



iSCSI Software User Guide for Linux

iSCSI Software Release Version 5

www.chelsio.com

Copyright © 2007 ~ 2015 by Chelsio Communications, Inc., 370 San Aleso Ave, Suite 100, Sunnyvale, CA 94085, U.S.A.
All rights reserved.

This document and related products are distributed under licenses restricting their use, copying, distribution, and reverse-engineering. No part of this document may be reproduced in any form or by any means without prior written permission by Chelsio Communications.

S3xx, N3xx, T3xx, S4xx, N4xx, T4xx and T5xx are trademarks of Chelsio Communications, Inc.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE USE OF THE SOFTWARE AND ANY ASSOCIATED MATERIALS (COLLECTIVELY THE "SOFTWARE") IS SUBJECT TO THE SOFTWARE LICENSE TERMS OF CHELSIO COMMUNICATIONS, INC.

Contents

1	<i>Introduction</i>	1
	Features.....	1
2	<i>Hardware Requirements</i>	3
	Adapter Requirements.....	3
	Storage Requirements.....	3
3	<i>Software Requirements</i>	4
	About the End User License Agreement (EULA).....	4
	Linux Requirements.....	4
	Base Adapter Driver Installed First.....	4
	TOE Installation.....	4
	Requirements for Installing the iSCSI Software.....	4
4	<i>Software Installation</i>	5
	Getting the Latest Underlying NIC Driver Software.....	5
	Regular NIC Driver (non-Chelsio).....	5
	Regular NIC Driver (Chelsio N-Series).....	5
	TOE Driver (Chelsio S-Series).....	5
	Getting the Latest iSCSI Software Stack Driver Software.....	6
	Compiling the iSCSI Software.....	6
	Installing the iSCSI Modules/Tools.....	8
5	<i>Software Configuration</i>	10
	Overview.....	10
	Command Line Tools.....	10
	iscsictl.....	10
	chisns.....	10
	iSCSI Configuration File.....	10
	“On the fly” Configuration Changes.....	11
	A Quick Start Guide for Target.....	11
	A Sample iSCSI Configuration File.....	11
	Basic iSCSI Control.....	12
	The iSCSI Configuration File.....	13
	Chelsio System Wide Global Entity Settings.....	13

Description	14
Table of Chelsio Global Entity Settings	14
iSCSI Entity Settings	16
Description	16
Table of iSCSI Entity Settings	16
Chelsio Entity Settings	18
Description	18
Table of Chelsio Entity Settings	18
Sample iSCSI Configuration File	24
Challenge-Handshake Authentication Protocol (CHAP)	25
Normal Session CHAP Authentication	25
Oneway CHAP authentication	25
Mutual CHAP authentication	25
Adding CHAP User ID and Secret	26
Discovery Session CHAP	26
Target Access Control List (ACL) Configuration	26
ACL Enforcement	27
Target Storage device Configuration	27
RAM Disk Details	28
FILE Mode Storage Device Details	28
Example Configuration of FILE Mode Storage	29
BLK Mode Storage Device Details	29
Multi-path Support	29
Target Redirection Support	30
ShadowMode for Local vs. Remote Redirection	30
Redirecting to Multiple Portal Groups	30
The Command Line Interface Tools “iscsictl” and “chisns”	32
iscsictl	32
chisns	32
iscsictl options	32
chisns options	36
Rules of Target reload (i.e. “on the fly” Changes)	36
System Wide Parameters	37
iscsi_login_complete_time	37
iscsi_auth_order	37
iscsi_target_vendor_id	38
iscsi_chelsio_ini_idstr	38
6 Software Compliance	39
7 Customer Support	40

1 INTRODUCTION

Thank you for choosing Chelsio Communications, Inc. (“Chelsio”) as the provider of iSCSI software driver suite in your storage environment. This document describes how to install and configure this software for use as a key element in your iSCSI SAN. The software works on Linux-based systems that use non-Chelsio based Ethernet adapters or for performance enhanced systems it is optimized for use with systems equipped with Chelsio’s Ethernet adapters. Chelsio’s adapters include offerings that range from stateless offload adapters (regular NIC) to the full line of TCP/IP Offload Engine (TOE) adapters.

The Chelsio iSCSI software implements RFC 3720, the iSCSI standard of the IETF. The software has been fully tested for compliance to that RFC and others and it has been exhaustively tested for interoperability with the major iSCSI vendors.

The software is delivered and is executed as one set of code for target. The software implements most of the iSCSI protocol in software running in kernel mode on the host with the remaining portion, which consists of the entire fast data path, in hardware when used with Chelsio’s TOE adapters. When standard NIC Adapters are used the entire iSCSI protocol is executed in software.

The performance of this iSCSI stack is outstanding and when used with Chelsio’s hardware it is enhanced further. Because of the tight integration with Chelsio’s TOE adapters, this software has a distinct performance advantage over the regular NIC. The entire solution, which includes this software, Chelsio TOE hardware, an appropriate base computer system – including a high end disk subsystem, has industry leading performance. This can be seen when the entire solution is compared to others based on other technologies currently available on the market in terms of throughput and IOPS.

The following are the currently shipping Chelsio Adapters that are compatible with this iSCSI software:

- **T3 ASIC based Adapters** (S3xx and N3xx series of products)
- **T4 ASIC based Adapters** (S4xx and N4xx series of products)
- **T5 ASIC based Adapters** (T5xx series of products)

In the above list, all but the N-series of adapters are TOE capable.

This product should provide a reliable, high-performance solution for your storage needs.

FEATURES

Chelsio’s iSCSI driver stack supports the iSCSI protocol in the Target mode. From henceforth “iSCSI Software Entity” term refers to the iSCSI target.

