NetApp® Block Access Management using Open Interfaces

Network Appliance, Inc.

March 2010

Executive Summary

NetApp storage systems support different block access protocols like iSCSI, FCP and block access entities like LUNs, initiator groups, and portsets. NetApp provides open interfaces like Data ONTAP APIs, SNMP, SMI-S agent for monitoring and managing various components of the NetApp storage system. This document provides the details of how to use NetApp open interfaces for managing the block access protocols and block access entities of NetApp storage systems.

Table of Contents

2 NetApp Block Access Concepts

2.1 iSCSI

iSCSI is an industry standard storage protocol defined and maintained by the Internet Engineering Task Force. Also known as Internet SCSI, iSCSI encapsulates SCSI block storage commands into Ethernet packets for transport over IP networks, enabling companies to leverage standard, familiar Ethernet networking infrastructures to create affordable SANs.

In an iSCSI network, there are two types of nodes: targets and initiators. Targets are storage systems, and initiators are hosts. Switches, routers, and ports are TCP/IP devices only and are not iSCSI nodes. Storage systems and hosts can be direct-attached or they can be connected through Ethernet switches. Both direct-attached and switched configurations use Ethernet cable and a TCP/IP network for connectivity.

The iSCSI protocol is a licensed service on the NetApp storage system that enables you to transfer block data to hosts using the SCSI protocol over TCP/IP. The iSCSI protocol is implemented in two ways. One approach is to implement Iscsi over the storage system's standard gigabit Ethernet interfaces using a software driver. Another approach is to use iSCSI target HBA to which some of the iSCSI protocol processing is offloaded. NetApp storage system listens for iSCSI connections on TCP port 3260.

2.2 FCP

Fibre Channel is a gigabit-speed network technology primarily used for Storage Area Networks. Fibre Channel Protocol (FCP) is the interface protocol of SCSI on the Fibre Channel. FCP maps the SCSI commands to the Fibre Channel transport layer.

In an FCP network, nodes include targets, initiators, and switches. Targets are storage systems, and initiators are hosts. Each FCP node is identified by a worldwide node name (WWNN). The FCP service is implemented over the target's and initiator's FCP ports. Each FCP port is identified by a worldwide port name (WWPN). Both the WWNN and the WWPN are a 64-bit address represented in the following format: nn:nn:nn:nn:nn:nn:nn:nn, where n represents a hexadecimal value. Storage systems and hosts have Host Bus Adapters (HBAs) so they can be connected directly to each other or to Fibre Channel switches with optical cable.

FCP is a licensed service on NetApp storage system that enables you to transfer block data to hosts using the SCSI protocol over a Fibre Channel fabric.

2.3 LUNs

In FCP and iSCSI networks, storage systems are targets that have storage target devices, which are referred to as LUNs (logical units).

From the storage system, a LUN is a logical representation of a physical unit of storage. It is a collection of, or a part of, physical or virtual disks configured as a single disk. When you create a LUN, it is automatically striped across many physical disks. From the host, LUNs appear as local disks on the host that you can format and manage to store data.

Data ONTAP is the operating system for the NetApp storage system. Using the Data ONTAP® operating system, you can configure the storage for Block access by creating LUNs. Data ONTAP manages LUNs at the block level, so it cannot interpret the file system or the data in a LUN. LUNs are identified by LUN serial number which is a unique 12-byte, ASCII string generated by the storage system.

You create LUNs in the root of a volume (traditional or flexible) or in the root of a qtree, with the exception of the root volume. Do not create LUNs in the root volume because it is used by Data ONTAP for system administration. The default root volume is /vol/vol0.

2.4 igroups

Initiator groups (igroups) are tables of host identifiers (FCP WWPNs or iSCSI node names) that are used to control hosts' access to LUNs.

igroups specify which initiators have access to which LUNs. igroups can be created either before or after LUNs are created, but they must be created before a LUN is mapped to an igroup. Initiator groups can have multiple initiators, and multiple igroups can have the same initiator. However, a LUN can not be mapped to multiple igroups that have the same initiator.

An initiator cannot be a member of igroups of differing ostypes. Also, a given igroup can be used for FCP or iSCSI, but not both.

2.5 Portsets

A portset consists of a group of target ports. You bind a portset to an igroup, to make the LUNs available only on a subset of the storage system's target ports. Any host in the igroup can access the LUNs only by connecting to the target ports in the portset.

If an igroup is not bound to a portset, the LUNs mapped to the igroup are available on all of the storage system's target ports. By using portsets, you can selectively control which initiators can access LUNs and the ports on which they access LUNs.

Portsets can only be used for LUNs that are accessed by FCP hosts. Portsets cannot be used for LUNs accessed by iSCSI hosts.

3.3 Data ONTAP SMI-S Agent

Data ONTAP SMI-S agent provides standards based storage management interface to discover, monitor, and manage NetApp storage systems. The specifications for SMI-S are developed by SNIA (Storage Networking Industry Association) and DMTF (Distributed Management Task Force) standards organizations.

The SMI-S agent is implemented as a proxy-based management solution. The agent needs to be installed on an external server (i.e. it is not implemented within ONTAP). Currently the supported platforms for SMI-S agent installation are Windows and Linux (Red Hat and SUSE) based hosts.

4 iSCSI Management using NetApp Open Interfaces

4.1 Managing iSCSI License

The following table provides details on how to verify / enable iSCSI licenses on the NetApp Storage.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Verify License	API	*License-list-info*	API returns *license-info[]* array. The *service* filed in the *license-info* object provides the service name. For iSCSI the value will be "*iscsi"*. The *' is-licensed'* field will be set to *true* if the license is enabled.
	SNMP	*iscsiIsLicensed (1.3.6.1.4.1.789.1.17.2)*	The return value of *true (2)* means iSCSI is enabled
Enable License	API	*license-add* 'code'	'code' is the license string
	SNMP	*Not Available*	

4.2 Managing iSCSI Service

The following table provides details on how to verify / start / stop the iSCSI service on the NetApp Storage

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Verify iSCSI service status	API	*iscsi-service-status*	The output parameter *is-available* is *true* if the service is running else it is *false*
	SNMP	*Not Available*	
Start iSCSI service	API	*iscsi-service-start*	
	SNMP	*Not Available*	
Stop iSCSI service	API	*iscsi-service-stop*	
	SNMP	*Not Available*	

4.3 Managing iSCSI target node name

The following table provides details on how to view / change the target node name of the NetApp Storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
View the node name	API	*iscsi-node-get-name*	
	SNMP	*Not Available*	
Change the node name	API	*iscsi-node-set-name*	
	SNMP	*Not Available*	

4.3.1 Changing the iSCSI Target Node Name

Changing the storage system's node name while iSCSI sessions are in progress does not disrupt the existing sessions. However, when you change the storage system's node name, you must reconfigure the initiator so that it recognizes the new target node name. If you do not reconfigure the initiator, subsequent initiator attempts to log in to the target will fail.

Node name rules: If you change the storage system's target node name, be sure the new name follows all of these rules:

A node name can be up to 223 bytes.

Uppercase characters are always mapped to lowercase characters.

A node name can contain alphabetic characters (a to z), numbers (0 to 9) and three special characters:

- Period (".")
- Hyphen ("-")
- Colon (":")

The underscore character ("_") is *not* supported.

4.4 Managing target alias

The target alias is an optional human-readable name for the iSCSI target. For example, if the target node name was iqn.1992-08.com.netapp:sn.33604646, the user might want an alias such as Filer_1. The alias is a text string with a maximum of 128 characters.

The alias is intended to be displayed by an initiator's user interface to make it easier for someone to identify the desired target in a list of targets. Depending on the initiator, the alias may or may not be displayed in the initiator's user interface.

The user can change the target alias or clear the alias at any time without disrupting existing sessions. The new alias will be sent to the initiators the next time they log in to the target.

