HP LeftHand SAN Solutions

Support Document

Application Notes

LeftHand Volumes with SUSE Linux iSCSI

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

LeftHand Volumes with SUSE Linux iSCSI

Novell's SUSE Linux Enterprise 9 and 10 both provide an embedded software iSCSI Initiator capable communicating to LeftHand volumes. In April 2005, the open source development of the linux-iscsi project ended with the combined efforts to focus on the openiscsi project and its next generation architecture. SUSE Linux 9 is based upon the linux-iscsi project and SUSE Linux 10 is based upon open-iscsi. Unfortunately, two unique methods for configuration must be followed for each operating system version. It is noted that inconsistencies can exist between code streams in the open source community versus code streams and patches included within a specific Linux distribution.

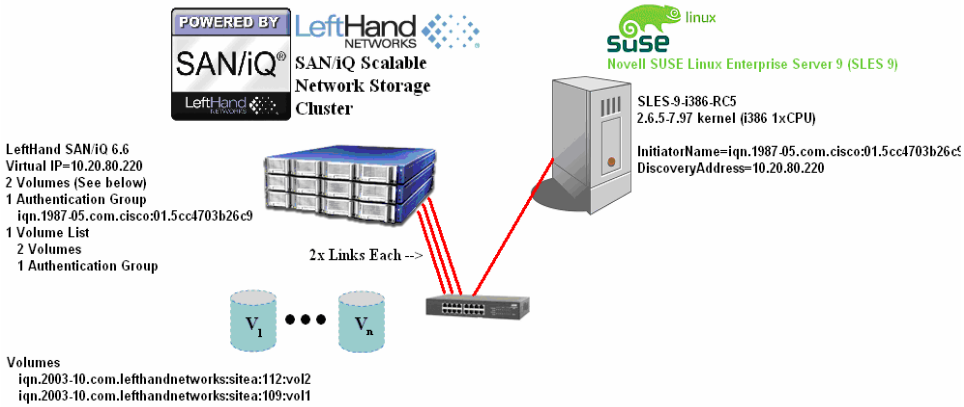
This document covers embedded iSCSI Initiators from SUSE Linux Enterprise 9 and 10 distributions only

Novell SUSE Linux Enterprise 9

Installing the SLES 9 Linux-iSCSI Initiator for use with LeftHand Volumes

This SLES 9 Section covers:

- Installing the linux-iscsi initiator via the YaST GUI
- Installing the linux-iscsi initiator via the terminal
- First-step configuration of the initiator
- Configuration highlights for the LeftHand SAN/iQ storage volume
- Configuring Persistent Bindings
- Preventing timing issues for raw devices
- iSCSI.conf File Documentation



Installing the linux-iscsi initiator via the YaST GUI

Novell SUSE Linux Enterprise Server 9 (SLES 9) was used for this document. Novell SUSE provides the Linux iSCSI Initiator via the YaST (Yet another Setup Tool) Install and Remove Software mechanism. The SUSE Linux Enterprise Server 9 CD3 will be required as the source.

Note that the http://linux-iscsi.sourceforge.net version 4.0.2 will only work on the 2.6.10 kernel. SLES 9 utilizes the 2.6.5-7 kernel.