The Chelsio iSCSI software provides the following high level features:

- Expanded NIC Support
 - Chelsio TCP Offload Engine (TOE) Support
 - T3 Based HBAs (S3xx Series cards)
 - T4 Based HBAs (S4xx Series cards)
 - T5 Based HBAs (T5xx Series cards)
 - Chelsio regular NIC – (non-TOE)
 - N Series (N3xx, N4xx)
 - Non-Chelsio
 - Runs on regular NICs

- From wireless through 10 GigE links
- Chelsio Terminator ASIC Support
 - Offloads iSCSI Fast Data Path with Direct Data Placement (DDP)
 - Offloads iSCSI Header and Data Digest Calculations
 - Offload Speeds at 1 Gb, 10 Gb and 40Gb
 - Offloads TCP/IP for NAS simultaneously with iSCSI
- Target Specific features
 - Full compliance with RFC 3720
 - Error Recovery Level 0 (ERL 0)
 - CHAP Support for both discovery and login chap including Mutual Authentication
 - Internet Storage Name Service (iSNS) Client
 - Target Access Control List (ACL)
 - Multiple Connections per Session
 - Multiple Targets
 - Multiple LUNs per Target
 - Multi Path I/O (MPIO)
 - Greater than 2 TByte Disk Support
 - Reserve / Release for Microsoft Cluster© Support
 - Persistent Reservation
 - Dynamic LUN Resizing
 - iSCSI Target Redirection
 - Multiple Target device types
 - Block
 - Virtual Block (LVM, Software RAID, EVMS, etc.)
 - Built in RAM Disk
 - Built in zero copy RAM Disk
 - Supports iSCSI Boot Initiators
 - An Intuitive and Feature Rich Management CLI

This user's guide will cover these features in detail.

2 HARDWARE REQUIREMENTS

ADAPTER REQUIREMENTS

The Chelsio iSCSI software can be used with or without hardware protocol offload technology. When used with protocol offload, a Chelsio TOE adapter must be used. When used with a regular NIC without offload, either one of the N-series adapters from Chelsio or a non-Chelsio NIC adapter can be used.

There are four modes of operation using the Chelsio iSCSI software on Ethernet-based adapters:

- **Regular NIC** – (Uses non-Chelsio or Chelsio's N-Series adapters). The Chelsio software can be used in non-offloaded (regular NIC) mode. Please note however that this is the least optimal mode of operating the software in terms of performance.
- **iSCSI HW Acceleration** – (Uses Chelsio's S Series adapters) In addition to offloading the TCP/IP protocols in hardware (TOE), this mode also takes advantage of Chelsio's ASIC capability of hardware assisted iSCSI data and header digest calculations as well as using the direct data placement (DDP) feature.

STORAGE REQUIREMENTS

When using the Chelsio iSCSI target a minimum of one hardware storage device is required. This device can be any of the device types that are supported (block, virtual block, RAM disk). Multiple storage devices are allowed by configuring the devices to one target or the devices to multiple targets. The software allows multiple targets to share the same device but use caution when doing this.

Chelsio's implementation of the target iSCSI stack has flexibility to accommodate a large range of configurations. For quick testing, using a RAM Disk as the block storage device works nicely. For deployment in a production environment a more sophisticated system would be needed. That typically consists of a system with one or more storage controllers with multiple disk drives attached running software or hardware based RAID.

3 SOFTWARE REQUIREMENTS

ABOUT THE END USER LICENSE AGREEMENT (EULA)

Before installing and using the Chelsio iSCSI software please read and agree to the End User License Agreement (EULA). It can be found in the distribution package.

LINUX REQUIREMENTS

The Chelsio iSCSI software runs on Linux-based platforms and therefore it is a base requirement for running the software. This software must run on recent versions of 2.6 of Linux. That includes full support of RHEL5, RHEL6, SLES10, and SLES11 distributions. See the next section on software installation for a list of specific kernel and distribution versions.

BASE ADAPTER DRIVER INSTALLED FIRST

As previously noted, the software can run on a regular NIC (the N series from Chelsio or others) or a TOE based adapter from Chelsio. In either case the underlying adapter driver must be installed first.

TOE Installation

Driver installation differs between Chelsio TOE adapter cards (S-series) and other regular NIC adapters such as Chelsio's N-series. The underlying TOE driver is normally module based but it can be built into the kernel. Please refer to the installation guide for instruction on installing and configuring the specific adapter that is being used.

REQUIREMENTS FOR INSTALLING THE iSCSI SOFTWARE

When installing the iSCSI software, it is required that the system have Linux kernel source or its headers installed in order to compile the iSCSI software as a kernel module. The source tree may be only header files, as for RHEL5 as an example, or a complete tree. The source tree needs to be configured and the header files need to be compiled. Additionally, the Linux kernel must be configured to use modules.

4 SOFTWARE INSTALLATION

There are three main steps to installing the Chelsio iSCSI software. They are:

1. **Installing the underlying Ethernet adapter driver** – This is generally covered in the user guide of the Ethernet adapter that is being used. Below is a small section on how to obtain the drivers required for the adapter that is being used.
2. **Installing the iSCSI software** – The majority of this section deals with how to install the iSCSI software.
3. **Configuring the iSCSI software** – Information on configuring the software can be found in a section further into this user's guide.

GETTING THE LATEST UNDERLYING NIC DRIVER SOFTWARE

An underlying NIC driver is required before the iSCSI software will work. Depending on the NIC used, the NIC driver can be from Chelsio or from another manufacturer.

Regular NIC Driver (non-Chelsio)

Please refer to the manufacturer of the non-Chelsio NIC Adapter for installation of the driver.

Regular NIC Driver (Chelsio N-Series)

The underlying Linux regular NIC driver for Chelsio's N-Series adapters can be downloaded from our support website at www.chelsio.com/support.

Additionally, the S-Series adapters can generally run as a regular NIC if explicitly configured to do so. Many of the newer Linux distributions such as those from RedHat and Novell ship with a T3 based driver. The newer kernel.org distribution too includes a Chelsio driver. All of these drivers work as a NIC only with the S-Series adapters and they also work with the N-Series adapters. Contact Chelsio support if you have questions about this.

