NOTE
This document is offered as an aid in designing and implementing a development and test network boot environment with NetApp® storage. While this document presents examples regarding the available options, several possible configurations exist. Therefore, your final configuration might vary based on your requirements and environment.
# TABLE OF CONTENTS

1 INTRODUCTION .................................................................................................................. 3
   1.1 INTENDED AUDIENCE ................................................................................................. 3
   1.2 NETWORK STORAGE IN DEVELOPMENT AND TEST ENVIRONMENTS ......................... 3
   1.3 NETAPP ADVANTAGE IN DEVELOPMENT AND TEST ENVIRONMENTS ...................... 4

2 PROJECT REQUIREMENTS AND GATHERING INFORMATION ............................................. 7

3 DESIGN CONSIDERATIONS AND OPTIONS ....................................................................... 8
   3.1 PHYSICAL VS. VIRTUAL DEVICES ............................................................................... 8
   3.2 ISCSI PROTOCOL INTRODUCTION ............................................................................. 10
   3.3 FIBRE CHANNEL PROTOCOL INTRODUCTION .......................................................... 11
   3.4 PROTOCOL CONSIDERATIONS .................................................................................... 12
   3.5 HARDWARE CONSIDERATIONS .................................................................................... 12

4 CONFIGURATION AND IMPLEMENTATION ....................................................................... 13
   4.1 GENERAL CONFIGURATION INFORMATION .............................................................. 13
   4.2 PHYSICAL ISCSI CONFIGURATION ............................................................................ 14
   4.3 PHYSICAL FIBRE CHANNEL CONFIGURATION ............................................................ 21
   4.4 VIRTUAL SERVER CONFIGURATION .......................................................................... 29
   4.5 STORAGE ALIGNMENT ................................................................................................. 45
   4.6 BOOT IMAGE CLONE PREPARATION .......................................................................... 47
   4.7 BLOCK-LEVEL CLONING ............................................................................................. 52
   4.8 FILE-LEVEL VIRTUAL MACHINE CLONING ................................................................. 54

APPENDIX 66

DOCUMENTATION ...................................................................................................................... 66
1 INTRODUCTION

1.1 INTENDED AUDIENCE
This document aids administrators who are implementing a network boot development and test environment with NetApp storage. The reader should have a working knowledge of server/workstation hardware, networking concepts, and NetApp technologies.

1.2 NETWORK STORAGE IN DEVELOPMENT AND TEST ENVIRONMENTS
Since the advent of hard disk drives in the 1950s, direct-attached storage (DAS) has generally been the most commonly used technology to store operating system data. With DAS, the local storage is unique to each device and is managed independently from other devices’ local storage. This usually creates additional management overhead in a development and test infrastructure due to the dynamic and nonpersistent nature of the data. With every new development and test cycle comes the possibility of new operating systems or configurations. The amount of change needed in the environment between testing cycles creates a proportional amount of management overhead.

Migrating from DAS toward a network boot environment with NetApp storage significantly reduces the management overhead involved. Storing operating system data on NetApp storage allows centralized management of system deployments and allows for rapid configuration and reconfiguration of devices. Instead of lengthy installation procedures performed on each device, an operating system can be installed once on NetApp storage, efficiently cloned as needed, and logically mapped to the necessary devices.

While the initial capital cost to deploy a DAS solution is typically lower than that for network storage, the ongoing operational cost of network storage is much lower. This operating expenditure (OPEX) savings is realized through reduced labor requirements needed to install and maintain the test environment. In addition, a capital expenditure (CAPEX) savings is realized by leveraging built-in NetApp Data ONTAP® technologies such as data deduplication and FlexClone® to increase storage efficiency. NetApp Data ONTAP includes several built-in technologies that increase data and administrative efficiency while maintaining operational simplicity.

Figure 1) Direct-attached storage vs. network storage.
1.3 NETAPP ADVANTAGE IN DEVELOPMENT AND TEST ENVIRONMENTS

NETAPP FLEXCLONE TECHNOLOGY

NetApp FlexClone technology is a key advantage in development and test environments. FlexClone technology allows instantaneous cloning of files and data containers that consume no additional physical storage space. Each clone references the same storage blocks as the source; therefore, numerous clones can be created without transferring or copying any bits and without performance penalty. FlexClone technology can be used to clone flexible volumes, LUNs, and individual files within flexible volumes and LUNs depending on the NetApp Data ONTAP version.

This technology provides a major efficiency advantage in a development and test environment. For example, an administrator can create a single “gold image” for each type of operating system needed and use FlexClone technology to clone the image for each device in the environment. FlexClone technology can provide these same advantages for application data (for example, databases and so on).

![Time to Provision w/o FlexClone](image)

Figure 2) Time to Provision without FlexClone
Figure 3) Space to provision an 8GB operating system image without FlexClone.

Figure 4) Time to provision with FlexClone.
Every NetApp controller runs the same Data ONTAP operating system and supports the full line of industry standard storage protocols, such as iSCSI, FCP, FCoE, NFS, and CIFS. This unified architecture allows NetApp storage controllers to serve multiple protocols and functions within a development and test environment, no longer requiring dedicated hardware to serve each purpose.

Similar to server virtualization, NetApp MultiStore® software allows for the creation of multiple virtual storage controllers called vFiler™ units within the same physical controller. Each vFiler unit can operate within its own IP space and security domain, providing multiple groups and projects with the flexibility to leverage the same physical controller, but access their own secure vFiler instance. NetApp MultiStore software further enhances the management of controller utilization and maximizes data availability by enabling controller portability, allowing vFiler units to be migrated or replicated from one physical NetApp storage controller to another. This process can take place within the same local infrastructure or across geographic locations, such as a coprocessing or DR site.

In most environments it is inevitable that multiple copies of the same data will be written to storage. In this case, valuable disk space is unnecessarily consumed with redundant copies. Deduplication reduces duplicate blocks of data to reclaim disk capacity. This technology works at the flexible volume level, identifying and removing redundant data blocks. Deduplication can lead to increased data storage efficiency and a lower total disk capacity requirement, depending upon the environment and type of data being stored.

Figure 5) Space to provision an 8GB operating system image with FlexClone.
2 PROJECT REQUIREMENTS AND GATHERING INFORMATION

This section explores the implementation options and requirements for a network boot environment based on NetApp storage. The answers to these questions will be useful in making design decisions, discussed in the following section. As with any design process, it is important to start by compiling a list of all necessary requirements. This can help to make sure the infrastructure meets all the expectations and demand for which it has been built.

SERVER/WORKSTATION

- What operating systems are necessary in the environment (for example, Windows® Server 2003, Red Hat Enterprise Linux® 5.0, and so on)?
  - This will provide information on what operating systems will be supported and distinguish which virtualization technologies could be leveraged.

- How many operating system instances are necessary?
  - This information will help to define the scale of the environment. Physical environments are fairly straightforward: the number of instances and physical servers are equal. Virtual environments are more complex, as they decouple physical servers from OS instances.

- What are the nature and distribution of the workload on the systems?
  - Understanding the nature and distribution of the workload is necessary to determine the size and diversity of resources (for example, memory, CPU, and storage) necessary and ultimately how many physical servers are needed to run an OS or virtualization hypervisor.

- What equipment, if any, already exists in your environment? Can it be leveraged for this deployment?
  - If existing equipment will be used for this deployment, it could affect some of the characteristics of the environment. For example, if the equipment has minimal resources (CPU, RAM, and so on) installed, this might weigh heavily on whether to use server virtualization in the environment. Typically server virtualization works best on resources with higher memory and processor counts. Keep in mind the possibility of upgrading CPU and RAM resources depending on the equipment (age, capability, and so on) and budget.

NETWORK

- Does your development and testing require the use of certain storage protocols (for example, iSCSI, FCP, and so on)?
  - It is important to define what protocols are needed for the workload, as this will influence both server and network design decisions. For example, traditional FCP will require a dedicated switching infrastructure.

- What storage protocols already exist in the environment?
  - If the infrastructure to support certain storage protocols already exists in the environment, this could be leveraged to minimize implementation cost.

STORAGE

- Is it possible to leverage an existing NetApp storage controller?
In many cases, existing NetApp storage can be leveraged in a development and test deployment due to the unified architecture supporting iSCSI, FCP, and NAS within the same controller. This opportunity will depend upon the size of the environment and available capacity of the controller.

- If there is an existing NetApp storage controller, what licenses are available?
- If using existing NetApp storage, check which licenses are installed on the system. This could affect decisions regarding which storage protocol to deploy in the development and test environment.

### 3 DESIGN CONSIDERATIONS AND OPTIONS

This section provides guidance on evaluating the requirements and options available toward making the optimal design decisions.