The following table provides details on how to view / add / delete the target alias for the NetApp Storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Viewing the target aliases	API	*iscsi-target-alias-get-alias*	
	SNMP	*Not Available*	
Adding the target alias	API	*iscsi-target-alias-set-alias*	
	SNMP	*Not Available*	
Deleting the target alias	API	*iscsi-target-alias-clear-alias*	
	SNMP	*Not Available*	

4.5 iSNS Server Configuration

An iSNS server uses the Internet Storage Name Service (iSNS) protocol to maintain information about active iSCSI devices on the network, including their IP addresses, iSCSI node names, and portal groups. The iSNS protocol enables automated discovery and management of iSCSI devices on an IP storage network. An iSCSI initiator can query the iSNS server to discover iSCSI target devices.

The storage system automatically registers its IP address, node name, and portal groups with the iSNS server when the iSCSI service is started and iSNS is enabled. After iSNS is initially configured, Data ONTAP automatically updates the iSNS server any time the storage system's configuration settings change. There can be a delay of a few minutes between the time of the configuration change and the update being sent; you can use the open interfaces to send an update immediately.

Note: *NetApp does not supply or resell iSNS servers. You obtain these servers from a vendor supported by NetApp. Be sure to check the NetApp iSCSI Support Matrix to see which iSNS servers are currently supported.*

The following table provides details on how to use open interfaces to configure iSNS on NetApp storage systems:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Configuring the iSNS Service	API	*iscsi-isns-config*	API takes *ipAddress* of the iSNS Server as input
	SNMP	*Not Available*	
Starting the iSNS	API	*iscsi-isns-start*	Start the iSNS Service on the NetApp storage system. Registers the device

Service			with the iSNS server as specified by iscsi-isns-config command.
	SNMP	*Not Available*	
Stopping the iSNS Service	API	*iscsi-isns-stop*	
	SNMP	*Not Available*	
Force iSNS Update	API	*iscsi-isns-update*	
	SNMP	*Not Available*	
Retrieve iSNS Server information	API	*iscsi-isns-get-info*	*Isns-ip-addr* output parameter provides the iSNS server ipaddress
	SNMP	*Not Available*	

4.5.1 Resolving iSNS Version incompatibility

The specification for the iSNS service is still in draft form. Some draft versions are different enough to prevent the storage system from registering with the iSNS server. Because the protocol does not provide version information to the draft level, iSNS servers and storage systems cannot negotiate the draft level being used.

By default, Data ONTAP versions prior to 7.1 used iSNS draft 18. In Data ONTAP 7.1, the default iSNS version is draft 22.

The following table provides the details on how to view / change the iSNS version information on NetApp Storage systems:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Viewing iSNS Version details	API	*options_get*	API to get the options of NetApp systems. The iSNS version option is *iscsi.isns.rev.*
	SNMP	*Not Available*	
Setting the iSNS version	API	*options_set*	API to set the options on NetApp systems. The iSNS version option is *iscsi.isns.rev.* The value can be 18 or 22 corresponding to iSNS version drafts 18 and 22.
	SNMP	*Not Available*	

4.6 Managing iSCSI service on Ethernet Interfaces

The iSCSI protocol on the storage system can be implemented over the storage system's standard Ethernet interfaces using software that is integrated into Data ONTAP. iSCSI can be implemented over multiple storage system Ethernet interfaces. The user can control which network interfaces are used for iSCSI communication. For example, user

can enable iSCSI communication over specific gigabit Ethernet (GbE) interfaces. By default, the iSCSI service is enabled on all Ethernet interfaces after enabling the license.

The following table provides details on how to manage Ethernet interfaces on NetApp storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
List iSCSI details for each interface	API	*iscsi-interface-list-info*	
	SNMP	*Not Available*	
Enable interface for iSCSI	API	*iscsi-interface-enable*	
	SNMP	*Not Available*	
Disable interface for iSCSI	API	*iscsi-interface-disable*	
	SNMP	*Not Available*	

4.7 Managing iSCSI Initiator Security

4.7.1 Authentication Basics

During the initial stage of an iSCSI session, the initiator sends a login request to the storage system to begin an iSCSI session. The storage system permits or denies the login request according to one of the following authentication methods:

Challenge Handshake Authentication Protocol (CHAP)—The initiator logs in using a CHAP user name and password. The user can specify a CHAP password or generate a random password. There are two types of CHAP user names and passwords:

- Inbound—The storage system authenticates the initiator. Inbound settings are required if CHAP authentication is used.
- Outbound—This is an optional setting to enable the initiator to authenticate the storage system. The outbound settings can be used only if the user has defined an inbound user name and password on the storage system.

deny—The initiator is denied access to the storage system.

none—The storage system does not require authentication for the initiator.

The user can define a list of initiators and their authentication methods. You can also define a default authentication method for initiators that are not on this list. If no list of initiators and authentication methods, the default method is none—any initiator can access the storage system without authentication.

4.7.2 Guidelines for CHAP Authentication

The following guidelines apply to CHAP authentication:

If you define an inbound user name and password on the storage system, you must use the same user name and password for outbound CHAP settings on the initiator.

If you also define an outbound user name and password on the storage system to enable bidirectional authentication, you must use the same user name and password for inbound CHAP settings on the initiator.

You cannot use the same user name and password for inbound and outbound settings on the storage system.

CHAP user names can be 1 to 128 bytes. A null user name is not allowed.

CHAP passwords (secrets) can be 1 to 512 bytes. Passwords can be hexadecimal values or strings. For hexadecimal values, enter the value with a prefix of "0x" or "0X". A null password is not allowed.

See the initiator's documentation for additional restrictions. For example, the Microsoft iSCSI software initiator requires both the initiator and target CHAP passwords to be at least 12 bytes if IPsec encryption is not being used. The maximum password length is 16 bytes regardless of whether IPsec is used.

4.7.3 Open Interfaces

The following provide the details of various initiator operations on NetApp storage systems:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Get default iSCSI initiator authentication	API	*iscsi-initiator-get-default-auth*	API will return whether default authentication for iSCSI initiators is CHAP, deny or none.
	SNMP	*Not Available*	
Set default iSCSI initiator authentication	API	*iscsi-initiator-set-default-auth*	API to set default iSCSI initiator authentication to CHAP, deny or none.
	SNMP	*Not Available*	
Get authentication list for all iSCSI initiators	API	*iscsi-initiator-auth-list-info*	If a initiator name is specified as an input then the API returns the details for the particular initiator. In this case the API is similar to *iscsi-initiator-get-auth*.
	SNMP	*Not Available*	
Get authentication details for a initiator	API	*iscsi-initiator-get-auth*	
	SNMP	*Not Available*	
Add initiator to	API	*iscsi-initiator-add-auth*	API to set the authentication mode

authentication list			for the initiator and add it to authentication list
	SNMP	*Not Available*	
Modify CHAP Params for existing initiator	API	*iscsi-initiator-modify-chap-params*	Modify CHAP parameters to an existing per-initiator authentication info whose auth-type equals CHAP
	SNMP	*Not Available*	
Remove initiator from authentication list	API	*iscsi-initiator-delete-auth*	API to remove the initiator from authentication list. In this case the authentication mode applied to initiator will be the *default* mode.
	SNMP	*Not Available*	
Generate random CHAP password	API	*iscsi-auth-generate-chap-password*	
	SNMP	*Not Available*	

4.8 Managing iSCSI target portal groups

A target portal group is a set of network portals within an iSCSI node over which an iSCSI session is conducted. In a target, a network portal is identified by its IP address and listening TCP port. For storage systems, each network interface can have one or more IP addresses and therefore one or more network portals. A network interface can be an Ethernet port, virtual local area network (VLAN), or
virtual interface (vif).

The assignment of target portals to portal groups is important for two reasons:

- The iSCSI protocol allows only one session between a specific iSCSI initiator port and a single portal group on the target.
- All connections within an iSCSI session must use target portals that belong to the same portal group.

By default, Data ONTAP maps each Ethernet interface on the storage system to its own default portal group. You can create new portal groups that contain multiple interfaces.

You can have only one session between an initiator and target using a given portal group. To support some multipath I/O (MPIO) solutions, you need to have separate portal groups for each path. Other initiators, including the Microsoft iSCSI initiator version 2.0, support MPIO to a single target portal group by using different initiator session IDs (ISIDs) with a single initiator node name.

