iSCSI Implementation for Dell EMC Storage Arrays Running PowerMaxOS

January 2022

H14531.5

White Paper

Abstract

This document provides an in-depth overview of the PowerMaxOS iSCSI implementation on Dell EMC PowerMax and VMAX All Flash storage arrays. The technology surrounding iSCSI is discussed as well as an in-depth review of the PowerMaxOS iSCSI target model.

Dell Technologies

Copyright

Published in the USA January 2022 H14531.5.

Contents

Executive summary

Overview

Dell Technologies is excited to offer iSCSI connectivity to our existing and new customers. The iSCSI storage protocol provides a potentially lower-cost-alternative connectivity method between hosts or virtual machines to Dell EMC PowerMaxOS based storage arrays. At a high-level glance, the primary benefits of the iSCSI storage protocol are as follows:

- Makes consolidated storage possible for a wide range of businesses
- Enables cost-effective, scalable, secure, and highly available storage area networks (SANs)
- Leverages existing management skills and network infrastructure
- Delivers performance comparable to Fibre Channel
- Provides interoperability using industry standards

The PowerMaxOS iSCSI solution has been architected to take advantage of virtual local area networks (VLANs) to provide customers with greater host, port, and connection densities. The ability to use VLANs also provides built in multitenancy capabilities since the front-end storage ports can be virtualized and partitioned. This design makes the PowerMaxOS iSCSI solution an ideal connectivity choice when considering lower-cost storage options for converged infrastructures and all virtualized environments.

Audience

This document is intended for Dell Technologies field personnel, including technology consultants, and for customer storage architects, administrators, and operators involved in managing, operating, or designing a storage infrastructure that contains PowerMaxOS based storage arrays.

Revisions

Date	Description
October 2016	Initial release
April 2018	Updated for PowerMaxOS
September 2018	Updates for PowerMaxOS Q3 2019 release
September 2020	Updates for PowerMaxOS Q3 2020 release
February 2021	Minor updates
January 2022	Template update

We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

Author: James Salvadore

Note: For links to other documentation for this topic, see the PowerMax Info Hub.

iSCSI overview

Introduction

iSCSI is a transport layer protocol that uses TCP/IP to transport SCSI packets, enabling the use of Ethernet-based networking infrastructure as a storage area network (SAN). Like Fibre Channel and other storage transport protocols, iSCSI transports block level data between an initiator on a server and a target on a storage device. IBM developed iSCSI as a proof of concept in 1998 and was ratified as a transport protocol by the Internet Engineering Task Force (IETF) in 2003. The current iSCSI standard is IETF RFC 7143 and can be found at https://tools.ietf.org/html/rfc7143.

Key iSCSI concepts and terminology

This white paper consistently uses or makes reference to specific concepts and terminology. The following table provides a detailed list of these terms and their definitions:

Table 1. Key iSCSI technologies and terminology

Terminology (first instance in document)	Equivalent term (later instances in document)	Definition
Open Systems Interconnection Model	OSI model	A seven-layer conceptual model that characterizes and standardizes the communication functions of a telecommunication or computer network system without regard to its underlying internal structure and technology. The primary layers are the application (Layer 7), Presentation (Layer 6), Session (Layer 5), Transport (Layer 4), Network (Layer 3), Datalink (Layer 2), Physical (Layer 1)
Ethernet	Ethernet	A family of computer networking technologies operating at the OSI physical layer (Layer 1) also providing services to the OSI datalink layer (Layer 2). Ethernet is comm*only used in local area networks (LAN) and wide area networks (WAN). Systems communicating over Ethernet based networks divide a stream of data into frames. Each frame contains source and destination addresses, and error-checking data so that damaged frames can be detected, discarded, and retransmitted when needed. Ethernet can use physical mediums of twisted pair and fiber optic links that can reach speeds of 10 Gbps (10 GbE), 25 Gbps, 40 Gbps, 50 Gbps, and now 100 Gbps.
Virtual Local Area Network (VLAN)	VLAN	Any broadcast domain that is partitioned and isolated in computer network at the datalink layer (Layer 2). VLANs work by applying tags to network packets and handling these tags in networking systems – creating the appearance and functionality of network traffic that is physically on a single network but acts as if it is split between separate networks.
Transmission Control Protocol/Internet Protocol	TCP/IP	A suite of communication protocols used to interconnect devices on communication networks. TCP/IP specifies how data can be exchanged over networks. TCP defines how applications can create channels of communication across a network. It manages how data is assembled into smaller packets before it is transmitted over the network and how it is to be reassembled at the destination address. In the OSI model, TCP provides services to the transport layer (Layer 4) and some services to the session layer (Layer 5). IP specifically defines how to address and route each packet to ensure it reaches the correct destination on the network. In the OSI model, IP provides services to the network layer (Layer 3).

Terminology (first instance in document)	Equivalent term (later instances in document)	Definition
Small Computer System Interface (SCSI)	SCSI	A set of standards for physically connecting and transferring data between computers and peripheral devices such as disk storage. The SCSI standards define commands, protocols, and electrical and optical interfaces.
Storage Area Network	SAN	A specialized, high-speed network that provides block-level network access to storage. A SAN consists of two types of equipment: initiator and target nodes. Initiators, such as hosts, are data consumers. Targets, such as disk arrays or tape libraries, are data providers. A SAN presents storage devices to a host such that the storage appears locally attached. SAN initiators and targets can be interconnected using various technologies, topologies, and transport layer protocols.
Internet Small Computer Serial Interface (iSCSI)	iSCSI	A transport layer protocol that uses TCP/IP to transport SCSI commands enabling Ethernet based networks to function as a storage area network (SAN). iSCSI uses TCP/IP to move block data between iSCSI initiators nodes and iSCSI target nodes
iSCSI Initiator Node	Initiator	Host-based hardware (virtual or physical) or software that sends data to and from iSCSI target nodes (storage arrays). The initiator makes requests for the data to be read from or written to the storage. In case of read operations, the initiator sends a SCSI READ command to the peer who acts as a target and in return the target sends the requested data back to the initiator. In the case of a write operation, initiator sends a SCSI WRITE command followed by the data packets to the target. The initiator always initiates the transactions.
iSCSI Target Node	Target	Storage arrays, tape drives, storage servers on a SAN. In iSCSI, targets can be associated with either virtual or physical entities. A storage array target exposes one or more SCSI LUNs to specific initiators. A target is the entity that processes the SCSI commands from the initiator. Upon receiving the command from the initiator, the target runs the command and then sends the requested data and response back to the initiator. A target cannot initiate any transaction.
iSCSI IP Interface (Network Portal)	IP Interface	Primary gateway for access to iSCSI nodes. IP Interfaces contain key network configuration information such as: IP Address, Network ID, VLAN information, and TCP Port Number. An IP Interface can only provide access to a single iSCSI target; however, an iSCSI target can be accessed through multiple IP Interfaces.
PowerMaxOS 5978 (microcode)	PowerMaxOS	The PowerMaxOS 5978 release supports PowerMax NVMe arrays, dedupe, and other software enhancements and is offered with VMAX All Flash arrays.
PowerMaxOS Network Identity	Network ID/NetID	A PowerMaxOS construct that is used internally by the system to associate an array IP interface with an array iSCSI target. The PowerMaxOS Network ID is specific to a single director on the array and is not visible to external switches or hosts.
iSCSI Qualified Names	IQN	Primary mechanism to identify iSCSI nodes on a network. These names are a human-readable ASCII string that can be either user or algorithmically generated; however, the iSCSI Name must be unique on a per network basis in order to avoid duplication.

Terminology (first instance in document)	Equivalent term (later instances in document)	Definition
iSCSI Protocol Data Unit (PDU)	PDU	SCSI commands encapsulated and placed into packets by the iSCSI Protocol at the session layer (Layer 5).
iSCSI Connection	Connection	A TCP/IP connection that ties the session components together. The IP addresses and TCP port numbers in the IP Interfaces define the end points of a connection.
iSCSI Session	Session	Primary communication linkage between iSCSI initiator and target nodes. The session is the vehicle for the transport of the iSCSI PDUs between the initiators and target nodes.
Challenge Handshake Authentication Protocol (CHAP)	CHAP	The most commonly used iSCSI authentication method. CHAP verifies identity using a hashed transmission of a secret key between initiator and target.

Primary benefits of iSCSI

With the proliferation of 10 GbE networking in the last few years, iSCSI has steadily gained footprint as a deployed storage protocol in data centers. For data centers with centralized storage, iSCSI offers customers many benefits. Developed by the Internet Engineering Task Force (IETF) as a response to the need for interoperability in networked storage, iSCSI lets businesses create TCP/IP based SANs that deliver the performance comparable to Fibre Channel, but at a lower cost.

The iSCSI protocol can achieve lower costs because the protocol allows for the encapsulation of SCSI commands on a standard TCP/IP connection and transported over an Ethernet based network. This means that host standard Ethernet network interface cards (NICs) and network switches can be used to carry storage traffic, eliminating the need for a more expensive specialized storage network using separate switches and host bus adapters (HBAs). Using fewer deployed ports means fewer deployed switches, which can result in lower infrastructure, administration, power consumption, and cooling costs. Cost reduction and consolidation of equipment are primary drivers behind the push to converged infrastructures; hence why iSCSI is a highly considered storage protocol for customers looking to go converged.

Core components of iSCSI

iSCSI architecture is made up of a set of core components. These components are initiator and target nodes, iSCSI names, IP Interfaces, sessions and connections, and security.

Initiators and target nodes

A storage area network (SAN) consists of two types of equipment: initiator and target nodes. Initiators, such as hosts, are data consumers. Targets, such as disk arrays or tape libraries, are data providers. iSCSI based SANs use initiators and targets in the same manner.

- **iSCSI initiator nodes** are typically host-based software or hardware that sends data to and from iSCSI target nodes (storage arrays). In data migration between storage arrays, the source array can act as an initiator.

- **iSCSI target nodes** expose one or more SCSI LUNs to specific iSCSI initiators. On the enterprise storage level, iSCSI target nodes are logical entities, not tied to a specific physical port.

iSCSI initiators must manage multiple, parallel communication links to multiple targets. Similarly, iSCSI targets must manage multiple, parallel communication links to multiple initiators. Several identifiers exist in the iSCSI protocol to make this happen, including iSCSI Name, ISID (iSCSI session identifiers), TSID (target session identifier), CID (iSCSI connection identifier) and iSCSI portals.

Names

iSCSI nodes are identified by a unique **iSCSI Name.** iSCSI names are a human readable ASCII string and must be unique on a per NetID/Network ID basis. iSCSI names can be both user and algorithmically generated. iSCSI Names are formatted in two different ways:

- Enterprise Unique Identifier (EUI): eui.0123456789ABCDEF
- iSCSI Qualified Name (IQN): Most commonly used naming format: iqn.2001-05.com.mircosoft:ProdHost

IP interfaces

iSCSI Nodes are accessed through **IP Interfaces** (sometimes called Network Portals). iSCSI IP Interfaces contain key network configuration information such as:

- IP Address
- VLAN information
- TCP Port Number

An iSCSI IP Interface can only provide access to a single iSCSI node; however, an iSCSI node can be accessed through multiple IP Interfaces.

Sessions and connections

iSCSI initiator and target nodes communicate by a linkage called an **iSCSI session**. The session is the vehicle for the transport of the iSCSI PDUs between the initiators and target nodes. Each session is started by the initiator logging into the iSCSI target. The session between the initiator and target is identified by an **iSCSI session ID**. Session IDs are not tied to the hardware and can persist across hardware swaps.

Session components are tied together by a TCP/IP **connection.** The IP addresses and TCP port numbers in the IP interfaces define the end points of a connection. The iSCSI protocol allows for multiple connections within a single session (MC/S) as means to provide connection resiliency to a target that is presenting volumes to the host; however, MC/S is rarely done with enterprise iSCSI connections as most enterprise implementations use host-based multipath I/O software (MPIO). Using host-based MPIO, a single host initiator can access the same devices by presenting them through multiple targets on the storage array. This allows the host to see the devices through multiple paths. Each path from the initiator to the targets will have its own session and connection. This connectivity method is often referred to as “port binding.” The diagram below shows these iSCSI connectivity methods:

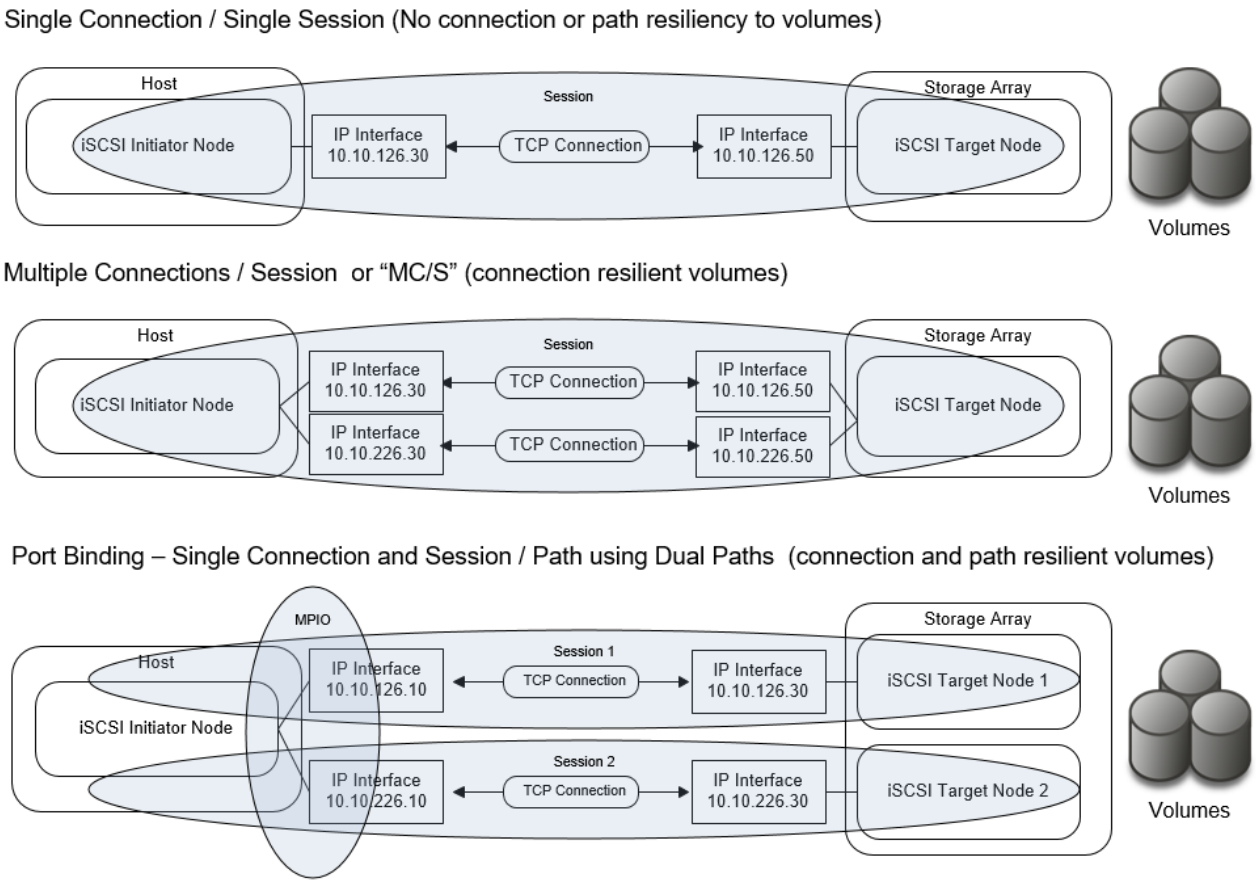


Figure 1. iSCSI connectivity methods

Security and authentication

The most commonly used iSCSI authentication method is Challenge Handshake Authentication Protocol (CHAP). CHAP verifies identity using a hashed transmission of a secret key between initiator and target. The iSCSI specification RFC 7143 defines that CHAP security is the only "must-support" authentication protocol. All other protocols such as Kerberos are considered to "in addition to" CHAP.

The CHAP secret key is a user-defined string up to 32 ASCII characters, or 64 binary characters (binary values should be prefixed with the string "0x"). Note: Windows users need to specify a secret key between 12 and 16 ASCII characters. The users also create a credential name (CHAP username) string between 8 and 256 ASCII characters. For more information about CHAP iSCSI considerations, refer to RFC 7143 section 9.2.1, which can be found at https://tools.ietf.org/html/rfc7143.

Using CHAP, the target initiates the challenge to the initiator for the secret key. It periodically repeats the challenge to guard against replay attacks. CHAP can be a unidirectional /one-way protocol, in which only the target authenticates the initiator, but it can be implemented in two directions (bidirectional and mutual) where the initiator also authenticates the target to provide security for both ends. The following bullets detail these methods:

- In **one-way** CHAP authentication, also called **unidirectional**, the target authenticates the initiator, but the initiator does not authenticate the target. With CHAP one-way authentication, the storage array challenges the host during the initial link negotiation process and expects to receive a valid credential and CHAP secret in response. When challenged, the host transmits a CHAP credential and CHAP secret to the storage array. The storage array looks for this credential and CHAP secret internally or on a network “RADIUS” server. Once a positive authentication occurs, the storage array sends an acceptance message to the host. However, if the storage array fails to find any record of the credential or secret pair, it sends a rejection message, and the link is closed.
- In **two-way** CHAP authentication, also called **bi-directional or mutual**, an additional level of security enables the initiator to authenticate the target after the target authenticates the initiator. With two-way CHAP authentication, the host challenges and authenticates the storage array. This provides an extra layer of authentication and security in the iSCSI configuration as both the target and initiator act as authenticators and peers.

How iSCSI works

As said earlier, the iSCSI protocol allows for the encapsulation of SCSI commands on a standard TCP/IP connection and transported over an Ethernet based network between a host and a storage array. These actions can be separated into two processes: the Login Process and the Data Transfer Process.

The login process

When an iSCSI host initiator wishes to communicate with an iSCSI storage array, it begins with a login request. The login request contains information about “who” is sending the request and “what” storage target the host wishes to communicate with. If CHAP is being used, the request will contain CHAP information. The iSCSI storage array will authenticate the host initiator using the CHAP information. If the authentication is successful, the login is able to complete, and a “session” is established between the host initiator and the storage array target. Once the session is established, the transfer of SCSI commands and data between the host initiator and storage array target can begin. It is not uncommon for iSCSI sessions to remain active for days or months. When either the host or the storage array decides to close the session, it will either issue a logout command. When the session closes, the ability transfer of SCSI commands and data between the host and storage will also end.

The data transfer process

iSCSI transports block level data between an initiator on a host and a target on a storage array in the following manner:

- The process starts with an application on a host generating a block level I/O (Layer 7)
- The I/O is sent to the presentation Layer 6 where the I/O is translated to the SCSI command set.
- At the session Layer 5 (where iSCSI operates), the iSCSI protocol encapsulates the SCSI commands and assembles them into packets called Protocol Data Units (PDUs).
- These PDUs are then sent to the Transport Layer 4, where it is encapsulated in a TCP segment (the i in iSCSI).
- It is then sent to the Network Layer 3 where it is placed into an IP datagram to form the TCP/IP packet.
- The TCP/IP packet is then placed into an Ethernet frame at the Datalink Layer 2
- The “iSCSI Ethernet” frame is then sent out onto the physical network Layer 1 to be sent to the storage target.

This process is shown in the following figure:

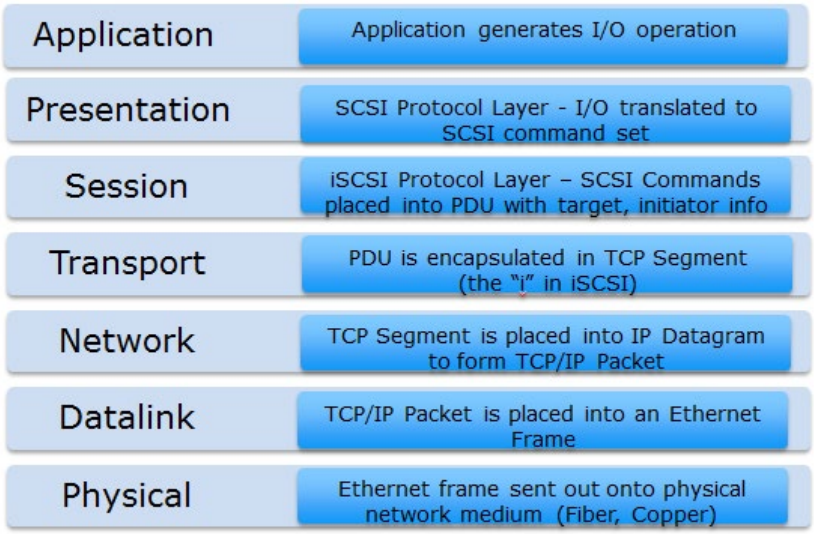


Figure 2. How iSCSI works

When the target side receives iSCSI Ethernet frames, the target datalink layer will remove the frame encapsulation and pass the results up to the Network Protocol Layer. The Network layer will remove the IP datagram encapsulation, and the Transport layer will remove the TCP segment encapsulation, leaving a PDU to be passed up to the session layer (iSCSI protocol layer). The iSCSI Protocol Layer will remove the SCSI data from the PDU and pass it to the presentation layer for interpretation and processing.

The figure below shows the different components in an iSCSI Ethernet frame.

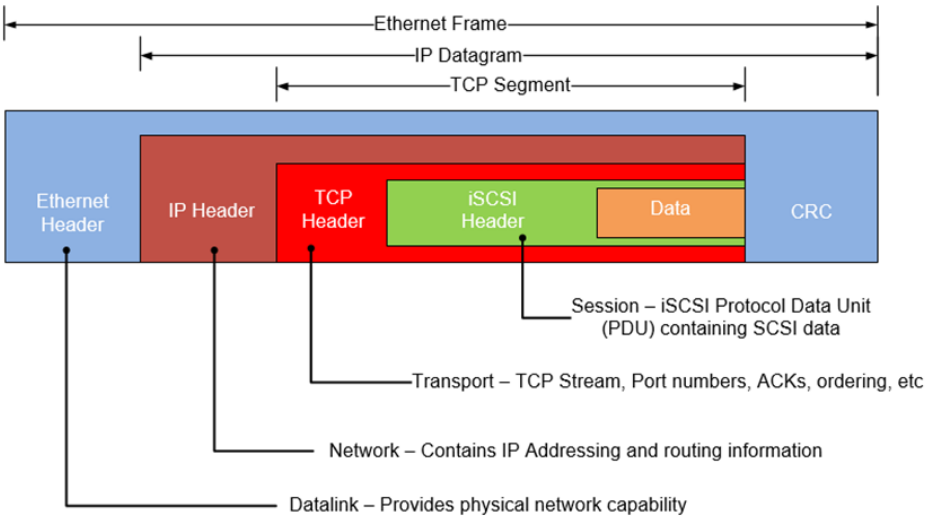


Figure 3. iSCSI Ethernet frame

Essentially there is no difference between an iSCSI Ethernet frame with a standard Ethernet frame except what is the payload in the TCP segment - the iSCSI PDU. There is nothing in the TCP Segment Header to indicate that the TCP Data Segment contains data of a specific protocol. The TCP/IP definition does not prevent iSCSI PDUs and other network data from being transmitted on the same network. Similarly, there is nothing that requires that they be mixed, so a network administrator can determine whether an isolated subnet for iSCSI is necessary or not. The ability to carry multiple types of data in TCP Segment header is what allows modern Ethernet switches to the transport of iSCSI, IP, and Fibre Channel over Ethernet (FCoE) on the same infrastructure.

How iSCSI compares with other storage transport protocols

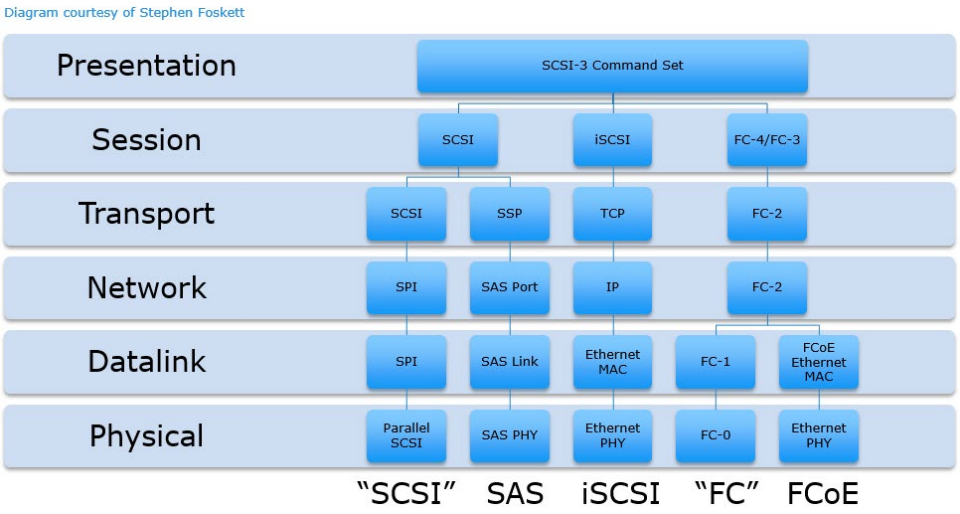


Figure 4. iSCSI and other SCSI transports

The primary storage transport protocols currently deployed in the data center today is Fibre Channel and Serial Attached SCSI (SAS) storage. With the proliferation of 10 GbE networks and movement to lower cost converged infrastructures in the data center over the last few years, iSCSI has seen a significant uptick in deployment. FCoE has seen some uptick in deployment as well in footprint but it still lags far behind FC, SAS, and iSCSI. This is primarily because FCoE requires Ethernet to be a lossless network that requires the implementation of additional technologies such as end to end Data Center Bridging (DCB). These additional requirements add cost and complexity to the Ethernet solution, greatly reducing any cost advantages that Ethernet has over traditional Fibre Channel.

The table below attempts to summarize the differences and advantages of Fibre Channel and iSCSI storage protocols. Where a protocol has an advantage is identified by the ☑ symbol.

Table 2. Fibre Channel and iSCSI comparison

	iSCSI	FC
Description	Interconnect technology that uses Ethernet and TCP/IP to transport SCSI commands between initiator and targets	Transporting protocol used to transfer SCSI command sets between initiators and targets
Architecture	Uses standard OSI-based network model—SCSI commands sent in TCP/IP packets over Ethernet	Uses its own five-layer model that starts at the physical layer and progresses through to the upper level protocols
Scalability Score	Good. No limits to the number of devices in specification but subject to vendor limitations. Larger implementations can see performance issues due to increasing number of hops, spanning tree, and other issues.	☑ Excellent. 16 million SAN devices with the use of switched fabric. Achieves linear performance profile as SAN scales outward using proper edge-core-edge fabric topologies
Performance Score	Good. Not particularly well suited for large amounts of small block IO (<=8 KB) due to TCP overhead. Requires Jumbo Frames end to end for best performance. Well suited for mixed workloads with low to mid IOPS requirements. Higher performance requires TCP offloading NICs to save CPU cycles on host and storage	☑ Excellent. Well suited for all IO types and sizes. Scales well as performance demands increase. Well suited for high IOPS environments with high throughput. No offloading required
Virtualization Capability Score	☑ Excellent. iSCSI storage can be presented directly to a virtual machine's initiator IQN by storage array	Fair to Good. FC SAN storage can be presented directly to a virtual HBA using N-Port ID Virtualization (NPIV). Note: The gen 7 specification will include integrated VM awareness and should close the gap with iSCSI in the future.

	iSCSI	FC
Investment Score	☑ Good to Excellent. Can use an existing Ethernet network; however, adding other technologies to make network lossless and to boost performance adds additional complexity and cost	Fair to Good. Initial FC infrastructure costs per port are high (although prices have declined in recent years). Other operation costs are incurred due to specialized network infrastructure. Specialized training required for administration.
IT Expertise Required Score	☑ Good. Network management teams understand Ethernet but could require some storage and IP cross-training.	Fair. Requires specialized FC networking training
Management Ease of Use Score	Fair. Can use existing network infrastructure, but host provisioning and device discovery requires ~3x the steps of Fibre Channel device provisioning. CHAP management in larger implementations can be daunting	☑ Good. Most HBAs allow for autodetection of new devices, with rescan. No host reboot required. Zoning on switch needs to be set up properly.
Security Score	Fair. Requires CHAP for authentication, VLANs or isolated physical networks for separation, IPSec for on wire encryption	☑ Excellent – specification has built in hardware level authentication and encryption, Switch port or WWPN zoning enables separation on Fabric
Strengths Summary	Cost, good performance, ease of virtualization, and pervasiveness of Ethernet networks in the data center and cloud infrastructures. Flexible feature vs. cost trade offs	High performance, scalability, enterprise-class reliability and availability. Mature ecosystem. Future ready protocol - 32 Gb FC is currently available for FC-NVMe deployments.
Weakness Summary	TCP overhead and workloads with large amounts of small block IO. CHAP, excessive host provisioning gymnastics. Questions about future - will NVMe and NVMeoF send iSCSI the way of the Dodo?	Initial investment is more expensive. Operational costs are higher as FC requires separate network infrastructure. Not well suited for virtualized or cloud-based applications.
Optimal Environments	SMBs and enterprise, departmental and remote offices. Very well suited for converged infrastructures and application consolidation. • Business applications running on top of smaller to mid-sized Oracle environments • All Microsoft Business Applications such as Exchange, SharePoint, SQL Server	Enterprise with complex SANs: high number of IOPS and throughput • Non-stop corporate backbone including mainframe • High intensity OLTP/OLAP transaction processing for Oracle, IBM DB2, Large SQL Server databases • Quick response network for imaging and data warehousing • All Microsoft Business Applications such as Exchange, SharePoint, SQL Server

The above table outlines the strengths and weaknesses of Fibre Channel vs. iSCSI when the protocols are being considered for implementation for SMB and enterprise-level SANs. Each customer has their own set of unique criteria to use in evaluating different storage interface for their environment. For most small enterprise and SMB environments looking to implement a converged, virtualized environment, the determining factors for a storage interface are upfront cost, scalability, hypervisor integration, availability, performance, and the amount of IT Expertise required to manage the environment. The above table shows that iSCSI provides a nice blend of these factors. When price to performance is compared between iSCSI and Fibre Channel, iSCSI does show itself to be compelling solution. In many data centers, particularly in the SMB space, many environments are not pushing enough IOPS to saturate even one Gbps bandwidth levels. At the time of this writing, 10 Gbps networks are becoming legacy in the data center and 25+ Gbps networks are being more commonly deployed for a network backbone. This makes iSCSI a real option for future growth and scalability as throughput demands increase.

Another reason that iSCSI is considered an excellent match for converged virtualized environments, is that iSCSI fits in extremely well with a converged network vision. Isolating iSCSI NICs on a virtualized host allows each NIC to have its own virtual switch and specific QoS settings. Virtual machines can be provisioned with iSCSI storage directly through these virtual switches, bypassing the management operating system completely and reducing I/O path overhead.

Making a sound investment in the right storage protocol is for a critical step for any organization. Understanding the strengths and weaknesses of each of the available storage protocol technologies is essential. By choosing iSCSI, a customer should feel confident that they are implementing a SAN that can meet most of their storage workload requirements at a potentially lower cost.

Deployment considerations for iSCSI

The following information needs to be considered and understood when deploying iSCSI into an environment.

Network considerations

Network design is key to making sure iSCSI works properly and delivers the expected performance in any environment. The following are best practice considerations for iSCSI networks:

- **10 GbE+ networks are essential** for enterprise production level iSCSI. Anything less than 10GbE should be relegated to test and development.
- iSCSI should be considered **a local-area technology**, not a wide-area technology, because of latency issues and security concerns.
- **Separate iSCSI traffic** from general traffic by using either separate physical networks or layer-2 VLANs. Best practice is to have a dedicated LAN for iSCSI traffic and not share the network with other network traffic. Aside from minimizing network congestion, isolating iSCSI traffic on its own physical network or VLAN is considered a must for security as iSCSI traffic is transmitted in an unencrypted format across the LAN.
- Implement **jumbo frames** (by increasing the default network MTU from 1500 to 9000) in order to deliver additional throughput especially for small block read and

write traffic. However, care must be taken if jumbo frames are to be implemented as they require all devices on the network to be jumbo frame compliant and have jumbo frames enabled. When implementing jumbo frames, set host and storage MTUs to 9000 and set switches to higher values such as 9216 (where possible).

- To minimize latency, **avoid routing iSCSI** traffic between hosts and storage arrays. Try to keep hops to a minimum. Ideally host and storage should co-exist on the same subnet and be one hop away maximum.
- **Enable "trunk mode"** on network switch ports. Many switch manufacturers will have their switch ports set using "access mode" as a default. Access mode allows for only one VLAN per port and is assuming that only the default VLAN (VLAN 0) will be used in the configuration. Once an additional VLAN is added to the default port configuration, the switch port needs to be in trunk mode, as trunk mode allows for multiple VLANs per physical port.

