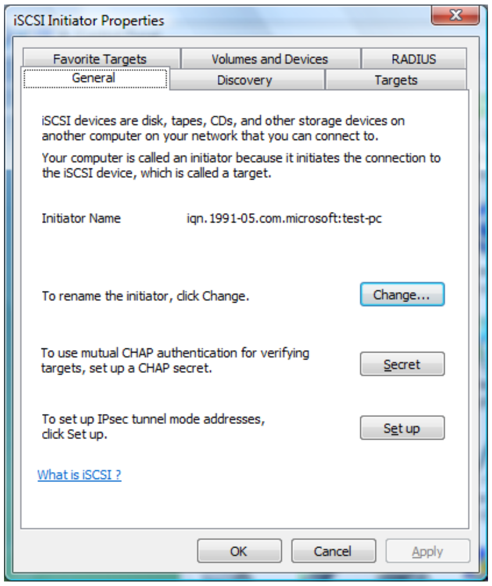
2. Microsoft Windows iSCSI Initiator

The Microsoft iSCSI Software Initiator enables connection of a Windows host to an external iSCSI Targets using Ethernet NICs. Windows Vista and Windows Server 2008 have the iSCSI initiator built-in.

Windows Vista

Connecting to a Target

To connect to an iSCSI Target using Vista or Server 2008, use the built-in iSCSI Initiator. Go to the **Control Panel** and start the **iSCSI Initiator** applet. Select the **Discovery** tab and click the **Add Portal...** button. Enter the Target’s IP address and port information and click **OK**.



iSCSI Initiator Properties

Favorite Targets | Volumes and Devices | RADIUS | General | Discovery | Targets

iSCSI devices are disk, tapes, CDs, and other storage devices on another computer on your network that you can connect to.

Your computer is called an initiator because it initiates the connection to the iSCSI device, which is called a target.

Initiator Name iqn.1991-05.com.microsoft:test-pc

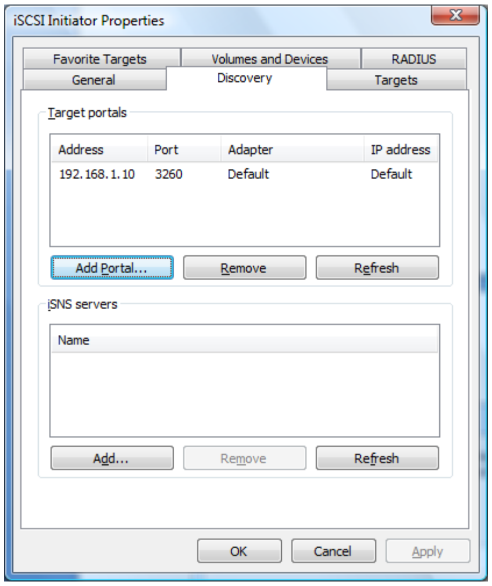
To rename the initiator, click Change. [Change...]

To use mutual CHAP authentication for verifying targets, set up a CHAP secret. [Secret]

To set up IPsec tunnel mode addresses, click Set up. [Set up]

What is iSCSI ?

OK | Cancel | Apply



iSCSI Initiator Properties

Favorite Targets | Volumes and Devices | RADIUS | General | Discovery | Targets

Target portals

Address	Port	Adapter	IP address
192.168.1.10	3260	Default	Default

Add Portal... | Remove | Refresh

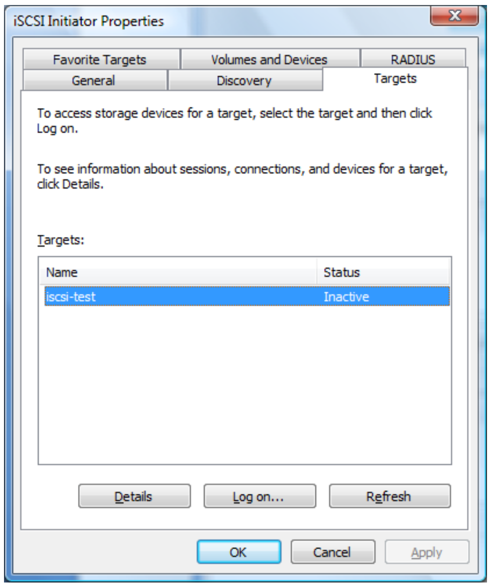
iSNS servers

Name

Add... | Remove | Refresh

OK | Cancel | Apply

Then click the **Targets** tab. A list of the Target’s storage devices appears.



iSCSI Initiator Properties

Favorite Targets | Volumes and Devices | RADIUS | General | Discovery | Targets

To access storage devices for a target, select the target and then click Log on.

To see information about sessions, connections, and devices for a target, click Details.

Targets:

Name	Status
iscsi-test	Inactive

Details | Log on... | Refresh

OK | Cancel | Apply

To connect to a device select it and click **Log on...**, which opens the logon dialog and provides an option to automatically connect to the target at startup and enables multipath. The **Advanced** button allows adapter configuration for the iSCSI connection, authentication, and IPSec information.