#### 3.1 PHYSICAL VS. VIRTUAL DEVICES

The decision to use physical or virtual devices affects the overall design of the environment. A virtualized environment can inherently provide many benefits difficult to achieve in a physical environment (for example, consolidation, device portability, and high availability). However, in some environments and implementations, virtualization can also introduce additional complexity and cost, which negate these benefits. It is a common misconception that server virtualization is the only form of virtualization. Even though server virtualization is the most well-known form, virtualization exists within many other layers of the infrastructure. Storage and network virtualization is becoming more prevalent as data centers focus on expanding capabilities and maximizing resource efficiency within the same or a smaller footprint. A combination of both physical and virtual technologies can be employed throughout all layers of the infrastructure to build the necessary network boot environment.

This decision requires examining several factors that are specific to each environment and should be based upon a discussion of requirements and cost.

- Existing hardware might not be adequate for virtualization or lack sufficient capacity to make virtualization cost-effective. Upgrading such systems could be an option, but might not be feasible for all budgets. In this case it might be necessary to leverage the existing hardware in a physical configuration instead.

- Another factor to consider is the type of operating systems and the number of instances needed in the environment. If a large number of instances is necessary and the hardware sufficiently supports virtualization, then the best option might be a virtual environment. It is important to review the compatibility guides associated with the hardware and hypervisors in question to make sure the necessary operating systems are supported. Some operating system compatibility information is provided in Table 2) Hypervisor protocol support. If high availability and resource scheduling are required, virtual environments can provide greater uptime and increased control of resources within a physical system.

- As with any decision making process, cost is also a factor. Remember that with some virtualization hypervisors there is a licensing cost associated with different features and capabilities. In addition to licensing the virtualization hypervisor, each individual operating system must be licensed. Licensing for every operating system instance is required in both physical and virtual environments.
Table 1) Hypervisor guest operating system support.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>VMware ESX</th>
<th>Microsoft Hyper-V</th>
<th>Citrix XenServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Windows Vista</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Windows Server 2003</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Windows XP</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Windows NT® 4.0</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CentOS 5.0, 5.1, 5.2</td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 5.0, 5.1, 5.2</td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 4</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 3</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 2.1</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Linux 9.0</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Linux 8.0</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Linux 7.3</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Red Hat Linux 7.2</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oracle® Enterprise Linux 5.0, 5.1, 5.2</td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>SUSE Linux Enterprise Desktop 10</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux Enterprise Server 10</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SUSE Linux Enterprise Server 9</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SUSE Linux Enterprise Server 8</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SUSE Linux 9.3</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux 9.2</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux 9.1</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux 9.0</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSE Linux 8.2</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu Linux 8.04 LTS</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu Linux 7.10</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ubuntu Linux 7.04</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FreeBSD 4.11</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FreeBSD 4.10</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FreeBSD 4.9</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetWare 6.5 Server</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetWare 6.0 Server</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NetWare 5.1 Server</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solaris™ 10 x86</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Latest virtualization hypervisor version as of this publication. Check hypervisor documentation for further version information.

Table 2) Hypervisor protocol support.

<table>
<thead>
<tr>
<th>Storage Protocols</th>
<th>VMware ESX</th>
<th>Microsoft Hyper-V</th>
<th>Citrix XenServer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAN/iSAN</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>iSCSI</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>FC</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NAS</td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>NFS</td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>CIFS</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Supports both software and hardware iSCSI.
3.2 ISCSI PROTOCOL INTRODUCTION

ISCSI OVERVIEW
The iSCSI protocol is a block-level storage protocol that utilizes an IP network to transfer SCSI commands between a client, known as the initiator, and a remote storage server, known as the target. Both an initiator and target device are required to create an iSCSI connection. Since iSCSI leverages Ethernet, all iSCSI devices require a unique MAC address and IP address similar to any standard Ethernet device. iSCSI devices also require another identifier, specific to the iSCSI protocol, in order to function properly. This identifier is known as an iSCSI qualified name, or IQN.

ISCSI QUALIFIED NAME
The IQN is a unique identifier configured on each device that serves as that device’s iSCSI address. The format for an iSCSI IQN is as follows:

\[ \text{iqn.<yyyy-mm>.<reversed domain name>:<identifying text>} \]

<table>
<thead>
<tr>
<th>iqn.1992-08.com.netapp:controller1</th>
</tr>
</thead>
</table>

- \(<yyyy-mm>\) = year and month in which the organization registered the domain
- \(<\text{reversed domain name}>\) = domain name reversed
- \(<\text{identifying text}>\) = text to uniquely identify the device

HARDWARE VS. SOFTWARE ISCSI INITIATOR
There are two types of iSCSI initiators, hardware and software, that can be used along with an iSCSI target to create an iSCSI connection. Hardware initiators use dedicated hardware (for example, iSCSI HBAs or multifunction NICs) in conjunction with firmware to implement iSCSI. Hardware initiators offload iSCSI processing from the CPU, which abstracts the iSCSI implementation from the OS and yields better performance in some cases. Software initiators utilize a special operating system driver to perform iSCSI operations over a standard NIC. Once an OS is loaded, the control of the iSCSI session is transferred to the native iSCSI initiator driver resident within the OS. In current OS implementations, iSCSI session parameters (for example, IP address, iSCSI initiator IQN, iSCSI target IQN, and so on) are passed from the NIC firmware to the OS using a standardized iSCSI boot firmware table (IBFT). Following this standard, the NIC firmware creates the IBFT in a specific memory location to store the iSCSI parameters for the software.
initiator to fetch and assume iSCSI operations once the OS running. Support for IBFT-compliant iSCSI boot is growing among most current OS distributions; however, legacy distributions are limited by the lack of software iSCSI support natively within the OS.

3.3 FIBRE CHANNEL PROTOCOL INTRODUCTION

FIBRE CHANNEL OVERVIEW
Fibre Channel (FC) is another block-level storage protocol that transfers SCSI commands between a client initiator and a storage server target over a network optimized for block-level data transport. To create a Fibre Channel connection between an FC initiator and FC target, both require a unique World-Wide Name (WWN) in order to function properly within the FC fabric.

WORLD-WIDE NAME (WWN)
A WWN is a unique 8-byte (64-bit) identifier assigned to each FC device, similar to a MAC address in Ethernet standards. WWNs are used to uniquely identify each FC device (initiators, switches, and targets) within the fabric.

Example: 50:0a:09:80:87:f9:57:de

ZONING
Administrators can create logical groupings referred to as zones within the fabric to restrict initiator to target visibility. Properly configured zones allow initiators to access only the necessary targets rather than all targets within the fabric. Two types of zones exist within an FC environment: hard zones and soft zones. Hard zones are created by grouping the FC switch physical ports connected to the initiator and target. Vulnerabilities with hard zoning can arise when an FC device is removed and a different FC device is connected to the same physical port. The new device will automatically become a member of the previous device’s zones and might have unrestricted access to the other devices within those zones.

Soft zoning uses WWNs to identify the members within each zone. With soft zoning, devices can be moved from one physical port to another while maintaining consistent zone membership. Also, if additional devices are connected to the fabric, their unique WWNs will not have access to any zoned FC devices until configured to do so.

Figure 7) FC topology.
MULTIPATHING

FC multipathing allows for redundant paths from an initiator to a target device to provide a higher level of availability. There are three multipathing modes available when using NetApp storage in an FC environment:

- **Active-active mode (recommended)**
  - Provides multiple paths to a single LUN
  - Uses multiple active paths for bandwidth aggregation
- **Active-passive mode**
  - Provides a single active path to a LUN with a second passive path available for redundancy
  - Single active path provides no bandwidth aggregation
- **Asymmetric active-active mode**
  - Provides multiple active and passive paths to a single LUN

When using the recommended active-active multipathing mode, the NetApp controller must be configured using a specific cluster failover mode (cfmode) called *single image cfmode*. This cfmode makes sure that multipathing works properly if a fault occurs. The cfmode determines the behavior of FC target ports in normal and cluster failover operations in order to preserve the FC connection between the initiator and the appropriate target/LUN.

3.4 PROTOCOL CONSIDERATIONS

Selecting the appropriate storage protocol for a development and test network boot environment is an important factor in the overall infrastructure design. There is no single right answer when selecting a storage protocol; in fact, technical requirements or existing infrastructure might lead to employing multiple storage protocols in the environment. Below are a few points to consider while making this decision.

- The infrastructure needed to support a Fibre Channel fabric can be more expensive than other storage protocols that only require an Ethernet network. Leveraging an existing FC infrastructure might be an option, depending upon available capacity and security concerns.
- There are Ethernet-based alternatives to Fibre Channel (for example, iSCSI) that only require the use of an Ethernet network to function properly. Ethernet networks have become a staple in infrastructure design, but their suitability as a storage network in each environment will depend upon available capacity, security concerns, and fundamental network design.
- If a specific virtualization hypervisor or operating system is required in the environment, make sure that the required storage protocols are supported. The appendix provides references to further documentation on hypervisor compatibility.