Multipathing and availability considerations

The following are iSCSI considerations with regard to multipathing and availability:

- Deploy **port binding** iSCSI configurations with multipathing software enabled on the host rather than use a multiple-connection-per-session (MC/S) iSCSI configuration. MC/S was created when most host operating systems did not have standard operating-system-level multipathing capabilities. MC/S is prone to command bottlenecks at higher IOPS. Also, there is inconsistent support for MC/S across vendors.
- Use the "**Round Robin (RR)**" load balancing policy for Windows host based multipathing software (MPIO) and Linux systems using DM-Multipath. Round Robin uses an automatic path selection rotating through all available paths, enabling the distribution of load across the configured paths. This path policy can help improve I/O throughput. For active/passive storage arrays, only the paths to the active controller will be used in the Round Robin policy. For active/active storage arrays, all paths will be used in the Round Robin policy.
- For Linux systems using DM-Multipath, change "path_grouping_policy" from "failover" to "multibus" in the multipath.conf file. This will allow the load to be balanced over all paths. If one fails, the load will be balanced over the remaining paths. With "failover" only a single path will be used at a time, negating any performance benefit. Ensure that all paths are active using "multipath –l" command. If paths display an "enabled" status, they are in failover mode
- Use the **"Symmetrix Optimized"** algorithm for Dell EMC PowerPath software. This is the default policy and means that administrators do not need to change or tweak configuration parameters. PowerPath selects a path for each I/O according to the load balancing and failover policy for that logical device. The best path is chosen according to the algorithm. Due to the propriety design and patents of PowerPath, the exact algorithm for this policy cannot be detailed here
- **Do not use NIC teaming** on NICs dedicated for iSCSI traffic. Use multipathing software such as native MPIO or PowerPath for path redundancy.

Resource consumption considerations

When designing an iSCSI SAN, one of the primary resource considerations must be focused around the CPU consumption required to process the iSCSI traffic throughput on both the host initiator and storage array target environments. As a guideline, network traffic processing typically consumes 1 GHz of CPU cycles for every 1 Gbps (125 MBps) of TCP throughput. It is important to note that TCP throughput can vary greatly depending on workload; however, many network design teams use this rule to get a “ballpark” idea of CPU resource sizing requirements for iSCSI traffic. The use of this rule in sizing CPU resources is best shown by an example.

Consider the following: the total throughput of an iSCSI workload is estimated to be 2.5 GBps. This means that both the host and storage environments must be sized properly from a CPU perspective to handle the estimated 2.5 GBps of iSCSI traffic. Using the general guideline, processing the 2.5 GBps of iSCSI traffic will consume:

$2.5\ GBps\ x\ \left(\frac{1000\ MBps}{1\ GBps}\right)\ x\ \left(\frac{1\ GHz}{125\ MBps}\right)$ = 20 GHz of CPU consumption

This means that an estimated 20 GHz of CPU resources will be consumed on both the host initiator and storage array target side of the environment to process the iSCSI traffic. To further examine the impact of this requirement, say the host initiator environment consists of a heavily virtualized dual node cluster. Each node has 2 x 16 core 2.5 GHz CPUs. This means that the host initiator environment has a total of:

$2\ nodes\ x\ \left(\frac{2\ CPUs}{node}\right) x\ \left(\frac{16\ cores}{CPU}\right) x\ \left(\frac{2.5\ GHz}{core}\right)$ = 160 GHz of CPU processing power

The estimated consumption of 20 GHz CPU cycles to process the 2.5 GBps of iSCSI traffic represents 12.5% of the total 160 GHz processing power of the host initiator environment. In many virtualized environments, a two-node cluster is considered a small implementation. Many modern virtualized environments will consist of many nodes, with each node being dual CPU and multi-core. In these environments, 20 GHz of CPU consumption might seem trivial; however, in heavily virtualized environments, every CPU cycle is valuable.

CPU resource conservation is even more important on the storage side of the environment as CPU resources are often more limited than on the host initiator side.

The impact on CPU resources by iSCSI traffic can be minimized by deploying the following into the iSCSI SAN environment:

- In order to fully access the environment CPU resources, **widely distribute** the iSCSI traffic across many host nodes and storage directors ports as possible.
- Employ NICs with a built-in **TCP Offload Engine (TOE)**. TOE NICs offload the processing of the datalink, network, and transport layers from the CPU and process it on the NIC itself.
- For heavily virtualized servers, use NICs that support **Single Root - IO Virtualization (SR-IOV)**. Using SR-IOV allows the guest to bypass the hypervisor and access I/O devices (including iSCSI) directly. In many instances, this can significantly reduce the server CPU cycles required to process the iSCSI I/O.

Note: The introduction of TOE and SR-IOV into an iSCSI environment can add complexity and cost. Careful analysis must be done to ensure that the additional complexity and cost is worth the increased performance.

PowerMaxOS iSCSI implementation overview

Introduction

The PowerMaxOS iSCSI target model is primarily being driven by market needs originating from the cloud or service-provider space, converged infrastructures, and heavily virtualized environments where slices of infrastructure (Compute, Network, and Storage) are assigned to different users (tenants). This model requires control and isolation of resources along with multitenancy capabilities not previously attainable with previous iSCSI implementations on previous generation of the VMAX.

Background

The implementation of iSCSI on many storage vendors closely follows the same model as FC and FCoE emulations where a user is presented a physical port linked together with a target node along with a pool of associated devices. Using masking, users can provision LUNs to individual hosts connected to this target. Besides LUN masking, this model provides almost no isolation and control of software and hardware resources on a per tenant basis. As a result, if a tenant required partial ownership of the IO stack, which is normally expected in cloud service environments, then each tenant would need to access its own physical port. In this type of situation, scalability immediately becomes a major obstacle with this design as front-end port counts on storage arrays are limited. Security and lack of network isolation are other concerns with this model, as resources (for example, volumes and authentication information) are shared among otherwise independent tenants.

PowerMaxOS iSCSI implementation design objectives

The PowerMaxOS iSCSI target model has been designed to meet customer demands regarding control and isolation of resources, as well as providing a platform for greater physical port utilization and efficiencies. The PowerMaxOS iSCSI target model accomplishes this by the following key design principles:

- PowerMaxOS groups director CPU resources (cores) together into logical pools. Each director dynamically allocates these pooled CPU resources to meet the workload demands placed upon the different types of front end and back-end connectivity options the director supports. These connectivity options and the resources they use are called “emulation instances.” PowerMaxOS supports iSCSI using the “SE instance.” A PowerMax director can have only one SE instance. The SE instance is dynamically allocated a certain number of cores, which are used to process the total amount of TCP traffic coming in through the director’s 10/25 GbE ports.
- Virtualization of the physical port. Users can create multiple iSCSI target nodes and IP interfaces for an individual port, which provides:
  - Individual iSCSI targets can be assigned one or more IP interfaces, which define access network paths for hosts to reach the target node.
  - The implementation supports configuration of routing and VLANs for traffic isolation.
- Storage side Quality of Service (QoS) is implemented at storage group (SG) level using host I/O limits and PowerMaxOS service levels.

Note: PowerMaxOS supports Ethernet PAUSE flow control; however, priority flow control (PFC) and data center bridging (DCB) are not supported.

PowrMaxOS iSCSI implementation core components

The PowerMaxOS iSCSI model achieves the design objectives by using the core components:

- A 4 x 25 GbE port or 4 x 10 GbE port interface module (required hardware), or both
- The PowerMaxOS iSCSI Target Node
- The PowerMaxOS iSCSI IP Interface
- CHAP Authentication
- IP Routing

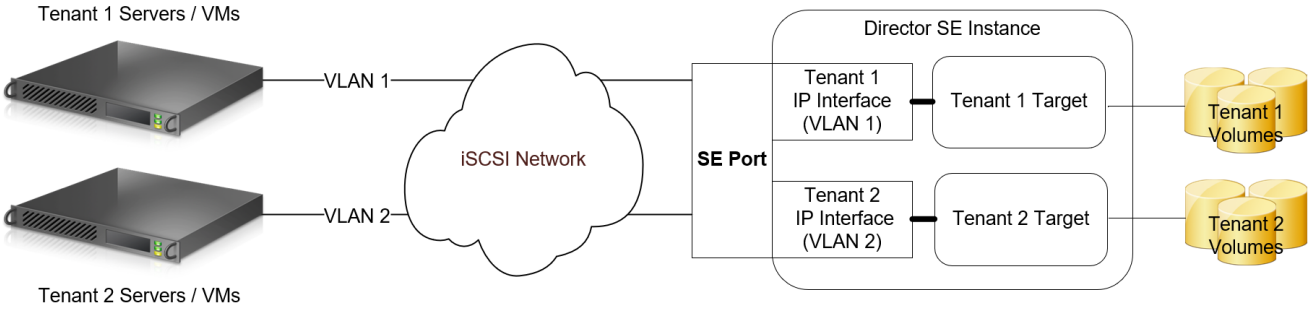


Figure 5. Typical PowerMaxOS iSCSI multitenant architecture

Supported PowerMax hardware: Quad-port 25 GbE interface module

The PowerMaxOS iSCSI target implementation uses a quad port 25 GbE hardware I/O module. This module has the following features:

- High-density quad-port 25 GbE interface (four SFP+ optical transceiver connectors)
- Support for up to four Modules/SE instance on PowerMax 2000, 3 Modules/SE instance on PowerMax 8000
- Auto Negotiation between 25 GbE to 10 GbE not supported
  - Cannot use 25 GbE and 10 GbE on 25 GbE interface module. Separate 10 GbE module required for 10 GbE
  - Can mix separate 25 GbE and 10 GbE interface modules on same director SE instance
- FRU and Hot Swappable
- Dimensions: 3” w x 1.25” h x 7.875”

Supported legacy PowerMax hardware: Quad-port 10 GbE interface module

The PowerMaxOS iSCSI target implementation uses a quad port 25 GbE hardware I/O module. This module has the following features:

- High-density quad-port 10 GbE interface (four SFP+ optical transceiver connectors)
- Support for up to four Modules/SE instance on PowerMax 2000, 3 Modules/SE instance on PowerMax 8000
- FRU and Hot Swappable
- Dimensions: 3” w x 1.25” h x 7.875”

PowerMaxOS iSCSI target node

The PowerMaxOS iSCSI target node is the backbone on how iSCSI is implemented on the PowerMax storage arrays. One can think of each PowerMaxOS iSCSI Target node as a virtual port as each physical port can have up to 64 targets. These target nodes are created and configured at the user’s discretion. An exception being the “bootstrapping target,” which can come preconfigured on a new all iSCSI PowerMax system. The bootstrapping target allows for initial target configuration on the system so that management hosts can discover and access the new storage array; however, with the introduction of Embedded Management, the user can remotely “http” to the embedded Unisphere on the storage array and create the initial iSCSI target, thus reducing the need for a factory preconfigured bootstrap target. The number of target nodes a user can configure is constrained to:

- Maximum of 64 targets per physical port
- Designed for maximum of 512 targets per director

An iSCSI Target can be in one of the two logical states: **online** or **offline**. Semantically these two states resemble the behavior of “port online” or “port offline,” where the online state indicates the Target Node is ready to accept and perform IO requests, and the latter one indicates it is not. Users will be able to control the target state through Unisphere and Solutions Enabler commands.

Most of the time (as is the case with port state) the Target Node will be in an online state. There are three common situations when the state will be offline:

- All newly created targets are in the offline state by default
- Configuration changes to the existing Target Node: All subsequent changes of Target attributes (for example, changing Target name) affect host connectivity in a detrimental way and require a specific Target to be in the offline state.
- For debugging, security, or other reasons, users may want to have a specific Target in the offline state and thus prevent any IO activity from hosts that are connected to this Target.

Another feature provided by the iSCSI target is device separation along target node lines. Devices that were previously provisioned on a per port basis are now allocated on a per Target basis. For example, if two Target Nodes are created for two different users (tenants) on the same SE instance, the new model allows for separation of devices assigned to each target. This practice cleanly isolates each user’s data from each other (true multitenancy). However, this is not a requirement, and users can technically still

assign the same device to several iSCSI Targets on the same director (even though a use case for this would be hard to justify). A top-level limitation is that the same volume cannot be assigned to more than 32 different Targets on the same director.

To create an iSCSI target on PowerMax, a user must supply:

- The physical director that is configured with the SE emulation
- A user-defined IQN (**Note**: If a user does not supply an IQN, one will be autogenerated)
- A Network ID: A Network ID is PowerMaxOS construct that is used internally by the system to associate an array IP interface with an array iSCSI target. The PowerMaxOS Network ID is specific to a single director SE emulation and is not visible to other directors or external switches and hosts. The default Network ID value is 0, and the range is 0 to 511 per director. Note: Different directors can make use the same network ID number value (NetID 10 on Dir 1F and NetID 10 on Dir 2F); however, these network IDs are unique specific to the director they reside on.
- The user can optionally specify a TCP port for the target (default is port 3260 is used if none is specified).

In the PowerMaxOS iSCSI implementation, port flags that previously needed to be set on the physical port are now set on the iSCSI target. The iSCSI target port flags are:

- SOFT_RESET
- ENVIRON_SET
- DISABLE_Q_RESET_ON_UA
- AVOID_RESET_BROADCAST
- SCSI_3
- SPC2_PROTOCOL_VERSION
- SCSI_SUPPORT1
- VOLUME_SET_ADDRESSING
- OPENVMS
- ISID_PROTECTED

Note: The SCSI_3, SPC2_PROTOCOL_VERSION, and SCSI_SUPPORT1 flags are enabled by default when a target is created on PowerMax.

Creating a PowerMaxOS iSCSI target using Solutions Enabler

Below are Solutions Enabler commands that can be used to create an iSCSI target:

```
symconfigure -sid 0536 -cmd "create iscsi_tgt dir 1E,
iqn=iqn.dellemc.0536.tenant1, network_id=80;" commit -noprompt
```

The following Solutions Enabler command creates an iSCSI target along with enabling the VOLUME_SET_ADDRESSING flag:

```
symconfigure -sid 0536 -cmd "create iscsi_tgt dir 1E,
iqn=iqn.dellemc.0536.tenant1, network_id=80,
VOLUME_SET_ADDRESSING=Enable;" commit -noprompt
```

Creating a PowerMaxOS iSCSI Target using Unisphere

To create an iSCSI target using Unisphere for VMAX, the user selects a POWERMAX or VMAX All Flash storage array; then goes to the iSCSI dashboard in “System;” and selects “Create iSCSI Target.” The create iSCSI target wizard is shown in the following screen.

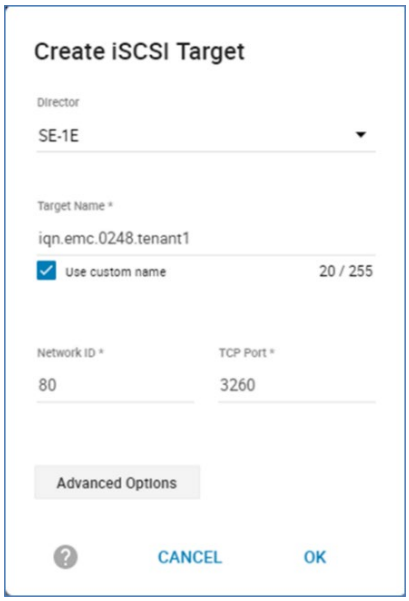


Figure 6. Creating an iSCSI target using Unisphere for PowerMax

After entering the required data, the user selects **OK** to create the target.

Note: By clicking “Advanced Options,” the user can set the port flags. The SCSI_3, SPC2_PROTOCOL_VERSION, and SCSI_SUPPORT1 flags are enabled by default when a target is created on PowerMax.

PowerMaxOS iSCSI IP interface

IP interfaces provide access to Target Nodes through one or more network paths. Similar to the iSCSI Target object, IP interfaces are managed by users, where they can create, modify, erase, and map them to an individual iSCSI Target. The number of IP interfaces a user can configure is constrained to:

- Maximum of 8 per target node
- Maximum of 64 per physical port
- Maximum of 1024 per engine (512 per director)
- A single IP Interface can be mapped to a single iSCSI target

To create an IP Interface on PowerMax, a user supplies:

- IP Address (either IPv4 or IPv6)
- Netmask for IP address provided as prefix length
- VLAN: VLAN tag information. If no VLAN is configured or not specified, the default value of zero (0) is used.
- Physical Director: Like the iSCSI target, an IP Interface can only be mapped to a single physical director using SE emulation.
- Physical Port number: The physical SE port number on the director the IP Interface is attached to.
- Network ID: A Network ID: A Network ID is PowerMaxOS construct that is used internally by the system to associate an array IP interface with an array iSCSI target. The PowerMaxOS Network ID is specific to a single director SE emulation and is not visible to other directors or external switches and hosts. The default Network ID value is 0, and the range is 0-511 per director. Note: Different directors can make use the same network ID number value (NetID 10 on Dir 1F and NetID 10 on Dir 2F); however, these network IDs are unique specific to the director they reside on.
- MTU size: This is an optional parameter that sets the maximum transmit size of Ethernet packet for the IP Interface. If not specified, the portal uses the default of 1500. To enable jumbo frames, set MTU to 9000 (maximum value allowed).

IP Interface configuration constraints:

- A single IP Interface can be mapped to only one iSCSI Target.
- Targets can make use of multiple IP interfaces (up to 8) on different SE ports on the same director; however, each IP interface must use the same Network ID as the target and must use a different subnet from the other IP interfaces.
- VLAN tag must be unique per physical SE port. VLAN tag zero implies there is no VLAN assigned.

Creating a PowerMaxOS IP interface using Solutions Enabler

Below is a Solutions Enabler command that will create an iSCSI IP Interface with an MTU size of 9000:

```
symconfigure -sid 0536 -cmd "create ip_interface dir 1E port 8,
ip_address=192.168.82.30, ip_prefix=24,network_id=80, vlanid=80,
mtu=9000;" commit -noprompt
```

Creating a PowerMaxOS IP interface using Unisphere

To create an iSCSI IP Interface using Unisphere, the user selects a storage array; then goes to the iSCSI dashboard in “System”; and selects “Create IP Interface.” The create iSCSI IP Interface wizard is shown in the screenshot below:

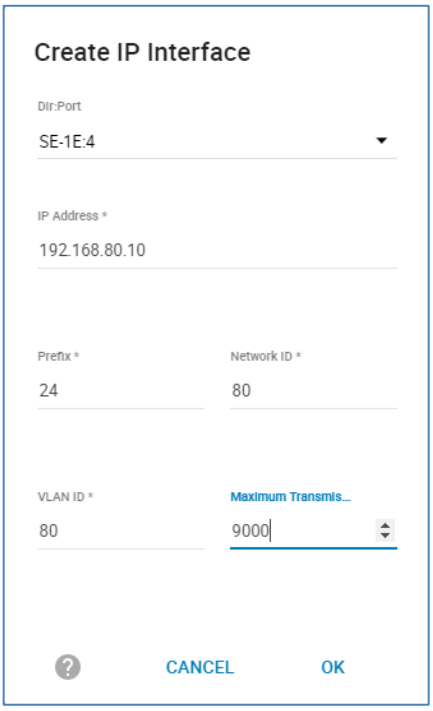


Figure 7. Creating a PowerMaxOS iSCSI IP Interface using Unisphere for VMAX

After entering the required information, the user selects OK to create the IP interface.

CHAP authentication

The PowerMaxOS iSCSI implementation supports the Challenge Handshake Authentication Protocol (CHAP) for initiators and targets. The implementation supports two types of CHAP authentication:

- One-way CHAP - In **one-way** CHAP authentication, also called **unidirectional**, the target authenticates the initiator, but the initiator does not authenticate the target.
- Two-way CHAP - In **two-way** CHAP authentication, also called **bi-directional or mutual**, an additional level of security enables the initiator to authenticate the target.

Setting one-way CHAP authentication

With CHAP one-way authentication, the storage array challenges the host initiator during the initial link negotiation process and expects to receive a valid credential and CHAP secret in response. When challenged, the host initiator transmits a CHAP credential and CHAP secret to the storage array. The storage array looks for this credential and CHAP secret, which is stored in the host initiator’s initiator group (IG) information in the ACLX database. Once a positive authentication occurs, the storage array sends an acceptance message to the host. However, if the storage array fails to find any record of the credential or secret pair, it sends a rejection message, and the link is closed.

CHAP Constraints:

- The CHAP protocol secret value is a user-defined string up to 32 ASCII characters, or 64 binary characters (binary values should be prefixed with the string “0x”) for UNIX users. Windows users need to specify a secret between 12 and 16 characters and a credential name string between 8 and 256 characters.
- Currently CHAP can only be set up using Solutions Enabler SYMCLI commands.
- The host initiator IQN must be in an initiator group prior to setting one-way CHAP as the initiator CHAP information is stored in the ACLX database for the initiator group. The initiator group does not need to be in a masking view (MV) at the time CHAP is enabled.
- Masking views are intended to provide storage isolation for specific initiators, while CHAP provides authentication.
- The use of Radius servers to store CHAP authentication data is not currently supported. This is under consideration for a future release.
- PowerMaxOS iSCSI currently does not support RADIUS for CHAP authentication. Radius support is under consideration for a future release but this support is dependent upon customer demand.

Setting iSCSI one-way CHAP authentication on PowerMax requires:

- The PowerMax storage array System ID (SID)
- The host initiator IQN
- The user-defined credential the host initiator will use to log in to the storage array with (often the host initiator IQN)
- A specific secret (password) the host needs to present to the storage array

The following SYMCLI command enables one-way CHAP for the iSCSI initiator (iqn.1991-05.com.microsoft.ENTTME0108) on the storage array:

```
symaccess -sid 0536 –iscsi iqn.1991-05.com.microsoft:ENTTME0108
set chap –cred iqn.1991-05.com.microsoft:ENTTME0108 –secret
<TargetSecret>
```

In the above “symaccess set chap” command, the –cred and –secret flags specify the credential and target password the specific host initiator (specified by the –iscsi flag) will need to send to the storage array for authentication.

On a Windows host, the specific host credential and the target secret it passes to the storage array can be found and customized using the advanced settings frame of Windows iSCSI Initiator Tool (see screenshot below).

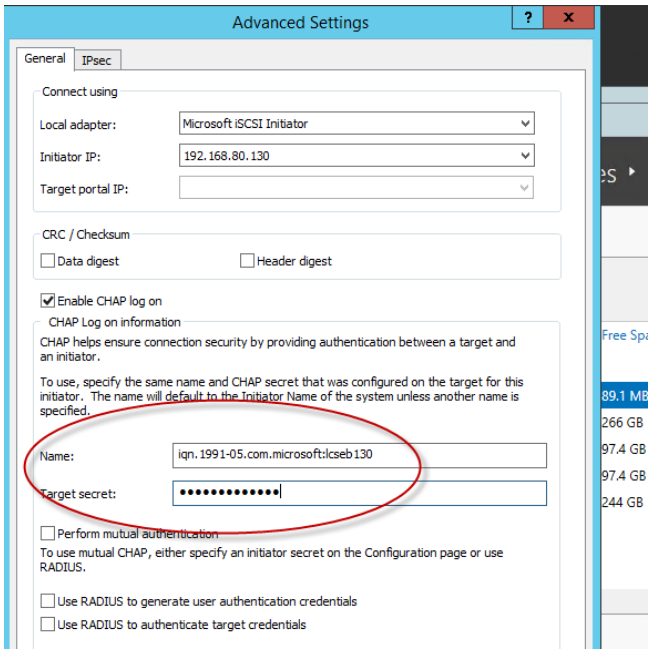


Figure 8. Customizing the host credential and target secret in the Windows iSCSI initiator tool

The values in the “Name” and “Target secret” text boxes on the advanced properties frame must match exactly the values entered in the –cred and –secret parameters used by the “symaccess set chap” command. Note that the values are case-sensitive. If there is a mismatch in either of these values, the host will not be able to authenticate on the storage array.

In the Windows iSCSI Tool, the host initiator IQN is always the default value used in the “Name” text box in the advanced settings frame. For easier management of Windows hosts on the storage array, use the Windows host IQN value for the –cred parameter in the “symaccess set chap” command. In most cases, Windows administrators will leave the default value (the host initiator IQN) in “Name” text box in advanced settings. If at some point the Windows administrator changes this value, then they must inform the storage administrator of this change as this will create a credential mismatch for the initiator on the PowerMax array. The host initiator will no longer be able to authenticate to the target and will lose access to its storage unless the “symaccess set chap” command is rerun for the initiator using the new credential value.

To examine the one-way CHAP credentials set up for the host initiator on the storage array, use the symaccess show <initiator group> command with the –detail flag using the name initiator group that the host initiator resides in:

```
symaccess -sid 0536 show ENTTME0108 -type initiator –detail
Symmetrix ID     : 000197900536
Initiator Group Name : ENTTME0108
Last update time   : 11:41:11 AM on Thu Aug 13,2015
Group last update time: 11:41:11 AM on Thu Aug 13,2015
Port Flag Overrides  : No
Consistent Lun     : No
  iSCSI Name       : iqn.1991-05.com.microsoft:ENTTME0108
  ...
```

```
Port Flag Overrides : No
CHAP Enabled      : Yes
 CHAP Credential   : iqn.1991-05.com.microsoft:ENTTME0108
Type          : iSCSI
```

In the above command, the host initiator IQN “iqn.1991-05.com.microsoft:ENTTME0108” has been previously placed into an initiator group named “ENTTME0108.” Again, this initiator group does not have to be in a masking view at the time one-way CHAP is enabled.

To disable CHAP authentications from an initiator, use the following command:

```
symaccess -sid 0536 –iscsi iqn.1991-05.com.microsoft:ENTTME0108
disable chap
```

Setting two-way CHAP authentication

Configuring two-way authentication between the host initiator and storage array iSCSI target requires the configuration of one-way authentication for the host initiator (as described in the previous section).

With two-way CHAP authentication, the host challenges and authenticates the storage array iSCSI targets also. This provides an extra layer of authentication and security in the iSCSI configuration as both the target and initiator act as authenticators and peers.

In two-way authentication, each target visible to the host must present an appropriate secret back to the host. In Windows, the initiator secret that the targets must present back to the host is set up in the Windows iSCSI Initiator tool Configuration tab as shown below:

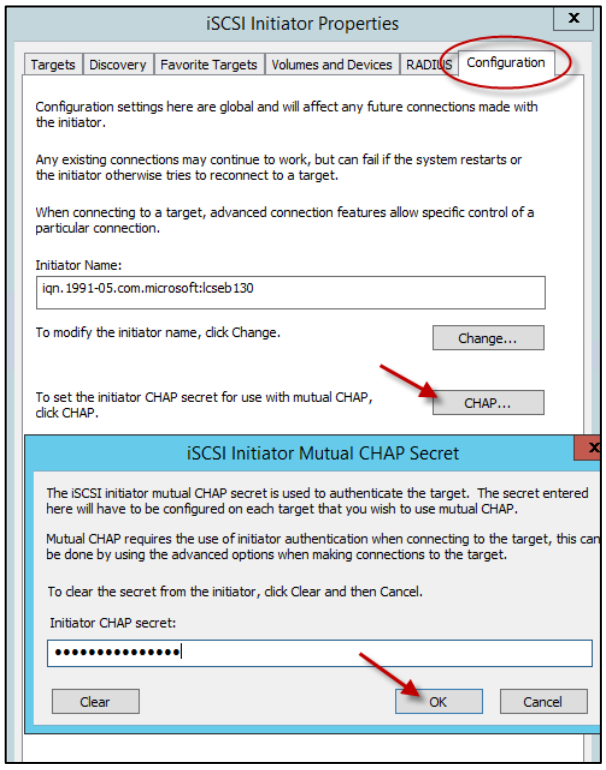


Figure 9. Setting the initiator secret using the Windows iSCSI Tool

This can also be accomplished by using the “set-IscsiChapSecret” PowerShell cmdlet on the host:

```
[LCSEB129] PS C:\ >Set-IscsiChapSecret -ChapSecret
<InitiatorCHAPSecret>
```

On the PowerMax array, two-way CHAP authentication is set up on the target using the following command:

```
symaccess -sid 0536 -iqn iqn.dellemc.0536.1F.prod1 set chap -cred
iqn.dellemc.0536.1F.prod1 -secret <InitiatorCHAPSecret>
```

In the above command, the IQN of the PowerMax iSCSI target that will be authenticated by the host initiator is the value used in the –iqn parameter. The IQN of the PowerMax iSCSI target is the value used in the –cred parameter (how the target presents itself to the host initiator in discovery). The secret that the target needs to present to the host initiator (as specified in the Windows iSCSI Tool Configuration tab) is the value used in the –secret parameter. If storage is to be presented to a host initiator through multiple PowerMax iSCSI targets, then the above command will need to be run for each target that will present itself to the host in order for successful two-way CHAP authentication.

Two-way CHAP authentication can also be set using the PowerMax iSCSI target’s associated director and virtual port combination as follows:

```
symaccess -sid 0536 –iscsi_dirport 1e:0 set chap -cred
iqn.dellemc.0536.1F.prod1 -secret <InitiatorCHAPSecret>
```

In the above command, the –iqn parameter has been replaced with the –iscsi_dirport parameter. A storage array iSCSI target’s associated director and virtual port can found using the following “symcfg” command:

```
symcfg -sid 0536 list -se all -iscsi_tgt

Symmetrix ID: 000197900536 (Local)
Dir:P  NetId Status IQN
------- ----- ------- ------------------------------
01E:000  80 Online iqn.dellemc.0536.1F.prod1
02E:000  81 Online iqn.dellemc.0536.2F.prod1
```

To examine two-way CHAP authentication set up on the PowerMax array, run the following symaccess command:

```
symaccess -sid 0536 list chap

Symmetrix ID      : 000197900536

Director Identification : SE-1F
Director Port      : 000
iSCSI Target Name    :
Protocol        : CHAP

Identifier        Type  State   Credential
------------------------ ----- -------- ------------------------
SE-1F:000         N/A  ENABLED  iqn.dellemc.0536.1F.prod1
```

To delete CHAP from a specific PowerMaxOS iSCSI target, use the following command:

```
symaccess -sid 0536 -iqn iqn.dellemc.0536.1F.prod1 delete chap
```

Routing instance

In many implementations, flat or single hop SAN networks are not possible, and the storage traffic will sometimes need to span across multiple subnets. For example, a host network might be on 10.240.180.xxx network while the storage might be on the 10.245.200.xxx network. In these cases, the PowerMaxOS iSCSI model must be able to properly route the iSCSI traffic across the different subnets being used in the environment. It does this by using an object called the routing instance. The routing instance object basically points the iSCSI traffic for a specific IP Interface IP Address (or group of IP addresses) used by a specific Network ID on a director to a specific gateway in which the iSCSI traffic is then forwarded on to other networks.

A PowerMaxOS routing instance is associated with a specific network ID on a single director. A user can create a maximum of 1024 routing instances per director. When creating a PowerMaxOS routing a user will need to specify:

- The director number
- IP address of default gateway
- Subnet Mask (prefix)
- Network ID number
- PowerMaxOS IP interface IP address

Creating a PowerMaxOS iSCSI IP route using Solutions Enabler

A user can specify an IP route for a specific IP address on a director by the following Solutions Enabler SYMCLI command:

```
symconfigure -sid 0536 -cmd "add ip_route dir 1F,
ip_address=0.0.0.0, ip_prefix=0, gateway=192.168.82.1,
network_id=10;" commit -nop
```

The above Solutions Enabler command will create a “catch all” routing instance that uses a default gateway of 192.168.82.1 for all IP interface IP address (0.0.0.0) and all subnets (0) using Network ID 10 on director 1F.

Note: Subnet mask 0.0.0.0/0 signifies all address visible on the network. In traditional networking best practices, the use of this subnet is discouraged because of the confusion in having a network and subnet with indistinguishable addresses. However, in networks with a few IP addresses, it can function as a useful “catch all” subnet to allow for broadcast to all visible IP address and subnets.

Creating a PowerMaxOS iSCSI IP route using Unisphere

A user can specify an IP route for a specific IP address on a director by the following Solutions Enabler SYMCLI command:

To create an iSCSI IP Interface using Unisphere, the user selects an array; then goes to the iSCSI dashboard in “System”; and selects “Add IP Route.” The Add iSCSI IP Route wizard is shown in the screen below.

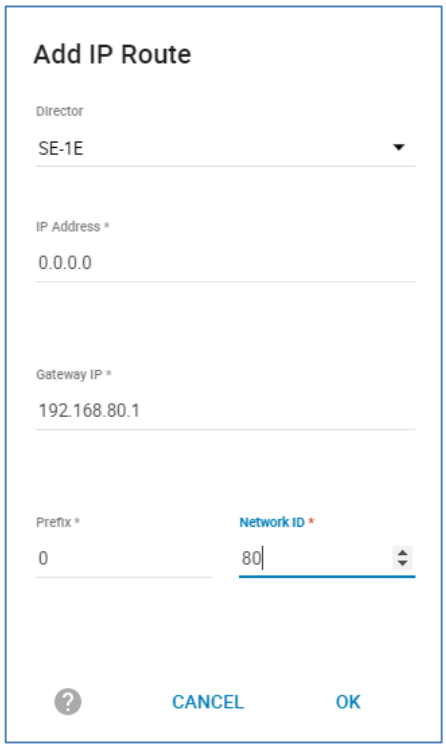


Figure 10. Creating a PowerMaxOS routing instance using Unisphere

After entering the required information, the user selects **OK** to create the IP route.

PowerMaxOS iSCSI host connectivity limits

The following table summarizes the host connectivity limits for PowerMaxOS iSCSI:

Table 3. PowerMaxOS iSCSI host connectivity limits

Component	Maximum values			
	Per SE port	Per SE instance	Per director	Per engine
VLANs	64	512	512	1024
SE Instance	NA	NA	1	2
Physical SE Ports	NA	16	16	32(1)
Network IDs	NA	512	512	1024
Routing Instances	NA	1024	1024	2048
IP Interfaces	64	512	512	1024
iSCSI Targets	64	512	512	1024
Host Connections	2048	8192	8192(2)	16384

(1) Maximums of 32 SE ports on the VMAX 250F and PowerMax 2000 and 24 ports on the VMAX 950F and PowerMax 8000

(2) The director host connection limit is a total that includes hosts using other front-end instances (Fibre Channel) that could be configured on the director.