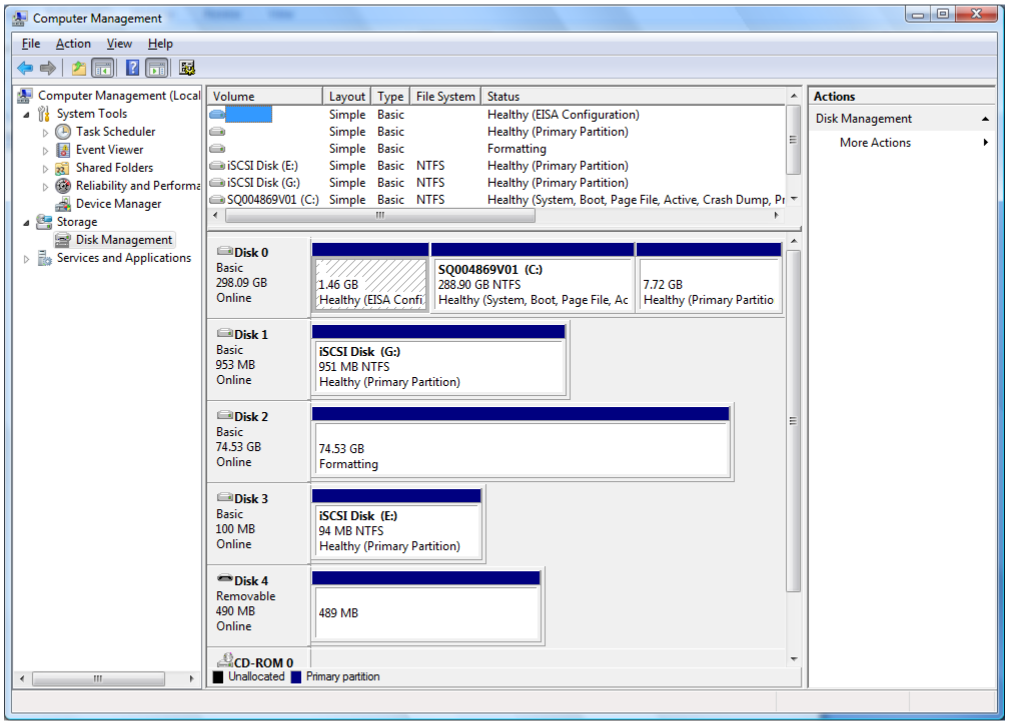
Click **OK** to complete the connection. If the target has been configured before, it will now automatically appear as a Windows volume.

To configure and enable CHAP Authentication, follow the instructions for Windows XP below.

Configuring a Target

Block devices can be configured with the Disk Management snap-in, which is part of the Computer Management console in Administrative Tools in Windows Vista. To open Disk Management, click **Start → Control Panel → System Maintenance** and **Administrative Tools**, then click on the **Computer Management** snap-in. In the **Computer Management** pane, expand **Storage**, and then click **Disk Management**.

The first time Disk Management starts with a new connection and the disk area has not been previously configured, the Disk Manager will prompt to initialize the disk with a particular partition style.



Volume	Layout	Type	File System	Status
	Simple	Basic		Healthy (EISA Configuration)
	Simple	Basic		Healthy (Primary Partition)
	Simple	Basic		Formatting
iSCSI Disk (E:)	Simple	Basic	NTFS	Healthy (Primary Partition)
iSCSI Disk (G:)	Simple	Basic	NTFS	Healthy (Primary Partition)
SQ004869V01 (C:)	Simple	Basic	NTFS	Healthy (System, Boot, Page File, Active, Crash Dump, Pr

Disk 0 Basic 298.09 GB Online	1.46 GB Healthy (EISA Confi	**SQ004869V01 (C:)** 288.90 GB NTFS Healthy (System, Boot, Page File, Ac	7.72 GB Healthy (Primary Partitio
Disk 1 Basic 953 MB Online	**iSCSI Disk (G:)** 951 MB NTFS Healthy (Primary Partition)		
Disk 2 Basic 74.53 GB Online	74.53 GB Formatting		
Disk 3 Basic 100 MB Online	**iSCSI Disk (E:)** 94 MB NTFS Healthy (Primary Partition)		
Disk 4 Removable 490 MB Online	489 MB		
CD-ROM 0			

■ Unallocated ■ Primary partition

The Disk Management window is in four parts. On the left is the Computer Management pane of the Computer Management console. The lower pane contains the information for all the block devices. The upper pane contains descriptions of those drives, including data on the amount of space used and free, the type of file system, and the health of that system. The right pane contains a list of actions you can perform. The list changes based on the object you select.

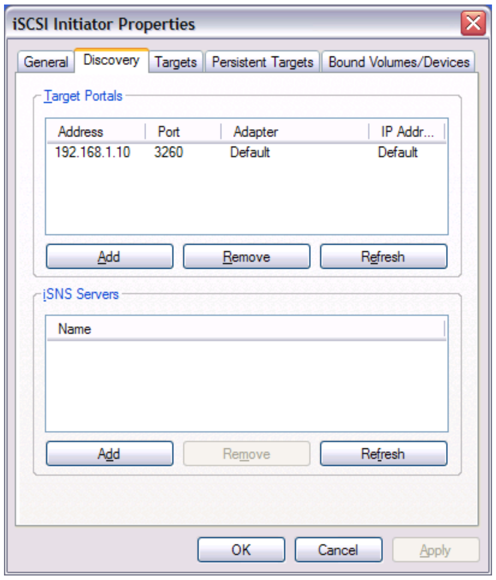
To use Disk Management to create or delete partitions, format drives, and create striped or volume sets, click the drive that you want to modify, and select the change that you want to make from the Action menu or Actions pane. Or right-click on the drive that you want to modify, and then select the appropriate action from the context menu.

Windows XP

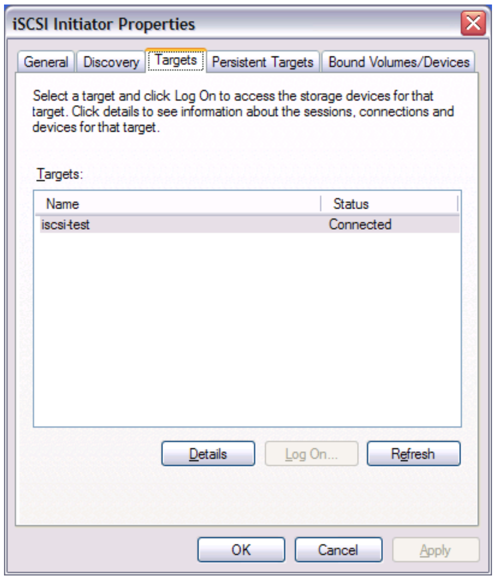
For Windows Server 2003, Windows XP, and Windows 2000, download and install the Microsoft Windows iSCSI Initiator version 2.08: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=12cb3c1a-15d6-4585-b385-befd1319f825

Connecting to a Target

Go to the **Control Panel** and start the **iSCSI Initiator** applet. Select the **Discovery** tab and click the **Add** button. Enter the Target’s IP address and port information and click **OK**.