3.5 HARDWARE CONSIDERATIONS

Once the questions of physical vs. virtual and storage protocol(s) have been answered, the next step is to determine the hardware requirements to implement in the environment. It is important to check hardware and software compatibility guides to make sure that the desired configuration is supported. It is a good strategy to evaluate different hardware and software vendors to make sure of compatibility and that the solution satisfies the defined requirements.

- Obtain evaluation equipment to better understand the hardware and its specific configuration options before making a purchasing decision.
- Perform stability and performance tests on evaluation equipment to make sure no problems arise as a result of the configuration.
4 CONFIGURATION AND IMPLEMENTATION

The following section provides information and examples regarding the configuration of devices in a network boot environment. While some examples are cited below, note that many additional configurations are supported.

4.1 GENERAL CONFIGURATION INFORMATION

VIRTUAL MEMORY
Physical memory (RAM) is an essential component to booting and running an operating system. Because computer systems have a limited amount of physical memory, operating systems implement virtual memory mechanisms to provide space for lower priority memory pages and reallocate space in physical memory for higher priority memory pages. These virtual memory processes are commonly referred to as paging or swapping.

In order to maximize the performance and efficiency of the environment, the placement of virtual memory should be examined when using network storage and the technologies built into storage processors. Virtual memory is dynamic and changes frequently due to number of memory pages being moved in and out of physical memory. If not implemented carefully, these characteristics can cause unnecessary consumption of disk space when leveraging Snapshot™ copies from NetApp storage. When a Snapshot copy is created, the associated blocks are retained for the life of the Snapshot copy, and subsequent writes allocate new blocks, which consume additional physical storage. Because the storage for virtual memory is used only temporarily, there is little value to including it in a Snapshot copy. Therefore, we will examine options for managing the storage allocated to virtual memory separately from the storage allocated to the rest of the operating system.

The first and often best option for alleviating the consumption of disk space by virtual memory is to eliminate the need for virtual memory. Virtual memory is used only when physical memory is fully allocated and lower priority memory pages need to be moved out to accommodate demand for higher priority memory pages. By adding physical memory to the system, more physical capacity is available for both low- and high-priority memory pages, which can entirely alleviate the need to use virtual memory. In some cases, upgrading the amount of physical memory in a system might not be possible. Another option is to investigate which applications are causing the system to use virtual memory and take steps to minimize or eliminate that demand. If these measures are not feasible or sufficient, another strategy is to relocate the storage used for virtual memory into a separate volume, isolated from the rest of the operating system and application data. When virtual memory storage is placed into a different volume, Snapshot copies can be disabled on that particular location, making sure that the least space possible is consumed by virtual memory. One caveat to this procedure is that Microsoft® Windows systems require a small page file (at least 2MB) to be located on the system partition. This page file is used for writing memory dumps, which can be useful when troubleshooting certain OS issues. With any configuration that includes more than one swap partition, make sure that each resides in a storage location with similar performance characteristics. For example, do not place one on local disk and another on network storage, as this can result in problems or errors within the operating system.

LUN IDS
It might be necessary to set the LUN ID of a bootable LUN to “0” for the boot process to complete successfully, depending upon the hardware involved. This requirement does not exist for every hardware/software vendor, but is generally considered to be a best practice within a SAN boot configuration. If a system is booting from direct-attached storage and using network-attached LUNs for application and other data, it is best to assign these LUNs with a nonzero LUN ID. This will accommodate any future decision to migrate from local boot to a SAN boot environment.

DHCP
The Dynamic Host Control Protocol (DHCP) is a useful tool for quickly and automatically assigning IP parameters (address, netmask, default gateway, and so on) to devices in a network. DHCP can assign IP parameters either dynamically from a pool of values or statically according to a device’s MAC address.
DHCP can also assign other configuration parameters in the form of DHCP options (for example, iSCSI target IP, iSCSI target IQN, root path, and so on). These options can be useful in further automating procedures such as setting up an iSCSI initiator or configuring PXE. When devices are powered on, their DHCP-enabled Ethernet interfaces broadcast a discovery message on the network. The DHCP servers on the network that receive the discovery broadcast, then reply with a unicast DHCP offer, which includes the MAC address of the client, an IP address offer for the client, and the DHCP server IP address. Since there can be multiple DHCP servers on the network and ultimately the client might receive multiple DHCP offers from different servers, the client sends a message back to each DHCP server that offered an IP address to inform them whether their offer was accepted and, if so, to request any additional configuration parameters. As a final step, the DHCP server that was accepted creates an IP address lease, which reserves that address for a specified time period and sends a final acknowledgment back to the client.

4.2 PHYSICAL iSCSI CONFIGURATION

REQUIRED COMPONENTS
Booting physical clients from NetApp storage using the iSCSI protocol is a simple procedure provided the following components exist in the environment:

- NetApp storage controller (physical controller or vFiler unit)
- NetApp storage iSCSI license
- iSCSI initiator for each physical client (software initiator, multifunction NIC, or hardware HBA)
- Operating system licenses
- Ethernet infrastructure

iSCSI CONFIGURATION
The following example details the steps involved in configuring a device to boot using the iSCSI protocol from NetApp storage. While there are many variations of iSCSI initiators on the market, the concepts and general configuration steps apply. It is important to be familiar with the hardware used, such as in blade server environments where internal interfaces might map to chassis switch modules in varying ways, and understanding these I/O mappings is crucial to identifying which ports need to be configured for each purpose. More information on advanced iSCSI configurations can be found in the appendix and on the NetApp NOW™ Web site at now.netapp.com.

The configuration steps outlined below will produce the following:

- A 10GB thin-provisioned iSCSI LUN on NetApp storage
- An iSCSI session between the device (iSCSI initiator) and the NetApp controller (iSCSI target)

---

Gather Target Information
1. Log into the NetApp Data ONTAP command line with the appropriate credentials. Check to make sure that iSCSI is licensed on the controller by typing `license` at the command prompt and then press **ENTER**. If licensed, a license code will appear next to `iscsi` in the list of possible licenses. If not licensed, use the `license add` command to add the appropriate license code.

   ```
   NetApp> license
   a_sis not licensed
   cifs not licensed
   cluster not licensed
   cluster_remote not licensed
   iscsi site ABCDEFG
   compression not licensed
   disk_sanitization not licensed
   fcp not licensed
   flex.clone not licensed
   flex_scale not licensed
   flexcache.nfs not licensed
   ```

2. Determine if the iscsi service is running using the `iscsi status` command. If the iscsi service is not running, use the `iscsi start` command to start the service.

   ```
   NetApp> iscsi status
   iSCSI service is not running

   NetApp> iscsi start
   Wed Mar 11 15:01:46 GMT [iscsi.service.startup:info]: iSCSI service startup
   iSCSI service started
   ```

3. Make a note of the IQN or iSCSI nodename of the NetApp controller. This can be displayed using the `iscsi nodename` command. Keep this information close as it will be used later in the procedure to configure the iSCSI initiator.

   ```
   NetApp> iscsi nodename
   ```

4. Make sure that the desired network interface is configured on the NetApp controller using the `ifconfig -a` command. Any Ethernet interface on the controller can be used as the iSCSI interface; just be sure it has been configured with the proper IP address. See the appendix for documentation on configuring network interfaces on NetApp storage. In this example, port e0b is used with an IP address of 192.168.10.200. Make note of the IP address of this port (the target port) next to the noted iscsi nodename. Both of these variables will be used later in the iSCSI initiator configuration.
NetApp> ifconfig -a
  e0a: flags=948043<UP,BROADCAST,RUNNING,MULTICAST,TCPCKSUM> mtu 1500
  inet 10.60.100.15 netmask 0xffffff00 broadcast 10.60.100.255
    ether 00:a0:98:09:d4:0c (auto-1000t-fd-up) flowcontrol full
  e0b: flags=948043<UP,BROADCAST,RUNNING,MULTICAST,TCPCKSUM> mtu 9000
  inet 192.168.10.200 netmask 0xffffff00 broadcast 192.168.10.255
  inet 192.168.20.200 netmask 0xffffff00 broadcast 192.168.20.255
    ether 00:a0:98:09:d4:0d (auto-1000t-fd-up) flowcontrol full
  lo: flags=1948049<UP,LOOPBACK,RUNNING,MULTICAST,TCPCKSUM> mtu 8160
  inet 127.0.0.1 netmask 0xff000000 broadcast 127.0.0.1
  ether 00:00:00:00:00:00 (VIA Provider)

### iSCSI Initiator Configuration

**NOTE** The next steps outline the configuration of the iSCSI initiator. The appearance of the setup/configuration utilities vary among vendors, but the iSCSI concepts and necessary variables apply to all.