Summary

The Dell EMC iSCSI implementation on PowerMaxOS based storage arrays provides a viable, lower-cost connectivity method for customers who are looking at alternatives to Fibre Channel. The Dell EMC PowerMaxOS iSCSI model is architected to support true multitenancy and other needs being driven by the market. The model is a good fit in the cloud or service-provider space, converging infrastructures, and heavily virtualized environments where slices of infrastructure (Compute, Network, and Storage) are assigned to different users (tenants).

PowerMax iSCSI use cases

Introduction

This section will provide a brief overview of the more common iSCSI implementations used by PowerMax customers. A step-by-step guide on how to implement these use cases using Unisphere for PowerMax and Solutions Enabler will be provided in the next section of this document.

Example 1: Basic port binding

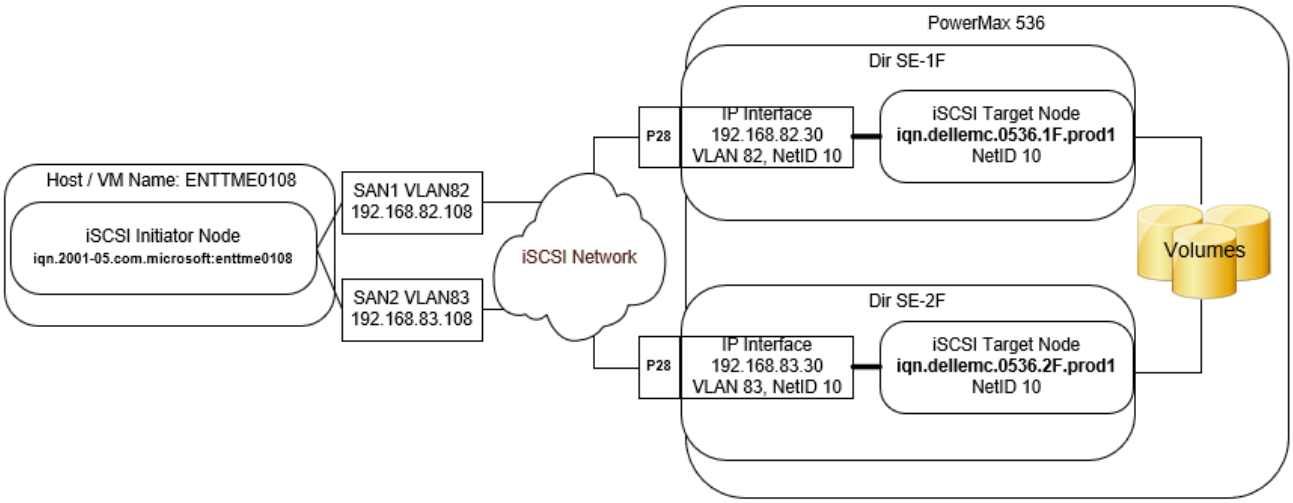


Figure 11. Example 1: Basic Port Binding Configuration

X The 1/1/1 configuration is a commonly used configuration by PowerMax iSCSI customers. Even though a PowerMax iSCSI target can make use of multiple IP interfaces on different ports (on the same director), many customers prefer to the more simplistic approach of having a single target and IP interface per SE port on the array. In order to enable multipathing, multiple iSCSI targets are placed into the port group of the masking view created for the host that has multipathing software (PowerPath/Multipath IO) installed.

A multipathing enabled masking view for the above configuration would include the following components:

- Storage group (SG): Volumes for application
- Port Group (PG): Two Target Node IQNs (iqn.dellemc.0536.1F.prod1 on Dir 1E, iqn.dellemc.0536.2F.prod1 on Dir 2E)
- Initiator Group (IG): Host/VM initiator IQN (iqn.2001-5.com.microsoft:enttme0108)

Note: In the above configuration example, the IP interfaces for the two targets use unique VLANs (VLAN 82 and VLAN 83). In many production implementations, only a single VLAN will be used for the entire iSCSI SAN. The use of a single VLAN for an iSCSI environment is shown in the next implementation example.

Example 2: PowerMaxOS iSCSI multitenancy or port consolidation

As network port speeds increase to 25 GbE and beyond, Ethernet implementations are moving towards port consolidation as the larger port speeds allow network administrators to consolidate different workloads onto “fewer but bigger” ports in the environment. This port count reduction and workload consolidation results in a reduction CAPEX and OPEX costs as power consumption, cabling, and overall management costs can be dramatically lowered.

The following configuration example expands upon the basic PowerMaxOS iSCSI Target Model by introducing the concepts of storage multitenancy and port consolidation. In this second example, a second storage environment (Prod2) is added to original Prod1 environment from example 1. The Prod2 environment uses two unique targets, each with its own IP Interface, which are sharing the same ports used by the Prod1 environment.

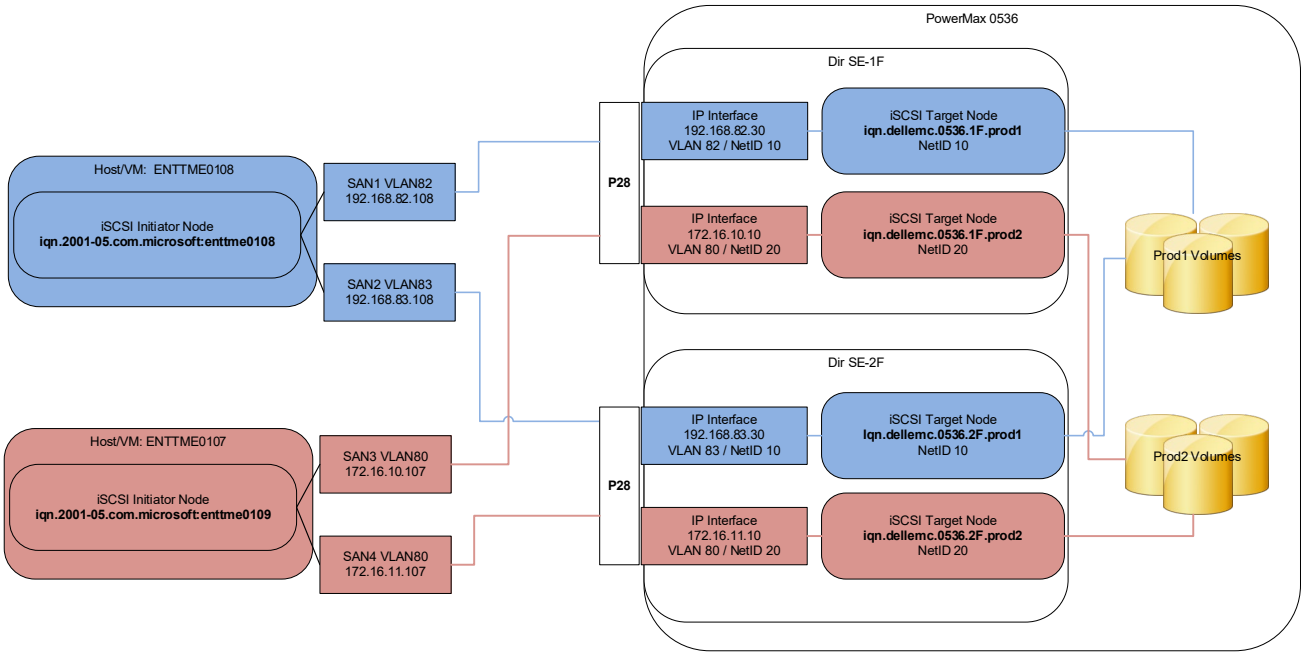


Figure 12. Example 2: PowerMaxOS iSCSI multitenant environment

In example 2, Prod2 uses completely different IP subnets and its own unique single VLAN (VLAN 82). The use of a unique separate VLAN for the Prod2 environment achieves storage network isolation from the Prod1 environment on the shared PowerMax SE ports. Volumes designated for the Prod1 environment are provisioned through the Prod1 targets, while volumes designated for the Prod2 environment are provisioned through the Prod2 targets. Because of the VLAN capability provided by the PowerMaxOS iSCSI target model, the Prod1 and Prod2 storage volumes can be accessed through the same port, while still being isolated from each other. This configuration example shows how the PowerMaxOS iSCSI target model allows for more efficient use resources in a multitenant environment.

A multipathing enabled masking view for the above Prod2 configuration would include the following components:

- Storage group (SG): Volumes used by Prod2 applications
- Port Group (PG): Two Target Node IQNs (**iqn.dellemc.0536.1F.prod2** on Dir 1E**, iqn.dellemc.0536.2F.prod2** on Dir 2E)
- Initiator Group (IG): Host/VM initiator IQN (**iqn.2001-5.com.microsoft:enttme0107**)

Implementing example 1: iSCSI port binding

Introduction

This section will demonstrate how to set up the PowerMax iSCSI port binding configuration described in example 1 in the previous section. This implementation example will demonstrate how to use the iSCSI configuration wizard in Unisphere for PowerMax to set up the complete Prod1 environment in three easy steps. It will also demonstrate how to use Solutions Enabler commands to accomplish the same task.

This implementation example will be broken down into the following parts:

- Optional: Document the current and desired environment
- Identify which SE ports are online and available for use on the array
- Use the iSCSI Configuration Wizard in Unisphere for PowerMax and Solutions Enabler commands to:
  - Create of initial Prod1 targets
  - Create of initial Prod1 IP interfaces
  - Attach IP interfaces to targets and enable targets
- Verify that the Prod1 IP interfaces can “ping” the remote host IP addresses
- Connectivity troubleshooting tips
- Create an iSCSI masking view using the Prod1 host IQN in the initiator group and use the Prod1 target IQNs in the port group.
- Optional: During the creation of the masking view using Solutions Enabler, One-Way CHAP will be set up for the Prod1 host initiator IQN.
- Acquiring the newly provisioned iSCSI PowerMax storage on the host. Format the devices and send IO.

Note: This example will use a host or virtual machine that is running Windows Server 2016 and PowerPath. It will demonstrate use of Windows PowerShell and the Windows Server Manager GUI. For detailed information about setting up iSCSI on RedHat Linux, go to the following link: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_storage_devices/getting-started-with-iscsi_managing-storage-devices

Document the current and desired configuration

As a best practice, it is good to create tables and diagrams like the ones shown in this section as it helps a storage administrator keep track of the components and relationships used in the PowerMax iSCSI environment. Although this is an optional step, detailed documentation greatly helps in management and in communicating the environment details to other teams such as the Networking and Database Administrators.

A table such as the following details the PowerMax parameters and values that comprise the Prod1 environment used in the example.

Configuration	PowerMax ID	iSCSI Director	Port	iSCSI Target Name	IP Interface IP Address	Prefix	Network ID	VLAN ID	MTU
Prod1	197900536	SE-1F	28	iqn.dellemc.0536.1F.prod1	192.168.82.30	24	10	82	9000
Prod1	197900536	SE-2F	28	iqn.dellemc.0536.2F.prod1	192.168.83.30	24	10	83	9000

The Prod1 environment in this example uses a PowerMax that has two SE directors (1F and 2F). The example shows both directors using a single physical port (port 28). The Prod1 environment will use two storage array iSCSI targets (iqn.dellemc.0536.1F.prod1 and iqn.dellemc.0536.2F.prod1) attached to two IP Interfaces using the IP address of 192.168.82.30 and 192.168.83.30. The Prod1 IP interfaces and targets use two VLANs (82-83) for SAN1 and SAN2. The use of VLANs require that they be set up previously on the network infrastructure by the Networking Team. Other details that are important to document are the switches and switch ports are being used; cable/trunk identifiers; and host information such as host IQNs and CHAP details.

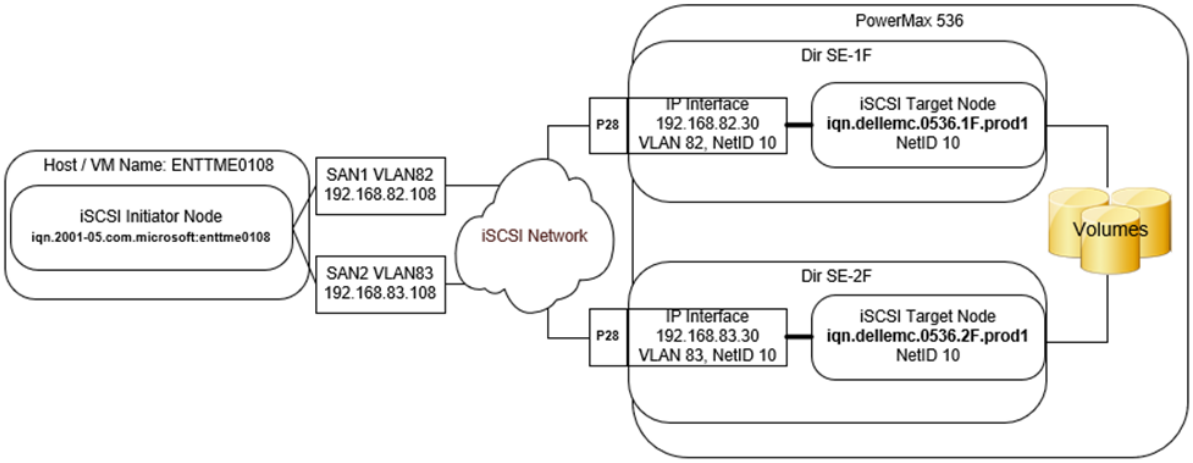


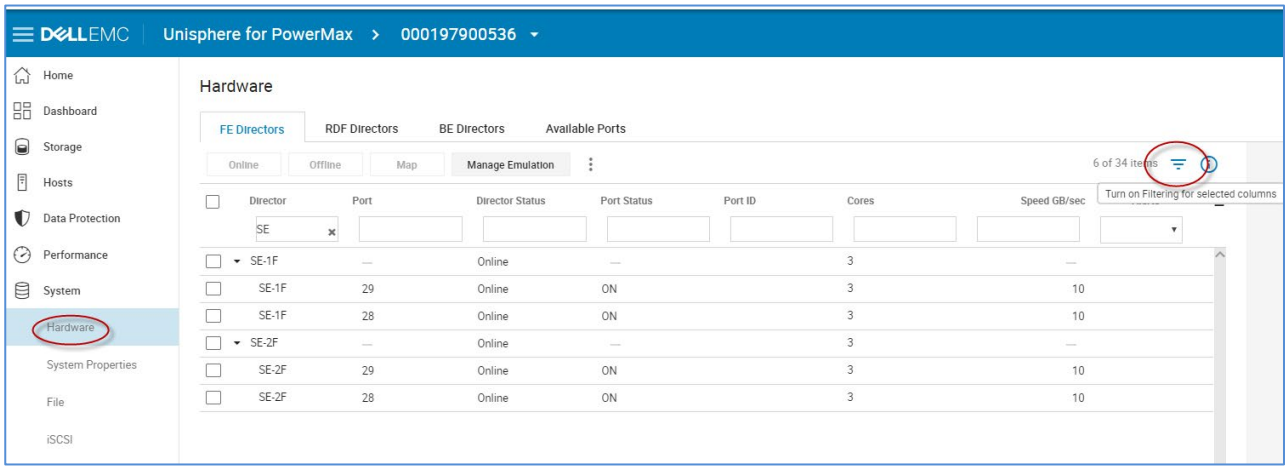
Figure 13. Completed Prod1 Environment Diagram

The above diagram shows how the completed Prod1 environment will look when finished. The components in the diagram correspond with the values shown in the previous Prod1 parameter table.

Identify all online PowerMax SE ports

The first step in the creating the initial iSCSI targets and IP Interfaces is to identify all of the online SE director ports on the PowerMax storage array.

Using Unisphere for PowerMax



In the above Unisphere output, it is shown that there are four online SE ports configured on the array—ports 28 and 29 on director 1F and ports 28 and 29 on director 2F. This example will use SE ports 1F:28 and 2F:28.

Using Solutions Enabler

Viewing and identifying the online SE ports can be done in Solutions Enabler using the “symcfg -sid ### list -se all -port -detail” command.

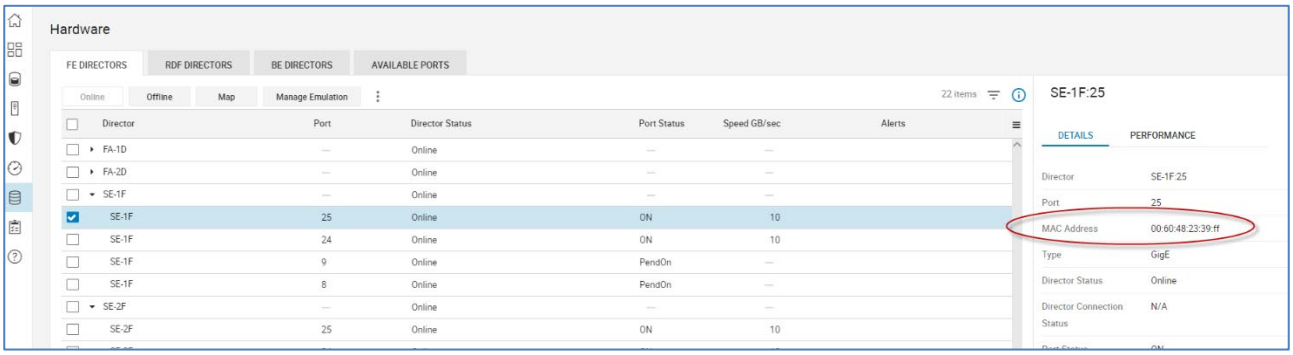
```
PS C:\> symcfg -sid 0536 list -se all -port -detail

Symmetrix ID: 000197900536 (Local)
     S Y M M E T R I X  D I R E C T O R  P O R T S

                                        Flags Speed
 Ident Port WWN         MAC Address     Type      ASCR  Gb/sec Status
 ----- ---- ---------------- ----------------- ------------ -----
------ -------
 SE-1F  28 N/A        00:60:48:23:6c:56 GigE      ..--     10 Online
 SE-1F  29 N/A        00:60:48:23:6c:57 GigE      ..--     10 Online
 SE-2F  28 N/A        00:60:48:22:fb:7c GigE      ..--     10 Online
 SE-2F  29 N/A        00:60:48:22:fb:7d GigE      ..--     10 Online
```

```
Legend:
 Flags:
   (A)CLX Enabled        : X = True, . = False, - = N/A
   (S)how ACLX device Enabled : X = True, . = False, - = N/A
   (C)HAP Enabled        : X = True, . = False, - = N/A
   (R)ADIUS Enabled       : X = True, . = False, - = N/A
```

The output above shows that four ports total (ports 28 and 29 on both iSCSI directors 1F and 2F) have been configured for iSCSI (SE emulation) and are online.

Note: When a port is in a pending online state, this typically means that the director port is active but is most likely not cabled yet.

One of the key troubleshooting enhancements added to the Unisphere and Solutions Enabler SE port listing output is the SE port's MAC Address. This greatly helps when trying to troubleshoot iSCSI connectivity issues between the host and array as the storage administrator can give the network team the SE port MAC addresses to verify if and what switch ports the SE ports are logging into. The SE port MAC address is shown in the above Solutions Enabler "symcfg -sid ### list -se all -port -detail" command and starting in Unisphere V9.2 by selecting the Array → Hardware → FE Directors tab and selecting a SE port. The MAC address for the selected port will appear in details window.



Create the Prod1 iSCSI configuration

Configuration	PowerMax ID	iSCSI Director	Port	iSCSI Target Name	IP Interface IP Address	Prefix	Network ID	VLAN ID	MTU
Prod1	197900536	SE-1F	28	iqn.dellemc.0536.1F.prod1	192.168.82.30	24	10	82	9000
Prod1	197900536	SE-2F	28	iqn.dellemc.0536.2F.prod1	192.168.83.30	24	10	83	9000

This section demonstrates the following steps:

- Create the iSCSI Targets
- Create the IP Interfaces
- Attach the IP Interfaces to the iSCSI Targets
- Enable the Targets

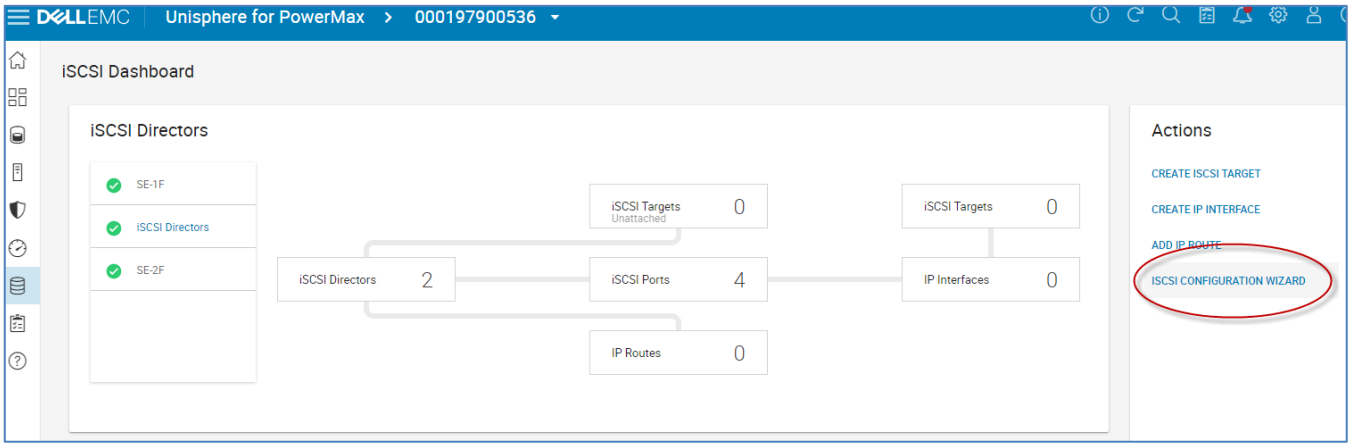
These steps will be demonstrated using the iSCSI Configuration Wizard in Unisphere for PowerMax and through the various Solutions Enabler commands.

Option 1: Using the iSCSI Configuration Wizard

This section will show how to build the iSCSI configuration using the iSCSI Configuration Wizard in Unisphere for PowerMax.

Step 1: Open the iSCSI Configuration Wizard

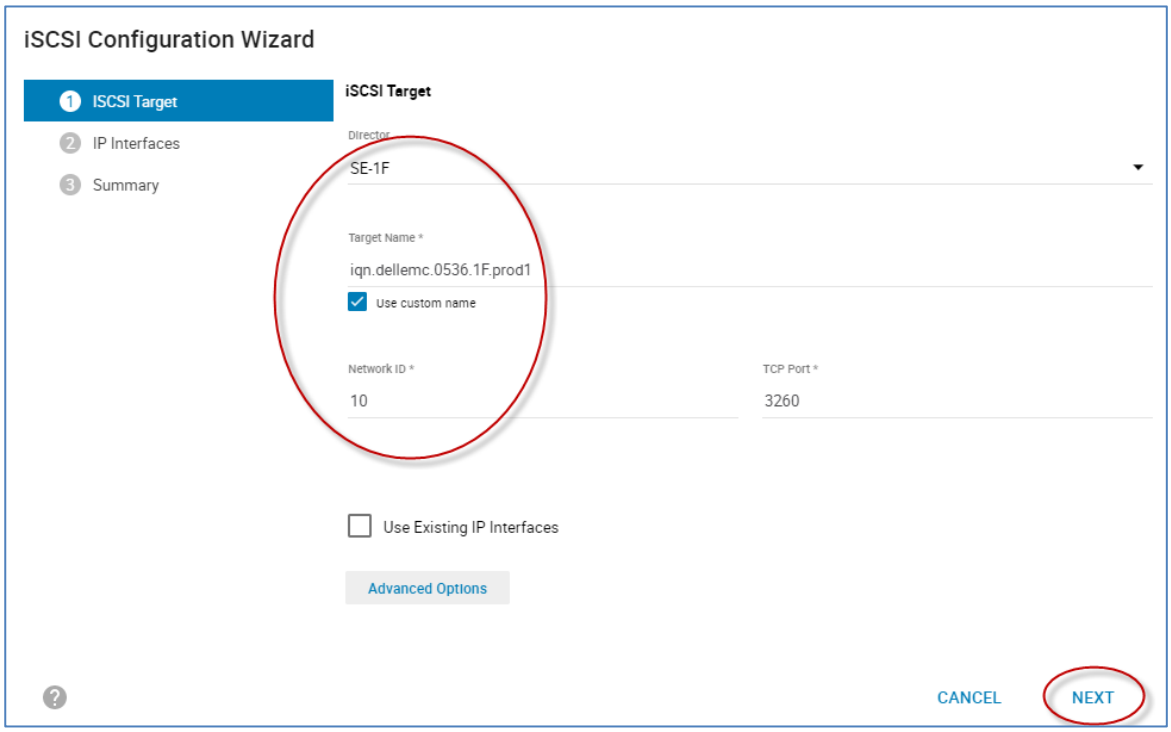
To access the wizard, select the PowerMax Array → System → iSCSI → iSCSI Configuration Wizard.



Step 2: Enter the first target information

Enter iSCSI Target information (director number, custom name, network id, leave defaults for TCP port and advanced options). In the example, director 1F is selected, a custom name is entered, and a Network ID of 10 is used (value of 10 chosen because it is a nice round number and easy to remember). In the example, all other defaults for TCP Port and Advanced Options are selected. “Next” is then clicked.

Author’s comments about iSCSI target naming: In the wizard, a user has two naming options when creating a target. One option is to use a custom name as done in the example, which is **iqn.dellemc.0536.1F.prod1,** or let the system create a unique name, which is often in the form **iqn.1992-04.com.emc:600009700bcbb8f83651012c00000006**. Each custom target name must begin with “iqn.” While letting the system generate a target name is quick and easy, the system-generated name can be non-intuitive (IMHO), making it difficult to interpret when many targets are created on the system or present in the environment. Using a custom name allows for some naming standards to be implemented, which can be much more intuitive when troubleshooting the environment. The target naming standard used in this example is:
iqn.dellemc.<SID###>.<director>.<environment name>

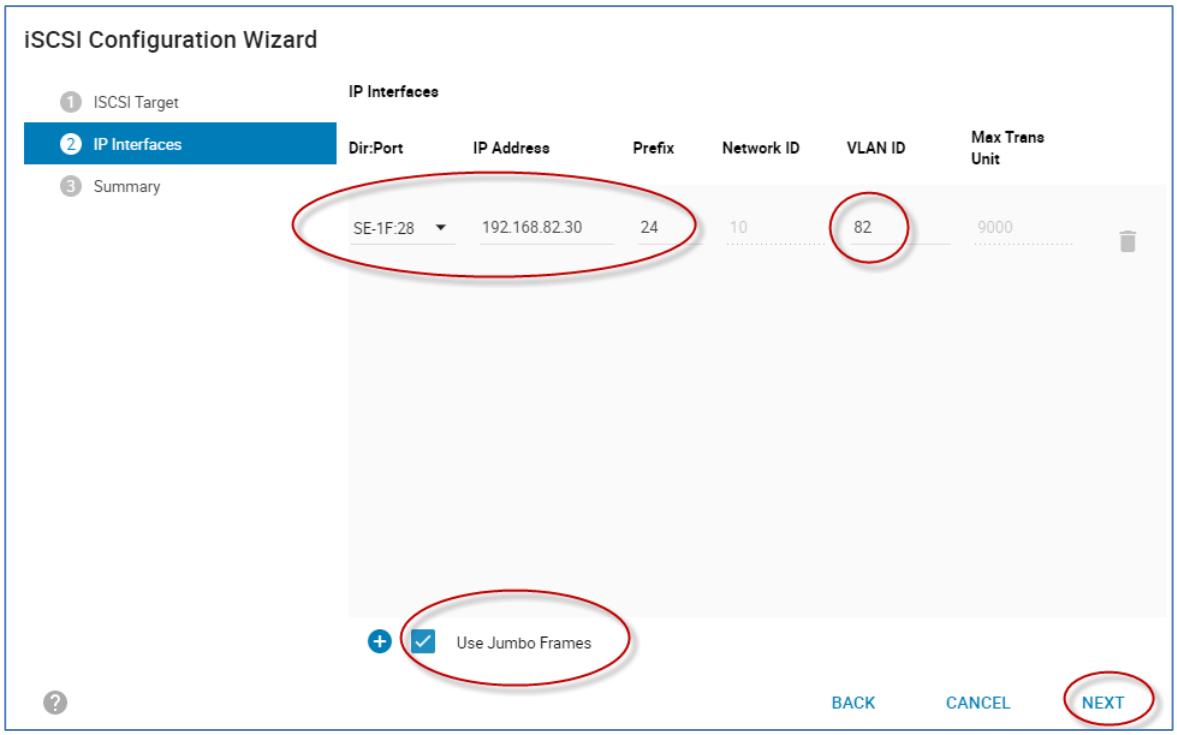


Step 3: Enter the first IP interface information

Once the target information is entered and “next” is selected, the wizard prompts for the entering of the associated IP Interface information such as the director port being used (1F:28), the IP address to be used (192.168.82.30), Subnet Prefix (24), and VLAN ID being used (82). Note that the wizard automatically selects the same Network ID (10) that was used when entering the target information. This is carried over from the previous screen as the target and its associated IP Interface must use the same Network ID.

One thing selected in the example that is not a default is “Use Jumbo Frames.” Although the use of Jumbo Frames is with iSCSI is considered a best practice, it needs to be implemented end to end from the host to the switch to the array. The use of Jumbo Frames in the environment requires coordination with the Network Team so in Unisphere “Use Jumbo Frames” is left as cleared as a default. A storage administrator can enable this when it is known that Jumbo Frames is enabled in the environment.

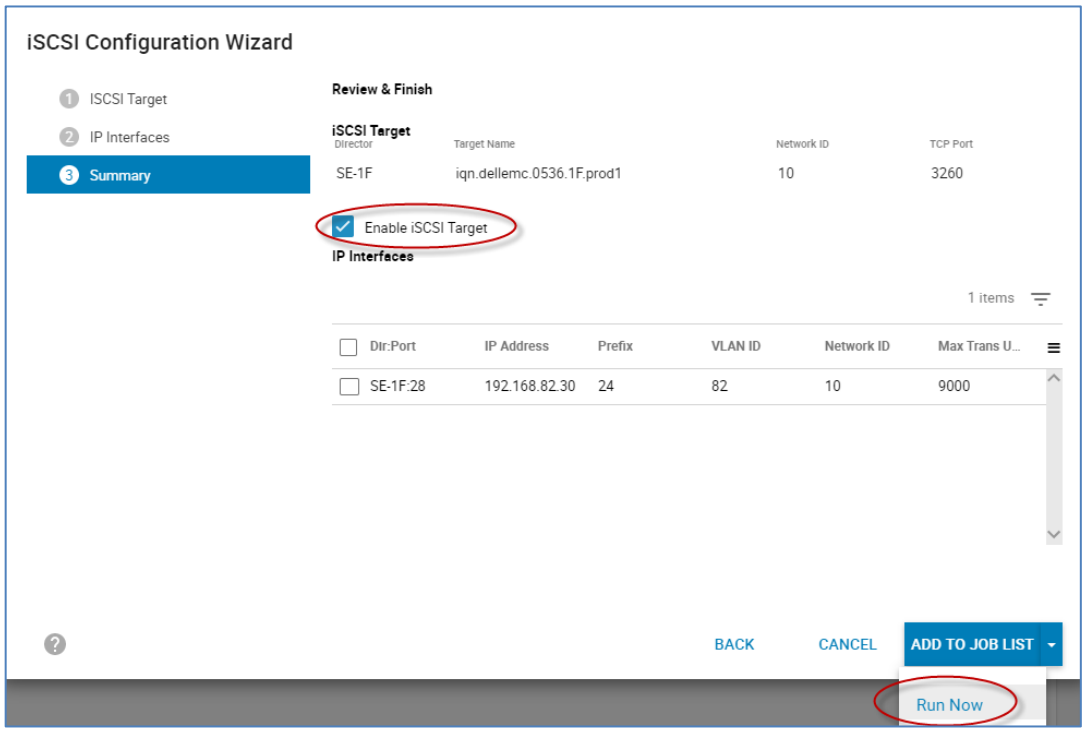
Once the IP Interface information is entered, select “Next.”



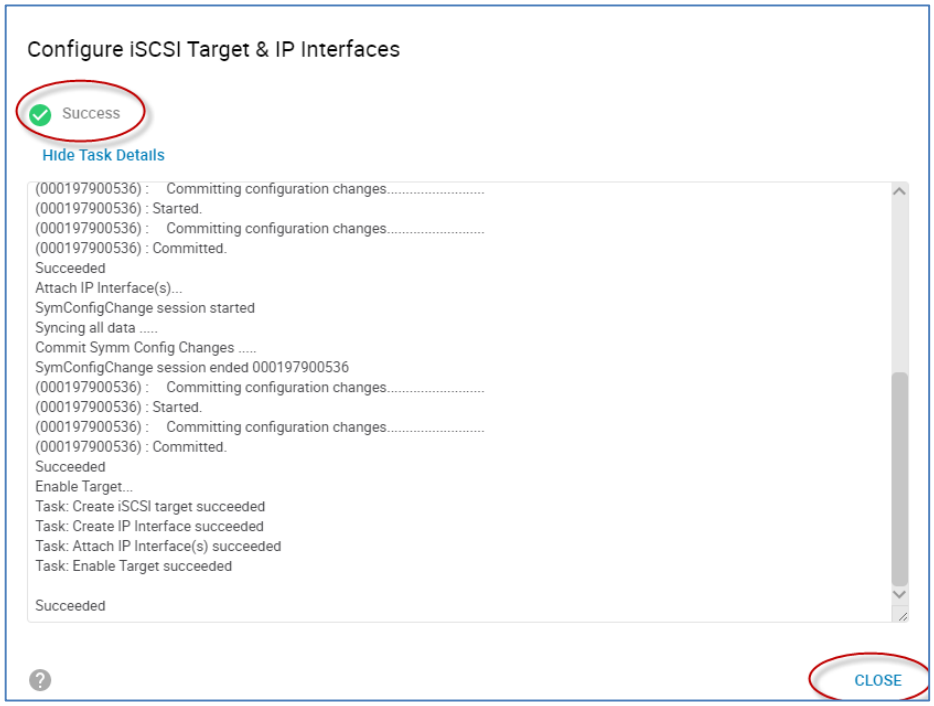
Step 4: Review the summary information and create the first target and IP interface

Once all the target and associated IP Interface information is entered, the wizard presents a summary screen of the configuration it will build. Review the information and select “Back” if any information needs to be updated or corrected. In the summary screen, there is a selectable option to “Enable iSCSI Target.” When iSCSI targets are created, the default option is for then to be disabled after the creation. This is done to allow the storage administrator the option to first create targets then enable later when the configuration and environment is ready. In the example, and in most customer implementations, the “Enable iSCSI Target” is selected. This saves the extra step of having to enable the target after its creation.