Address	Port	Adapter	IP Addr...
192.168.1.10	3260	Default	Default



Select a target and click Log On to access the storage devices for that target. Click details to see information about the sessions, connections and devices for that target.

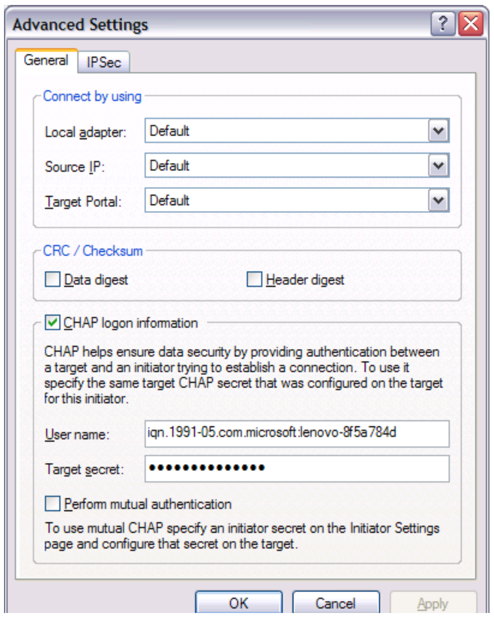
Name	Status
iscsi-test	Connected

Then select the **Targets** tab. A list of the target’s storage devices appears. To connect to a device select it and click **Log On...**, which opens the logon dialog and provides an option to automatically connect to the target at startup and enables multipath. The **Advanced** button allows adapter configuration for the iSCSI connection, authentication, and IPSec information.

Enabling CHAP Initiator Authentication

To enable CHAP Authentication for the Initiator, first make sure the Initiator is not logged into a Target. Then configure CHAP Initiator authentication in the LIO Target, as described above.

In the iSCSI Initiator, click **Targets**, select your Target, click **Log On... → Advanced → General**. Check **CHAP logon information**, enter your **Target secret** and click **OK**.


Advanced Settings

General | IPSec

Connect by using
Local adapter: Default
Source IP: Default
Target Portal: Default

CRC / Checksum
☐ Data digest ☐ Header digest

☑ CHAP logon information

CHAP helps ensure data security by providing authentication between a target and an initiator trying to establish a connection. To use it specify the same target CHAP secret that was configured on the target for this initiator.

User name: iqn.1991-05.com.microsoft:lenovo-8f5a784d
Target secret: ••••••••••••••

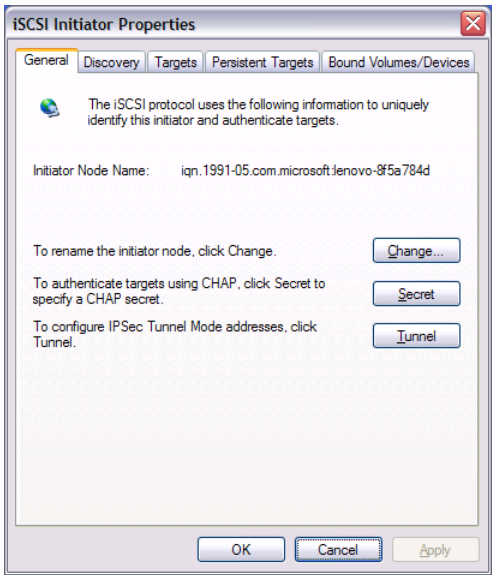
☐ Perform mutual authentication

To use mutual CHAP specify an initiator secret on the Initiator Settings page and configure that secret on the target.

Enabling CHAP Mutual Authentication

To enable CHAP Mutual Authentication, first make sure the Initiator is not logged into a Target. Then enable CHAP Mutual Authentication in the LIO Target, as described above.

In the **iSCSI Initiator**, click **Secret** and enter a "secure secret" (the password required for the Target to login to the client) and click **OK**.


iSCSI Initiator Properties

General | Discovery | Targets | Persistent Targets | Bound Volumes/Devices

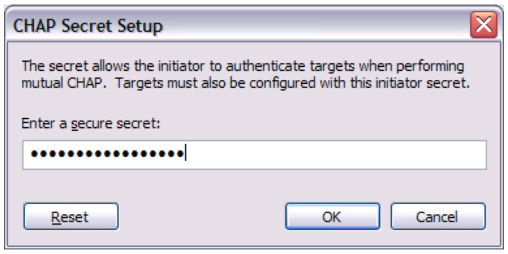
The iSCSI protocol uses the following information to uniquely identify this initiator and authenticate targets.

Initiator Node Name: iqn.1991-05.com.microsoft:lenovo-8f5a784d

To rename the initiator node, click Change. [Change...]