The following table provides details on how to manage target portal groups on NetApp storage systems:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
List all	API	*iscsi-tpgroup-list-entry-info*	

target portal groups	SNMP	*Not Available*	
Create a target portal group	API	*iscsi-tpgroup-create*	
	SNMP	*Not Available*	
Destroy a target portal group	API	*iscsi-tpgroup-destroy*	
	SNMP	*Not Available*	
Add interface to portal group	API	*iscsi-tpgroup-interface-add*	
	SNMP	*Not Available*	
Remove interface from portal group	API	*iscsi-tpgroup-interface-delete*	
	SNMP	*Not Available*	
Change ALUA parameters on target portal group	API	*iscsi-tpgroup-alua-set*	Data ONTAP supports SCSI ALUA functionality for managing multi-pathed SCSI devices
	SNMP	*Not Available*	

4.9 Managing iSCSI Sessions

4.9.1 Displaying iSCSI initiators, connections and sessions

The user can retrieve information on iSCSI Initiators connected to the NetApp storage system. An iSCSI session can have zero or more connections. Typically a session has at least one connection. Connections can be added and removed during the life of the iSCSI session.

The following APIs provide the details:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Retrieving connected iSCSI Initiators	API	*iscsi-initiators-list-info*	List of initiators that are connected to the filer. The information displayed for each initiator includes the target session identifier handle (TSIH) assigned to the session, the target portal group tag of the group the initiator is connected to, the iSCSI initiator alias (if provided by the initiator), and the initiator's iSCSI node name and initiator session

			identifier (ISID).
	SNMP	*initiatorListTable (1.3.6.1.4.1.789.1.17.16.3)*	
Retrieving iSCSI connections	API	*iscsi-connection-list-info*	List iSCSI connections on filer. It provide TCP IP details for each connection
	SNMP	*iscsiConnectionAttributesTable (1.3.6.1.3.9999.1.12.1.1)*	
Retrieving iSCSI Session details	API	*iscsi-session-list-info*	List of active iSCSI Sessions on a storage system. The iSCSI Session provide details on iSCSI session parameters, connections associated with each session, active iSCSI commands per active connection. In order to associate initiator with a session, use ISID and TSIH provided in both session list and initiator list.
	SNMP	*iscsiSessionAttributesTable (1.3.6.1.3.9999.1.11.1)*	

4.10 iSCSI Statistics

The following table provides details on how to retrieve iSCSI statistics from NetApp Storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Retrieve iSCSI Statistics	API	*iscsi-stats-list-info*	API to provide iSCSI statistics. Returns statistics in 4 categories: ***iscsi-cdb-stats*:** Statistics of Command Descriptor Blocks ***iscsi-error-stats*:** Statistics of errors. ***iscsi-received-stats*:** Statistics of PDUs received. ***iscsi-transmitted-stats*:** Statistics of PDUs transmitted.
	SNMP	*Not Available*	
Reset iSCSI Statistics	API	*iscsi-reset-stats*	
	SNMP	*Not Available*	

4.10.1.1 Interpreting iSCSI Statistics

iSCSI PDUs received: This section lists the iSCSI Protocol Data Units (PDUs) sent by the initiator. It includes the following statistics:

SCSI-CMD—SCSI-level command descriptor blocks.

LoginReq—Login request PDUs sent by initiators during session setup.

DataOut—PDUs containing write operation data that did not fit within the PDU of the SCSI command. The PDU maximum size is set by the storage system during the operation negotiation phase of the iSCSI login sequence.

Nop-Out—A message sent by initiators to check whether the target is still responding.

Logout-Req—A request sent by initiators to terminate active iSCSI sessions or to terminate one connection of a multi-connection session.

SNACK—A PDU sent by the initiator to acknowledge receipt of a set of DATA_IN PDUs or to request retransmission of specific PDUs.

SCSI TaskMgtCmd—SCSI-level task management messages, such as ABORT_TASK and RESET_LUN.

Text-Req—Text request PDUs that initiators send to request target information and renegotiate session parameters.

iSCSI PDUs transmitted: This section lists the iSCSI PDUs sent by the storage system and includes the following statistics:

SCSI-Rsp—SCSI response messages.

LoginRsp—Responses to login requests during session setup.

DataIn—Messages containing data requested by SCSI read operations.

Nop-In—Responses to initiator Nop-Out messages.

Logout-Rsp—Responses to Logout-Req messages.

R2T—Ready to transfer messages indicating that the target is ready to receive data during a SCSI write operation.

SCSI TaskMgtRsp—Responses to task management requests.

TextRsp—Responses to Text-Req messages.

Asyncmsg—Messages the target sends to asynchronously notify the initiator of an event, such as the termination of a session.

Reject—Messages the target sends to report an error condition to the initiator, for example:

- Data Digest Error (checksum failed)
- Target does not support command sent by the initiator
- Initiator sent a command PDU with an invalid PDU field

iSCSI CDBs: Provide statistics associated with the handling of iSCSI Command Descriptor Blocks, including the number of blocks of data transferred, and the number of SCSI-level errors and successful completions.

iSCSI Errors: Provide details of failures and other SCSI protocol errors.

5 FCP Management using NetApp Open Interfaces

5.1 Managing FCP License

The following table provides details on how to verify / enable FCP licenses on the NetApp Storage.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Verify License	API	*license-list-info*	API returns *license-info[]* array. The *service* filed in the *license-info* object provides the service name. For FCP the value will be "*fcp"*. The *' is-licensed'* field will be set to *true* if the license is enabled.
	SNMP	*fcpIsLicensed (1.3.6.1.4.1.789.1.17.1)*	The return value of *true (2)* means FCP is enabled
Enable License	API	*license-add* 'code'	'code' is the license string
	SNMP	*Not Available*	

5.2 Managing FCP Service

The following table provides details on how to verify / start / stop the FCP service on the NetApp Storage

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Verify fcp service status	API	*fcp-service-status*	The output parameter *is-available* is *true* if the service is running else it is *false*
	SNMP	*Not Available*	
Start fcp service	API	*fcp-service-start*	Adapters are brought online when the service starts
	SNMP	*Not Available*	
Stop fcp service	API	*fcp-service-stop*	Adapters are brought offline when the service stops
	SNMP	*Not Available*	

5.3 Managing FCP node name

Each FCP node is identified by a worldwide node name (WWNN).The WWNN of a storage system is generated by a serial number in its NVRAM, but it is stored on disk. If you ever replace a storage system chassis and reuse it in the same Fibre Channel SAN, it is possible, although extremely rare, that the WWNN of the replaced storage system is duplicated. In this unlikely event, you can change the WWNN of the storage system.

The following table provides details on how to view / change the FCP node name of the NetApp Storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
View the node name	API	*fcp-node-get-name*	The nodename is in the form of a Fibre Channel world wide name, which is in the form XX:XX:XX:XX:XX:XX:XX:XX where X is a hexadecimal digit.
	SNMP	*fcpTargetNN* (1.3.6.1.4.1.789.1.17.17.1.1.3)	
Change the node name	API	*fcp-node-set-name*	<node-name> input is in the form of in the form XX:XX:XX:XX:XX:XX:XX:XX where X is a hexadecimal digit.
	SNMP	*Not Available*	

5.4 Managing FC Adapter

Depending on the specific model, NetApp storage system can have two types of target ports:

Host Bus Adapter (HBA) ports: The storage system has a target FCP HBA with two ports that are labeled Port A and Port B (if there is a second port). FAS900 series systems use target HBAs.

Onboard ports: The following systems have onboard FCP adapters, or ports, that you can configure to connect to disk shelves or to operate in SAN target mode:

1. FAS270 models—These storage systems have a port labeled Fibre Channel C (with an orange label). You can configure the Fibre Channel C port in initiator or target mode. You use initiator mode to connect to tape backup devices such as in

a TapeSAN backup configuration. You use target mode to communicate with SAN hosts or a front end SAN switch.