TOE Driver (Chelsio S-Series)

To take advantage of iSCSI offload or just pure TCP/IP offload (TOE) with Chelsio's S-Series adapters, the underlying TOE driver must be installed first. These drivers can be downloaded from our support website at www.chelsio.com/support.

Currently the underlying TOE driver is available at the above website for the following Linux versions. Please check with Chelsio for updates as this list will be expanded over time.

Terminator 3, 4 and 5 ASIC (S-Series adapters)

- Kernel.org Linux Kernels 2.6.18+
- Kernel.org Linux Kernels 3.0+
- RHEL5, RHEL6, RHEL7
SLES11, SLES12

GETTING THE LATEST iSCSI SOFTWARE STACK DRIVER SOFTWARE

Obtaining the iSCSI software stack is typically done as a download from the Chelsio support website (www.chelsio.com/support). The license key must be obtained through interaction with the Chelsio sales channel and/or the support organization. Please contact sales@chelsio.com or support@chelsio.com for more information.

The iSCSI software is available for use with most installations of the Linux kernel version 2.6. The software is dependent on the underlying NIC adapter driver and thus the limitation on what version of the 2.6 Linux kernel it can run on is mostly dependent on the NIC driver's limitations.

COMPILING THE iSCSI SOFTWARE

The software is distributed in a compressed tar package and is comprised of user space and kernel space libraries and source code.

In order to compile the Chelsio iSCSI software, a configured and compiled Linux kernel source tree is required. Additionally, the `/lib/modules` must have been set up for this particular kernel (i.e. "make modules_install" has been run with the Linux kernel source tree).

For users running RHEL make sure the kernel-devel package is installed. It includes the kernel header files necessary for building a third-party kernel module. To install the kernel-devel package, run the following command:

```
[chelsio@]# up2date kernel-devel
Or
[chelsio@]# yum install kernel-devel
```

Follow the steps below for building the iSCSI software in preparation for use.

1. Untar the Chelsio iSCSI package:

```
[chelsio@]# tar -xzf <chelsio iscsi source tar ball.tar.gz>
Or
[chelsio@]# tar -xjf <chelsio iscsi source tar ball.tar.bz2>
[chelsio@]# cd <chelsio iscsi src>
```

2. Compile the Chelsio iSCSI package:

Note the following cases when using the make command during this build step.

- **Case 1:**
If Chelsio's T4 or T5 TOE driver (i.e., ChelsioUwire package) for the T4 or T5 S-Series cards is installed, then an extra option `CXGB4TOE_SRC=<cxgb4 src directory>` is needed immediately after the make:

```
make CXGB4TOE_SRC=<cxgb4toe_source_dir>
```

Example:

The ChelsioUwire package used is ChelsioUwire-2.1.0.10.tar.gz and is uncompressed under /usr/src, the cxgb4 source directory would be /usr/src/ChelsioUwire-2.1.0.10/src/network, the make command would be:

```
"make CXGB4TOE_SRC=/usr/src/ChelsioUwire-1.1.0.10/src/network".
```

- **Case 2:**

If Chelsio's T3 TOE driver (i.e., cxgb3toe package) for the T3 S-Series cards is installed, then an extra option CXGB3TOE_SRC=<cxgb3toe's source directory> is needed immediately after the make:

```
make CXGB3TOE_SRC=<cxgb3toe_source_dir>
```

Example:

The cxgb3toe package used is cxgb3toe-1.5.tar.gz and is uncompressed under /usr/src, the cxgb3toe source directory would be /usr/src/cxgb3toe-1.5/src, the make command would be:

```
"make CXGB3TOE_SRC=/usr/src/cxgb3toe-1.5/src".
```

- **Case 3:**

If a non-Chelsio card is used or a Chelsio N-Series card is used or a Chelsio S-Series card is used without enabling TOE functionality then "CDIR" option is not needed when using make.

Example: "make".

```
Case 1: [chelsio@]# make CXGB4TOE_SRC=<ChelsioUwire source
/src/network >

or

Case 2: [chelsio@]# make CXGB3TOE_SRC=<cxgb3toe source/src>

or

Case 3: [chelsio@]# make
```

If make fails because of an error "Unable to locate the kernel source", then run make and pass in the KDIR=<kernel_source_tree> variable. The <kernel_source_tree> is the location of the Linux kernel source files. It may be a kernel with or without the Chelsio driver.

```
Case 1: [chelsio@]# make CXGB4TOE_SRC=<ChelsioUwire source/
src/network> KDIR=<linux_source_tree>

or

Case 2: [chelsio@]# make CXGB3TOE_SRC=<cxgb3toe source/src>
KDIR=<linux_source_tree>

or

Case 2: [chelsio@]# make KDIR=<linux source tree>
```

This will compile

- the `iscsictl` and `chisns` tool in the `user/` subdirectory,
- the iSCSI kernel base module `chiscsi_base.ko` in the `base/` subdirectory,
- the iSCSI T4 or T5 offload module `chiscsi_t4.ko` in the `t4/` subdirectory in case 1,
- the iSCSI T3 offload module `chiscsi_t3.ko` in the `t3/` subdirectory in case 2.

INSTALLING THE iSCSI MODULES/TOOLS

To install the iSCSI software, the installer must be the root user.

1. Install the iSCSI module and tools

Run “make install” to install the iSCSI modules and the tools for all cases.

```
[chelsio@]# make install
```

The iSCSI module will be installed in the `/lib/modules/<linux_kernel_version>/kernel/drivers/iscsi` directory. The modules database will be updated by the installer. This allows the iSCSI module to be located when using the `modprobe` utility. The actual module `chiscsi.ko` can be found inside the package under `/chiscsi-5.xxx/kernel`.