To authenticate targets using CHAP, click Secret to specify a CHAP secret. [Secret]

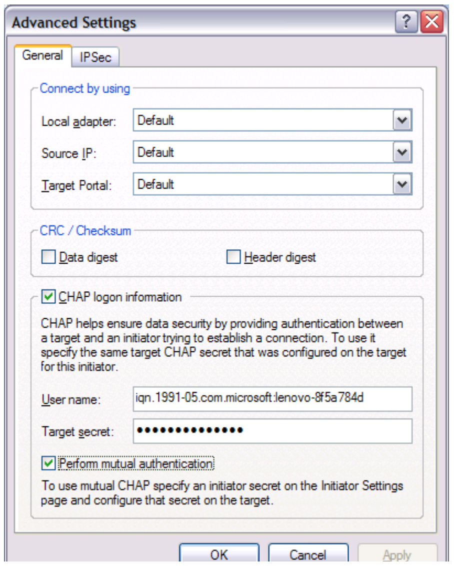
To configure IPSec Tunnel Mode addresses, click Tunnel. [Tunnel]


CHAP Secret Setup

The secret allows the initiator to authenticate targets when performing mutual CHAP. Targets must also be configured with this initiator secret.

Enter a secure secret: •••••••••••••••••

Select the **Targets** tab, click **Log On… → Advanced**, select **CHAP logon information** and check **Perform mutual authentication**. Enter the Initiator's secure secret (password) that the Target expects, and rerun the Disk Manager applet.


Advanced Settings

General | IPSec

Connect by using

Local adapter: Default

Source IP: Default

Target Portal: Default

CRC / Checksum

☐ Data digest ☐ Header digest

☑ CHAP logon information

CHAP helps ensure data security by providing authentication between a target and an initiator trying to establish a connection. To use it specify the same target CHAP secret that was configured on the target for this initiator.

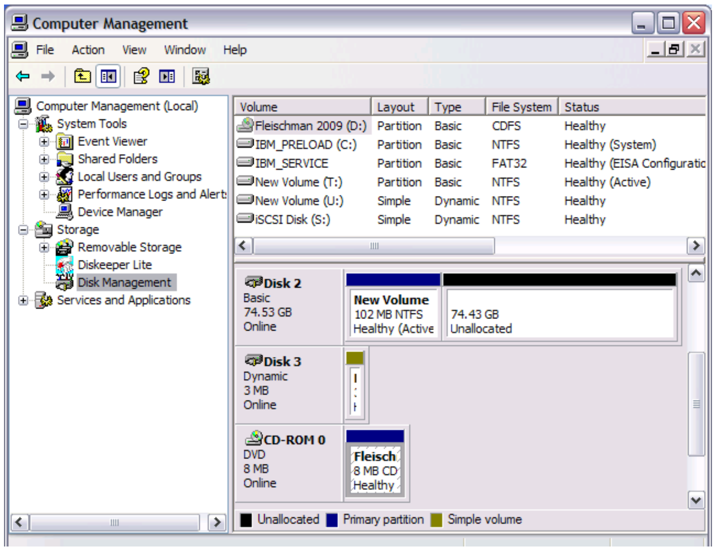
User name: iqn.1991-05.com.microsoft:lenovo-8f5a784d

Target secret: ••••••••••••••

☑ Perform mutual authentication

To use mutual CHAP specify an initiator secret on the Initiator Settings page and configure that secret on the target.

OK | Cancel | Apply


Computer Management

File Action View Window Help

Computer Management (Local)
- System Tools
  - Event Viewer
  - Shared Folders
  - Local Users and Groups
  - Performance Logs and Alerts
  - Device Manager
- Storage
  - Removable Storage
  - Diskeeper Lite
  - Disk Management
- Services and Applications

Volume	Layout	Type	File System	Status
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuratic
New Volume (T:)	Partition	Basic	NTFS	Healthy (Active)
New Volume (U:)	Simple	Dynamic	NTFS	Healthy
iSCSI Disk (S:)	Simple	Dynamic	NTFS	Healthy

Disk 2
Basic
74.53 GB
Online
New Volume
102 MB NTFS
Healthy (Active
74.43 GB
Unallocated

Disk 3
Dynamic
3 MB
Online

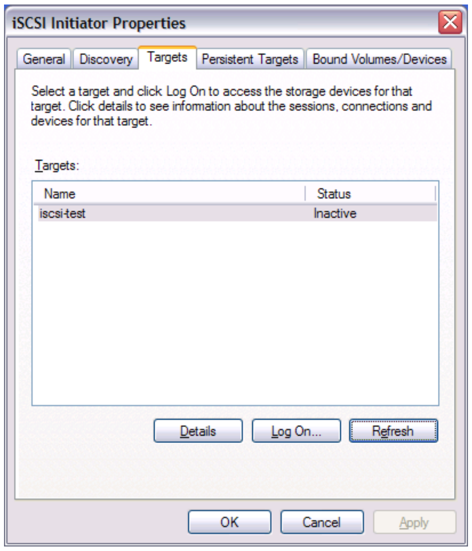
CD-ROM 0
DVD
8 MB
Online
Fleisch
8 MB CD
Healthy

■ Unallocated ■ Primary partition ■ Simple volume

All the iSCSI disks should be accessible again, now with CHAP Mutual Authentication enforced between the iSCSI Initiator and the Target.

Reconnecting to a Target

Start the Microsoft iSCSI Initiator, click **Target → Refresh**. Your iSCSI Target shows up as Inactive in the iSCSI Initiator, and is not yet visible in the Microsoft Disk Manager snap-in:


iSCSI Initiator Properties

General | Discovery | Targets | Persistent Targets | Bound Volumes/Devices

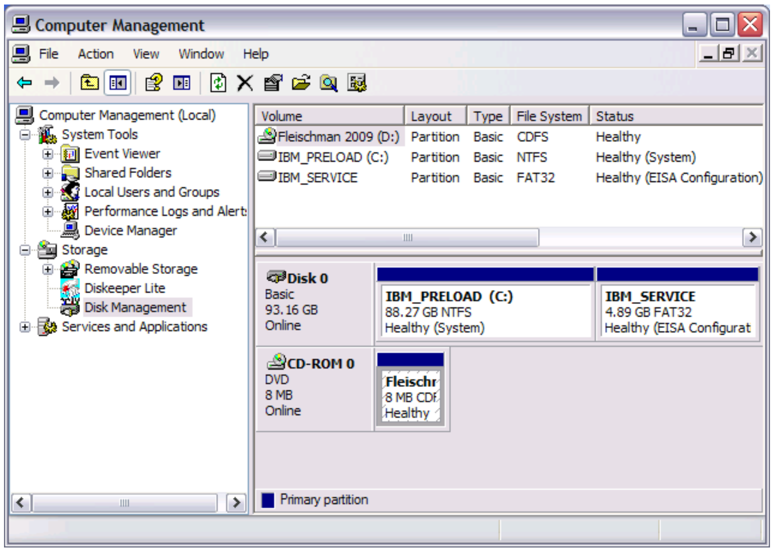
Select a target and click Log On to access the storage devices for that target. Click details to see information about the sessions, connections and devices for that target.

Targets:

Name	Status
iscsi-test	Inactive

Details | Log On... | Refresh

OK | Cancel | Apply


Computer Management

File Action View Window Help

Computer Management (Local)
- System Tools
  - Event Viewer
  - Shared Folders
  - Local Users and Groups
  - Performance Logs and Alerts
  - Device Manager
- Storage
  - Removable Storage
  - Diskeeper Lite
  - Disk Management
- Services and Applications

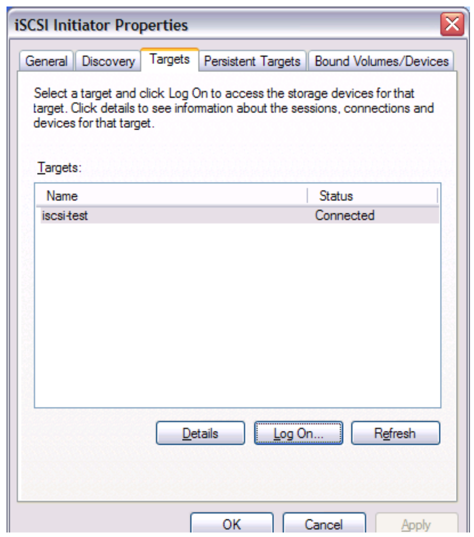
Volume	Layout	Type	File System	Status
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuration)

Disk 0
Basic
93.16 GB
Online
IBM_PRELOAD (C:)
88.27 GB NTFS
Healthy (System)
IBM_SERVICE
4.89 GB FAT32
Healthy (EISA Configurat

CD-ROM 0
DVD
8 MB
Online
Fleischr
8 MB CDF
Healthy

■ Primary partition

In the Microsoft iSCSI Initiator, select the desired iSCSI Target and click **Log On…** The iSCSI Target is now connected, but in the Disk Manager, the volume is still shown as offline.

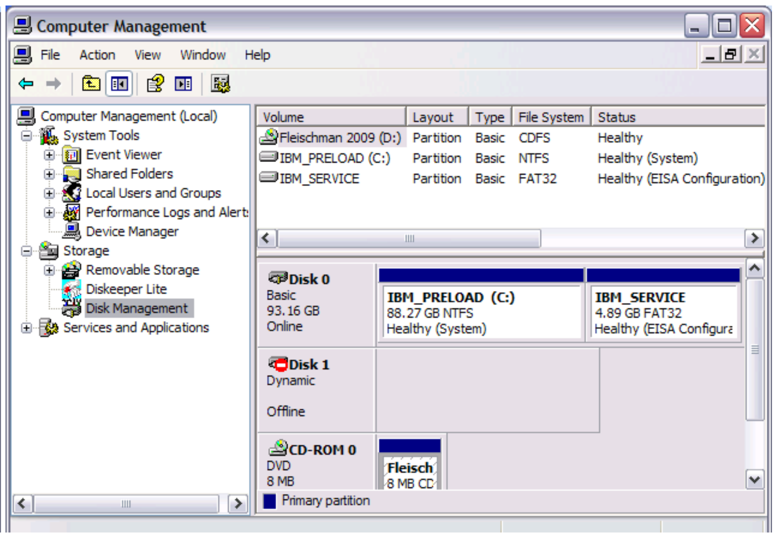


iSCSI Initiator Properties

Select a target and click Log On to access the storage devices for that target. Click details to see information about the sessions, connections and devices for that target.

Targets:

Name	Status
iscsi-test	Connected



Computer Management

Volume	Layout	Type	File System	Status
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuration)

Disk 0
Basic
93.16 GB
Online

IBM_PRELOAD (C:)
88.27 GB NTFS
Healthy (System)

IBM_SERVICE
4.89 GB FAT32
Healthy (EISA Configura

Disk 1
Dynamic

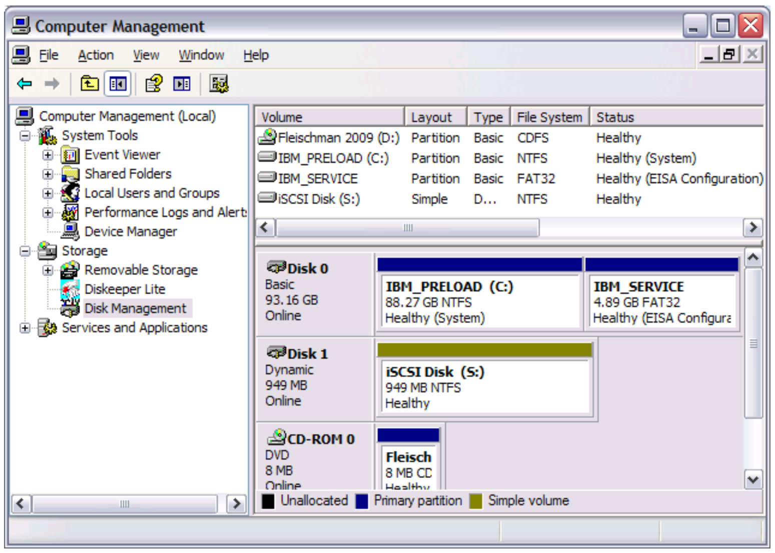
Offline

CD-ROM 0
DVD
8 MB

Fleisch
8 MB CD

■ Primary partition

Right click on **iSCSI Disk → Reactivate Disk**. The disk is now back online:



Computer Management

Volume	Layout	Type	File System	Status
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuration)
iSCSI Disk (S:)	Simple	D...	NTFS	Healthy

Disk 0
Basic
93.16 GB
Online

IBM_PRELOAD (C:)
88.27 GB NTFS
Healthy (System)

IBM_SERVICE
4.89 GB FAT32
Healthy (EISA Configura

Disk 1
Dynamic
949 MB
Online

iSCSI Disk (S:)
949 MB NTFS
Healthy

CD-ROM 0
DVD
8 MB
Online

Fleisch
8 MB CD
Healthy

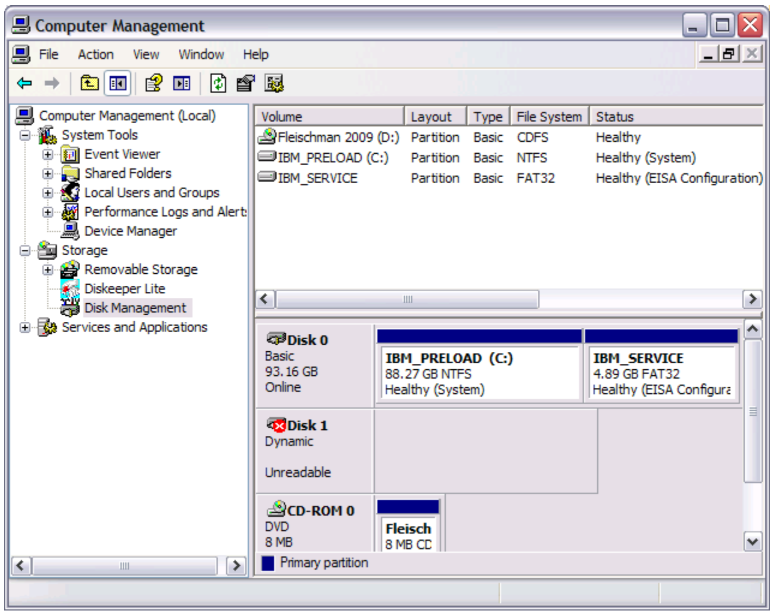
■ Unallocated ■ Primary partition ■ Simple volume

Configuring a Target

Run Computer Management: Click **Start → Run** and type *compmgmt.msc* → click **Ok**. For more information, see http://support.microsoft.com/kb/309000

In the console tree, click **Storage → Disk Management**. The Disk Management window appears. Your disks and volumes appear in a graphical view and list view. To customize how you view your disks and volumes in the upper and lower panes of the window, point to **Top** or **Bottom** on the **View** menu, and then click the view that you want to use.

An "Unreadable" Dynamic disk appears. Right-click on **Disk 1 → Convert to a Basic Disk → Ok**. Optionally, click again to **Convert to a Dynamic Disk**.

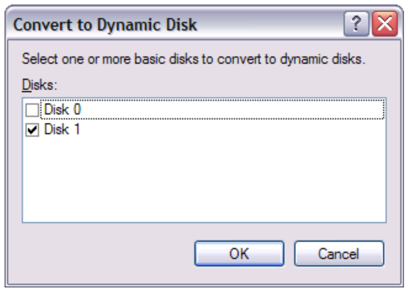


Volume	Layout	Type	File System	Status
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuration)

Disk 0
Basic
93.16 GB
Online
IBM_PRELOAD (C:)
88.27 GB NTFS
Healthy (System)
IBM_SERVICE
4.89 GB FAT32
Healthy (EISA Configura

Disk 1
Dynamic
Unreadable

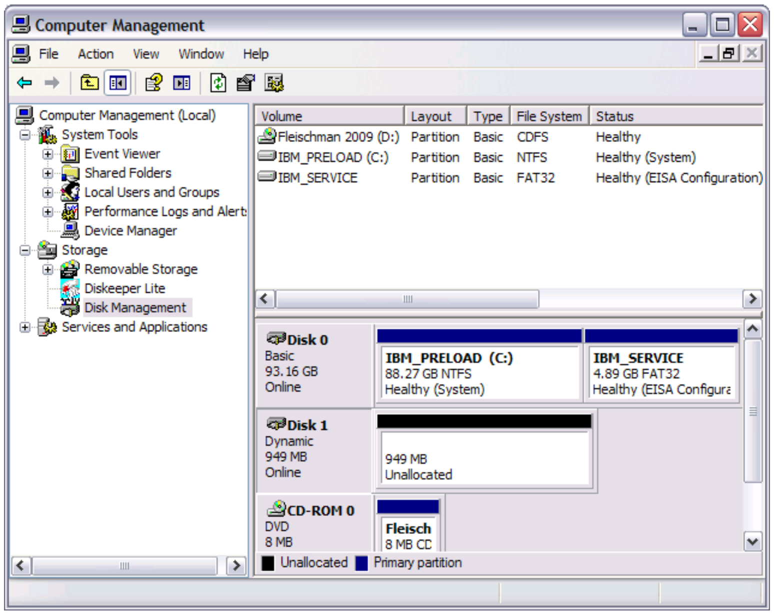
CD-ROM 0
DVD
8 MB
Fleisch
8 MB CD



Select one or more basic disks to convert to dynamic disks.