2. FAS3000 models—These storage systems have four onboard Fibre Channel ports that have orange labels and are numbered 0a, 0b, 0c, and 0d. You use the *fcadmin* command to configure the ports to operate in SAN target mode or initiator mode. In SAN target mode, the onboard ports connect to Fibre Channel switches or fabric. In initiator mode, they connect to disk shelves.
3. FAS6000 models—These storage systems have eight onboard Fibre Channel ports that have black and gray labels and are numbered 0a, 0b, 0c, 0d, 0e, 0f, 0g, and 0h. You use the *fcadmin* command to configure the ports to operate in SAN target mode or initiator mode. In SAN target mode, the onboard ports connect to Fibre Channel switches or fabric. In initiator mode, they connect to disk shelves.

5.4.1 Managing FC Target Adapter Configuration

The following table provides details on how to view / configure the target FCP Adapters of NetApp Storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
List the details of all FC HBAs	API	*fcp-adapter-list-info*	The output is an array of adapter configuration information *fcp-config-adapter-info[]*
	SNMP	`fcpTargetTable` *(1.3.6.1.4.1.789.1.17.17.1)*	
Bring the adapter online	API	*fcp-adapter-config-up*	Adapter can not be brought online if the FCP service is not started
	SNMP	*Not Available*	
Bring the adapter offline	API	*fcp-adapter-config-down*	
	SNMP	*Not Available*	
Sets the link type on the Fibre Channel target adapter	API	*fcp-adapter-config-media-type*	It can be configured to establish a point-to-point link, a loop configuration or to automatically sense whether the connection type is a point-to-point or loop link. If the adapter is online, it must be brought offline and then online in order for a new mediatype to take effect.
	SNMP	*Not Available*	
Get the list of initiators	API	*fcp-adapter-initiators-list-info*	
	SNMP	*Not Available*	

currently connected to the specified Fibre Channel target adapter			
Set the speed on the Fibre Channel target adapter	API	*fcp-adapter-set-speed*	FC Adapter can be configured to run at 1Gb, 2Gb, 4Gb, or to autonegotiate. If the adapter is online it must be brought offline before setting the speed, and then online in order for a new speed to take effect.
	SNMP	*Not Available*	

5.4.2 Managing FC Adapter Functionalities

Data ONTAP supports three separate functions with Fibre channel controllers:

Initiator mode for FC storage loops, FC TapeSAN, and OPR functionality
Target mode for SAN attach (FCP/Blocks access) functionality
FC-VI mode for MetroCluster/SlideRuler interconnect functionality

These three functions are controlled by separate drivers. As DOT boots, the device driver attached to each FC adapter determines which function an adapter will support.

The fc-config APIs provide a user programmable mechanism to configure the personality of Fibre channel adapters by changing what driver attached to the adapter. Not all Fibre channel adapters support FC configuration. Currently only "embedded" Fibre channel adapters can be programmed with the fc-config APIs.

Following table shows the APIs used to get the list of Adapters and their details:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
View the adapter configuration information	API	*fc-config-list-iter-**	The output is an array of adapter configuration information *fcp-config--info[]*
	SNMP		

Some of the adapter details got from the above APIs:

adapter-name	FC adapter name (e.g. 0c)
adapter-state	Indicates what the adapter configuration state is. Possible values: "UNDEFINED" - The default state. The adapter has never been configured. "CONFIGURED" - The adapter port is configured and the adapter is operational. "UNCONFIGURED" - The adapter is unconfigured. The initiator driver is attached, but the adapter is not operational. "PENDING" - The adapter is waiting for a filer reboot to effect an fc-type change. While in the PENDING state the adapter can not be used
adapter-status	Possible values: "online" - adapter driver is enabled "offline" - adapter driver is disabled
adapter-type	Indicates which driver is attached to the adapter. Possible values: "initiator" - the storage Initiator driver (default) "vi" - the FC-VI cluster interconnect driver "target" - the FCP Target driver
pending-fc-type	If adapter-state is PENDING, this value tells what the last fc-config-set-adapter-fc-type command was.

Following APIs are used to bring the adapter online/offline by calling the enable/disable functions of the Adapter driver:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Bring the adapter online	API	*fc-config-adapter-enable*	Under some circumstances an adapter can not be brought online. (e.g. when that adapter is in the UNCONFIGURED state, or when there is no cable attached to the adapter port). When this happens, an appropriate error messages is returned.
	SNMP	*Not Available*	
Bring the adapter offline	API	*fc-config-adapter-disable*	Under some circumstances an adapter can not be put offline. (e.g. when the adapter is being used by the RAID sub-system to provide disks in a volume). In some cases, manual intervention is required. When this happens, an appropriate error messages is returned.
	SNMP	*Not Available*	

Each configurable adapter has an adapter-type and adapter-state. The adapter-type indicates which driver is attached to the adapter, the adapter-state indicates what the configuration state of the adapter is. The fc-type attribute of the adapter is used to modify both the adapter-type and the adapter-state.

The following table provides the APIs to change both the adapter type and adapter state:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Set the fc-type of the adapter	API	*fc-config-set-adapter-fc-type*	
	SNMP	*Not Available*	

Possible inputs for this API:
"unconfigured" - set adapter-type to "initiator" and adapter-state to UNCONFIGURED
"initiator" - set adapter-type to "initiator" and adapter-state to CONFIGURED
"target" - set adapter-type to "target" and adapter-state to CONFIGURED
"vi" - set adapter-type to "vi" and adapter-state to CONFIGURED

After setting the adapter fc-type a filer reboot is sometimes needed to make the change effective. Use fc-config-list-info API to determine if a filer reboot is needed.

5.4.3 FCP Statistics

The following table provides details on how to retrieve FCP statistics from NetApp Storage:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Retrieve FCP Statistics	API	*fcp-adapter-stats-info*	Returns an array of type fcp-adapter-stats-info[] with each element in the representing the FCP Statistics for each adapter in the storage system
	SNMP		
Reset FCP Statistics	API	*fcp-adapter-reset-stats*	
	SNMP	*Not Available*	

5.4.3.1 Interpreting FCP Statistics

adapter-resets	Number of adapter resets occurred.
crc-errors	Total CRC errors occurred.
frame-overruns	Number of frame overruns detected by the adapter during write requests.
frame-underruns	Number of frame underruns detected by the adapter during read requests.
initiators-connected	Total number of initiators connected to this target adapter.
link-breaks	Number of times that the link breaks.
lip-resets	Number of times that a selective Reset LIP (Loop Initialization Primitive) occurred. LIP reset is used to preform a verndorspecific reset at the loop port specified by the AL-PA value.
queue-depth	Counts the queue depth on the target HBA.
scsi-requests-dropped	Number of SCSI requests being dropped.
spurious-interrupts	Count of spurious interrupts.
total-logins	Counts the times of initiators added. Each time a new initiator is added, the total logins is incremented by 1. Each time an initiator is removed, the total logouts is incremented by 1.
total-logouts	Counts the times of initiators removed. Each time a new initiator is added, the total logins is incremented by 1. Each time an initiator is removed, the total logouts is incremented by 1.

5.5 Managing FCP in Clustered Environment

In a storage cluster environment, Data ONTAP provides multiple modes of operation required to support homogeneous and heterogeneous host operating systems. The FCP cfmode setting controls how the target ports

log into the fabric.

Handle local and partner traffic for a cluster, in normal operation and in takeover.

Provide access to local and partner LUNs in a cluster.

