iSCSI Software Boot Deployment Example for Linux Operating Systems

0.80

Last Saved 11/30/2005 12:17 PM

First Edition (October 2005)

Revision Log

Version	**Content**
0.10	Creation (based off of iscsi linux wp 0.00.00.91) What is here: In band and Out of band discussion, lloyds tool introduction, both DHCP methods What is to be done: Lloyd tool example, OS image specifics(deltas from NBP), OOB failures What to be considered: An appendix with quick start/worksheets for the admin
0.20	Changed structure of document Updated the SLES imaging Updated the blade config Updated RHEL (still a work in progress) Added error trees for config deployment
0.30	Cleaned up links Added temp info for RedHat…full RedHat description 0.40 targeted for 9/22/05
0.40	Clarified RHEL
0.50	Noted the restriction of iSCSI software boot when iSCSI HBA present
0.60	Added info on using static IP addresses for SLES Noted restrictions with em64T and BSE
0.70	Updates and fixes for Redhat deployments
0.80	Update restrictions concerning Blade diagnostics Update restrictions concerning continual/repetitive boot stress testing

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1.0 Intended Audience and Document Map

This document is intended to provide the foundation for deploying iSCSI in BladeCenter as a boot service using the software based iSCSI initiator. This document is broken into three sections: Section 1 provides an overview of the solution. Section 2 provides the details involved with deploying iSCSI software boot using in-band configuration. Section 3 provides the details involved with deploying iSCSI software boot using out-of-band configuration.

1.1 Document Scope for the Novice

For the novice interested in in-band configuration, Chapters 2.0 Introduction to iSCSI Solutions for Bladecenter and 3.0 iSCSI Software Boot via In-band provide introductory and conceptual view of the steps required to deploy iSCSI boot solutions in BladeCenter. Chapters 3.0 iSCSI Software Boot via In-band and 4.0 iSCSI Software Boot via Out-of-band provide an overview of iSCSI deployment using in band configuration and iSCSI deployment using out of band configuration. Chapter 5.0 Deployment Steps provides detail discussions on each of the steps. Chapter 6.0 Restrictions and Debug Strategies present limitations and debug strategies.

1.2 Document Scope for the Expert

For the expert interested in in-band configuration, already fluent in Linux and at least familiar with iSCSI concepts, it may be appropriate to go straight to chapter 3.0 iSCSI Software Boot via In-band to determine the exact steps involved and use chapter 5.0 Deployment Steps for detailed references and 6.0 Restrictions and Debug Strategies as a guide for limitations and debug strategies.

For the expert interested in out-of-band configuration, already fluent in Linux and at least familiar with iSCSI concepts, it may be appropriate to go straight to chapter 4.0 iSCSI Software Boot via Out-of-band to determine the exact steps involved and use chapter 5.0 Deployment Steps for detailed references and 6.0 Restrictions and Debug Strategies as a guide for limitations and debug strategies.

2.0 Introduction to iSCSI Solutions for Bladecenter

The integration of iSCSI solutions into BladeCenter introduce a new paradigm for provide storage to BladeCenter Blades. Today, a BladeCenter blade can either use Direct Attach Storage (DAS) via disks residing on a given blade or use a Storage Area Network (SAN) via a Fibre Channel (FC) fabric. While DAS is very cost effective, it has limited sharing and scalability attributes while FC SANs offer excellent sharing and scalability at a premium. An iSCSI solution in BladeCenter offers an optimal solution be providing sharing and scalability while being more cost effective.

In an iSCSI solution, iSCSI initiators, residing as software on a generic Network Interface Controller (NIC) or as hardware Host Bus Adapter (HBA) on a given blade, can access iSCSI targets, residing external to the given blade and providing the actual storage capacity, over standard Ethernet fabrics. With the storage behavior similar to a FC SAN at more cost effective implementations, opportunities to provide shared scalable storage to wider set of implementations are possible. The SAN behavior of sharing the storage, scaling the storage into the Terabyte (TB) range, and performing content management such as snapshot, backup, and provisioning provide the solution with a rich set of tools to better manage the content. The cost effective nature of Ethernet and Internet Protocol (IP) networks provide a more optimal storage solutions on common and pervasive transport.

This white paper discusses how to implement an IP based SAN in the BladeCenter paradigm. Several usage models are discussed as well as the components required and the implementation steps are discussed.

2.1 Using iSCSI for System Boot

Using iSCSI for system boot involves configuring the system to use an iSCSI remote target as the storage hosting the operating system and such. With the storage remote to the blade, there is an opportunity to remove the disk drives residing on a given blade and, thus, open up the BladeCenter solution to diskless blades and provisioning.

2.1.1 Overview of System Boot Options

There are two dimensions possible in defining the methods to use iSCSI to boot a system up. The first dimension articulates the method that iSCSI configuration parameters are acquired at boot time. The second dimension articulates the implementation of iSCSI initiator used to access the remote disk access.

There are three methods for obtaining iSCSI configuration parameters. The first method, defined as “in-band”, uses network services such as DHCP to acquire the iSCSI configuration parameters. The second method, defined as “out-of-band”, uses the BladeCenter internal management fabric to pass down iSCSI configuration parameters. The third method, defined as “manual”, uses panels native to BIOS or iSCSI initiator to enter in iSCSI configuration parameters. Each of these methods can provide value to the appropriate customer. For example, customers comfortable with network services may favor in-band configuration while customers more interested in security may favor out-of-band or manual methods of configuration

There are two implementations of iSCSI that can be used for iSCSI boot. The first method, defined as “iSCSI software boot”, provides iSCSI capabilities by running iSCSI software on the blade processor and using standard NICs as transport off the blade. The second method, defined as “iSCSI hardware boot” provides iSCSI capabilities by offloading the entire iSCSI service and transport onto an iSCSI HBA.

These two dimensions provide significant flexibility to the customer. As part of the iSCSI initiator for BladeCenter, a unified approach to these dimensions provides a cohesive solution for the customer to leverage the best configuration approach and best iSCSI implementation for their needs. Over time, as various pieces of the unified approach become available, this document will be revised to include them so that a current perspective of system booting is available to the customer.

This paper focuses on the steps to deploy an iSCSI boot solution via a software initiator using either an in-band configuration approach or out-of-band approach.

2.1.2 System Boot using iSCSI Software Initiator

This system boot scenario involves providing the ability at boot time for the blade's Basic Input Output System (BIOS) to access remote storage residing on an external target. With this capability, the BIOS can load the operating system and applications of choice to a given blade that does not contain any local disks. Once the operating system (OS) is loaded and with the paging resource defined, the OS uses the remote storage as if it is a local hard disk.

To accomplish this, the iSCSI solution, implementing an IP SAN solution, involves the BIOS level iSCSI driver to discover and access the appropriate iSCSI target. Once that target is accessed, through the setup of an iSCSI session, the storage residing on the appropriate iSCSI target appears to the BIOS as a local hard drive. The BIOS accesses the appropriate iSCSI target to retrieve the operating system Master Boot Record (MBR) which starts the loading of the operating system. As the operating system loads, at the appropriate point, an enhanced OS driver is used to maintain connectivity to the iSCSI target for OS usage.

Since the remote iSCSI target appears to the BIOS and operating system as a local drive, the disk image residing on the remote iSCSI target must "look" identical to the disk image that would be provided by a local disk. To create this "look", tools are used to create this disk image and have that image placed on the on the iSCSI target. With this disk image in place, the BIOS and operating system can treat the iSCSI target as if it was a local hard disk.

The steps involved with deploying an iSCSI solution to support system booting are discussed in the following subsections and sections.

2.1.3 In-Band iSCSI Configuration for System Boot

One of the key aspects of iSCSI boot is the determination of the appropriate iSCSI target. As stated earlier, there are several ways to provide the appropriate configuration information to the BIOS level iSCSI driver. In-band configuration is one supported approach for BIOS level iSCSI driver to acquire iSCSI system boot parameters.

In-band configuration leverages networks services such as the Dynamic Host Configuration Protocol (DHCP) protocol to manage and present the iSCSI boot configuration to BIOS level iSCSI driver. In essence, when the BIOS level iSCSI driver starts up, it uses the DHCP network protocol to identify and retrieve iSCSI boot configuration parameters from the selected DHCP server on the network. Drilling down on this taxonomy, there are methods of in-band configuration: Method 1 uses standard base DHCP options to provide a very basic configuration. Method 2 uses standard vendor DHCP options to provide a rich flexible configuration.

The advantages to this method of system boot configuration include a centralized parameter management resource for the coordination of system boot of all servers and the ability to integrate into the current network resource configuration architecture the customer may have in place. From a centralized and current configuration architecture perspective, in-band configuration provides a way to leverage and continue to standardize on standard configuration architecture.

One disadvantage to this method of system boot configuration is that it lacks of a reasonable security model since passing authentication parameters via DHCP exposes them to all stations on the network. Another disadvantage involves the concern of network behavior in terms of traffic storms when many servers boot up concurrently and in terms of configuration ambiguity with assigned IP addresses and network station movement.

2.1.4 Out-of-Band iSCSI Configuration for System Boot

Realizing that some customers may view the in-band configuration approach disadvantages as significant, another supported configuration approach is out-of-band iSCSI configuration.

Out-of-band configuration leverages the BladeCenter internal management resources to manage, deploy, and present iSCSI system boot parameters to the BIOS level iSCSI driver. In essence, a configuration tool uses the BladeCenter management infrastructure to deploy iSCSI parameters to a given Blade's non volatile memory space in the service processor such that these parameters can be used at boot by the BIOS level iSCSI driver to find the appropriate iSCSI target.

The advantages to this method of system boot configuration include a strong security model and the prevention of any network traffic congestion during boot due to many servers booting concurrently. From a security model perspective, authentication parameters can be deployed such that not all network stations have visibility to them. From a traffic congestion perspective, the network transactions to identify the DHCP server and acquire the configuration parameters are eliminated.

One disadvantage to this method of system boot configuration is that it relies on more complex configuration management model involving several management subsystem components in BladeCenter.

3.0 iSCSI Software Boot via In-band

This section describes how to using iSCSI for system boot using an in-band parameter configuration and iSCSI software boot services.

3.1 iSCSI Software Boot via In-band Prerequisites

There are five major subsystems needed to support iSCSI software boot via in-band.

First, one or more recipient blades that will use the iSCSI remote storage as its boot resource must be present in the BladeCenter chassis. These blades must have the appropriate level of BIOS that contains the BIOS iSCSI initiator. This level of BIOS is provided on newer levels of blades for BladeCenter but also can be downloaded for some older HS20 level blades by accessing **https://www-1.ibm.com/quickplace/iSCSIsw/Main.nsf** .

Second, one or more iSCSI target devices that will provide the iSCSI remote storage for all blades in all the BladeCenter chassis must be present and visible in the network. These iSCSI targets provide a common pool of storage that can be allocated as needed based on customer needs. It should be noted that each piece of storage destined for use by a recipient blade is uniquely created and can have a unique target name. To simplify the concept of a piece of storage, the term of Logical Unit Number (LUN) can be used. This term may be familiar to readers familiar with fibre channel storage. It should also be noted that the term iSCSI target references the entire storage appliance providing ALL the storage while the term iSCSI target LUN references a specific subset of storage providing by the storage appliance.

Third, network services specifically DHCP, will provide iSCSI configuration parameters to all requesting recipient blades must be present in the system. These network services receive requests from recipient blades asking for the iSCSI configuration parameters for the given recipient blade based on its identification.

Fourth, one producer blade that will aid in the creation of the disk images that will reside on the iSCSI target must be in the network during disk image creation but does not have to remain in the network after disk image creation and deployment is completed. The only difference between a recipient blade and producer blade is that the producer blade has local disks attached. As a result, once all the disk images have been created, the producer blade can be transformed, by removing the local drives, into a recipient blade or can be reallocated to another function within the organization

Fifth, and finally, one management/administration console to access the various subsystems in the solution must be present. This console provides control and monitoring of the resources.

In Figure 1: Solution Topology, the solution topology is shown with the various subsystems.

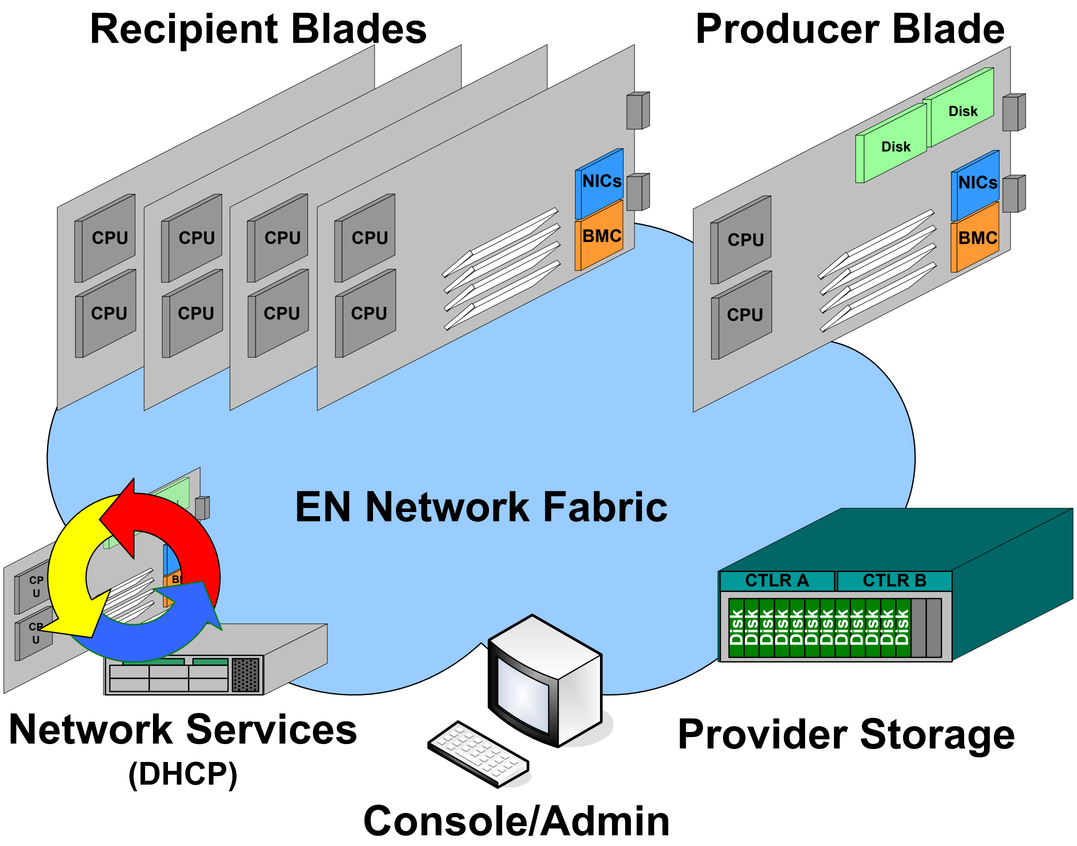


Figure 1: Solution Topology

3.2 iSCSI Software Boot via In-band Flow

The general execution flow for iSCSI software boot via in-band involves several steps.

Step 1 involves the creating the solution definition. Defining the solution involves establishing guidelines such as naming formats and establishing the subsystem relationships such as that between a given recipient blade and a given iSCSI target LUN relationships for each pair. For example, a recipient blade A with a unique iSCSI name and the corresponding iSCSI target LUN 1 with a unique name must be related to each other (details later).

Step 2 involves setting up the target by creating the LUNs with the appropriate iSCSI names, Access Control List (ACL) settings, and defining the size of the LUN. The iSCSI names and ACL settings should adhere to the solution definition desired by the administrator. For example, if the administrator plans to have a specific format for the iSCSI target name for each LUN, and then when creating the LUN, this format should be adhered to.

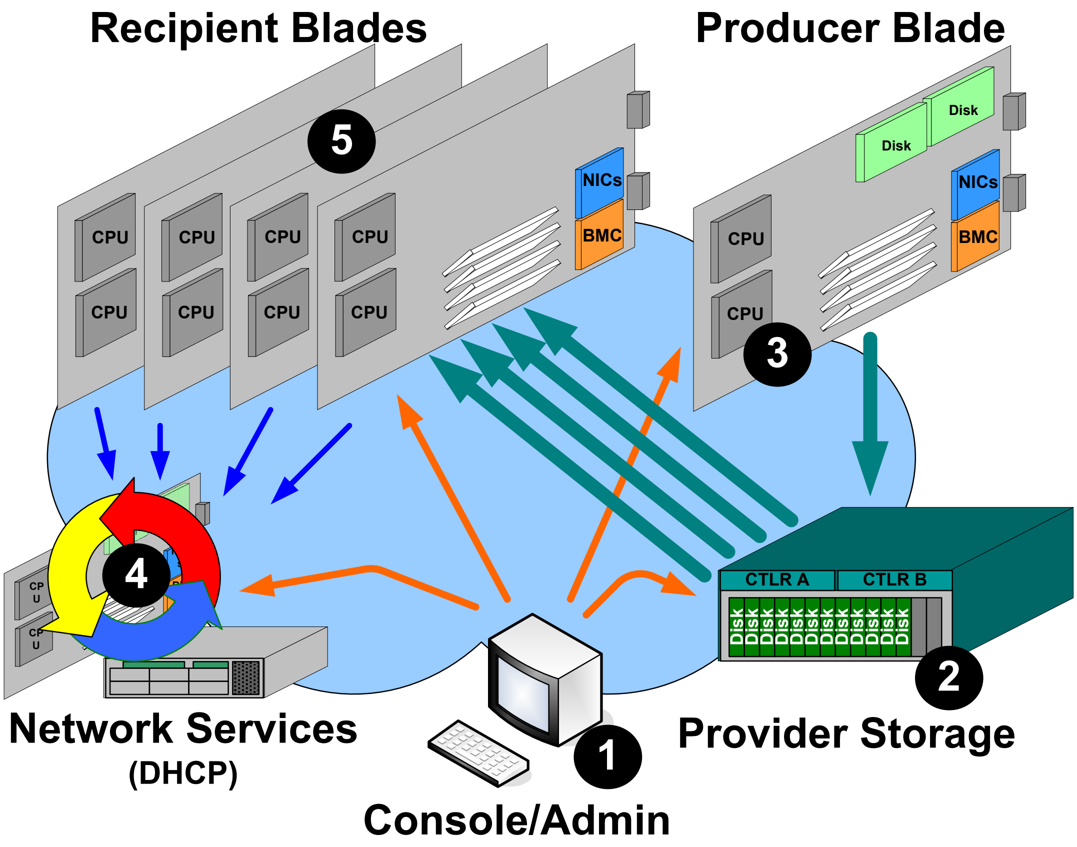
Step 3 involves creating the disk images to be placed on the iSCSI target. These disk images are created on the producer blade. Although the goal is get the OS installed, configured, and subsequently created on these disk images, the administrator can also choose to install appropriate applications at this time or can choose to install applications from the recipient blades at a later time.

Step 4 involves configuring the network services so that the recipient blades can retrieve their respective iSCSI configuration parameters and iSCSI initiator executable. Here too, the iSCSI naming conventions desired by the administrator should be adhered to so that iSCSI initiator and iSCSI target can successfully establish an iSCSI session.

Step 5, after the previous three steps have been completed, is to power up and configure the recipient blade. Once powered up, the recipient blade boot settings should indicate that iSCSI boot is desired. After iSCSI boot has been selected, the recipient blade will acquire the iSCSI configurations parameters from the network services then the recipient blade, at the BIOS stage, will access the iSCSI target to use the remote storage.

In Figure 2: Solution Flow, the general steps involved in iSCSI software boot via in-band is presented. In subsequent sections, each step is examined in more detail.

Figure 2: Solution Flow



3.3 iSCSI Software Boot via In-band Steps: Solution Definition

Examining the solution definition in more detail, the administrator needs to define many of the details for the iSCSI solution. Once these details or parameters are defined, the administrator needs to use them appropriately when configuring various subsystems. In Table 1: Basic iSCSI Parameters to be defined, the basic iSCSI parameters and their definitions are presented.

Parameter	**Definition**
Initiator IP address	The IP address a given recipient blade is to use when accessing the network

Initiator iSCSI name	The iSCSI name a given recipient blade is to use to access the iSCSI target
Initiator Storage	The amount of storage a given recipient blade is to use for operation
Network subnet	The IP network subnet used for the network where the solution resides
Target IP address	The IP address a given iSCSI target is to use when accessing the network
Target TCP port	The TCP port a given iSCSI target is to use when accessing the network
Target iSCSI name	The iSCSI name a given iSCSI target LUN is to use to present storage

Table 1: Basic iSCSI Parameters to be defined

As the administrator becomes more fluent with iSCSI, additional parameters are available to provide additional features and functions.

With the basic parameters defined, the solution definition is now possible. In Table 2: Example of Basic iSCSI Solution, a simple example of the solution definition is provided. Several points to note when examining this example: First, the IP address and iSCSI name for each recipient blade is different. Second, while the IP address of the iSCSI target is constant, the target iSCSI name is unique for each LUN presented to the network. The administrator should make the names short and reader friendly to be able to mentally parse this as the solution configuration proceeds.

Parameter Name	**Parameter Description**	**Parameter Value**
Network Subnet	Subnet containing all devices	1.1.1.0
Recipient Blade A		
Initiator Storage	Storage to be carved from iSCSI target	10GB
Initiator IP address	iSCSI Initiator IP address	1.1.1.1
Initiator iSCSI name	iSCSI Initiator iSCSI name	Iqn.2003-05.com.ibm.blade_a
Target IP address	iSCSI Target IP address	1.1.1.10
Target iSCSI name	iSCSI Target name	Iqn.2003-05.com.ibm.lun_a
Target iSCSI TCP port	iSCSI Target TCP port	3260
Recipient Blade B		
Initiator Storage	Storage to be carved from iSCSI target	10GB
Initiator IP address	iSCSI Initiator IP address	1.1.1.2
Initiator iSCSI name	iSCSI Initiator iSCSI name	Iqn.2003-05.com.ibm.blade_b
Target IP address	iSCSI Target IP address	1.1.1.10
Target iSCSI name	iSCSI Target name	Iqn.2003-05.com.ibm.lun_b
Target iSCSI TCP port	iSCSI Target TCP port	3260
Recipient Blade C		
Initiator Storage	Storage to be carved from iSCSI target	10GB
Initiator IP address	iSCSI Initiator IP address	1.1.1.3
Initiator iSCSI name	iSCSI Initiator iSCSI name	Iqn.2003-05.com.ibm.blade_c
Target IP address	iSCSI Target IP address	1.1.1.10
Target iSCSI name	iSCSI Target name	Iqn.2003-05.com.ibm.lun_c
Target iSCSI TCP port	iSCSI Target TCP port	3260

Table 2: Example of Basic iSCSI Solution

A detailed discussion on defining the solution is provided in section 5.2 Sample Deployment Solution Definition.

3.4 iSCSI Software Boot via In-band Steps: Target Configuration

Examining the target configuration in more detail, iSCSI target configuration involves the configuration of the iSCSI target storage controller as well as the creation of storage LUNs and configuring each of the LUNs per the solution definition. Several general steps are involved with configuring the iSCSI target and LUNs.

Step 1 involves initially logging into the target through the administrator interface. This interface can be via a Graphical User Interface (GUI) or via a Command Line Interface (CLI). Once logged in, the administrator can define the IP address for the target. Note that DHCP can be used for establishing IP addresses as well. Refer to appendix D for details on using DHCP for IP address definition.

Step 2 involves creating one or more storage LUNs from the general pool of storage. From the administrator console CLI or GUI, create storage amount of storage defined in the storage definition and set the Redundant Array of Inexpensive Disks (RAID) level as appropriate.

Step 3 involves configure one or more storage LUNs for usage. From the administrator console CLI or GUI, configure the iSCSI target name and ACL for each LUN created. These configuration settings provide the iSCSI target storage controller with the information to determine which LUN should be presented to which recipient blade.

A detailed discussion on defining the solution is provided in section 5.3 Sample Deployment Storage Configuration.

3.5 iSCSI Software Boot via In-band Steps: Disk Image Management

Examining the disk image management in more detail, the disk image management includes the creation of the disk image as well as the deployment of the disk image to the iSCSI target.

With the producer blade in place with its local disks, several general steps are involved with creating a deploying the disk image.

Disk Image Creation and Deployment

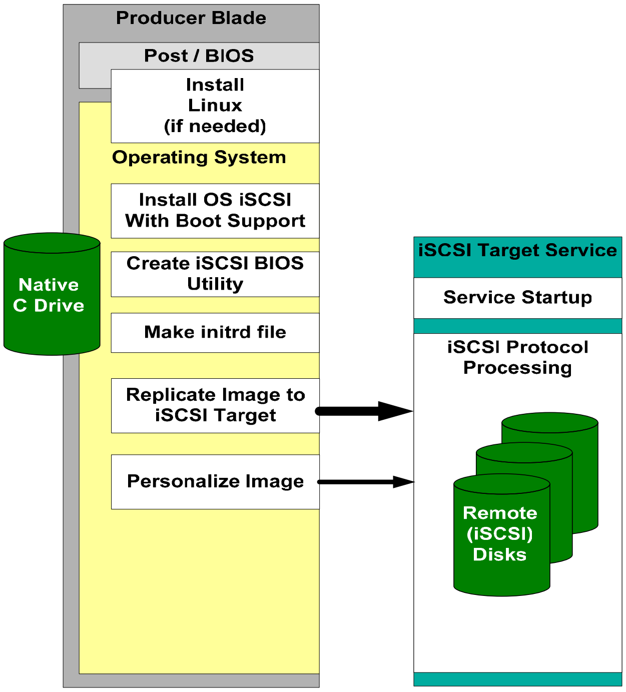


Figure 3: Disk Image Creation and Deployment

Step 1 involves the traditional installation of the Linux operating system on the producer blade (recall the producer blade has local disks). This involves the inserting the installation CD into the CD tray and proceeding through the install process. Once the operating system is installed, desired applications or appropriate drivers can be installed. In addition, any desired applications can be installed at this time.

Step 2 involves installing the operating system device driver supporting iSCSI boot. This driver is available either in the Linux distribution used at install or can be obtained from the source forge sit. This driver, providing sda drive functionality for Linux, is installed such that during recipient blade booting the transition between the BIOS functionality and operating system functionality is transparent to the user and applications. Once the driver is installed, appropriate configuration steps ensure the driver is integrated into the operating system.

Step 3 involves the creation of the initrd file used during the early Linux load phase during boot. Recall Linux has a two step process to Linux boot up. Phase 1 establishes a simple basic environment with minimal capabilities. Phase 2 uses the basic environment established in phase 1 to load the entire Linux kernel and all associated drivers and applications.

Step 4 involves connecting to the iSCSI target and replicating and personalizing the image for the recipient blade to use during booting. In addition, the initrd file needs to be placed on the TFTP server so that it is available at boot.

A detailed discussion on defining the solution is provided in section 5.4 Sample Deployment Disk Image Creation and Deployment.

3.6 iSCSI Software Boot via In-band Steps: Network Services

Examining network services configuration in more detail, the DHCP service is required for the configuration deployment.

3.6.1 DHCP Configuration Overview

The DHCP service provides the initial configuration parameters during the boot process. Basically, the recipient blade accesses the DHCP to acquire basic network parameters such as a TCP/IP address and pointers to the other subsequent services in the network.

DHCP configuration involves creating the DHCP configuration file entry for entry for each recipient blade and enabling the DHCP service on a blade or server. There are two general steps involved with creating and configuring the DHCP service.

Step 1 involves identifying the DHCP service in the network. This step entails establishing the DHCP service on a blade or server visible to all the subsystems in the solution. Access the blade or server providing DHCP services allows the administrator to configure the DHCP service and allows the recipient blades to access the DHCP service to retrieve iSCSI configuration parameters.