Linux-iscsi-4.0.1-88.13.i586.rpm

Files

/etc/fstab.iscsi

/etc/init.d/iscsi

/etc/initiatorname.iscsi

/etc/iscsi.conf

/sbin/iscsi-device

/sbin/iscsi-id

/sbin/iscsi-iname

/sbin/iscsi-ls

/sbin/iscsi-mountall

/sbin/iscsi-umountall

/sbin/iscsid

/sbin/rciscsi

/usr/share/doc/packages/linux-iscsi

/usr/share/doc/packages/linux-iscsi/COPYING

/usr/share/doc/packages/linux-iscsi/README

/usr/share/man/man1/iscsi-ls.1.gz

/usr/share/man/man5/iscsi.bindings.5.gz

/usr/share/man/man5/iscsi.conf.5.gz

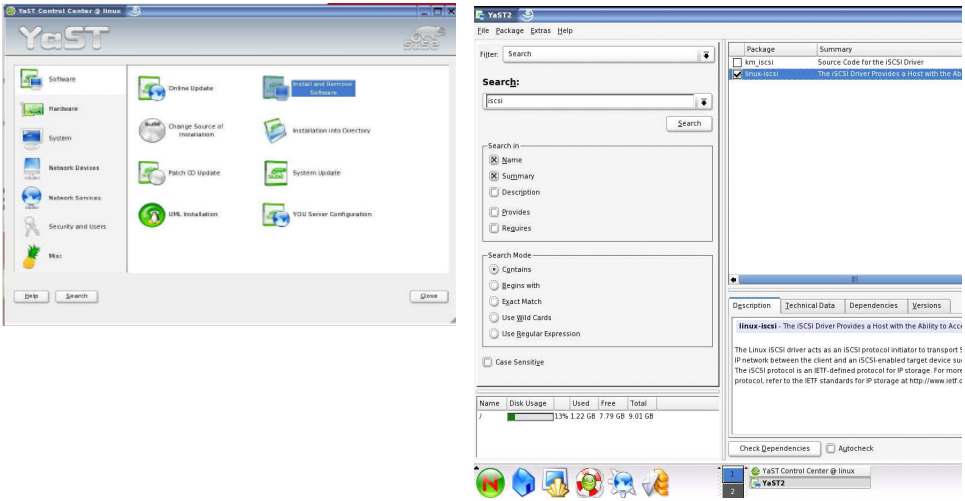
/usr/share/man/man8/iscsid.8.gz

/var/lib/iscsi

/var/lib/iscsi/bindings

Select the Red N to Browse and Start Applications Select System, YaST

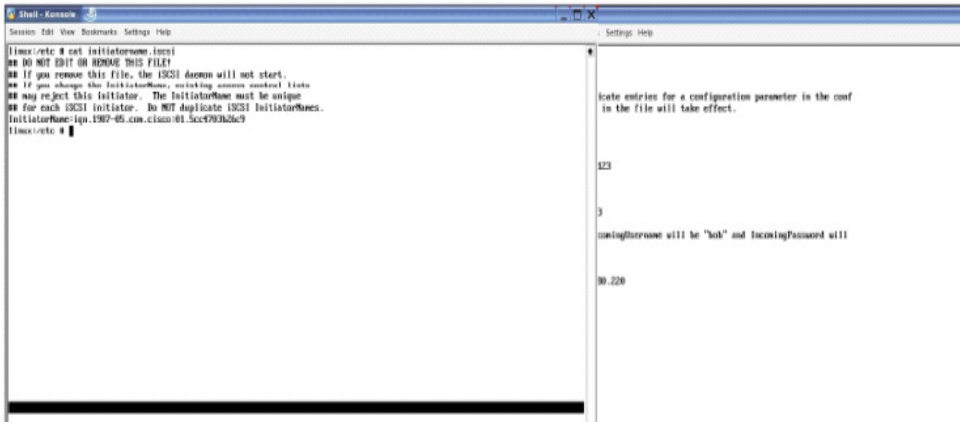
Yet another Setup Tool (YaST) is an operating system setup and configuration tool.



Under Software, Select Install and Remove Software. Search for iscsi. Select “linux-iscsi”. Note on the Technical Data tab the version is “linux-iscsi-4.1.1-88.13.i586.rpm”. Select Accept. Note that a dialog box will request the insertion of the SUSE Linux Enterprise Server 9 CD3.

Installing the linux-iscsi initiator via the terminal

The Novell SUSE Linux Enterprise Server 9 CD3 contains /suse/i586/ linux-iscsi-4.0.1-88.13.i586.rpm. Insert the CD, provide access to the cdrom. (mount /dev/cdrom /mnt/cdrom) Change directories to the location of the linux-iscsi installation file. (cd /mnt/cdrom/suse/i586) Install the linux-iscsi package. (rpm –install linux-iscsi-4.0.1-88.13.i586.rpm)



First-step configuration of the initiator

Once installation is complete, open a terminal console. Note that root equivalent access may be requi red for modifying the following configurationsettings.

Configure the iSCSI Initiator IQN Name in /etc/initiatorname.iscsi (vi may also be used)

A default iSCSI Initiator Node name will be pre-populated in the configuration file. This name must be unique on the iSCSI SAN. Modify the file or create a new file with the following command.

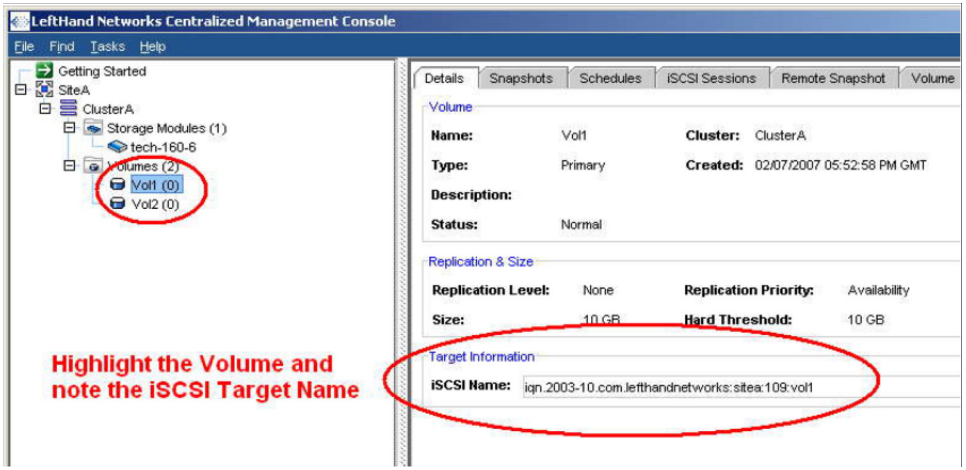
> echo “InitiatorName=iqn.1987-05.com.cisco:01.5cc4703b26c9” > /etc/initiatorname.iscsi

Configure the Discovery Address of the LeftHand SAN/iQ iSCSI Target Portal in /etc/iscsi.conf

> echo “DiscoveryAddress=10.20.80.220” > /etc/iscsi.conf

Global settings are configured prior to the first DiscoveryAddress, specific target CHAP settings are configured by setting options after the DiscoveryAddress and prior to the next DiscoveryAddress (if multiple target portals are defined).

Configuration highlights for the LeftHand SAN/iQ storage volume

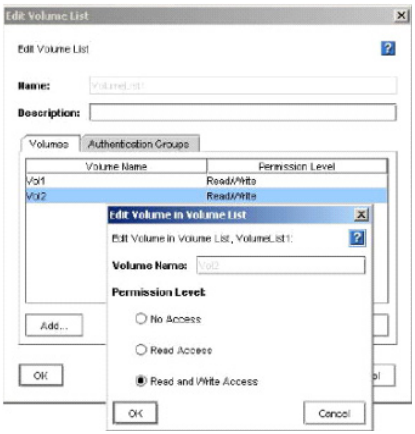


Note the iSCSI target information for the volumes in the LeftHand Central Management Console.

Create the authentication group and define the access list of authentication groups for each storage volume. The Linux iSCSI InitiatorName is written to the /etc/initiatorname.iscsi file. This must match the Initiator defined in the Authentication Group which will be assigned to the Volume list for access. Note that you are able to define and create one or two way CHAP authentication for the target (Target Secret and Initiator Secret) . If this is done, ensure that appropriate changes are made to the /etc/iscsi.conf file. (See below and Appendix A for further information.)