After reviewing the information and selecting “Enable iSCSI Target,” select “Run Now.” This will launch a batch operation will create the target.

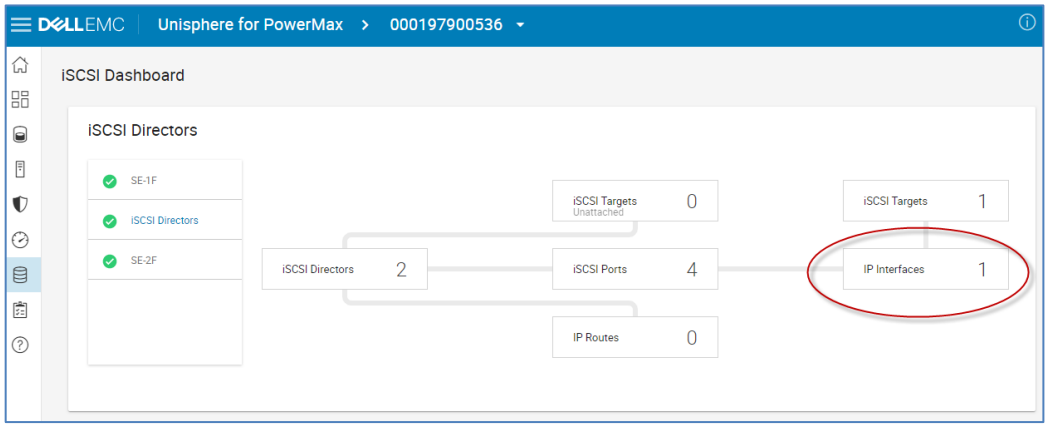


As the batch operation proceeds, monitor the status and verify that it completes successfully. Press “Close” when done.

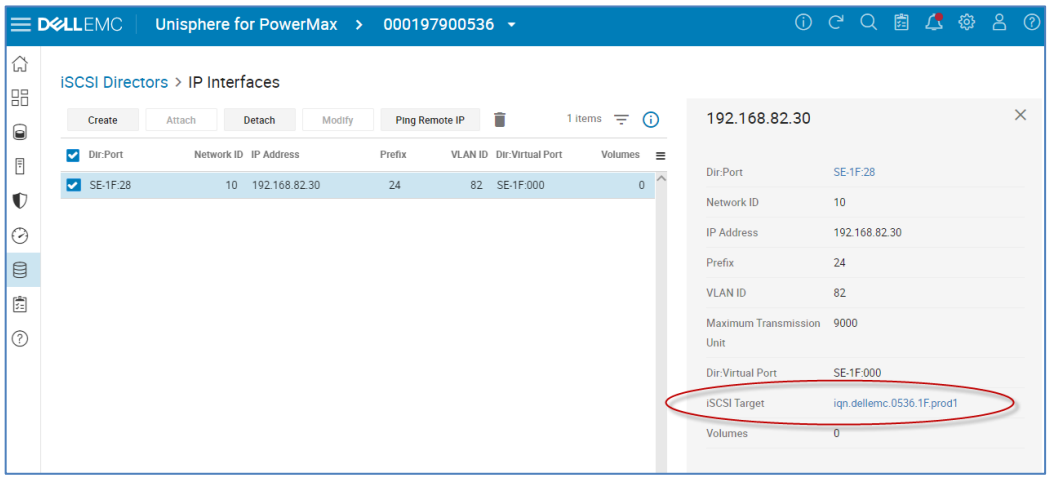


Step 5: Step 5: Optional: Examine the newly created iSCSI Target and IP Interface details

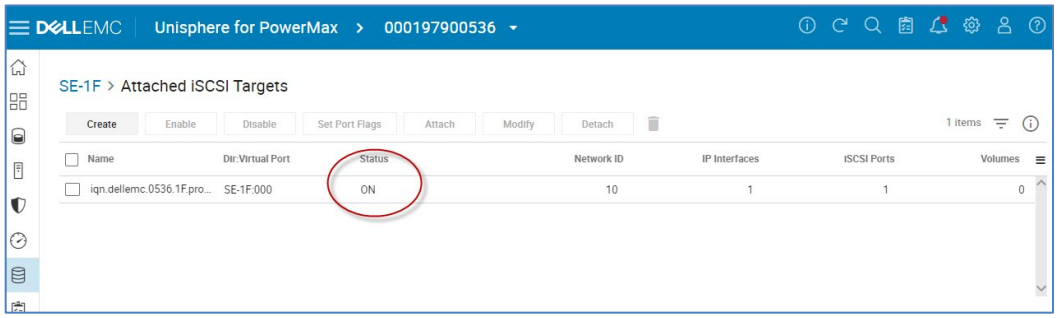
After the first iSCSI target and its IP Interface are created, go to the iSCSI dashboard (select the PowerMax Array → System → iSCSI) and double-click on “IP Interfaces.”



Examine the details of the newly created IP Interface and then double-click its associated iSCSI target.

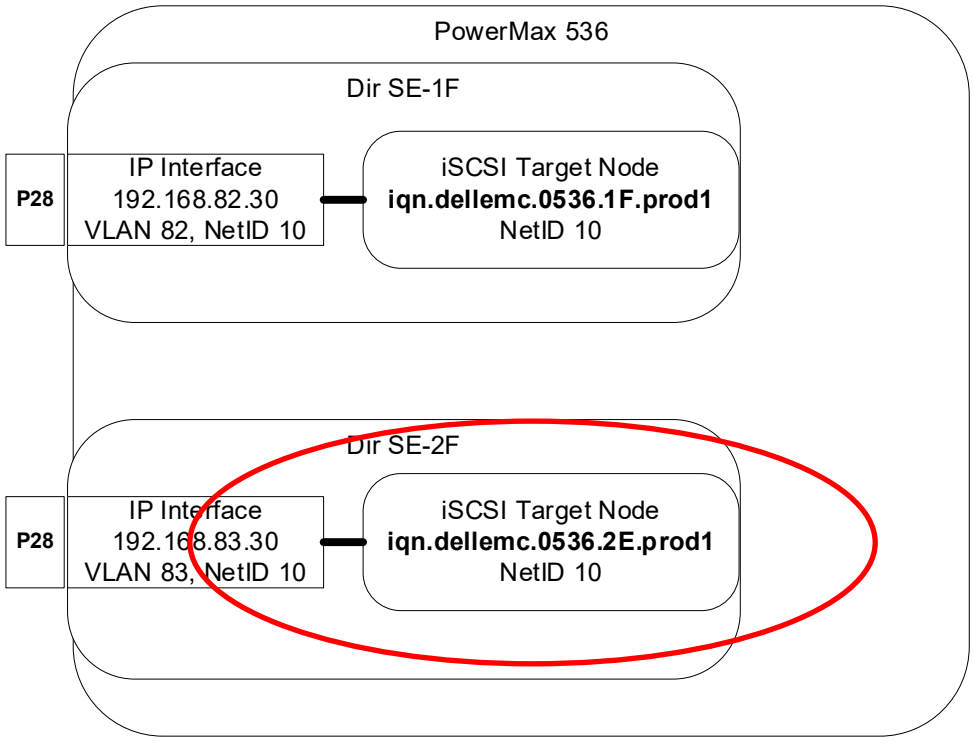


Examine the details of the associated iSCSI target, and verify that i’s status is “On” (enabled).

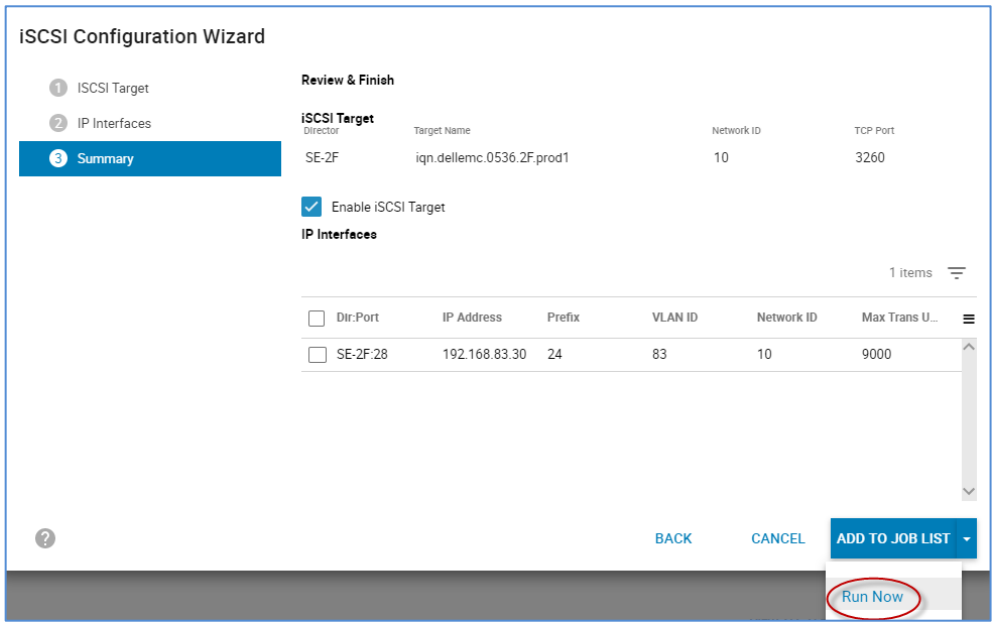


Step 6: Repeat steps 1 to 5 to create the second iSCSI target and IP interface

Using the iSCSI Configuration Wizard, repeat the previous steps (1 – 5) to create the second iSCSI target and IP interface for the Prod1 environment. The following diagram illustrates the values to use for the second iSCSI target and IP interface.



Once the values are entered in the wizard for the second iSCSI target and IP interface, review the summary information and then select “Run Now” to create the components. As before, confirm that the operation completes successfully.



Option 2: Using Solutions Enabler

This section will assume that the Prod1 environment (iSCSI Targets and IP Interfaces) have not been built using the Unisphere iSCSI Configuration Wizard discussed in the previous section. Some of the information presented in this section such as PowerMax iSCSI target naming will be a repeat from the previous section.

As stated previously, the example's PowerMax Prod1 environment iSCSI components will use the following values:

Configuration	PowerMax ID	iSCSI Director	Port	iSCSI Target Name	IP Interface IP Address	Prefix	Network ID	VLAN ID	MTU
Prod1	197900536	SE-1F	28	iqn.dellemc.0536.1F.prod1	192.168.82.30	24	10	82	9000
Prod1	197900536	SE-2F	28	iqn.dellemc.0536.2F.prod1	192.168.83.30	24	10	83	9000

Step 1: Create the IP Interfaces for the Prod1 Environment

To create this example with two IP Interfaces using the previously specified parameters, use the following SYMCLI symconfigure commands

```
PS C:\> symconfigure -sid 0536 -cmd "create ip_interface dir 1F
port 28, ip_address=192.168.82.30, ip_prefix=24,network_id=10,
vlanid=82, mtu=9000;" commit -noprompt

A Configuration Change operation is in progress. Please wait...
  Establishing a configuration change
session..............Established.
  Processing symmetrix 000197900536
  Performing Access
checks.................................Allowed.
  Checking Device
Reservations............................Allowed.
  Committing configuration
changes.......................Started.
  Committing configuration
changes.......................Committed.
  Terminating the configuration change session.............Done.
The configuration change session has successfully completed.

PS C:\> symconfigure -sid 0536 -cmd "create ip_interface dir 2F
port 28, ip_address=192.168.83.30, ip_prefix=24,network_id=10,
vlanid=83, mtu=9000;" commit -noprompt

A Configuration Change operation is in progress. Please wait...
  Establishing a configuration change
session..............Established.
  Processing symmetrix 0001979000536
  Performing Access
checks.................................Allowed.
  Checking Device
Reservations............................Allowed.
  Committing configuration
changes.......................Started.
  Committing configuration
```

```
changes.........................Committed.
  Terminating the configuration change session.............Done.
The configuration change session has successfully completed.
```

Step 2: (Optional) Verify that initial IP Interfaces have been created successfully

After the initial IP interfaces have been created, examine them to ensure that they have been created on the appropriate iSCSI SE director:port combination and are using the correct parameters (VLAN IDs, IP Address, MTU size) To examine the IP interfaces using Solutions Enabler, use the “symcfg list –ip” command.

```
PS C:\> symcfg -sid 0536 list -ip

Symmetrix ID: 000197900536 (Local)
                                        iSCSI
Dir:P NetId Vlan  IP Address                  Mtu  Port
------ ----- ---- ---------------------------------------- ----
-----
01F:28  10   82 192.168.82.30/24              9000   -
02F:28  10   83 192.168.83.30/24              9000   -
```

Note: The “-” for iSCSI port indicates that the specific IP interface is not attached to an iSCSI Target.

Step 3: Create the iSCSI Targets for the Prod1 environment

In this step, two iSCSI targets will be created - one created for each SE director used by the IP Interfaces created in the previous step. In the example, the SE directors are SE-1F and SE-2F. When creating the Targets, the IQN is user definable. When defining the IQN for the target, use a nomenclature that makes it easily identifiable on the host iSCSI environment. In the example, the nomenclature used follows “iqn.dellemc.<SID###>.<iSCSI director>.<environment name>”.

For the Prod1 example, recall that the targets will be called:

- iqn.dellemc.0536.1F.prod1
- iqn.dellemc.0536.2F.prod1

When creating the targets, it is important that the target and the eventual IP interface it is attached to share the same Network ID. In the example, the Prod1 IP Interfaces used network IDs of 10. These same network IDs will be used by the Prod1 targets.

The following “symconfigure create iscsi_tgt” commands will create the two “production” iSCSI targets used by this example. The associated iSCSI director on the PowerMax is specified by the “dir” parameter. The IQN assigned to the targets are specified by the “iqn” parameter. The network ID is specified by the “network_id” parameter.

```
PS C:\> symconfigure -sid 0536 -cmd "create iscsi_tgt dir 1F,
iqn=iqn.dellemc.0536.1F.prod1, network_id=10;" commit -noprompt

A Configuration Change operation is in progress. Please wait...
...
  created IQN : iqn.dellemc.0536.1F.prod1
  Committing configuration
```

```
changes.........................Committed.
  Terminating the configuration change session.............Done.
The configuration change session has successfully completed.

PS C:\> symconfigure -sid 0536 -cmd "create iscsi_tgt dir 2F,
iqn=iqn.dellemc.0536.2F.prod1, network_id=10;" commit -noprompt

A Configuration Change operation is in progress. Please wait...
...
  created IQN : iqn.dellemc.0536.2F.prod1
  Committing configuration
changes.........................Committed.
  Terminating the configuration change session.............Done.
The configuration change session has successfully completed.
```

Note: In the above commands, a specific TCP port number could have been specified by including the "tcp_port=####" parameter where #### is the user specified TCP port. When this parameter is omitted, the default iSCSI TCP port of 3260 is used. Also, specific port flags could have been specified in the commands as well.

Step 4: (Optional) Verify the iSCSI targets were created successfully

To examine the newly created targets using Solutions Enabler, use the following SYMCLI "symcfg list" command. Using the –se flag filters specifically for iSCSI SE directors and the –iscsi_tgt parameter will display the targets.

```
PS C:\> symcfg -sid 0536 list -se all -iscsi_tgt

Symmetrix ID: 000197900536 (Local)
Dir:P  NetId Status IQN
------- ----- ------- ----------------------------------------
01F:000  10 Offline iqn.dellemc.0536.1F.prod1
02F:000  10 Offline iqn.dellemc.0536.2F.prod1
```

In the above output, note the Dir:P column. The first part of the entry in the Dir:P column specifies the director and the second part designates the iSCSI virtual port the target has been assigned on the director. The initial target on any director will always be 000. Make a note of the iSCSI virtual port assigned to each director.

Step 5: Attach the Prod1 iSCSI targets to the Prod1 IP Interfaces

Once the targets have been created, they can then be attached to the IP Interfaces. Recall that each PowerMaxOS iSCSI target can be attached up to eight IP Interfaces. In this example, a single target will be attached to a single IP Interface. The target and any of the IP Interfaces it will be attached to need to use be associated with the same SE director and use the same network ID. The network IDs used in this example's Prod1 environment are 10 on both directors 1F and 2F. The target iqn.dellemc.0536.1F.prod1. will be attached to IP Interface with the address of 192.168.82.30 as they are both associated with Dir 1F and Network ID 10. The target iqn.dellemc.0536.2F.prod1 will be attached to the IP interface with the address of 192.168.83.30 as they both are associated with director SE-2F and use the same Network ID of 10.

To attach an IP interface to an iSCSI target using Solutions Enabler, use the “symconfigure ‘attach ip_interface’” command by specifying the IP address of the IP interface using the ‘ip_address’ parameter and the IQN of the target using the ‘iscsi_tgt’ parameter. The commands below attach the example’s IP Interfaces with the appropriate iSCSI target.

```
PS C:\> symconfigure -sid 0536 -cmd "attach ip_interface
ip_address=192.168.82.30 to iscsi_tgt
iqn=iqn.dellemc.0536.1F.prod1;" commit -noprompt


A Configuration Change operation is in progress. Please wait...
  Establishing a configuration change
session..............Established
...
The configuration change session has successfully completed.


PS C:\> symconfigure -sid 0536 -cmd "attach ip_interface
ip_address=192.168.83.30 to iscsi_tgt
iqn=iqn.dellemc.0536.2F.prod1;" commit -noprompt


A Configuration Change operation is in progress. Please wait...
  Establishing a configuration change
session..............Established.
...
The configuration change session has successfully completed.
```

Step 6: (Optional) Examine the attached “production” iSCSI target and IP interface combinations

Once the “production” targets have been attached to their IP interface, it is helpful to take a detailed look at the iSCSI configuration before moving to online the targets. In the detailed examination, look to ensure that the target has been attached to the desired interface; the status of the target; and re-examine the iSCSI flags.

To examine the iSCSI configuration in detail using Solutions Enabler, use the SYMCLI “symcfg list –se all –iscsi_tgt –detail” command.

```
PS C:\> symcfg -sid 0536 list -se all -iscsi_tgt -detail
Symmetrix ID: 000197900536 (Local)
  Director Identification: SE-01F
    iSCSI Name            : iqn.dellemc.0536.1F.prod1
    iSCSI Virtual Port        : 01F:000
    Status              : Offline <----
    Network ID            : 10
    IP Addresses
     {
     ------------------------------------------------------------
                                Tcp
     IP Address                  Vlan  Port Mtu
     ------------------------------------------------ ---- -----
```

```
     192.168.82.30                    82   3260 9000
     }
    iSCSI Flags
     {
      Soft_Reset(S)          : Disabled
      Environ_Set(E)          : Disabled
      Disable_Q_Reset_on_UA(D)     : Disabled
      Avoid_Reset_Broadcast(ARB)    : Disabled
      SCSI_3(SC3)           : Enabled
      SPC2_Protocol_Version(SPC2)   : Enabled
      SCSI_Support1(OS2007)      : Enabled
      Volume_Set_Addressing(V)     : Disabled
     }
  Director Identification: SE-02F
    iSCSI Name             : iqn.dellemc.0536.2F.prod1
    iSCSI Virtual Port         : 02F:000
    Status               : Offline <----
    Network ID             : 10
    IP Addresses
     {
     ------------------------------------------------------------
                                Tcp
     IP Address                  Vlan   Port Mtu
     ------------------------------------------- ---- -----
     192.168.83.30                    83   3260 9000
     ...
```

Step 7: Bring the Prod1 iSCSI targets online (enabling the target)

The final step in creating the “production” iSCSI configuration is to bring the iSCSI targets online. To bring the targets online, use the “symcfg -sid ### –se ## -iqn <name> online –nop” command where SE director value is used with the –SE parameter and target name is used with -iqn parameter.

```
PS C:\> symcfg -sid 0536 online -SE 1F -iqn
iqn.dellemc.0536.1F.prod1 -noprompt

A port 'Online' operation execution is
in progress for Symmetrix unit '000197900536'. Please wait...
The port 'Online' operation successfully executed for
Symmetrix Unit '000197900536'.

PS C:\> symcfg -sid 0536 online -SE 2F -iqn
iqn.dellemc.0536.2F.prod1 -noprompt

A port 'Online' operation execution is
in progress for Symmetrix unit '000197900536'. Please wait...
The port 'Online' operation successfully executed for
Symmetrix Unit '000197900536'.
```

Step 8: (Optional) Examine the completed Prod1 iSCSI configuration using Solutions Enabler

Once the “production” iSCSI targets are online, the initial Prod1 iSCSI configuration is completed on the PowerMax. Before moving on to provision storage, a good practice is to reverify that the targets are online and re-examine the overall iSCSI configuration.

To verify that the initial targets are online, use the “symcfg list –se all –iscsi_tgt” command and examine that status column in the output. The targets should be in the “Online” state.

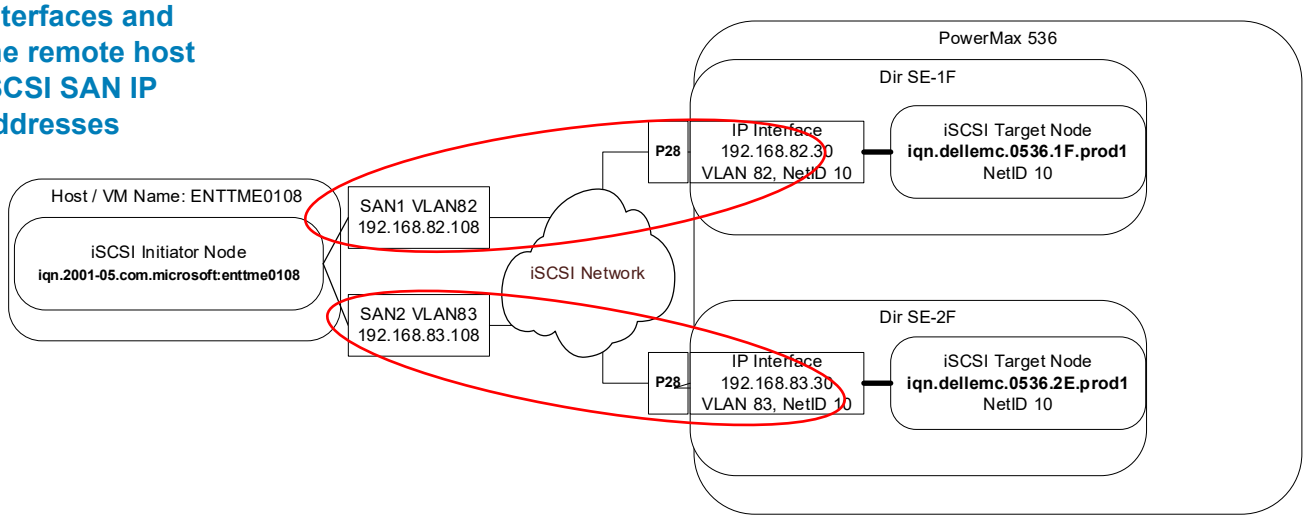
```
PS C:\> symcfg -sid 0536 list -se all -iscsi_tgt

Symmetrix ID: 000197900536 (Local)
Dir:P  NetId Status  IQN
------- ----- ------- ----------------------------------------
-------------
01F:000  10 Online iqn.dellemc.0536.1F.prod1
02F:000  10 Online iqn.dellemc.0536.2F.prod1
```

To examine the details of the configuration, the “–detail” parameter can be used with the “symcfg list –se all –iscsi_tgt” command.

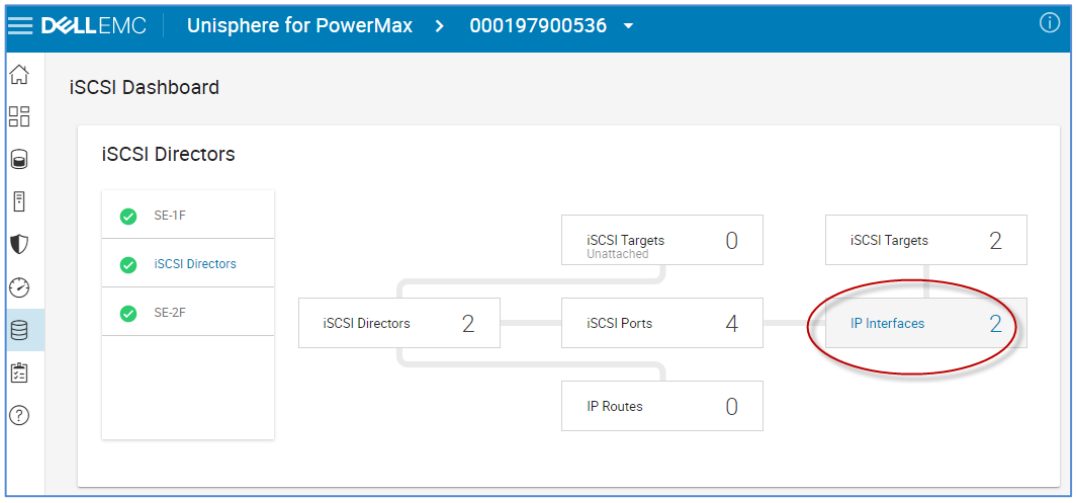
```
PS C:\> symcfg -sid 0536 list -se all -iscsi_tgt -detail

Symmetrix ID: 000197900536 (Local)
  Director Identification: SE-01F
    iSCSI Name            : iqn.dellemc.0536.1F.prod1
    iSCSI Virtual Port        : 01F:000
    Status              : Online <----
    Network ID            : 10
    IP Addresses
     {
     ------------------------------------------------------------
                                Tcp
     IP Address                  Vlan  Port Mtu
     ------------------------------------------- ---- -----
     192.168.82.30                  82  3260 9000
    ...

  Director Identification: SE-02F
    iSCSI Name            : iqn.dellemc.0536.2F.prod1
    iSCSI Virtual Port        : 02F:000
    Status              : Online <----
    Network ID            : 10
    IP Addresses
     {
     ------------------------------------------------------------
                                Tcp
     IP Address                  Vlan  Port Mtu
     ------------------------------------------- ---- -----
     192.168.83.30                  83  3260 9000
     ...
```

Verify connectivity between the new Prod1 IP interfaces and the remote host iSCSI SAN IP addresses

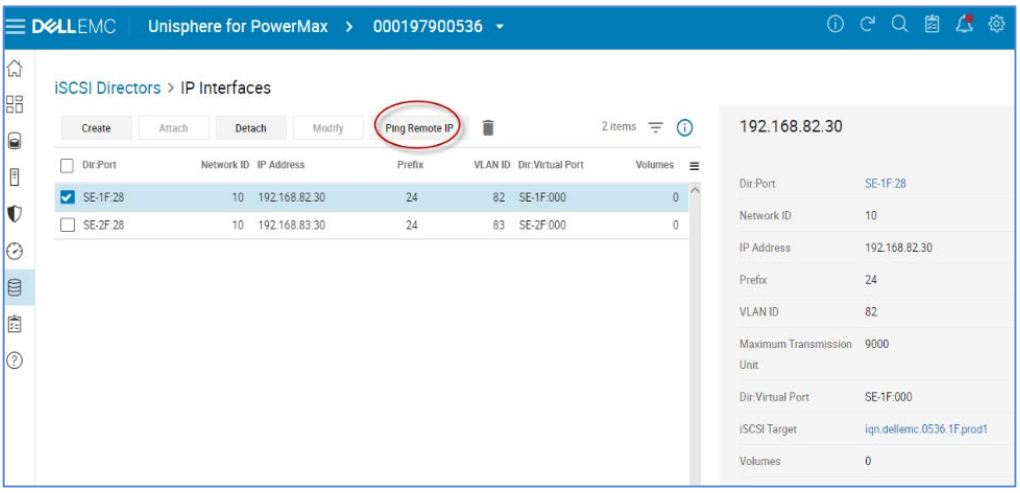
After the Prod1 components have been successfully created, it is important to verify connectivity between the newly created IP Interfaces and the remote host IP addresses. This can be done using the “ping” utility in either Unisphere for PowerMax or Solutions Enabler.



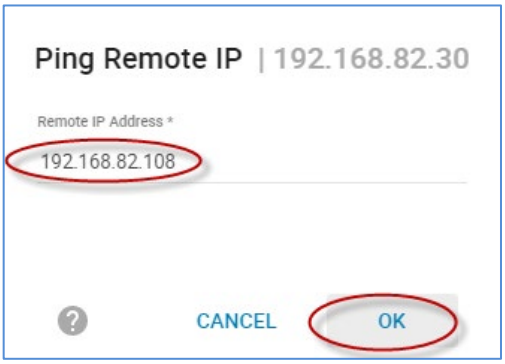
Using the ping utility in Unisphere for PowerMax



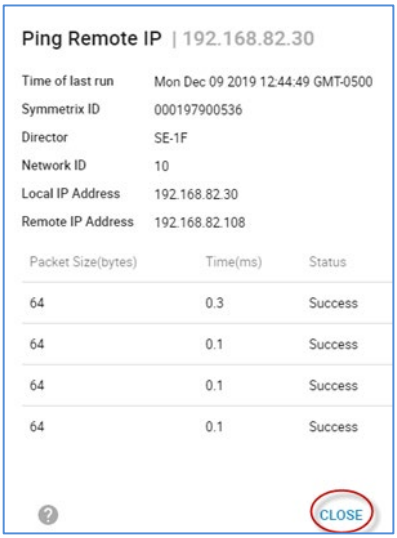
Highlight the first created Prod1 IP Interface (192.168.82.30) and then click the “Ping Remote IP” tab.



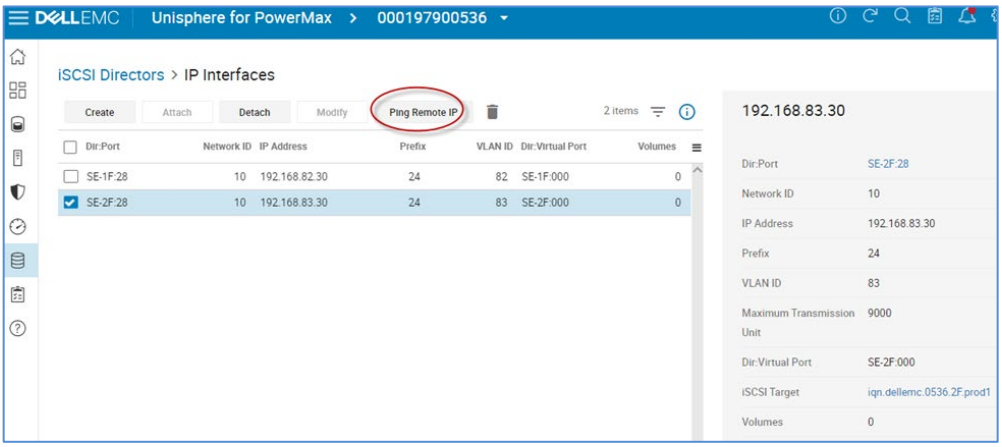
Enter the IP Address of the remote host (192.168.82.108) that is on the same network (.82) as the first IP interface and click “OK.”



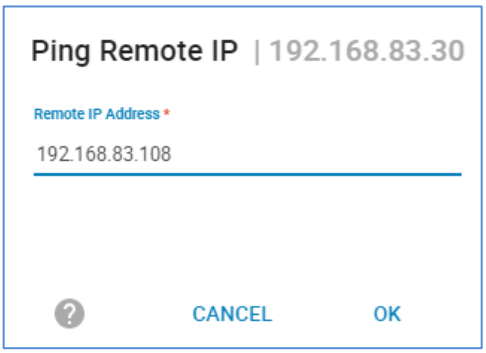
Verify that the first IP Interface can ping the specified remote host IP Address then click “Close.”



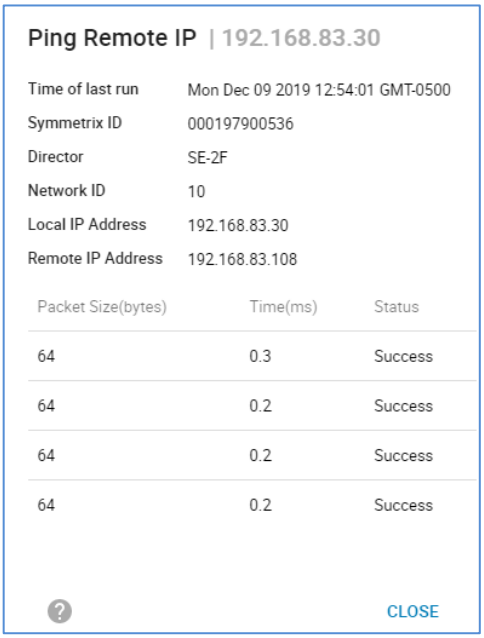
Repeat previous ping steps for the second created Prod1 IP Interface (192.168.83.30). Highlight the second IP Interface and select the “Ping Remote IP” tab.