Also this setting controls if the filer should use multiple virtual target adapters per physical target adapter

Following table explains the different cfmode settings, their description and their benefits.

cfmode	Description	Benefits/Limitations
partner	In this mode the storage system connects to the FC fabric in ptp mode. The B-port on each FC target adapter proxies commands for the partner over the cluster interconnect.	Supports all host OS types. Supports all switches. Requires multi-pathing software to be installed on the host.
single_image	In this mode the storage system connects to the FC fabric in ptp mode by default but is configurable, and all luns in the cluster are visible on all FC target ports. In this mode all hosts require multipathing software.	Supports all host OS types. Supports all switches. Makes all LUNs available on all target ports.
dual_fabric	In this mode the storage system connects to the FC fabric in loop mode and all initiators need to have multipathing software	Supports all host OS types. Requires fewer switch ports. Does not support all switches. Requires switches that support public loop.
standby	In this mode the B-port is in standby mode and takes over the partner WWNN and WWPN during a cluster takeover. A ptp connection is used in this case.	Requires more switch ports. Supports only Windows and Solaris hosts.
mixed	In this mode the storage system connects to the FC fabric in loop mode and assigns multiple virtual identities to each physical FC target adapter.	Supports all operating systems. Does not support all switches. Requires switches that support public loop.

Guideline for setting cfmode:

fcp service must be stopped before setting/changing the cfmode.

On filers with only one FCP target port the valid cfmodes are 'dual_fabric', and 'single_image'

On storage systems with more than one FCP target port the valid cfmodes are 'mixed', 'partner', 'standby' and 'single_image'.

When setting the cfmode to 'single_image' configuration checks are performed. If these checks fail an EPERM error will be returned.

The following table provides details on how to get/set FCP cfmode settings:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To get the current cfmode setting for the system	API	*fcp-get-cfmode*	
	SNMP	*fcpCfMode (1.3.6.1.4.1.789.1.17.18.0)*	

To set the current cfmode setting for the system	API	*fcp-set-cfmode*	
	SNMP	*Not Available*	
To set the name of the partner which the local adapter can takeover during CFO	API	*fcp-adapter-set-partner*	
	SNMP	*Not Available*	
To remove the name of the partner which the local adapter should takeover during CFO	API	*fcp-adapter-clear-partner*	
	SNMP	*Not Available*	

6 igroup Management using NetApp Open Interfaces

6.1 Managing igroups

The following table provides details on how to manage the igroups:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Get the details of all the igroups or the details of a given igroup	API	*igroup-list-info*	Returns an array *initiator-group-info[]* which has details of all the igroups or the details of a given igroup
	SNMP	*initiatorGroupTable (1.3.6.1.4.1.789.1.17.16.1)* *initiatorGroupMemberTable (1.3.6.1.4.1.789.1.17.16.2)*	*initiatorGroupTable* returns the igroup information and *initiatorGroupMemberTable* provides details of initiators mapped to a particular igroup
Create a new initiator group	API	*igroup-create*	Valid inputs: Valid os-types are: "solaris", "windows", "hpux", "aix", "linux", "netware" and "vmware". Valid initiator-group-type are: “iscsi” and “fcp”
	SNMP	*Not Available*	
Add an initiator to an existing igroup	API	*igroup-add*	An initiator cannot be a member of igroups of differing types. For example, if you have an initiator that already belongs to a solaris igroup, you cannot add this initiator to an aix igroup. The initiator to be added should the iSCSI node name for ‘‘iscsi’’ initiator-group-type and FCP WWPN for “fcp” initiator-group-type
	SNMP	*Not Available*	
Remove an initiator to an existing igroup	API	*igroup-remove*	The operation is prohibited if there are existing lun maps defined for that group. The force option set to "true" can be used to forcibly remove the node regardless of mappings.
	SNMP	*Not Available*	

Rename an existing igroup	API	*igroup-rename*	
	SNMP	*Not Available*	
Destroy an existing igroup	API	*Igroup-destroy*	By default a group cannot be destroyed if there are existing lun maps defined for that group. This behaviour can be overridden with the use of force option set to "true" which will destroy the initiator group and any associated lun maps.
	SNMP	*Not Available*	

6.2 Associating igroups with Portsets

Binding a portset to an igroup will make the LUN available only on a subset of the storage system's target ports. Any host in the igroup can access the LUNs only by connecting to the target ports in the portset.

If an igroup is not bound to a portset, the LUNs mapped to the igroup are available on all of the storage system's FCP target ports. By using portsets, you can selectively control which initiators can access LUNs and the ports on which they access LUNs.

The following table provides details on how to bind/unbind portsets to the igroups:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Bind an existing igroup to a given portset	API	*igroup-bind-portset*	
	SNMP	*Not Available*	
Unbind an existing igroup to a given portset	API	*igroup-unbind-portset*	
	SNMP	*Not Available*	

6.3 Managing igroups' Attributes

The following table provides details on how to set the igroup attributes:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Set an attribute to an existing igroup	API	*igroup-set-attribute*	Currently supported attributes are: os-type, throttle_reserve, throttle_borrow, and alua
	SNMP	*Not Available*	

6.3.1 igroups' Attributes' details:

os-type

The valid values for "os-type" are "default", "solaris", "windows", "hpux", "aix", "linux", "netware" and "vmware". Setting the "os-type" attribute will perform checks with the cluster partner if this filer is running in the 'single_image' fcp cfmode and this igroup is an FCP igroups. The optional force argument can be used to override these checks if the cluster partner can not be reached. It is also strongly recommended the "default" os-type not be used and for the caller of this API to always require the proper OS type information to be obtained. Using "default" is dangerous because it could potentially cause problems with LUN access.

ALUA (Asymmetric Logical Unit Access):

ALUA defines a standard set of SCSI commands for discovering and managing multiple paths to LUNs on Fibre Channel and iSCSI SANs. Although MPIO software is still required to run on the host, additional host-specific plug-ins are no longer required, as long as the host supports the ALUA standard.

ALUA is disabled by default for Data ONTAP 7.2, you must manually set ALUA to On for each igroup. While ALUA works with all iSCSI storage systems, it only works in single_image cfmode with clustered Fibre Channel storage systems.

Make sure that your host supports ALUA before enabling it. Enabling ALUA for a host that does not support it can cause host failures during cluster failover.

Throttle

Each physical port on the target HBA in the storage system has a fixed number of command blocks for incoming initiator requests. When initiators send large numbers of requests, they can monopolize the command blocks and prevent other initiators from accessing the command blocks at that port.

igroup throttles can help to:

- limit the number of concurrent I/O requests an initiator can send to the storage system.
- prevent initiators from flooding a port and preventing other initiators from accessing a LUN.
- ensure that critical initiators are guaranteed access to the queue resources and that less-critical initiators are not flooding the queue resources.

Data ONTAP calculates the total amount of command blocks available and allocates the appropriate number to reserve for an igroup, based on the percentage specified when a throttle is created for that igroup. Data ONTAP does not allow to reserve more than 99 percent of all the resources. The remaining command blocks are always unreserved and are available for use by igroups without throttles.

throttle-reserve attribute of the igroups specifies what percentage of the queue resources they can reserve for their use. For example, if you set an igroup's throttle to be 20 percent, 20 percent of the queue resources available at the storage system's ports are reserved for the initiators in that igroup. The remaining 80 percent of the queue resources are unreserved. In another example, if you have four hosts and they are in separate igroups, you might set the igroup throttle of the most critical host at 30 percent, the least critical at 10 percent, and the remaining two at 20 percent, leaving 20 percent of the resources unreserved.