Each Volume List must match the group of the volumes which are to be accessed and the Authentication Groups (Initiators) accessing those volumes. This indirectly maps the Linux iSCSI Initiator and its defined Authentication Group to the Volumes which it will discover and have access to. You may define different permission levels restricting or providing access.

Securing access to LeftHand SAN/iQ volumes is achieved through Authentication Groups and CHAP. The Challenge-Handshake Authentication Protocol (CHAP) is a standard authentication protocol to validate the identity of remote iSCSI clients authenticating to their associated targets. In 1 -way CHAP, initiators must log in with a target secret to access the volume. This secret proves the identity of the initiator to the target. In 2 -way CHAP, initiators not only implement 1 -way, the target must also prove its identity to the initiator using the initiator secret.



Edit /etc/iscsi.conf below each DiscoveryAddress

; 1-way CHAP

OutgoingUsername=<CHAP Name> ; Matches Authentication Group CHAP Name

OutgoingPassword=<Target Secret> ; Matches Authentication Group Target Secret

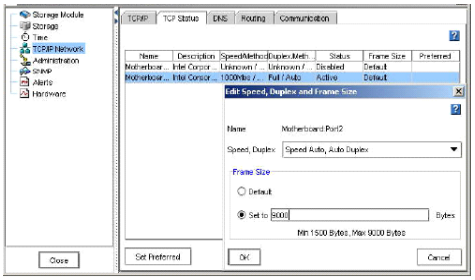
; 2-way CHAP in addition to 1-way above

IncomingUsername=<CHAP Name> ; Matches Authentication Group CHAP Name

IncomingPassword=<Initiator Secret> ; Matches Authentication Group Initiator Secret

Enabling jumbo frame support increases performance characteristics in most SANs. For reasons of backwards compatibility, the gigabit Ethernet standard allows the use of 1500 byte frames in the original 100 Mbit Ethernet specification, as well as jumbo frames of up to 9,000 bytes. The default of Ethernet equipment is usually shipped with the 1500 byte frame size to ensure the widest compatibility. Smaller frame sizes will require more frames in order to send the same amount of information relayed by jumbo frames. Since the overhead involved in handling fra mes could affect maximum potential performance, it is important to ensure that every point on the iSCSI SAN supports and has jumbo frames enabled. In the case of the a Linux iSCSI Initiator, this will include the appropriate driver load parameter (if applicable) for the gigabit Ethernet host adapter and/or the ifconfig MTU

value switch (consult your Ethernet Adapter driver documentation), the appropriate between end points switch capability and setting (if it is a managed switch), and the setting on the SAN/iQ target. Note that for the target, this must be configured on each SAN/iQ node for each Ethernet interface.

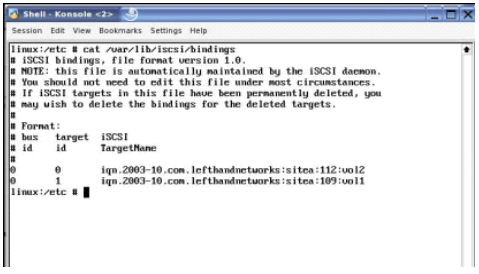


Within the LeftHand SAN/iQ Centralized Management Console, edit each node configuration, select TCP/IP Network on the left window pane and the TCP Status tab. Highlight each interface and select Edit. Change the Frame Size from radio selected Default to Set to and specify 9000 Bytes. Select OK and repeat for the other interface.

Repeat for each node in the storage cluster communicating to the Linux iSCSI Initiator.

Configuring Persistent Bindings

Persistent Bindings ensure that discovered targets are repeatedly enumerated as the same device across reboots. If this approach is not utilized, first login discovery will dictate the enumeration of iSCSI targets. Applications which rely upon consistent configuration of where data resides could fail.



Start the iSCSI service for discovered targets to be enumerated.

/etc/init.d/iscsi start

With SUSE's version of the linux-iscsi initiator, the iSCSI daemon will maintain a mapping of all discovered targets. This discovery mapping is found in /var/lib/iscsi/bindings.

Note that SLES 9 is preconfigured for udev to enumerate iSCSI devices based upon the TargetName.

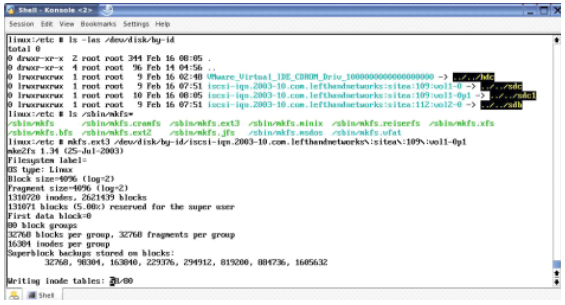
These mapping soft links can be found in /dev/disk/by-id.

List the devices. (ls –las /dev/disk/by-id)

Note the naming convention /dev/disk/by-id/iscsi-<Target IQN Name>-0. Discovered partitions will be enumerated in a similar manner, however, are not required for raw devices. For example, /dev/disk/by-id/iscsi-<Target IQN Name>-0p1 will map to the first partition.

Create a partition by addressing the udev enumeration. Note that you could use /dev/sdc for the first target, however, the intent is to abstract enumeration to the full targetname. (Tab completion may be used)

fdisk /dev/disk/by-id/
iscsi-iqn.2003-10.com.lefthandnetworks\:sitea\:109\:vol1-0



Create a new primary partition, write the table to disk and exit. Note the enumeration change for the newly created partition in /dev/disk/by-id/.