Step 2 involves defining the actual DHCP configuration. This step defines how the network and iSCSI configuration parameters are set and presented to the requesting recipient blade during boot. DHCP configuration is typically done via a simple text file or, as in the case of a windows DHCP server, via a GUI configuration aid. However, the given the flexibility of DHCP, this simple text file can be very complex.

Details of the DHCP configuration procedure is provided in the sample deployment section for both standard base DHCP options for simple configurations and for standard vendor DHCP options for rich configurations

A detailed discussion on defining the solution is provided in section 5.5 Sample Deployment Network Services Configuration.

3.7 iSCSI Software Boot via In-band Steps: Recipient Blade Boot Up

The system boot flow is depicted in Figure 4: iSCSI Boot Flow. This generalized flow is operating system independent and intended to provide background to the steps involved.

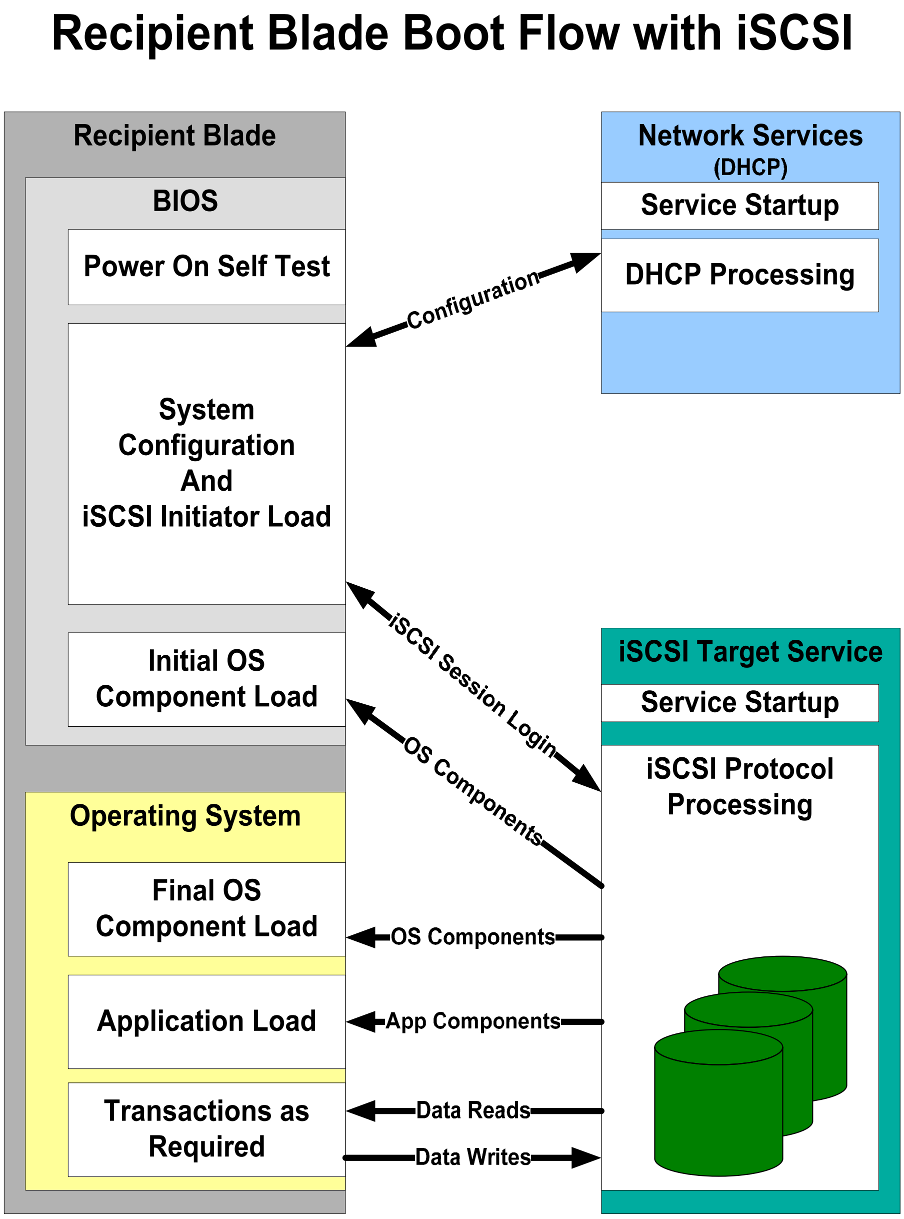


Figure 4: iSCSI Boot Flow

Step 1 involves the given recipient blade starting the power up and performs its Power On Self Test (POST). This step ensures the blade hardware is operating correctly.

Step 2 involves the given recipient blade starting the BIOS execution. The BIOS configures the hardware for appropriate operation. The configuration includes processor, memory, network interface controller (NIC) settings. Once the fundamental hardware is configured, the BIOS starts to set up the interrupt chains or trees for various services. One such service is the Interrupt 13 (INT13) used for presenting the BIOS with the storage resources available to the BIOS. At this point, the BIOS uses pre OS network drivers to access the network services to retrieve configuration parameters.

Step 3 involves the BIOS using the pre OS network driver to acquire the configuration. Within this step, the first task is to acquire the basic network parameters from DHCP.

Step 4 involves the given recipient blade establishing the connection with the iSCSI target. The BIOS initiator initializes and establishes the iSCSI session with the target using the configuration parameters acquired from network services. In order for the BIOS initiator to establish an iSCSI session, a light weight TCP service and a light weight iSCSI service is started. With these parameters and services in place, the initiator logs into the appropriate iSCSI target. With a successful login, the iSCSI initiator returns success back to the BIOS.

Step 5 involves the given recipient blade's BIOS requesting its BIOS iSCSI initiator to retrieve the Master Boot Record (MBR) for the operating system. The MBR contains the very initial instruction sequence of the operating system including what other disk blocks to read into the blade. As these additional disk blocks are read in, the BIOS can construct the image of the kernel loader. Once assembled, the kernel loader starts execution and the BIOS no longer participate in the system boot up.

Step 6 involves the given recipient blade's OS loader to pull in network services and iSCSI services. With theses services now in memory, the OS loader can start the TCP/IP network service, then, subsequently, start the iSCSI service for the "/sda" drive of the OS.

Step 7 involves the given recipient blade's OS to start other applications and finish the boot up sequence. Loading any subsequent applications and finishing the boot up places the OS in the state to start processing tasks spawned by users or applications.

A detailed discussion on defining the solution is provided in section 5.8 Sample Deployment Diskless Blade Configuration.

4.0 iSCSI Software Boot via Out-of-band

This section describes how to using iSCSI for system boot using an out-of-band parameter configuration and iSCSI software boot services.

4.1 iSCSI Software Boot via Out-of-band Prerequisites

There are five major subsystems needed to support iSCSI software boot via out-of-band.

First, one or more recipient blades that will use the iSCSI remote storage as its boot resource must be present in the BladeCenter chassis. These blades must have the appropriate level of BIOS that contains the BIOS iSCSI initiator. This level of BIOS is provided on newer levels of blades for BladeCenter but also can be downloaded for some older HS20 level blades by accessing **https://www-1.ibm.com/quickplace/iSCSIsw/Main.nsf** .

Second, one or more iSCSI target devices that will provide the iSCSI remote storage for all blades in all the BladeCenter chassis must be present and visible in the network. These iSCSI targets provide a common pool of storage that can be allocated as needed based on customer needs. It should be noted that each piece of storage destined for use by a recipient blade is uniquely created and has a unique target name. To simplify the concept of a piece of storage, the term of Logical Unit Number (LUN) can be used. This term may be familiar to readers familiar with fibre channel storage. It should be noted that the term iSCSI target references the entire storage appliance providing ALL the storage while the term iSCSI target LUN references a specific subset of storage providing by the storage appliance.

Third, iSCSI Configuration Manager that will provide iSCSI configuration parameters to all requesting recipient blades must be present in the system. This configuration manager uses the BladeCenter management infrastructure to deploy iSCSI configuration parameters to all the blades in the solution. When deployed, these parameters reside in the given blades service processor space and available for subsequent boots of the blade. Note that the iSCSI Configuration Manager can reside on the console platform if desired.

Fourth, one producer blade that will aid in the creation of the disk images that will reside on the iSCSI target must be in the network during disk image creation but does not have to remain in the network after disk image creation and deployment is completed. The only difference between a recipient blade and producer blade is that the producer blade has local disks attached. As a result, once all the disk images have been created, the producer blade can be transformed, by removing the local drives, into a recipient blade or can be reallocated to another function within the organization

Fifth, and finally, one management/administration console to access the various subsystems in the solution must be present. This console provides control and monitoring of the resources.

In Figure 1: Solution Topology, the solution topology is shown with the various subsystems.

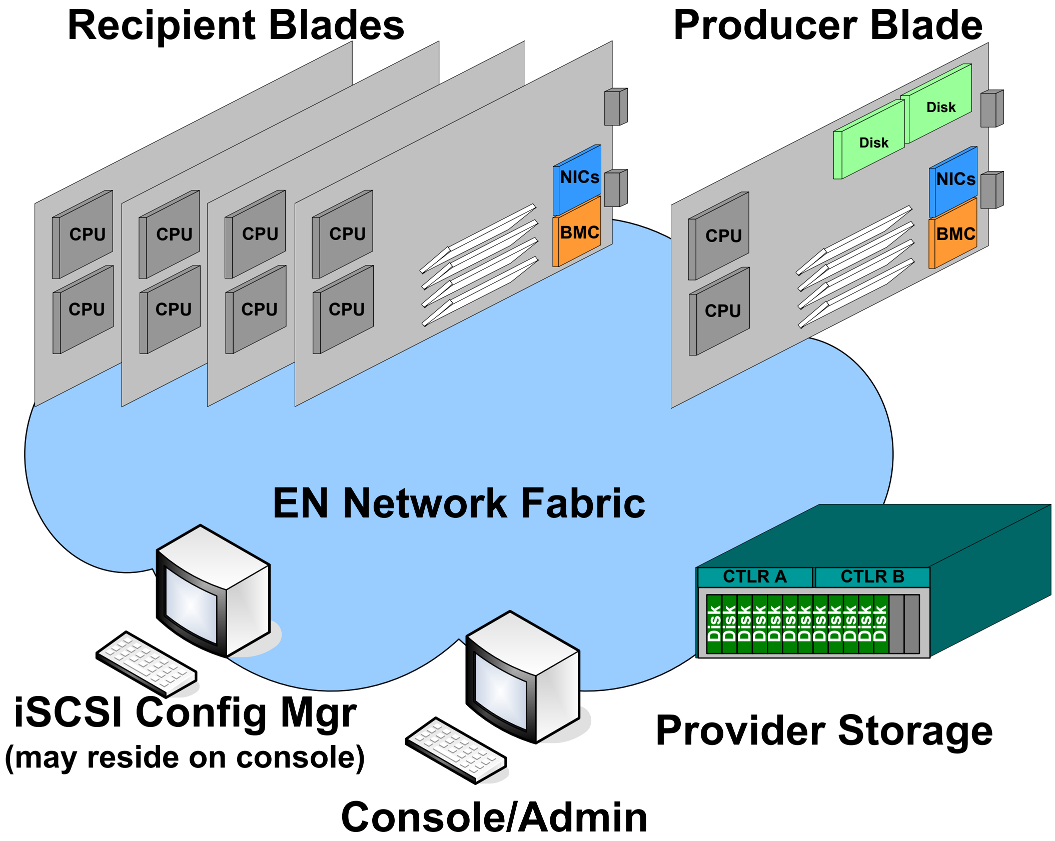


Figure 5: Solution Topology

4.2 iSCSI Software Boot via Out-of-band Flow

The general execution flow for iSCSI software boot via out-of-band involves several steps.

Step 1 involves the creating the solution definition. Defining the solution involves establishing guidelines such as naming formats and establishing the subsystem relationships such as that between a given recipient blade and a given iSCSI target LUN for each pair. For example, a recipient blade A with a unique iSCSI name and the corresponding iSCSI target LUN 1 with a unique name must be related to each other (details later).

Step 2 involves setting up the target by creating the LUNs with the appropriate iSCSI names, Access Control List (ACL) settings, and defining the size of the LUN. The iSCSI names and ACL settings should adhere to the solution definition desired by the administrator. For example, if the administrator plans to have a specific format for the iSCSI target name for each LUN, and then when creating the LUN, this format should be adhered to.

Step 3 involves creating the disk images to be placed on the iSCSI target. These disk images are created on the producer blade. Although the goal is get the OS installed, configured, and subsequently created on these disk images, the administrator can also choose to install appropriate applications at this time or can choose to install applications from the recipient blades at a later time.

Step 4 involves configuring the given blades using the iSCSI Configuration Manager so that the recipient blades are given the iSCSI configuration parameters upon startup. Here too, the iSCSI naming conventions desired by the administrator should be adhered to so that iSCSI initiator and iSCSI target can successfully establish an iSCSI session. Note that for some iSCSI targets, the iSCSI Configuration Manager will query the designated target to determine the iSCSI storage configuration that should be made available to the iSCSI initiators.

Step 5, after the previous three steps have been completed, is to power up and configure the recipient blade. Once powered up, the recipient blade boot settings should indicate that iSCSI boot is desired. After iSCSI boot has been selected, the recipient blade will first acquire the iSCSI initiator executable from network services then acquire the iSCSI configurations parameters from the network services then the recipient blade, at the BIOS stage, will access the iSCSI target to use the remote storage.

In Figure 2: Solution Flow, the general steps involved in iSCSI software boot via out-of-band is presented. In subsequent sections, each step is examined in more detail.

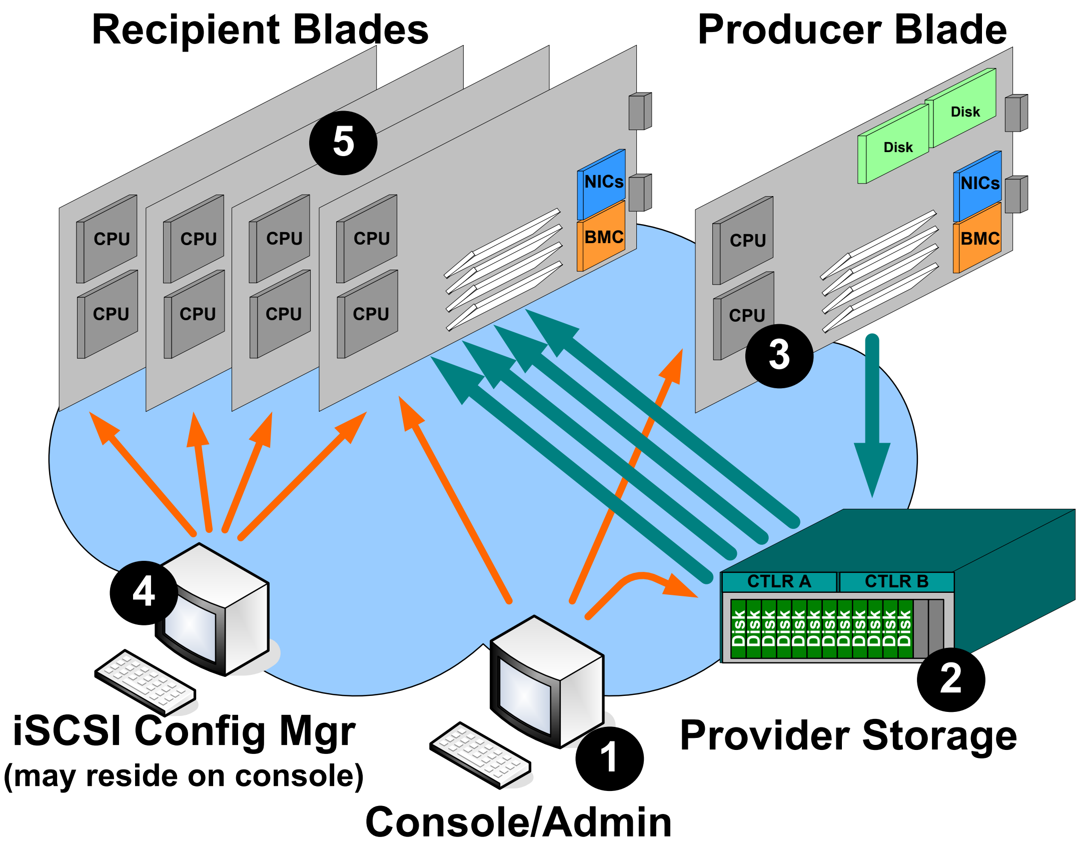


Figure 6: Solution Flow

4.3 iSCSI Software Boot via Out-of-band Steps: Solution Definition

Examining the solution definition in more detail, the administrator needs to define many of the details for the iSCSI solution. Once these details or parameters are defined, the administrator needs to use them appropriately when configuring various subsystems. In Table 1: Basic iSCSI Parameters to be defined, the basic iSCSI parameters and their definitions are presented.

Parameter	**Definition**
Initiator IP address	The IP address a given recipient blade is to use when accessing the network
Initiator iSCSI name	The iSCSI name a given recipient blade is to use to access the iSCSI target
Initiator Storage	The amount of storage a given recipient blade is to use for operation
Network subnet	The IP network subnet used for the network where the solution resides
Target IP address	The IP address a given iSCSI target is to use when accessing the network
Target TCP port	The TCP port a given iSCSI target is to use when accessing the network
Target iSCSI name	The iSCSI name a given iSCSI target LUN is to use to present storage

Table 3: Basic iSCSI Parameters to be defined

As the administrator becomes more fluent with iSCSI, additional parameters are available to provide additional features and functions.

With the basic parameters defined, the solution definition is now possible. In Table 2: Example of Basic iSCSI Solution, a simple example of the solution definition is provided. Several points to note when examining this example: First, the IP address and iSCSI name for each recipient blade is different. Second, while the IP address of the iSCSI target is constant, the target iSCSI name is unique for each LUN presented to the network. The administrator should make the names short and reader friendly to be able to mentally parse this as the solution configuration proceeds.

Parameter Name	**Parameter Description**	**Parameter Value**
Network Subnet	Subnet containing all devices	1.1.1.0
Recipient Blade A		
Initiator Storage	Storage to be carved from iSCSI target	10GB
Initiator IP address	iSCSI Initiator IP address	1.1.1.1
Initiator iSCSI name	iSCSI Initiator iSCSI name	Iqn.2003-05.com.ibm.blade_a
Target IP address	iSCSI Target IP address	1.1.1.10
Target iSCSI name	iSCSI Target name	Iqn.2003-05.com.ibm.lun_a
Target iSCSI TCP port	iSCSI Target TCP port	3260
Recipient Blade B		
Initiator Storage	Storage to be carved from iSCSI target	10GB
Initiator IP address	iSCSI Initiator IP address	1.1.1.2
Initiator iSCSI name	iSCSI Initiator iSCSI name	Iqn.2003-05.com.ibm.blade_b
Target IP address	iSCSI Target IP address	1.1.1.10
Target iSCSI name	iSCSI Target name	Iqn.2003-05.com.ibm.lun_b
Target iSCSI TCP port	iSCSI Target TCP port	3260
Recipient Blade C		
Initiator Storage	Storage to be carved from iSCSI target	10GB
Initiator IP address	iSCSI Initiator IP address	1.1.1.3
Initiator iSCSI name	iSCSI Initiator iSCSI name	Iqn.2003-05.com.ibm.blade_c
Target IP address	iSCSI Target IP address	1.1.1.10
Target iSCSI name	iSCSI Target name	Iqn.2003-05.com.ibm.lun_c
Target iSCSI TCP port	iSCSI Target TCP port	3260

Table 4: Example of Basic iSCSI Solution

A detailed discussion on defining the solution is provided in section 5.2 Sample Deployment Solution Definition.

4.4 iSCSI Software Boot via Out-of-band Steps: Target Configuration

Examining the target configuration in more detail, iSCSI target configuration involves the configuration of the iSCSI target storage controller as well as the creation of storage LUNs and configuring each of the LUNs per the solution definition. Several general steps are involved with configuring the iSCSI target and LUNs.

Step 1 involves initially logging into the target through the administrator interface. This interface can be via a Graphical User Interface (GUI) or via a Command Line Interface (CLI). Once logged in, the administrator can define the IP address for the target. Note that DHCP can be used for establishing IP addresses as well. Refer to appendix D for details on using DHCP for IP address definition.

Step 2 involves creating one or more storage LUNs from the general pool of storage. From the administrator console CLI or GUI, create storage amount of storage defined in the storage definition and set the Redundant Array of Inexpensive Disks (RAID) level as appropriate.

Step 3 involves configure one or more storage LUNs for usage. From the administrator console CLI or GUI, configure the iSCSI target name and ACL for each LUN created. These configuration settings provide the iSCSI target storage controller with the information to determine which LUN should be presented to which recipient blade.

A detailed discussion on defining the solution is provided in section 5.3 Sample Deployment Storage Configuration.

4.5 iSCSI Software Boot via Out-of-band Steps: Disk Image Management

Examining the disk image management in more detail, the disk image management includes the creation of the disk image as well as the deployment of the disk image to the iSCSI target.

With the producer blade in place with its local disks, several general steps are involved with creating a deploying the disk image.

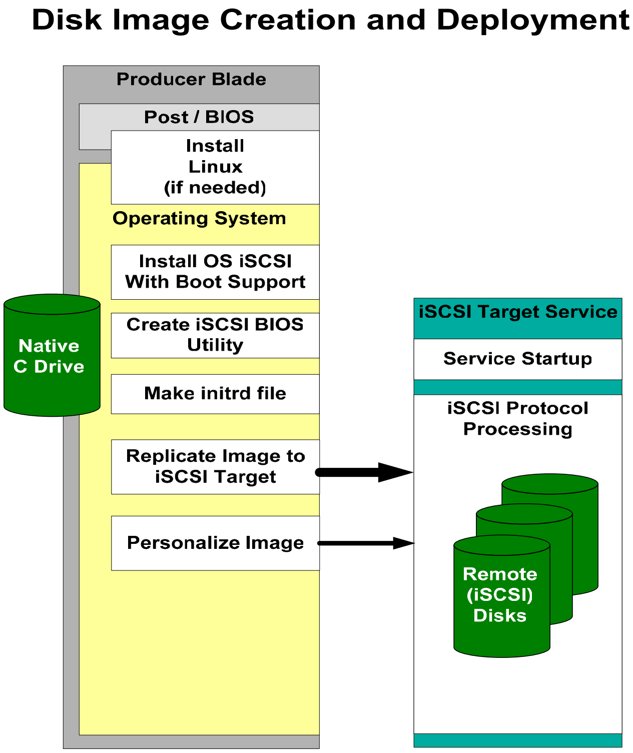

Disk Image Creation and Deployment
Producer Blade
Post / BIOS
Install Linux (if needed)
Operating System
Install OS iSCSI With Boot Support
Create iSCSI BIOS Utility
Make initrd file
Replicate Image to iSCSI Target
Personalize Image
Native C Drive
iSCSI Target Service
Service Startup
iSCSI Protocol Processing
Remote (iSCSI) Disks

Figure 7: Disk Image Creation and Deployment

Step 1 involves the traditional installation of the Linux operating system on the producer blade (recall the producer blade has local disks). This involves the inserting the installation CD into the CD tray and proceeding through the install process. Once the operating system is installed, desired applications or appropriate drivers can be installed. In addition, any desired applications can be installed at this time.

Step 2 involves installing the operating system device driver supporting iSCSI boot. This driver is available either in the Linux distribution used at install or can be obtained from the source forge sit. This driver, providing sda drive functionality for Linux, is installed such that during recipient blade booting the transition between the BIOS functionality and operating system functionality is transparent to the user and applications. Once the driver is installed, appropriate configuration steps ensure the driver is integrated into the operating system.

Step 3 involves the creation of the initrd file used during the early Linux load phase during boot. Recall Linux has a two step process to Linux boot up. Phase 1 establishes a simple basic environment with minimal capabilities. Phase 2 uses the basic environment established in phase 1 to load the entire Linux kernel and all associated drivers and applications.

Step 4 involves connecting to the iSCSI target and replicating and personalizing the image for the recipient blade to use during booting. In addition, the initrd file needs to be placed on the TFTP server so that it is available at boot.

A detailed discussion on defining the solution is provided in section 5.4 Sample Deployment Disk Image Creation and Deployment.

4.6 iSCSI Software Boot via Out-of-band Steps: Configuration Services

Examining iSCSI solution configuration in more detail, the out-of-band iSCSI configuration via the iSCSI Configuration Manager is used to configure many blades in a solution.

4.6.1 iSCSI Configuration Manager Overview

The iSCSI Configuration Manager (ICM) is a stand alone JAVA tool that can run on any computer that has connectivity to the BladeCenter. The ICM provides several ways to allow the administrator to remotely define and deploy the iSCSI configuration parameters to a given blade. Moreover, with supported iSCSI targets, the ICM can query those targets to determine the storage configuration implemented and allow the administrator to use those resources to support the given blades configuration. To provide rich management tracking, the iSCSI configuration can be saved to and retrieved from an eXtensible Markup Language (XML) file.

Step 1 involves installing and configuring the ICM for usage. This step entails installing the ICM on either the console computer (assuming a remote computer is used access the BladeCenter) or on a designated computer that will be used as the ICM platform. In either case, the ICM must have connectivity to the BladeCenter Management Module(s) and may have connectivity to the iSCSI target if it is desired to have ICM query the target for storage configuration information.

Step 2 involves defining the actual iSCSI configuration. This step defines how the network and iSCSI configuration parameters are set and deployed to the given blade prior to the blade boot. The ICM GUI can be used for a human friendly interactive configuration of the blade parameters. In addition, advanced users can develop the appropriate iSCSI configuration XML file and pipe it through the ICM to the BladeCenter.

A detailed discussion on defining the solution is provided in section 5.6 Sample Deployment Using iSCSI Configuration Manager.

4.7 iSCSI Software Boot via Out-of-band Steps: Recipient Blade Boot Up

The system boot flow is depicted in Figure 4: iSCSI Boot Flow. This generalized flow is operating system independent and intended to provide background to the steps involved.

Recipient Blade Boot Flow with iSCSI

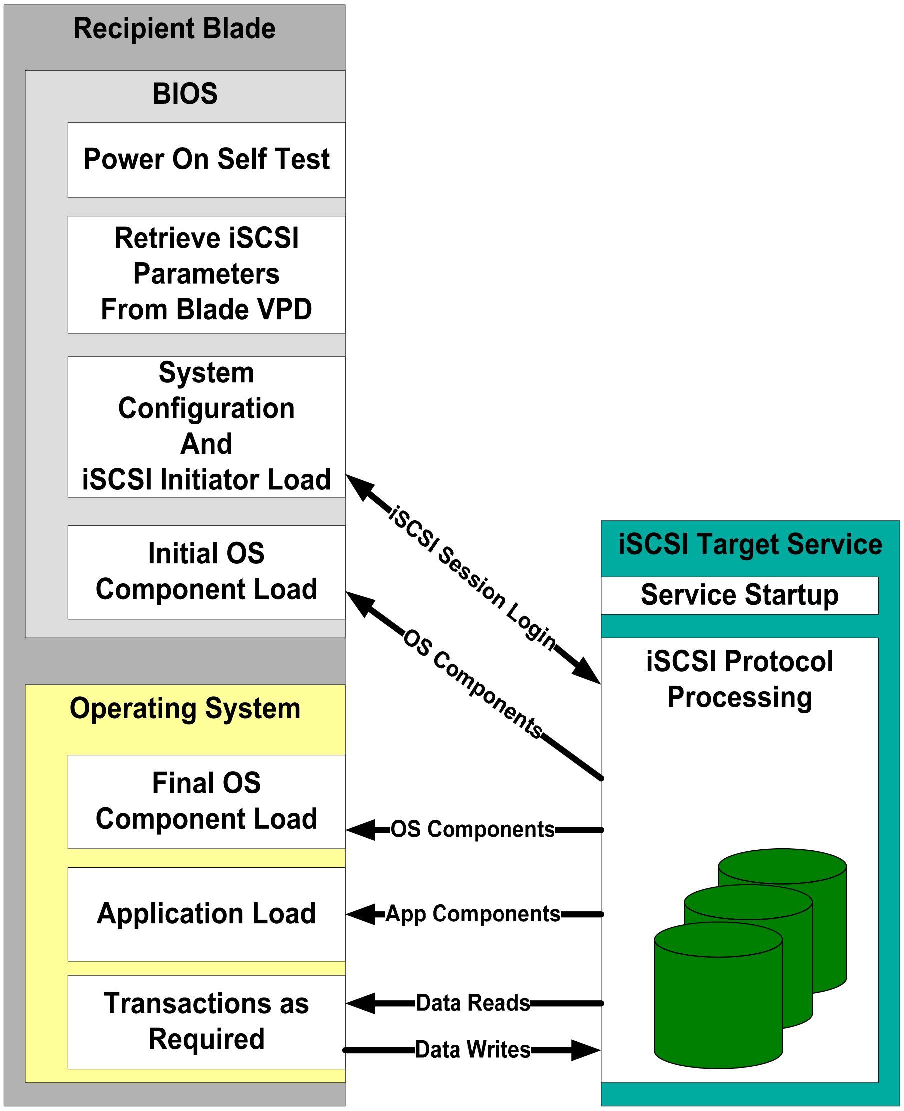


Figure 8: iSCSI Boot Flow

Step 1 involves the given recipient blade starting the power up and performs its Power On Self Test (POST). This step ensures the blade hardware is operating correctly.