The `iscsictl` tool and the `chisns` tool will be installed in `/sbin`. The `chisns` tool starts the iSNS client. The `iscsictl` tool is provided for configuring and managing the iSCSI targets and iSNS client. It also provides control for iSCSI global settings.

To load the module run `modprobe` as follows:

```
Case 1: Chelsio T4 or T5 S-series adapters are installed:
[root@]# modprobe chiscsi_t4

and/or

Case 2: Chelsio T3 S-series adapters are installed:
[root@]# modprobe chiscsi_t3

Or
```

```
Case 3: Chelsio N4 or N3 series adapters are installed:  
[root@]# modprobe chiscsi_base
```

A sample iSCSI configuration file will be installed in `/etc/chelsio-iscsi/chiscsi.conf`. This file should be edited using a standard text editor and customized to fit your environment.

2. Set iSCSI service to automatically start at bootup

The `chelsio-target` service scripts are installed to `/etc/init.d` and the parameters for the script are installed at `/etc/sysconfig/chiscsi`. The script is installed as a system service.

To auto-start the iSCSI target service at a certain runlevel, e.g. runlevel 3, `chkconfig` can be used on RedHat and Novell / SuSE based systems as follows:

```
[root@]# chkconfig --level 3 chelsio-target on
```

The `chelsio-target` service scripts do basic checks before starting the iSCSI target service, loads the kernel module, and starts all the targets configured by default. It can also be used to stop the targets, and restart/reload configuration.

3. Support

For any distribution specific questions or problems, please check ERRATA included in the `chiscsi` release or contact support@chelsio.com for assistance.

5 SOFTWARE CONFIGURATION

OVERVIEW

The Chelsio iSCSI software needs configuration before it can become useful. The following sections describe how this is done.

There are two main components used in configuring the Chelsio iSCSI software, the configuration file and the iSCSI control tool. This section describes in some detail what they are and their relationship they have with one another.

COMMAND LINE TOOLS

There are two command line tools, one for control of the iSNS client and one for control of the iSCSI target nodes.

iscsictl

The Chelsio iSCSI control tool, `iscsictl`, is a Command Line Interface (CLI) user space program that allows administrators to:

- Start/Stop the iSCSI Target
- Start the iSNS client
- Get/Set the iSCSI driver global settings
- Get/Set/Remove the iSCSI Target configuration settings
- Retrieve active sessions' information of an iSCSI Target
- Manually flush data to the iSCSI Target disks
- Reload the iSCSI configuration file
- Write current iSCSI configuration to a file
- Save the current iSCSI configuration to a file

chisns

The Chelsio iSNS client, `chisns`, can be started independently of `iscsictl`.

iSCSI CONFIGURATION FILE

The iSCSI configuration file is the place where information about the Chelsio iSCSI software is stored. The information includes global data that pertains to all targets as well as information on each specific iSCSI target node. Most of the information that can be placed in the configuration file has default values that only get overwritten by the values set in the configuration file.

There are only a few global configuration items that can be changed.

There are many specific parameters that can be configured, some of which are iSCSI specific and the rest being Chelsio specific. An example of an iSCSI specific item is "HeaderDigest" which is defaulted to "None" but can be overridden to "CRC32C". An example of a Chelsio specific configurable item is "ACL" (for Access Control List). "ACL" is one of the few items that have no default.

Before starting any iSCSI target, an iSCSI configuration file must be created. An easy way to create this file is to use the provided sample configuration file and modify it. This file can be named anything and placed in any directory but it must be explicitly specified when using `iscsictl` by using the `-f` option. To avoid this, put configuration file in the default directory (`/etc/chelsio-iscsi`) and name it the default file name (`chiscsi.conf`).

“On the fly” Configuration Changes

Parameters for the most part can be changed while an iSCSI node is running. However, there are exceptions and restrictions to this rule that are explained in a later section that describes the details of the iSCSI control tool `iscsictl`.

A QUICK START GUIDE FOR TARGET

This section describes how to get started quickly with a Chelsio iSCSI target. It includes:

- Basic editing of the iSCSI configuration file.
- Basic commands of the iSCSI control tool including how to start and stop a target.

A Sample iSCSI Configuration File

The default Chelsio iSCSI configuration file is located at `/etc/chelsio-iscsi/chiscsi.conf`. If this file doesn't already exist then one needs to be created.

To configure an iSCSI target, there are three required parameters (in the form of key=value pairs) needed as follows:

- **TargetName** – A worldwide unique iSCSI target name.
- **PortalGroup** – The portal group tag associating with a list of target IP address(es) and port number(s) that service the login request. The format of this field is a Chelsio specific iSCSI driver parameter which is described in detail in the configuration file section.
- **TargetDevice** – A device served up by the associated target. A device can be:
 - A block device (for example, `/dev/sda`)
 - A virtual block device (for example, `/dev/md0`)
 - A RAM disk
 - A regular file

A target can serve multiple devices, each device will be assigned a Logical Unit Number (LUN) according to the order it is specified (i.e., the first device specified is assigned LUN 0, the second one LUN 1, ..., and so on and so forth). Multiple **TargetDevice** key=value pairs are needed to indicate multiple devices.

Here is a sample of a minimum iSCSI target configuration located at `/etc/chelsio-iscsi/chiscsi.conf`:

```
target:
  TargetName=iqn.2006-02.com.chelsio.diskarray.san1
  TargetDevice=/dev/sda
  PortalGroup=1@192.0.2.178:3260
```

The **TargetDevice** value must match with the storage device in the system. The **PortalGroup** value must have a matching IP address of the Ethernet adapter card in the system.

For more information about TargetDevice configuration please refer to the later chapter titled “Target Storage Device Configuration”.

Basic iSCSI Control

Control of the Chelsio iSCSI software is done through `iscsictl`, the command line interface control tool. The following are the basic commands needed for effective control of the target.