Throttle-borrow attribute of an igroup enables it to exceed its limit by borrowing from the following resources:

- The pool of unreserved resources to handle unexpected I/O requests
- The pool of unused reserved resources, if those resources are available

7 Portset Management using NetApp Open Interfaces

Port sets are associated with igroups. If an igroup is not associated with a port set then an initiator that belongs to that igroup can see its target luns on all ports. If a port set is bound to an igroup then the initiator that belongs to that igroup can only see its target luns on the ports that belong to the port set.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Get the details of all the portsets or for a given portset	API	*portset-list-info*	Returns an array of *portset-info[]* which has the details for each of the portset on the system or details of the given portset
	SNMP		
Create a new portset	API	*portset-create*	Possible values for 'portset-type' input parameter are: 'iscsi' or 'fcp'. Currently only 'fcp' is supported
	SNMP	*Not Available*	
Add a port to an existing portset	API	*portset-add*	Input parameter '*portset-port-name'* can be input in two styles. The filername:slotletter format will add the port from a specific filer. For example: "buxton:4a" The slotletter format will add the port from both the local and partner filers. For example: "4a"
	SNMP	*Not Available*	
Delete a port from an existing portset	API	*portset-remove*	the name of the port that is to be removed from the portset can be input in two styles. The filername:slotletter format will remove the port from a specific filer. For example: "buxton:4a" The slotletter format will remove the port from both the local and partner filers. For example: "4a"
	SNMP	*Not Available*	
Destroy an existing portset	API	*portset-destroy*	By default a set cannot be destroyed if there are existing igroups associated with that portset. '*force'* option will forcibly destroy the portset, even if there are existing igroup bindings

	SNMP	*Not Available*	

8 LUN Management using NetApp Open Interfaces

8.1 Setting up a LUN

The following are the steps involved in setting up a LUN so that it is ready to be used by the hosts:

1. Determine the space required by the LUN.
2. Check if the Volume is suitable to create a LUN
    a. Check that the volume has enough space to accommodate the LUN.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To check what is the maximum possible size of a lun on a given volume or qtree	API	*lun-get-maxsize*	Returns the maximum size for different types of luns and the possible maximum size with or without snapshot reserves.
	SNMP	Not Available	
To check what is the minimum possible size of a lun on a given volume or qtree	API	*lun-get-minsize*	Returns the minimum size for different types of luns (based on the specified OS type). Space reservation does not affect the minimum lun size, thus only a single minimum size is returned.
	SNMP	Not Available	

    b. Data ONTAP requires that the path of a volume or qtree containing a LUN is in the Unicode format.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Check the volume option *create_ucode*	API	*volume-options-list-info*	The options should list *create_ucode=on* if the volume path is in the Unicode format
	SNMP	*volOptions (1.3.6.1.4.1.789.1.5.8.1.7)*	

c. Do not create any LUNs in the system's root volume. Data ONTAP uses this volume to administer the storage system. The default root volume is /vol/vol0.

3. Create the LUN
   a. The LUN can be created in three different ways:
      i. LUN can be created in a volume/qtree by specifying the size of the LUN
      ii. LUN can be created from an existing file.
      iii. LUN can be cloned for an existing LUN
   b. The LUN type parameter should be selected while creating the LUN. LUN type reflects to what use the LUN will be put to and what type of hosts will be accessing the LUN. Though this is an optional input, it is strongly recommended to be set to proper OS type so as to make sure that there are no performance penalties in accessing the LUN.
   c. The LUN can have space reserved for it so as to make sure that the write operations to the LUN do not fail due to unavailability of space. If you disable space reservations, write operations to a LUN might fail due to insufficient disk space and the host application or operating system might crash. By default space-reservation is set to 'on'.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Create a new LUN for the given size	API	*lun-create-by-size*	The lun is created at the lun_path given. No file should already exist at the given lun_path
	SNMP	Not Available	
Create a new LUN from an existing file	API	*lun-create-from-file*	A new lun is created, at the given lun_path (which must be at a qtree root). A hard link is created to the existing file.
	SNMP	Not Available	
Create a clone for the LUN	API	*lun-create-clone*	
	SNMP	Not Available	

4. Map the LUN to an igroup

   A LUN can not be accessed by a host until it is mapped to the igroup that contains the hosts that will access the LUN.

   Following are some guidelines for mapping the LUN to igroup:

a. You can map two different LUNs with the same LUN ID to two different igroups without having a conflict, provided that the igroups do not share any initiators or only one of the LUNs is online at a given time.
b. You can map a LUN only once to an igroup or a specific initiator.
c. You can add a single initiator to multiple igroups. but the initiator can be mapped to a LUN only once. You cannot map a LUN to multiple igroups that contain the same initiator.
d. You cannot use the same LUN ID for two LUNs mapped to the same igroup.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To map a LUN to an existing igroup	API	*lun-map*	'force' option will forcibly map the lun, disabling mapping conflict checks with the cluster partner. If not specified all conflict checks are performed. Returns the LUN ID that is assigned to this map
	SNMP	Not Available	

8.2 Managing LUN availability

The LUN can be made unavailable in following ways:

8.2.1 Using LUN offline/online APIs

- The lun online and lun offline APIs control the availability of LUNs while preserving LUN igroup mappings.
- Before you bring a LUN online or take it offline, make sure that you quiesce or synchronize any host application accessing the LUN

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To bring a LUN online so as to enable block-protocol access to it	API	*lun-online*	'*force'* option will forcibly online the lun, disabling mapping onflict checks with the cluster partner. If not specified all conflict checks are performed

	SNMP	Not Available	
To bring a LUN offline so as to disable block-protocol access to it	API	*lun-offline*	
	SNMP	Not Available	

8.2.2 Mapping/Unmapping a LUN from igroups

- LUNs can be made selectively available/unavailable to specific hosts by mapping/unmapping the LUN to specific igroups.
- LUNs should be brought online before mapping them to an igroup.
- LUN should be taken offline before unmapping them from an igroup.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Check the initiator groups to which the LUN is mapped	API	*lun-map-list-info*	Returns the *initiator-group-info[]* array which has the details of the each initiator group that is mapped to the LUN
	SNMP	*lunMapTable (1.3.6.1.4.1.789.1.17.15.3)*	
To map a LUN to an initiator group	API	*lun-map*	Returns the LUN ID that is assigned to this map
	SNMP	*Not Available*	
To unmap a LUN from an initiator group	API	*lun-unmap*	
	SNMP	*Not Available*	

8.2.3 Removing a LUN

- LUN can be made permanently unavailable by destroying the LUN
- LUN should be brought offline and unmapped from all the igroups it is mapped to before destroying it

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To remove a LUN permanently	API	*lun-destroy*	'*force'* option can be used to destroy a LUN even if it is online or mapped.
	SNMP	Not Available	

8.3 Managing LUN Clones

A LUN clone is a point-in-time, writable copy of a LUN in a Snapshot copy. Changes made to the parent LUN after the clone is created are not reflected in the clone. A LUN clone shares space with the LUN in the backing Snapshot copy. The clone does not require additional disk space until changes are made to it. You cannot delete the backing Snapshot copy until you split the clone from it. When you split the clone from the backing Snapshot copy, you copy the data from the Snapshot copy to the clone. After the splitting operation, both the backing Snapshot copy and the clone occupy their own space.

Cloning is not NVLOG protected, so if the storage system panics during a clone operation, the operation is restarted from the beginning on a reboot or takeover.

Some uses of cloning LUNs:

You can use LUN clones to create multiple read/write copies of a LUN. You might want to do this for the following reasons:

- You need to create a temporary copy of a LUN for testing purposes.
- You need to make a copy of your data available to additional users without giving them access to the production data.
- You want to create a clone of a database for manipulation and projection operations, while preserving the original data in unaltered form.
- You want to access a specific subset of a LUN's data (a specific logical volume or file system in a volume group, or a specific file or set of files in a file system) and copy it to the original LUN, without restoring the rest of the data in the original LUN. This works on operating systems that support mounting a LUN and a clone of the LUN at the same time.

Following table provides the list of APIs that can be used to do LUN cloning operations:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To create a clone for the LUN	API	*lun-create-clone* *lun-create-from-snapshot*	
	SNMP	Not Available	
To start splitting a clone from its parent LUN	API	*lun-clone-split-start*	The API '*lun-clone-start'* used to do the same thing but it is deprecated now
	SNMP	Not Available	
To stop splitting a clone from its parent LUN	API	*lun-clone-split-stop*	The API '*lun-clone-stop'* used to do the same thing but it is deprecated now
	SNMP	Not Available	
To get the status of the split operation on a Clone	API	*lun-clone-split-status-list-info*	The API '*lun-clone-status-list-info'* used to do the same thing but it is deprecated now Return values are: blocks completed and total blocks to clone.
	SNMP	Not Available	

8.4 Managing LUN Characteristics

8.4.1 Resizing LUN

LUN size can be increased or decreased dynamically. However, the host operating system must be able to recognize changes to its disk partitions.