After creating a partition, a file system must be created on that partition. This example creates an ext3 file system. Note the other type of file systems available in the default SLES 9 installation.

mkfs.ext3 /dev/disk/by-id/
iscsi-iqn.2003-10.com.lefthandnetworks\:sitea\:109\:vol1-0p1

Note that this file system may now be mounted.

The /etc/fstab.iscsi configuration file defines the mount points for iSCSI devices. This file is checked at the start of the iscsi service.

Edit the /etc/fstab.iscsi configuration file adding the following line:

/dev/disk/by-id/iscsi-iqn.2003-10.com.lefthandnetworks:sitea:109:vol1-0p1 /mnt/iscsi1 ext3 acl,user_xattr 1 2

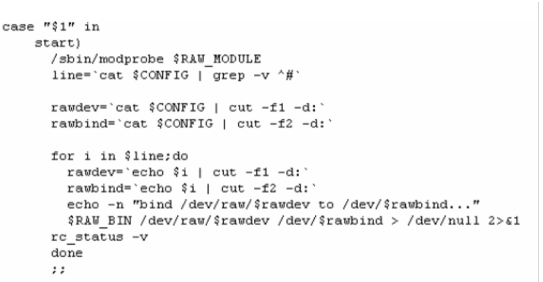
This will map the iSCSI Target Name to the mount point /mnt/iscsi1. The file system type is ext3.

"/etc/init.d/iscsi restart" restarts the service which enumerates this file. Note the volume is now mounted and accessed at /mnt/iscsi1. Repeat with any additional volumes.

Preventing Timing Issues for Raw Devices

Some applications may require direct support of an iSCSI target as a raw device. The raw devices script can start before the iSCSI initiator establishes access to all volumes. The mapping of raw devices may therefore encounter timing issues in presenting all configured targets. Several approaches solve this issue: The raw service may be started after a delay; The raw service may be started manually after the iSCSI service has completed login and discovery of all targets; Modification of the raw device service script could implement delays. The following prevents this by adding retry and timeout logic to the /etc/init.d/raw script.

Locate the section to the right in the /etc/init.d/raw device script file and replace the previous code with the following:



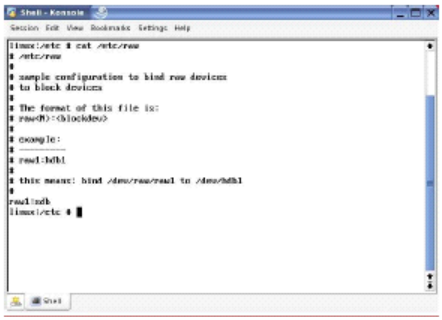

```
case "$1" in
   start)
     /sbin/modprobe
$RAW_MODULE
     line=`cat $CONFIG
| grep -v ^#`

     rawdev=`cat $CONFIG | cut -f1 -d:`
     rawbind=`cat $CONFIG | cut -f2 -d:`

     for i in $line;do
       rawdev=`echo $i | cut -f1 -d:`
       rawbind=`echo $i | cut -f2 -d:`
       retrycount=12
       while [ $retrycount -gt 0 ];do
         echo -n "bind /dev/raw/$rawdev to /dev/
         $rawbind..."
         $RAW_BIN /dev/raw/$rawdev /dev/$rawbind > /dev/
         null 2>&1
         if [ $? -ne 0 ]; then
           sleep 5
           retrycount=$((retrycount-1))
```

```
      echo "*** Waiting for iSCSI Initiator...,
      $retrycount "
    else
      break
    fi
  done
    rc_status -v
done
;;
```

Ensure that the raw service is configured to start at the same levels of the iscsi service.

/sbin/chkconfig --list iscsi raw ; Lists init levels of iscsi and raw

/sbin/chkconfig --level 345 raw on ; Enables raw to start at init levels 3,4 and 5



The /etc/raw configuration file will bind raw devices to block devices. This example will bind the /dev/sdb device (persistently mapped earlier) to the /dev/raw/raw1 rawdevice.

Useful commands supporting the iSCSI service:

Start the iSCSI service

/etc/init.d/iscsi start

Stopping the iSCSI service

/etc/init.d/iscsi stop

Report the currently connected devices if the iSCSI service is started.

/sbin/iscsi-ls

Close all open iSCSI sessions

/sbin/iscsi-kill-session

To add the iSCSI service to start automatically

/sbin/chkconfig --list	
/sbin/chkconfig --add iscsi ;	If iscsi is not listed, add the iSCSI service
/sbin/chkconfig --level 345 iscsi on	
/sbin/chkconfig --list iscsi ;	Verify that iSCSI is set appropriately
/sbin/chkconfig --list raw ;	Verify that all instances of raw devices matches iscsi
	; This is required to match raw devices expected support
/sbin/chkconfig --level 345 raw on	

iSCSI.conf File Documentation

EDITING THE ISCSI.CONF FILE

The /etc/iscsi.conf file is used to control the operation of the iSCSI driver by allowing the user to configure the values for a number of programmable parameters. These parameters can be setup to apply to specific configuration types or they can be setup to apply globally. The configuration types that are supported are:

- DiscoveryAddress = SCSI routing instance IP address with format a.d.c.d or a.b.c.d:n or hostname.
- TargetName = Target name in 'iqn' or 'eui' format
  e.g.: TargetName = iqn.1987 -05.com.cisco:00.0d1d898e8d66.t0
- Subnet = Network portal IP address with format a.b.c.d/n or a.b.c.d&hex

The complete list of parameters that can be applied either globally or to the configuration types listed above are shown below. Not all parameters are applicable to all configuration types. Some of the parameters are actual iSCSI login keys, and others are just configuration options for the initiator.