Step 2 involves the given recipient blade starting the BIOS execution. The BIOS configures the hardware for appropriate operation. The configuration includes processor, memory, network interface controller (NIC) settings. Once the fundamental hardware is configured, the BIOS start to set up the interrupt chains or trees for various services. One such service is the Interrupt 13 (INT13) used for presenting the BIOS with the storage resources available to the BIOS. At this point, the BIOS retrieves the iSCSI configuration parameters from the service processor non volatile memory and passes these parameters, upon invocation, to the BIOS level iSCSI driver.

Step 3 involves the BIOS level iSCSI driver to validate the iSCSI parameters passed to it. With valid parameters in place, the BIOS level iSCSI driver proceeds to establish a session with the iSCSI target. With a successful negotiation and login into the target, the BIOS level ISCSI driver post a successful status message to the BIOS.

Step 4 involves the given recipient blade's BIOS requesting its BIOS iSCSI initiator to retrieve the Master Boot Record (MBR) for the operating system. The MBR contains the very initial instruction sequence of the operating system including what other disk blocks to read into the blade. As these additional disk blocks are read in, the BIOS can construct the image of the kernel loader. Once assembled, the kernel loader starts execution and the BIOS no longer participate in the system boot up though the BIOS level iSCSI driver will continue until step 5. Note that BIOS level iSCSI driver passes the configuration parameters it used to start the boot process up to the OS level iSCSI driver to continue the boot process.

Step 5 involves the given recipient blade's OS loader to pull in network services and iSCSI services. With theses services now in memory, the OS loader can start the TCP/IP network service, then, subsequently, start the iSCSI service for the "/sda" drive of the OS.

Step 6 involves the given recipient blade's OS to start other applications and finish the boot up sequence. Loading any subsequent applications and finishing the boot up places the OS in the state to start processing tasks spawned by users or applications.

A detailed discussion on defining the solution is provided in section 5.8 Sample Deployment Diskless Blade Configuration.

5.0 Deployment Steps in Detail

5.1 Sample Deployment Overview

In this sample deployment, the objective is to deploy iSCSI to 4 diskless blades that use the DS300 as iSCSI target presenting the boot drive. In order to support this deployment, a DHCP server and a blade with disks, acting as the producer blade, is used to complete the solution. Note the producer blade may be one of the 4 recipient blades deployed, but for clarity, it is assumed to be a separate blade. The DHCP service is provided by a typical DHCP server found in Linux Server environments. The IBM DS300 is used here, but other iSCSI targets can be used just as easily. The last component is a console that can access the various devices for configuration

5.2 Sample Deployment Solution Definition

Recall that the first step is to define the overall iSCSI solution desired. Here, 4 processor blades, an IBM DS300, a producer blade, a DHCP service, and a console need to be identified. All the devices are interconnected by Ethernet fabrics.

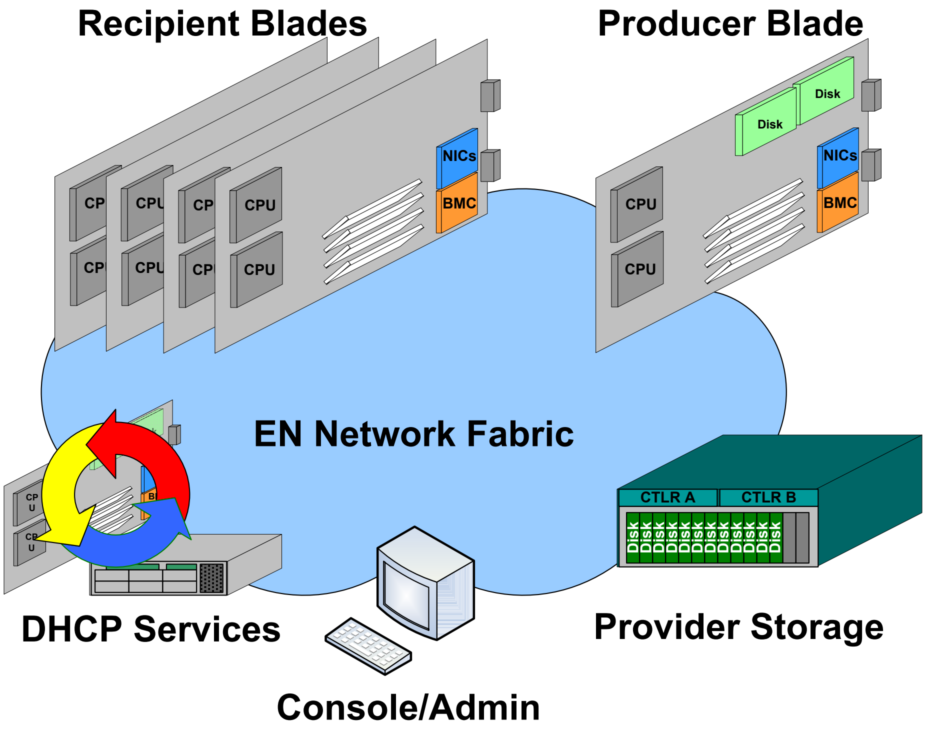


Figure 9: Example iSCSI Solution

5.2.1 Configuration Conventions

Once all the components have been identified, several parameter ranges must be defined for usage in the specific configurations.

Parameter Sets	Definition
Subnet of the local link	The subnet address and subnet mask of interest for this solution
IP address range	IP address range for all the components in the solution
Initiator name format	The naming format of the initiators
Target name format	The naming format for the targets

Table 5: iSCSI Configuration Parameter Ranges to Define

With these general parameter sets defined, the individual components can be defined. Namely, using these parameter sets as pools of available parameters, the specific configuration of the solution can be defined.

5.2.2 iSCSI Initiator Naming Definition

The iSCSI initiator name is used by the initiator to identify itself to the target.

5.2.2.1 iSCSI Initiator Name When Using Base DHCP Options

When using the standard base DHCP options, the initiator name is not explicitly defined but, instead, is built up from several parameters.

Specifically, the format of the iSCSI initiator name is the following:

iSCSI Initiator Name = iqn.1986-03.com.ibm.<MAC address>.<Option 12>

Where the MAC address is the Ethernet MAC address of the currently used port and the host name is used and provided by the DHCP server. So, for example, if the MAC address is equal to 11:11:11:11:11:11 and the host name is, defined in the DHCP config file, equal to "blade_a" then the iSCSI initiator name would be the following:

Iqn.1986-03.com.ibm.11:11:11:11:11:11.blade_a

Note in the DHCP file, either option 12 or "use-host-decl-names" command directs the DHCP server to provide the host name to the DHCP client (given blade).

5.2.2.2 iSCSI Initiator Name When Using Vendor DHCP Options

When using the standard vendor DHCP options, the iSCSI initiator name is explicitly defined, providing a layer of virtualization or abstraction from the actual hardware.

Specifically, the format of the iSCSI initiator name is the following:

iSCSI Initiator Name = <iqn compliant arbitrary string>

For example, since not needing the MAC address or option 12 (host name), the iSCSI name equivalent to the given blade above could be:

Iqn.2005-07.com.ibm.myisci.blade_a

Note that this string is presented to the given blade in vendor option 203.

5.2.2.3 iSCSI Initiator Name When Using Out of Band Configuration

When using the out of band configuration approach, the iSCSI initiator name is explicitly defined, providing a layer of virtualization or abstraction from the actual hardware.

Specifically, the format of the iSCSI initiator name is the following:

iSCSI Initiator Name = <iqn compliant arbitrary string>

For example, the iSCSI name equivalent to the given blade above could be:

Iqn.2005-07.com.ibm.myisci.blade_a

Note that this string is presented to the given blade using the iSCSI Configuration Manager.

5.2.3 Sample Deployment Configuration Parameters

For each recipient blade, the IP and iSCSI settings need to be defined. Below are the definitions for this sample deployment.

Recipient Blade 001			
Parameter	**Example Value**	**Mandatory**	**Definition**
Host Name	Blade_a	Yes	Host name assigned to given blade by DHCP
Host MAC Address	11:11:11:11:11:11	Yes	EN MAC address of the port on iSCSI initiator
Initiator IP address	123.123.123.001	Yes	iSCSI initiator IP address
Initiator name	Iqn.1986-03..com.ibm. 11:11:11:11:11:11.blade_a	Yes	Initiator when using std DHCP options
Initiator name	Iqn.2005-7.com.ibm.blade_a	Yes	Initiator when using std vendor DHCP options
Initiator name	Iqn.2005-7.com.ibm.blade_a	Yes	Initiator when using out of band config
Subnet Mask	255.255.255.0	Yes	IP network subnet mask
Gateway IP address	123.123.123.050	No	IP network gateway mask
Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
Target TCP port	3260	Yes	iSCSI TCP port on target IP address
Target name	Iqn.2005-03.com.ibm.lun_a	Yes	iSCSI iqn tag
Target Boot LUN	0	Yes	Boot LUN residing on 1st Target - option

Table 6: Configuration Parameters for Recipient Blade 1

Recipient Blade 002			
Parameter	**Example Value**	**Mandatory**	**Definition**
Host Name	Blade_b	Yes	Host name assigned to given blade by DHCP
Host EN MAC Address	22:22:22:22:22:22	Yes	EN MAC address of the port on iSCSI initiator
Initiator IP address	123.123.123.002	Yes	iSCSI initiator IP address
Initiator name	Iqn.1986-03..com.ibm. 22:22:22:22:22:22.blade_b	Yes	iSCSI in tag
Initiator name	Iqn.2005-7.com.ibm.blade_b	Yes	Initiator when using std vendor DHCP options
Initiator name	Iqn.2005-7.com.ibm.blade_b	Yes	Initiator when using out of band config
Subnet Mask	255.255.255.0	Yes	IP network subnet mask
Gateway IP address	123.123.123.050	No	IP network gateway mask
Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
Target TCP port	3260	Yes	iSCSI TCP port on target IP address
Target name	Iqn.2005-03.com.ibm.lun_b	Yes	iSCSI iqn tag
Target Boot LUN	0	Yes	Boot LUN residing on 1st Target - option

Table 7: Configuration Parameters for Recipient Blade 2

Recipient Blade 003			
Parameter	**Example Value**	**Mandatory**	**Definition**
Host Name	Blade_c	Yes	Host name assigned to given blade by DHCP
Host EN MAC Address	33:33:33:33:33:33	Yes	EN MAC address of the port on iSCSI initiator
Initiator IP address	123.123.123.003	Yes	iSCSI initiator IP address
Initiator name	Iqn.1986-03..com.ibm. 33:33:33:33:33:33.blade_c	Yes	iSCSI in tag
Initiator name	Iqn.2005-7.com.ibm.blade_c	Yes	Initiator when using std vendor DHCP options
Initiator name	Iqn.2005-7.com.ibm.blade_c	Yes	Initiator when using out of band config
Subnet Mask	255.255.255.0	Yes	IP network subnet mask
Gateway IP address	123.123.123.050	No	IP network gateway mask
Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
Target TCP port	3260	Yes	iSCSI TCP port on target IP address
Target name	Iqn.2005-03.com.ibm.lun_c	Yes	iSCSI iqn tag
Target Boot LUN	0	Yes	Boot LUN residing on 1st Target - option

Table 8: Configuration Parameters for Recipient Blade 3

Recipient Blade 004			
Parameter	**Example Value**	**Mandatory**	**Definition**
Host Name	Blade_d	Yes	Host name assigned to given blade by DHCP
Host EN MAC Address	44:44:44:44:44:44	Yes	EN MAC address of the port on iSCSI initiator
Initiator IP address	123.123.123.004	Yes	iSCSI initiator IP address
Initiator name	Iqn.1986-03..com.ibm. 44:44:44:44:44:44.blade_d	Yes	iSCSI in tag
Initiator name	Iqn.2005-7.com.ibm.blade_d	Yes	Initiator when using std vendor DHCP options
Initiator name	Iqn.2005-7.com.ibm.blade_d	Yes	Initiator when using out of band config
Subnet Mask	255.255.255.0	Yes	IP network subnet mask
Gateway IP address	123.123.123.050	No	IP network gateway mask
Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
Target TCP port	3260	Yes	iSCSI TCP port on target IP address
Target name	Iqn.2005-03.com.ibm.lun_d	Yes	iSCSI iqn tag
Target Boot LUN	0	Yes	Boot LUN residing on 1st Target - option

Table 9: Configuration Parameters for Recipient Blade 4

Producer Blade			
Parameter	**Example Value**	**Mandatory**	**Definition**
Host Name	donor	Yes	Host name assigned to given blade by DHCP
Host EN MAC Address	55:55:55:55:55:55	Yes	EN MAC address of the port on producer
Initiator IP address	123.123.123.005	Yes	Producer iSCSI initiator IP address
Initiator name	Iqn.1986-03..com.ibm. 55:55:55:55:55:55.donor	Yes	iSCSI in tag
Initiator name	Iqn.2005-7.com.ibm.donor	Yes	Initiator when using std vendor DHCP options
Initiator name	Iqn.2005-7.com.ibm.donor	Yes	Initiator when using out of band config
Subnet Mask	255.255.255.0	Yes	IP network subnet mask
Gateway IP address	123.123.123.050	No	IP network gateway mask
Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
Target TCP port	3260	Yes	iSCSI TCP port on target IP address
Target name	Iqn.2005-03.com.ibm.lun_a Iqn.2005-03.com.ibm.lun_b Iqn.2005-03.com.ibm.lun_c Iqn.2005-03.com.ibm.lun_d	Yes	iSCSI iqn tag for LUNs the producer blade should access
Target Boot LUN	0 0 0 0	Yes	Boot LUNs for each LUN accessed by producer

Table 10: Configuration Parameters for Producer Blade

Storage Controller			
Parameter	**Example Value**	**Mandatory**	**Definition**
Ctlr A port 1 IP address	123.123.123.011	Yes	iSCSI target IP address
Ctlr A port 2 IP address	123.123.123.012	Yes	iSCSI target IP address
Ctlr B port 1 IP address	123.123.123.021	Yes	iSCSI target IP address
Ctlr B port 2 IP address	123.123.123.022	Yes	iSCSI target IP address
Ctlr A mgmt IP address	123.123.123.019	Yes	Mgmt IP address for configuration
Ctlr B mgmt IP address	123.123.123.029	Yes	Mgmt IP address for configuration
Subnet Mask	255.255.255.0	Yes	IP network subnet mask
Gateway IP address	123.123.123.050	No	IP network gateway mask
1st Target name	Iqn.2005-03.com.ibm.lun_a	Yes	iSCSI iqn tag
1st Target owning host 1	Iqn.1986-03.com.ibm. 11:11:11:11:11:11.blade_a	Yes	iSCSI initiator name of recipient blade (note naming depends on config method)
1st Target owning host 2	Iqn.1986-03.com.ibm. 55:55:55:55:55:55.donor	Yes	iSCSI initiator name of donor blade (note naming depends on config method)
1st Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
1st Target TCP port	3260	Yes	iSCSI TCP port on target IP address
1st Target LUN size	10GB	Yes	Size of the LUN
1st Target LUN config	RAID 5, Write thru	Yes	LUN configuration – note target specific
1st Target specific cfg	As appropriate	Yes	Any storage vendor unique configuration
2nd Target name	Iqn.2005-03.com.ibm.lun_b	Yes	iSCSI iqn tag
2nd Target owning host 1	Iqn.1986-03.com.ibm. 22:22:22:22:22:22.blade_b	Yes	iSCSI initiator name of recipient blade (note naming depends on config method)
2nd Target owning host 2	Iqn.1986-03.com.ibm. 55:55:55:55:55:55.donor	Yes	iSCSI initiator name of donor blade (note naming depends on config method)
2nd Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
2nd Target TCP port	3260	Yes	iSCSI TCP port on target IP address
2nd Target LUN size	10GB	Yes	Size of the LUN
2nd Target LUN config	RAID 5, Write thru	Yes	LUN configuration – note target specific
2nd Target specific cfg	As appropriate	Yes	Any storage vendor unique configuration
3rd Target name	Iqn.2005-03.com.ibm.lun_c	Yes	iSCSI iqn tag
3rd Target owning host 1	Iqn.1986-03.com.ibm. 33:33:33:33:33:33.blade_c	Yes	iSCSI initiator name of recipient blade (note naming depends on config method)
3rd Target owning host 2	Iqn.1986-03.com.ibm. 55:55:55:55:55:55.donor	Yes	iSCSI initiator name of donor blade (note naming depends on config method)
3rd Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage
3rd Target TCP port	3260	Yes	iSCSI TCP port on target IP address
3rd Target LUN size	10GB	Yes	Size of the LUN
3rd Target LUN config	RAID 5, Write thru	Yes	LUN configuration – note target specific
3rdTarget specific cfg	As appropriate	Yes	Any storage vendor unique configuration
4th Target name	Iqn.2005-03.com.ibm.lun_d	Yes	iSCSI iqn tag
4th Target owning host 1	Iqn.1986-03.com.ibm. 44:44:44:44:44:44.blade_d	Yes	iSCSI initiator name of recipient blade (note naming depends on config method)
4th Target owning host 2	Iqn.1986-03.com.ibm. 55:55:55:55:55:55.donor	Yes	iSCSI initiator name of donor blade (note naming depends on config method)
4th Target IP address	123.123.123.011	Yes	iSCSI target IP address of storage

Storage Controller			
Parameter	**Example Value**	**Mandatory**	**Definition**
4th Target TCP port	3260	Yes	iSCSI TCP port on target IP address
4th Target LUN size	10GB	Yes	Size of the LUN
4th Target LUN config	RAID 5, Write thru	Yes	LUN configuration – note target specific
4th Target specific cfg	As appropriate	Yes	Any storage vendor unique configuration

Table 11: Configuration Parameters for Storage Controller

Other Network Components			
Parameter	**Example Value**	**Mandatory**	**Definition**
DHCP Server	123.123.123.51	Yes	Address of DHCP server boot config options
Gateway IP address	123.123.123.50	No	Address for gateway/router used to leave subnet
1st DNS server	123.123.123.31	Yes	Address of the DNS server if required
2nd DNS server	123.123.123.31	Yes	Address of the DNS server if required
Other desired server	123.123.123.xxx	Yes	Address of some server not related to iSCSI
Other desired server	123.123.123.yyy	Yes	Address of some server not related to iSCSI

Figure 10:Configuration Parameters of Other Network Services

A graphical view of the iSCSI configuration parameters are presented in Figure 11: iSCSI Example Solution Enumerated

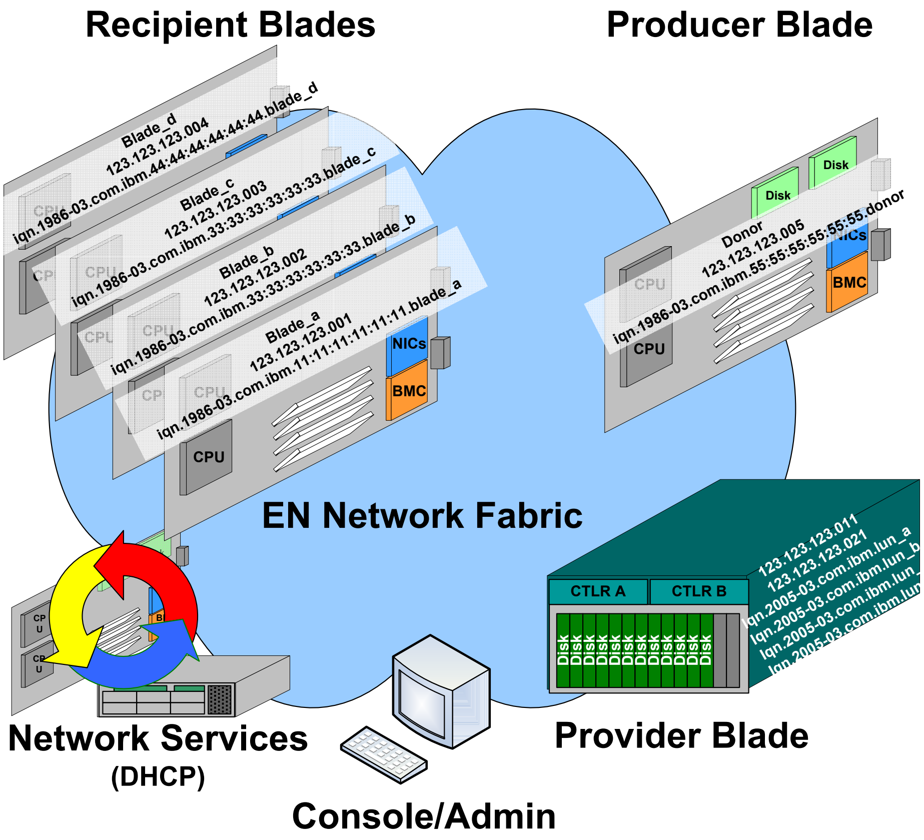


Figure 11: iSCSI Example Solution Enumerated

5.3 Sample Deployment Storage Configuration

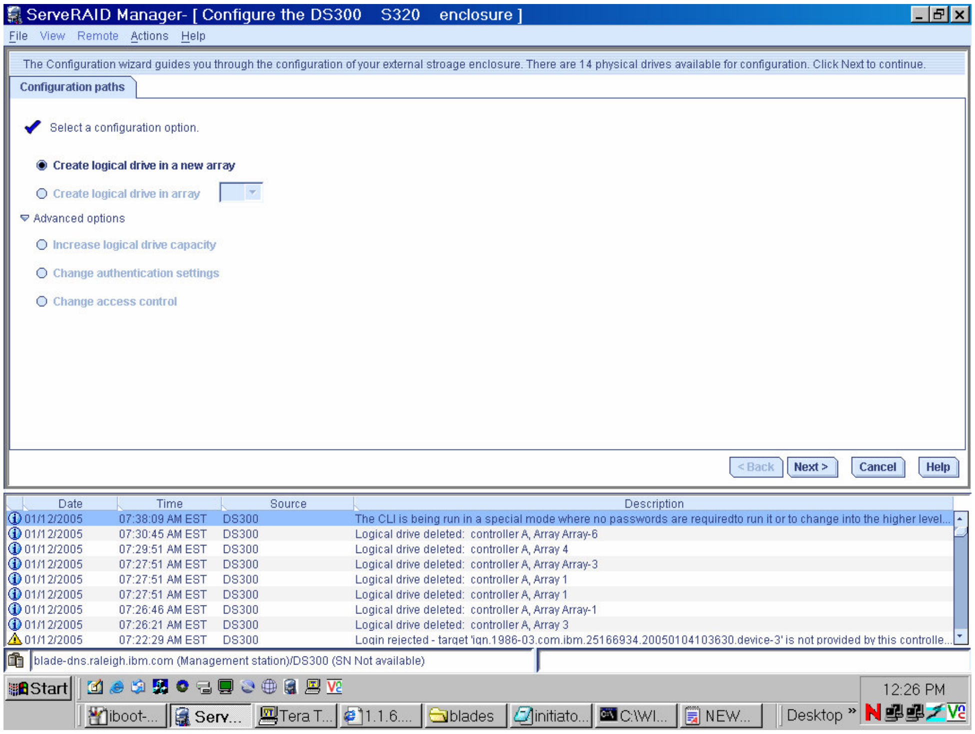
With the solution defined, the storage can be configured. For each target LUN, use the solution definition parameters to configure the LUN settings. Namely, the LUN size, the RAID level and cache configuration, and any storage vendor specific parameters such as access control lists and such.

There are a variety of iSCSI targets available in the market. Each iSCSI target has specific methods for configuring and monitors the presented storage. This appendix presents the details in configuring an IBM DS300 iSCSI target. Other iSCSI targets are configured with their respective steps and the iSCSI solution creator should refer the iSCSI target vendor's information on configuring the storage.

The IBM DS300 can be configured via Serve RAID manager or via a CLI interface. The Serve RAID manager provides a GUI based approach to configuration while the CLI provides a scriptable API interface for advanced users. While the method specifics are unique to each configuration approach, the tasks involved are the same. As a result, the Serve RAID manager approach is presented because it is more intuitive and insightful. Note, refer to the DS300 and Serve RAID document to gain better understanding the specific tasks such as administration access to the DS300.

5.3.1 DS300 Example Configuration Panels

As an example of configuring a DS300 to provide the iSCSI boot disks, the several panels cited below show the actual DS300 configuration screens



The Configuration wizard guides you through the configuration of your external strogage enclosure. There are 14 physical drives available for configuration. Click Next to continue.

✓ Select a configuration option.