1. Power on or reboot the server and enter the setup for the iSCSI card or initiator installed in the system. The key sequence to enter setup varies between vendors. In this example, **Ctrl + S** is used to enter the Configuration Menu for the HP NC373m (Broadcom NetXtreme II) NIC. Also, this card is a dual port card, so be sure to enter the Configuration Menu for the correct port. This example uses the second port, in which case **Ctrl + S** is pressed after the second interface is shown.

2. In the Configuration Menu, enter the iSCSI Cfg setup by pressing **Ctrl + I**.
3. Select **General Parameters** from the **Main Menu**.

4. On the **General Parameters** page, make sure **Boot to iSCSI target** is **Enabled**. DHCP can be used to assign certain variables on the initiator card; if DHCP is being used in the environment, make sure it is **Enabled**. In this example, the variables for the iSCSI initiator will be statically assigned (manually entered), so DHCP will be **Disabled** on the card. CHAP authentication may also be enabled from this menu; use as desired. CHAP will be **Disabled** in this environment. Once set, press **ESC** to exit back to the **Main Menu**.
5. Select **Initiator Parameters** from the **Main Menu**.

6. On the **Initiator Parameters** page, enter the card’s **IP Address**, **Subnet Mask** and **iSCSI Name (IQN)**. Caution: Be sure to enter the *initiator* information on this page, not the NetApp controller (target) information. There might already be a default value for **iSCSI Name**; this can optionally be renamed to better describe the initiator. Be sure to note the iSCSI Name (IQN) of the initiator; this will be used later in the configuration procedure. If using CHAP authentication, enter the appropriate values for the initiator CHAP configuration on this page as well. The **Default Gateway** parameter is only needed if the iSCSI traffic will be routed. The **Primary and Secondary DNS** parameters are optional. Once the necessary parameters are entered, press **ESC** to exit back to the **Main Menu**.
7. Select **1st Target Parameters** from the **Main Menu**.

8. On the **1st Target Parameters** page, make sure the **Connect** parameter is set to **Enabled** and enter the **IP Address** and **iSCSI Name** of the NetApp controller (target) noted in the steps above. Make sure the **TCP Port** parameter is set to **3260** and the **Boot LUN** parameter is set to **0**. Once the necessary values have been set, press **ESC** twice to exit to the **Main Menu** and then select **Exit** and **Save Configuration**.
1. Now that the iSCSI initiator has been configured, the NetApp controller (target) needs to be configured to accept iSCSI connections from the initiator. Log back into the NetApp controller command line and use the `igroup create` command to set up an igroup for the initiator configured above.

   **NOTE** With the `igroup create` command, an `-i` or `-f` argument must be used to specify whether the LUN will be used in conjunction with the iSCSI or FC protocol. In this case make sure to use the `-i` option. Also be sure to specify the `-t <ostype>` argument, as this will make sure the file system is properly aligned when loaded on the LUN.

   `igroup create` command usage:
   
   `igroup create {-i | -f} -t <ostype> <initiator_group_name> <iqn>`

   NetApp> igroup create -i -t windows initiator1 iqn.1995-05.com.broadcom:iscsiboot
   
   NetApp> igroup show
   
   `initiator1 (iSCSI) (ostype: windows):
   iqn.1995-05.com.broadcom:iscsiboot (not logged in)`

2. Now create the necessary LUN to connect to the initiator using iSCSI with the `lun create` command. In the example below a 10GB LUN is created for a Windows installation. The `-o noreserve` option is the `thin-provisioning` option allowing the LUN to be created but initially take up no space on disk. Only until the operating system or application data is written to the LUN does it consume space.

   **lun create** command usage:
   
   `lun create -s <size> -t <ostype> [-o noreserve ] <lun_path>`

   NetApp> lun create -s 10g -t windows -o noreserve /vol/test/win2k3_gold
   
   `lun create: created a LUN of size: 10.0g (10742215680)`

   NetApp> lun show
3. Now that the iSCSI initiator and target have been configured, map the newly created LUN to the appropriate igroup for the newly configured initiator using the `lun map` command.

   **lun map command usage:**
   `lun map <lun_path> <igroup> [ <lun_id> ]`

   With the `lun map` command, make sure to specify a desired `<lun_id>` of 0 to make sure the initiator always boots from lun id 0.

   ```
   NetApp> lun map /vol/test/win2k3_gold initiator1 0
   Wed Mar 11 14:55:53 GMT [lun.map:info]: LUN /vol/test/win2k3_gold was mapped to initiator group initiator1=0
   ```

   ```
   NetApp> lun show -m
   LUN path          Mapped to            LUN ID  Protocol
   ************************************************************************
   /vol/test/win2k3_gold  initiator1      0     iSCSI
   ```

4. Once the server is rebooted, the iSCSI initiator should log into the NetApp controller (iSCSI target), which can be viewed from the NetApp Data ONTAP command line.

5. The 10GB iSCSI attached LUN is now attached to the server and is ready to receive an operating system or application data as needed. If loading an operating system, be sure to check the availability of native (already installed) device drivers for your iSCSI adapter/HBA. If not natively supported, you will need to provide the device drivers during the OS installation. For example, load third-party drivers by pressing F6 during a Microsoft Windows installation.

### PHYSICAL FIBRE CHANNEL CONFIGURATION

#### REQUIRED COMPONENTS
Booting physical clients from NetApp storage using the Fibre Channel protocol is a simple procedure provided the following components exist in the environment:

- NetApp storage controller (physical controller required, vFiler units not supported)
- NetApp storage FC license
- FC initiator HBA
- Operating system licenses
- FC switch

#### FCP CONFIGURATION
As a best practice, create zones based on single initiator ports. In other words, make sure each FC initiator in the fabric has its own unique zone. Target ports might exist in multiple zones, but initiator ports should exist only in a single zone. For more information on advanced FC configurations, refer to the documentation in the appendix, the NetApp NOW Web site at [now.netapp.com](http://now.netapp.com), and the appropriate documentation provided by your FC switch vendor.

The configuration steps outlined below will produce the following:

- A 10GB thin-provisioned FC LUN on NetApp storage
- An FC connection between the device (FC initiator) and the NetApp controller (FC target)
1. Log into the NetApp Data ONTAP command line with the appropriate credentials. Make sure Fibre Channel is licensed on the controller by typing `license` at the command prompt, and then press `ENTER`. If licensed, a license code will appear next to `fcp` in the list of possible licenses. If not licensed, use the `license add` command to add the appropriate license code.

```bash
NetApp> license
    a_sis not licensed
    cifs not licensed
    cluster not licensed
    cluster_remote not licensed
    compression not licensed
    disk_sanitization not licensed
    fcp site ABCDEFG
    flex_clone not licensed
    flex_scale not licensed
    flexcache_nfs not licensed
    gateway_hitachi not licensed
```

2. Determine whether the FC service is running using the `fcp status` command. If the fcp service is not running, use the `fcp start` command to start the service. The controller attempts to bring online any FC ports configured as target ports in the system (such as the 0d port in the example below).

```bash
NetApp> fcp status
FCP service is not running.
NetApp> fcp start
Wed Mar 11 13:19:24 EST [fcp.service.startup:info]: FCP service startup
Wed Mar 11 13:19:24 EST [scsitarget.ispfct.onlining:notice]: Onlining Fibre Channel target adapter 0d.
```

3. Make a note of the WWN or fcp nodename of the NetApp controller. This can be displayed using the `fcp nodename` command. Keep this information close as this will be used later in the procedure to configure the proper zone within the FC fabric.

```bash
NetApp> fcp nodename
Fibre Channel nodename: 50:0a:09:80:87:f9:57:de (500a098087f957de)
```

**FC Initiator Configuration**

**NOTE** The next steps outline the configuration of the FC initiator. The appearance of the setup/configuration utilities varies among vendors; however, the FC concepts and necessary variables are standard.

4. Power on or reboot the server. Use the correct key sequence for the vendor's HBA to enter the configuration menu. In this example, using a QLogic FC Adapter, **Ctrl + Q** is used to open the Fast!UTIL as seen below.
5. From the Qlogic Fast!UTIL main menu, select **Configuration Settings**.

6. Enter **Adapter Settings**.
7. On the **Adapter Settings** page, make sure the **Host Adapter BIOS is Enabled**. Also make note of the 16-character **Adapter Port Name** or **WWN**. This will be used later in the procedure to create the proper zoning within the FC fabric.

8. Press **ESC** to exit **Adapter Settings** and enter **Selectable Boot Settings**.
9. On the Selectable Boot Settings page, make sure the Selectable Boot option is set to Enabled. This allows the card to boot from a target device using this FC adapter.