Start Target: To start all of the iSCSI targets specified in the iSCSI configuration file, execute `iscsictl` with the “-S” option followed by “target=ALL”.

```
[chelsio@]# iscsictl -f /etc/chelsio-iscsi/chiscsi.conf -S target=ALL
```

To start a specific target execute `iscsictl` with “-S” followed by the target.

```
[chelsio@]# iscsictl -f /etc/chelsio-iscsi/chiscsi.conf -S  
target=iqn.2006-02.com.chelsio.diskarray.san1
```

Stop Target: To stop the all the iSCSI target(s), execute `iscsictl` with “-s” option followed by “target=ALL”.

```
[chelsio@]# iscsictl -s target=ALL
```

To stop a specific target execute `iscsictl` with “-s” followed by the target name.

```
[chelsio@]# iscsictl -s target=iqn.2006-02.com.chelsio.diskarray.san1
```

View Configuration: To see the configuration of all the active iSCSI targets, execute `iscsictl` with “-c” option.

```
[chelsio@]# iscsictl -c
```

To see the more detailed configuration settings of a specific target, execute `iscsictl` with “-c” option followed by the target name.

```
[chelsio@]# iscsictl -c target=iqn.2006-02.com.chelsio.diskarray.san1
```

View Global Settings: To see Chelsio global settings, execute `iscsictl` with “-g” option.

```
[chelsio@]# iscsictl -g
```


Change Global Settings: To change Chelsio global settings, execute `iscsiictl` with “-G” option.

```
[chelsio@]# iscsiictl -G iscsi_login_complete_time=300
```

View Help: To print help to `stdout`, execute `iscsiictl` with “-h” option.

```
[chelsio@]# iscsiictl -h
```

THE iSCSI CONFIGURATION FILE

The iSCSI configuration file consists of a series of blocks consisting of the following types of iSCSI entity blocks:

1. global:
2. target:

There can be only one global entity block whereas multiple target entity blocks are allowed. The global entity block is optional but there must be at least one target entity block.

An entity block begins with a block type (global or target). The content of each entity block is a list of parameters specified in a “key=value” format. An entity block ends at the beginning of the next entity block or at the end-of-file.

The parameter list in an entity block contains both:

- iSCSI parameters that override the default values
- Parameters that facilitate passing of control information to the iSCSI module

All lines in the configuration file that begin with “#” character are treated as comments and will be ignored. White space is not significant except in key=value pairs.

For the “key=value” parameters the <value> portion can be a single value or a list of multiple values. When <value> is a list of multiple values, they must be listed on one line with a comma “,” to separate their values. Another way to list the values instead of commas is to list their values as key=value pairs repeatedly, each on a new line, until they are all listed.

There are three categories of key=value parameter, the first category belongs to the global entity block whereas the second and third categories belong to target and initiator entity blocks:

1. The Chelsio Global Entity Settings of key=value pairs
2. The iSCSI Entity Settings of key=value pairs
3. The Chelsio Entity Settings of key=value pairs

The following sub-sections describe these three categories and list in tables the details of their key=value parameters.

Chelsio System Wide Global Entity Settings

Description

Chelsio System Wide Global Entity Parameters pass system control information to the iSCSI software which affects all targets in the same way. More detail of these parameters below can be found in a later section entitled "System Wide Parameters".

Table of Chelsio Global Entity Settings

<i>Key</i>	<i>Valid Values</i>	<i>Default Value</i>	<i>Multiple Values</i>	<i>Description</i>
iscsi_auth_order	"ACL" "CHAP"	"CHAP"	No	Authorization order for login verification on the target. Valid only when a target's ACL_Enable=Yes ACL: ACL first then CHAP CHAP: CHAP first then ACL <i>Applies to Target(s) Only</i>
DISC_AuthMethod	"CHAP" "NONE"	None	No	To choose an authentication method for discovery phase.
DISC_Auth_CHAP_Policy	"Oneway" "Mutual"	"Oneway"	No	Oneway or Mutual (two-way) CHAP
DISC_Auth_CHAP_Initiator	"<user id>":"<secret>"		Yes	CHAP user id and secret for the initiator. <user id> must be less than 256 characters. Commas "," are not allowed. <secret> must be between 6 and 255 characters. Commas "," are not allowed. The initiator user id and secret are used by the target to authenticate the initiator <i>NOTE: The double quotes are required as part of the format.</i>
DISC_Auth_CHAP_Target	"<user id>":"<secret>"		Yes	CHAP user id and secret for the target. <user id> must be less than 256 characters. Commas "," are not allowed. <secret> must be between 6 and 255 characters. Commas "," are not allowed. The target user id and secret are used by the initiator to authenticate the target while doing Mutual chap. <i>NOTE: The double quotes are required as part of the format.</i>
iscsi_chelsio_ini_idstr	a string of maximum of 255 characters	"cxgb4i"	No	To enable additional optimization when Chelsio Adapters and drivers are used at both ends (initiator and target) systems.

Key	Valid Values	Default Value	Multiple Values	Description
				Make sure the initiator name contain the substring set in iscsi_chelsio_ini_idstr when using Chelsio iscsi initiator driver.
iscsi_target_vendor_id	a string of maximum of 8 characters	"CHISCS I"	No	The target vendor ID part of the device identification sent by an iSCSI target in response of SCSI Inquiry command.
iscsi_login_complete_time	0 to 3600	300	No	Time allowed (in seconds) for the initiator to complete the login phase. Otherwise, the connection will be closed <i>NOTE: value zero means this check is NOT performed.</i>

iSCSI Entity Settings

Description

iSCSI Entity Parameters pass iSCSI protocol control information to the Chelsio iSCSI module. This information is unique for each entity block. The parameters follow the IETF iSCSI standard RFC 3720 in both definition and syntax. The descriptions below are mostly from this RFC.