● Create logical drive in a new array

○ Create logical drive in array

Advanced options

○ Increase logical drive capacity

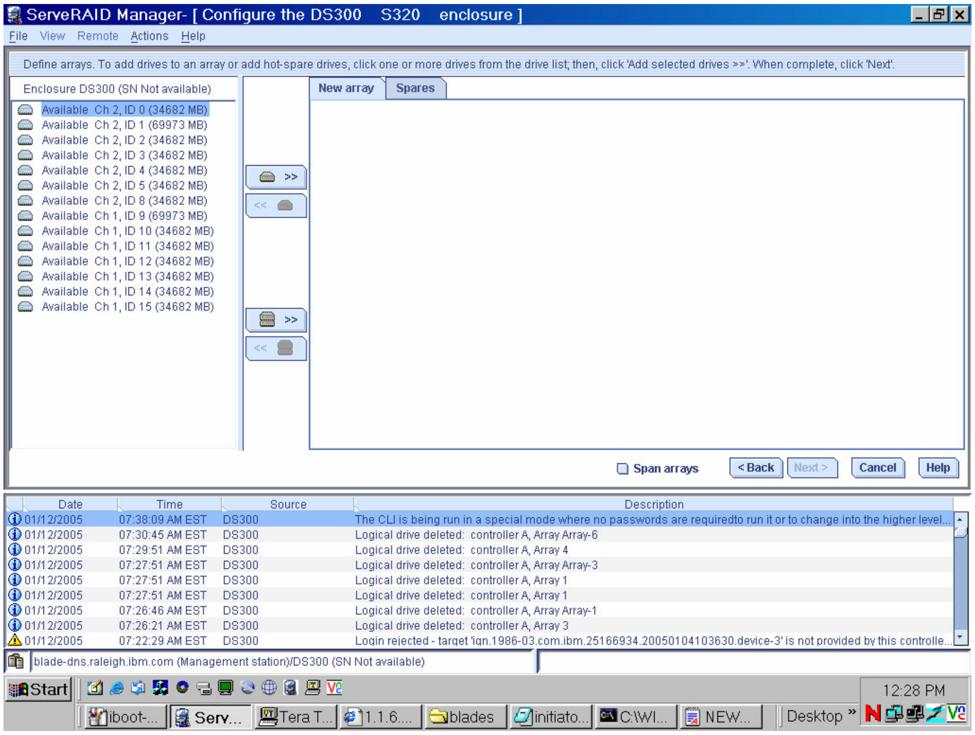
○ Change authentication settings

○ Change access control

```
Date        Time             Source   Description
01/12/2005  07:38:09 AM EST  DS300    The CLI is being run in a special mode where no passwords are requiredto run it or to change into the higher level...
01/12/2005  07:30:45 AM EST  DS300    Logical drive deleted:  controller A, Array Array-6
01/12/2005  07:29:51 AM EST  DS300    Logical drive deleted:  controller A, Array 4
01/12/2005  07:27:51 AM EST  DS300    Logical drive deleted:  controller A, Array Array-3
01/12/2005  07:27:51 AM EST  DS300    Logical drive deleted:  controller A, Array 1
01/12/2005  07:27:51 AM EST  DS300    Logical drive deleted:  controller A, Array 1
01/12/2005  07:26:46 AM EST  DS300    Logical drive deleted:  controller A, Array Array-1
01/12/2005  07:26:21 AM EST  DS300    Logical drive deleted:  controller A, Array 3
01/12/2005  07:22:29 AM EST  DS300    Login rejected - target 'iqn.1986-03.com.ibm.25166934.20050104103630.device-3' is not provided by this controlle...
```

blade-dns.raleigh.ibm.com (Management station)/DS300 (SN Not available)

Figure 12: Creating a Physical Drive on the DS300



Define arrays. To add drives to an array or add hot-spare drives, click one or more drives from the drive list; then, click 'Add selected drives >>'. When complete, click 'Next'.

Enclosure DS300 (SN Not available)

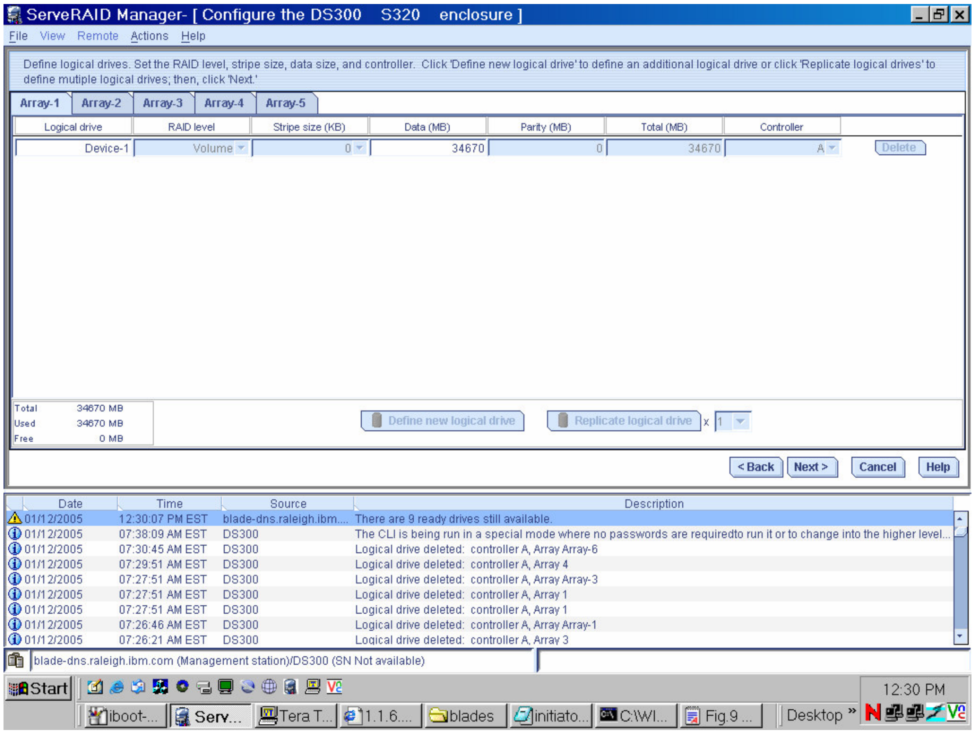
- Available Ch 2, ID 0 (34682 MB)
- Available Ch 2, ID 1 (69973 MB)
- Available Ch 2, ID 2 (34682 MB)
- Available Ch 2, ID 3 (34682 MB)
- Available Ch 2, ID 4 (34682 MB)
- Available Ch 2, ID 5 (34682 MB)
- Available Ch 2, ID 8 (34682 MB)
- Available Ch 1, ID 9 (69973 MB)
- Available Ch 1, ID 10 (34682 MB)
- Available Ch 1, ID 11 (34682 MB)
- Available Ch 1, ID 12 (34682 MB)
- Available Ch 1, ID 13 (34682 MB)
- Available Ch 1, ID 14 (34682 MB)
- Available Ch 1, ID 15 (34682 MB)

□ Span arrays

```
Date        Time             Source   Description
01/12/2005  07:38:09 AM EST  DS300    The CLI is being run in a special mode where no passwords are requiredto run it or to change into the higher level...
01/12/2005  07:30:45 AM EST  DS300    Logical drive deleted:  controller A, Array Array-6
01/12/2005  07:29:51 AM EST  DS300    Logical drive deleted:  controller A, Array 4
01/12/2005  07:27:51 AM EST  DS300    Logical drive deleted:  controller A, Array Array-3
01/12/2005  07:27:51 AM EST  DS300    Logical drive deleted:  controller A, Array 1
01/12/2005  07:27:51 AM EST  DS300    Logical drive deleted:  controller A, Array 1
01/12/2005  07:26:46 AM EST  DS300    Logical drive deleted:  controller A, Array Array-1
01/12/2005  07:26:21 AM EST  DS300    Logical drive deleted:  controller A, Array 3
01/12/2005  07:22:29 AM EST  DS300    Login rejected - target 'iqn.1986-03.com.ibm.25166934.20050104103630.device-3' is not provided by this controlle...
```

blade-dns.raleigh.ibm.com (Management station)/DS300 (SN Not available)

Figure 13: Creating a Logical Drive on a DS300


ServeRAID Manager- [Configure the DS300 S320 enclosure]
File View Remote Actions Help

Define logical drives. Set the RAID level, stripe size, data size, and controller. Click 'Define new logical drive' to define an additional logical drive or click 'Replicate logical drives' to define mutiple logical drives; then, click 'Next.'

Array-1 Array-2 Array-3 Array-4 Array-5

Logical drive	RAID level	Stripe size (KB)	Data (MB)	Parity (MB)	Total (MB)	Controller
Device-1	Volume	0	34670	0	34670	A

Total	34670 MB
Used	34670 MB
Free	0 MB

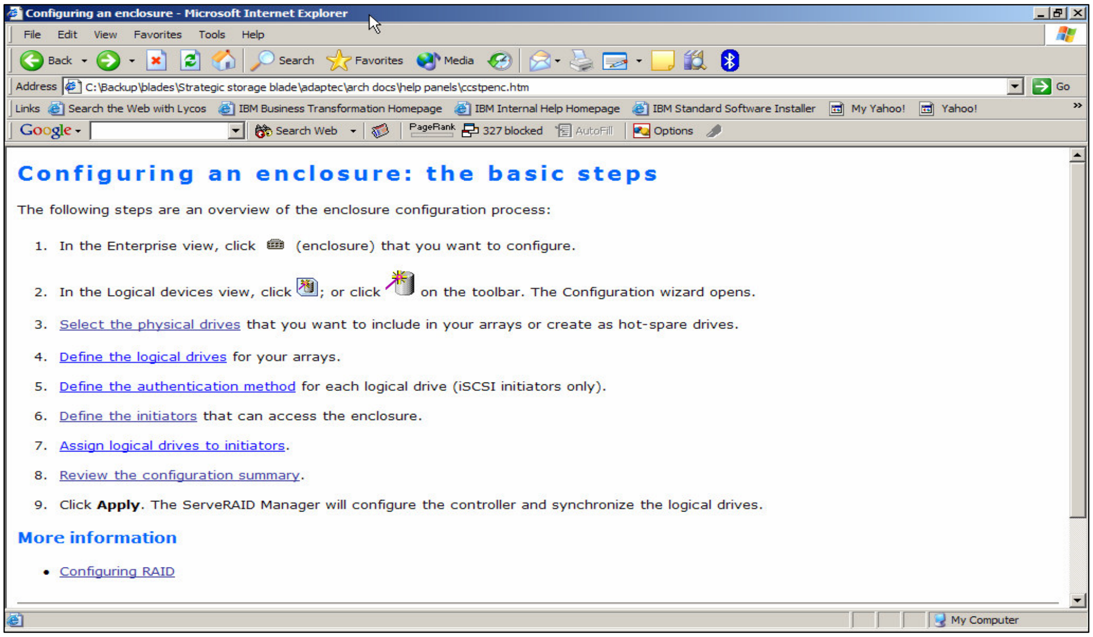
```
Date        Time             Source                Description
01/12/2005  12:30:07 PM EST  blade-dns.raleigh.ibm....  There are 9 ready drives still available.
01/12/2005  07:38:09 AM EST  DS300                 The CLI is being run in a special mode where no passwords are requiredto run it or to change into the higher level...
01/12/2005  07:30:45 AM EST  DS300                 Logical drive deleted:  controller A, Array Array-6
01/12/2005  07:29:51 AM EST  DS300                 Logical drive deleted:  controller A, Array 4
01/12/2005  07:27:51 AM EST  DS300                 Logical drive deleted:  controller A, Array Array-3
01/12/2005  07:27:51 AM EST  DS300                 Logical drive deleted:  controller A, Array 1
01/12/2005  07:27:51 AM EST  DS300                 Logical drive deleted:  controller A, Array 1
01/12/2005  07:26:46 AM EST  DS300                 Logical drive deleted:  controller A, Array Array-1
01/12/2005  07:26:21 AM EST  DS300                 Logical drive deleted:  controller A, Array 3
```

blade-dns.raleigh.ibm.com (Management station)/DS300 (SN Not available)

Figure 14: Defining a Logical Drive on a DS300

5.3.2 DS300 Configuration via Serve RAID Manager Help Panels

The on-line help panels for Serve RAID manager provide an overview of the configuration flow. In Figure 15: Serve RAID Configuration Help Panel, the introduction to the basic steps required to configure a DS300 is presented. This panel outlines the steps necessary to get the storage created and presented to the iSCSI network.


Configuring an enclosure - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Backup\blades\Strategic storage blade\adaptec\arch docs\help panels\ccstpenc.htm
Go
Links Search the Web with Lycos IBM Business Transformation Homepage IBM Internal Help Homepage IBM Standard Software Installer My Yahoo! Yahoo!
Google Search Web PageRank 327 blocked AutoFill Options
My Computer

Configuring an enclosure: the basic steps

The following steps are an overview of the enclosure configuration process:

1. In the Enterprise view, click (enclosure) that you want to configure.
2. In the Logical devices view, click ; or click on the toolbar. The Configuration wizard opens.
3. Select the physical drives that you want to include in your arrays or create as hot-spare drives.
4. Define the logical drives for your arrays.
5. Define the authentication method for each logical drive (iSCSI initiators only).
6. Define the initiators that can access the enclosure.
7. Assign logical drives to initiators.
8. Review the configuration summary.
9. Click **Apply**. The ServeRAID Manager will configure the controller and synchronize the logical drives.

More information

- Configuring RAID

Figure 15: Serve RAID Configuration Help Panel

5.4 Sample Deployment Disk Image Creation and Deployment

The disk images are created and deployed in the following manner

5.4.1 SuSE SLES 9 Image Creation and Deployment

You will need:

- SuSE SLES 9 CDs 1, 2, 3, and 4
- SuSE SLES 9 SP2 CDs 1, 2, and 3

The intention is to provide iSCSI Network Boot support in the installer as soon as possible. Until then we outline this manual procedure to allow you to experiment with Linux iSCSI Network Boot.

5.4.1.1 Install the OS

1. Boot SuSE SLES 9 SP2 CD1
2. Select Installation
3. When requested, insert SLES 9 CD 1
4. On License Agreement, enter “I agree”
5. Language: <select your language>
6. Installation Settings:
   a. New installation
   b. Software
      i. Full installation
      ii. Detailed selection…
      iii. Filter -> Search -> tftp
         1. Select tftp

            2. Accept
        iv. Automatic Changes -> Continue
    c. Accept -> Yes, install

7. Formatting and Installation proceeds through the various CDs as prompted…
8. The system reboots
9. Password for root: <enter a password>
10. Network Configuration: Next
11. Test Internet Connection: <No>
12. Service Configuration: Skip Configuration
13. User Authentication Method: <Local>
14. Users: <add a user>
15. Release Notes…, Next
16. Hardware Configuration: Use Following Configuration, Next
17. Congratulations!, Finish
18. login as root
19. "N" -> Control Center
    a. YaST2 modules
    b. Network Services
    c. Network Services (initrd)
    d. Enable
    e. telnet=on
    f. Finish
    g. Quit Control Center, File -> Quit
20. Unassign the Media Tray from the Blade
21. Reboot

5.4.1.2 iSCSI Boot Configuration – One Time Only

1. Remove the iSCSI daemon from the boot script (note: the minus minus):
    a. chkconfig --del iscsi

5.4.1.3 iSCSI Boot Configuration - For Each Replication

1. cd /
2. Add your iSCSI Target information to the configuration file.

   edit /etc/iscsi.conf and add the following:

   DiscoveryAddress=<iSCSI Target IP Address>

3. Make sure your iSCSI Target is configured to accept a connection from this system. The Initiator iqn name is in "/etc/initiatorname.iscsi".

   To maintain simplicity and ensure proper drive mapping provide only one LUN to each diskless Blade.

4. Load the iSCSI module:
    a. modprobe iscsi
5. establish an iSCSI session
    a. /sbin/iscsid

6. To verify the established session:

   ```
   iscsi-ls <-c>
   ls -la /dev/disk/by-id
   ```

   Make note of the iSCSI disk /dev/sd<?> letter. It should be “a” if you reassigned the Media Tray per a previous step.

7. Partition the iSCSI disk. In the typical install, there are two disk partitions. There is one for swap and one for the file system.

   ```
   fdisk /dev/sd<?>
   ```

   Create a clean disk: o (lower case letter oh)
   Create the swap partition: n, p, 1, <cr>, +<RAM Size MB * 2>M
   Create the root partition: n, p, 2, <cr>, <cr>
   Write out the changes and exit: w

8. Prepare the swap partition
   a. mkswap /dev/sd<?>1
9. Select a unique label for the partition and format the file system
   a. mkfs -t reiserfs /dev/sd<?>2 -label<unique partition label>
10. Create an iSCSI Boot enabled initrd file. Look in /boot for the full vmlinuz filename, however the name “vmlinuz” should link to the file you want. Note the “tg3” below is referencing a specific network driver for the blade model in use.
    a. If dynamic IP addresses via DHCP is used, use the following:
       i. /sbin/mkinitrd –k vmlinuz –I initrd-<unique name> -m “iscsi ipv6 tg3 reiserfs” –D eth0
       ii. put vmlinuz-…
    b. If static IP addresses via fixed config is used, use the following:
       i. /sbin/mkinitrd –k vmlinuz –I initrd-<unique name> -m “iscsi ipv6 tg3 reiserfs” –I eth0
       ii. Edit linuxrc and add “sleep 5” after the IPconfig command
11. Mount the iSCSI disk
    a. mount /dev/sd<?>2 /mnt
12. cd /
13. Replicate Linux to the iSCSI drive (note the minus minus):
    a. tar --exclude=/mnt -cf - / | (cd /mnt ; tar -xvpf -)
14. Create the mnt folder excluded previously
    a. cd /mnt
    b. mkdir mnt
15. edit /mnt/etc/sysconfig/network/ifcfg-eth-<mac address> and change:

    ```
    STARTMODE=’off’
    # UNIQUE=…
    ```

16. edit /mnt/etc/fstab
    a. Make sure your root is LABEL = <unique partition label> (the partition label entered in step 9 above) and swap is /dev/sda<?>
17. Inhibit the network shutdown script
    a. cd /mnt/etc/rc.d

   b. mv network network.org

18. Fix the InitiatorName
    a. cd /mnt/etc
    b. edit initiatorname.iscsi and change the name to a new unique value
19. Set the initrd symbolic link. This is the same file from step 10 for each replication.
    a. cd /mnt/boot
    b. rm initrd
    c. ln –s initrd-sles9sp1 initrd
20. If you plan to use a BIOS extension for iSCSI Network Boot instead of pxelinux see the GRUB section.
21. cd /
22. umount /mnt
23. cd /boot
24. copy initrd-<unique name> and vmlinuz-<ver> to your tftp server
    a. ls –la
    b. tftp <TFTP Server IP Address>
        i. binary
        ii. put vmlinuz-…
        iii. put initrd-…
        iv. quit
25. Edit the pxelinux config file on your TFTP server
26. Make sure your iSCSI Target is configured to accept a connection from the system booting the replicated image. The Initiator iqn is the one you entered in the pxelinux configuration file.
27. Shutdown
28. remove local disk
29. reboot via iSCSI

5.4.1.4 SuSE GRUB

If you have an iSCSI Boot BIOS extension module then you can use it rather than pxelinux to boot. It’s not required, or necessarily preferred, but it’s possible. Remember to alter your DHCP “filename” parameter appropriately as well.

5.4.1.4.1 Installing GRUB on the target partition:

1.cd /mnt/boot/grub

2. cp device.map device.map.new
3. Edit /mnt/boot/grub/device.map.new:
   (hd0) /dev/sd<?>
4. run: /usr/sbin/grub –device-map=/mnt/boot/grub/device.map.new
5. The grub command prompt should appear. Type:
   root (hd0,1)
   setup (hd0)
   quit
6. Update the grub configuration file: (See below for examples of static and DHCP usage)
   a. Edit /mnt/boot/grub/menu.lst
   b. Set the ‘root’ parameter to the root partition label (the label selected in 4.5.1.3.9):
      root=LABEL=<unique partition label>
   c. Add the Initiator IP address and subnet if addressing is static otherwise omit

d. Add the InitiatorName and DiscoveryAddress parameters appropriate for your setup.
e. Add the InitiatorName=<initiator name>
f. Add the DiscoveryAddress=<target discovery address> .
g. Change the initrd filename to match the name you used previously.

Here is an example of the /boot/grub/menu.lst contents for the case where DHCP is providing the initiator IP address (note the various line wraps in the text below):

```
# Modified by YaST2. Last modification on Thu Feb 24 08:15:25 2005

color white/blue black/light-gray
default 0
timeout 8
gfxmenu (hd0,1)/boot/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title Linux
    kernel (hd0,1)/boot/vmlinuz root=/dev/sda2 vga=0x314 selinux=0
splash=silent resume=/dev/sda1 showopts elevator=cfq
InitiatorName=iqn.2003-05.com.ibm:sles9sp1 DiscoveryAddress=192.168.70.99
    initrd (hd0,1)/boot/initrd-sles9sp1

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe
    kernel (hd0,1)/boot/vmlinuz root=/dev/sda2 showopts ide=nodma apm=off
acpi=off vga=normal noresume selinux=0 barrier=off nosmp noapic maxcpus=0  3
    initrd (hd0,1)/boot/initrd
```

Here is an example of the /boot/grub/menu.lst contents for the case where static IP addresses for the initiator are used (note the various line wraps in the text below):

```
# Modified by YaST2. Last modification on Thu Feb 24 08:15:25 2005

color white/blue black/light-gray
default 0
timeout 8
gfxmenu (hd0,1)/boot/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title Linux
    kernel (hd0,1)/boot/vmlinuz root=/dev/sda2 vga=0x314 selinux=0
splash=silent resume=/dev/sda1 showopts elevator=cfq
InitiatorName=iqn.2003-05.com.ibm:sles9sp1 DiscoveryAddress=192.168.70.99
ip=9.148.28.232::9.148.28.225:255.255.255.224:mickey-mouse:eth0:none
    initrd (hd0,1)/boot/initrd-iboot-static-sles9sp1

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe
    kernel (hd0,1)/boot/vmlinuz root=/dev/sda2 showopts ide=nodma apm=off
acpi=off vga=normal noresume selinux=0 barrier=off nosmp noapic maxcpus=0  3
    initrd (hd0,1)/boot/initrd
```

5.4.2 Red Hat Enterprise Server Image Creation and Deployment

iSCSI boot for Red Hat Enterprise Server is somewhat more complex and in flux due to Red Hat not including iSCSI capabilities natively in the distribution. The steps outlined in this section outline the process for creating an image for Red Hat Enterprise Server. However, it should be noted that these steps will be refined over the next several months. Of particular note is the procedure for making the initrd file since this is the procedure that deviates most significantly from the SUSE approach. The other procedures for Red Hat are similar to procedures for SUSE.

In order to prepare a new image for Linux iSCSI boot, you need to perform the following steps:

1. Create a volume on the iSCSI target big enough to contain your Linux image (outside the scope of this manual)
2. Create a "master" or a base Linux installation on regular machine (or use an existing one)
3. Create an initrd image
4. Replicate the "master" image into the iSCSI target.
5. Make the necessary adjustments on the target image.

In this example, several conventions are used to minimize confusion. The directories on the donor system (blade with disks used to create the image) are prefaced with **/mnt/iscsi/src** while any directors on the target system (iSCSI storage controller) is prefaced with **/mnt/iscsi/dst**. Moreover, the references to the device the target will be mapped to will be in the form of **<DEV_TARGET>**. Also, when device names (i.e. /dev/sdb1 ...) presented explicitly - it should be absolutely clear to the user that this names should reflect the system when it iBoots (i.e. if you iBoot diskless blade your device will probably be /dev/sda and if you iBoot a blade with a local disk the relevant device will probably be /dev/sdb). The /etc/fstab , /etc/mtab, and grub.conf should be changed accordingly.

5.4.2.1 Installing Linux

Follow the regular Red Hat Linux install procedure. Please use the following options.

- Partition the hard drive: Choose "manual partition" and create one swap partition and one ext3 partition mounted to "/" .
- Select GRUB boot loader
- Network settings – make sure eth0 is selected

5.4.2.2 Install iSCSI Utilities

- Install the iSCSI utilities rpm "iscsi-initiator-utils-4.0.3.0-2.i386.rpm", the sysfs utilities rpm "sysfsutils-1.2.0-1.i386.rpm".
- Both rpm's are located on the RHAS4 installation CD's

5.4.2.3 Verify Your "Master" Image

Boot you machine from your master image (locally). Make sure you can connect to the target (see instructions in section 5.4.2.4.1 Connect to Target).

5.4.2.4 Image Replication

5.4.2.4.1 Connect to Target

You can use the iSCSI driver documentation to find the instruction which describe how to connect to the target. Here are the main steps:

- Edit the file */etc/initiatorname.iscsi*. This file should contain a valid initiator name.
- Edit the /etc/iscsi.conf to point to your target, You can just adjust the following fields:
  DiscoveryAddress=
  TargetName=

Connect to the target by running the iSCSI daemon: *"/etc/init.d/iscsi {start/stop/restart} "*. The iSCSI volume should appear as a regular SCSI device (i.e. /dev/sda, etc.).

You can use the command "iscsi-ls –l" to identify the exact device name.

We will refer to the device the target is mapped to as DEV_TARGET (usually if you have one local SCSI disk this will be /dev/sdb)

5.4.2.4.2 Create Target Partitions and Filesystems

```
Create the partitions (assuming that the target device is /dev/sdb):
1.Run: fdisk <DEV_TARGET>
2.Delete all existing partitions (command d)
3.Create two partitions <DEV_TARGET>1 for the root and <DEV_TARGET>2 for
  swap. Use 'n' command, choose primary partition (p) and choose each
  partition size by +sizeM (size in Mb).
4.Change the type of the swap partition using the 't' command.
5.Change the bootable flag on the boot partition (DEV_TARGET). Use 'a'
  command and choose the bootable partition.
6.Write the changes and exit (Command 'w').

Create the filesystems:
1.mkfs.ext3 <DEV_TARGET>1 (you can also use mkfs –t ext3)
2.Adjust the partition label to a unique value.  In this example, the label
  is "iscsiBoot".  So the command is
       tune2fs –L iscsiBoot <DEV_TARGET>1
3.mkswap <DEV_TARGET>2
```

5.4.2.4.3 Replicate the Image

We will copy the content of the local root partition (assuming /dev/sda1) to the target root partition (assuming /dev/sdb1).

Create source and destination directories:

- mkdir –p /mnt/iscsi/src
- mkdir –p /mnt/iscsi/dst

Mount the local and target root partitions:

- mount /dev/sda1 /mnt/iscsi/src
- mount <DEV_TARGET>1 /mnt/iscsi/dst

Copy the data from the local partition to the target:

- (cd /mnt/iscsi/src; tar cfpl - .) | (cd /mnt/iscsi/dest; tar xfp -)

You may get some error messages during the copying process ignore these errors. This command makes 'tar' to your local partition and directs it to stdout. The second 'tar' command will receive this stream as input and will untar it on the destination.

5.4.2.4.4 Creating the Initrd Image

We will create a regular initrd image and then adjust it to enable iBoot.
Create the initrd with the following command (adjusting the kernel version as necessary):

```
/sbin/mkinitrd --with=md5 --with=crc32c --with=libcrc32c --with=tg3 --with=iscsi_sfnet --nocompress
initrd-2.6.9-22.ELsmp.iscsi_fixed.img 2.6.9-22.ELsmp
```

5.4.2.4.5 Adjust the Target Image

Make these changes on the target drive (/mnt/dst) – **NOT** on your local drive. In order to make these changes you need to mount the target's partitions. We will make changes to the /root and /boot partitions, we will refer to the target mounted partitions as ROOT and BOOT

1. Edit /mnt/iscsi/dst/boot/grub/grub.conf
   - Change the 'root' parameter to point the root partition LABEL (e.g. root=/dev/sdb1).
   - Verify that the initrd image points to the newly created image (/boot/initrd-2.6.9.22.ELsmp/iscsi_fixed.img)
2. Edit the following files (to prevent the network from being shutdown before syncing the disks):
   - In /mnt/iscsi/dst/etc/init.d/halt Comment out the following lines:
     ```
     runcmd $"Sending all processes the TERM signal..." /sbin/killall5 -15
     Sleep 5
     runcmd $"Sending all processes the KILL signal..." /sbin/killall5 -9
     ```
   - In /mnt/iscsi/dst/etc/init.d/network Comment out the following lines:
     ```
     if ! check_device_down $DEVICE; then
       action $"Shutting down interface $i: " ./ifdown $i boot
     fi
     ```
3. Verify that the file /ROOT/etc/fstab contain the correct labels for the partitions. The relevant lines should be like this:

   ```
   Label=iscsiBoot   / ext3 defauts 1 1
   /dev/sdb2         swap   swap   defaults      0 0
   ```

   (if a diskless blade is used, replace /dev/sdb2 with /dev/sda2)

4. In the files /mnt/iscsi/dst/etc/sysconfig/network-scripts/ifcfg-eth* Change the flag: *OnBoot=no*. This is done in order to prevent network interface from being reloaded at boot time (the network interface is loaded from the driver).
5. Change the hostname on the target by editing the HOSTNAME parameter in /mnt/iscsi/dst/etc/sysconfig/network.