There are following restrictions on resizing a LUN:

On Windows systems, resizing is supported only on basic disks. Resizing is not supported on dynamic disks.
Resizing LUNs is not supported on VxVM version 3.5 or lower
If you want to increase the size of the LUN, the SCSI disk geometry imposes an upper limit of ten times the original size of the LUN. Data ONTAP also imposes a maximum increase to 2 TB.

Following table provides the details of the APIs for changing the LUN size:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Get the size occupied by the LUN in the active FS	API	*lun-get-occupied-size*	Returns the size occupied by the LUN in the Active FS in bytes. This indicates the amount of space used by the LUN.
	SNMP	*lunSizeHigh (1.3.6.1.4.1.789.1.17.15.2.1.5.1) lunSizeLow (1.3.6.1.4.1.789.1.17.15.2.1.4.1)*	LUN size is a 64 bit unsigned integer. *lunSizeHigh* gives most significant 32 bits and *lunSizeLow* gives least significant 32 bits of the 64 bit unsigned integer.
Change the size of the given LUN	API	*lun-resize*	'*force*' option should be set to reduce the size of the LUN. This is required to avoid accidentally reducing the LUN size
	SNMP	Not Available	

8.4.2 Accessing a LUN with NAS protocols

When you create a LUN, it can be accessed only with the iSCSI or FCP protocol by default. However, you can use NAS protocols to make a LUN available to a host if the NAS protocols are licensed and enabled on the storage system.

Restrictions in NAS mode:

A LUN cannot be extended or truncated using NFS or CIFS protocols.

If you want to write to a LUN using a NAS protocol, you must take the LUN offline or unmap it to prevent an iSCSI or FCP host from overwriting data in the LUN.

To make a LUN accessible to a host that uses a NAS protocol use the following APIs:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To enable file system protocol based access to the given LUN	API	*lun-set-share*	Possible values for the input 'share-type': all, none, read, write
	SNMP	Not Available	

8.4.3 Renaming a LUN

The original name of the LUN can be renamed. Also a LUN can be moved to different path. The only restriction is, if you are organizing LUNs in qtrees, the existing path and the new path must be in the same qtree.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Rename or move the given LUN	API	*lun-move*	
	SNMP	Not Available	

8.5 Viewing and Modifying LUN Details

LUN details like description, name, serial number, space-reservation status, type can be viewed or modified as described in the table below:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To get the details of the given LUN or all the LUNs on the storage system	API	*lun-list-info*	Output is an array of *lun-info[]* objects with each lun-info object corresponding to a particular LUN
	SNMP	*lunTable (1.3.6.1.4.1.789.1.17.15.2)*	
Get SCSI disk geometry for a given LUN	API	*lun-get-geometry*	The output provides bytes, sectors, tracks, cylinders details of the LUN
	SNMP	Not Available	
Get the SCSI INQUIRY response data for the igroup that the LUN is mapped to	API	*lun-get-inquiry-info*	The output provides product id, vendor id and firmware revision
	SNMP	Not Available	
Check if the given LUN has SCSI reservations	API	*lun-has-scsi-reservations*	The output is 'true' if given LUN has a scsi-2 or scsi-3 style reservation held, 'false' otherwise
	SNMP	Not Available	

Check if the given initiator port has SCSI reservations	API	*lun-port-has-scsi-reservations*	The output is 'true' if given port has a scsi-2 or scsi-3 style reservation held, 'false' otherwise
	SNMP		
Get a named attribute for the given LUN	API	*lun-get-attribute*	
	SNMP	Not Available	
Set a named attribute for the given LUN	API	*lun-set-attribute*	
	SNMP	Not Available	
Clear a named attribute for the given LUN	API	*lun-unset-attribute*	
	SNMP	Not Available	
Get the description for a LUN	API	*lun-get-comment*	
	SNMP	*lunComment (1.3.6.1.4.1.789.1.17.15.2.1.3.1)*	
Set the description for a LUN	API	*lun-set-comment*	
	SNMP	Not Available	
Get the select attribute for a LUN.	API	*lun-get-select-attribute*	Possible values for the select attribute of a LUN: "active" - this is an active LUN "copy" - this is a clone or copy of an active LUN "mirror" - this is a mirror of an active LUN
	SNMP	Not Available	
Set the select attribute for a LUN.	API	*lun-set-select-attribute*	The select attribute is used by mult-pathing software on hosts to discriminate between luns when mirrored or cloned copies of a vdisk are mapped to the same host.
	SNMP	Not Available	
Get the serial number for the given LUN.	API	*lun-get-serial-number*	The serial number is a 12 character string
	SNMP	*lunSerialNumber (1.3.6.1.4.1.789.1.17.15.2.1.7.1)*	
Set the serial number for the given LUN.	API	*lun-set-serial-number*	The LUN should be made offline before setting the serial number.
	SNMP	Not Available	

Get the space reservation setting for the given LUN	API	*lun-get-space-reservation-info*	
	SNMP	*lunSpaceReserved (1.3.6.1.4.1.789.1.17.15.2.1.20.1)*	
Set the space reservation setting for the given LUN	API	*lun-set-space-reservation-info*	
	SNMP	Not Available	
Get the state of the restore for the given LUN	API	*lun-restore-status*	Returns 'true' if file restore is currently running for this vdisk, 'false' otherwise
	SNMP	Not Available	

8.6 Managing LUN Performance

LUN performance depends on the **amount of traffic** and the **type of traffic** from the initiators that are mapped to the LUN. LUNs have fixed number of command blocks, if the amount of traffic is more, then the LUN command blocks will be full and the subsequent request from the initiators will result in QUEUEFULL status. If the LUN is not laid out in sequential blocks, sequential read commands take longer to complete because each command might require an additional disk seek operation.

8.6.1 Measuring LUN Performance Statistics

The following APIs provide the read/write statistics for a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To get the block protocol access statistics for LUN	API	*lun-stats-list-info*	An array of *lun-stats-info[]* objects, each object represents the statistics for a particular on the storage system
	SNMP	*lunTable (1.3.6.1.4.1.789.1.17.15.2)*	
To reset the block protocol access statistics of LUN to zeroes	API	*lun-reset-stats*	
	SNMP	Not Available	

The API provides the following LUN statistics:

block-size: Disk block size for this LUN in bytes.
last-zeroed: Total number of seconds since the statistics for this lun were last zeroed.
other-ops: Total number of other scsi ops executed.
Path: path of the LUN. (for example, "/vol/vol0/lun1")
read-blocks: Number of disk blocks read.
read-ops: Total number of scsi read ops executed.
write-blocks: Number of disk blocks written.
write-ops: Total number of scsi write ops executed.

8.6.2 Improving the Read/Write Performance of the LUN

If a LUN is not laid out in sequential blocks, sequential read commands take longer to complete because each command might require an additional disk seek operation. Sequential block layout improves the read/write performance of host applications that access data on the storage system. Reallocation scans can be used to ensure that blocks in a LUN are laid out sequentially.

A reallocation scan evaluates how the blocks are laid out in a LUN. Data ONTAP performs the scan as a background task, so applications can rewrite blocks in the LUN during the scan. Repeated layout checks during a scan ensure that the sequential block layout is maintained during the current scan. A reallocation scan does not necessarily rewrite every block in the LUN. Rather, it rewrites whatever is required to optimize the layout of the LUN.

Reallocation scans can be performed on LUNs when they are online. You do not have to take them offline. You also do not have to perform any host-side procedures when you perform reallocation scans.

Reallocation scans are managed by performing the following tasks:

8.6.2.1 Enable/Disable the Reallocation scan

Reallocation scans are disabled by default. You must enable reallocation scans globally on the storage system before you run a scan or schedule regular scans.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To enable system wide global reallocation jobs	API	*reallocate-on*	
	SNMP	Not Available	
To disable system wide global reallocation jobs	API	*reallocate-off*	
	SNMP	Not Available	

8.6.2.2 Define a Reallocation scan

Following are some of the parameters that define a reallocation scan for the LUN:

Threshold- It is a number between 3 (layout is moderately optimal) and 10 (layout is not optimal). The default is 4.