KEYS	DESCRIPTION	DEFAULT VALUE
OutgoingUsername	CHAP username used for initiator authentication by the target.	–
OutgoingPassword	CHAP password used for initiator authentication by the target.	–
IncomingUsername	CHAP username for target authentication by the initiator.	–
IncomingPassword	CHAP password for target authentication by the initiator.	–
HeaderDigest	Type of header digest support the initiator is requesting of the target.	prefer-off
DataDigest	Type of data digest support the initiator is requesting of the target.	prefer-off
SLPUnicast	Single IP address of the SLP service or directory agent.	–
SLPMulticast	Multicast IP addresses used to discover SLP service or directory agents.	–
PollInterval	Time interval between successive SLP queries sent out.	5 min

KEYS	DESCRIPTION	DEFAULT VALUE
LoginTimeout	Time interval to wait for a response to a login request to be received from a target before failing a connection attempt.	15 sec
AuthTimeout	Time interval to wait for a response to a login request containing authentication information to be received from a target before failing a connection attempt.	45 sec
IdleTimeout	Time interval to wait for on a connection before sending a ping when there are active tasks in the session.	60 sec
ActiveTimeout	Time interval to wait for a ping response after a ping is sent before failing a connection.	5 sec
PingTimeout	Time interval to wait for a ping response after a ping is sent before failing a connection.	5 sec
ConnFailTimeout	Time interval to wait before failing SCSI commands back to an application for unsuccessful commands.	0 sec

KEYS	DESCRIPTION	DEFAULT VALUE
AbortTimeout	Time interval to wait for a abort command to complete before declaring the abort command failed.	10 sec
ResetTimeout	Time interval to wait for a reset command to complete before declaring the reset command failed.	30 sec
Enabled	Enable/Disable a target or list of targets.	yes
InitialR2T	Enabling/disabling of R2T flow control with the target.	no
ImmediateData	Enabling/disabling the sending of unsolicited data burst with the iSCSI command PDU.	yes
MaxRecvDataSegment	Maximum number of bytes that the initiator can receive in an iSCSI PDU.	128K
FirstBurstLength	Maximum number of bytes of unsolicited data the initiator is allowed to send.	Length 256K
MaxBurstLength	Maximum number of bytes for the SCSI payload negotiated by initiator.	16M

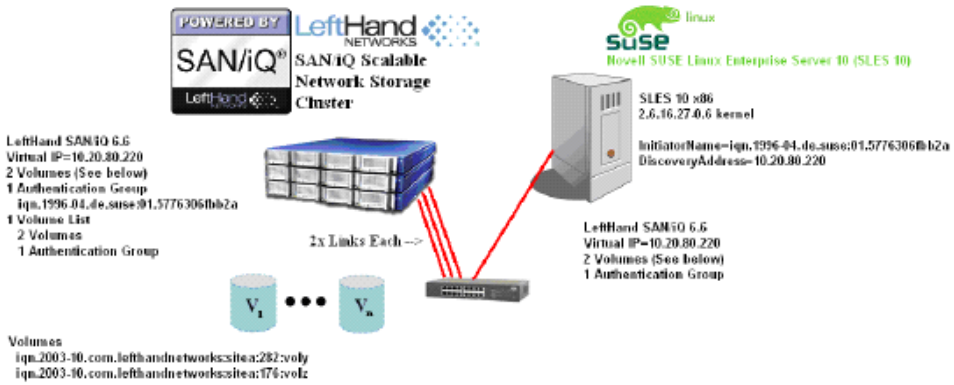
KEYS	DESCRIPTION	DEFAULT VALUE
TCPWindowSize	Maximum number of bytes that can be sent over a TCP connection by the initiator before receiving an acknowledgement from the target.	256K
Continuous	Enabling/disabling the discovery session to be kept alive. If Continuous = "yes" and a "Reject(0x3f)" message is received from the target in response to the ping (NOP_OUT) message sent by the initiator, the Continuous parameter will be set to "no" and no further ping messages will be sent by the initiator.	yes
SendAsyncText	Enable/disable sending "X" keys related to Async events; Cisco target only.	yes

A detailed description for each of these parameters is included in both the man page and the included sample iscsi.conf file. Please consult these sources for examples and more detailed programming instructions.

Installing the SLES 10 open-iSCSI Initiator for use with LeftHand Volumes

This SLES 10 Section covers:

- Installing the open-iscsi initiator via the YaST GUI
- Installing the open-iscsi initiator via the terminal
- Configuration of the initiator and persistent bindings
- Configuration highlights for the LeftHand SAN/iQ storage volume



Installing the open-iscsi initiator via the YaST GUI

Novell SUSE Linux Enterprise Server 10 (SLES 10) 2006-07-05 Release DVD was used for this document. Novell SUSE provides the Linux iSCSI Initiator via the YaST (Yet another Setup Tool) Install and Remove Software mechanism. The SUSE Linux Enterprise Server 10 DVD will be required as the source.

Note that the http://www.open-iscsi.org version is not covered in this document. SLES 10 includes open -iscsi-0.5.545-9.12.