5.4.2.4.6 Adjust Initrd Image

Adjust the initrd image for iSCSI.

```
1.  mkdir tmp_initrd
2.  cd tmp_initrd
3.  cat /mnt/iscsi/dst/boot/initrd-2.6.9-22.ELsmp.iscsi.img | cpio –extract
```

Under tmp_initrd you should see the extracted initrd image

```
4.  mkdir –p var/run  lib/tls  usr/lib
5.        cp /bin/ash /sbin/ifconfig bin
6.        cp   /usr/lib/libsysfs.so.1 usr/lib
```

7. *cp //lib/tls/libc.so.6 lib/tls*
8. *cp /lib/ld-linux.so.2 /lib/ld-linux.so.2 lib*
9. *cp /sbin/iscsid sbin*
10. *cp /etc/iscsi.conf /etc/initiatorname.iscsi etc*

For a dynamic IP configuration you will also need:

11. *cp /bin/bash bin/*
12. *cp /sbin/dhclient /sbin/dhclient-script sbin/*
13. *cp /lib/libtermcap.so.2 /lib/libdl.so.2 lib/*
14. *cp /sbin/route sbin/*
15. *cp /bin/hostname bin/*
16. *mkdir -p var/lib/dhcp/*

5.4.2.4.7 Edit the Initrd Init Script

For a static IP configuration configuration:

Attached below is the adjusted initrd script, the necessary changes are marked in bold (adjust the IP address as necessary).

```
#!/bin/nash
mount -t proc /proc /proc
setquiet
echo Mounted /proc filesystem
echo Mounting sysfs
mount -t sysfs none /sys
echo Creating /dev
mount -o mode=0755 -t tmpfs none /dev
mknod /dev/console c 5 1
mknod /dev/null c 1 3
mknod /dev/zero c 1 5
mkdir /dev/pts
mkdir /dev/shm
echo Starting udev
/sbin/udevstart
echo -n "/sbin/hotplug" > /proc/sys/kernel/hotplug
echo "Loading scsi_mod.ko module"
insmod /lib/scsi_mod.ko
echo "Loading sd_mod.ko module"
insmod /lib/sd_mod.ko
echo "Loading mptbase.ko module"
insmod /lib/mptbase.ko
echo "Loading mptscsih.ko module"
insmod /lib/mptscsih.ko
echo "Loading jbd.ko module"
insmod /lib/jbd.ko
echo "Loading ext3.ko module"
insmod /lib/ext3.ko
echo "Loading md5.ko module"
insmod /lib/md5.ko
echo "Loading libcrc32c.ko module"
insmod /lib/libcrc32c.ko
echo "Loading crc32c.ko module"
insmod /lib/crc32c.ko
echo "Loading tg3.ko module"
insmod /lib/tg3.ko
echo "Loading scsi_transport_iscsi.ko module"
insmod /lib/scsi_transport_iscsi.ko
echo "Loading iscsi_sfnet.ko module"
insmod /lib/iscsi_sfnet.ko
/bin/ifconfig eth0 <IP ADDRESS>
/sbin/udevstart
sleep 5
/bin/ash -c "/sbin/iscsid &"
```

```
sleep 5
/sbin/udevstart
echo Creating /dev
mkdevices /dev
echo Creating root device
mkrootdev /dev/root
umount /sys
echo Mounting root filesystem
mount -o defaults --ro -t ext3 /dev/root /sysroot
mount -t tmpfs --bind /dev /sysroot/dev
echo Switching to new root
switchroot /sysroot
umount /initrd/dev
```

For a dynamic IP configuration configuration:

Attached below is the adjusted initrd script (the file 'init' in the initrd's root directory), the necessary changes are marked in bold:

```
echo Switching to new root
switchroot /sysroot
umount /initrd/dev

#!/bin/nash

mount -t proc /proc /proc
setquiet
echo Mounted /proc filesystem
echo Mounting sysfs
mount -t sysfs none /sys
echo Creating /dev
mount -o mode=0755 -t tmpfs none /dev
mknod /dev/console c 5 1
mknod /dev/null c 1 3
mknod /dev/zero c 1 5
mkdir /dev/pts
mkdir /dev/shm
echo Starting udev
/sbin/udevstart
echo -n "/sbin/hotplug" > /proc/sys/kernel/hotplug
echo "Loading scsi_mod.ko module"
insmod /lib/scsi_mod.ko
echo "Loading sd_mod.ko module"
insmod /lib/sd_mod.ko
echo "Loading mptbase.ko module"
insmod /lib/mptbase.ko
echo "Loading mptscsih.ko module"
insmod /lib/mptscsih.ko
echo "Loading jbd.ko module"
insmod /lib/jbd.ko
echo "Loading ext3.ko module"
insmod /lib/ext3.ko
echo "Loading md5.ko module"
insmod /lib/md5.ko
echo "Loading libcrc32c.ko module"
insmod /lib/libcrc32c.ko
echo "Loading crc32c.ko module"
insmod /lib/crc32c.ko
echo "Loading tg3.ko module"
insmod /lib/tg3.ko
echo "Loading scsi_transport_iscsi.ko module"
insmod /lib/scsi_transport_iscsi.ko
echo "Loading iscsi_sfnet.ko module"
insmod /lib/iscsi_sfnet.ko
```

```
/bin/dhclient eth0
sleep 5
/sbin/udevstart
sleep 5
/bin/ash -c "/sbin/iscsid &"
sleep 5
/sbin/udevstart
echo "Creating /dev"
mkdevices /dev


echo Creating root device
mkrootdev /dev/root
umount /sys
echo Mounting root filesystem
mount -o defaults --ro -t ext3 /dev/root /sysroot
mount -t tmpfs --bind /dev /sysroot/dev
echo Switching to new root
switchroot /sysroot
umount /initrd/dev
```

You will also have to edit the file sbin/dhclient-script. Below is the modified file, with changes marked in bold:

```
#!/bin/bash
# dhclient-script for Linux. Dan Halbert, March, 1997.
# Updated for Linux 2.[12] by Brian J. Murrell, January 1999.
# No guarantees about this. I'm a novice at the details of Linux
# networking.

# Notes:

# 0. This script is based on the netbsd script supplied with dhcp-970306.

# 1. ifconfig down apparently deletes all relevant routes and flushes
# the arp cache, so this doesn't need to be done explicitly.

# 2. The alias address handling here has not been tested AT ALL.
# I'm just going by the doc of modern Linux ip aliasing, which uses
# notations like eth0:0, eth0:1, for each alias.

# 3. I have to calculate the network address, and calculate the broadcast
# address if it is not supplied. This might be much more easily done
# by the dhclient C code, and passed on.

#function save_previous() {
#  if [ -e $1 -a ! -e $1.predhclient ]; then
#    mv $1 $1.predhclient
#  fi
#}
#make_resolv_conf() {
#  if [ "${PEERDNS}" == "no" ]; then
#     return
#  fi
#
#  if [ x$reason == xRENEW ]; then
#     if [ "$new_domain_name" == "$old_domain_name" ] && [ "$new_domain_servers" ==
          "$old_domain_servers" ]; then
#          return
#     fi
#  fi
#
#  if [ -n "$new_domain_name" ] || [ -n "$new_domain_name_servers" ]; then
#    save_previous /etc/resolv.conf
#    rscf=`mktemp /tmp/XXXXXX`;
#   echo '; generated by /sbin/dhclient-script' > $rscf
#   if [ -n "$SEARCH" ]; then
#         echo search $SEARCH >> $rscf
#   else
#         if [ -n "$new_domain_name" ]; then
#             echo search $new_domain_name >> $rscf
#      fi
#   fi
#   for nameserver in $new_domain_name_servers; do
#     echo nameserver $nameserver >> $rscf
#   done
#   change_resolv_conf $rscf
#   rm -f $rscf
# fi
#}

# Must be used on exit.   Invokes the local dhcp client exit hooks, if any.
exit_with_hooks() {
  exit_status=$1
  if [ -f /etc/dhclient-exit-hooks ]; then
    . /etc/dhclient-exit-hooks
  fi
# probably should do something with exit status of the local script
  exit $exit_status
}

# Invoke the local dhcp client enter hooks, if they exist.
if [ -f /etc/dhclient-enter-hooks ]; then
  exit_status=0
  . /etc/dhclient-enter-hooks
  # allow the local script to abort processing of this state
  # local script must set exit_status variable to nonzero.
  if [ $exit_status -ne 0 ]; then
    exit $exit_status
  fi
fi

# Import Red Hat Linux configuration
#cd /etc/sysconfig/network-scripts;
#. /etc/sysconfig/network-scripts/network-functions
#. /etc/rc.d/init.d/functions

#[ -f ../network ] && . ../network
#[ -f ../networking/network ] && . ../networking/network

CONFIG=$interface

#need_config ${CONFIG}

#if [ -f "${CONFIG}" ]; then
#   source_config
#else
#   echo $"$0: configuration for $interface not found. Continuing with defaults." >&2
#fi

#source_config

#release=`uname -r`
#relmajor=`echo $release |/bin/cut -f1 -d'.'`
#relminor=`echo $release |/bin/cut -f2 -d'.'`

if [ x$new_broadcast_address != x ]; then
  new_broadcast_arg="broadcast $new_broadcast_address"
fi
if [ x$old_broadcast_address != x ]; then
  old_broadcast_arg="broadcast $old_broadcast_address"
fi
if [ x$new_subnet_mask != x ]; then
```

```
  new_subnet_arg="netmask $new_subnet_mask"
fi
if [ x$old_subnet_mask != x ]; then
  old_subnet_arg="netmask $old_subnet_mask"
fi
if [ x$alias_subnet_mask != x ]; then
  alias_subnet_arg="netmask $alias_subnet_mask"
fi

if [ x$reason = xMEDIUM ]; then
  # Linux doesn't do mediums (ok, ok, media).
  exit_with_hooks 0
fi

added_old_broadcast_route=0;
if [ x$reason = xPREINIT ]; then
  if [ x$alias_ip_address != x ]; then
    # Bring down alias interface. Its routes will disappear too.
    ifconfig $interface:0- inet 0
  fi
  if [ x$keep_old_ip == xyes ]; then
    ifconfig $interface up
#  elif [ $relmajor -lt 2 ] || ( [ $relmajor -eq 2 ] && [ $relminor -eq 0 ] )   then
#    ifconfig $interface inet 0.0.0.0 netmask 0.0.0.0 \
#                         broadcast 255.255.255.255 up
#    # Add route to make broadcast work. Do not omit netmask.
#    route add default dev $interface netmask 0.0.0.0 && added_old_broadcast_route=1;
  else
    ifconfig $interface 0 up
  fi

  # We need to give the kernel some time to get the interface up.
#  sleep 1

  exit_with_hooks 0
fi

if [ x$reason = xARPCHECK ] || [ x$reason = xARPSEND ]; then
  exit_with_hooks 0
fi

if [ x$reason = xBOUND ] || [ x$reason = xRENEW ] || \
   [ x$reason = xREBIND ] || [ x$reason = xREBOOT ]; then

#  if [ "${PEERNIS}" = no ]; then
#    :
#  elif [ -n "$new_nis_domain" ]; then
#           domainname "$new_nis_domain"
#           save_previous /etc/yp.conf
#           echo '# generated by /sbin/dhclient-script' > /etc/yp.conf
#           if [ -n "$new_nis_servers" ]; then
#              for I in $new_nis_servers; do
#                        echo "domain $new_nis_domain server $I" >> /etc/yp.conf
#              done
#           else
#              echo "domain $new_nis_domain broadcast" >> /etc/yp.conf
#           fi
#  elif [ -n "$new_nis_servers" ]; then
#       save_previous /etc/yp.conf
#       echo '# generated by /sbin/dhclient-script' > /etc/yp.conf
#       for I in $new_nis_servers; do
#             echo "ypserver $I" >> /etc/yp.conf
#       done
#  fi

#  if [ "${PEERNTP}" = no ]; then
#    :
#  elif [ -n "$new_ntp_servers" ] && [ -e /etc/ntp.conf ]; then
#     save_previous /etc/ntp.conf
#     cat <<EOF > /etc/ntp.conf
## generated by /sbin/dhclient-script
#restrict default ignore
#restrict 127.0.0.1
#driftfile /var/lib/ntp/drift
#broadcastdelay 0.008
#authenticate yes
#keys /etc/ntp/keys
#EOF
#     save_previous /etc/ntp/step-tickers
#     echo '# generated by /sbin/dhclient-script' > /etc/ntp/step-tickers
#     for I in $new_ntp_servers; do
#        echo "restrict $I nomodify notrap noquery" >> /etc/ntp.conf
#          echo "server $I" >> /etc/ntp.conf
#          echo "$I" >> /etc/ntp/step-tickers
#     done
# fi

# if [ -n "$new_host_name" ] && need_hostname; then
     hostname $new_host_name
# fi

  if [ x$old_ip_address != x ] && [ x$alias_ip_address != x ] && \
                      [ x$alias_ip_address != x$old_ip_address ]; then
   # Possible new alias. Remove old alias.
   ifconfig $interface:0- inet 0
  fi
  if [ x$old_ip_address != x ] && [ x$old_ip_address != x$new_ip_address ]; then
   # IP address changed. Bringing down the interface will delete all routes,
   # and clear the ARP cache.
   ifconfig $interface inet 0 down

  fi
  if [ x$old_ip_address = x ] || [ x$old_ip_address != x$new_ip_address ] || \
    [ x$reason = xBOUND ] || [ x$reason = xREBOOT ]; then

   ifconfig $interface inet $new_ip_address $new_subnet_arg \
                                                                  $new_broadcast_arg
   # Add a network route to the computed network address.
#    if [ $relmajor -lt 2 ] || \
#                      ( [ $relmajor -eq 2 ] && [ $relminor -eq 0 ] ); then
#     route add -net $new_network_number $new_subnet_arg dev $interface
#     if [ $added_old_broadcast_route -eq 1 ]; then
#           route del default;
#     fi;
#    fi
   for router in $new_routers; do
     route add default gw $router && break; # there can be only one !
   done
   # static routes
   if [ "x$new_static_routes" != x ]; then
     len=`echo $new_static_routes | awk '{print NF}'`
     i=1
     while [ $i -lt $len ]; do
       target=`echo $new_static_routes | awk "{print \\$$i}"`
       let i=i+1
       gateway=`echo $new_static_routes | awk "{print \\$$i}"`
       let i=i+1
       route add -host $target gw $gateway $interface
     done
   fi
  fi
  if [ x$new_ip_address != x$alias_ip_address ] && [ x$alias_ip_address != x ];
   then
   ifconfig $interface:0- inet 0
   ifconfig $interface:0 inet $alias_ip_address $alias_subnet_arg
   route add -host $alias_ip_address $interface:0
  fi
#  make_resolv_conf
  exit_with_hooks 0
fi

if [ x$reason = xEXPIRE ] || [ x$reason = xFAIL ] || [ x$reason = xRELEASE ] \
   || [ x$reason = xSTOP ]; then
  if [ x$alias_ip_address != x ]; then
    # Turn off alias interface.
    ifconfig $interface:0- inet 0
  fi
  if [ x$old_ip_address != x ]; then
    # Shut down interface, which will delete routes and clear arp cache.
    ifconfig $interface inet 0 down
  fi
  if [ x$alias_ip_address != x ]; then
    ifconfig $interface:0 inet $alias_ip_address $alias_subnet_arg
    route add -host $alias_ip_address $interface:0
  fi
  exit_with_hooks 0
fi

if [ x$reason = xTIMEOUT ] && [ "x$new_routers" != 'x' ]; then
  if [ x$alias_ip_address != x ]; then
    ifconfig $interface:0- inet 0
  fi
  ifconfig $interface inet $new_ip_address $new_subnet_arg \
                                                                   $new_broadcast_arg
  set $new_routers
  if ping -q -c 1 -w 10 $1; then
    if [ x$new_ip_address != x$alias_ip_address ] && \
                                           [ x$alias_ip_address != x ]; then
      ifconfig $interface:0 inet $alias_ip_address $alias_subnet_arg
      route add -host $alias_ip_address dev $interface:0
    fi
    if [ $relmajor -lt 2 ] || \
                        ( [ $relmajor -eq 2 ] && [ $relminor -eq 0 ] ); then
      route add -net $new_network_number
    fi
    for router in $new_routers; do
      route add default gw $router
    done
#    if [ "${PEERDNS}" != "no" ]; then
#      make_resolv_conf
#    fi
    exit_with_hooks 0
  fi
  ifconfig $interface inet 0 down
  exit_with_hooks 1
elif [ x$reason = xTIMEOUT ]; then
  exit_with_hooks 1
fi

exit_with_hooks 0
```

Rebuild and compress the initrd image

```
cd  tmp_initrd
( find . | cpio -c -o --quiet | gzip -9 ) >   /mnt/iscsi/dst/initrd-2.6.9-22.ELsmp.iscsi.img
```

5.4.2.4.8 Change the GRUB Boot Loader

We now copy the grub boot loader to the target:

1. cd /mnt/iscsi/dst/boot/grub ; cp device.map device.map.new
2. Edit device.map.new
   (fd0)   /dev/fd0
   (hd0)   <DEV_TARGET>
3. Run */sbin/grub --device-map=<device-map file you created in the previous step>. This will put you in the* grub shell using your device map file.
4. Enter the following commands in the grub shell to install grub:
   a. *"root (hd0 ,0)"*
   b. "setup (hd0)"
   c. "quit"

Unmount the partitions:

```
umount /mnt/iscsi/src
umount /mnt/iscsi/dst
```

Stop iSCSI daemon

```
service iscsi stop
```

The target image is ready for booting

5.5 Sample Deployment Network Services Configuration

The configuration of the network services involves using the solution definition parameters to set up the behaviors on DHCP. It is assumed the network services have been installed on some system that has been defined in the solution definition. DHCP network services can be already resident and used currently or it can be a new service to add to the network.

5.5.1 DHCP Background

Before discussing the sample deployment, some background information on DHCP may prove helpful. The DHCP Protocol is defined by the Internet Engineering Task Force (IETF) in Request For Comment (RFC) standards 2131, 2132, and 3396. Recall that DHCP can be used to acquire configuration information from a network resource such as DHCP server. Implied here is that the server trying to acquire such information may not have any intelligence at all about the network. Fortunately, the DHCP protocol assumes that and defines identifying the appropriate resources as part of the protocol. Also, with DHCP rich flexibility, there are many ways to customize the configuration and configuration information deployment. Given that the reader may not be fluent the DHCP capabilities, this section is intended to introduce these concepts and provide a launching board to other DHCP documentation for more detailed information.

5.5.1.1 DHCP Protocol Flow

The DHCP protocol has 4 basic phases used to acquire configuration information. These are DHCPDISCOVERY, DHCPOFFER, DHCPREQUEST, and DHCPACK. The DHCP protocol has several advanced phases, such as DHCPNACK, but that is beyond the scope of this document and, thus, left for more detailed reading.

The DHCPDISCOVER phase defines the process used by a given Blade to announce to the network that its looking for a DHCP server to acquire configuration information from. In essence, the Blade launches a broadcast frame to all stations on the network declaring itself and its desire to find a DHCP server.

The DHCPOFFER phase defines the process used by one or more DHCP servers to announce to the Blade that it can provide configuration information. In essence, the DHCP server determines it can provide the desired information and launches a broadcast frame (as a response to DHCPDISCOVER) to all stations on the network declaring it can provide the desired information. Given that more than one DHCP servers can respond, the given Blade determines the best DHCP server to use.

The DHCPREQUEST phase defines the process used by a given blade to announce that it selected a specific DHCP server and is now explicitly asking that server for the desired information. This frame is a broadcast as well to inform the other DHCP servers that they were not selected.

The DHCPACK phase defines the process used by the selected DHCP server to provide the desired information back to the given blade.

In Figure 16: DHCP Protocol Overview, simple overview is provided.

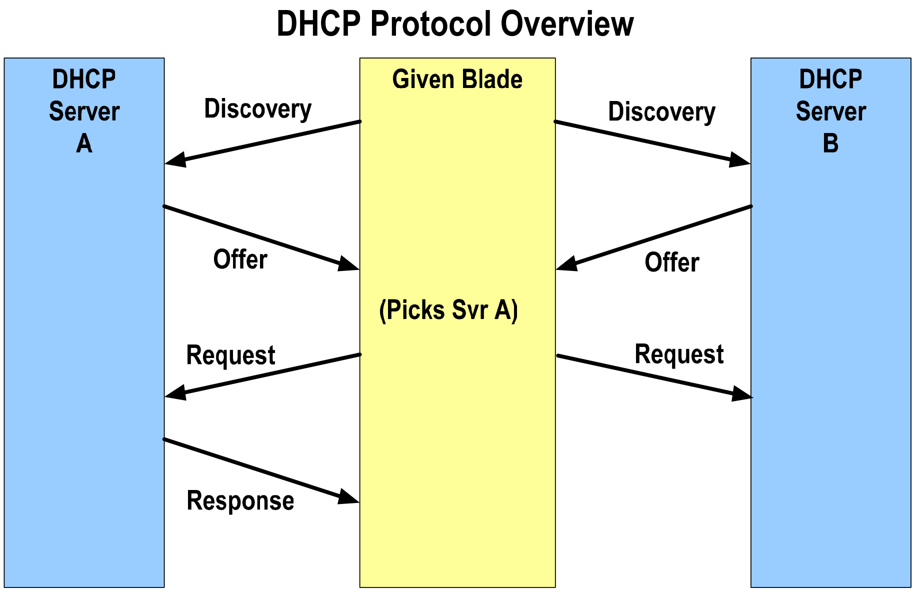


Figure 16: DHCP Protocol Overview

5.5.1.2 Standard Base and Vendor DHCP Options

As part of the DHCP RFCs, there is a capability to use DHCP options to provide specific types of information back to the given blade noting that some blades may want some types of information while others may want different types of information. Within the options construct, there are two types of options. Standard base options are part of the base RFC standard and are anticipated to be used widely throughout the industry. Standard vendor options are part of the base RFC standard as well but allows a given vendor (or user/admin) to customize the type of information (not the actual values) represented by the option. The intent of vendor options is to allow a vendor/customer/etc to tailor the types of information presented for the specific new or emerging device or for a specific configuration the vendor or customer wishes to implement.

In the context of iSCSI boot, the customer can choose to use the standard base DHCP options to configure the given blade or can choose to use the standard vendor DHCP options. The key difference in the approaches is that the standard base DHCP option approach provides a very simple, and thus limited, configuration while the standard vendor DHCP option approach provides a very rich and flexible configuration. As an example of feature differences, the standard base approach provides for a single target the given blade can access for booting while the standard vendor approach provides for 2 iscsi targets for high availability booting.

In the context of iSCSI boot, the standard base DHCP options of interest are defined Table 12: Standard Base DHCP Options in the Context of iSCSI boot. Note that the given Blades IP address is provided inherently in the protocol and, thus, does not need a specific option to provide it. Also note that there are other standard base DHCP options in the RFC standard though they are not pertinent to the iSCSI boot discussion.

DHCP Base Option	**Definition**	**Description**
1	IP Subnet Mask	Defines the subnet mask to use for determining if an IP address is on the local subnet or needs to be routed to another subnet.
3	IP Router Address	Defines the router or gateway address to be used when accessing a device on another subnet
17	Root path	Defines the DHCP method of using the root path option to define the iSCSI target to use during boot. If a solution is going to use the root path option to define a single target, then the root path definition in the DHCP server configuration file must be scoped within the vendor ID scope. This is necessary to ensure that a DHCP client using the Ethernet MAC address as the identifier does not received the root path associated with a PXE boot or some other boot methodology.
12 (host name)	Host name	Defines the DHCP method to define the initiator. If a solution going to use DHCP option 12 in conjunction with option 17, the initiator name used to login into the target is a string built up from the MAC address and the option 12 name. Note that DHCP servers can provide the host name via the "use-host-decle-names" flag

Table 12: Standard Base DHCP Options in the Context of iSCSI boot

In the context of iSCSI boot, the standard vendor DHCP options of interest are defined Table 14: Standard Vendor DHCP Options in the Context of iSCSI and Table 14: Standard Vendor DHCP Options in the Context of iSCSI. Note that the standard vendor DHCP options complement the standard base options with the exception of option 17 and option 12 because they are not needed when standard vendor DHCP options are used. As a result, the appropriate standard base options and standard vendor options are presented

DHCP Base Option	Definition	Description
1	IP Subnet Mask	Defines the subnet mask to use for determining if an IP address is on the local subnet or needs to be routed to another subnet.
3	IP Router Address	Defines the router or gateway address to be used when accessing a device on another subnet
43	Vendor Option Response	Defines the encapsulation of all options associated with this vendor identifier. The DHCP client can assume that if a 43 response is provided, then this DHCP server is iSCSI aware. The format of this option is 43:len:optA:lenA:valueA:optB:lenB:valueB:…. Given the maximum length of any option including 43 is 255 bytes, RFC 3396 defines a means of sending larger responses to the DHCP client. Basically, 3396 defines an orderly concatenation of multiple option responses.
55	DHCP Parameter List	Defines the list of options the DHCP client expects to receive from the DHCP server. As defined in RFC2132, there is ambiguity on the usage of DHCP option 55 with some servers ignoring it. To better manage the ambiguity, a DHCP client can either include option 55 with ALL desired DHCP options or no include option 55 at all. As a result, a given implementer has the choice of implementing DHCP option 55. In the context of this specification, it is assumed DHCP option 55 is neither present nor enumerated.
57	DHCP message size	Defines the message size of the DHCP messages that should be used between the DHCP server and client. The default is 576 bytes of which 504 is the true DHCP message.
60	Vendor Class Identifier	Defines the class of vendor requesting server. The DHCP server uses this option to scope the class of option responses that should be used. In the context of iSCSI solutions, this option uses the following format to aid in determine which DHCP server to use and to aid the DHCP server to determine which options to use. If SVID field is defined, the format of option 60 is SVID field (if defined):VERS field:LVL field:Initiator mfg:Initiator type:initiator level. If the SVID is not defined, the format of option 60 is “IBM ISAN”:VER field:LVL field:initiator mfg: initiator type:initiator revision. An example of the latter format would be “IBM ISAN:29:58:IBM :01:01” where the VER[7:0]=29, LVL[7:0]=58, mfg=”IBM “,type=01,rev 01. See Option 60 discussion below for more details. Note that the 29 and 58 must be translated to a string form representing the value. The vendor class identifier is analogous to identifying the seating section in a stadium
61	Client Identifier	Defines the id of this specific DHCP client. The DHCP server uses this option to determine the exact values to respond with. The default is to use the EN MAC address of the port used but in the future, option should use the CAID field if defined. The client identify is analogous to identifying the seat in the identified section in a stadium. Note when using Microsoft DHCP server, the option 61 cannot be used. Instead, the EN MAC, preceded by a byte with the value 0x01 is used to distinguish one client from another.