Enter in the second IP Address of the Remote host (192.168.83.108) that is on the same network (.83) as the second Prod1 IP Interface and press “OK.”



Verify that the second IP Interface can ping the remote host IP address, then click “Close.”



Using the ping utility in Solutions Enabler

Verify connectivity between the Prod1 IP Interfaces and the remote host IP Addresses using the Solutions Enabler “symsan ping” command as follows:

```
PS C:\ >symsan -sid 0536 ping -SE 1F -network_id 10 -remote_ip
192.168.82.108
Symmetrix ID       : 000197900536 (Local)

SE Director       : 1F
Network ID        : 10
Local IP Address     : 192.168.82.30
Remote IP Address    : 192.168.82.108

   P I N G S T A T U S

Packet Size Time   Status
(bytes)    (ms)
----------- -------- ---------

64      0.2    Success


PS C:\>symsan -sid 0536 ping -SE 2F -network_id 10 -remote_ip
192.168.83.108

Symmetrix ID       : 000197900536 (Local)

SE Director       : 2F
Network ID        : 10
Local IP Address     : 192.168.83.30
Remote IP Address    : 192.168.83.108

   P I N G S T A T U S

Packet Size Time   Status
(bytes)    (ms)
----------- -------- ---------

64      0.2    Success
```

Section summary

This section showed how to create a basic iSCSI port binding configuration on a PowerMax. In the example, this configuration is called the Prod1 environment. This section showed how to create iSCSI IP Interfaces and iSCSI targets using both Solutions Enabler and Unisphere for PowerMax. The targets were attached to the IP interfaces and brought online. Throughout the example, techniques were shown on how to examine the configuration at different points during the construction.

The next section will show how to create an iSCSI masking view and how to present the devices to a server or virtual machine running Windows Server 2016.

Create an iSCSI masking view for the Prod1 host

This section will create an masking view using the IQN from the Prod1 iSCSI host / VM (ENTTME0108). It will demonstrate how to create the host and the host masking view using Unisphere for PowerMax. The masking view will use host IQN as the initiator for the initiator group; it will create and contain a total 200 GB using four volumes in the storage group, and the Prod1 iSCSI targets in the view's port group. An optional step in this section is setting up unidirectional (or One-Way) CHAP authentication for the host initiator and initiator group using Solutions Enabler.

Note: Setting CHAP authentication on an iSCSI initiator is currently not available in Unisphere for PowerMax. It can only be done through Solutions Enabler.

The diagram below is the completed Prod1 environment used by this example. The masking view created in this section will have the following components:

- The initiator group will contain the host initiator IQN.
- The storage group will contain the six volumes (4 x 50 GB) that will be presented to the Prod1 host (ENTTME0108).

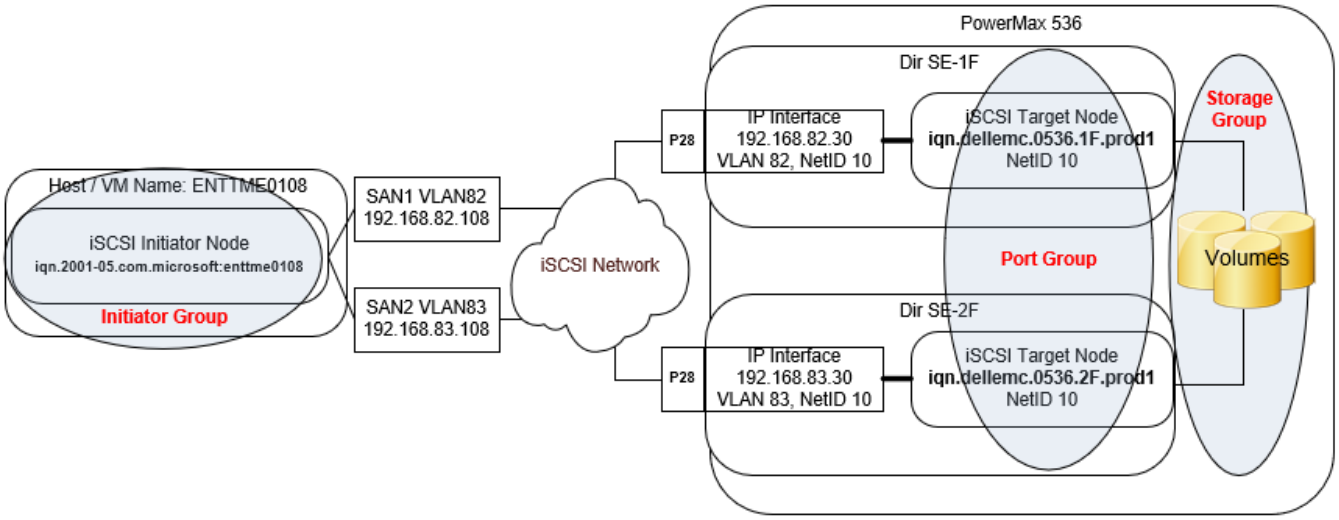


Figure 14. Initial masking view components for Prod1 host ENTTME0108

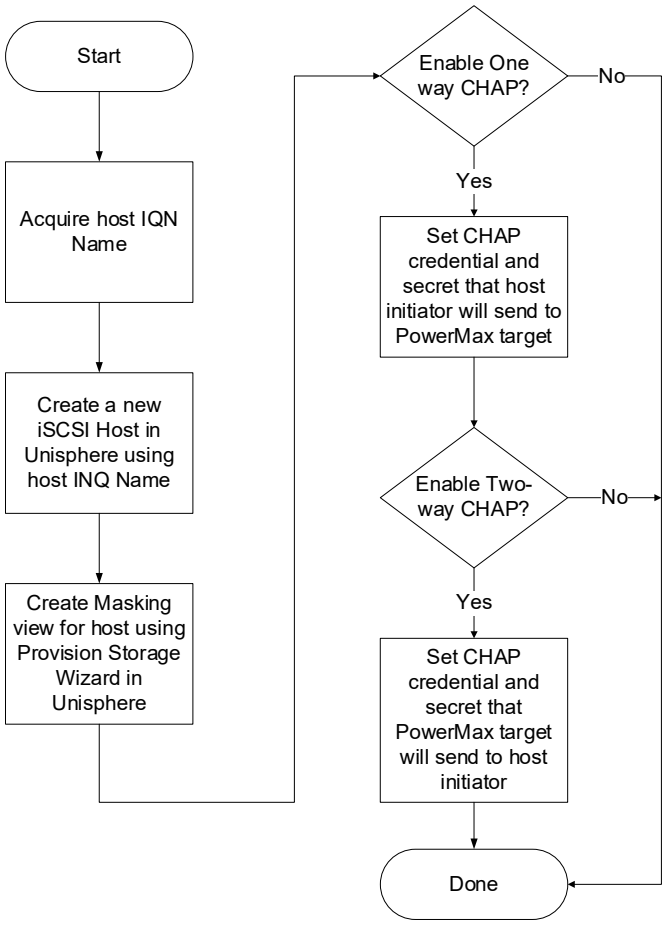
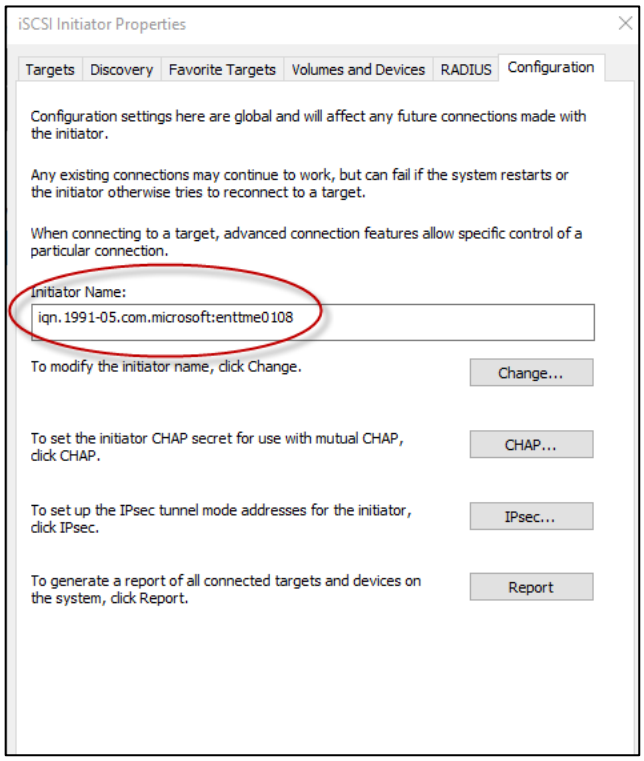


Figure 15. Process Flow Chart: Creating an iSCSI masking view on PowerMax

Create an iSCSI host in Unisphere

Step 1: Acquire the host or virtual machine initiator Name

The host used in the example (ENTTME0108) is a Windows Server 2016 host. If the host or virtual machine has not been attached to the network and / or has not previously attempted to log into the PowerMax, then the host administrator will have to provide the initiator IQN to the PowerMax storage administrator. On Windows servers and virtual machines, the host initiator IQN can be found by opening the iSCSI Initiator tool and going to the “Configuration” tab. The host IQN will can be found in the “Initiator Name:” text box.

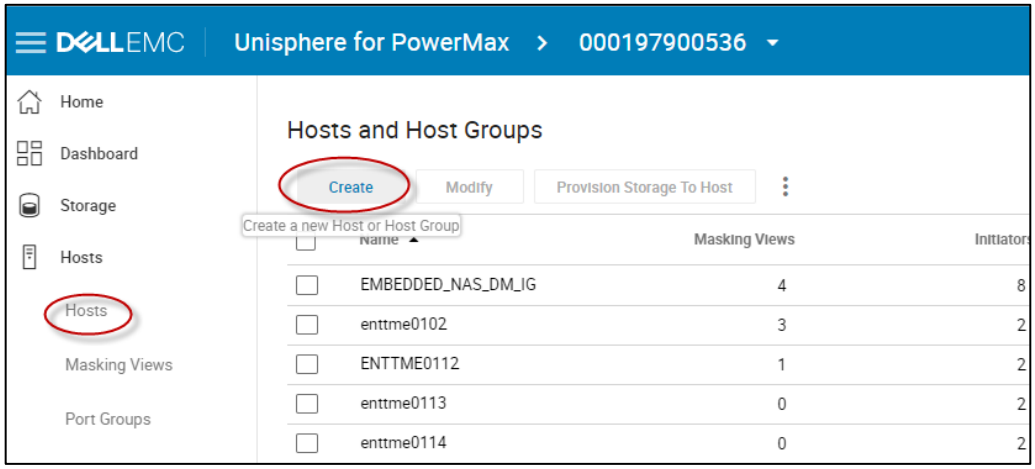


The host initiator name can also be determined by using the following PowerShell “one liner” command from the Windows host or virtual machine.

```
PS C:\>(get-initiatorport | where {$_.portaddress -like'*ISCSI*'}).nodeaddress
iqn.1991-05.com.microsoft:enttme0108
```

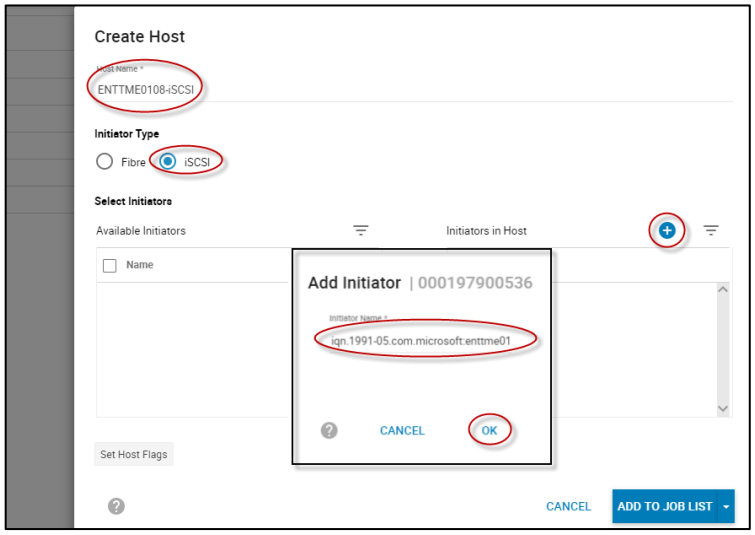
Step 2: Create the iSCSI host in Unisphere using the host IQN

After the host IQN has been acquired, create the host in Unisphere by going to PowerMax Array → Hosts → Hosts and Host Groups and select the “Create” tab.



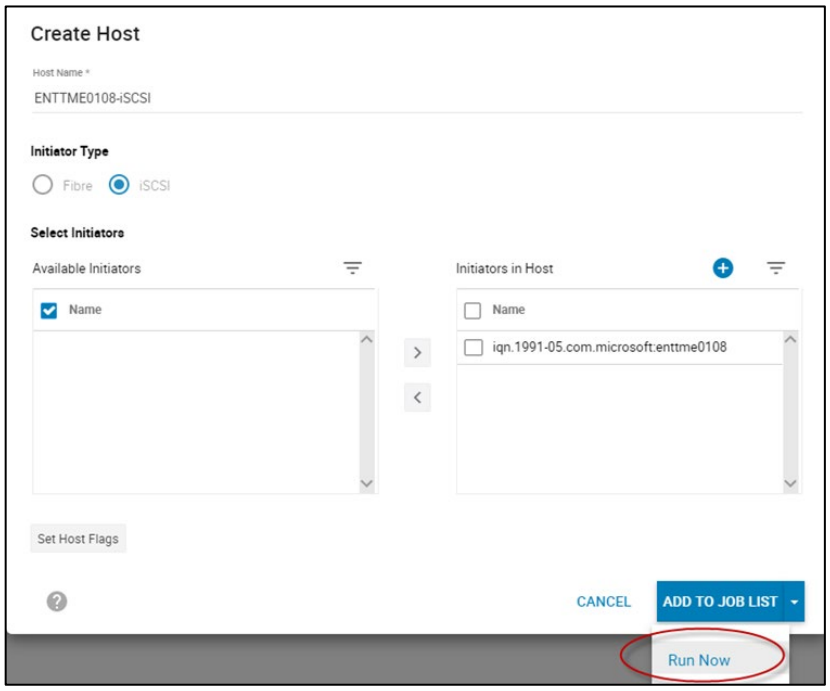
Once “Create” is selected, the “Create Host” wizard will open. Enter a name for the new host (ENTTME0108-iSCSI); select iSCSI for initiator type; select the “+” button to manually type or copy and paste the host IQN. Press “OK” to add the host IQN to the “Initiators in Host” box. If necessary, host flag options can be selected by clicking “Set

Host Flags" in the bottom-left corner of the wizard panel. In the example, the default host flags are used.



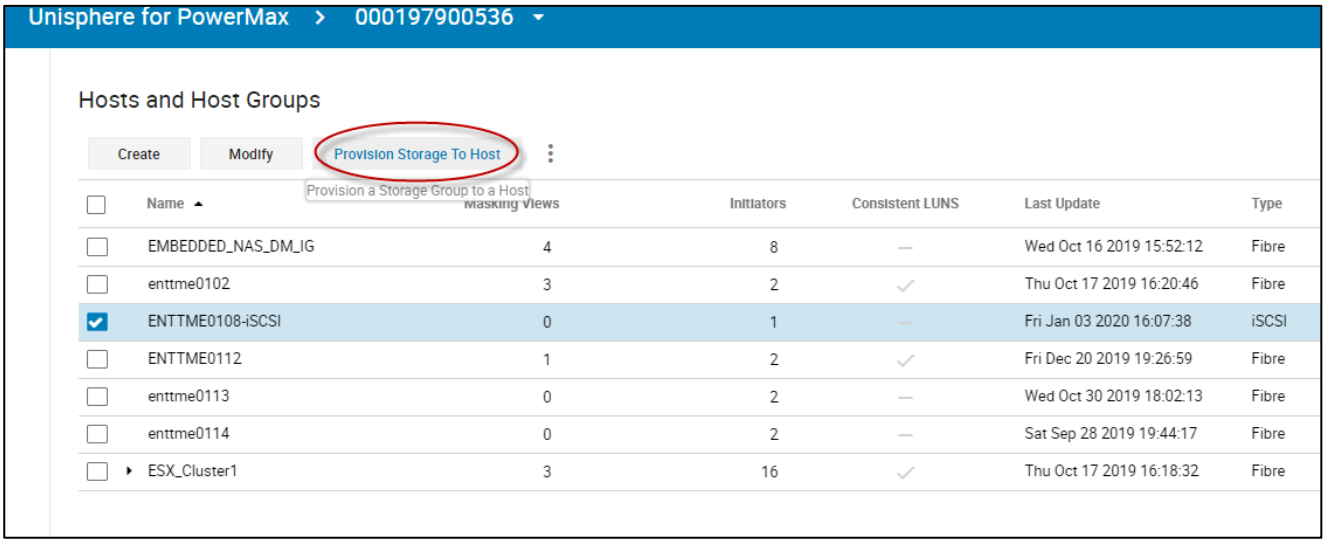
If the host had previously logged into the PowerMax, its IQN would already be in the PowerMax internal Login History Database. If this is the case, then the host IQN would show up in the "Available Initiators" box and could be selected without having to enter the host IQN manually.

After adding the initiator to the "Initiators in Host" box, select "Run Now" to create the host.



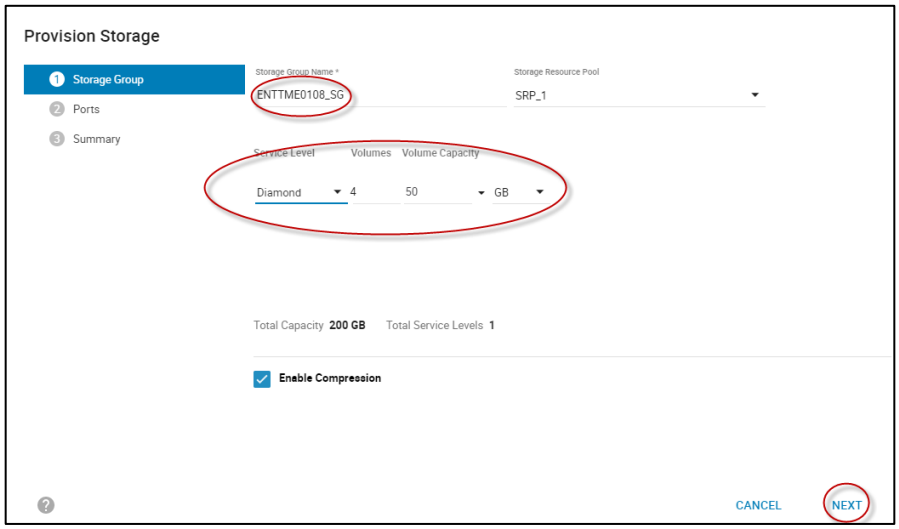
Create a Masking View for the new iSCSI Host

After creating the iSCSI host, the next step is to create its masking view. This can be easily done by going back to the hosts listings in Unisphere (PowerMax Array → Hosts → Hosts and Host Groups). Select the newly created host and click on the "Provision Storage to Host" tab.

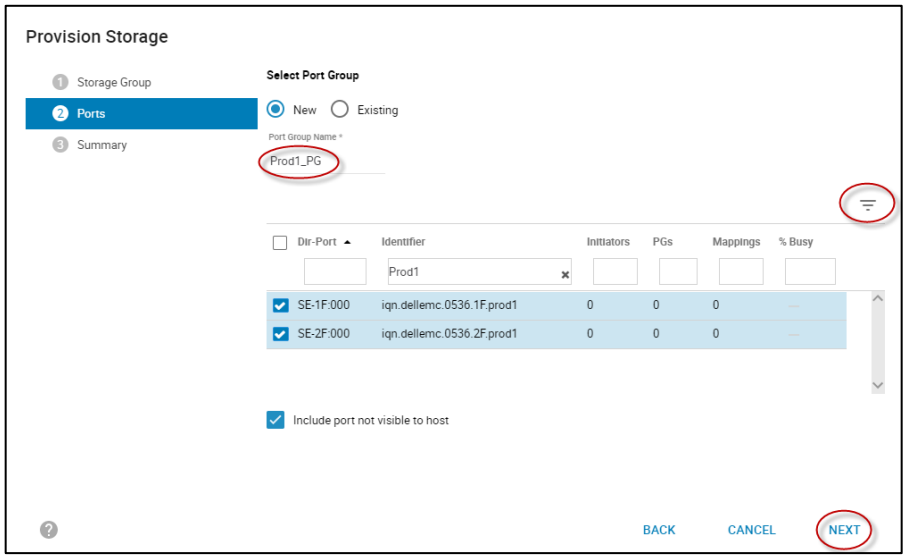


In an iSCSI masking view in PowerMax, the initiator group uses the host/VM initiator IQN. The Unisphere "Provision Storage to Host" wizard uses the host's name and IQN and to automatically create the initiator group for the masking view behind the scenes (ENTME0108-iSCSI will be the IG name in the PowerMax and the initiator it contains will be iqn.1991-05.com.microsoft:enttme0108).

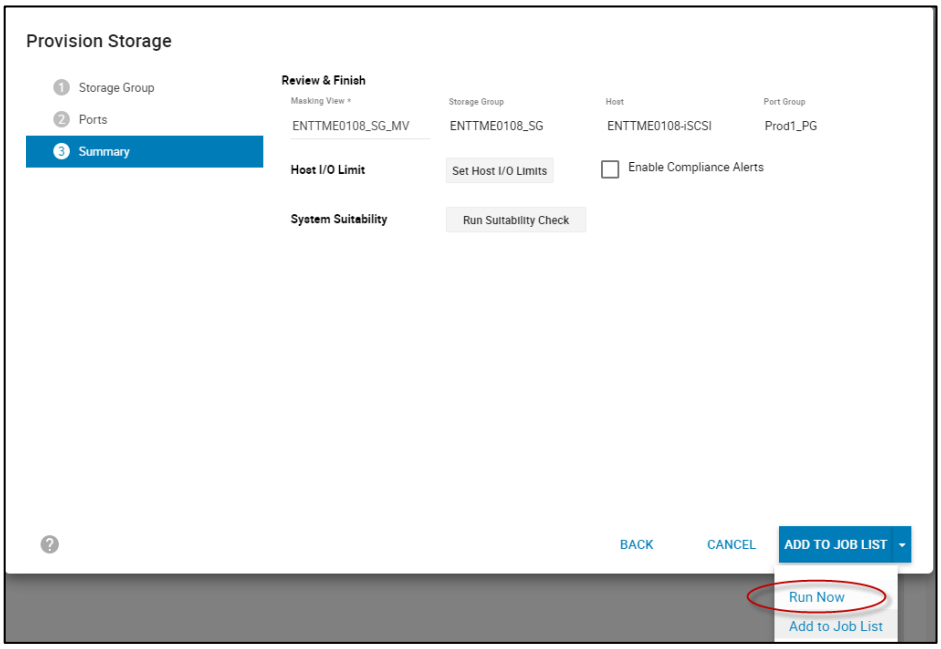
The first part of the wizard that requires user input is the creation of the storage group. In the example, the storage group will be named “ENTTME0108_SG” and it will use 4 x 50 GB volumes for a total storage capacity of 200 GB. The example’s storage group will use the “Diamond” service level. This paper will not discuss PowerMax service levels. A link to the PowerMax Service Level White Paper will be provided in the reference section of this document. PowerMax data reduction (both compression and deduplication) is enabled by default. Click “Next” to move on to creating the masking view port group.



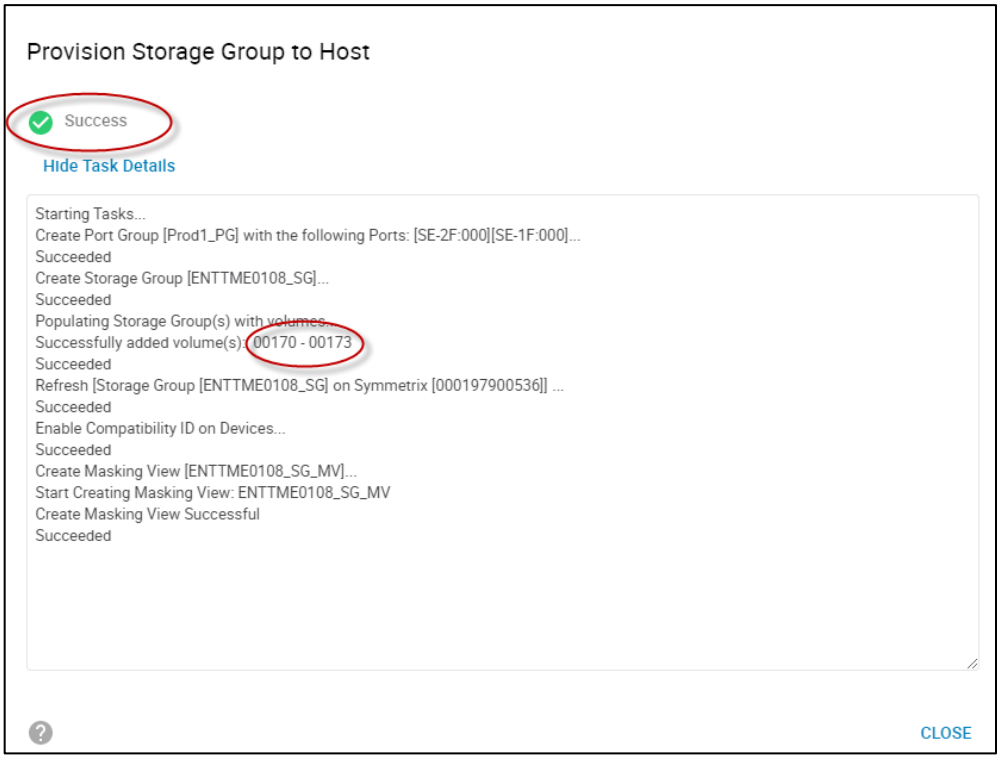
The next step in the wizard is to create a new port group used by the masking view. In the example, the port group will be named “Prod1_PG” and will use the Prod1 iSCSI targets. Use the filter icon to bring up the filter bar, and then type “Prod1” to quickly identify the two Prod1 iSCSI targets on the array. Select the two “Prod1” iSCSI targets and the click “Next.”



The final screen of the wizard is the summary screen. The user can at this point set up Host I/O Limits, Enable Compliance Alerts, and run a suitability check. The masking view gets a default name, which is <storage group name>_MV. This name can be changed later by the user. Click “Run Now” to create the masking view.



Monitor the create masking view operation and verify that it completes successfully. After it completes, note the volume numbers that were added to the storage group. If enough volumes of the specific size were already created and available to provide the capacity required, they would be added to the storage group. If there were not enough volumes of the specific size to be added to provide the required capacity, they would be created from free space and then added to the storage group during the creation of the masking view.



Optional: Set up CHAP authorization on the Prod1 host initiator

This example will set up One-Way CHAP for the “Prod1” host initiator. Setting up CHAP on the initiator is done using the “symaccess set chap” command as follows.

```
PS C:\> symaccess -sid 0536 -iscsi iqn.1991-05.com.microsoft:enttme0108 set chap -
cred iqn.1991-05.com.microsoft:enttme0108 -secret FreeTomBrady
```

In the above command, the “–iscsi” flag specifies the host initiator IQN. The value used with the “ –cred” flag represents the initiator credential. The initiator and the secret specified by the “-secret” flag must be presented exactly as typed in the above “symaccess” command by the initiator when it attempts to log in to the iSCSI target on the PowerMax array and create an iSCSI session.

The CHAP information for the host initiator is stored in the initiator group properties in the PowerMax ACLX db. This is why an initiator must be in an initiator group or in a masking view prior to enabling CHAP on it. To verify if CHAP is set for a specific initiator, a user does a “symaccess show -details” command on the initiator’s initiator group.

```
PS C:\ > symaccess -sid 0536 show ENTTME0108-iSCSI -type initiator
-detail

Symmetrix ID     : 000197900536

Initiator Group Name : ENTTME0108-iSCSI
Last update time   : 11:26:14 AM on Tue Dec 31,2019
Group last update time: 11:26:14 AM on Tue Dec 31,2019
Port Flag Overrides  : No
Consistent Lun    : No

  iSCSI Name      : iqn.1991-05.com.microsoft:enttme0108
  User-generated Name : /
  FCID Lockdown    : N/A
  Heterogeneous Host  : No
  Port Flag Overrides : No
  CHAP Enabled     : Yes <---
   CHAP Credential  : iqn.1991-05.com.microsoft:enttme0108
  Type         : iSCSI
```

The above output shows that CHAP has been enabled and what the required credential is for the host initiator.

Note: The secret is never presented in any command output. If the secret is forgotten, CHAP will need to be disabled and the reenabled on the initiator with a new secret. The host will have to use this new secret in order to reestablish the iSCSI session with the PowerMax targets.

Discover PowerMax iSCSI storage on the host

Once iSCSI has been set up and a masking view has been created for the host on the PowerMax, the host can discover the iSCSI targets and acquire the storage devices presented to it through the masking view. Once this is done, the volumes can be formatted. This section will demonstrate how to do this for a using the Prod1 iSCSI components and masking view created in the previous sections. As said earlier, the

example's host is running Windows Server 2016 The techniques shown in this section will use PowerShell, the Windows iSCSI Initiator Tool, and the Windows Server Manager UI.

Note: All the techniques shown in this section are well documented at Microsoft TechNet.

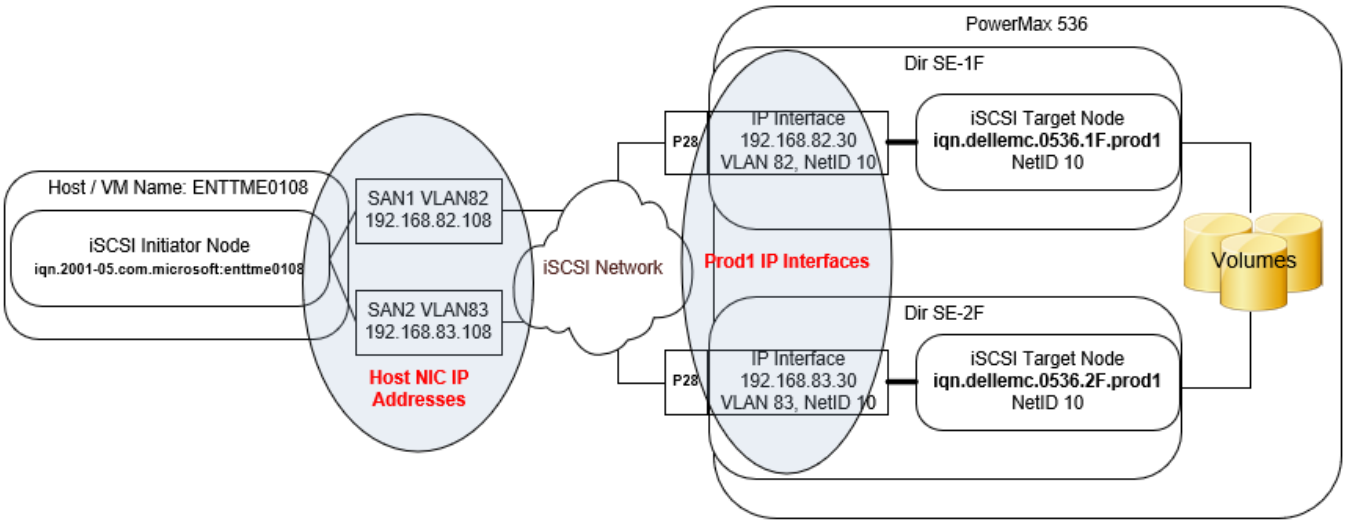


Figure 16. "Prod1" environment iSCSI network configuration

Note: This section assumes that the Windows host/virtual machine has been previously configured with the appropriate network information for use with the PowerMax iSCSI SAN (IP addresses, VLAN IDs). It is also assumed that the host/virtual machine has been properly configured to use the Microsoft iSCSI Software Initiator and that multipathing software (MPIO or PowerPath) has been installed. Details on how to set up a Windows host configuration for use with an iSCSI SAN is shown in the appendix section of this document.

Discover the PowerMax Prod1 IP Interfaces using PowerShell

In the example, there are two Prod1 IP interfaces on the PowerMax: 192.168.82.30 (for target iqn.dellemc.0536.1F.prod1) and 192.168.83.30 (for target iqn.dellemc.0536.2F.prod11). These interfaces are labeled SAN1 and SAN2 on the example's Windows host. To discover the two storage array iSCSI target IP Interfaces from the Windows host, use the PowerShell "New-IscsiTargetPortal" cmdlet or use the Windows iSCSI Initiator Tool UI.