Open-iscsi-0.5.545-9.12.i586.rpm

Files

/etc/init.d/boot.open-iscsi

/etc/init.d/open-iscsi

/etc/iscsid.conf

/sbin/iscsi-iname

/sbin/iscsiadm

/sbin/iscsid

/sbin/iscsistart

/sbin/rcopen-iscsi

/usr/share/doc/packages/open-iscsi

/usr/share/doc/packages/open-iscsi/COPYING

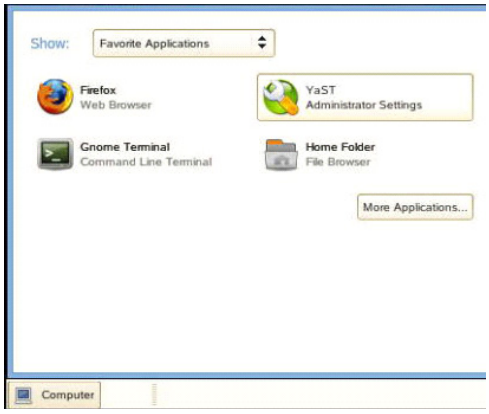
/usr/share/doc/packages/open-iscsi/README

/usr/share/man/man8/iscsiadm.8.gz

/usr/share/man/man8/iscsid.8.gz

/var/adm/fillup-templates/sysconfig.open-iscsi

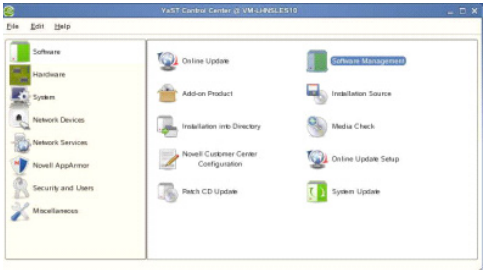
/var/lib/open-iscsi



Select the Computer on the Task Bar

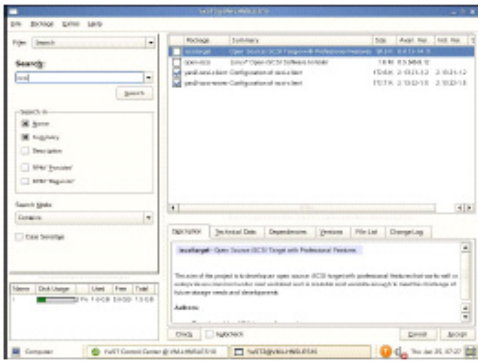
Select YaST Administrator Settings

Yet another Setup Tool (YaST) is an operating system setup and configuration tool.



Under Software, select Software Management.

Search for iscsi. Select “open-iscsi”. Select Accept. Note that a dialog box will request the insertion of the SUSE Linux Enterprise Server 10 DVD. Note the version as 0.5.545-9-12.



Installing the open-iscsi initiator via the terminal

The Novell SUSE Linux Enterprise Server 10 DVD contains /suse/i586/ open-iscsi-0.5.545-9.12.i586.rpm. Insert the DVD, if not auto mounted, provide access to the cdrom. (mount /dev/cdrom /media/cdrom) Change directories to the location of the open-iscsi installation file. (cd /media/cdrom/suse/i586) Install the open-iscsi package. (rpm --install open-iscsi-0.5.545-9.12.i586.rpm)

Configuration of the initiator and persistent bindings

Once installation is complete, open a terminal console. Note that root equivalent access may be required for modifying the following configuration settings.

Ensure that the open-iscsi service is started.

```
/etc/init.d/open-iscsi status
```

The iSCSI initiator service should be “running” for configuration. If the service is not running, start the service

```
/etc/init.d/open-iscsi start
```

The open-iscsi service should start in all appropriate init levels.

```
/sbin/chkconfig --list open-iscsi
open-iscsi                    0:off 1:off 2:off 3:on 4:off 5:on 6:off
/sbin/chkconfig --level 345 open-iscsi on
/sbin/chkconfig --list open-iscsi
open-iscsi                    0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

Configure the iSCSI Initiator IQN Name in /etc/initiatorname.iscsi (vi may also be used).

A default iSCSI Initiator Node name will be pre-populated in the configuration file. This name must be unique on the iSCSI SAN. Modify the file or create a new file with the following command.

echo “InitiatorName= iqn.1996-04.de.suse:01.5776306fbb2a” > /etc/initiatorname.iscsi

Differing from previous iscsi initiators, open-iscsi configures its discovery database and nodes in a database table, /var/db/iscsi/discover.db and /var/db/iscsi/node.db. The iscsiadm utility is the tool for populating entries and interfacing into the discovery and node databases.