Table 13: Standard Base DHCP Options in the Context of iSCSI

DHCP Vendor Option	Definition	Description
201	1st Target identification	Defines IP and iSCSI parameters set for the 1st target. The format is "iscsi:<1st target>:<protocol>:<port>:<LUN>:<iscsi name>"
202	2nd Target identification	Defines IP and iSCSI parameters set for the 1st target. The format is "iscsi:<1st target>:<protocol>:<port>:<LUN>:<iscsi name>"
203	Initiator identification	Defines the iSCSI information of the initiator. Note that the IP information is defined in the DHCP message elsewhere The format is Initiator iSCSI name in IQN or EUI format
204	Discovery identification	Defines the parameters needed for additional discovery. Basically, this option allows the customer to use the current DHCP server to tell the DHCP client to find the iSCSI ONLY parameters on another DHCP server. The format is "scope:<scope>:disc:<dhcp>:id:<client alt id>:flags:<flag bits>" Note: to be backward compatible, if the last character is NOT a colon, then there are no flag bits coming and the following rules apply: Use the scope as defined (i.e. scope:<IBM ISAN or other>:) If disc = NULL (i.e. disc:255.255.255.255:), then dhcp broadcast is used If id = NULL (i.e. id:null"), then client ID is EN MAC address In the future, flag bits (not to exceed 12 presented as 3 ASCII hex chars) will be defined for advanced functions.
205	iSCSI option space	Defines a DHCP option to pass iSCSI option bits to the DHCP client. This is used only for future or advanced configuration approaches. The intent is that this response matches the OPT field defined in the iSCSI parameter data structure. The format is "B0:<LSB>:B1:<LMSB>:B2:<HMSB>:B3:<MSB>" Note that this DHCP option is optional in the DHCP exchange
206	iSCSI Link Behavior	Defines the behavior iSCSI session establishment in terms of timeouts and retries to the 1st target and to the 2nd target. This option is for future advanced configurations where the specific link behavior needs to be specified. The format is "R1:<retry1>:R2:<retry2>:T1:<timeout1>:T2:<timeout2>"

Table 14: Standard Vendor DHCP Options in the Context of iSCSI

Standard vendor DHCP Options 204,205,206 are of particular interest. Option 204 provides a way to segregate and virtualize the iSCSI environment in terms of supporting an additional iSCSI centric DHCP server, allowing the customer to uniquely identify the iSCSI solution context, and allowing the customer to uniquely define the given client blade independent of the actual blade hardware fixed configuration. Option 205 provides a way to impart additional configuration options to the given blade. Option 206, for advanced iSCSI initiators, provides link behavior configuration to aid in high availability booting.

5.5.1.3 iSCSI Initiator Naming Definition

The iSCSI initiator name is used by the initiator to identify itself to the target.

5.5.1.3.1 iSCSI Initiator Name When Using Base DHCP Options

When using the standard base DHCP options, the initiator name is not explicitly defined but, instead, is built up from several parameters.

Specifically, the format of the iSCSI initiator name is the following:

iSCSI Initiator Name = iqn.1986-03.com.ibm.<MAC address>.<Option 12>

Where the MAC address is the Ethernet MAC address of the currently used port and the host name is used and provided by the DHCP server. So, for example, if the MAC address is equal to 11:11:11:11:11:11 and the host name is, defined in the DHCP config file, equal to "blade_a" then the iSCSI initiator name would be the following:

Iqn.1986-03.com.ibm.11:11:11:11:11:11.blade_a

Note in the DHCP file, either option 12 or "use-host-decl-names" command directs the DHCP server to provide the host name to the DHCP client (given blade).

5.5.1.3.2 iSCSI Initiator Name When Using Vendor DHCP Options

When using the standard vendor DHCP options, the iSCSI initiator name is explicitly defined, providing a layer of virtualization or abstraction from the actual hardware.

Specifically, the format of the iSCSI initiator name is the following:

iSCSI Initiator Name = <iqn compliant arbitrary string>

For example, since not needing the MAC address or option 12 (host name), the iSCSI name equivalent to the given blade above could be:

Iqn.2005-07.com.ibm.myisci.blade_a

Note that this string is presented to the given blade in vendor option 203.

5.5.2 DHCP Configuration using Standard Base DHCP Options

Recall that the DHCP service is configured and controlled by a configuration file. This file specifies how the DHCP server should respond to a given recipient blade.

Below, in Figure 17: DHCP Config File using Standard Base DHCP Options Only, the DHCP config file for the current sample deployment is presented for NBP boot. There are several highlights worth noting.

First, for each recipient blade, an entry defines the specific behavior required. For additional blades, additional entries are required.

Second, The DHCP server uses the Ethernet MAC address to determine which entry to use for the given recipient blade.

Third, a given entry defines a fixed IP address for the initiator, the path to the iSCSI target LUN (enumerated in the root path entry), pointer to the PXE config directory, and a pointer to the TFTP server.

Fourth, the comments indicate how to specify the options in the context of a windows server.

```
#############
# DHCP config File
############
#DHCP 3.0
use-host-decl-names on;
allow booting;
allow bootp;

ddns-update-style ad-hoc;

subnet 123.123.123.0  netmask 255.255.255.0 {
option dhcp-lease-time -1;
option routers 123.123.123.31;
option domain-name-servers 123.123.123.32,123.123.123.33;
option domain-name "haifa.ibm.com";

    host blade_a {
    option root-path "iscsi:123.123.123.11:8:3260:0:iqn.2005-03.com.ibm.lun_a";
    hardware ethernet 11:11:11:11:11:11;
    fixed-address 123.123.123.1;
    }
    host blade_b {
    option root-path "iscsi:123.123.123.11:8:3260:0:iqn.2005-03.com.ibm.lun_b";
    hardware ethernet 22:22:22:22:22:22;
    fixed-address 123.123.123.2;
    }
    host blade_c {
    option root-path "iscsi:123.123.123.11:8:3260:0:iqn.2005-03.com.ibm.lun_c";
    hardware ethernet 33:33:33:33:33:33;
    fixed-address 123.123.123.3;
    }
    host blade_d {
    option root-path "iscsi:123.123.123.11:8:3260:0:iqn.2005-03.com.ibm.lun_d";
    hardware ethernet 44:44:44:44:44:44;
    fixed-address 123.123.123.4;
    }
```

Figure 17: DHCP Config File using Standard Base DHCP Options Only

If the DHCP service is to be provided by Microsoft Server, then please refer to the Microsoft DHCP server configuration. This service is native to Microsoft server 2000 and server 2003 operating systems.

5.5.3 DHCP Configuration using Standard Vendor DHCP Options

Recall that the DHCP service is configured and controlled by a configuration file. This file specifies how the DHCP server should respond to a given recipient blade.

Below, in Figure 18: DHCP Config File using Standard Base DHCP Options Only, the DHCP config file for the current sample deployment is presented for NBP boot. There are several highlights worth noting.

First, for each recipient blade, an entry defines the specific behavior required. For additional blades, additional entries are required.

Second, The DHCP server uses the Ethernet MAC address to determine which entry to use for the given recipient blade.

Third, a given entry defines a fixed IP address for the initiator, the path to the iSCSI target LUN (enumerated in the root path entry), pointer to the PXE config directory, and a pointer to the TFTP server.

Fourth, the comments indicate how to specify the options in the context of a windows server.

```
#############
# DHCP config File
############
#DHCP 3.0
allow booting;
allow bootp;

ddns-update-style ad-hoc;

subnet 123.123.123.0  netmask 255.255.255.0 {
option dhcp-lease-time -1;
option routers 123.123.123.31;
option domain-name-servers 123.123.123.32,123.123.123.33;
option domain-name "haifa.ibm.com";

#define option space
option space iSCSI;
option iSCSI.initiator code 203 = string;
option iSCSI.target1 code 201 = string;
option iSCSI.target2 code 202 = string;
#more iscsi options

#the actual vendor-class-identifier sent by the client = “IBM ISAN:3:1:IBM :29:56”

class “iscsi” {
   Match if substring(option vendor-class-identifier,0,8) = ”IBM ISAN”;    #increase 8 to catch more of string
   #Global class options as needed
   group {
            subclass “vendor classes” “IBM ISAN” {
               vendor-option-space iSCSI;
            }

            host blade_a {
               hardware Ethernet = 11:11:11:11:11:11;
               fixed address = 123.123.123.1;
               option iSCSI.initiator = “iqn.2005-03.com.ibm.blade_a”;
               option iSCSI.target1 = “iscsi:123.123.123.11:6:3260:0:iqn.2005-03.com.ibm.lun_a1”;
               option iSCSI.target2 = “iscsi:123.123.123.21:6:3260:0:iqn.2005.03.com.ibm.lun_a2”;
               #more iSCSI and DHCP options as needed
            } #end blade_a

            host blade_b {
               hardware Ethernet = 22:22:22:22:22:22;
               fixed address = 123.123.123.2;
               option iSCSI.initiator = “iqn.2005-03.com.ibm.blade_b”;
               option iSCSI.target1 = “iscsi:123.123.123.11:6:3260:0:iqn.2005-03.com.ibm.lun_b1”;
               option iSCSI.target2 = “iscsi:123.123.123.21:6:3260:0:iqn.2005.03.com.ibm.lun_b2”;
               #more iSCSI and DHCP options as needed
            } #end blade_b

            host blade_c {
               hardware Ethernet = 33:33:33:33:33:33;
               fixed address = 123.123.123.3;
               option iSCSI.initiator = “iqn.2005-03.com.ibm.blade_c”;
               option iSCSI.target1 = “iscsi:123.123.123.11:6:3260:0:iqn.2005-03.com.ibm.lun_c1”;
               option iSCSI.target2 = “iscsi:123.123.123.21:6:3260:0:iqn.2005.03.com.ibm.lun_c2”;
               #more iSCSI and DHCP options as needed
            } #end blade_c

            host blade_d {
               hardware Ethernet = 44:44:44:44:44:44
               fixed address = 123.123.123.4;
               option iSCSI.initiator = “iqn.2005-03.com.ibm.blade_d”;
               option iSCSI.target1 = “iscsi:123.123.123.11:6:3260:0:iqn.2005-03.com.ibm.lun_d1”;
               option iSCSI.target2 = “iscsi:123.123.123.21:6:3260:0:iqn.2005.03.com.ibm.lun_d2”;
               #more iSCSI and DHCP options as needed
            } #end blade_d
   } #end group

}
```

Figure 18: DHCP Config File using Standard Base DHCP Options Only

5.6 Sample Deployment Using iSCSI Configuration Manager

The deployment configuration using ICM involves invoking the ICM and configuring the iSCSI parameter for the 4 designated blades.

5.6.1 ICM Background

The iSCSI Initiator Configuration Wizard is a standalone Java application used to configure Initiators on blades in a BladeCenter chassis. The Wizard can communicate with the iSCSI Target enclosure Madrid and retrieve the configuration of the target LUNs. The user of the Wizard can then map target LUNs to Initiators using the Wizard's (Graphical User Interface) GUI. If the Target device is not available or not supported by the Wizard then the user can manually configure the Target's parameters using the Wizard's GUI and then map the manually added Targets to Initiators

Once the user has mapped Targets to Initiators, the Wizard will format the parameters into the BIOS layout and send the commands to the BladeCenter Management Module necessary to cause the Management Module to write the BIOS of the Initiator Blades. The Wizard can then save the parameters to a XML file for subsequent downloads.

The figure below shows the Wizard running on a Management Station. This Management Station can be any device that supports Java applications and is connected to the required IP Network, the IP Network connected to the Management Module. Optionally, the Management Station can connect to the IP Network of the Target device (which may be the same network connected to the Management Module).

The figure below shows the Wizard running on a Management Station. This Management Station can be any device that supports Java applications and is connected to the required IP Network, the IP Network connected to the Management Module. Optionally, the Management Station can connect to the IP Network of the Target device (which may be the same network connected to the Management Module).

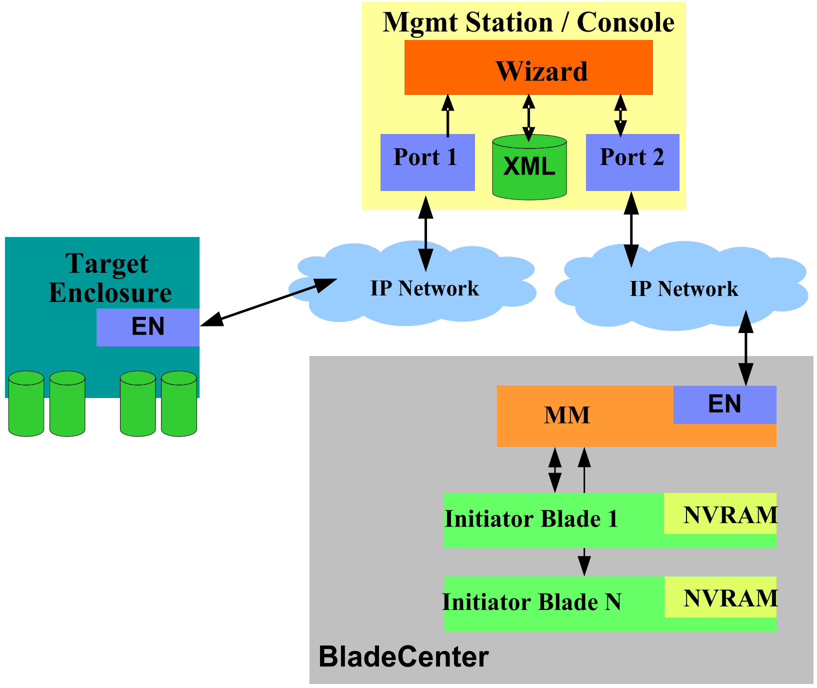


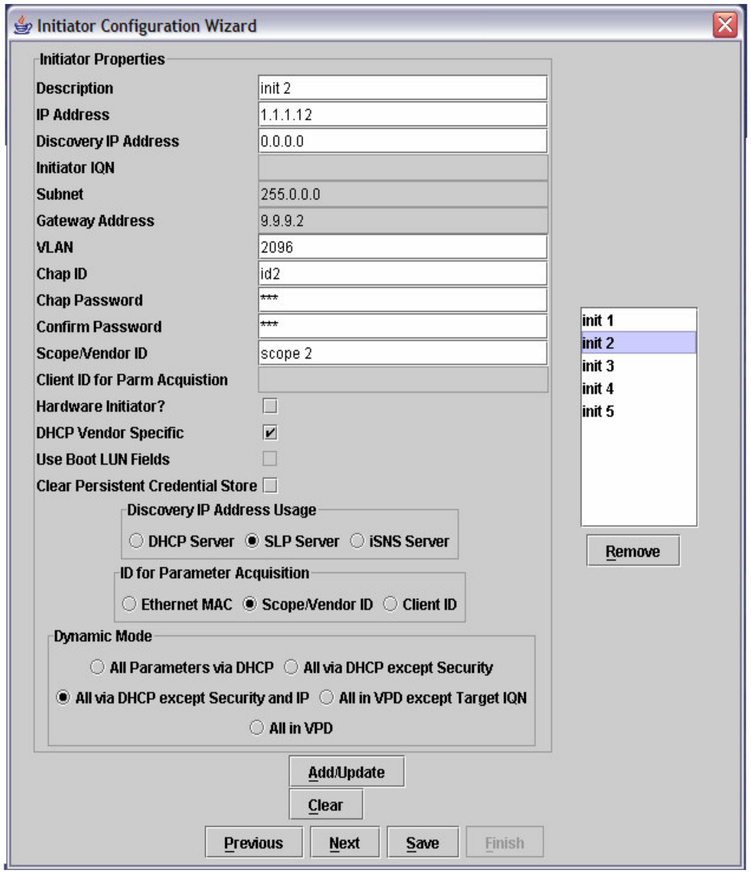
Figure 19: ICM Architecture and Topology

5.6.1.1 ICM Initiator Data Panel

The Initiator Data Panel is used to enter and display initiator parameters. If the “Retrieve Data from BladeCenter” option was selected on the environment panel then the Wizard will display the retrieved data on this panel. If the “Enter BladeCenter Data Manually” option was selected then the user must enter initiator parameters on this panel. The parameters and their descriptions are contained in the tables below.

The user can enter and display initiator parameters for multiple initiators using the “Add/Update”, “Clear” and “Remove” buttons. The initiator selection box on the far right displays the available initiators. The user can add initiators by entering the desired parameters in the “Initiator Properties” box and pressing the “Add/Update” button. The new initiator will then display in the initiator selection box. The user can modify initiator parameters by selecting the initiator in the initiator selection box, modifying the desired parameters in the “Initiator Properties” box, then pressing the “Add/Update” button. The user can set the values in the “Initiator Properties” box to their defaults by pressing the “Clear” button. The user can remove an initiator by selecting the initiator in the initiator selection box then pressing “Remove”.

For a detailed description of the individual entries, please refer to the ICM whitepaper located at **https://www-1.ibm.com/quickplace/iSCSIsw/Main.nsf**. When all the initiator data has been entered, press the “Next” button to move to the Blade Data Panel.



Initiator Configuration Wizard

Initiator Properties

Description	init 2
IP Address	1.1.1.12
Discovery IP Address	0.0.0.0
Initiator IQN	
Subnet	255.0.0.0
Gateway Address	9.9.9.2
VLAN	2096
Chap ID	id2
Chap Password	***
Confirm Password	***
Scope/Vendor ID	scope 2
Client ID for Parm Acquistion	
Hardware Initiator?	□
DHCP Vendor Specific	☑
Use Boot LUN Fields	□
Clear Persistent Credential Store	□

Discovery IP Address Usage

○ DHCP Server ● SLP Server ○ iSNS Server

ID for Parameter Acquisition

○ Ethernet MAC ● Scope/Vendor ID ○ Client ID

Dynamic Mode

○ All Parameters via DHCP ○ All via DHCP except Security
● All via DHCP except Security and IP ○ All in VPD except Target IQN
○ All in VPD

init 1
init 2
init 3
init 4
init 5

Figure 20: ICM Initiator Data Panel

Initiator Properties	Description
IP Address	iSCSI initiator IP address. This field is inactive when “All Parameters via DHCP” or “All Parameters via DHCP except Security” under “Dynamic Mode” is selected.
Discovery IP Address	This discovery IP address is an optional address used in cases where the initiator IP address is defined via the static/parameter push approach or via F1 BIOS Panels. This option aids in the dynamic or parameter acquisition approach where, for a variety of reasons, the initiator must access a specific IP address to acquire the parameters. The discovery IP address (with the appropriate Discovery IP Address Usage set to use DHCP option) is used by the initiator to identify and unicast to a specific DHCP server to acquire some or all of the iSCSI parameters. With the support of unicast, DHCP broadcast storms can be eliminated. In the future, solutions using SLP or iSNS discovery services, this discovery IP address is used for identifying the SLP or iSNS server in the network. This field is only active when “All Parameters via DHCP except Security and IP” under “Dynamic Mode” is selected.
Initiator IQN	iSCSI qualified name of initiator. This field is inactive when “All Parameters via DHCP”, “All Parameters via DHCP except Security” or “All Parameters via DHCP except Security and IP” under “Dynamic Mode” is selected.
Subnet	This network subnet mask is an optional mask used in cases where the subnet mask is defined via static/parameter push approach or via F1 BIOS panels. The mask defines the local network scope of all the IP addresses on this particular subnet. Specifically, this mask defines the local network containing stations that may be accessed directly from this station (i.e. no router or gateways involved).

Initiator Properties	Description
	This field is inactive when "All Parameters via DHCP", "All Parameters via DHCP except Security" or "All Parameters via DHCP except Security and IP" under "Dynamic Mode" is selected.
Gateway Address	This network gateway/router IP address is an optional address used in cases where the subnet mask is defined via static/parameter pus approach or via F1 BIOS panels. Note that this address defines either the gateway or the router to reach outside the current subnet and it is IETF compliant. This field is inactive when "All Parameters via DHCP", "All Parameters via DHCP except Security" or "All Parameters via DHCP except Security and IP" under "Dynamic Mode" is selected.
VLAN	The VLAN tag defines the VLAN virtual LAN to use for the iSCSI traffic within the subnet. A value of zero in this field means the initiator NIC should not insert a VLAN tag. This field is inactive when "All Parameters via DHCP" under "Dynamic Mode" is selected.
Chap ID	CHAP ID or 1st half of security key. This field is inactive when "All Parameters via DHCP" under "Dynamic Mode" is selected.
Chap Password	CHAP PW or 2nd half of security key. This field is inactive when "All Parameters via DHCP" under "Dynamic Mode" is selected.
Confirm Password	Confirmation of above. This field is inactive when "All Parameters via DHCP" under "Dynamic Mode" is selected.
Scope/Vendor ID	This Scope/vendor ID is an **optional** address used in cases where parameters are acquired from a DHCP service and some scope or vendor casting is needed to aid the DHCP service in determining the parameters to return to the DHCP client. For example, this field can be used to identify the DHCPREQUEST or DHCPINFORM transaction is within the scope of iSCSI parameter acquisition. Note that since this field is per instance and per initiator, finer levels of scoping are possible. This field is only active when "DHCP Vendor Specific" is selected.
Client ID for Parm Acquisition	This Client Alternate ID is an optional address used in cases where parameters are acquired from a DHCP service and a client ID different from EN MAC address or Scope/Vendor casting is needed to aid the DHCP service in determining the parameters to return to the DHCP client. For example, this field can be used to identify the DHCPREQUEST or DHCPINFORM transaction is within the scope of IP parameter or iSCSI parameter acquisition. Note that since this field is per instance and per initiator, finer levels of scoping are possible. Also note that the initiator may use this ID for either IP or iSCSI and use Scope/Vendor for iSCSI or IP to segment the context of acquisition. This field is only active when "All Parameters via DHCP", "All Parameters via DHCP except Security" or "All Parameters via DHCP except Security and IP" under "Dynamic Mode" is selected.
Hardware Initiator	This is a hardware initiator versus a software initiator.
DHCP Vendor Specific	This defines the appropriate DHCP options to use to acquire iSCSI parameters. Namely, whether to use the internet draft using DHCP Option 17 to acquire iSCSI path information or whether to used customer/site specific options defined in this document. This option may be superseded by the initiator and DHCP functionality. Specifically, the initiator can choose to ignore this option and ask DHCP for both Option 17 and Site/User specific Options. In turn, the DHCP server may respond with the valid options leaving the initiator to interrogate the DHCP server response to determine which options are valid. This field is only active when "All Parameters via DHCP", "All Parameters via DHCP except Security" or "All Parameters via DHCP except Security and IP" under "Dynamic Mode" is selected.
Use Boot LUN fields	This field indicates whether to use the target boot LUN field. If an initiator is to use the defined LUN number in the boot LUN fields, then this option should be disabled. If the initiator is to determine the boot LUN through other means such as intelligence or discovery, then this option should be enabled. This field is inactive when "All Parameters via DHCP", "All Parameters via DHCP except Security" or "All Parameters via DHCP except Security and IP" under "Dynamic Mode" is selected.

Initiator Properties	Description
Clear Persistent Data Store	This field is used to indicate to iSCSI service (iSCSI HBA based services namely) whether to clear IPSec certificates if they are in persistent storage.

Table 15: ICM Initiator Properties

The Discovery IP Address field indicates whether the discovery IP address should be used to access a DHCP server or should be used to access a SLP server (DA). These fields are only active when “All Parameters via DHCP except Security and IP” under “Dynamic Mode” is selected.

Discovery IP Address	Description
DHCP Server	The Discovery IP address points to a DHCP server.
SLP Server	The Discovery IP address points to a SLP server.
iSNS Server	The Discovery IP address points to an iSNS server.

Table 16: ICM Discovery IP Address Field

The Scope IDs field indicates what to use as the client ID for iSCSI parameter acquisition when querying a DHCP server for iSCSI parameters. If not present, then the parameters must be acquired from DHCP. Note that usage of Scope/Vendor ID as an additional usage scope tool is independent of this option. This field is only active when “All Parameters via DHCP”, “All Parameters via DHCP except Security” or “All Parameters via DHCP except Security and IP” under “Dynamic Mode” is selected.

Scope ID	Description
Ethernet MAC	Use Ethernet MAC address of current port. as ID.
Scope/Vendor ID	Use Scope/Vendor ID as ID.
Client ID	Use Client Alternate ID as ID.

Table 17: ICM Scope ID Field

The Parameter Acquisition Mode field is used by BIOS to determine if the iSCSI parameters are located in VPD space, for static mode, or should be acquired by a discovery service, in dynamic mode.

Parameter Acquisition Mode	Description
All Parameters via DHCP	All parameters acquired via DHCP acquisition.
All Parameters via DHCP except Security	All parms acquired via DHCP except: - Security parameters
All Parameters via DHCP except Security and IP	All parms acquired via DHCP except: -Security parameters -Initiator IP address and Discovery IP address
All in VPD except Target IQN	All parameters are present in data structure except: - Target name parms
All in VPD	All parameters are present in the data structure

Table 18: ICM Parameter Acquisition Mode Field

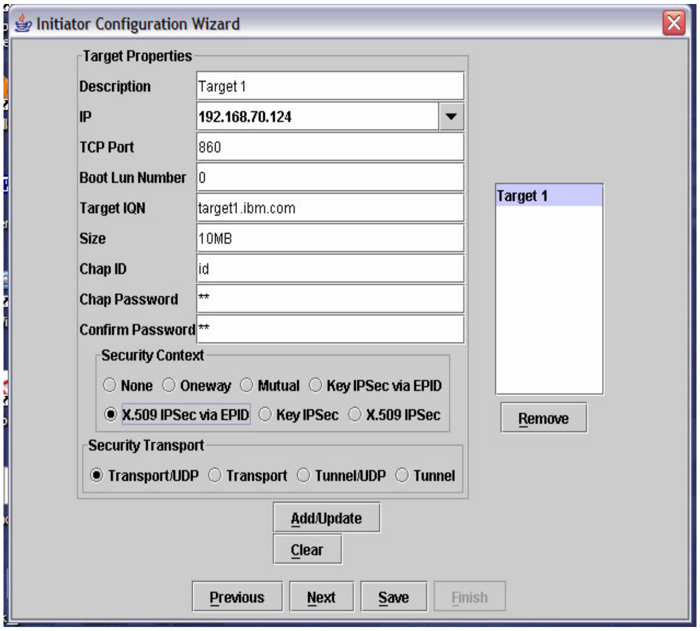
5.6.1.2 ICM Target Data Panel

The Target Data Panel is used to enter and display target parameters that are relevant to the Initiators. If the "Retrieve Data from Target" option was selected on the previous panel then the Wizard will display the retrieved data on this panel. If the "Enter Target Data Manually" option was selected then the user must enter target parameters on this panel. In both cases, target parameters entered on this panel will not be sent back to the target. The Parameters and their descriptions are contained in the tables below.

The user can enter and display target parameters for multiple targets using the "Add/Update", "Clear" and "Remove" buttons. The target selection box on the far right displays the available targets. The user can add targets by entering the desired parameters in the "Target Properties" box and pressing the "Add/Update" button. The new target will then display in the target selection box. The user can modify target parameters (in the Wizard GUI only) by selecting the target in the target selection box, modifying the desired parameters in the "Target Properties" box, and then pressing the "Add/Update" button. The user can set the values in the "Target Properties" box to their defaults by pressing the "Clear" button. The user can remove a target by selecting the target in the target selection box then pressing "Remove".

For a detailed description of the individual entries, please refer to the ICM whitepaper cited earlier in this document

When all the target data has been entered, press the "Next" button to move to the Initiator Data Panel.



Target Properties

Description	Target 1
IP	192.168.70.124
TCP Port	860
Boot Lun Number	0
Target IQN	target1.ibm.com
Size	10MB
Chap ID	id
Chap Password	**
Confirm Password	**

Security Context: ○ None ○ Oneway ○ Mutual ○ Key IPSec via EPID ● X.509 IPSec via EPID ○ Key IPSec ○ X.509 IPSec

Security Transport: ● Transport/UDP ○ Transport ○ Tunnel/UDP ○ Tunnel

Target 1

Figure 21: ICM Target Data Panel

Target Properties	Description
IP	iSCSI target IP address of storage.
TCP Port	iSCSI TCP port on target IP address.
Boot LUN Number	Boot LUN.
Target IQN	iSCSI qualified name of target.
Size	Optional size of storage.
Chap ID	CHAP ID or 1st half of security key.
Chap Password	CHAP PW or 2nd half of security key.
Confirm Password	Confirmation of above.