A scan checks the block layout of a LUN before reallocating blocks. If the current layout is below the threshold, the scan does not reallocate blocks in the LUN. If the current layout is equal to or above the threshold, the scan reallocates blocks in the LUN.

Interval - It is the interval, in hours, minutes, or days, at which the scan is performed. The default interval is 24 hours.

The countdown to the next scan begins only after the first scan is complete. For example, if the interval is 24 hours and a scan starts at midnight and lasts for an hour, the next scan begins at 1:00 a.m. the next day—24 hours after the first scan is completed.

The following APIs are used to define a reallocation scan for a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To define a new reallocation scan for a LUN	API	*reallocate-start*	
	SNMP	Not Available	

8.6.2.3 Managing a Reallocation scan schedule

Reallocation scans can be run according to a schedule. The schedule created replaces any interval specified while defining the reallocation scan for the LUN. The existing reallocation scan schedule that is defined for a LUN can be deleted. If you delete a schedule, the scan runs according to the interval that was specified while defining the reallocation scan for the LUN.

The following APIs are used to manage the reallocation scan schedules for a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To set a schedule for a reallocation job for a LUN	API	*reallocate-set-schedule*	The schedule parameter is in the format "*minute hour day_of_month day_of_week*" *minute* is a value from 0 to 59. *hour* is a value from 0 (midnight) to 23 (11:00 p.m.). *day_of_month* is a value from 1 to 31. *day_of_week* is a value from 0 (Sunday) to 6 (Saturday). You can enter a number, a range (x-y), or

			a comma-separated list of values (x,y) for a field or "*". A wildcard character (*) indicates every value for that field
	SNMP	Not Available	
To delete a schedule for a reallocation job for a LUN	API	*reallocate-delete-schedule*	
	SNMP	Not Available	

8.6.2.4 Managing a Reallocation scan operation

Reallocation scan for a LUN can be temporarily paused by quiescing the scan or it can be forced to start asynchronously by restarting the scan.

You can quiesce a reallocation scan that is in progress and restart it later. The scan restarts from the beginning of the reallocation process. For example, if you want to back up a LUN, but a scan is already in progress, you can quiesce the scan.

The following APIs are used to quiesce the reallocation scan schedules for a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To quiesce the reallocation scan for a LUN	API	*reallocate-quiesce*	
	SNMP	Not Available	

You might restart a scan for the following reasons:

You quiesced the scan by using the reallocate quiesce command, and you want to restart it.

You have a scheduled scan that is idle (it is not yet time for it to run again), and you want to run it immediately.

The following APIs are used to restart the reallocation scan schedules for a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To restart the reallocation scan that was	API	*reallocate-restart*	'ignore-checkpoint' parameter can be set to start the scan from beginning rather than from the check-pointed state

quiesced	SNMP	Not Available	

8.6.2.5 Getting the details of a scan

The following are some of the details of scan that can be got:

Interval: Interval between reallocation jobs
layout-factor: Current allocation layout factor. The range is from 1 (ideal layout) upwards.
scan-detail: If a reallocation job is running, detail on the current scan, including the progress when known.
Schedule: Schedule for reallocation.
State: The current state of the reallocation job. The states are Idle, Checking, reallocating, Deleting or Quiesce.
Threshold: The reallocation threshold.

The following APIs are used to get the reallocation scan details for a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
Get the details of the reallocation scan defined for a LUN	API	*reallocate-list-info*	Use 'verbose' option for getting detailed info
	SNMP	Not Available	

8.6.2.6 Stopping the scan for a LUN

If you delete a LUN, you do not delete the reallocation scan defined for it. If you take the LUN offline, delete it, and then reconstruct it, you still have the reallocation scan in place. To delete a reallocation scan for a LUN, you need to stop the scan.

The following APIs are used to stop the reallocation scan of a LUN:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To delete the reallocation scan for a LUN	API	*reallocate-stop*	
	SNMP	Not Available	

8.7 Troubleshooting LUN Configuration settings

LUN configuration settings can have different issues. Some of the configuration problems could be OS type mismatch, ALUA setting mismatch between partner filers, FCP settings not configured properly, HBA status, improper cluster mode settings etc.

8.7.1 Checking LUN/FCP Configuration warnings

The following APIs return a list of lun/fcp configuration warnings. These warnings are not related to filer cluster configuration.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To get the list of warnings pertaining to the LUN/fcp configuration settings	API	*lun-config-check-info*	
	SNMP	Not Available	

The API provides the following configuration warnings:

lua-setting-mismatch-info: list of initiator groups for which the ALUA (Asymmetric Logical Unit Access) settings do not match between local and partner filers. ALUA is typically used by host multi-path software to recognize primary and secondary paths to a Logical Unit when more than one path are available to the Logical Unit. If the ALUA setting does not match between the local and partner filers, it would affect the host multi-path software's ability to distinguish primary and secondary paths. This could lead to incorrect system behavior.

fcp-down-hbas: list of HBA names that have no link or are configured down. Possible states for the down HBA are: STARTUP, UNINITIALIZED, INITIALIZING, FIRMWARE, LINK NOT CONNECTED, WAITING FOR LINK UP, ONLINE LINK DISCONNECTED, RESETTING, OFFLINE,OFFLINED BY USER/SYSTEM, Unknown state

invalid-vsa-settings: Information about an initiator group with an invalid Volume Set Addressing (VSA) setting for its ostype. Only 'hpux' initiator groups should have VSA enabled. All other initiator groups should have it disabled. Incorrect settings can cause hosts to not be able to access some or all of their luns.

mixed-ostype: List of initiators which are members of initiator groups of differing ostypes. An initiator can only be a member of initiator groups which have the same ostype across all the initiator groups it is a member of.

mixed-vsa-initiators: List of initiators which are members of initiator groups with differing VSA settings. An initiator can only be a member of initiator groups which have the same VSA setting across all the initiator groups it is a member of.

8.7.2 Checking initiator groups and FCP cfmode Configuration warnings

The following APIs return a list of configuration warnings related to initiator groups and the fcp cfmode setting:

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To get the list of warnings pertaining to the initiator groups and fcp cfmode settings	API	*lun-config-check-cfmode-info*	
	SNMP	Not Available	

The API provides the following configuration warnings:

invalid-cfmode-settings: Local and partner cfmode settings if they do not match. cfmodes need to match on both filers in the cluster or cluster failover will not work properly

invalid-ostype-cfmode-settings: List of initiator groups with invalid ostypes for the fcp-cfmode. Some host operating systems are only compatible with certain fcp-cfmodes

invalid-use-partner-cfmode-settings: List of initiator groups with invalid use_partner settings for the FCP cluster failover mode (fcp-cfmode). Incorrect settings can cause hosts to lose luns during takeover.

invalid-use-partner-ostype-settings: List of initiator groups with invalid use_partner settings for the initiator group's ostype. Some host operating systems should only have certain use_partner settings to ensure proper behavior during failover.

8.7.3 Checking single_image' fcp cfmode Configuration warnings

The following APIs return a list of configuration warnings that pertain to problems specific to the 'single_image' fcp cfmode. These errors must be fixed prior to upgrading any filer cluster to run in 'single_image' mode.

Operation	**Open Interface**	**Command Details** (API for ONTAPI, OID for SNMP)	**Comments**
To get the list of warnings pertaining to the 'single_image' fcp cfmode	API	*lun-config-check-single-image-info*	
	SNMP	Not Available	

The API provides the following configuration warnings:

conflicting-initiators: List of initiators which belong to initiator groups on the local filer which have different os type settings from the initiator groups on the partner.

conflicting-maps: Only one lun in the cluster can be mapped to an initiator at a given lun-id. If a lun on each filer is mapped to the same initiator at the same lun-id there will be a conflict. These conflicts need to be resolved before a filer can be upgraded to run in the 'single_image' fcp cfmode. A conflict can be resolved by unmapping one lun and remapping it to an unused lun-id.

invalid-nodename-settings: When running in the the single_image cfmode the fcp nodename needs to be same on each filer in the cluster. If the nodenames are different the nodenames will be returned. If the cfmode is not set to single_image, then the nodenames will not be checked.