```
supportnode1:/etc # cat ./iscsid.conf
#
# Open-iSCSI default configuration.
# Could be located at /etc/iscsid.conf or ~/.iscsid.conf
#
node.active_cnx = 1
node.startup = automatic
#node.session.auth.username = dima
#node.session.auth.password = aloha
node.session.timeo.replacement_timeout = 120
node.session.err_timeo.abort_timeout = 10
node.session.err_timeo.reset_timeout = 30
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.session.iscsi.DefaultTime2Wait = 0
node.session.iscsi.DefaultTime2Retain = 0
node.session.iscsi.MaxConnections = 0
node.conn[0].iscsi.HeaderDigest = None
node.conn[0].iscsi.DataDigest = None
node.conn[0].iscsi.MaxRecvDataSegmentLength = 65536
#discovery.sendtargets.auth.authmethod = CHAP
#discovery.sendtargets.auth.username = dima
#discovery.sendtargets.auth.password = aloha
```

The default configuration file is /etc/iscsid.conf.

This file contains default configuration information populated for each node which may be overwritten by iSCSI Discovery or manual updates through the iscsiadm utility.

Changing any of these values will only have affect onnewly discovered nodes not previously found in the database. After a node is added to the database, any of its node properties may be changed with iscsiadm which will have effect only at the next target login.

Note that node.startup has been changed from manual to automatic. Use an editor to change this value for new nodes added to the database.

The first step is to add the discovery portal where the targets will found. This will be the Virtual IP address found in the LeftHand Centralized Management Console (CMC). Note that targets advertised through the sendtargets will still require appropriate access controls defined through the LeftHand CMC Volume List containing SLES 10 accessible volumes and the access group containing the SLES 10 Initiator IQN. (Alternate methods include 1 or 2-way CHAP Authentication)

iscsiadm -m discovery -t sendtargets -p 10.20.80.220:3260

[7b4943]
10.20.80.220:3260,1 iqn.2003-10.com.lefthandnetworks:sitea:282:voly

[bb4925]
10.20.80.220:3260,1 iqn.2003-10.com.lefthandnetworks:sitea:176:volz

In this particular example, only 2 targets are defined within the LeftHand CMC. Note that record ID 7b4943 contains an entry for a target discovered at 10.20.80.220:3260 with an iSCSI target IQN of iqn.2003-10.com.lefthandnetworks:sitea:282:voly. Record ID bb4925 contains an entry for a target discovered at 10.20.80.220:3260 with an iSCSI target IQN of iqn.2003-10.com.lefthandnetworks:sitea:176:volz.

Both of these targets are now added to the node database. List all targets found in the node database.

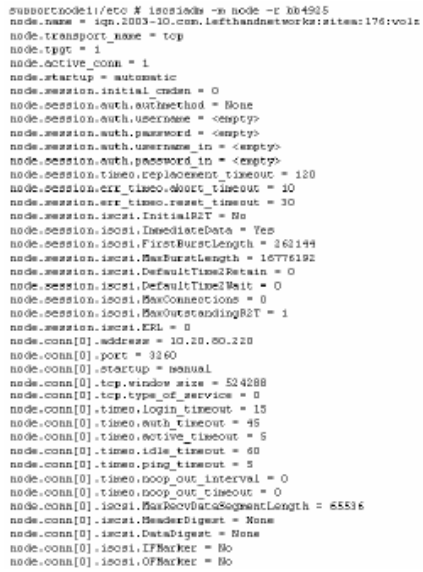
iscsiadm -m node

[7b4943]
10.20.80.220:3260,1 iqn.2003-10.com.lefthandnetworks:sitea:282:voly

[bb4925]
10.20.80.220:3260,1 iqn.2003-10.com.lefthandnetworks:sitea:176:volz

List the properties of a specific node.

iscsiadm –m node –r bb4925



Note that the record number is used to reference the specific node entry. Also note that the in addition to default properties, the properties defined in /etc/iscsid.conf have populated the node entry. If iscsid.conf is not present, default entries will still populate the node properties. Any of these values may be modified in the database by using the iscsiadm utility.

iscsiadm –m node –r bb4925 –o update –n node.conn[0].startup –v automatic

This command will update node record bb4925 to automatically login to the target upon the service start. The change may be verified by listing the properties of the node again.

Note that node.conn[0].startup = automatic may also be added to the /etc/iscsid.conf file to affect any additional nodes that are added to the database.

Repeat this for the second node in the database 7b4943.

iscsiadm –m node –r 7b4943 –o update –n node.conn[0].startup –v automatic

Since these entries were performed after population of the database, you may either restart the open-iscsi service or manually login to each target with iscsiadm utility. (“--login” is equivalent to “-l”)

iscsiadm –m node –r bb4925 --login

iscsiadm –m node –r 7b4943 --login

Obtain the status of logged in targets and the open-iscsi service.

/etc/init.d/open-iscsi status

Checking for iSCSI initiator service: running

Active connections:

iqn.2003-10.com.lefthandnetworks:sitea:282:voly

iqn.2003-10.com.lefthandnetworks:sitea:176:volz

If an error is returned attempting to login and the connections are not seen in status or in the LeftHand CMC, restart the service.

/etc/init.d/open-iscsi restart

Stopping iSCSI initiator service: done

Starting iSCSI initiator service: done

Logging into iqn.2003-10.com.lefthandnetworks:sitea:282:voly done

Logging into iqn.2003-10.com.lefthandnetworks:sitea:176:volz done

Targets are enumerated as devices by several methods: by-path, by-id, and by-uuid.

ls –las /dev/disk/by-path

ls –las /dev/disk/by-id

ls –las /dev/disk/by-uuid

By-path enumerates the targets by the IQN. It is recommended to mount by the path for persistent binding of that volume to known mount points. By-uuid enumerates the targets by a Universally Unique Identifier. Note that rawdevices and some partition types may not implement UUIDs.

Now that the iSCSI target shows up as a device, the target may be partitioned with fdisk.

```
fdisk /dev/disk/by-path/
ip-10.20.80.220\:3260-iscsi-iqn.2003-10.com.lefthandnetworks\:sitea\:28
2\:voly
```

Note that the ":" must be represented by "\:". Tab completion may be used for populating the target enumeration. The new path for voly partition 1 is

```
/dev/disk/by-path/
ip-10.20.80.220:3260-iscsi-iqn.2003-10.com.lefthandnetworks:sitea:282:vo
ly-part1
```

This example will show creation of an ext3 file system on the target.

```
mkfs.ext3 /dev/disk/by-path/
ip-10.20.80.220\:3260-iscsi-iqn.2003-10.com.lefthandnetworks\:sitea\:28
2\:volypart1
```

Repeat with all targets.

A mount point may be created (mkdir /mnt/voly /mnt/volz) and the devices mounted.

```
mount /dev/disk/by-path/
ip-10.20.80.220\:3260-iscsi-iqn.2003-10.com.lefthandnetworks\:sitea\:28
2\:voly-part1/mnt/voly
```

Automatic mounting is still handled by /etc/fstab, however some problems will exist if the volume is entered as usual. The system will attempt to mount the volume at boot time before networking is up which will present a problem since iSCSI volumes are located on the network. Since the SLES distribution system removed the netfs script, fstab mount options like _netdev will no longer work. SUSE will require the hotplug functions controlled by the HAL system to support iSCSI volumes in the /etc/fstab file.

Open the /etc/fstab file.

```
vi /etc/fstab
```

Add the following entry to a separate line in the fstab file. Examine the man entry for additional information.