Table 19: ICM Target Properties

Security Context Field	Description
None	No security context to be used.
One-way	One way security to be used in logging into target (target authenticates initiator) CHAP only.
Mutual	Mutual security to be used in logging into target (target authenticates initiator / initiator authenticates target) CHAP only.
Key IPSec via EPID	Pre-shared key based IPSec authentication. The Distinguished Name is supplied via the EPID.
X.509 IPSec via EPID	X.509 Certificate based IPSec authentication. The Distinguished Name is supplied via the EPID.
Key IPSec	Pre-shared key based IPSec authentication.
X.509 IPSec	X.509 Certificate based IPSec authentication.
Confirm Password	Confirmation of above.

Table 20: ICM Security Context Field

This is an **optional** field indicating the security transport mode to be used when IPSec security context is selected. If CHAP authentication is specified, then these option bits are meaningless.

Security Transport Field	Description
Transport/UDP	Target security transport is Transport Mode / UDP Encapsulation. This field is only active when one of the IPSec options in Security Context is selected.
Transport	Target security transport is Transport Mode. This field is only active when one of the IPSec options in Security Context is selected.
Tunnel/IUDP	Target security transport is Tunnel Mode / UDP Encapsulation. This field is only active when one of the IPSec options in Security Context is selected.
Tunnel	Target security transport is Tunnel Mode. This field is only active when one of the IPSec options in Security Context is selected.
X.509 IPSec via EPID	X.509 Certificate based IPSec authentication. The Distinguished Name is supplied via the EPID.
Key IPSec	Pre-shared key based IPSec authentication.
X.509 IPSec	X.509 Certificate based IPSec authentication.

Table 21: ICM Security Transport Field

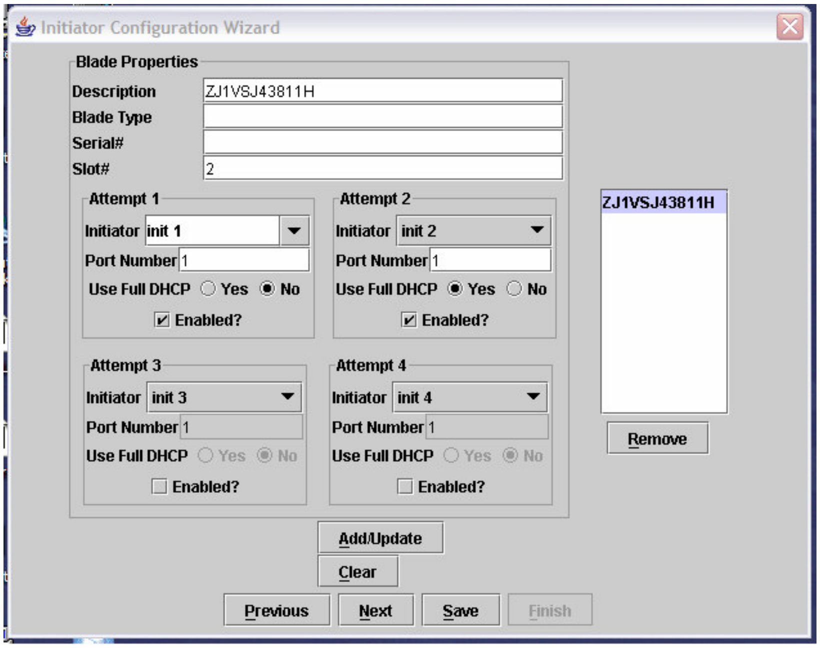
5.6.1.3 ICM Blade Data Panel

The Blade Data Panel is used to enter and display blade parameters. If the “Retrieve Data from BladeCenter” option was selected on the environment panel then the Wizard will display the retrieved data on this panel. If the “Enter BladeCenter Data Manually” option was selected then the user must enter initiator parameters on this panel. The parameters and their descriptions are contained in the tables below.

The user can enter and display blade parameters for multiple blades using the “Add/Update”, “Clear” and “Remove” buttons. The blade selection box on the far right displays the available blades. The user can add blades by entering the desired parameters in the “Blade Properties” box and pressing the “Add/Update” button. The new blade will then display in the blade selection box. The user can modify blade parameters by selecting the blade in the blade selection box, modifying the desired parameters in the “Blade Properties” box, then pressing the “Add/Update” button. The user can set the values in the “Blade Properties” box to their defaults by pressing the “Clear” button. The user can remove a blade by selecting the blade in the blade selection box then pressing “Remove”.

For a detailed description of the individual entries, please refer to the ICM whitepaper located cited earlier in this document.

When all the blade data has been entered, press the “Next” button to move to the Initiator/Target Mapping Panel.



Initiator Configuration Wizard

Blade Properties

Description: ZJ1VSJ43811H

Blade Type:

Serial#:

Slot#: 2

Attempt 1
- Initiator: init 1
- Port Number: 1
- Use Full DHCP ○ Yes ● No
- ☑ Enabled?

Attempt 2
- Initiator: init 2
- Port Number: 1
- Use Full DHCP ● Yes ○ No
- ☑ Enabled?

Attempt 3
- Initiator: init 3
- Port Number: 1
- Use Full DHCP ○ Yes ● No
- □ Enabled?

Attempt 4
- Initiator: init 4
- Port Number: 1
- Use Full DHCP ○ Yes ● No
- □ Enabled?

ZJ1VSJ43811H

Figure 22: ICM Blade Data Panel

Blade Properties	**Description**
Blade Type	An optional field describing the type of blade.
Serial#	An optional field containing the serial number of the blade.
Slot#	The slot in BladeCenter where this blade resides. If the blade occupies more than one slot then this is the slot number of the slot containing the LEDs and power button.

Table 22: ICM Blade Properties

Each blade can have up to four initiator attempts. The are attempted one at a time in order one to four until a target is contacted. The parameters for the four attempts are identical and described once in the table below.

Blade Properties	**Description**
Initiator	The description of the initiator configured on the Initiator Data Panel to use for this attempt. This field is only active when "Enabled" is selected and "Use Full DHCP" is set to "no".
Port Number	The Ethernet port number to use for this attempt. This field is only active when "Enabled" is selected.
Use Full DHCP	If this is set to "Yes" then all configured initiator parameters for this attempted are ignored and retrieved from a DHCP server. This field is only active when "Enabled" is selected.
Enabled	Whether or not this attempt is enabled.

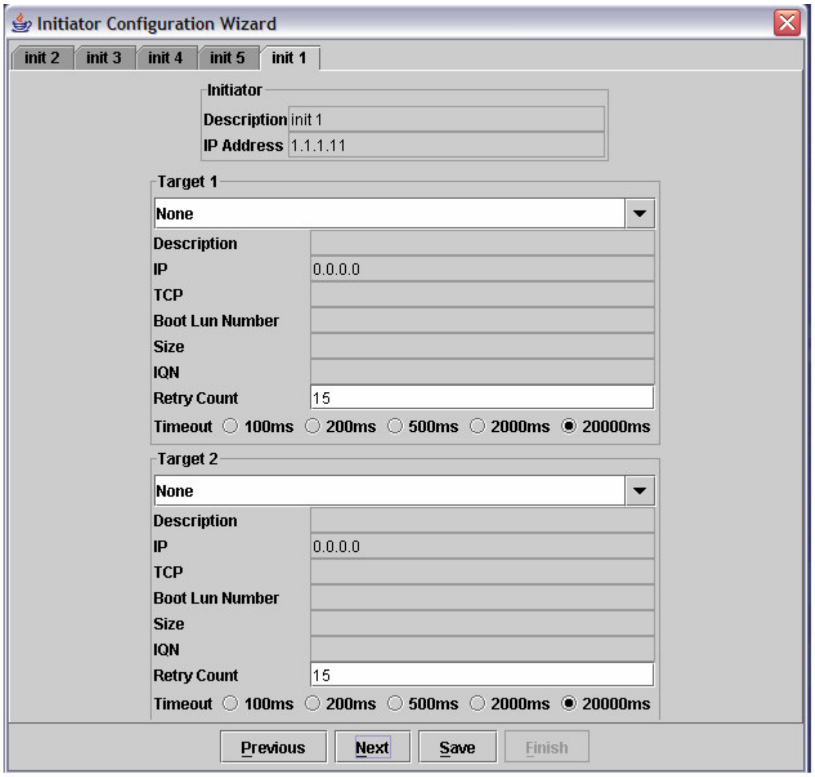
5.6.1.4 iSCSI Initiator⬄iSCSI Target Mapping Panel

This panel is used to assign targets to initiators. Each initiator can be assigned up to two targets. The initiators configured on the previous panels are displayed has tabs along the top of this panel. To select the initiator press the tab containing the description of the desired initiator. The configured targets are contained in the pull downs in the boxes marked “Target 1” and “Target 2”. A target can only be assigned once to an initiator. When an initiator is configured in a blade’s attempt on the Blade Data Panel, the targets assigned to the initiator will be contacted when that attempt is activated during the boot process.

The fields Retry Count and Timeout are only active when “All in VPD” or “All in VPD except Target IQN” in selected in the Dynamic Mode box of the Initiator Data Panel.

For a detailed description of the individual entries, please refer to the ICM whitepaper located cited earlier in this document.

Press the “Next” button to move to the Blade Configuration Download Panel.



Initiator

Description: init 1

IP Address: 1.1.1.11

Target 1

None

Description:

IP: 0.0.0.0

TCP:

Boot Lun Number:

Size:

IQN:

Retry Count: 15

Timeout ○ 100ms ○ 200ms ○ 500ms ○ 2000ms ● 20000ms

Target 2

None

Description:

IP: 0.0.0.0

TCP:

Boot Lun Number:

Size:

IQN:

Retry Count: 15

Timeout ○ 100ms ○ 200ms ○ 500ms ○ 2000ms ● 20000ms

Figure 23: iSCSI Initiator to iSCSI Target Mapping Panel

Initiator⇔Target Properties	**Description**
Initiator	
Initiator description	User friendly description of initiator.
Initiator IP address	IP address of the initiator.
Target 1	
Target 1 description	User friendly description of target.
Target 1 IP address	IP address of the target.
Target 1 TCP port	TCP port used on target IP address
Target 1 Boot LUN number	LUN number used on target for booting
Target 1 Size	Size of the boot LUN on target
Target 1 IQN	iSCSI name of the target
Target 1 Retry Count	Number of retries to attempt against target before declaring unavailable

Initiator⇔Target Properties	**Description**
Target 1 Timeout	Timeout value to for each try against target
Target 2	
Target 2 description	User friendly description of target.
Target 2 IP address	IP address of the target.
Target 2 TCP port	TCP port used on target IP address
Target 2 Boot LUN number	LUN number used on target for booting
Target 2 Size	Size of the boot LUN on target
Target 2 IQN	iSCSI name of the target
Target 2 Retry Count	Number of retries to attempt against target before declaring unavailable
Target 2 Timeout	Timeout value to for each try against target

Table 23: ICM Initiator - Target Mapping Properties

5.7 ICM Sample Configuration

5.8 Sample Deployment Diskless Blade Configuration

5.8.1 Supported Blades

The blade generation supported is 8678 class blades. Any earlier generation of blades that are desired to be included in an iSCSI boot deployment requires contacting IBM to discuss the requirement and understand the deployment differences.

5.8.2 Firmware Updates

It is prudent to make sure the various subsystems in BladeCenter are at the current firmware levels. These subsystems include Blade BIOS, Blade service processor, Bladecenter Management module, and Ethernet switches. The latest firmware can be found at

http://www.ibm.com/support/us

5.8.3 Blade Configuration

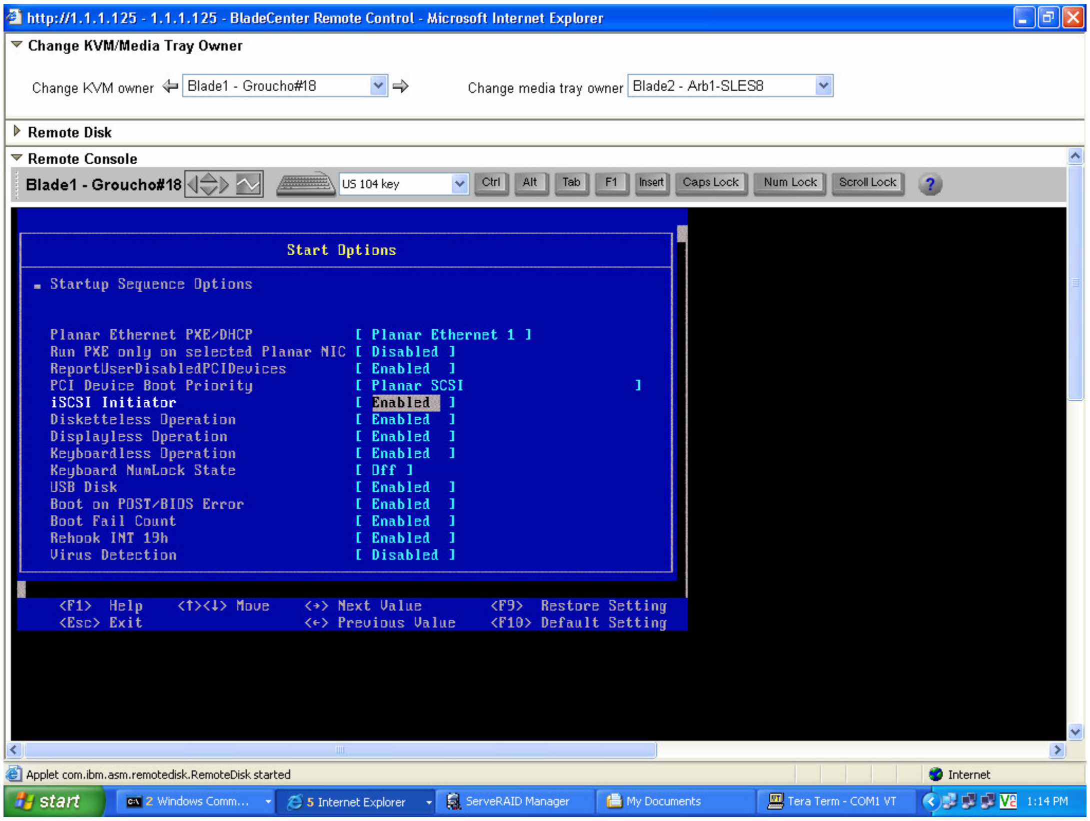
The following steps guide the configuration of a given blade

1. Select the given blade
2. if install is to be done on this blade, ensure local disks are present. If a donor, recipient model is preferred, ensure the donor blade has local disks
3. Power up the blade and enter F1/BIOS setup
4. Load default settings
5. Refer to the next section on NIC configuration
6. Assign the KVM and media tray to the blade using the management module interface

5.8.4 Blade Configuration

On the F1/BIOS panels, there are several NIC specific settings that can be applied. In the 8643 blades, there are two Ethernet NICs resident on the base blade. In general, the blade configuration entails two steps. First, the ICM is used to deploy iSCSI parameters to the given blade, even if full DHCP approach is going to be used to acquire the actual parameters. Note that in future releases, the need for ICM to initialize the data structures will be removed. Second, on the given blade, iSCSI boot needs to be selected..

Below is a screen shot of the F1/BIOS panel displaying the selection of iSCSI boot



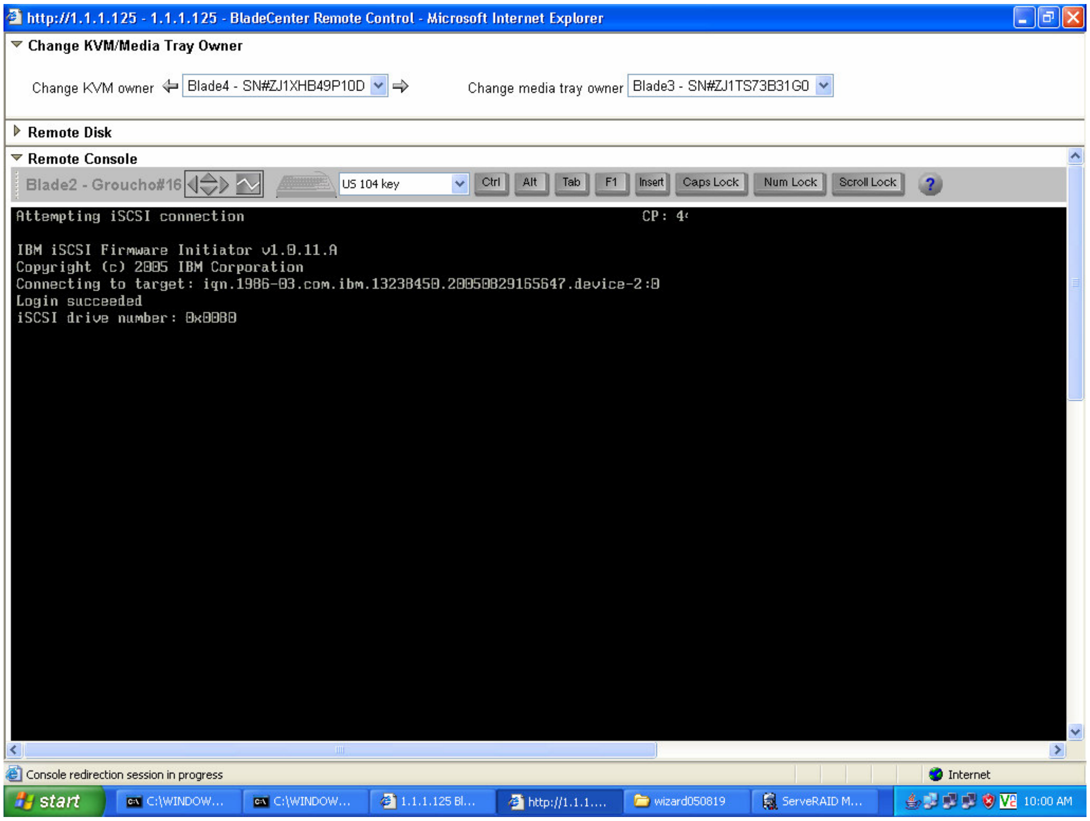
```
                          Start Options

 ▪ Startup Sequence Options

  Planar Ethernet PXE/DHCP             [ Planar Ethernet 1 ]
  Run PXE only on selected Planar NIC  [ Disabled ]
  ReportUserDisabledPCIDevices         [ Enabled  ]
  PCI Device Boot Priority             [ Planar SCSI                    ]
  iSCSI Initiator                      [ Enabled  ]
  Disketteless Operation               [ Enabled  ]
  Displayless Operation                [ Enabled  ]
  Keyboardless Operation               [ Enabled  ]
  Keyboard NumLock State               [ Off ]
  USB Disk                             [ Enabled  ]
  Boot on POST/BIOS Error              [ Enabled  ]
  Boot Fail Count                      [ Enabled  ]
  Rehook INT 19h                       [ Enabled  ]
  Virus Detection                      [ Disabled ]

     <F1>  Help   <↑><↓> Move      <→> Next Value      <F9>  Restore Setting
     <Esc> Exit                    <←> Previous Value  <F10> Default Setting
```

Figure 24: F1/BIOS Panel for Enabling iSCSI boot

After these two steps as well as all the configuration and disk image deployment steps, the given blade can be booted up using iSCSI as the boot device.

If the blade booting is successful, the BIOS screen will indicate a successful login.

Below is a screen shot of a successful login.



Change KVM owner: Blade4 - SN#ZJ1XHB49P10D

Change media tray owner: Blade3 - SN#ZJ1TS73B31G0

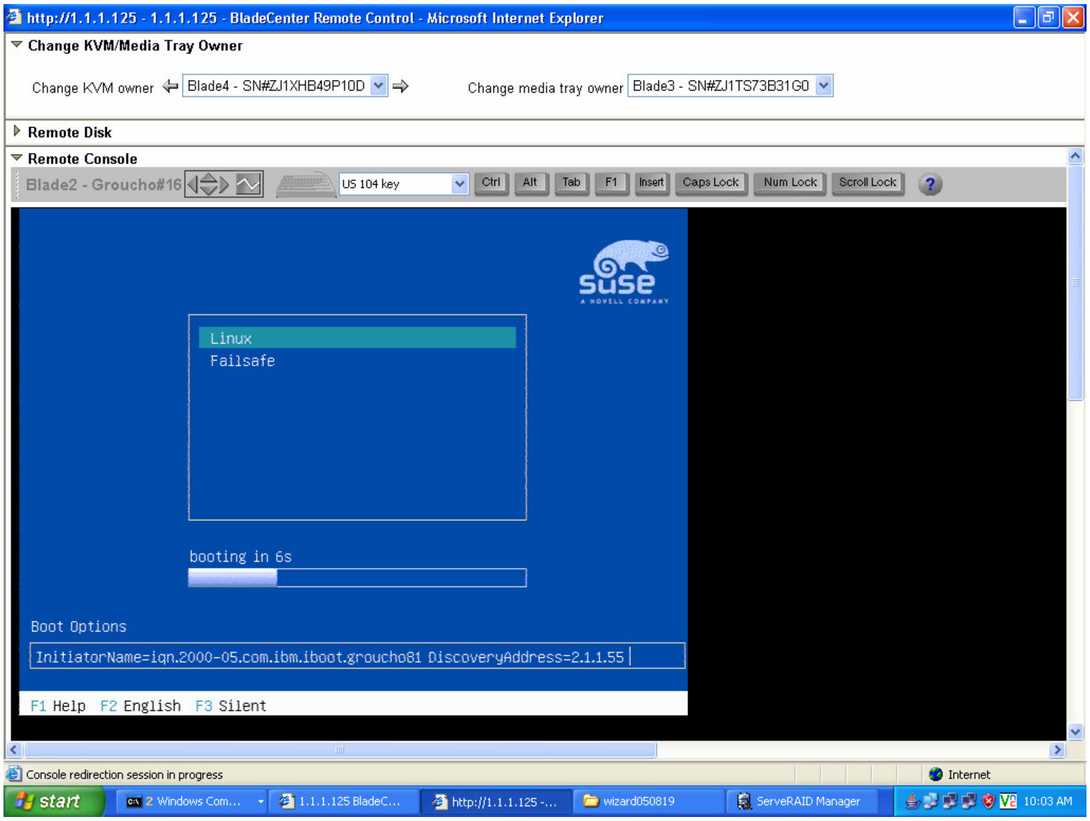
US 104 key

```
Attempting iSCSI connection                                            CP: 4¢

IBM iSCSI Firmware Initiator v1.0.11.A
Copyright (c) 2005 IBM Corporation
Connecting to target: iqn.1986-03.com.ibm.13238450.20050829165647.device-2:0
Login succeeded
iSCSI drive number: 0x0080
```

Figure 25: BIOS Panel of a Successful iSCSI Login to an iSCSI Target

As the booting process proceeds, additional information is provided.

Below is a screen shot of a SUSE boot on a diskless blade



http://1.1.1.125 - 1.1.1.125 - BladeCenter Remote Control - Microsoft Internet Explorer

Change KVM/Media Tray Owner

Change KVM owner: Blade4 - SN#ZJ1XHB49P10D

Change media tray owner: Blade3 - SN#ZJ1TS73B31G0

US 104 key

```
Linux
Failsafe

booting in 6s

Boot Options
InitiatorName=iqn.2000-05.com.ibm.iboot.groucho81 DiscoveryAddress=2.1.1.55
```

Console redirection session in progress

Figure 26: SUSE GRUB Screenshot of a Successful iSCSI boot

6.0 Restrictions and Debug Strategies

6.1 iSCSI Restrictions for In band Deployment

There are several restrictions to the usage of the iSCSI software boot via in-band.

Restriction 1: VLAN usage is confined to port VLAN groups. This restriction is the result of the network interface not configured for VLAN at boot time (only after the OS is up and running). With port VLANs, all the iSCSI solution components must be "visible" within a common port group.

Restriction 2: It should be noted that if the network or iSCSI service is taken off line during the normal operation of the blade, connectivity for the boot and swap partitions will be lost and the blade will, in essence, "freeze". As a result, the user or admin must be very careful when altering the iSCSI behavior during operation.

Restriction 3: When the iSCSI HBA is installed, using iSCSI software initiator for booting is not supported. Conceptually, there may be limited value in using iSCSI software boot if an iSCSI HBA and associated infrastructure is in place. Specifically, if there is a need to support both on a given blade, please contact IBM to discuss enhancing support to accommodate this configuration.

Restriction 4: When the Blade Storage Expansion (BSE) is installed, using iSCSI software initiator for booting is not supported. This issue revolves around device enumeration at POST time. There are efforts to resolve this shortly, please contact IBM for status update.

Restriction 5: When using 64 bit Linux operating systems based off EM64T, using iSCSI software initiator for booting is not supported. While EM64T has been verified informally, formal test verification of iSCSI software boot in an EM64T environment has not been 100% successful. As a result, this restriction has been put in place until the EM64T usage has passed all testing. There are efforts to resolve this shortly, please contact IBM for status update.

Restriction 6: When executing the Blade Diagnostics, the results for the disk test are invalid because the disk is actually located remote from the system. As a result, the Blade Diagnostics will reflect accurate results for all other components but the disk drive. To determine if the remote iSCSI disk is failing, please refer to the iSCSI target problem determination procedures. In future releases, enhancements will be made to the iSCSI BIOS service to provide some rudimentary testing of the remote disks when Blade Diagnostics are executed and reflect the results of such tests will be presented in the context of the Blade Diagnostics (appropriate screens). Please contact IBM for updates regarding Blade Diagnostic behavior.

Restriction 7: If there is a desire to perform continual or perpetually repeating iSCSI booting as some sort of stress test scenario, then there is a restriction currently in place where the time span between reboots of such scenario should be at least 1 minute. This restriction is currently in place to allow the iSCSI BIOS initiator, Ethernet network, and iSCSI target sufficient time to sufficiently complete all tasks from the previous boot sequence. This is restriction is akin to repetitively logging in and logging out of an iSCSI target as fast as possible while still have disks transactions pending in the iSCSI target cache. There are efforts to better characterize and resolve this restriction in terms of iSCSI initiator behavior as well as iSCSI target behavior. Please contact IBM for updates if this type of usage scenario is desired.

6.2 iSCSI Error Tree for In band Deployment

As a debug aid, the following error tree can provide insight for problem determination

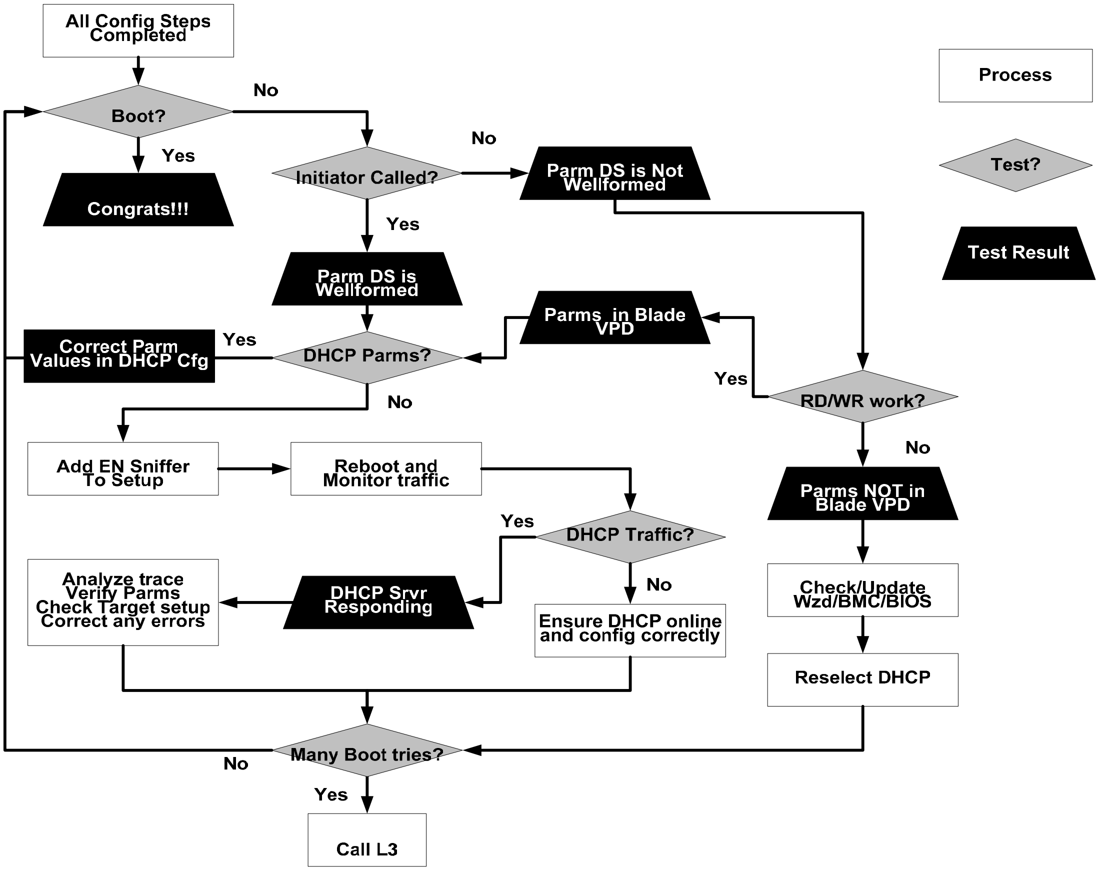


Figure 27: Error Tree for In Band iSCSI Deployment

6.3 iSCSI Parameter Debug for In band Deployment

For the iSCSI parameter deployment, it is very helpful to have a network sniffer such as Ethereal on the network to monitor the flow of packets. Using such a tool, one can monitor the transactions between the recipient blade and DHCP service or iSCSI target. As a result, several failure scenarios can be explored. In this discussion, the failure scenarios are focused on actual parameter deployments and not more global issues such as not having a DHCP service in the solution.