10. Once complete, press ESC until prompted to Save changes or until the FastUTIL Options menu is displayed. If no changes were made, you will not be prompted to save your changes. Save the changes if necessary and leave the server at the FastUTIL Options menu.
**FC Target Configuration**

**NOTE**
The next steps outline the configuration of the NetApp storage controller as the FC target device.

8. With the `igroup create` command, an `-i` or `-f` argument must be used to specify whether the initiator group will be used in conjunction with the iSCSI or FC protocol. In this case make sure to use the `-f` option. Also be sure to specify the `-t <ostype>` argument as this will make sure the file system is properly aligned when loaded on the LUN. The `<wwn>` is the value that was noted above when configuring the FC initiator called **Adapter Port Name** or **WWN**.

```
igroup create command usage:
igroup create {-i | -f} -t <ostype> <initiator_group_name> <wwn>
```

```
NetApp> igroup add initiator1 50:01:43:80:00:c5:1c:28
NetApp> igroup show
  initiator1 (FCP) (ostype: windows):
      50:01:43:80:00:c5:1c:28 (not logged in)
```

9. Now create the necessary LUN to connect to the initiator using FC with the `lun create` command. In the example below a 10GB LUN is created for a Windows installation. The `-o noreserve` option is the **thin-provisioning** option allowing the LUN to be created but initially takes up no space on disk. Only until the operating system or application data is written to the LUN does it consume space.

```
lun create command usage:
lun create -s <size> -t <ostype> [ -o noreserve ] <lun_path>
```

```
NetApp> lun create -s 10g -t windows -o noreserve /vol/test/win2k3_gold
lun create: created a LUN of size:  10.0g (10742215680)
NetApp> lun show
  /vol/test/win2k3_gold         10.0g (10742215680)  (r/w, online)
```
Now that the FC initiator and target have been configured, map the newly created LUN to the appropriate igroup for the newly configured initiator using the `lun map` command.

```
lun map command usage:
lun map <lun_path> <igroup> [ <lun_id> ]

With the `lun map` command, make sure to specify a desired "<lun_id>" of 0 to make sure the initiator always boots from lun id 0.

NetApp> lun map -f /vol/test/win2k3_gold initiator1 0
Wed Mar 11 14:08:00 GMT [lun.map:info]: LUN /vol/test/win2k3_gold was mapped to initiator group initiator1=0

NetApp> lun show -m
LUN path Mapped to LUN ID Protocol
/vol/test/win2k3_gold initiator1 0 FCP
```

**NOTE**
The next step is to configure the Fibre Channel zoning to segment devices and allow the appropriate initiator(s) to see the appropriate target(s). This example will not show the step-by-step configuration of the fabric but will provide information on the simple zone configuration used for this example. For more information on FC fabric configurations and procedures, refer to your FC switch vendor’s documentation or the NetApp NOW Web site at now.netapp.com.

```
brocade4100:admin> zoneshow
Defined configuration:
cfg: FC_example_1
  FC_soft_zone
zone: FC_soft_zone
  FAS3070_0d; Server
 alias:FAS3070_0d 50:0a:09:81:87:f9:57:de
 alias:Server H(50:01:43:80:00:c5:1c:28)

Effective configuration:
cfg: FC_example_1
zone: FC_soft_zone
  50:0a:09:81:87:f9:57:de
  50:01:43:80:00:c5:1c:28
```

After configuring the appropriate zones within the fabric, return to the server and the initiator `Fast!UTIL Option` menu. Choose Scan Fibre Devices. If configured correctly, the FC initiator should be able to see the NetApp LUN with the ID of 0 as seen below.
12. If the initiator cannot see the NetApp LUN, double check the WWN provided when creating the igroup, the zone configuration, and that the FC service has been started within Data ONTAP. Once properly configured, exit FastUTIL by pressing ESC and then reboot the server. As the server scans the PCI bus during POST, the FC Adapter (initiator) should log into the NetApp controller (target).
13. The 10GB FC attached LUN is now attached to the server and is ready to receive an operating system or application data as needed. If loading an operating system, be sure to check the availability of native (already installed) device drivers for your FC Adapter/HBA. If not natively supported, you will need to provide the device drivers during the OS installation. For example, load third-party drivers by pressing F6 during a Microsoft Windows installation.

4.3 VIRTUAL SERVER CONFIGURATION

Many development and test environments employ multiple storage protocols, each performing a specific function. In a virtual server environment, it is common to boot the virtualization hypervisor using one type of storage protocol while using a different storage protocol to store virtual machine files.

NFS STORAGE FOR VIRTUAL MACHINES

This example details the configuration steps to create a NetApp NFS export and attach it to VMware Virtual Infrastructure 3 (VI3) using the NFS protocol. Similar procedures can also be performed with other virtualization hypervisor platforms, such as Citrix XenServer. For more information on attaching NFS storage to various virtualization hypervisors, refer to the appropriate vendor documentation in the appendix.

The following steps describe how to produce a new aggregate and volume on the NetApp controller and present it as an NFS data store to the VMware Virtual Infrastructure.

**VMware Network Configuration**

<table>
<thead>
<tr>
<th>NOTE</th>
<th>Perform the following steps (1-10) for configuring a VMkernel port on each host in the VMware data center that needs access to the NFS storage.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Current topology within VMware vCenter (Virtual Center).</td>
</tr>
</tbody>
</table>
2. Current network configuration.

3. Add a VMkernal port on each host in the data center. Begin by selecting a host in the left topology. This example uses ktc5b1.

4. Select the Configuration tab (1) at the top and then the Networking section (2) on the sidebar menu. Then select Add Networking… (3) in the upper-right corner.

5. Select VMkernel and click Next to continue.
6. Select the vSwitch location for the VMkernel port. In this example, both physical NICs (vmnic0 and vmnic1) are attached to vSwitch0. Click Next to continue.

7. Enter the appropriate VLAN (VLAN ID) for the NFS traffic. Also enter a valid IP Address and Subnet Mask for the VMkernel port. Make sure that the VMkernel port is either on the same subnet as the desired interface on the NetApp controller or routable to that destination. Click Next to continue.

8. Click Finish.

**NOTE** A popup window might appear and ask if you would like to configure a default gateway address for the VMkernel port. If your VMkernel port is on the same subnet as the NetApp Controller interface, choose No. If the NetApp Controller interface is on a different subnet than the VMkernel port, choose Yes and enter the appropriate default gateway IP address.

9. Apply the VMkernel port to the ESX host's network configuration.
10. From the ESX host's command line interface, verify the connection from the VMkernel port within ESX to the interface you want to use for NFS traffic on the NetApp controller. Use the `vmkping` command to ping from the VMkernel port to a destination IP address. For this example, the IP address of the NetApp interface is 192.168.10.200.

   **Ping from ESX host VMkernel interface to NetApp interface:**
   
   ```
   [root@ktc5b1 root]# vmkping 192.168.10.200
   PING 192.168.10.200 (192.168.10.200): 56 data bytes
   64 bytes from 192.168.10.200: icmp_seq=0 ttl=64 time=0.048 ms
   64 bytes from 192.168.10.200: icmp_seq=1 ttl=64 time=0.016 ms
   64 bytes from 192.168.10.200: icmp_seq=2 ttl=64 time=0.015 ms
   --- 192.168.10.200 ping statistics ---
   3 packets transmitted, 3 packets received, 0% packet loss
   round-trip min/avg/max = 0.015/0.026/0.048 ms
   ```

   **Ping from NetApp interface to ESX Host VMkernel interface:**
   
   ```
   NetApp>ping 192.168.10.190
   192.168.10.190 is alive
   ```

**NetApp Storage Configuration**

1. If necessary, create an aggregate on the NetApp storage controller to hold the flexible volumes to be used as an NFS data store. Using NetApp FilerView® administration tool, expand the **Aggregates** section in the sidebar menu, select **Add**, and follow the wizard to configure the aggregate.

2. Select **Manage** in the **Aggregates** section of the sidebar menu. The new aggregate should now be visible in the list of current aggregates. In this example, `aggr0` is the aggregate containing the controller root volume and `aggr1` is the newly created aggregate.
3. To create a new flexible volume, expand the Volumes section in the sidebar menu and select Add.

4. The Volume Wizard opens in a new window. Click Next.

5. In the Volume Type Selection window, select Flexible and then click Next.
6. In the Volume Parameters window, enter the desired name for the flexible volume, then click Next.

7. In the Flexible Volume Parameters window, select the desired aggregate to place the flexible volume. Also select the appropriate Space Guarantee option. The Space Guarantee option determines the amount of storage space reserved upon creation of the volume. The Space Guarantee option of none enables thin provisioning of the VM data store and is recommended but not required.

8. In the Flexible Volume Size window, select the appropriate Volume Size Type, Volume Size, and Snapshot Reserve. When choosing these options, keep in mind the number of VMs that
8. In the **Flexible Volume Size** window, select the appropriate **Volume Size Type**, **Volume Size**, and **Snapshot Reserve**. When choosing these options, keep in mind the number of VMs that might be cloned inside the data store as well as the size of each VM's virtual disk. Set the **Total Size**, **Usable Size**, and **Snapshot Reserve** as appropriate.