Table of iSCSI Entity Settings

Key	Valid Values	Default Value	Multiple Values	Description
MaxConnections	1 to 65535	1	No	Initiator and target negotiate the maximum number of connections requested/acceptable.
InitialR2T	"Yes" "No"	"Yes"	No	To turn on or off the default use of R2T for unidirectional and the output part of bidirectional commands.
ImmediateData	"Yes" "No"	"Yes"	No	To turn on or off the immediate data.
FirstBurstLength	512 to 16777215 ($2^{24} - 1$)	65536	No	The maximum negotiated SCSI data in bytes of unsolicited data that an iSCSI initiator may send to a target during the execution of a single SCSI command.
MaxBurstLength	512 to 16777215 ($2^{24} - 1$)	262144	No	The maximum negotiated SCSI data in bytes, of a Data-In or a solicited Data-Out iSCSI sequence between the initiator and target.
DefaultTime2Wait	0 to 3600	2	No	The minimum time, in seconds, to wait before attempting an explicit / implicit logout or connection reset between initiator and target.
DefaultTime2Retain	0 to 3600	20	No	The maximum time, in seconds, after an initial wait.
MaxOutstandingR2T	1 to 65535	1	No	The maximum number of outstanding R2Ts per task.
DataPDUInOrder	"Yes" "No"	"Yes"	No	To indicate the data PDUs with sequence must be at continuously increasing order or can be in any order. <i>Chelsio only supports "Yes".</i>
DataSequenceInOrder	"Yes" "No"	"Yes"	No	To indicate the Data PDU sequences must be transferred in continuously non-decreasing sequence offsets or can be transferred in any order. <i>Chelsio only supports "Yes".</i>

Key	Valid Values	Default Value	Multiple Values	Description
ErrorRecoveryLevel	0 to 2	0	No	To negotiate the recovery level supported by the node. <i>Chelsio only supports 0.</i>
HeaderDigest	"None" "CRC32C"	"None"	Yes	To enable or disable iSCSI header cyclic integrity checksums.
DataDigest	"None" "CRC32C"	"None"	Yes	To enable or disable iSCSI data cyclic integrity checksums.
AuthMethod	"CHAP" and "None"	"None,CHAP"	Yes	To choose an authentication method during login phase.
TargetName	"<target name>"		No	A worldwide unique iSCSI target name. <i>Target only.</i>
TargetAlias	"<target alias>"		No	A human-readable name or description of a target. It is not used as an identifier, nor is it for authentication. <i>Target only.</i>
MaxRecvDataSegmentLength	512 to 16777215 ($2^{24} - 1$)	8192	No	To declare the maximum data segment length in bytes it can receive in an iSCSI PDU.
OFMarker	"Yes" "No"	"No"	No	To turn on or off the initiator to target markers on the connection. <i>Chelsio only supports "No".</i>
IFMarker	"Yes" "No"	"No"	No	To turn on or off the target to initiator markers on the connection. <i>Chelsio only supports "No".</i>
OFMarkInt	1 to 65535	2048	No	To set the interval for the initiator to target markers on a connection.
IFMarkInt	1 to 65535	2048	No	To set the interval for the target to initiator markers on a connection.

Chelsio Entity Settings

Description

Chelsio Entity Parameters pass control information to the Chelsio iSCSI module. The parameters are specific to Chelsio's implementation of the iSCSI node (target or initiator) and are unique for each entity block. The parameters consist of information that can be put into three categories:

1. Challenge Handshake Authentication Protocol (CHAP).
2. Target specific settings. All of the following parameters can have multiple instances in one target entity block (i.e., they can be declared multiple times for one particular target).
 - Portal Group
 - Storage Device
 - Access Control List (ACL)

Table of Chelsio Entity Settings

Key	Valid Values	Default Value	Multiple Values	Description
Chelsio CHAP Parameter (Target)				
Auth_CHAP_Initiator	"<user id>": "<secret>"		Yes	<p>CHAP user id and secret for the initiator.</p> <p><user id> must be less than 256 characters. Commas "," are not allowed.</p> <p><secret> must be between 6 and 255 characters. Commas "," are not allowed.</p> <p>The initiator user id and secret are used by the target to authenticate the initiator.</p> <p><i>NOTE: The double quotes are required as part of the format.</i></p>
Auth_CHAP_Target	"<user id>": "<secret>"		No	<p>CHAP user id and secret for the target.</p> <p><user id> must be less than 256 characters. Commas "," are not allowed.</p> <p><secret> must be between 6 and 255 characters. Commas "," are not allowed.</p> <p>The target user id and secret are used by the initiator to authenticate the target while doing Mutual chap.</p> <p><i>NOTE: The double quotes are required as part of the format.</i></p>

Key	Valid Values	Default Value	Multiple Values	Description
Auth_CHAP_Challenge Length	16 to 1024	16	No	CHAP challenge length
Auth_CHAP_Policy	"Oneway" or "Mutual"	"Oneway"	No	Oneway or Mutual (two-way) CHAP
Chelsio Target Specific Parameter				
PortalGroup	<p><portal group tag> @<target IP address> [:<port number>] . . . [,<target IP address> [:<port number>]] [,timeout= <timeout value in seconds>] [,portalgroup1, portalgroup2,... portalgroupn]</p>		Yes	<p>The portal group name associates the given target with the given list of IP addresses (and optionally, port numbers) for servicing login requests. It's required to have at least one per target.</p> <p><portal group tag> is a unique tag identifying the portal group. It must be a positive integer.</p> <p><target IP address> is the IP address associated with the portal group tag.</p> <p><port number> is the port number associated with the portal group tag. It is optional and if not specified the well-known iSCSI port number of 3260 is used.</p> <p><timeout> is optional, it applies to all the portals in the group. The timeout value is in seconds. It is used to detect loss of communications at the iSCSI level.</p> <p><i>NOTE: There can be multiple target IP address/port numbers per portal group tag. This enables a target to operate on multiple interfaces for instance.</i></p> <p><portalgroupX>The portalgroup to which login requests should be redirected to.</p> <p><i>NOTE: There can be multiple redirection target portalgroups specified for a particular target portal group and the redirection will happen to these in a round robin manner.</i></p>
ShadowMode	"Yes" "No"	"No"	No	To turn ShadowMode on or off for iSCSI Target Redirection
TargetSessionMaxCmd	1 to 2048	64	No	The maximum number of outstanding iSCSI commands per session.