Disks:
☐ Disk 0
☑ Disk 1

Note that a *dynamic disk* (a disk initialized for dynamic storage) can hold simple volumes, spanned volumes, mirrored volumes, striped volumes, and RAID-5 volumes. With dynamic storage, disk and volume management can be performed without having to restart the operating system. Upgrading a disk to dynamic storage is a one-way process. In order to change back to basic disk format, the drive must be repartitioned, see http://support.microsoft.com/kb/175761/.



Volume	Layout	Type	File System	Status
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuration)

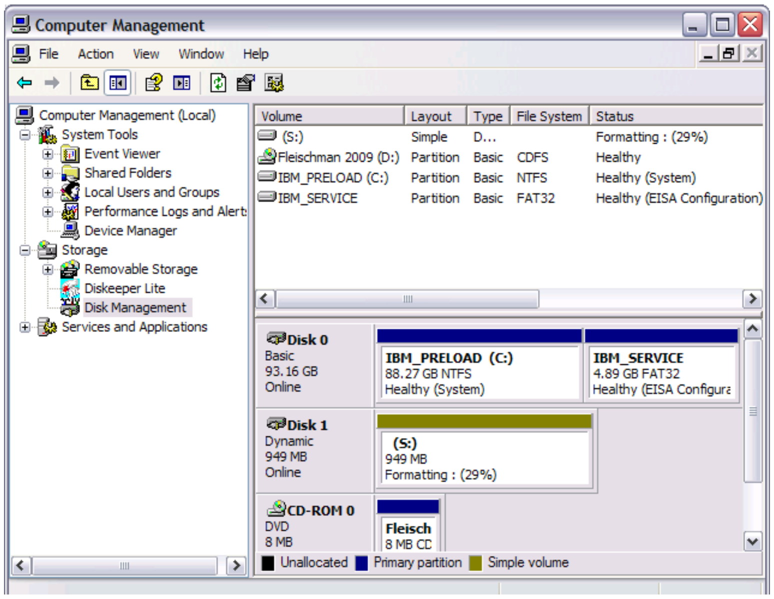
Disk 0
Basic
93.16 GB
Online
IBM_PRELOAD (C:)
88.27 GB NTFS
Healthy (System)
IBM_SERVICE
4.89 GB FAT32
Healthy (EISA Configura

Disk 1
Dynamic
949 MB
Online
949 MB
Unallocated

CD-ROM 0
DVD
8 MB
Fleisch
8 MB CD

Right-click on **Disk 1 → New Volume**. The New Volume Wizard appears. Work through the ensuing popup menus, select your appropriate parameters, and click **Finish**. Windows now formats your iSCSI drive and assigns a drive letter to it.

If necessary, run **Actions → Rescan** to make the Microsoft SCSI layer look at the write protect mode page again.



Volume	Layout	Type	File System	Status
(S:)	Simple	D...		Formatting : (29%)
Fleischman 2009 (D:)	Partition	Basic	CDFS	Healthy
IBM_PRELOAD (C:)	Partition	Basic	NTFS	Healthy (System)
IBM_SERVICE	Partition	Basic	FAT32	Healthy (EISA Configuration)

Disk 0
Basic
93.16 GB
Online

IBM_PRELOAD (C:)
88.27 GB NTFS
Healthy (System)

IBM_SERVICE
4.89 GB FAT32
Healthy (EISA Configura

Disk 1
Dynamic
949 MB
Online

(S:)
949 MB
Formatting : (29%)

CD-ROM 0
DVD
8 MB

Fleisch
8 MB CD

■ Unallocated 🟦 Primary partition 🟨 Simple volume

3. RisingTide Snapshots

The RisingTide IP Storage software stack currently only supports LVM-based snapshots, so for enabling snapshots on a Storage Object, it needs to be on a logical volume (created by the LVM).

Initializing, Starting, Stopping and Status

Initialize snapshots

tcm_node --lvsnapinit <HBA>/<LVM StorageObjectName> <Count#> <Size#> <Interval#>

Example: Initialize LVM snapshot on the logical volume my_lv0 with the following attributes:

- Preserve up to 10 round-robin snapshots
- Reserve 100MB of storage space for the 10 snapshots (valid units are: K, M, G)
- Create a new snapshot every minute (valid units are: [S]econds, [M]inutes, [H]ours)

```
vsnapinit iblock_0/my_lv0 10 100M 1M
Set attribute: lv_group=vg0
Set attribute: lvc_size=100M
Set attribute: max_snapshots=10
Set attribute: contact=root@localhost
Set attribute: check_interval=60
Set attribute: create_interval=60
Set attribute: usage_warn=75
Set attribute: vgs_usage_warn=80
Successfully set default snapshot attributes for
/sys/kernel/config/target/core/iblock_0/my_lv0
You can futher customize these settings with tcm_node --lvsnapattrset and --
lvsnapattrshow
```

Snapshots are now initialized but not yet activated – which requires to start the snapshot demon.

Activate snapshots

tcm_node --lvsnapstart <HBA>/<LVM StorageObjectName>

Example: Activate snapshots on the logical volume my_lv0:

```
# tcm_node --lvsnapstart iblock_0/my_lv0
Started tcm_snap daemon at pid: 8551 for
/sys/kernel/config/target/core/iblock_0/my_lv0
[root@northdome ~]#
  Logical volume "snap-2009-08-10-18_47_16.624103" created
  Logical volume "snap-2009-08-10-18_48_16.934139" created
  Logical volume "snap-2009-08-10-18_49_17.311924" created
```

Disable snapshots

This just stops the snapshot demon.

tcm_node --lvsnapstop <HBA>/<LVM StorageObjectName>

Example: Disable snapshots on the logical volume my_lv0:

```
# tcm_node --lvsnapstop iblock_0/my_lv0
Successfully stopped tcm_snap daemon for
/sys/kernel/config/target/core/iblock_0/my_lv0
```

This actually kills the snapshot demon; however, the snapshot demon is stateless, so it can be restarted transparently for snapshots to resume seamlessly.