9. The **Commit** window displays a summary of the options chosen. Verify that the desired selections have been made and click **Commit**.

10. Click **Close** when these updates have been applied.

11. Select **Manage** in the **Volumes** section of the sidebar menu. The new flexible volume should now be visible in the list of current volumes. The volume `NFS_datastore_001` was used in this example.
12. Snapshot copies of VMware data stores need to be coordinated with VMware ESX; therefore, Scheduled Snapshots should be disabled within NetApp Data ONTAP to avoid unnecessary storage consumption. Expand the Volumes -> Snapshots section in the sidebar menu and select Configure.

13. On the Configure Snapshots panel, select the volume to be used for the data store and disable the Scheduled Snapshots feature for this volume. Click Apply.
14. To configure the newly created flexible volume as an NFS export, expand the **NFS** section in the sidebar menu and select **Manage Exports**.

15. On the **Manage NFS Exports** panel, the new flexible volume might automatically appear in the list of NFS exports; otherwise, click **Add Export** to add it. Click the path of the flexible volume to be used for the VM data store to configure the NFS export options. The volume `/vol1/NFS_datastore_001` is used in this example.
16. The NFS Export Wizard opens. The VMkernel ports of every ESX host that needs to mount this NFS export will require both read/write and root access permissions. Enable the Read-Write Access, Root Access, and Security export options and click Next.

17. In the NFS Export Path panel, enter the NFS export path for this volume that will be presented to NFS clients. The flexible volume path is used by default. /vol/NFS_datastore_001 is used in this example.
18. In the Read-Write Access panel, uncheck the All Hosts box and then click Add.

19. Enter the IP address of a VMkernel port from an ESX host that requires access to this NFS data store. Click OK.

20. To support VMotion or HA across an ESX clustered resource pool, all of the ESX hosts in the cluster require access to the same VM data store. Continue to add all appropriate VMkernel port IP addresses for each ESX host as illustrated below. Click Next when complete.
21. In the **Root Access** panel, uncheck the **All Hosts** box and follow the same procedure to add the appropriate VMkernel port IP addresses for each ESX host that requires access to this VM data store. Click **Next** when complete.

22. In the **Security** panel, make sure that **Unix Style** security is selected and click **Next**.
<table>
<thead>
<tr>
<th></th>
<th>Review the summary of updates and click <strong>Commit</strong>.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>NFS Export Wizard - Commit</strong>&lt;br&gt;Below is a summary of your changes.</td>
</tr>
<tr>
<td></td>
<td>&lt; Back</td>
</tr>
<tr>
<td></td>
<td>Click <strong>Close</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>NFS Export Wizard - Success</strong>&lt;br&gt;1. Export updated successfully.</td>
</tr>
<tr>
<td></td>
<td>Close</td>
</tr>
</tbody>
</table>

|   | The **Manage NFS Exports** panel should now reflect the changes made to the NFS export options for the flexible volume. /vol/NFS_datastore_001 is used in this example. |
NOTE: The next step is to connect each ESX host in the Data Center to the NFS export to be used as a VM data store. This procedure must be performed for each ESX host in the VMware Data Center that requires access to the data store.

26. Select an ESX host.

27. On the Configuration tab for this ESX host, select the Storage panel from the Hardware sidebar menu. Click Add Storage…

28. When the Add Storage Wizard opens, select the Network File System storage type and click Next.
The next panel is used to configure the NFS mount options for the ESX host. In the **Server** property, enter the IP address of the NetApp controller interface to be accessed by the VMKernel port of this ESX host. In the **Folder** property, enter the NFS export path defined for the flexible volume to be used as a VM data store. Make sure that the **Mount NFS read only** option is unchecked. Provide a name to reference this VM data store within ESX and click **Next**.

Note: The DNS name for the NetApp controller could also be used in the **Server** property; however, this is not recommended due to connectivity issues that could occur if DNS services are interrupted.
30. Verify the configuration and click **Finish**.

31. The NFS-based VM data store should now appear in the **Storage** window as illustrated below. Virtual Machines can now be created and stored using this NFS-attached data store.
4.4 STORAGE ALIGNMENT

Storage alignment is an important factor in storage design for both physical and virtual server environments. Proper alignment involves making sure that all layers of storage management, from the storage controller to the OS file system, are appropriately coordinated to achieve the best possible performance. The following sections provide an overview of how storage alignment affects both physical and virtual deployment environments. Refer to NetApp TR-3747 for further details on this topic.

PHYSICAL ALIGNMENT

In a physical server environment, operating systems allocate and manage file system storage in logical units of equally sized blocks within a storage container, typically a disk partition or LUN. Likewise, storage controllers manage physical disk geometry and data reliability schemes such as RAID in terms of physical units of equally sized blocks on disk. When OS file system logical block boundaries are not aligned with the storage controller’s blocks, each OS storage request could result in multiple I/O requests from the storage controller, which will ultimately impair I/O performance. Operating systems often inadvertently introduce misalignment by injecting special identification data into a storage container before allocating their logical file system blocks. If this logical block offset does not correspond with an underlying storage block boundary, it will result in misalignment of the entire OS file system.

NOTE

The NFS data store must be added to every host that requires access to this NFS data store. It is not automatically connected to all hosts in a VMware Data Center upon configuring a single host. Once the NFS data store has been attached to the appropriate ESX hosts, virtual machines can be created and their associated files can be stored within the configured data store.
NetApp storage controllers provide easy solutions to help simplify physical storage alignment. NetApp LUN creation procedures allow storage administrators to specify which type of operating system will use the LUN, and the NetApp controller automatically compensates for any necessary OS file system offset to help maintain storage alignment.

VIRTUAL ALIGNMENT

Virtual server environments require even greater attention to storage alignment due to the additional layers of storage abstraction introduced by VM disk emulation. In these environments, storage performance degradation due to misalignment can be compounded by both the hypervisor file systems and the guest OS file systems housed within each VM’s virtual disk files. Storage alignment issues with virtual server environments vary by hypervisor type and storage methods used to host the virtual machines. For example, one VMware ESX configuration could use a VMFS file system to host VM virtual disk files. The following figures illustrate how this additional storage abstraction layer can further compound the detrimental effects of storage misalignment and how proper alignment enables optimal I/O performance.

Figure 5) Physical file system layout, properly aligned.

Figure 6) Virtual file system layout, not aligned.
4.5 BOOT IMAGE CLONE PREPARATION

Operating system installations typically contain identification and configuration data that is unique to each installation and associated to specific underlying hardware. It is important to prepare these aspects of the operating system installation prior to using it as a source for cloning operations. If this unique identification and configuration data is not properly prepared, conflicts can arise when using clones of the OS installation on additional servers or different hardware. The following sections provide further information on preparing various types of operating system installations for cloning.

WINDOWS IMAGE PREPARATION

Microsoft provides a set of tools, called Sysprep (sysprep.exe) and Setup Manager (setupmgr.exe), which are used to prepare a Windows installation for deployment to multiple computers. Microsoft distributes these utilities with the installation media for each Windows platform version and service pack. Sysprep removes system-specific information from the Windows installation so that the same image can easily be deployed to different systems. Sysprep assigns a unique security identifier (SID) to the Windows installation and configures the system to request new setup information the next time it is restarted. Setup Manager can be used to automate delivery of this new setup information. Setup Manager will ask a series of questions and generate a configuration file (sysprep.inf) containing all necessary information requested in Windows setup. After the system starts from a Sysprep-configured installation, Windows setup automatically retrieves the necessary values from the configuration file.

NOTE: This procedure can be used to prepare both physical and virtual machine Windows installations for clone deployments. Some virtualization hypervisors also include integrated tools to further assist with these preparations, such as VMware’s Customization Specifications.

**Image Preparation Steps**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Install the desired version of Windows operating system and make any necessary customizations (that is, install updates, applications, and so on)</td>
</tr>
<tr>
<td>2.</td>
<td>Create a Sysprep folder on the %systemdrive% of the installed platform (for example, C:\Sysprep).</td>
</tr>
<tr>
<td>3.</td>
<td>From the Windows platform installation media, extract the contents of \Support\Tools\Deploy.cab to the Sysprep folder.</td>
</tr>
<tr>
<td>4.</td>
<td>From the extracted Sysprep files, start the Setup Manager wizard (setupmgr.exe) and click <strong>Next</strong> to proceed.</td>
</tr>
</tbody>
</table>

Figure 7) Virtual file system layout, aligned.
5. Select **Create new** to generate a new answer file and click **Next** to continue.