Key	Valid Values	Default Value	Multiple Values	Description
TargetDevice	<path/name> [,FILE MEM BLK] [,NULLRW] [,SYNC] [,RO] [,size=xMB] [,ID=xxxxxx] [,WWN=xxxxxxxxxx] [,SN= xxxxxx]		No	<p>A device served up by the associated target.</p> <p>The device mode can be a:</p> <ul style="list-style-type: none"> Block Device (e.g. /dev/sda) Virtual Block Device (e.g. /dev/md0) RamDisk Regular File <p><path/name> is the path to the device - with the exception of when a RAM Disk is specified, where it is a unique name given to the device. If multiple RAM Disks are used for a target then each name must be unique within the target.</p> <p>NULLRW specifies that random data is returned for reads, and for writes data is dropped. Useful for testing network performance.</p> <p>SYNC specifies that the device will function in the write-through mode (i.e., the data will be flushed to the device before the response is returned to the initiator). <i>NOTE: SYNC is only applicable with FILE mode.</i></p> <p>RO specifies the device as a read-only device.</p> <p>FILE specifies this device should be accessed via the kernel's VFS layer. This mode is the most versatile, and it is the default mode in the cases where there is no mode specified.</p> <p>BLK specifies this device should be accessed via the kernel's block layer. This mode is suitable for high-speed storage device such as RAID Controllers.</p> <p>MEM specifies this device should be created as a RAM Disk.</p> <p>size=xMB is used with "MEM", to specify the RamDisk size. If not specified, the default RamDisk size is 16MB (16 Megabytes). The</p>

Key	Valid Values	Default Value	Multiple Values	Description
				<p>minimum value of x is 1 (1MB) and the maximum value is limited by system memory.</p> <p>SN is a 16 character unique value. ID is a 24 character unique value. WWN is a 16 character unique value.</p> <p>It is recommended when using a multipath aware initiator , the optional ID (short form for SCSI ID), SN and WWN values should be set manually for the TargetDevice. These values will be returned in Inquiry response (VPD 0x83).</p> <p>Multiple TargetDevice key=value pairs are needed to indicate multiple devices.</p> <p>There can be multiple devices for any particular target. Each device will be assigned a Logical Unit Number (LUN) according to the order it is specified (i.e., the first device specified is assigned LUN 0, the second one LUN 1, ..., and so on and so forth).</p> <p>NOTE: FILE mode is the most versatile mode, if in doubt use FILE mode.</p>
ACL_Enable	"Yes" "No"	"No"	No	<p>Defines if Chelsio's Access Control List (ACL) method will be enforced on the target:</p> <p>Yes: ACL is enforced on the target No: ACL is not enforced on the target</p> <p>NOTE: ACL flag is not allowed to be updated on the fly. Target must be restarted for new ACL flag to take effect.</p>
ACL	[iname=<name1>];<sip=<sip1>[:dip=<dip1>][:lun=<lun_list:permissions>]		Yes	<p>The ACL specifies which initiators and how they are allowed to access the LUNs on the target.</p> <p>iname=<Initiator Name> specifies one or more initiator names, the name must be a fully qualified iSCSI initiator name.</p>

Key	Valid Values	Default Value	Multiple Values	Description
				<p>sip=<Source IP address> specifies one or more IP addresses the initiators are connecting from.</p> <p>dip=<Destination IP address> specifies one or more IP addresses that the iSCSI target is listening on (i.e., the target portal IP addresses).</p> <p>NOTE: when configuring an ACL at least one of the above three must be provided:</p> <ul style="list-style-type: none"> • iname, and/or • sip, and/or • dip. <p>lun=<lun list>:<permission> controls how the initiators access the luns.</p> <p>The supported value for <lun list> is ALL.</p> <p><permissions> can be: R: Read Only RW or WR: Read and Write If permissions are specified then the associated LUN list is required.</p> <p>If no lun=<lun list>:[R RW] is specified then it defaults to ALL:RW.</p> <p>NOTE: ACL permission are not allowed to be changed on the fly. Target must be restarted first to take new ACL changes in effect.</p> <p>NOTE: For the Chelsio Target Software release with lun-masking included, <lun list> is in the format of <0..N 0~N ALL> Where: 0..N: only one value from 0 through N 0~N: a range of values between 0 through N ALL: all currently supported LUNs Multiple lists of LUN numbers are</p>

<i>Key</i>	<i>Valid Values</i>	<i>Default Value</i>	<i>Multiple Values</i>	<i>Description</i>
				allowed. When specifying the list separate the LUN ranges by a comma.
RegisteriSNS	"Yes" "No"	"Yes"	No	To turn on or off exporting of target information via iSNS