First (if needed), identify the host iSCSI NIC IP addresses (initiator interfaces). In the example, these are the NIC interfaces created for VLAN 82 and VLAN 83. The following PowerShell one liner identifies the IP addresses for the initiator interfaces on the example's host:

```
[ENTTME0108] PS C:\>(Get-NetIPAddress -AddressFamily ipv4 | ?
{$_.interfacealias -like "SAN*"}).IPAddress

192.168.83.108
```

```
192.168.82.108
```

Note: To get the hostname to appear in the command prompt use the following PowerShell command:

```
PS C:\>function prompt {"[$env:computername] PS $(get-location)>"}
[ENTTME0108] PS C:\>
```

Using PowerShell

Discover the storage array iSCSI target IP interfaces using the IP address for the host initiator interfaces with the “New-IscsiTargetPortal” command as follows without CHAP:

```
[ENTTME0108] PS C:\>New-IscsiTargetPortal -TargetPortalAddress
192.168.82.30 -InitiatorPortalAddress 192.168.82.108
InitiatorInstanceName : ROOT\ISCSIPRT\0000_0
InitiatorPortalAddress : 192.168.82.108
IsDataDigest      : False
IsHeaderDigest     : False
TargetPortalAddress  : 192.168.82.30
TargetPortalPortNumber : 3260
PSComputerName     :

[ENTTME0108] PS C:\>New-IscsiTargetPortal -TargetPortalAddress
192.168.83.30 -InitiatorPortalAddress 192.168.83.108


InitiatorInstanceName : ROOT\ISCSIPRT\0000_0
InitiatorPortalAddress : 192.168.83.108
IsDataDigest      : False
IsHeaderDigest     : False
TargetPortalAddress  : 192.168.83.30
TargetPortalPortNumber : 3260
PSComputerName
```

In the above commands, a user specifies the network path between the host initiator and the storage array iSCSI target. The host initiator IP interface IP address is specified with “–InitiatorPortalAddress” flag and the associated target IP interface IP address is specified with the “–TargetPortalAddress” flag.

To discover the IP Interfaces if using CHAP:

```
[ENTTME0108] PS C:\>New-IscsiTargetPortal -TargetPortalAddress
192.168.82.30 -InitiatorPortalAddress 192.168.82.108 -
AuthenticationType ONEWAYCHAP -ChapSecret FreeTomBrady
InitiatorInstanceName : ROOT\ISCSIPRT\0000_0
InitiatorPortalAddress : 192.168.82.108
IsDataDigest      : False
IsHeaderDigest     : False
TargetPortalAddress  : 192.168.82.30
TargetPortalPortNumber : 3260
PSComputerName     :
```

```
[ENTTME0108] PS C:\>New-IscsiTargetPortal -TargetPortalAddress
192.168.83.30 -InitiatorPortalAddress 192.168.83.100 -
AuthenticationType ONEWAYCHAP -ChapSecret FreeTomBrady

InitiatorInstanceName : ROOT\ISCSIPRT\0000_0
InitiatorPortalAddress : 192.168.83.108
IsDataDigest       : False
IsHeaderDigest     : False
TargetPortalAddress  : 192.168.83.30
TargetPortalPortNumber : 3260
PSComputerName     :
```

If One-Way CHAP has been enabled, the host needs to specify the appropriate CHAP authentication type with the "-AuthenticationType" flag, along with the appropriate CHAP secret it must present to the PowerMax. This secret was specified when CHAP was set up on the initiator IQN on the PowerMax.

Note: The three valid options for authentication type in the above "New-IscsiTargetPortal" command are "NONE," "ONEWAYCHAP," and "MUTUALCHAP" – all in capital letters. There is an error in the PowerShell 4.0 documentation, which states that the valid options are "None," "OneWayChap," and "MutualChap." This is incorrect and will be updated in a future release of PowerShell from Microsoft. This also applies to the upcoming "Connect-IscsiTarget" command.

Using the Windows iSCSI Initiator Tool UI

To discover the storage array iSCSI target IP interfaces using the Windows iSCSI Initiator Tool, open the tool through server manager → tools → iSCSI Initiator and go to the "Discovery" tab. In the Discovery tab, select "Discover Portal…"

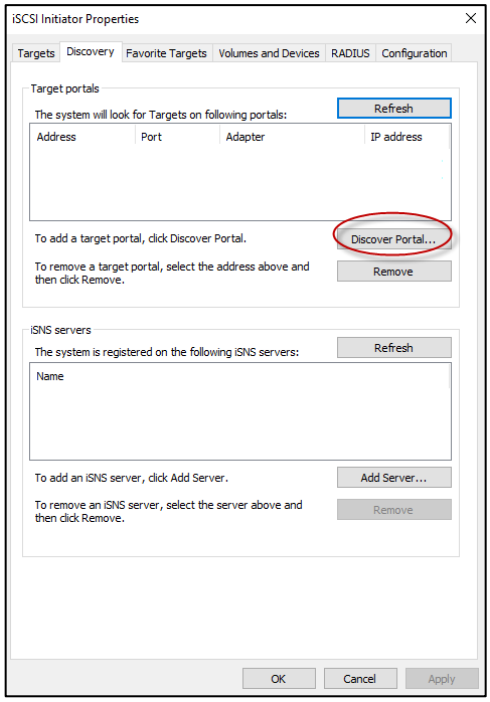
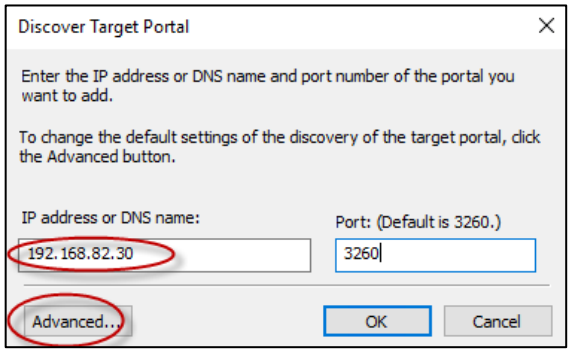


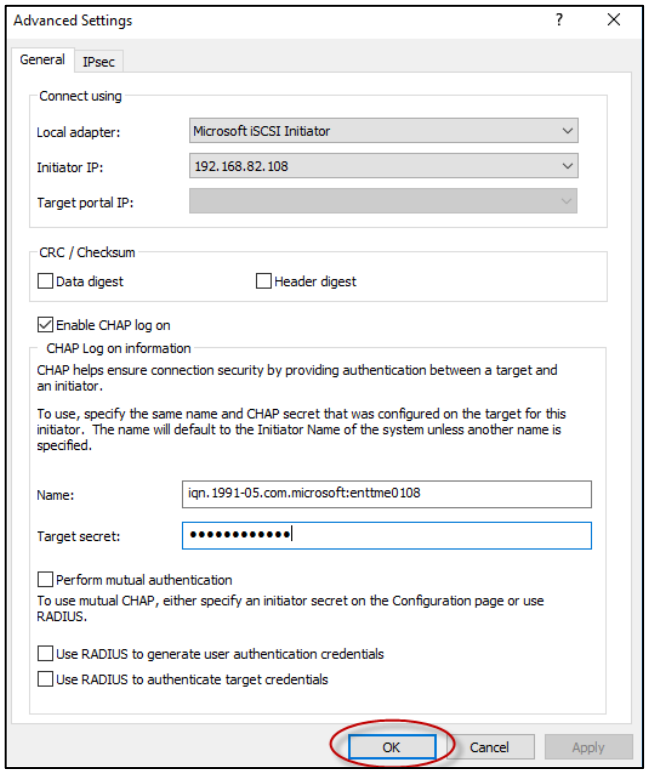
Figure 17. Discover the IP Interfaces using the Windows iSCSI Initiator Tool

The “Discover Target Portal” window appears. Enter the IP address of the storage array iSCSI target IP interface and then select “Advanced…”

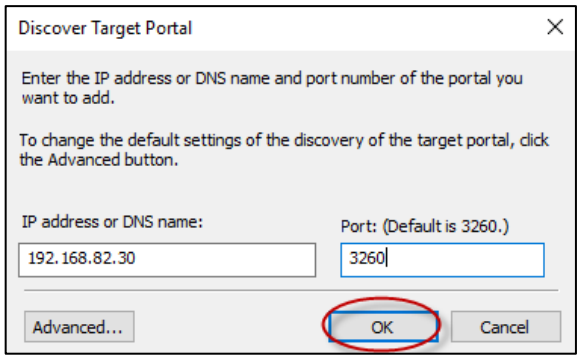


By clicking “Advanced…,” the “Advanced Setting” window opens. This where the storage array iSCSI target/IP interface discovery and connection information used by the host initiator is entered and stored. In the window, the user specifies the “Microsoft iSCSI Initiator” in the “Local adapter” drop down and the host initiator IP address used for connecting to the target on the POWERMAX in the “Initiator IP:” drop down.

If CHAP is enabled, the user enters the relevant CHAP information by selecting “Enable CHAP log on.” Here the initiator credential is shown in “Name” text box (the default is the IQN of the initiator) and the target CHAP secret is entered in the “Target secret” text box. As said earlier, the credential and target secret entered here are what is passed to the POWERMAX for authentication by the target. These values must match exactly the values what was entered in the “symaccess set chap” command –cred and –secret parameters.



Click “OK” to save the connection information entered. This will close “Advanced Settings” and go back to the “Discover Target Portal” window. Click “OK” to discover the PowerMax IP Interface.



Repeat the previous steps using the Windows iSCSI Initiator tool to discover the second IP Interface (192.168.83.30).

Connect to the host to the PowerMax iSCSI Targets.

Once the target IP interfaces have been discovered, the host will be able to see the specific storage array iSCSI targets associated with the target IP interfaces. The next step in the process is to establish a connection from the host to the target. This can be done on a Windows host using either PowerShell or the Windows iSCSI Initiator Tool UI.

Using PowerShell

Once the IP interfaces have been discovered, examine the iSCSI targets that are associated with the portal IP addresses. This can be done using the “get-iscsitarget” cmdlet.

```
[ENTTME0108] PS C:\>Get-IscsiTarget | ft -AutoSize

IsConnected NodeAddress        PSComputerName
----------- -----------        --------------
    False iqn.dellemc.0536.1F.prod1
    False iqn.dellemc.0536.2F.prod1
```

In the above output, the two targets are visible to the host; however, there are no connections created yet between the host initiators and targets. To connect host initiator though its first interface (192.168.82.108) to PowerMax iSCSI target (iqn.dellemc.0536.1F.prod1), use the “Connect-IscsiTarget” cmdlet if CHAP is not being used. Successful completion of this command will result in a newly created session between the host initiator and PowerMax iSCSI target (iqn.dellemc.0536.1F.prod1). The session details are shown in the command output.

```
[ENTTME0108] PS C:\>Connect-IscsiTarget -NodeAddress iqn.dellemc.0536.1F.prod1 -
InitiatorPortalAddress 192.168.82.108 -IsMultipathEnabled $true -IsPersistent
$true


AuthenticationType    : NONE                                ← NO CHAP being used
InitiatorInstanceName  : ROOT\ISCSIPRT\0000_0
```

```
InitiatorNodeAddress  : iqn.1991-05.com.microsoft:enttme0108    ← Host IQN
InitiatorPortalAddress : 192.168.82.108                         ← First Initiator
IP
InitiatorSideIdentifier : 400001370000
IsConnected       : True
IsDataDigest      : False
IsDiscovered      : True
IsHeaderDigest     : False
IsPersistent      : True
NumberOfConnections   : 1
SessionIdentifier    : ffffc00f984e6010-400001370000000a        ← Session ID
TargetNodeAddress    : iqn.dellemc.0536.1f.prod1          ← PowerMax Target
TargetSideIdentifier  : 0100
PSComputerName     :

[ENTTME0108] PS C:\>Connect-IscsiTarget -NodeAddress iqn.dellemc.0536.2F.prod1 -
InitiatorPortalAddress 192.168.83.108 -IsMultipathEnabled $true -IsPersistent
$trueget-disk


AuthenticationType   : NONE                                  ← No CHAP being used
InitiatorInstanceName  : ROOT\ISCSIPRT\0000_0
InitiatorNodeAddress  : iqn.1991-05.com.microsoft:enttme0108    ← Host IQN
InitiatorPortalAddress : 192.168.83.108                         ← Second
Initiator IP
InitiatorSideIdentifier : 400001370000
IsConnected       : True
IsDataDigest      : False
IsDiscovered      : True
IsHeaderDigest     : False
IsPersistent      : False
NumberOfConnections   : 1
SessionIdentifier    : ffffc00f984e6010-400001370000000b  ← Session ID
TargetNodeAddress    : iqn.dellemc.0536.2f.prod1          ← PowerMax Target
TargetSideIdentifier  : 0100
PSComputerName     :
```

If using CHAP:

```
[ENTTME0108] PS C:\>Connect-IscsiTarget -NodeAddress iqn.dellemc.0536.1F.prod1 -
InitiatorPortalAddress 192.168.82.108 -IsMultipathEnabled $true -IsPersistent
$true -AuthenticationType ONEWAYCHAP -ChapSecret FreeTomBrady
AuthenticationType   : ONEWAYCHAP                            ← One-way CHAP being
used
InitiatorInstanceName  : ROOT\ISCSIPRT\0000_0
InitiatorNodeAddress  : iqn.1991-05.com.microsoft:enttme0108    ← Host IQN
InitiatorPortalAddress : 192.168.82.108                         ← First Initiator
IP
InitiatorSideIdentifier : 400001370000
IsConnected       : True
IsDataDigest      : False
```

```
IsDiscovered      : True
IsHeaderDigest     : False
IsPersistent      : True
NumberOfConnections   : 1
SessionIdentifier    : ffffc00f984e6010-400001370000000a        ← Session ID
TargetNodeAddress    : iqn.dellemc.0536.1f.prod1          ← PowerMax Target
TargetSideIdentifier  : 0100
PSComputerName     :
```

When using “Connect-IscsiTarget,” the connecting initiator interface IP address is specified with the “InitiatorPortalAddress” flag. The target to connect to is specified with “NodeAddress” flag. Also in this command, the CHAP authentication type and the CHAP target secret must be included if CHAP has been enabled (See notes about CHAP specified previously with the New-IscsiTargetPortal cmdlet). The other parameters used in the cmdlet specify that host initiator has enabled Multipath IO and will use this when logging into the target and that the resulting session is persistent and to be automatically reconnected after each host reboot.

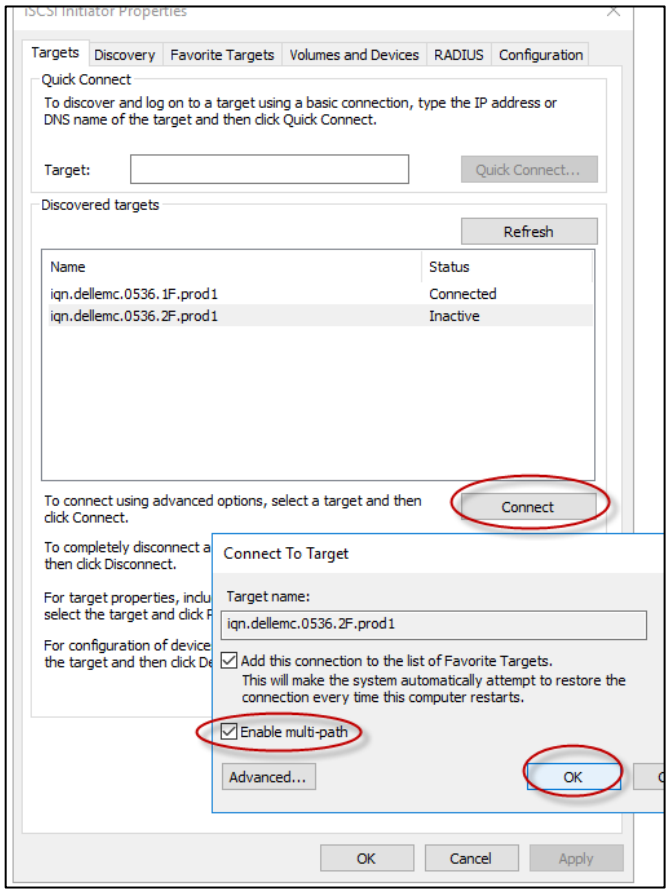
After running the “Connect-IscsiTarget cmdlet for both PowerMax iSCSI targets, reexamine iSCSI targets on host to verify that the connection state is "True".

```
[ENTTME0108] PS C:\>Get-IscsiTarget | ft -AutoSize

IsConnected NodeAddress        PSComputerName
----------- -----------        --------------
   True iqn.dellemc.0536.1F.prod1
   True iqn.dellemc.0536.2F.prod1
```

Note: The discovering and connecting steps only needs to be performed one time whenever a new target is presented to the host or virtual machine. After the targets have been discovered and connected, a host simply has to do a storage rescan in order to acquire additional iSCSI storage presented through the connected targets.

Using the Windows iSCSI Initiator Tool UI

Connecting to a newly discovered target is straightforward using the Windows iSCSI Initiator Tool UI. To connect to a new target, go to the “Targets” tab, select a target, and then click “Connect.” The iSCSI tool will prompt to enable the connection for multipathing. Click “OK” connect to the target. Once the target is connected, the status will change to “Connected.” Repeat this step for additional targets.

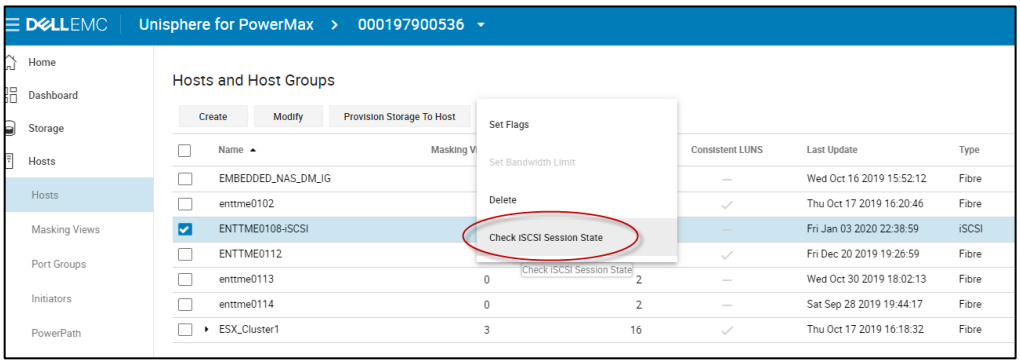


Note: The Windows iSCSI Initiator Tool will connect to the targets using the parameters entered previously in the “Advanced Settings” window (Initiator IP Interface, CHAP secret). Those parameters do not have to be reentered when connecting to the discovered targets.

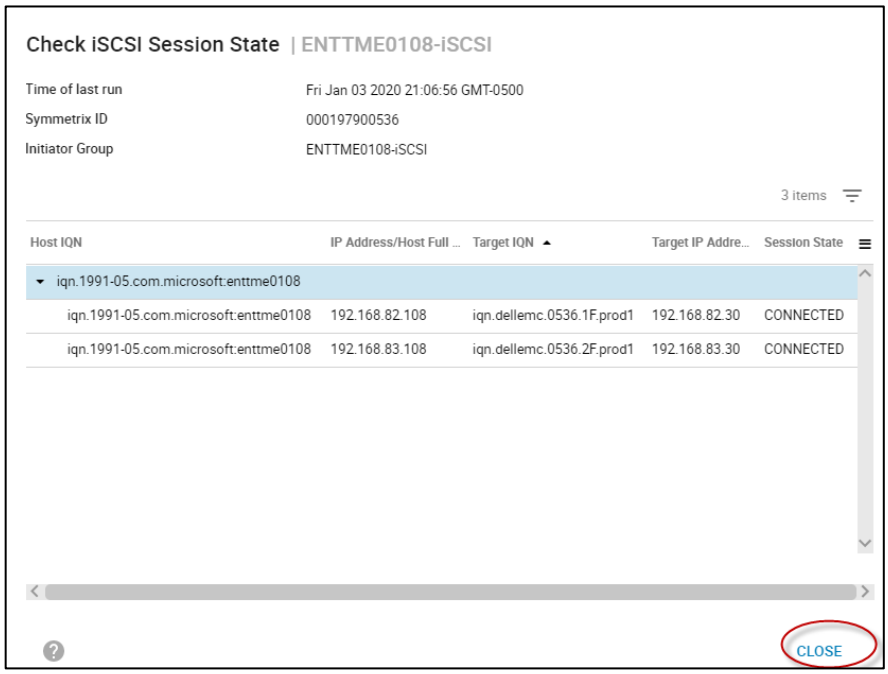
Troubleshooting tip: Verify the host iSCSI session status on the PowerMax

This dialog displays iSCSI sessions for an initiator group on arrays running PowerMax OS 5978_Q219SR or above. iSCSI sessions on SE directors for an initiator group with iSCSI initiators are displayed along with the session state. The feature is intended to help users perform basic troubleshooting when iSCSI hosts lose connectivity to PowerMax arrays.

In order to see the session status in Unisphere for PowerMax, select the host (ENTTME0108-iSCSI), click three dots (more actions), and select “Check iSCSI Session State.”



The iSCSI session state for the host IQN will be show in the output. In the example, the session state is CONNECTED. Select close when done.



This information can also be seen in Solutions enabler with the following “symsan” command:

```
PS C:\>symsan -sid 0536 show -iscsi_sessions -ig ENTTME0108-iSCSI
-se all

Symmetrix ID          : 000197900536
Initiator Group       : ENTTME0108-iSCSI

Host IQN           : iqn.1991-05.com.microsoft:enttme0108
iSCSI Sessions
  {
  Director Identification  : SE-01F
  Director Port        : 028
  Host IP Address      : 192.168.82.108
  Target IQN         : iqn.dellemc.0536.1F.prod1
```

```
Target IP Address     : 192.168.82.30
Target IP Prefix Length  : 24
Session State         : CONNECTED

Director Identification  : SE-02F
Director Port         : 028
Host IP Address       : 192.168.83.108
Target IQN         : iqn.dellemc.0536.2F.prod1
Target IP Address     : 192.168.83.30
Target IP Prefix Length  : 24
Session State         : CONNECTED
```

Rescan the storage on host

Although not always necessary, it is always a good idea to do a storage rescan anytime the host storage configuration changes or is updated.

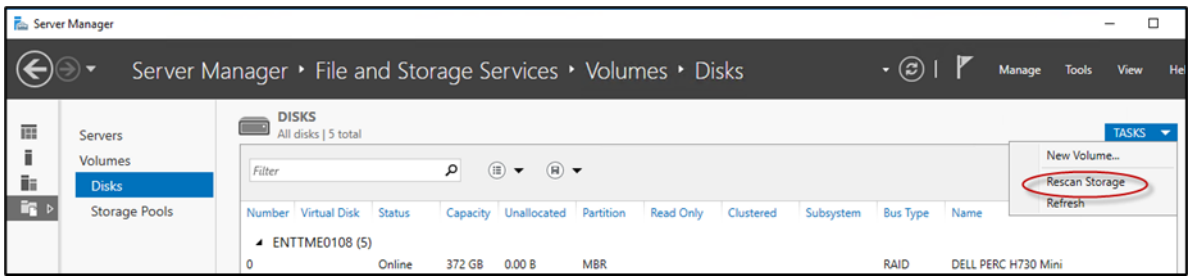
Using PowerShell

The PowerShell “Update-HostStorageCache” cmdlet will refresh the storage configuration on the host or virtual machine.

```
[ENTTME0108] PS C:\> Update-HostStorageCache
```

Using Windows Server Manager UI

To rescan the storage bus using Windows Server Manager, go to “Volumes” and select “Disks.” Select the “TASKS” drop down and then select “Rescan Storage.”



Verify the PowerMax volumes are visible to the host

After the establishment of the iSCSI session to the storage array iSCSI targets and storage rescan, the new devices should be visible to the host but will be in an “Offline” status.

Using PowerShell

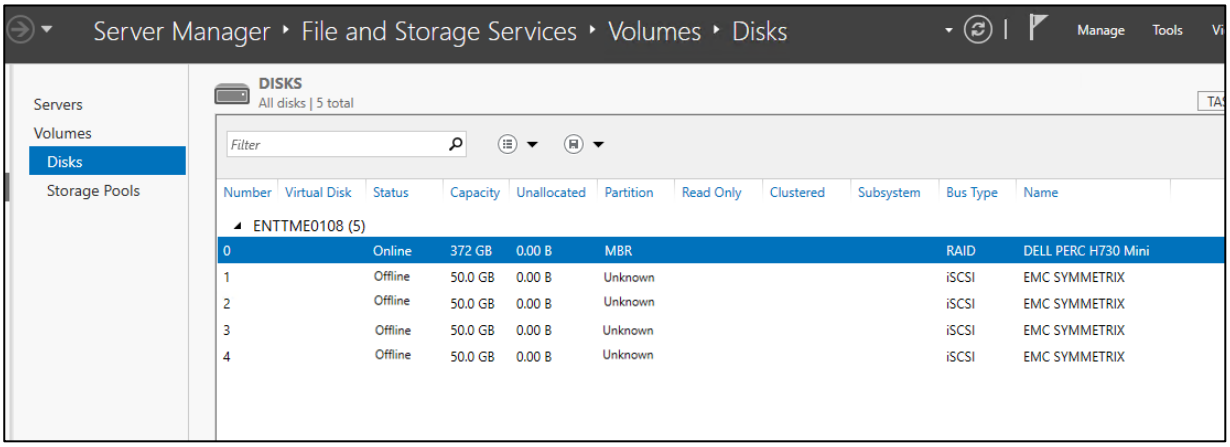
To examine the storage visible to the host or virtual machine, use the “get-disk” cmdlet. In the example below, the four new iSCSI devices will have a status of “Offline.” Also notice that the friendly name specifies "EMC Symmetrix."

```
[ENTTME0108] PS C:\>get-disk | ft -AutoSize

Number Friendly Name    Serial Number          HealthStatus OperationalStatus
Total Size Partition Style
------ -------------    -------------          ------------ ----------------- ----
------ ---------------
```

```
0   DELL PERC H730 Mini 00fe965c1c5ed1042300c85c82a06d86 Healthy   Online
372 GB MBR
1   EMC SYMMETRIX    900536170000           Healthy   Offline        50 GB
2   EMC SYMMETRIX    900536171000           Healthy   Offline        50 GB
3   EMC SYMMETRIX    900536172000           Healthy   Offline        50 GB
4   EMC SYMMETRIX    900536173000           Healthy   Offline        50 GB
```

Using Windows Server Manager UI

After the “Rescan Storage” task completes, the new devices will appear in the “DISKS” window with a status of “Offline.”



Using PowerPath

The example’s host has PowerPath installed. To examine the presented devices in PowerPath use the “powermt display dev=all” command

```
[ENTTME0108] PS C:\>powermt display dev=all
Pseudo name=harddisk1
Symmetrix ID=000197900536
Logical device ID=00170
Device WWN=60000970000197900536533030313730
state=alive; policy=SymmOpt; queued-IOs=0
==============================================================================
--------------- Host ---------------   - Stor - -- I/O Path --   -- Stats ---
### HW Path         I/O Paths   Interf. Mode    State   Q-IOs Errors
==============================================================================
  3 port3\path0\tgt1\lun0 c3t1d0   SE 2f:28 active   alive    0    0
  3 port3\path0\tgt0\lun0 c3t0d0   SE 1f:28 active   alive    0    0

Pseudo name=harddisk2
Symmetrix ID=000197900536
Logical device ID=00171
Device WWN=60000970000197900536533030313731
state=alive; policy=SymmOpt; queued-IOs=0
==============================================================================
--------------- Host ---------------   - Stor - -- I/O Path --   -- Stats ---
### HW Path         I/O Paths   Interf. Mode    State   Q-IOs Errors
==============================================================================
```

```
  3 port3\path0\tgt1\lun1 c3t1d1   SE 2f:28 active  alive   0   0
  3 port3\path0\tgt0\lun1 c3t0d1   SE 1f:28 active  alive   0   0

Pseudo name=harddisk3
Symmetrix ID=000197900536
Logical device ID=00172
Device WWN=60000970000197900536533030313732
state=alive; policy=SymmOpt; queued-IOs=0
==============================================================================
--------------- Host ---------------  - Stor - -- I/O Path --  -- Stats ---
### HW Path        I/O Paths  Interf. Mode   State  Q-IOs Errors
==============================================================================
  3 port3\path0\tgt1\lun2 c3t1d2   SE 2f:28 active  alive   0   0
  3 port3\path0\tgt0\lun2 c3t0d2   SE 1f:28 active  alive   0   0

Pseudo name=harddisk4
Symmetrix ID=000197900536
Logical device ID=00173
Device WWN=60000970000197900536533030313733
state=alive; policy=SymmOpt; queued-IOs=0
==============================================================================
--------------- Host ---------------  - Stor - -- I/O Path --  -- Stats ---
### HW Path        I/O Paths  Interf. Mode   State  Q-IOs Errors
==============================================================================
  3 port3\path0\tgt1\lun3 c3t1d3   SE 2f:28 active  alive   0   0
  3 port3\path0\tgt0\lun3 c3t0d3   SE 1f:28 active  alive   0   0
```

Optional: Online, initialize, and create a new file system on the iSCSI volumes

Once the disks are visible to the operating system, they can be brought online, initialized, and formatted. This can be done using either PowerShell or Windows Server Manager.

Using PowerShell

The following is a sample PowerShell script that will online, initialize, and format all the EMC disks visible to the host OS that are in the “Offline” state. Scripts like this are beneficial as they can be used to work on multiple disks at a time.

Note: The following script should be used with discretion. There is no error checking included in the script and it might take a few moments to run. As each device is formatted, it will appear in the output.

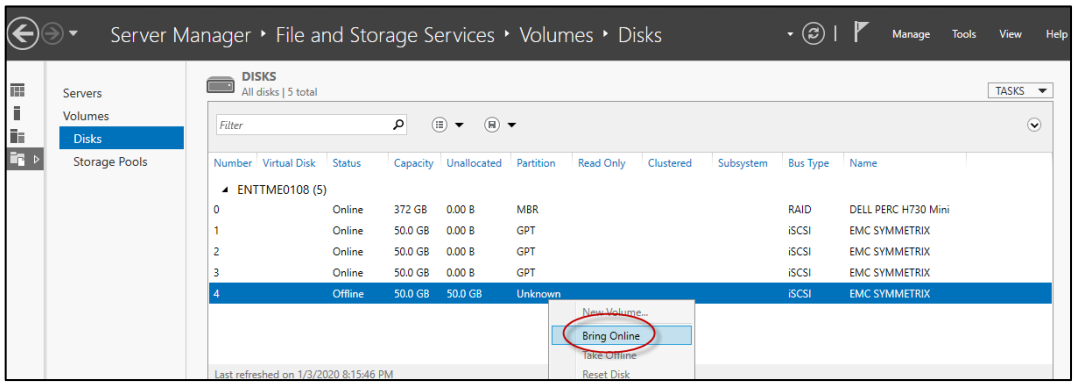
```
#############################################################################
##
#PowerShell script to initialize, partition, and format new EMC offline disks
#############################################################################
##
```

```
#Get disk numbers from all offline EMC disks visible to host OS
$disknumbers = (get-disk | ? {$_.FriendlyName -like "EMC*" -and
$_.OperationalStatus -eq "Offline"}).number

#Starting foreach loop through disk numbers
$disknumbers | % {
  #place disk number from input stream into variable
  $disknum = $_

  #create volume label to be used later
  $newvollabel = "Disk"+$_

  #inner foreach loop - initialize, online, format, and create new volume on disk
  Initialize-Disk -number $disknum -PartitionStyle GPT -PassThru |
  % {
    $_ | set-disk -IsReadOnly 0
    $_ | Set-disk -IsOffline 0
    $_ | New-Partition -AssignDriveLetter -UseMaximumSize | Format-Volume -
FileSystem  NTFS -NewFileSystemLabel $newvollabel -Confirm:$false
  }
}
#End of Script
################################################################################
###
<output of script>
DriveLetter FileSystemLabel FileSystem DriveType HealthStatus OperationalStatus
SizeRemaining   Size
----------- --------------- ---------- --------- ------------ ----------------- --
-----------   ----
E      Disk1      NTFS    Fixed   Healthy   OK           49.78 GB 49.87 GB
F      Disk2      NTFS    Fixed   Healthy   OK           49.78 GB 49.87 GB
G      Disk3      NTFS    Fixed   Healthy   OK           49.78 GB 49.87 GB
H      Disk4      NTFS    Fixed   Healthy   OK           49.78 GB 49.87 GB
```

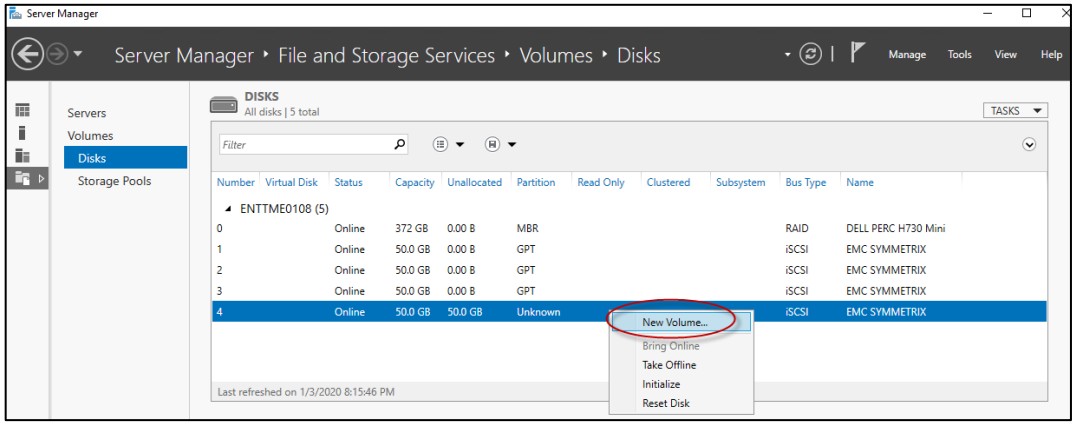
Using Windows Server Manager

With the release of Windows Server 2012, Microsoft introduced Windows Server Manager. Server Manager is Microsoft's preferred way for Windows administrators to manage their Windows Server environments. Prior to Windows 2012, “Disk Manager” was the primary utility to perform volume management operations. Disk manager is still available on Windows platforms, but Server Manager provides a more wizard-driven procedure (New Volume Wizard) to manage volumes on the Windows 2016 host.