6. Select **Sysprep setup** as the type of answer file to generate and click **Next**.
7. Select the type of Windows platform installation, then click **Next**.

8. Choose **Yes, fully automate the installation**, then click **Next**.
9. The remaining questions collect the configuration information that would otherwise be requested during the course of an interactive Windows setup. Once complete, a `sysprep.inf` file is created in the Sysprep folder.

**NOTE** Sysprep will automatically remove the Sysprep folder and all contents when it executes, so save a copy if you would like to reuse this automated setup configuration on another master installation.

10. Verify that all desired OS and preinstallation are correct, then execute `sysprep.exe` in the Sysprep folder.

11. Click **OK** to proceed.

12. Make sure that **Shut down** is selected as the **Shutdown mode** and click **Reseal**.
13. Click OK to confirm.

The Sysprep utility will reconfigure and shut down the system. The master installation is now ready to be cloned and deployed to multiple systems.

NOTE: If the master installation is booted again, these Sysprep procedures must be repeated to reconfigure the installation for cloning. If the master installation requires further customization, a good practice is to preserve the original master in state and instead boot and customize a clone of the original. This approach provides a versioning style of management to boot image preparation.

**LINUX AND UNIX IMAGE PREPARATION**

Linux and UNIX® platforms do not provide mass deployment configuration tools analogous to Sysprep for Windows. However, similar preparations are required in order for cloned boot images of these platforms to function properly. A common problem with cloning Linux and UNIX installations involves Ethernet MAC addresses stored within network interface configuration files. If these stored values are replicated by clone deployments to multiple computers, the cloned Linux and UNIX OS will consider them as legacy interfaces and will create additional Ethernet interface configurations for the MAC addresses discovered on the new hardware. For example, a master boot image created on a machine with two Ethernet interfaces will have two interfaces configured, named eth0 and eth1. If this master boot image is cloned without modification and deployed to a different computer with two Ethernet interfaces, the cloned OS will recognize the legacy interface configurations, regardless of the change in underlying hardware, and also discover the two new MAC addresses and create additional interface configurations named eth2 and eth3. The eth0 and eth1 configurations would persist as “stale” interfaces only because the legacy MAC address information was not removed from their configuration files before cloning. Although this issue might not affect the fundamental network connectivity of the cloned OS, any applications or services that were initially configured using the now stale interfaces might not be able to adapt to these changes. To eliminate these issues and make sure that existing network interfaces are configured in place after cloning, remove all occurrences of MAC address values from the master boot image when preparing the OS installation for cloning. This configuration information is typically located in one or more files within the /etc directory tree, varying with Linux and UNIX platforms. The following procedure can be used to locate any files that contain MAC address values within the /etc directory.

NOTE: Commands and available switches might vary for the procedure below depending on version and
1. Use `ifconfig` to list all network interface MAC addresses.

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:F7:B9:9A
          inet addr:10.60.119.191  Bcast:10.60.119.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe7:b99a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1523 errors:0 dropped:0 overruns:0 frame:0
          TX packets:295 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:110481 (107.8 KiB)  TX bytes:40827 (39.8 KiB)
          Interrupt:177 Base address:0x1400
```

2. Within applicable operating systems, use `grep` with the following parameters to locate all occurrences of each MAC address within the `/etc` directory. Enter the MAC address in the correct format using colons between each byte as seen below.

```
grep -ri <MAC Address> <search directory>
```

```
#grep -ri 00:0C:29:F7:B9:9A /etc
```

### 4.6 BLOCK-LEVEL CLONING

**OVERVIEW**

NetApp Data ONTAP provides block-level cloning capabilities to efficiently replicate storage-level objects, such as LUNs and volumes. LUNs can store block-level OS or application data for iSCSI- or FC-attached systems. Remote systems can directly access OS or application data on volumes using NAS protocols such as CIFS and NFS. NetApp FlexClone technology can be used to efficiently and quickly clone a LUN or a volume object regardless of the data it stores. Two methods are available to clone block-level objects, with and without a dependency upon persistent Snapshot copies. This section focuses on the method that requires the use of persistent Snapshot copies, as the second method is exactly the same as the file cloning method in the section above. The only difference from file-level cloning is that the source and destination paths will be LUN paths rather than individual file paths.

**REQUIREMENTS**

**Persistent Cloning Based on Snapshot**

This cloning method requires the use of persistent NetApp snapshot copies to clone individual LUNs or volumes.

- NetApp Data ONTAP version 7.0 and beyond
- NetApp FlexClone license
- NetApp FCP and/or iSCSI license (only if objects being cloned are LUNs)

**Nonpersistent Cloning Based on Snapshot**

This cloning method does not require the use of persistent NetApp Snapshot copies to clone master images or volumes, leveraging the same technology applied toward file-level cloning. The only difference is that the source and destination paths will be full paths to LUNs or a single volume name depending on the object type being cloned.

- NetApp Data ONTAP version 7.3.1 and beyond
- NetApp FlexClone license
• NetApp FCP and/or iSCSI license (only if objects being cloned are LUNs)

**PROCEDURE**

<table>
<thead>
<tr>
<th>Cloning Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> After preparing a master boot image, unmap the newly created gold LUN from the current igroup using the <code>lun unmap</code> command.</td>
</tr>
</tbody>
</table>
| NetApp> lun unmap /vol/test/win2k3_gold initiator1  
Thu Mar 12 09:38:00 GMT [lun.map.unmap:info]: LUN /vol/test/win2k3_gold unmapped from initiator group initiator1 |
| **2.** In this example, the gold LUN `/vol/test/win2k3_gold` resides in the test volume. Using the `snap create` command, create a Snapshot copy of the volume that contains the gold LUN:

`snap create <volume> <snap name>`

Observe the created Snapshot copy with the `snap list` command:

`snap list <volume>`

| NetApp> lun show  
/vol/test/win2k3_gold 10.0g (10742215680) (r/w, online) |
| NetApp> snap create test GOLD |
| NetApp> snap list test  
Volume test  
working... |
| %/used %/total date name  
---------- ---------- ----------  
48% (48%) 0% ( 0%) Mar 12 08:00 hourly.0  
65% (48%) 0% ( 0%) Mar 12 00:00 nightly.0  
74% (48%) 0% ( 0%) Mar 11 20:00 hourly.1  
79% (48%) 0% ( 0%) Mar 11 16:00 hourly.2  
82% (49%) 0% ( 0%) Mar 11 12:00 hourly.3  
85% (48%) 0% ( 0%) Mar 11 08:00 hourly.4  
87% (48%) 0% ( 0%) Mar 11 00:00 nightly.1  
88% (48%) 0% ( 0%) Mar 10 20:00 hourly.5 |
| **3.** Now that a Snapshot copy of the volume has been created, create a clone of the gold LUN that references the Snapshot copy. Use the `lun clone create` command:

`lun clone create <clone_lunpath> [-o noreserve] -b <parent_lunpath> <parent_snap>`

| NetApp> lun clone create /vol/test/win2k3_clone -o noreserve -b /vol/test/win2k3_gold GOLD |
| NetApp> lun show  
/vol/test/win2k3_clone 10.0g (10742215680) (r/w, online)  
/vol/test/win2k3_gold 10.0g (10742215680) (r/w, online) |
| NetApp> lun show -v /vol/test/win2k3_clone |
4. • Now the clone can be mapped to desired iSCSI or FC initiator groups using the `lun map` command. This procedure can be repeated and scripted to create large numbers of clones referencing the same Snapshot copy.

4.7 FILE-LEVEL VIRTUAL MACHINE CLONING

OVERVIEW

File-level FlexClone is a NetApp technology that enables thin-provisioned cloning of individual files. This capability has many beneficial applications and is particularly useful for efficiently provisioning the files that support virtual machines. Most virtualization hypervisors include native features for cloning virtual machines, but use extremely inefficient methods to replicate the underlying virtual machine data. Native hypervisor tools often employ “thick” cloning methods by merely copying the files that compose the virtual machine, consuming additional storage with redundant data. Leveraging the NetApp file-level FlexClone technology to replicate virtual machine files in conjunction with native hypervisor VM import procedures provides an extremely rapid and efficient solution for virtual machine cloning with minimal storage consumption.

REQUIREMENTS

• NetApp Data ONTAP version 7.3.1 or later
• NetApp FlexClone license
• NetApp NFS or CIFS license (depending on hypervisor)

PROCEDURE

The following example provides a step-by-step procedure for using the NetApp file-level FlexClone technology to clone a virtual machine on VMware Virtual Infrastructure 3. This technology can also be leveraged with other hypervisor types. The procedure can easily be automated through APIs available for NetApp Data ONTAP and the corresponding hypervisor. NetApp also offers a free VMware Virtual Center plug-in called the Rapid Cloning Utility (RCU), which automates the cloning procedure below using NetApp FlexClone technology. The RCU is available for download from the NetApp NOW ToolChest.

**File-level Cloning Procedure**

1. A single Windows Server 2003 Standard Edition virtual machine with a 10GB virtual disk will be used as the clone source. This master virtual machine is named `vm_gold`.
Each ESX host in the VMware cluster has two NFS data stores attached, gold and vm_1. Both data stores are NFS exports from a NetApp controller and are qtrees that reside in the same flexible volume on storage. NetApp file-level FlexClone technology requires both source and destination file paths to reside in the same flexible volume. In this example, the master virtual machine vm_gold resides on the gold. The clone virtual machine will be named vm_clone_1 and will reside in the vm_1 data store.

Note: The source and destination files may also reside within the same data store; however, this example uses qtrees to illustrate how VMs may also be cloned across data stores.

<table>
<thead>
<tr>
<th>Storage</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>Device</td>
<td>Capacity</td>
<td>Free</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>NFS</td>
<td>ktc5b1</td>
<td>90.75 GB</td>
<td>60.20 GB</td>
</tr>
<tr>
<td>gold</td>
<td>192.168.10.203/vmfs</td>
<td>380.00 GB</td>
<td>239.36 GB</td>
</tr>
<tr>
<td>vm_1</td>
<td>192.168.10.203/vmfs</td>
<td>380.00 GB</td>
<td>239.36 GB</td>
</tr>
</tbody>
</table>

2. From the ESX command line, navigate to /vmfs/volumes. This directory lists all the data stores that are currently connected to this ESX host.