Display status information

tcm_node --lvsnapstat <HBA>/<LVM StorageObjectName>

Example: Display the status of snapshots on the logical volume my_lv0:

```
# tcm_node --lvsnapstat iblock_0/my_lv0
  LV VG Attr LSize Origin Snap% Move Log Copy%
Origin LV: lv0 vg0 owi-ao 100.00M
  snap-2009-08-10-18_47_16.624103 vg0 sri-a- 100.00M lv0 0.01
  snap-2009-08-10-18_48_16.934139 vg0 sri-a- 100.00M lv0 0.01
  snap-2009-08-10-18_49_17.311924 vg0 sri-a- 100.00M lv0 0.01
  snap-2009-08-10-18_50_17.745782 vg0 sri-a- 100.00M lv0 0.01
  snap-2009-08-10-18_51_18.333669 vg0 sri-a- 100.00M lv0 0.01
  snap-2009-08-10-18_52_18.850523 vg0 sri-a- 100.00M lv0 0.01
  snap-2009-08-10-18_53_19.395417 vg0 sri-a- 100.00M lv0 0.01
Total snapshot usage percentage for LV: 0.0
Total volume group usage percentage: 1.0
```

Attributes

Display snapshot attributes

tcm_node --lvsnapattrshow <HBA>/<LVM StorageObjectName>

Example: Show the snapshots attributes of the logical volume my_lv0:

```
# tcm_node --lvsnapattrshow iblock_0/my_lv0
vgs_usage_warn=80
usage_warn=75
usage=0
create_interval=60
check_interval=60
max_warn=0
max_snapshots=10
permissions=0
enabled=1
pid=9796
lvc_size=100M
lv_group=vg0
contact=root@localhost
```

The attributes are specifically:

- *vgs_usage_warn* determines the high-watermark of the utilization of the volume group that contains the original volume and snapshots that will lead to an email being sent the *contact*
- *usage_warn* determines the high-watermark of the utilization of the all active snapshots on the associated with the particular Storage Object that will lead to an email being sent the *contact*
- *create_interval* determines how often a new snapshot is created
- *check_interval* determines the wakeup interval for the snapshot housekeeping demon
- *max_warn* is not implemented yet
- *max_snapshots* is the maximum number of snapshots kept before replacing them
- *permissions* is a readonly flag for the volume
- *enabled* determines whether snapshots are enabled
- *pid* depicts the process ID of the snapshot demon
- *lvc_size* is the size of the snapshot volume
- *lvc_group* is the name of the volume group from which the snapshots are being crated
- *contact* is the email address for the administrative contact

Set a snapshot attribute

tcm_node --lvsnapattrset <HBA>/<LVM StorageObjectName> <AttrName>=<AttrValue>

Example: Set the snapshot demon wakeup interval to 30 seconds:

```
# tcm_node --lvsnapattrset iblock_0/my_lv0 check_interval=30
Successfully updated snapshot attribute: check_interval=30 for
/sys/kernel/config/target/core/iblock_0/my_lv0
```

4. Glossary

Backstore: A physical storage object that provides the actual storage underlying an iSCSI Endpoint.

CDB (Command Descriptor Block): The standard format for SCSI commands. CDBs are commonly 6, 10, or 12 bytes long, though they can be 16 bytes or of variable length.

CHAP (Challenge Handshake Authentication Protocol): An authentication technique for confirming the identity of one computer to another. Described in RFC 1994.

CID (Connection Identifier): A 16-bit number, generated by the Initiator, that uniquely identifies a connection between two iSCSI devices. This number is presented during the login phase.

Endpoint: The combination of an iSCSI Target Name with an iSCSI TPG (IQN + Tag).

EUI (Extended Unique Identifier): A 64-bit number that uniquely identifies every device in the world. The format consists of 24 bits that are unique to a given company, and 40 bits assigned by the company to each device it builds.

Initiator: The originating end of a SCSI session. Typically a controlling device such as a computer.

IPS (Internet Protocol Storage): The class of protocols or devices that use the IP protocol to move data in a storage network. FCIP, iFCP, and iSCSI are all examples of IPS protocols.

IQN (iSCSI Qualified Name): A name format for iSCSI that uniquely identifies every device in the world (e.g. iqn.5886.com.acme.tapedrive.sn-a12345678).

ISID (Initiator Session Identifier): A 48-bit number, generated by the Initiator, that uniquely identifies a session between the Initiator and the Target. This value is created during the login process, and is sent to the target with a Login PDU.

MCS (Multiple Connections per Session): A part of the iSCSI specification that allows multiple TCP/IP connections between an Initiator and a Target.

MPIO (MultiPah I/O): A method by which data can take multiple redundant paths between a server and storage.

Network Portal: The combination of an iSCSI Endpoint with an IP address plus a TCP port. The TCP port number for the iSCSI protocol defined by IANA is 3260.

SAM (SCSI Architectural Model): A document that describes the behavior of SCSI in general terms, allowing for different types of devices communicating over various media.

Target: The receiving end of a SCSI session, typically a device such as a disk drive, tape drive, or scanner.

Target Group: A list of SCSI Target Ports that are all treated the same when creating views. Creating a view can help facilitate LUN mapping. Each view entry specifies a target group, host group, and a LUN.

Target Port: The combination of an iSCSI Endpoint with one or more LUNs.

TPG (Target Portal Group): A list of IP addresses and TCP port numbers that determines which interfaces a specific iSCSI target will listen to.

TSID (Target Session Identifier): A 16-bit number, generated by the target, that uniquely identifies a session between the initiator and the target. This value is created during the login process, and is sent to the initiator with a Login Response PDU.