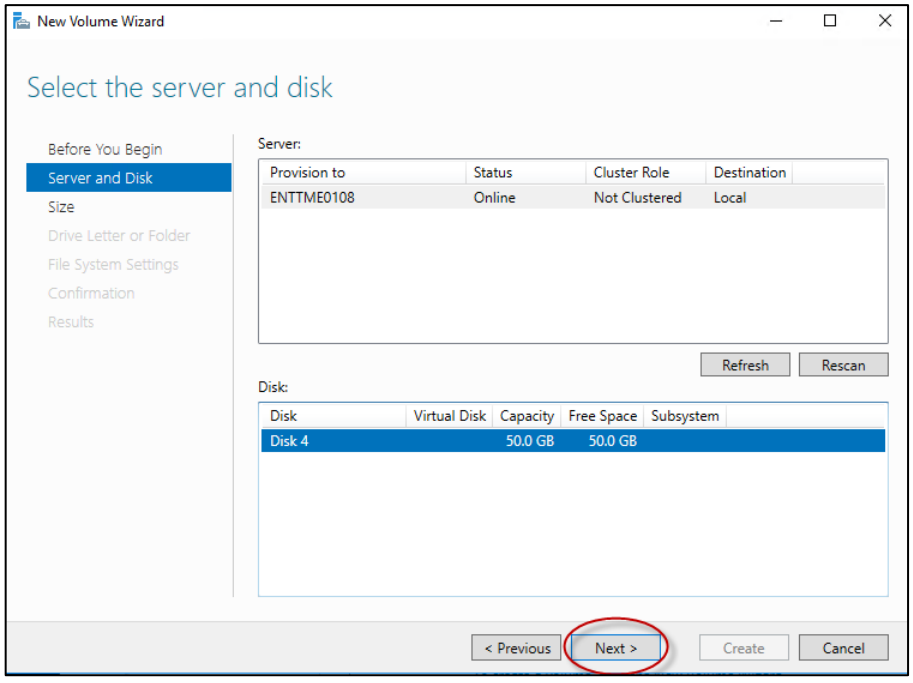
In Server Manager, go to disks, select the offline disk, and click Bring Online.



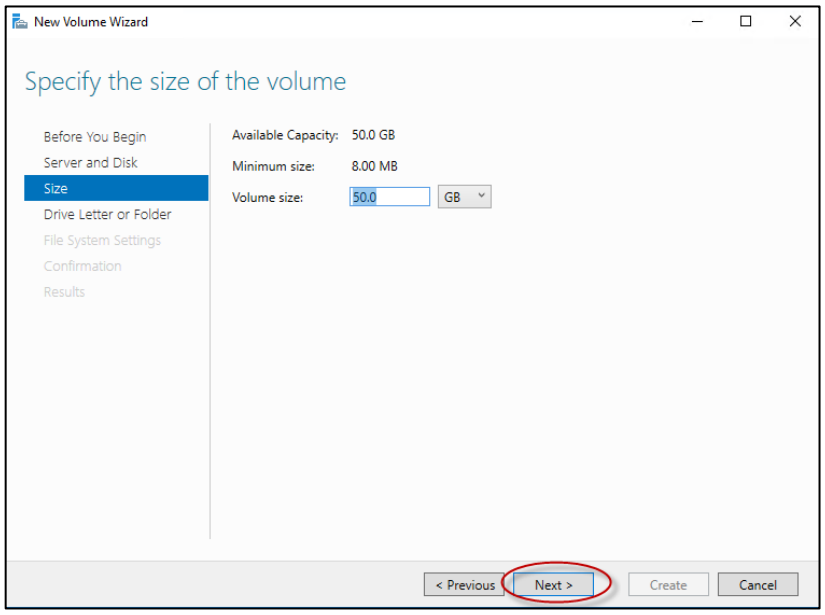
Once the disk is Online, select the disk and launch the New Volume wizard.



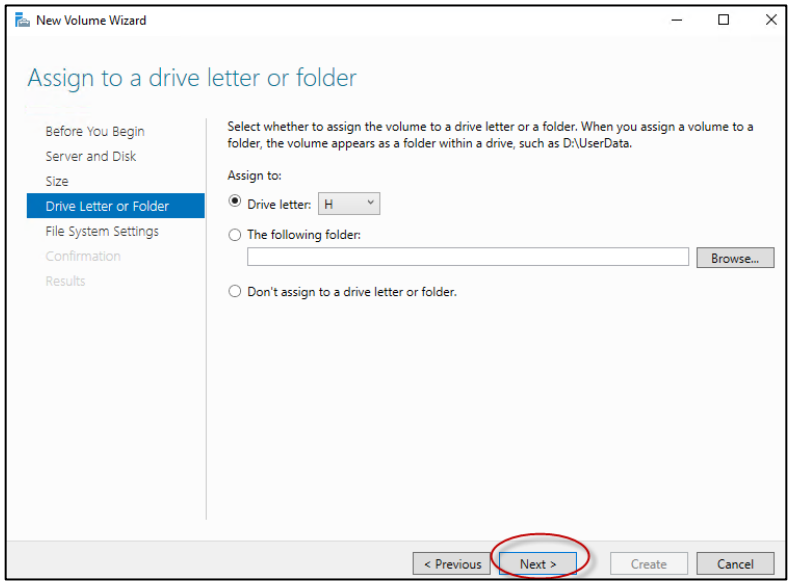
Select disk to create the new volume on, click Next.



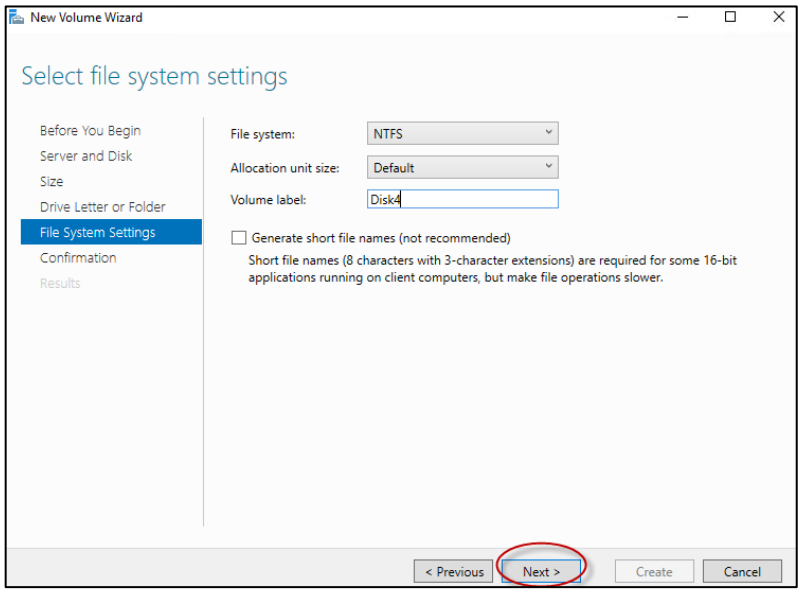
Enter volume size. Default is to use all available capacity. Click Next.



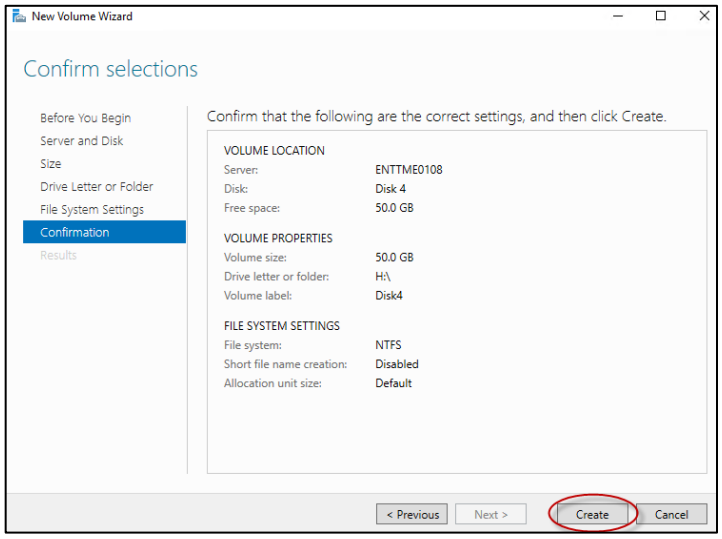
Assign drive letter for new volume. Next available drive letter is default selection. Click Next.



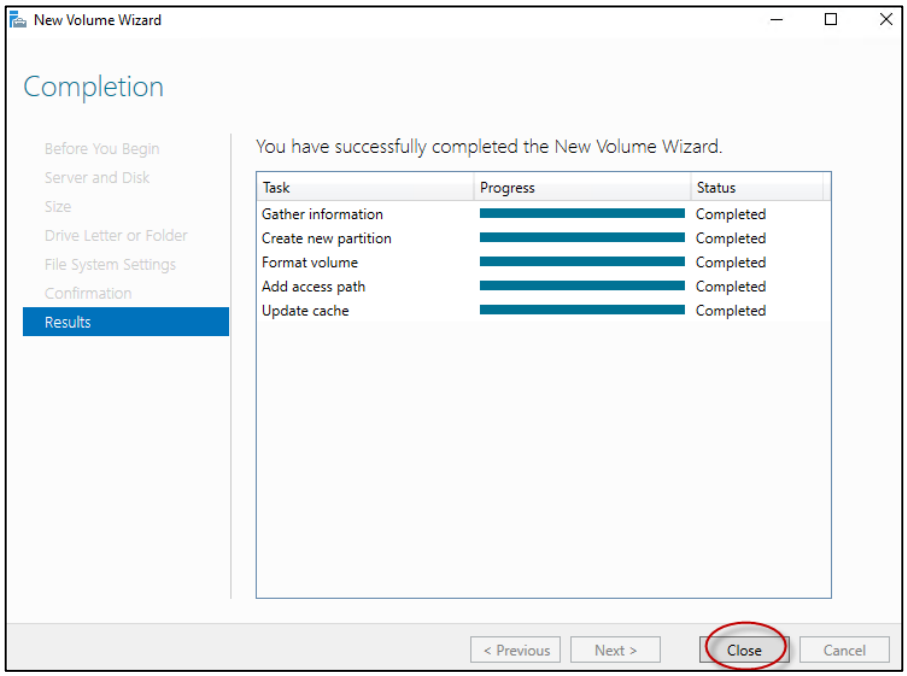
Select file system type (default is NTFS) and specify a volume label. Disk4 is entered as an example. Click Next.



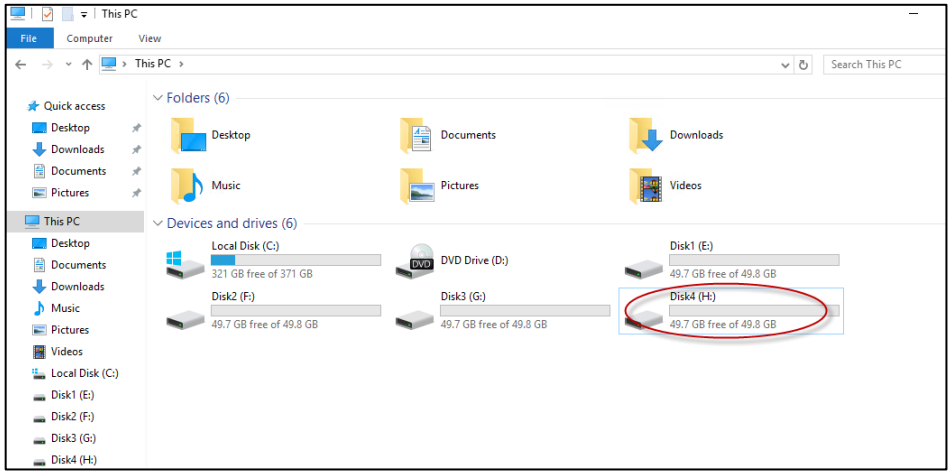
Confirm selections and then click Create.



Monitor volume creation progress. Click close when done.



Verify new volume in Windows Explorer.



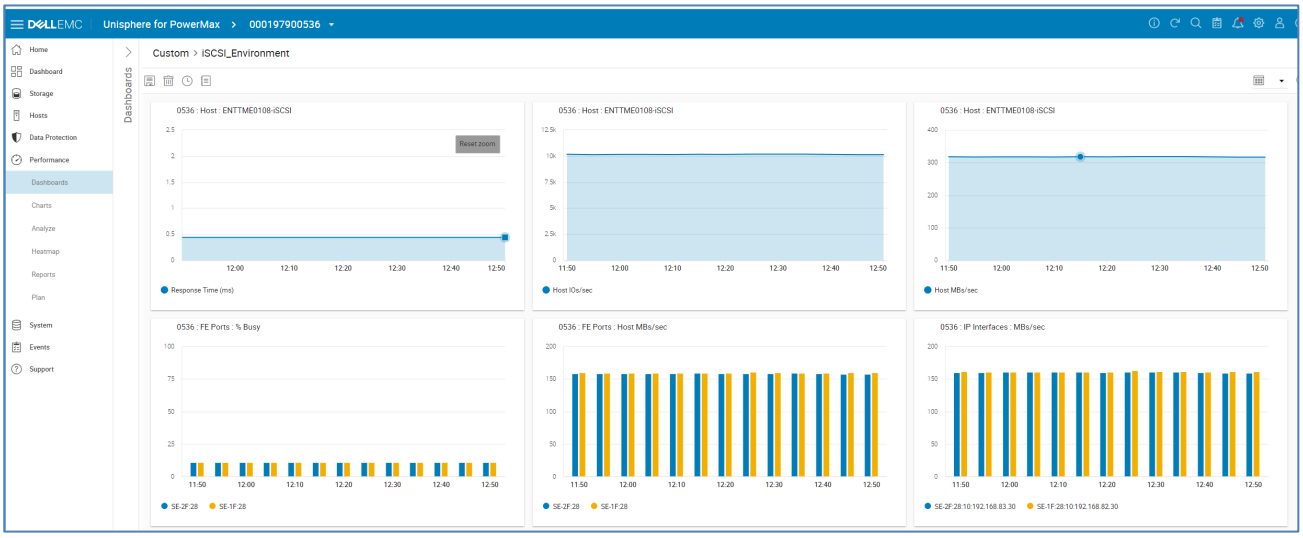
Optional: Send I/O from Prod1 host to PowerMax iSCSI storage

After the iSCSI volumes have been provisioned and acquired by the Prod1 host, it can then send IO to the array. The IO and its performance can be monitored from the array perspective using Unisphere for PowerMax and CloudIQ (if installed). Performance can be monitored from the host perspective using host-based IO tools such as Windows Resource Monitor or the PowerPath (if installed) “powermt display performance dev=all” command.

Unisphere for PowerMax allows users to create custom dashboards to monitor specific hosts or an application's storage group's performance profiles. Alerts can be set up and triggered when thresholds are crossed. A typical iSCSI-based host or application's storage group performance dashboard could contain information around:

- Average response time (millisecond) for the host or application storage group
- Total host or application IOPS being sent to the array
- Throughput (MB/Sec) for the host or application storage group
- Associated SE Port % Busy
- Associated SE Port Throughput (MB/sec)
- Associated SE Port IP Interface Throughput (MB/Sec)

A sample dashboard that includes the above information is shown below:



In the above dashboard, the example's Prod1 host ENTTME0108-iSCSI workload is sending ~ 10000 IOPS and a total ~320 MB/sec of throughput split between the two SE Ports SE-1F:28 and SE-2F:28. This workload is consuming about 10% of the resources (~ 10% Busy) on the SE ports. The associated throughput and for the SE Ports and associated SE Port IP Interfaces is also shown in the dashboard.

Section summary

This section demonstrated how to discover and connect a Windows Server 2016 host or virtual machine to PowerMax iSCSI storage. The PowerMax iSCSI storage provisioned through the targets was acquired and brought online, initialized, and formatted with NTFS file systems. Much of the techniques shown in this section are documented on Microsoft TechNet. See this site for more information about using the Windows iSCSI Initiator Tool or Windows Server Manager.

This document is not intended to be a tutorial on PowerShell. For more detailed information about PowerShell go to Microsoft TechNet or the following websites:

http://www.powershellpro.com/powershell-tutorial-introduction/

https://technet.microsoft.com/en-us/scriptcenter/dd901334.aspx

An excellent book on PowerShell is “Windows PowerShell Step by Step” by Ed Wilson.

Implementing example 2: iSCSI multitenancy

Introduction

This section calls for the expansion of the PowerMax iSCSI target configuration built in the previous section by implementing a second set of targets with their own IP Interfaces (the “Prod2” environment) using the same SE ports as the Prod1 targets and IP interfaces. Implementing the Prod2 iSCSI components alongside the original Prod1 configuration demonstrates the concepts of storage isolation and multitenancy made possible by the PowerMaxOS iSCSI model.

Document the current and desired environments

As said in the earlier section, as a best practice, it is good to create tables and diagrams like the ones shown in this section as it helps a storage administrator keep track of the components and relationships used in the PowerMax iSCSI environment. Although this is an optional step, detailed documentation greatly helps in management and in communicating the environment details to other teams such as the Networking and Database Administrators.

The new Prod2 environment components are detailed in the table and diagram below.

Table 5. Components used by the Prod1 and Prod2 environments

Configuration	PowerMax ID	iSCSI Director	Port	iSCSI Target Name	IP Interface IP Address	Prefix	Network ID	VLAN ID	MTU
Prod1	197900536	SE-1F	28	iqn.dellemc.0536.1F.prod1	192.168.82.30	24	10	82	9000
Prod1	197900536	SE-2F	28	iqn.dellemc.0536.2F.prod1	192.168.83.30	24	10	83	9000
Prod2	197900536	SE-1F	28	iqn.dellemc.0536.1F.prod2	172.16.10.10	24	20	80	9000
Prod2	197900536	SE-2F	28	iqn.dellemc.0536.2F.prod2	172.16.11.10	24	20	80	9000

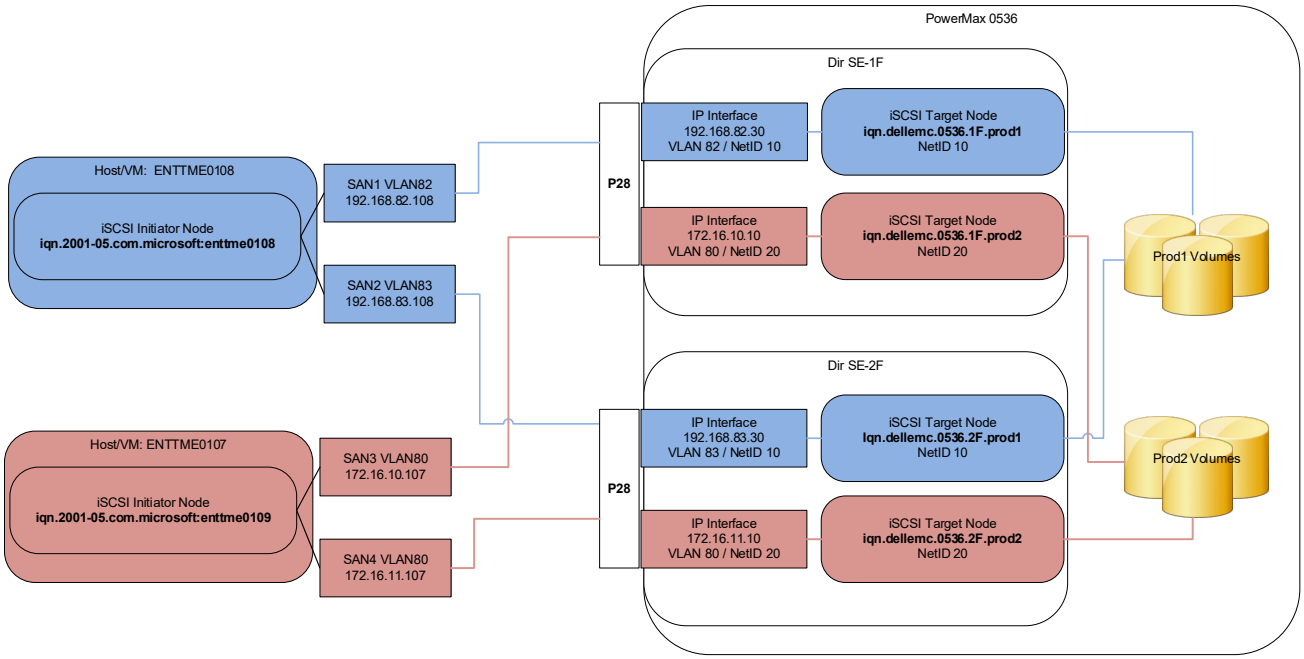


Figure 18. Completed Prod1 and Prod2 Environment Diagram

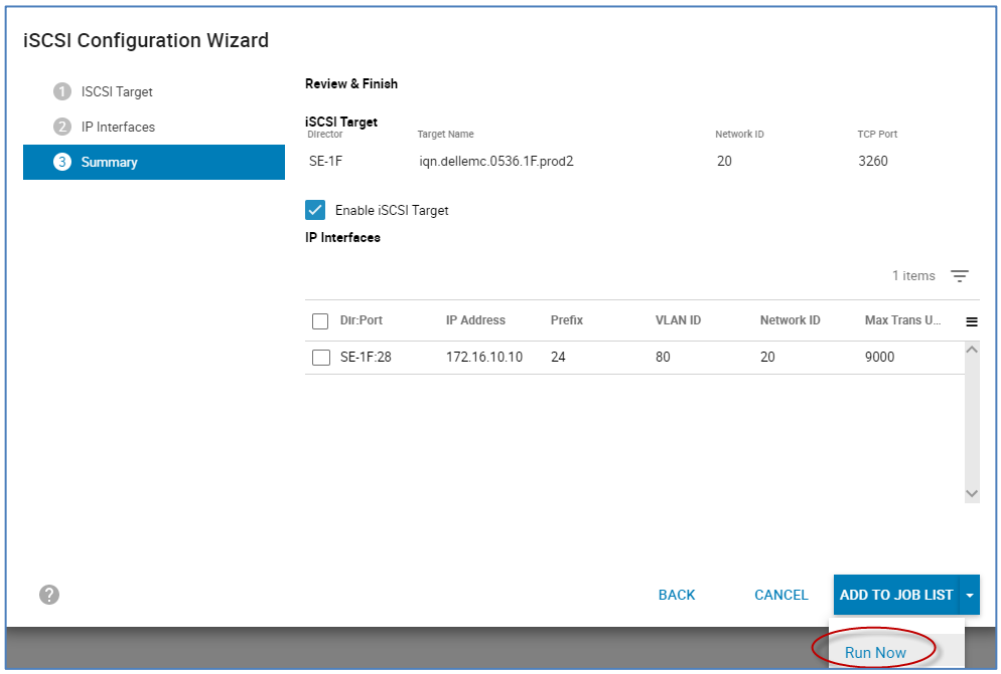
In the example, two additional storage array iSCSI targets are required as well as two additional IP interfaces. The Prod2 environment requires the creation additional VLAN (VLAN 82) to isolate the Prod2 traffic from the Prodd1 traffic. In the example, only a single VLAN will be used for the entire environment. Also, an additional host or virtual machine is required (Windows Server 2016 or Linux) to act as the Prod2 server. In the example, the Prod2 server (ENTTME0107) is running Windows Server 2016.

Create the Prod2 iSCSI configuration

The steps to create the Prod1 components are identical to the steps into create the Prod1 components. These steps will not be detailed in this section as they were earlier. The tool used to create the components in this section is the Unisphere iSCSI Configuration Wizard. Solutions Enabler commands will not be shown. For details on how to use the iSCSI Configuration Wizard and the associated Solutions commands to create IP Interfaces and iSCSI targets refer to the previous section “Implementing Example 1.”

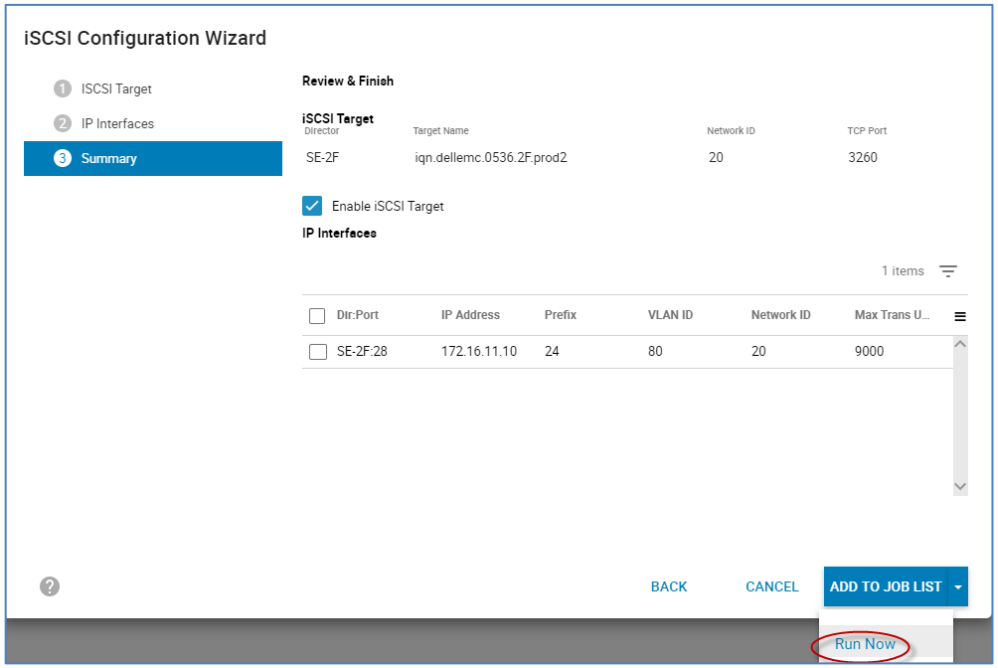
Create the Prod2 target and IP interface on Director 1F

Use the Unisphere iSCSI Configuration Wizard to create the Prod2 iSCSI target and IP interface using director 1F and SE port 28. Review the Prod2 component details as to what values to use. Complete the steps in the wizard, review the summary screen, and then click “Run Now.”



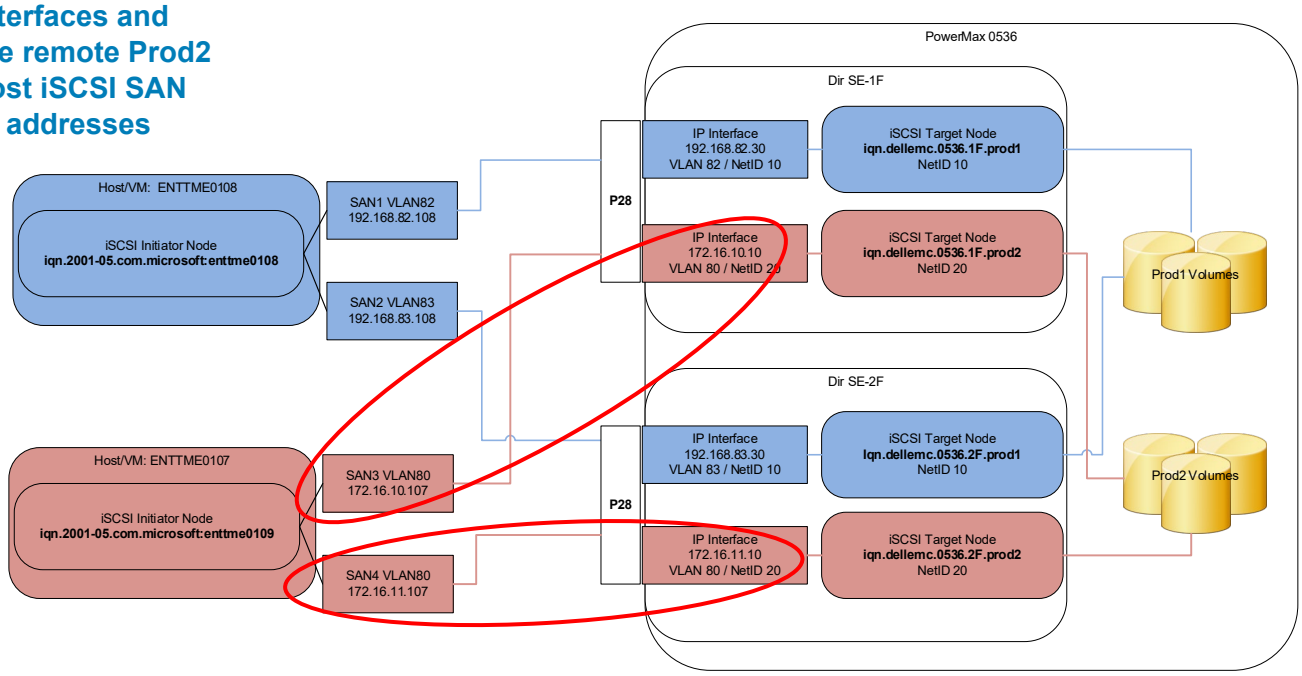
Create the Prod2 target and IP interface on Director 2F

Use the Unisphere iSCSI Configuration Wizard to create the Prod2 iSCSI target and IP interface using director 2F and SE port 28. Review the Prod2 component details as to what values to use. Complete the steps in the wizard, review the summary screen, and then click “Run Now.”

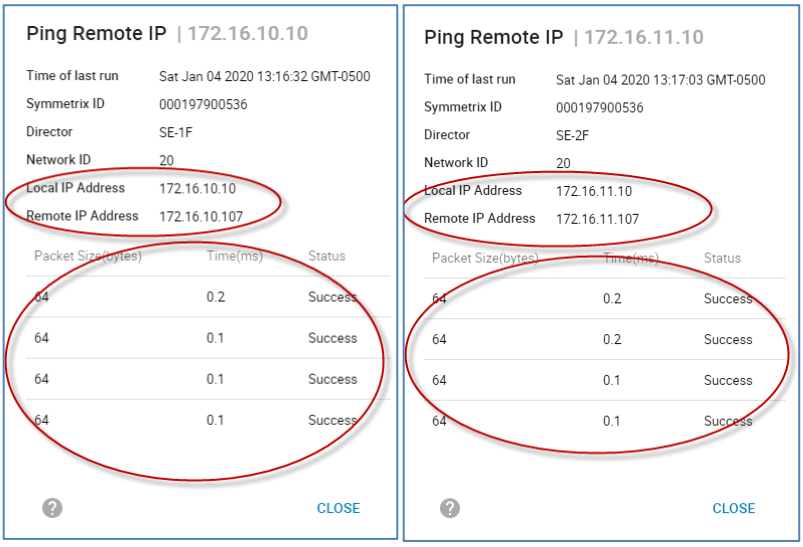


Verify connectivity between the new Prod2 IP Interfaces and the remote Prod2 host iSCSI SAN IP addresses

After the Prod2 components have been successfully created, it is important to verify connectivity between the newly created IP Interfaces and the Prod2 remote host IP addresses.



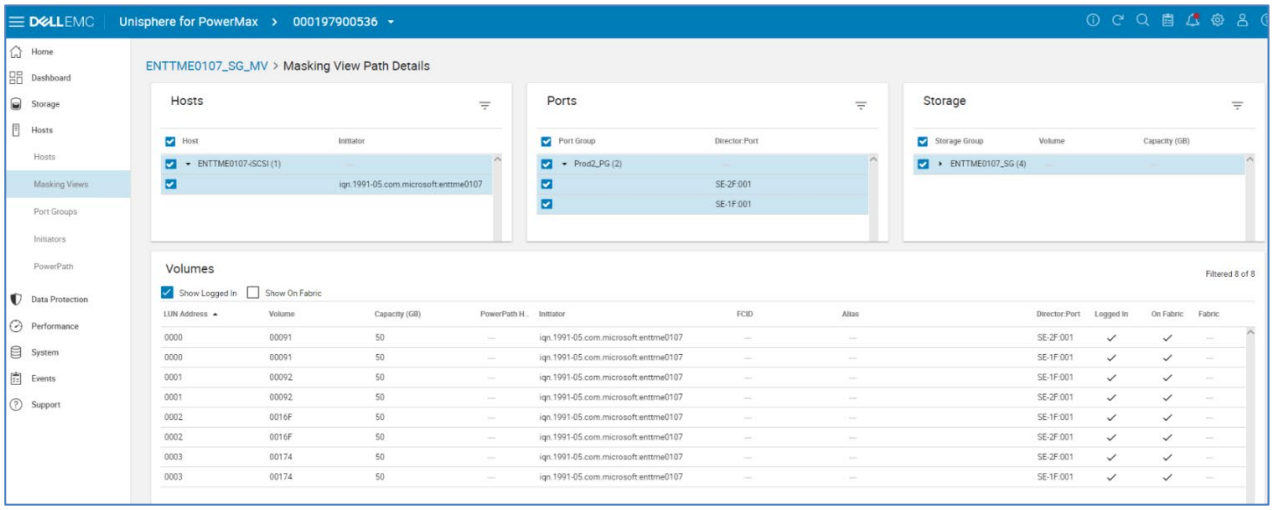
Verify that both Prod2 IP interfaces can ping the associated host SAN network IP address.