Failure scenario 1: The recipient blade cannot find DHCP server. This is a result of either not finding the actual DHCP service or that the DHCP service did not respond. The network sniffer can assist on determine whether the DHCP server respond to the recipient blade. Without getting into the DHCP protocol in too much detail, the packets to look for are DHCPACK packets. If they are not found in the network trace, then the DHCP service is not available (either because it's not there or not active or, in general, a reservation has not been entered for this given recipient's Ethernet MAC address, or something fundamentally wrong with the service itself). If they are found, the options the DHCP service responded with should include the options defined for this recipient blade (among others, option 17 or option 201,202,203 are present). Ensure the appropriate options are defined and configured appropriately.

Failure scenario 2: Incorrect iSCSI parameters provided. This is a result of having appropriate options responded but the content of each option is not valid. Use the network sniffer verifies the string associated with the iSCSI options. Correct the DHCP configuration as needed.

Failure scenario 3: Unable to find or login into the iSCSI target. This is the result of not having the iSCSI target on the network or the target configuration correct. Recall, the iSCSI target must be aware of the iSCSI initiator name that is targeting the given LUN, so the configuration of the iSCSI target must be consistent with the DHCP configuration for the given recipient blade.

Failure scenario 4: There are DHCP conflicts on the network. This is the result of DHCP service definition and setup. For example, several DHCP servers may be serving up iSCSI parameters (base or vendor options) against the same DHCP client Ethernet MAC address. As a result, non iSCSI aware DHCP servers should not serve up iSCSI parameters and, thus, an iSCSI aware and iSCSI unaware DHCP servers can coexist on the network. Using the network sniffer such as Ethereal (www.ethereal.com) to debug the network traffic, ensure that the desired iSCSI parameter definitions are present on the network. For example, ensure that 2 iSCSI aware DHCP servers are not responding to the given client request where the DHCP server of interest (only one is intended to provide the iSCSI parameters for the given client) is not the second DHCP server response as seen by the client. For this situation, resolve the conflict by either making a high availability DHCP configuration is set up so that only one server is responding for any given requests (active passive configuration) or that each server in a high availability configuration is correctly configured to load balance (load balance configuration) or making sure each DHCP server is responding to specific clients (recipient blade) requests. If option 17 is used, then this approach must be folded into the high availability DHCP server pair already in the network or that there is only 1 DHCP server in the network. This is because the iSCSI unaware DHCP server may have an option 17 setup for bootp for this specific recipient blade.

6.4 iSCSI Restrictions for Out of Band Deployment

There are several restrictions to the usage of the iSCSI software boot via out-of-band.

Restriction 1: VLAN usage is confined to port VLAN groups. This restriction is the result of the network interface not configured for VLAN at boot time (only after the OS is up and running). With port VLANs, all the iSCSI solution components must be “visible” within a common port group.

Restriction 2: It should be noted that if the network or iSCSI service is taken off line during the normal operation of the blade, connectivity for the boot and swap partitions will be lost and the blade will, in essence, “freeze”. As a result, the user or admin must be very careful when altering the iSCSI behavior during operation.

Restriction 3: When the iSCSI HBA is installed, using iSCSI software initiator for booting is not supported. Conceptually, there may be limited value in using iSCSI software boot if an iSCSI HBA and associated infrastructure is in place. Specifically, if there is a need to support both on a given blade, please contact IBM to discuss enhancing support to accommodate this configuration.

Restriction 4: When the Blade Storage Expansion (BSE) is installed, using iSCSI software initiator for booting is not supported. This issue revolves around device enumeration at POST time. There are efforts to resolve this shortly, please contact IBM for status update.

Restriction 5: When using 64 bit Linux operating systems based off EM64T, using iSCSI software initiator for booting is not supported. While EM64T has been verified informally, formal test verification of iSCSI software boot in an EM64T environment has not been 100% successful. As a result, this restriction has been put in place until the EM64T usage has passed all testing. There are efforts to resolve this shortly, please contact IBM for status update.

Restriction 6: When executing the Blade Diagnostics, the results for the disk test are invalid because the disk is actually located remote from the system. As a result, the Blade Diagnostics will reflect accurate results for all other components but the disk drive. To determine if the remote iSCSI disk is failing, please refer to the iSCSI target problem determination procedures. In future releases, enhancements will be made to the iSCSI BIOS service to provide some rudimentary testing of the remote disks when Blade Diagnostics are executed and reflect the results of such tests will be presented in the context of the Blade Diagnostics (appropriate screens). Please contact IBM for updates regarding Blade Diagnostic behavior.

Restriction 7: If there is a desire to perform continual or perpetually repeating iSCSI booting as some sort of stress test scenario, then there is a restriction currently in place where the time span between reboots of such scenario should be at least 1 minute. This restriction is currently in place to allow the iSCSI BIOS initiator, Ethernet network, and iSCSI target sufficient time to sufficiently complete all tasks from the previous boot sequence. This is restriction is akin to repetitively logging in and logging out of an iSCSI target as fast as possible while still have disks transactions pending in the iSCSI target cache. There are efforts to better characterize and resolve this restriction in terms of iSCSI initiator behavior as well as iSCSI target behavior. Please contact IBM for updates if this type of usage scenario is desired.

6.5 iSCSI Error Tree for Out of Band Deployment

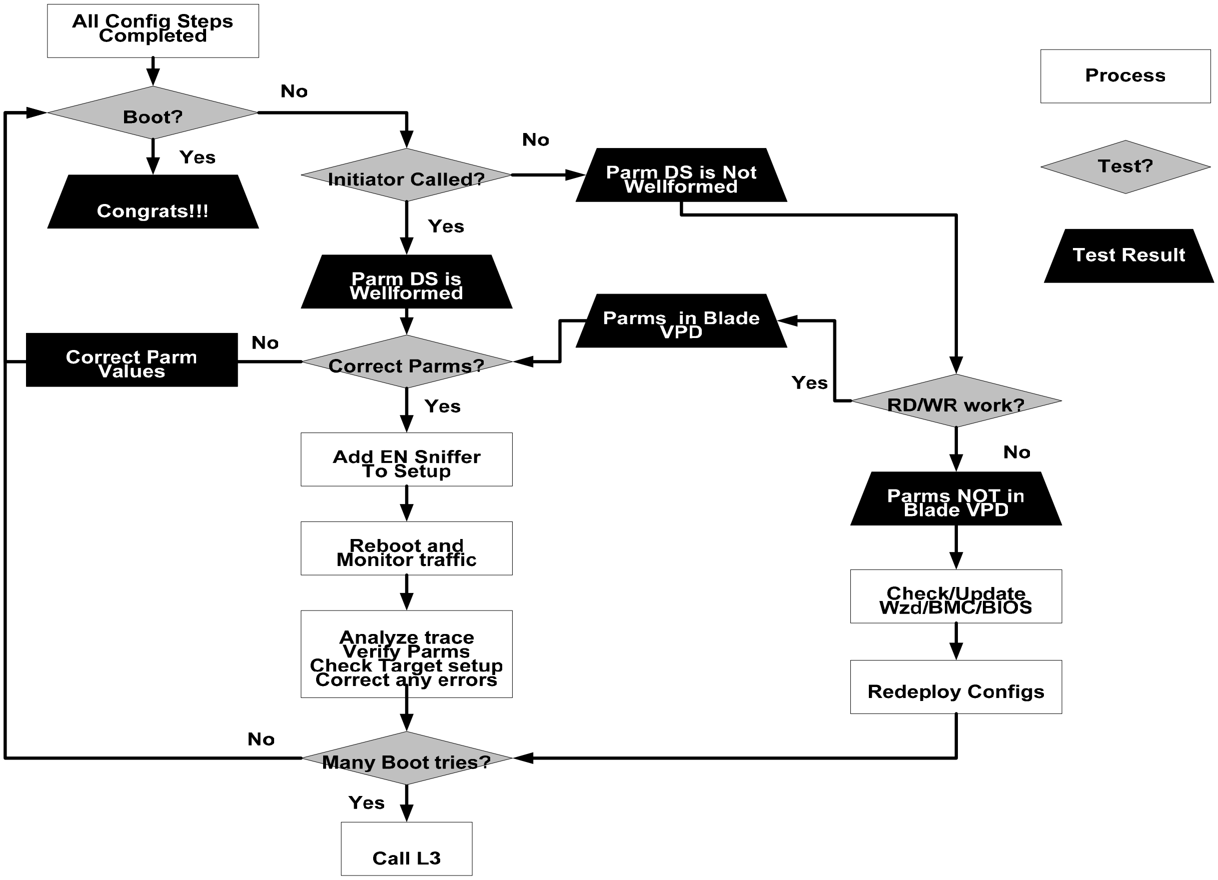


Figure 28: Error Tree for iSCSI Out of Band Deployment

6.6 iSCSI Parameter Debug for Out of Band Deployment

For the iSCSI parameter deployment, it is very helpful to have a network sniffer such as Ethereal on the network to monitor the flow of packets. Using such a tool, one can monitor the transactions between the recipient blade and iSCSI target. As a result, several failure scenarios can be explored. In this discussion, the failure scenarios are focused on actual parameter deployments and not more global issues such as not having an iSCSI target in the solution.

Failure scenario 1: The ICM cannot successfully write the iSCSI parameters to the given blade. As discussed in the fault tree, the admin must ensure the correct level of BMC firmware is installed on the blade. Examine the BMC firmware to determine the installed version and if downlevel from the support version, please upgrade the BMC. Once up to the correct BMC firmware version, test the deployment of the configuration by writing then reading back the parameters deployed to ensure they are actually in the BMC. Make sure that the read back parameters are present and not the written parameters (via the saved config file).

Failure scenario 2: If the BIOS has not launched the iSCSI initiator, confirm that the BIOS panels have enabled iSCSI as a boot device. Next ensure valid parameters reside in the BMC for iSCSI. This can be done by reading out the parameters using the ICM. If all seems appropriate, note any error message and contact the appropriate support personnel.

Failure scenario 3: If the BIOS launches the iSCSI boot initiator but the boot initiator fails to successfully log into the target, refer to the debug methods described in the inband section. Namely, consider using a network sniffer such as Ethereal (www.ethereal.com) to debug and monitor the network traffic. One arcane scenario to consider is the authentication setting mismatch, which would be presented in the sniffer log as authentication error or login failure due to authentication error. In this case, ensure the authentication settings at the initiator and target are in agreement.

6.7 iSCSI Image Creation and Deployment Debug

During the creation and deployment of the disk images, it may be necessary to verify operation.

Failure scenario 1: Can't move the disk image to the iSCSI target. This is the result of either not configuring the producer blade correctly to access the target, not correctly configuring the iSCSI target allow the producer blade to access LUN x. The first things to check are to ensure the configuration on each is correct. Once confirmed, use the producer blade to access (not move any images) the target and see that the LUN is visible and of the right size. The Linux open source iSCSI initiator can be used for this. If this is successful, create a disk image and deploy it to another partition on the producer blades local disk. If that works, then attempt to move the disk image to the target via the steps mentioned in the deployment example. If that does not work, review the disk image creation and deployment steps and adjust as appropriate.

Failure scenario 2: During recipient boot, Linux indicates that there is a problem with the sda drive (either cant find the master boot record or some file or directory is missing). This is the result that the deployment steps having a problem. Using the producer blade, check the iSCSI target to verify that the appropriate components are there. If not, update the iSCSI target by repeating the deployment steps. If so, contact IBM for guidance.

Failure scenario 3: Once Linux is up and running, an unexpected event occurs causing a system crash. This can be the result of one of several events. Before proceeding, note the information presented on the screen. First, ensure the iSCSI target is up and operational. If not, proceed to get the iSCSI target back on line. If so, ensure there are no catastrophic network issues such as power outage along the path to the iSCSI target. If no outages, record the application configuration (i.e. the applications installed on the LUN in the iSCSI target) and the crash dump information. Contact IBM with this information for assistance.

Failure scenario 4: During image creation, a failure in the process prevented image creation from completing successfully. This is the result of missing components on the producer blade source image. For example, if the wrong network driver is used or the iSCSI initiator is missing. Verify that the missing components are there. If not, ensure they are added and configured appropriately. If so, contact IBM with the so

Appendix A: Screen Dialogue for Configuration

Below presents the screen dialogue involved with configuring the disk image

Iboot_Ft Worth SuSE 9 setup

```
Groucho83-SLES9-SP2-iboot-fw: # chkconfig --del iscsi
iscsi                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
Groucho83-SLES9-SP2-iboot-fw: # kate /etc/iscsi.conf
# iSCSI configuration file - see iscsi.conf(5)
DiscoveryAddress=2.1.1.55
# Authentication Settings
Groucho83-SLES9-SP2-iboot-fw: # kate /etc/initiatorname.iscsi
## DO NOT EDIT OR REMOVE THIS FILE!
## If you remove this file, the iSCSI daemon will not start.
## If you change the InitiatorName, existing access control lists
## may reject this initiator.  The InitiatorName must be unique
## for each iSCSI initiator.  Do NOT duplicate iSCSI InitiatorNames.
InitiatorName=iqn.2000-05.com.ibm.iboot.groucho83
Groucho83-SLES9-SP2-iboot-fw:/ # /etc/init.d/iscsi start
Starting iSCSI: iscsi iscsid fsck/mount                              done
Groucho83-SLES9-SP2-iboot-fw:/ # iscsi-ls -c
***************************************************************************
        Cisco iSCSI Driver Version ... 4.0.188.18 ( 21-May-2004 )
***************************************************************************
TARGET NAME             : iqn.1986-03.com.ibm.13238450.20050826105134.device-1
TARGET ALIAS            : Device-1
HOST NO                 : 3
BUS NO                  : 0
TARGET ID               : 1
TARGET ADDRESS          : 2.1.1.55:3260
SESSION STATUS          : ESTABLISHED AT Sat Aug 27 13:43:15 2005
NO. OF PORTALS          : 2
PORTAL ADDRESS 1        : 2.1.1.55:3260,1
PORTAL ADDRESS 2        : 3.1.1.200:3260,1
SESSION ID              : ISID 00023d000001 TSID 1d5

TARGET CONFIGURATION KEYS :
-------------------------
INITIAL R2T             : Yes
IMMEDIATE DATA          : Yes
HEADERDIGEST            : NONE
DATA DIGEST             : NONE
FIRSTBURSTLENGTH        : 65536
MAXBURSTLENGTH          : 262144
MAXRECVDATASEGMENTLENGTH: initiator - 131072
                          target    - 65536
LOGIN TIMEOUT           : 15
AUTH TIMEOUT            : 45
ACTIVE TIMEOUT          : 5
IDLE TIMEOUT            : 60
PING TIMEOUT            : 5
Groucho83-SLES9-SP2-iboot-fw:/ # ls -la /dev/disk/by-id
total 1
drwxr-xr-x  2 root root 472 Aug 27 13:43 .
drwxr-xr-x  4 root root  96 Aug 24 00:27 ..
lrwxrwxrwx  1 root root   9 Aug 24 19:57 IOMEGA-Mini1GB*IOM2B5 -> ../../sdc
lrwxrwxrwx  1 root root  10 Aug 24 19:57 IOMEGA-Mini1GB*IOM2B5p1 -> ../../sdc1
lrwxrwxrwx  1 root root   9 Aug 24 00:27 LG-CD-ROM_CRN-8245B -> ../../sr0
lrwxrwxrwx  1 root root   9 Aug 27 13:21 TEAC-CD-224E -> ../../sr0
lrwxrwxrwx  1 root root   9 Aug 27 13:21 TEAC-FD-05PUB -> ../../sdb
lrwxrwxrwx 1 root root  9 Aug 27 13:43 iscsi-iqn.1986-03.com.ibm.13238450.20050826105134.device-1-0
-> ../../sdc
lrwxrwxrwx  1 root root  10 Aug 27 13:43
iscsi-iqn.1986-03.com.ibm.13238450.20050826105134.device-1-0p1 -> ../../sdc1
lrwxrwxrwx  1 root root  10 Aug 27 13:43
iscsi-iqn.1986-03.com.ibm.13238450.20050826105134.device-1-0p2 -> ../../sdc2
```

```
Groucho83-SLES9-SP2-iboot-fw:/ # fdisk -l

Disk /dev/sda: 36.3 GB, 36362518528 bytes
255 heads, 63 sectors/track, 4420 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1         131     1052226   82  Linux swap
/dev/sda2             132        4420    34451392+  83  Linux

Disk /dev/sdc: 41.9 GB, 41911582720 bytes
64 heads, 32 sectors/track, 39970 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
Groucho83-SLES9-SP2-iboot-fw:/ # fdisk /dev/sdc

The number of cylinders for this disk is set to 39970.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): o
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.


The number of cylinders for this disk is set to 39970.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
Groucho83-SLES9-SP2-iboot-fw:/ # fdisk /dev/sdc

The number of cylinders for this disk is set to 39970.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sdc: 41.9 GB, 41911582720 bytes
64 heads, 32 sectors/track, 39970 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-39970, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-39970, default 39970): +1024M
```

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
Groucho83-SLES9-SP2-iboot-fw:/ # fdisk /dev/sdc

The number of cylinders for this disk is set to 39970.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sdc: 41.9 GB, 41911582720 bytes
64 heads, 32 sectors/track, 39970 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1        1024     1048560   83  Linux

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (1025-39970, default 1025):
Using default value 1025
Last cylinder or +size or +sizeM or +sizeK (1025-39970, default 39970):
Using default value 39970

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
Groucho83-SLES9-SP2-iboot-fw:/ # fdisk -l

Disk /dev/sda: 36.3 GB, 36362518528 bytes
255 heads, 63 sectors/track, 4420 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1         131     1052226   82  Linux swap
/dev/sda2             132        4420    34451392+  83  Linux

Disk /dev/sdc: 41.9 GB, 41911582720 bytes
64 heads, 32 sectors/track, 39970 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1               1        1024     1048560   83  Linux
/dev/sdc2            1025       39970    39880704   83  Linux
Groucho83-SLES9-SP2-iboot-fw:/ # mkswap /dev/sdc1
Setting up swapspace version 1, size = 1073721 kB
Groucho83-SLES9-SP2-iboot-fw:/ # mkfs -t reiserfs /dev/sdc2 -labeliscsi_iboot83
mkfs.reiserfs 3.6.19 (2003 www.namesys.com)

A pair of credits:
Oleg Drokin was the debugger for  V3 during most of the time that  V4 was under
development,  and was quite  skilled and fast at it.  He wrote  the large write
optimization of V3.

Hans Reiser was the project initiator,  source of all funding for the first 5.5
years. He is the architect and official maintainer.
```

```
Guessing about desired format.. Kernel 2.6.5-7.191-smp is running.
Format 3.6 with standard journal
Count of blocks on the device: 9970176
Number of blocks consumed by mkreiserfs formatting process: 8516
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 51cdae34-bfa1-47aa-bfbf-dc81e33b4ea0
LABEL: abeliboot83
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/sdc2'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok

Tell your friends to use a kernel based on 2.4.18 or later, and especially not a
kernel based on 2.4.9, when you use reiserFS. Have fun.

ReiserFS is successfully created on /dev/sdc2.
Groucho83-SLES9-SP2-iboot-fw:/ # /sbin/mkinitrd -k vmlinuz -i initrd-iboot -m "iscsi ipv6 tg3
reiserfs" -D eth0
Root device:    /dev/sda2 (mounted on / as reiserfs)
Module list:    iscsi ipv6 tg3 reiserfs af_packet

Kernel image:   /boot/vmlinuz
Initrd image:   /boot/initrd-iboot
Shared libs:    lib/ld-2.3.3.so lib/libblkid.so.1.0 lib/libc.so.6 lib/libselinux.so.1
lib/libuuid.so.1.2
Modules:        kernel/drivers/scsi/scsi_mod.ko kernel/drivers/scsi/sd_mod.ko extra/iscsi.ko
kernel/net/ipv6/ipv6.ko kernel/drivers/net/tg3.ko kernel/fs/reiserfs/reiserfs.ko
kernel/net/packet/af_packet.ko
Including:      iscsi udev dhcp(eth0)
Bootsplash:     SuSE-SLES (800x600)
Groucho83-SLES9-SP2-iboot-fw:/ # mount /dev/sdc2 /mnt/
Groucho83-SLES9-SP2-iboot-fw:/ # cd /
Groucho83-SLES9-SP2-iboot-fw:/ # tar --exclude=/mnt -cf - / | (cd /mnt ; tar -xvpf - )
(This is to Replicate Linux to the iSCSI drive)

Groucho83-SLES9-SP2-iboot-fw:/ # cd /mnt/
Groucho83-SLES9-SP2-iboot-fw:/mnt # mkdir /mnt/
mkdir: cannot create directory `/mnt/': File exists
Groucho83-SLES9-SP2-iboot-fw:/mnt # kate
/mnt/etc/sysconfig/network/ifcfg-eth-id-00\:0d\:60\:5b\:99\:0e
BOOTPROTO='static'
MTU=''
REMOTE_IPADDR=''
STARTMODE='off'
#UNIQUE='...'
_nm_name='bus-pci-0000:05:01.0'
BROADCAST='2.1.7.255'
IPADDR='2.1.1.160'
NETMASK='255.255.248.0'
NETWORK='2.1.0.0'
Groucho83-SLES9-SP2-iboot-fw:/mnt # kate /mnt/etc/fstab
Label=iscsi_iboot    /                    reiserfs   acl,user_xattr        1 1
/dev/sda2            swap                 swap       pri=42                0 0
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
proc                 /proc                proc       defaults              0 0
usbfs                /proc/bus/usb        usbfs      noauto                0 0
sysfs                /sys                 sysfs      noauto                0 0
/dev/cdrom       /media/cdrom      subfs    fs=cdfss,ro,procuid,nosuid,nodev,exec,iocharset=utf8
0 0
You have new mail in /var/spool/mail/root
Groucho83-SLES9-SP2-iboot-fw:/mnt # ls
.  ..  .rnd  bin  boot  dev  etc  home  lib  media  opt  proc  root  sbin  srv  sys  tmp  usr  var
```

```
Groucho83-SLES9-SP2-iboot-fw:/mnt # cd /mnt/etc/rc.d/
Groucho83-SLES9-SP2-iboot-fw:/mnt/etc/rc.d # mv network network.org
Groucho83-SLES9-SP2-iboot-fw:/mnt/etc/rc.d # cd /mnt/etc/
Groucho83-SLES9-SP2-iboot-fw:/mnt/etc # cat initiatorname.iscsi
## DO NOT EDIT OR REMOVE THIS FILE!
## If you remove this file, the iSCSI daemon will not start.
## If you change the InitiatorName, existing access control lists
## may reject this initiator.  The InitiatorName must be unique
## for each iSCSI initiator.  Do NOT duplicate iSCSI InitiatorNames.
InitiatorName=iqn.2000-05.com.ibm.iboot.groucho83
Groucho83-SLES9-SP2-iboot-fw:/mnt/etc # cd /mnt/boot/
Groucho83-SLES9-SP2-iboot-fw:/mnt/etc # ls
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot # rm initrd
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot # ln -s initrd-iboot initrd

Install SuSE GRUB on the Target partition:
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot # cd /mnt/boot/grub/
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot/grub # cp device.map device.map.new
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot/grub # kate /mnt/boot/grub/device.map.new
(hd0)   /dev/sd<c>


Groucho83-SLES9-SP2-iboot-fw:/mnt/boot/grub # /usr/sbin/grub
--device-map=/mnt/boot/grub/device.map.new
GNU GRUB  version 0.94  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename. ]

grub> root (hd0,1)
 Filesystem type is reiserfs, partition type 0x83

grub> setup (hd0)
 Checking if "/boot/grub/stage1" exists... yes
 Checking if "/boot/grub/stage2" exists... yes
 Checking if "/boot/grub/reiserfs_stage1_5" exists... yes
 Running "embed /boot/grub/reiserfs_stage1_5 (hd0)"...  19 sectors are embedded.
succeeded
 Running "install /boot/grub/stage1 (hd0) (hd0)1+19 p (hd0,1)/boot/grub/stage2
/boot/grub/menu.lst"... succeeded
Done.

grub> quit
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot/grub # kate /mnt/boot/grub/menu.lst
# Modified by YaST2. Last modification on Tue Aug 23 09:33:47 2005


color white/blue black/light-gray
default 0
timeout 8
gfxmenu (hd0,1)/boot/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title Linux
    kernel (hd0,1)/boot/vmlinuz root=LABEL= iscsi_iboot83 vga=0x31a selinux=0 splash=silent
resume=/dev/sda3 elevator=cfq showopts InitiatorName=iqn.2000-05.com.ibm.iboot.test
DiscoveryAddress=2.1.1.55
    initrd (hd0,1)/boot/initrd

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe
    kernel (hd0,1)/boot/vmlinuz root=/dev/sda2 showopts ide=nodma apm=off acpi=off vga=normal
noresume selinux=0 barrier=off nosmp noapic maxcpus=0  3
    initrd (hd0,1)/boot/initrd
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot/grub # iscsi-ls
*******************************************************************************
        Cisco iSCSI Driver Version ... 4.0.188.18 ( 21-May-2004 )
*******************************************************************************
```

```
TARGET NAME              : iqn.1986-03.com.ibm.13238450.20050826105134.device-1
TARGET ALIAS             : Device-1
HOST NO                  : 1
BUS NO                   : 0
TARGET ID                : 1
TARGET ADDRESS           : 2.1.1.55:3260
SESSION STATUS           : ESTABLISHED AT Sat Aug 27 10:45:40 2005
NO. OF PORTALS           : 2
PORTAL ADDRESS 1         : 2.1.1.55:3260,1
PORTAL ADDRESS 2         : 3.1.1.200:3260,1
SESSION ID               : ISID 00023d000001 TSID 189
*********************************************************************************
Groucho83-SLES9-SP2-iboot-fw:/mnt/boot/grub # cd /
Groucho83-SLES9-SP2-iboot-fw:/ # umount /mnt/
Groucho83-SLES9-SP2-iboot-fw:/ # /etc/init.d/iscsi stop
Stopping iSCSI: sync umount sync iscsid                                              done
```