```
/dev/disk/by-path/
ip-10.20.80.220:3260-iscsi-iqn.2003-10.com.lefthandnetworks:sitea:282:vo
ly-part1 /mnt/voly ext3 hotplug,rw 0 2
```

The last configuration requirement for persistent binding at boot time will require modification of the open-iscsi start/stop script. A small delay at service startup has been found to be required to ensure connections are restored at boot time.

Open the /etc/init.d/open-iscsi file.

```
vi /etc/init.d/open-iscsi
```

Immediately before the declaration of variables, add a sleep command for 10 seconds. Note that some configurations

may require additional time, some less.

```
…
### END INIT INFO
sleep 10
PID_FILE=/var/run/iscsi.pid
…
```

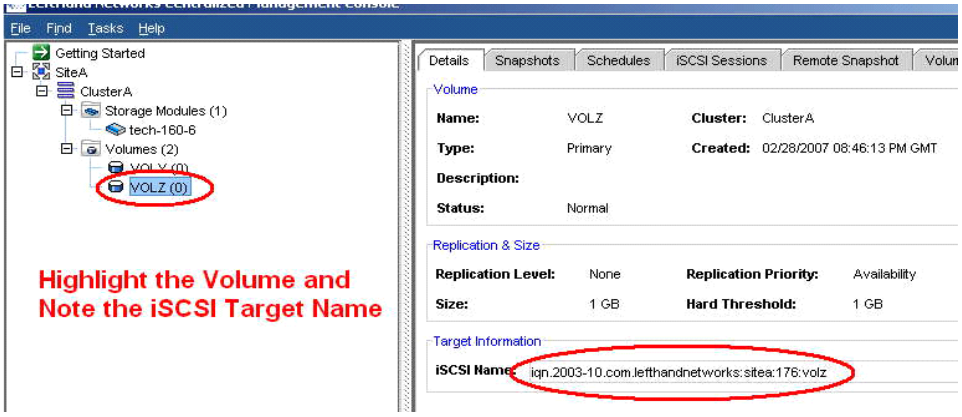
In addition, some configurations may require a delay in the iscsi_login_all_nodes() function. Add a sleep command for

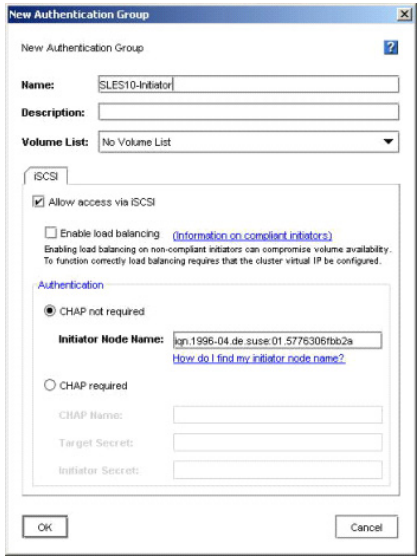
4 seconds further down the file.

```
…
iscsi_login_all_nodes()
{
sleep 4
TARGETS=$($ISCSIADM -m node 2> /dev/null | sed 's@\[\(.*\)\]
*@\1@g')
…
```

A reboot of the system should now verify that automatic logging in to the targets is successful and that the target is mounted appropriately through the hotplug system declared in the fstab file.

Configuration highlights for the LeftHand SAN/iQ storage volume





Note that in this configuration 2 volumes were created VOLY and VOLZ.

Note that an Authentication Group for the SLES10 Initiator is also created. Although CHAP could have been defined and then added for each entry with either the /etc/iscsid.conf file or manually entered for each target with the iscsiadm utility, only Initiator Node Name authentication will be used in this example. For chap support, iscsiadm must edit the following values which will correspond to CHAP Name, Target Secret and Initiator Secret found here in the CMC.

node.session.auth.authmethod = CHAP

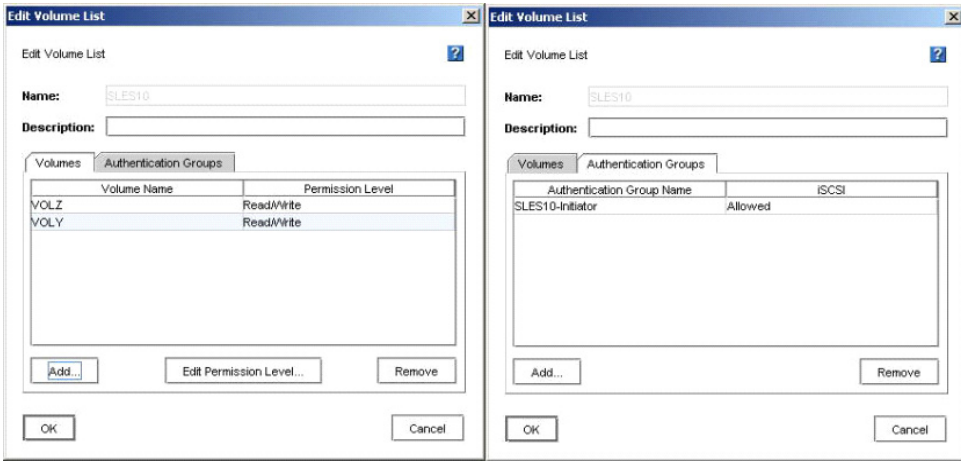
node.session.auth.username

node.session.auth.password

node.session.auth.username_in

node.session.auth.password_in

Note that the Initiator Node Name must match the value that is found in the /etc/initiatorname.iscsi file.



A Volume List provides access to the Volumes with the appropriate Authentication Groups. This configuration shows a SLES10 volume list with the VOLZ and VOLY volumes for both Read/Write to the Initiator defined in the SLES10-Initiator Authentication Group.