Create an iSCSI masking view for the Prod2 host

Create a masking view for the example's Prod2 iSCSI host/VM (ENTTME0109). To do this, repeat the steps documented in Create an iSCSI masking view for the Prod1 host using the Prod2 host and array iSCSI information. Setting up CHAP is optional. The

completed masking view path details for the example's Prod2 host ENTTME0107 looks as follows:



Discover and acquire PowerMax iSCSI storage on the Prod2 host

After the masking view has been created for the example's Prod2 host (ENTTME0108), the storage can be discovered and acquired. Once this is done, the storage can be formatted, and IO be sent. The steps used to accomplish this are detailed in Discover PowerMax iSCSI storage on the host. Repeat these same steps if the host is a Windows 2016 Server. For Linux, consult the various RedHat, CentOS, Ubuntu user documentation for details around how to perform these steps.

The primary point of this section is to demonstrate how two different storage environments can make use of the same SE port resources. This is a key premise behind enabling multitenancy on the PowerMax storage array. Performance charts and user created performance dashboards in Unisphere for PowerMax can easily show this concept in action. The following dashboard shows the two hosts used in the Prod1 and Prod2 examples sending IO to the array. The dashboard shows:

- Average response time (millisecond) for each host
- Total IOPS being sent to the array by each host
- Total Throughput (MB/Sec) being sent to the array by each host
- Prod1 and Prod2 shared SE Port % Busy
- Prod1 and Prod2 shared SE Port Throughput (MB/sec)
- Associated Prod1 and Prod2 SE Port IP Interface Throughput (MB/Sec)

Conclusion

With the proliferation of 10 GbE and now 25 GbE+ networks in enterprise data centers, the use of iSCSI as a primary storage protocol has surged. This paper has demonstrated how iSCSI can be implemented on the PowerMax, highlighting how the PowerMaxOS iSCSI target model is well suited for enterprise environments in which scalability and multitenancy are key design requirements. The key components of the PowerMaxOS iSCSI target model design were explained in detail. These components were shown being used and implemented in real world examples using both Solutions Enabler and Unisphere for PowerMax. This paper also highlighted how to use Windows Server 2016 UI tools and PowerShell cmdlets to acquire PowerMax iSCSI storage. In summary, this paper has attempted to bring together various disparate pieces of knowledge or documentation into a best practices and procedure guide for storage administrators who are considering PowerMax and iSCSI for their storage environment.

Appendix A: Configuring the iSCSI initiator and MPIO on a Windows Server 2016 Host

This section will detail setting up iSCSI and Multipath IO (MPIO) on a Windows Server 2016 host using PowerShell cmdlets. Although this is not a mandatory section, proper iSCSI setup on a host or VM is a critical part of ensuring successful provisioning of HYPERMAX OS iSCSI storage. As said earlier, the host in the example is a Windows Server 2016 using Intel X520 Dual Port 10 GbE NICs with the ProSet driver and tools. The commands listed in this section could be used as a template for a PowerShell script to automatically set up any iSCSI on a Windows Server 2016 host.

Note: This guide will not provide a tutorial on PowerShell syntax. For full detailed information about PowerShell and setting up iSCSI on Windows Server 2016, consult Microsoft TechNet.

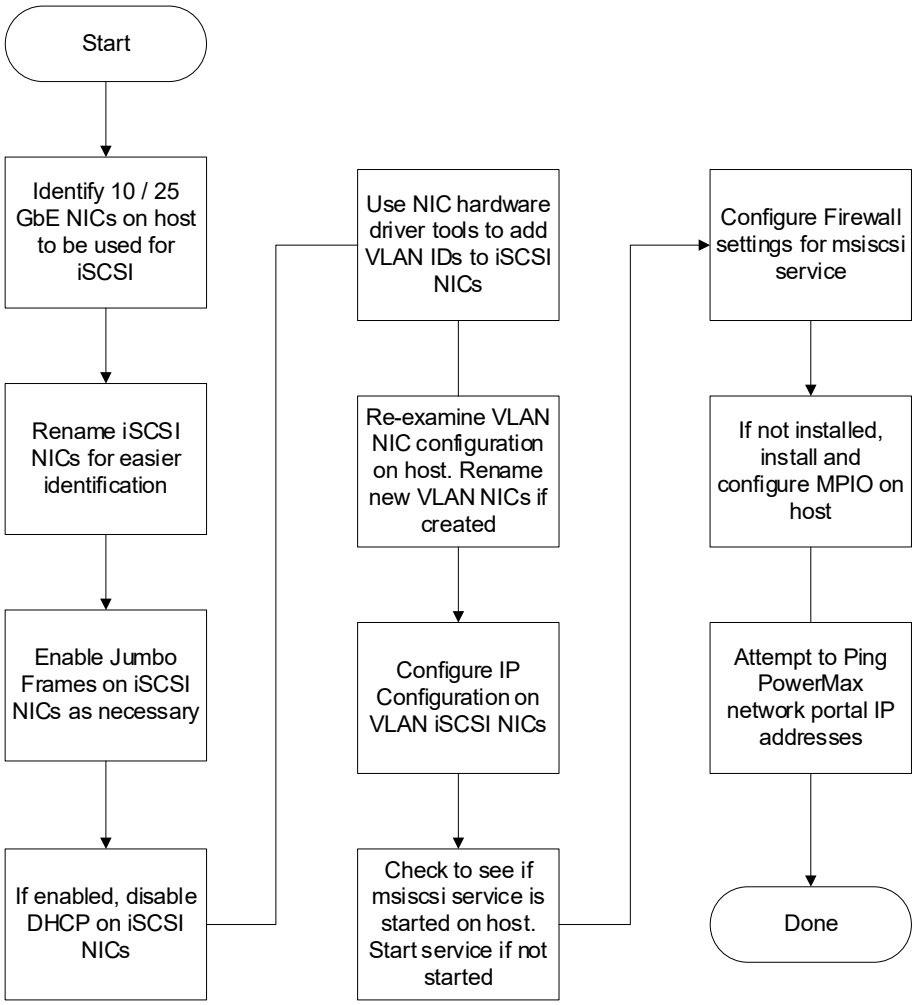


Figure 19. Process Flowchart: Configuring iSCSI and Multipath IO on Windows Server 201

Identify NICs that will be used for iSCSI on host

```
PS C:\> get-netadapter | where-object {$_.status -eq "up"} | ft –autosize
Name    InterfaceDescription          ifIndex Status MacAddress    LinkSpeed
----    --------------------          ------- ------ ----------    ---------
Ethernet 3 Intel(R) Ethernet Server Adapter X520-2   16 Up   90-E2-BA-4C-04-01  10
Gbps <---
Ethernet 4 Intel(R) Ethernet Server Adapter X...#2   17 Up   90-E2-BA-4C-04-00  10
Gbps <---
Ethernet  Cisco 1GigE I350 LOM             12 Up   F8-72-EA-A3-BE-08  1 Gbps
```

Rename iSCSI NICs and LAN NICs for easier identification

```
PS C:\> rename-netadapter -name "Ethernet 3" SAN1
PS C:\> rename-netadapter -name "Ethernet 4" SAN2
PS C:\> rename-netadapter -name "Ethernet" LAN

PS C:\> get-netadapter | where-object {$_.status -eq "up"} | ft –autosize

Name InterfaceDescription          ifIndex Status MacAddress    LinkSpeed
---- --------------------          ------- ------ ----------    ---------
SAN1 Intel(R) Ethernet Server Adapter X520-2   16 Up   90-E2-BA-4C-04-01  10 Gbps
SAN2 Intel(R) Ethernet Server Adapter X...#2   17 Up   90-E2-BA-4C-04-00  10 Gbps
LAN Cisco 1GigE I350 LOM             12 Up   F8-72-EA-A3-BE-08  1 Gbps
```

Enable jumbo frames on iSCSI NICs if supported on network

Jumbo frames have shown a boost in iSCSI performance for small block reads and writes by as much as 20%. Jumbo frames do not provide as much benefit for iSCSI performance for larger block sizes (>64K). Always configure Jumbo Frames according to the NIC manufacturer instructions. Configure Jumbo Frames end-to-end on the network, with the smallest setting governing the end-to-end packet size. Enabling jumbo frames and Flow Control on the host NICs depends on the network configuration. If these features are not enabled on the network, then enabling them for the host NICs will provide no benefit. Before enabling these features on the host NICs, consult with the local network administrators and see if Jumbo Frames has been enabled on the network.

Examine jumbo frame settings on iSCSI NICs:

```
PS C:\> get-netadapteradvancedproperty –name SAN* -displayname "Jumbo
Packet","Flow Control" | ft –property Name, Displayname, Displayvalue,
validdisplayvalues -AutoSize
Name Displayname Displayvalue  validdisplayvalues
---- ----------- ------------  ------------------
SAN1 Flow Control Rx & Tx Enabled {Disabled, Tx Enabled, Rx Enabled, Rx & Tx
Enabled} <--- Flow control enabled
```

```
SAN1 Jumbo Packet Disabled    {Disabled, 4088 Bytes, 9014 Bytes}          <---
Jumbo Frames disabled
SAN2 Flow Control Rx & Tx Enabled {Disabled, Tx Enabled, Rx Enabled, Rx & Tx
Enabled}
SAN2 Jumbo Packet Disabled    {Disabled, 4088 Bytes, 9014 Bytes}
```

Enable jumbo frames on iSCSI NICs.

```
PS C:\> set-netadapteradvancedproperty -name SAN* -displayname "Jumbo Packet" -
DisplayValue "9014 Bytes"
```

Verify that jumbo frames have been enabled on iSCSI NICs.

Note: The valid display values shown in the output of the get-netadapteradvancedproperty command will vary depending upon the device driver provided by the NIC manufacturer. For example, Broadcom NICs use a jumbo frame value of “9014” while Intel NICs will use a value of “9014 Bytes.”

```
PS C:\> get-netadapteradvancedproperty -name SAN* -displayname "Jumbo Packet" | ft
-property Name, Displayname, Displayvalue, validdisplayvalues -AutoSize
Name Displayname Displayvalue validdisplayvalues
---- ----------- ------------ ------------------
SAN1 Jumbo Packet 9014 Bytes  {Disabled, 4088 Bytes, 9014 Bytes} <--- Jumbo Frames
Enabled
SAN2 Jumbo Packet 9014 Bytes  {Disabled, 4088 Bytes, 9014 Bytes} <--- Jumbo Frames
Enabled
```

Optional: If enabled, disable DHCP on iSCSI NICs

Verify if DHCP is enabled on iSCSI NICs:

```
PS C:\> Get-NetAdapter -Name SAN* | Get-NetIPConfiguration -
Detailed
ComputerName        : ENTTME0108
InterfaceAlias       : SAN1
InterfaceIndex       : 16
InterfaceDescription    : Intel(R) Ethernet Server Adapter X520-2
NetAdapter.LinkLayerAddress : 90-E2-BA-4C-04-01
NetAdapter.Status      : Up
NetProfile.Name       : Unidentified network
NetProfile.NetworkCategory : Public
NetProfile.IPv6Connectivity : NoTraffic
NetProfile.IPv4Connectivity : LocalNetwork
IPv6LinkLocalAddress    : fe80::1f3:4489:cc1:2928%16
IPv4Address         : 169.254.41.40
IPv6DefaultGateway     :
IPv4DefaultGateway     :
NetIPv6Interface.NlMTU   : 9000
NetIPv4Interface.NlMTU   : 9000
NetIPv6Interface.DHCP    : Enabled <----
NetIPv4Interface.DHCP    : Enabled <----
DNSServer          : fec0:0:0:ffff::1
```

```
                fec0:0:0:ffff::2
                fec0:0:0:ffff::3
ComputerName         : ENTTME0108
InterfaceAlias       : SAN2
InterfaceIndex       : 17
InterfaceDescription    : Intel(R) Ethernet Server Adapter X520-2 #2
NetAdapter.LinkLayerAddress : 90-E2-BA-4C-04-00
NetAdapter.Status      : Up
NetProfile.Name       : Unidentified network
NetProfile.NetworkCategory : Public
NetProfile.IPv6Connectivity : NoTraffic
NetProfile.IPv4Connectivity : LocalNetwork
IPv6LinkLocalAddress    : fe80::b1f0:78da:cdeb:a441%17
IPv4Address          : 169.254.164.65
IPv6DefaultGateway     :
IPv4DefaultGateway     :
NetIPv6Interface.NlMTU   : 9000
NetIPv4Interface.NlMTU   : 9000
NetIPv6Interface.DHCP    : Enabled <----
NetIPv4Interface.DHCP    : Enabled <----
DNSServer           : fec0:0:0:ffff::1
                fec0:0:0:ffff::2
                fec0:0:0:ffff::3
```

If DHCP is enabled, disable DHCP:

```
PS C:\> Get-NetAdapter -Name SAN* | Set-NetIPInterface -Dhcp
Disabled
```

Verify that DHCP has been disabled on iSCSI NICs:

```
PS C:\> Get-NetAdapter -Name SAN* | Get-NetIPConfiguration -
Detailed
ComputerName         : ENTTME0108
InterfaceAlias       : SAN1
InterfaceIndex       : 16
InterfaceDescription    : Intel(R) Ethernet Server Adapter X520-2
NetAdapter.LinkLayerAddress : 90-E2-BA-4C-04-01
NetAdapter.Status      : Up
NetProfile.Name       : Unidentified network
NetProfile.NetworkCategory : Public
NetProfile.IPv6Connectivity : NoTraffic
NetProfile.IPv4Connectivity : LocalNetwork
IPv6LinkLocalAddress    : fe80::1f3:4489:cc1:2928%16
IPv4Address          : 169.254.41.40
IPv6DefaultGateway     :
IPv4DefaultGateway     :
NetIPv6Interface.NlMTU   : 9000
NetIPv4Interface.NlMTU   : 9000
NetIPv6Interface.DHCP    : Disabled <----
NetIPv4Interface.DHCP    : Disabled <----
```

```
DNSServer          : fec0:0:0:ffff::1
                     fec0:0:0:ffff::2
                     fec0:0:0:ffff::3
ComputerName       : ENTTME0108
InterfaceAlias     : SAN2
InterfaceIndex     : 17
InterfaceDescription    : Intel(R) Ethernet Server Adapter X520-2 #2
NetAdapter.LinkLayerAddress : 90-E2-BA-4C-04-00
NetAdapter.Status       : Up
NetProfile.Name       : Unidentified network
NetProfile.NetworkCategory : Public
NetProfile.IPv6Connectivity : NoTraffic
NetProfile.IPv4Connectivity : LocalNetwork
IPv6LinkLocalAddress    : fe80::b1f0:78da:cdeb:a441%17
IPv4Address         : 169.254.164.65
IPv6DefaultGateway     :
IPv4DefaultGateway     :
NetIPv6Interface.NlMTU   : 9000
NetIPv4Interface.NlMTU   : 9000
NetIPv6Interface.DHCP    : Disabled <----
NetIPv4Interface.DHCP    : Disabled <----
DNSServer          : fec0:0:0:ffff::1
                     fec0:0:0:ffff::2
                     fec0:0:0:ffff::3
```

Use NIC hardware driver tools to add VLAN IDs to iSCSI NICs

The host used in the example uses Intel X-520 Dual Port 10 GbE NICs. These NICs use Intel's ProSet Driver. This driver comes with a PowerShell toolkit that can be used to add VLANs to the NICs. Each NIC vendor will have different tool kits included with their drivers. Be sure to consult the vendor documentation on how to add VLANs to that vendor's NICs. This next section will use the ProSet PowerShell command set to perform this task.

Ensure that the identical VLAN information is used across the entire configuration—from the PowerMax iSCSI IP Interfaces, through the switch infrastructure (refer to your switch vendor documentation on setting up VLANs on switch ports) to the host NICs.

Run the get-intelnetadapter command to detail the host NICs from the Intel driver point of view:

```
PS C:\> get-intelnetadapter
Location    Name                    ConnectionName    LinkStatus
--------    ----                      --------------   ----------
1:0:0:0     Cisco 1GigE I350 LOM           LAN          1.00 Gbps/...
1:0:1:0     Cisco 1GigE I350 LOM #2          Ethernet 2      Not Available
3:0:0:0     Intel(R) Ethernet Server Adapter X520   SAN2         10.00 Gbps... <--
-
3:0:1:0     Intel(R) Ethernet Server Adapter X520   SAN1         10.00 Gbps... <--
-
```

Add VLAN 82 to NIC "SAN1":

```
PS C:\> get-intelnetadapter | ? {$_.connectionname -eq "SAN1"} | Add-IntelNetVLAN
-VLANID 82
VLANID VLANName                    ParentName
------ --------                    ----------
82   VLAN82                     Intel(R) Ethernet Server Adapter X520-2
```

Add VLAN 83 to NIC "SAN2":

```
PS C:\> get-intelnetadapter | ? {$_.connectionname -eq "SAN2"} | Add-IntelNetVLAN
-VLANID 83
VLANID VLANName                    ParentName
------ --------                    ----------
83   VLAN83                     Intel(R) Ethernet Server Adapter X520-2 #2
```

Examine the VLANs for the iSCSI NICs to ensure they were properly created:

```
PS C:\> Get-IntelNetVLAN
VLANID VLANName ParentName
------ -------- ----------
83   VLAN83  Intel(R) Ethernet Server Adapter X520-2 #2
82   VLAN82  Intel(R) Ethernet Server Adapter X520-2
```

Reexamine VLAN NICs on host

Examine the NIC instances the driver created for each VLAN. The Intel ProSet driver creates a new instance of the physical NIC that the VLAN is assigned to. It will create a new NIC instance for each VLAN assigned. The new NIC VLAN instance assumes all the parameters of the parent NIC instance, including jumbo frame settings; however, the new VLAN NIC is created with its own name, interface index, and interface description.

```
PS C:\> Get-NetAdapter | ft -AutoSize
Name    InterfaceDescription              ifIndex Status MacAddress    LinkSpeed
----    --------------------              ------- ------ ----------    ---------
Ethernet 3 Intel(R) Ethernet ... X520-2 #2 - VLAN : VLAN83 29   Up   90-E2-BA-4C-
04-00  10 Gbps <----
Ethernet  Intel(R) Ethernet ... X520-2 - VLAN : VLAN82  33   Up   90-E2-BA-4C-04-
01  10 Gbps <----
SAN1    Intel(R) Ethernet ... X520-2          16   Up   90-E2-BA-4C-04-01  10 Gbps
SAN2    Intel(R) Ethernet ... X...#2          17   Up   90-E2-BA-4C-04-00  10 Gbps
LAN    Cisco 1GigE I350 LOM              12   Up   F8-72-EA-A3-BE-08  1 Gbps
PS C:\> Get-IntelNetVLANJumboPacket
ParentName                   VLANID DisplayValue RegistryValue
----------                   ------ ------------ -------------
Intel(R) Ethernet Server Adapter X520-2 #2   83   9014 Bytes  9014
Intel(R) Ethernet Server Adapter X520-2    82   9014 Bytes  9014
```

Rename VLAN NIC instances for easier identification

```
PS C:\> Rename-NetAdapter -name "Ethernet" "SAN1 VLAN82"
PS C:\> Rename-NetAdapter -name "Ethernet 3" "SAN2 VLAN83"
PS C:\> Get-NetAdapter | ft -AutoSize

Name    InterfaceDescription                    ifIndex Status    MacAddress
LinkSpeed
```

```
----    --------------------                    ------- ------    ----------    --
-------
SAN2 VLAN83 Intel(R) Ethernet Server Adapter X520-2 #2 - VLAN : VLAN83   29 Up
90-E2-BA-4C-04-00  10 Gbps <----
SAN1 VLAN82 Intel(R) Ethernet Server Adapter X520-2 - VLAN : VLAN82     33 Up
90-E2-BA-4C-04-01  10 Gbps <----
SAN1    Intel(R) Ethernet Server Adapter X520-2             16 Up      90-E2-BA-
4C-04-01  10 Gbps
SAN2    Intel(R) Ethernet Server Adapter X...#2             17 Up      90-E2-BA-
4C-04-00  10 Gbps
Ethernet 2 Cisco 1GigE I350 LOM #2                    13 Disconnected F8-72-EA-
A3-BE-09   0 bps
LAN     Cisco 1GigE I350 LOM                       12 Up      F8-72-EA-A3-BE-08  1
Gbps
```

Configure IP address and subnet information for VLAN NICs

In the get-netadapter output, identify the VLAN NICs and make a note of their interface indexes (ifIndex). Use the new-netipaddress command to set the IP address and subnet specifying the ifindex for each VLAN NIC. In the example, the subnet prefix of 24 (255.255.255.0) is used. Again, be sure that the network configuration information is consistent across the entire iSCSI infrastructure (PowerMax iSCSI IP interfaces, switch ports, host NICs).

```
PS C:\> New-NetIPAddress -InterfaceIndex 33 -IPAddress
192.168.82.108 -PrefixLength 24
IPAddress      : 192.168.82.108
InterfaceIndex  : 33
InterfaceAlias  : SAN1 VLAN82
AddressFamily   : IPv4
Type       : Unicast
PrefixLength   : 24
PrefixOrigin   : Manual
SuffixOrigin   : Manual
AddressState   : Tentative
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : ActiveStore
IPAddress      : 192.168.82.108
InterfaceIndex  : 33
InterfaceAlias  : SAN1 VLAN82
AddressFamily   : IPv4
Type       : Unicast
PrefixLength   : 24
PrefixOrigin   : Manual
SuffixOrigin   : Manual
AddressState   : Invalid
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : PersistentStore
```

```
PS C:\> New-NetIPAddress -InterfaceIndex 29 -IPAddress
192.168.83.108 -PrefixLength 24
IPAddress      : 192.168.83.108
InterfaceIndex  : 29
InterfaceAlias  : SAN2 VLAN83
AddressFamily   : IPv4
Type       : Unicast
PrefixLength   : 24
PrefixOrigin   : Manual
SuffixOrigin   : Manual
AddressState   : Tentative
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : ActiveStore
IPAddress      : 192.168.83.108
InterfaceIndex  : 29
InterfaceAlias  : SAN2 VLAN83
AddressFamily   : IPv4
Type       : Unicast
PrefixLength   : 24
PrefixOrigin   : Manual
SuffixOrigin   : Manual
AddressState   : Invalid
ValidLifetime   : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : PersistentStore
```

Verify network connectivity to POWERMAX IP interfaces

Before installing MPIO or configuring the iSCSI Services on the host, it is important to verify network connectivity to the POWERMAX IP interface. This can be done simply by pinging the IP addresses of the POWERMAX IP interfaces.

```
PS C:\> ping 192.168.83.30

Pinging 192.168.83.30 with 32 bytes of data:
Reply from 192.168.83.30: bytes=32 time<1ms TTL=64
Reply from 192.168.83.30: bytes=32 time<1ms TTL=64
PS C:\> ping 192.168.82.30

Pinging 192.168.82.30 with 32 bytes of data:
Reply from 192.168.82.30: bytes=32 time<1ms TTL=64
Reply from 192.168.82.30: bytes=32 time<1ms TTL=64
```

Verify that the jumbo frames can be transmitted using a ping on Windows. The 28 Bytes of overhead must be subtracted from the 9000 MTU value, therefore the size of the test packet needs to be 8972 Bytes (Size of packet = 9000 Byte MTU 28 Bytes overhead = 8972).

```
PS C:\> ping -f -l 8972 192.168.82.30
```

```
Pinging 192.168.82.30 with 8972 bytes of data:
Reply from 192.168.82.30 : bytes=8972 time=1ms TTL=255
Reply from 192.168.82.30: bytes=8972 time<1ms TTL=255
PS C:\> ping -f -l 8972 192.168.83.30

Pinging 192.168.82.30 with 8972 bytes of data:
Reply from 192.168.83.30: bytes=8972 time=1ms TTL=255
Reply from 192.168.83.30: bytes=8972 time<1ms TTL=255
```

If for some reason the jumbo packet fragments in the above ping command as shown below, recheck to ensure that jumbo packets are enabled across the entire iSCSI network including the POWERMAX IP interfaces, all switch ports (all hops), and host NICs.

```
PS C:\> ping -f -l 8972 192.169.82.30

Pinging 192.168.82.30 with 8972 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
```

At this point, the network connectivity to the Management operating system partition on the Hyper-V host is complete. The virtual switches have been created and the vNICs have successfully assumed the IP configuration of the physical NICs.

Verify the Microsoft iSCSI Initiator (MSiSCSI) service is started on the host

The Microsoft iSCSI Initiator Service (MSiSCSI) is responsible for accessing iSCSI devices. It manages the iSCSI sessions from the host initiators to the remote iSCSI target devices. If this service is stopped, the host will not be able to log in or access iSCSI targets. If this service is disabled, any services that explicitly depend on it will fail to start.

Examine MSiSCSI service on Windows host to see start mode and running state. For detailed information about this service, consult Microsoft TechNet.

```
PS C:\> get-wmiobject -class win32_service | ? {$_.name -eq
"msiscsi"}
ExitCode : 1077
Name   : MSiSCSI
ProcessId : 0
StartMode : Manual  <----
State   : Stopped <----
Status  : OK
```

If the service is not running, start MSiSCSI service and set start mode to auto.

```
PS C:\> set-service -name msiscsi -status running -startuptype
auto
```

Verify MSiSCSI service has started and start mode has been set to auto.

```
PS C:\> get-wmiobject -class win32_service | ? {$_.name -eq
"msiscsi"}
ExitCode : 0
Name   : MSiSCSI
ProcessId : 200
```

```
StartMode : Auto    <----
State     : Running <----
Status    : OK
```

Configure Windows firewall settings for the MSiSCSI service

In order for the MSiSCSI service to function properly, the Windows firewall settings need to be modified to allow for incoming and outgoing iSCSI traffic.

Examine the firewall rules pertaining to the MSiSCSI service. In the output, Enabled should be set to "true" for both msiscsi-in-tcp and msiscsi-out-tcp rules.

```
PS C:\> get-netfirewallservicefilter -Service msiscsi | get-netfirewallrule
Name          : MsiScsi-In-TCP
DisplayName       : iSCSI Service (TCP-In)
Description       : Inbound rule for the iSCSI Service to allow communications with
an iSCSI server or device. [TCP]
DisplayGroup      : iSCSI Service
Group         : @FirewallAPI.dll,-29002
Enabled         : False <-----
...
Name          : MsiScsi-Out-TCP
DisplayName       : iSCSI Service (TCP-Out)
Description       : Outbound rule for the iSCSI Service to allow communications
with an iSCSI server or device. [TCP]
DisplayGroup      : iSCSI Service
Group         : @FirewallAPI.dll,-29002
Enabled         : False <-----
...
```

If either firewall rule is not enabled, enable them with the following commands.

```
PS C:\> set-NetFirewallRule -Name MsiScsi-in-TCP -Enabled True
PS C:\> set-NetFirewallRule -Name MsiScsi-out-TCP -Enabled True
```

Examine the firewall rules again to ensure that they are now enabled.

```
PS C:\> get-netfirewallservicefilter -Service msiscsi | get-netfirewallrule
Name          : MsiScsi-In-TCP
DisplayName       : iSCSI Service (TCP-In)
Description       : Inbound rule for the iSCSI Service to allow communications with
an iSCSI server or device. [TCP]
DisplayGroup      : iSCSI Service
Group         : @FirewallAPI.dll,-29002
Enabled         : True <-----
...
Name          : MsiScsi-Out-TCP
DisplayName       : iSCSI Service (TCP-Out)
Description       : Outbound rule for the iSCSI Service to allow communications
with an iSCSI server or device. [TCP]
DisplayGroup      : iSCSI Service
Group         : @FirewallAPI.dll,-29002
Enabled         : True <-----
...
```

If not already installed, install multipathing software on Windows host

Install multipathing software such as PowerPath or Microsoft Multipath I/O (MPIO). There currently are published guides on how to install PowerPath on Windows platforms so this section will focus on installing Window's native Multipath I/O utility (MPIO). MPIO is an optional feature in Windows Server 2016, and is not installed by default. The following section will demonstrate how to install MPIO on a running Windows Server 2016 host.

Check to see if Multipath-IO is installed or available to be installed on the server:

```
PS C:\> Get-WindowsFeature -name multipath-io
Display Name      Name              Install State
------------      -------------                -------------
[ ] Multipath I/O   MultipathIO          Available <----
```

If not "Installed," install Multipath IO:

Use the "install-WindowsFeature" cmdlet to install the Multipath IO feature.

```
PS C:\> Install-WindowsFeature -name multipath-io

Success Restart Needed Exit Code   Feature Result
------- -------------- ---------   --------------
True  No       Success    {Multipath I/O}
```

Verify that Multipath-IO now shows up in the installed Windows features output.

```
PS C:\> Get-WindowsFeature | ? installed
Display Name                        Name            Install State
------------                        ----            -------------
[X] File and Storage Services              FileAndStorage-Services    Installed
  [X] Storage Services                  Storage-Services        Installed
[X] .NET Framework 4.5 Features             NET-Framework-45-Fea...    Installed
  [X] .NET Framework 4.5                NET-Framework-45-Core     Installed
  [X] WCF Services                    NET-WCF-Services45       Installed
    [X] TCP Port Sharing                NET-WCF-TCP-PortShar...    Installed
[X] Multipath I/O                     Multipath-IO          Installed <----
[X] SMB 1.0/CIFS File Sharing Support          FS-SMB1            Installed
[X] User Interfaces and Infrastructure         User-Interfaces-Infra     Installed
  [X] Graphical Management Tools and Infrastructure  Server-Gui-Mgmt-Infra
Installed
  [X] Server Graphical Shell              Server-Gui-Shell        Installed
[X] Windows PowerShell                  PowerShellRoot         Installed
  [X] Windows PowerShell 4.0              PowerShell           Installed
  [X] Windows PowerShell ISE              PowerShell-ISE         Installed
[X] WoW64 Support                     WoW64-Support         Installed
```

Set MPIO to automatically claim new iSCSI devices when they are presented to the host:

```
PS C:\> Enable-MSDSMAutomaticClaim -BusType iSCSI
PS C:\> Get-MSDSMSupportedHW
VendorId ProductId
-------- ---------
```

```
Vendor 8 Product    16
MSFT2005 iSCSIBusType_0x9 <----
```

Set MPIO load-balancing policy to be round robin (RR):

```
PS C:\> Set-MSDSMGlobalDefaultLoadBalancePolicy -Policy RR
```

Optional: Discover and attempt to connect to the PowerMax IP interfaces

The final step in configuring the Windows host for iSCSI is to discover and attempt to connect to the PowerMax using the new-iscsitargetportal PowerShell command specifying the IP addresses of the PowerMax IP interfaces. The first time this command is run, it might fail as a successful connection to the storage array iSCSI target portals requires either:

- The host initiator uses the proper CHAP credentials required by the storage array.
- A masking view created using the IQN of the host initiator in the initiator group.

Even though the new-iscsitargetportal command most likely will result in an error the first time it is run from the host, its execution will enter the IQN in the login history table on the PowerMax. By performing a "symaccess list logins -v" command, a storage administrator will be able to harvest the host IQN from the login history for the iSCSI target. The storage administrator will then be able to use the host IQN in an initiator group for the initial masking being presented to the host. Once the masking view for the host has been created, the host will be able to rerun the new-iscsitargetportal command and connect to the PowerMax IP Interface.

```
PS C:\> New-IscsiTargetPortal -TargetPortalAddress 192.168.82.30

New-IscsiTargetPortal : Authorization Failure. <---- Improper CHAP
credential or no masking view created
...

PS C:\> New-IscsiTargetPortal -TargetPortalAddress 192.168.83.30

New-IscsiTargetPortal : Authorization Failure. <---- Improper CHAP
credential or no masking view created
```

Even though the command failed, the IQN from the host has registered in the PowerMax login history tables as shown below:

```
PS C:\> symaccess -sid 0536 list logins -v
Symmetrix ID      : 000197900536
...
Director Identification : SE-1F
Director Port      : 000
iSCSI Target Name    : iqn.dellemc.0536.1F.prod1
  ...
  Originator Node wwn : N/A
  Originator Port wwn : N/A
  iSCSI Name : iqn.1991-05.com.microsoft:ENTTME0108 <----
  ip Address     : 192.168.83.108
  Type        : iSCSI
```

```
  User-generated Name : /
  FCID        : N/A
  Logged In       : No <----
  On Fabric       : Yes
  Last Active Log-In : 11:26:42 AM on Wed Jun 24,2015
Director Identification : SE-2F
Director Port       : 000
iSCSI Target Name     : iqn.dellemc.0536.2F.prod1
  ...

  Originator Node wwn : N/A
  Originator Port wwn : N/A
  iSCSI Name : iqn.1991-05.com.microsoft:ENTTME0108 <---
  ip Address      : 192.168.83.108
  Type        : iSCSI
  User-generated Name : /
  FCID        : N/A
  Logged In       : No <----
  On Fabric       : Yes
  Last Active Log-In : 11:26:42 AM on Wed Jun 24,2015
```

At this point, the host initiator will have logged its IQN into the login history table of the PowerMax. The storage administrator can then begin creating a masking view as described in Create an iSCSI masking view for the Prod1 host.

Appendix B: Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage and data protection technical white papers and videos provide expertise that helps to ensure customer success with Dell EMC storage and data protection products.

The following table lists related resources:

Table 6. Related resources

Document title	Collateral type	Part number
Dell EMC PowerMax Family Overview	White Paper	H17118
Dell EMC VMAX All Flash Family Overview	White Paper	H14920.3
Dell EMC Service Levels for PowerMaxOS	White Paper	H17108
Dell EMC Embedded Management on PowerMax, VMAX All Flash, and POWERMAX	White Paper	H16856
Data Reduction with Dell EMC PowerMax	White Paper	H17072
Dell EMC PowerMax Reliability, Availability, and Serviceability Technical Note	White Paper	H17064
VMAX All Flash iSCSI Deployment Guide for Oracle Databases	White Paper	H15132.1
VMAX All Flash iSCSI Deployment Guide for Windows	White Paper	H15143