```
[root@ktc5b1 volumes]# pwd
/vmfs/volumes
```

```
[root@ktc5b1 volumes]# ls
496433c4-cd70ad88-7059-001a648abb18
f1360744-795c1cc2
f621d74d-155b07bd
gold
ktc5b1:storage1
vm_1
```

3. Navigate into the gold data store that contains a directory for each VM within that data store. In this example, the gold data store contains a directory for the master virtual machine vm_gold.
Navigate into the `vm_gold` directory and note the collection of files that compose the virtual machine. Here is a short description of some of the VMware file types:

- `*.vmdk` – virtual disk descriptor file (specifies geometry and information about `*.vmdk`)
- `*.flat.vmdk` – virtual disk data file
- `*.vmx` – virtual machine configuration file
- `*.nvram` – stores the state of the virtual machine BIOS
- `*.vswp` – virtual machine hypervisor swap file created when VM is powered on (note this is not the paging/swap data managed by the guest OS)

File-level FlexClone technology will be used to create a "thin" clone of the `*.vmdk` and `*.flat.vmdk` files. This will result in two sets of `*.vmdk` and `*.flat.vmdk` files that will initially share the same underlying storage blocks. VMware’s native features will first be used to clone the remaining virtual machine files.
4. To omit the virtual disk descriptor and data files from the VMware cloning method, the virtual disk must be detached from the VM configuration. From the Virtual Infrastructure Client, power down the VM and click **Edit Settings**…

4. In the **Hardware** tab of the VM properties panel, select **Hard Disk 1** in the left pane and then click **Remove**.

5. The **Hard Disk 1** hardware will be struck out. Under **Removal Options**, select **Remove from virtual machine**. Important note: **Do not** select the option to also delete files from disk. Click **OK**.
Now that the virtual machine disk files have been detached from the VM configuration, use the native hypervisor tools to clone the VM. Remember that since the virtual disks have been removed, this will only clone the remaining VM files (*.vmx, *.nvram, and so on).

6. Right click the VM and choose **Clone…**

7. When the Clone Virtual Machine Wizard opens, enter the new clone name (vm_clone_1 in this example) and select the desired **Inventory Location**. Click **Next**.
8. Select the VMware Host or Cluster to place the VM clone. Click **Next**. If a VMware Cluster is selected, choose a specific VMware Host in the next window and click **Next**.

9. Select the appropriate data store to place the cloned VM configuration files. In this example, the cloned VM will be placed in the **vm_1** data store. Click **Next**.

10. In the next window, choose **Do not customize**. Click **Next**.

11. Review the summary and click **Finish**.
The new virtual machine clone will be defined in the Virtual Infrastructure Client.

From the ESX host command line, navigate to the directory for the cloned VM. In this example it is /vfms/volumes/vm_1/vm_clone_1/. The VM configuration files have been cloned, omitting the previously detached virtual disk files.

```
[root@ktc5b1 vm_clone_1]# pwd
/vfms/volumes/vm_1/vm_clone_1

[root@ktc5b1 vm_clone_1]# ls
vm_clone_1.nvram
vm_clone_1.vmsd
vm_clone_1.vmx
vmware-1.log
vmware-2.log
vmware.log
```

Now that the virtual machine configuration files have been cloned using the hypervisor’s native methods, use the NetApp FlexClone technology to efficiently clone the virtual disk files.

12. From the NetApp Data ONTAP command line, enable the advanced privilege mode. This mode is designated by an asterisk (*) in the command line prompt as illustrated below.

```
NetApp> priv set advanced
Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.
NetApp>
```
13. In this example, the `vm_gold.vmdk` and `vm_gold-flat.vmdk` virtual disk files will be cloned. You must include the fully qualified path (that is, `/vol/<volume>/<qtree>/<VM>/etc`) to both the source and destination files for the FlexClone operation. Use the `clone start` command as follows:

**Command usage:**
```
clone start <source file path> <destination file path>
```

**Example:**
```
clone start /vol/NFS_datastore_001/gold/vm_gold/vm_gold.vmdk
/vol/NFS_datastore_001/vm_1/vm_clone_1/vm_clone_1.vmdk
```

14. Perform the `clone start` operation on both the `*.vmdk` and `*–flat.vmdk` virtual disk files. Then navigate to the cloned VM directory on the ESX host (that is, `/vmfs/volumes/vm_1/vm_clone_1`) and confirm the presence of both the `*.vmdk` and `*–flat.vmdk` clone files.

```
[root@ktc5b1 vm_clone_1]# pwd
/vmfs/volumes/vm_1/vm_clone_1

[root@ktc5b1 vm_clone_1]# ls
vm_clone_1-flat.vmdk
vm_clone_1.vmdk
vm_clone_1.nvram
vm_clone_1.vmsd
vm_clone_1.vmx
vm_clone_1.vmxfs
vmware-1.log
vmware-2.log
vmware.log
```

15. Open the `*.vmdk` virtual disk descriptor file with a text editor and change the stored name of the original master `*–flat.vmdk` file to the new name of the cloned `*–flat.vmdk` file. In the example below `vm_gold-flat.vmdk` must be changed to `vm_clone_1-flat.vmdk`. If this change is not made, the virtual disk data file will not be recognized by ESX and cannot be attached to a VM.

```
# Disk DescriptorFile
version=1
CID=859cfc1c
parentCID=ffffffff
createType="vmfs"

# Extent description
RW 20971520 VMFS "vm_gold-flat.vmdk"

# The Disk Data Base
#DDB

ddb.toolsVersion = "7302"
ddb.adapterType = "lsilogic"
ddb.geometry.sectors = "63"
ddb.geometry.heads = "255"
ddb.geometry.cylinders = "1305"
ddb.uuid = "60 00 C2 9e 37 69 f0 26-45 fb ad d0 7c ec 2b 3b"
ddb.virtualHWVersion = "4"
```

Now that the virtual disk files have been cloned, they must be attached to the cloned VM.
configuration files.

16. In the Virtual Infrastructure Client, select the VM clone `vm_clone_1` and click **Edit Settings**...

17. On the **Hardware** tab of the VM properties panel, click **Add**...

18. Select **Hard Disk** as the type of device to add and click **Next**.
19. Select the **Use existing virtual disk** option and click **Next**.

20. Click **Browse** and select the cloned *.vmdk file. In this example **vm_clone_1.vmdk** was used. Click **OK**, then **Next**.
21. No changes are required in the Specify Advanced Options panel, so click Next.

22. Review the summary and click Finish.
23. Observe the new **Hard Disk** added in the hardware list and click **OK**.

24. The efficiently cloned virtual machine *vm_clone_1* is now prepared for power on and use.
APPENDIX

DOCUMENTATION

VMWARE
VMware Infrastructure 3 Documentation
http://vmware.com/support/pubs/vi_pages/vi_pubs_35u2.html


VMware Compatibility Guides
http://www.vmware.com/resources/compatibility

VMware Virtual Infrastructure 3 Documentation Library
http://www.vmware.com/support/pubs/vi_pubs.html

MICROSOFT
Microsoft Hyper-V: Supported Guest Operating Systems

Microsoft Windows Server 2008/Hyper-V Technical Resource Library
CITRIX
Citrix XenServer 5: Requirements and Specifications

http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=1681139

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer’s responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. This document and the information contained herein must be used solely in connection with the NetApp products discussed in this document.