Sample iSCSI Configuration File

Following is a sample configuration file. While using iSCSI node (target), irrelevant entity block can be removed or commented.

```
#
# Chelsio iSCSI Global Settings
#
global:
    iscsi login complete time=300
    iscsi_auth_order=CHAP
    DISC_AuthMethod=None
    DISC_Auth_CHAP_Policy=Oneway
    DISC_Auth_CHAP_Initiator="initiator id1":"initiator sec1"
    DISC_Auth_CHAP_Target="target_id1":"target_secret1"

#
# an iSCSI Target "iqn.2006-02.com.chelsio.diskarray.san1"
# being served by the portal group "5". Setup as a RAM Disk.
#
target:
    TargetName=iqn.2006-02.com.chelsio.diskarray.san1
    # lun 0: a ramdisk with default size of 16MB
    TargetDevice=ramdisk, MEM
    PortalGroup=5@192.0.2.178:3260

#
# an iSCSI Target "iqn.2005-8.com.chelsio:diskarrays.san.328"
# being served by the portal group "1" and "2"
#
target:
    #
    # iSCSI configuration
    #
    TargetName=iqn.2005-8.com.chelsio:diskarrays.san.328
    TargetAlias=iTarget1
    MaxOutstandingR2T=1
    MaxRecvDataSegmentLength=8192
    HeaderDigest=None, CRC32C
    DataDigest=None, CRC32C
    ImmediateData=Yes
    InitialR2T=No
    FirstBurstLength=65535
    MaxBurstLength=262144

    #
    # Local block devices being served up
    # lun 0 is pointed to /dev/sda
    # lun 1 is pointed to /dev/sdb

    TargetDevice=/dev/sda, ID=aabbccddeeffgghh, WWN=aaabbbcccddeeeef
    TargetDevice=/dev/sdb

    #
    # Portal groups served this target
    #

    PortalGroup=1@102.50.50.25:3260
    PortalGroup=2@102.60.60.25:3260

    #
    # CHAP configuration
    #
    Auth_CHAP_Policy=Mutual
    Auth_CHAP_Initiator="iInitiator1":"InitSecret1"
    Auth_CHAP_Initiator="iInitiator2":"InitSecret2"
    Auth_CHAP_Target="iTarget1ID":"iTarget1Secret"
```

```

Auth CHAP ChallengeLength=16

#
# ACL configuration
#

# initiator "iqn.2006-02.com.chelsio.san1" is allowed full access
# to this target
ACL=iname=iqn.2006-02.com.chelsio.san1

# any initiator from IP address 102.50.50.101 is allowed full access of
# this target
ACL=sip=102.50.50.101

# any initiator connected via the target portal 102.60.60.25 is allowed
# full access to this target
ACL=dip=102.60.60.25

# initiator "iqn.2005-09.com.chelsio.san2" from 102.50.50.22 and
# connected via the target portal 102.50.50.25 is allowed read only access
# of this target
ACL=iname=iqn.2006-
02.com.chelsio.san2;sip=102.50.50.22;dip=102.50.50.25;lun=ALL:R

```

CHALLENGE-HANDSHAKE AUTHENTICATION PROTOCOL (CHAP)

The Chelsio iSCSI software supports Challenge-Handshake Authentication Protocol (CHAP). CHAP is a protocol that is used to authenticate the peer of a connection and uses the notion of a challenge and response, (i.e., the peer is challenged to prove its identity).

The Chelsio iSCSI software supports two CHAP methods: oneway and mutual (i.e., two way). CHAP is supported for both login and discovery sessions.

Normal Session CHAP Authentication

For a normal Session, the CHAP authentication is configured on a per-target basis

Oneway CHAP authentication

With Oneway CHAP (also called unidirectional CHAP) the target uses CHAP to authenticate the initiator. The initiator does not authenticate the target. This method is the default method.

For Oneway CHAP, the initiator CHAP id and secret are configured and stored on a per-initiator with Chelsio Entity parameter "Auth_CHAP_Initiator".

Mutual CHAP authentication

With mutual CHAP (also called bidirectional CHAP), the target uses CHAP to authenticate the initiator. The initiator uses CHAP to authenticate the target.

For mutual CHAP, in addition to the initiator CHAP id and secret, the target CHAP id and secret are required. They are configured and stored on a per target basis with Chelsio Entity parameter "Auth_CHAP_Target".

Adding CHAP User ID and Secret

A single `Auth_CHAP_Target` key and multiple `Auth_CHAP_Initiator` keys could be configured per target:

```
target:
  TargetName=iqn.2006-02.com.chelsio.diskarray.san1
  TargetDevice=/dev/sda
  PortalGroup=1@192.0.2.178:8000
  Auth_CHAP_Policy=Oneway
  Auth_CHAP_Initiator="remoteuser1":"remoteuser1_secret"
  Auth_CHAP_Initiator="remoteuser2":"remoteuser2_secret"
  Auth_CHAP_Target="targetid1":"target1_secret"
```

In the above example, target “iqn.2005-com.chelsio.diskarray.san1” has been configured to authenticate two initiators, and its own id and secret are configured for use in the case of mutual CHAP.

Discovery Session CHAP

CHAP authentication is also supported for the discovery sessions where an initiator queries of all available targets.

Discovery session CHAP is configured through the global section in the configuration file. List of keys to provision discovery chap are:

- `DISC_AuthMethod`: disable or enable discovery session CHAP.
- `DISC_Auth_CHAP_Policy`: oneway or mutual (i.e., two-way) authentication
- `DISC_Auth_CHAP_Initiator`: initiator chap user id and secret (required for Oneway CHAP)
- `DISC_Auth_CHAP_Target`: target chap user id and secret. (required for Mutual CHAP)

A sample below enables the discovery session chap and mutual chap authentication is required:

```
#
# Chelsio iSCSI Global Settings
#
global:
  DISC_AuthMethod=CHAP
  DISC_Auth_CHAP_Policy=Mutual
  DISC_Auth_CHAP_Initiator="initiator_id1":"initiator_secret1"
  DISC_Auth_CHAP_Target="target_id1":"target_secl"
```

TARGET ACCESS CONTROL LIST (ACL) CONFIGURATION

The Chelsio iSCSI target supports iSCSI initiator authorization via an Access Control List (ACL).

ACL configuration is supported on a per-target basis. The creation of an ACL for a target establishes:

- Which iSCSI initiators are allowed to access it
- The type of the access: read-write or read-only
- Possible SCSI layer associations of LUNs with the initiator

More than one initiator can be allowed to access a target and each initiator's access rights can be independently configured.

The format for ACL rule is as follows:

```
ACL=[iname=<initiator name>][;<sip=<source ip addresses>]
    [<dip=<destination ip addresses>][<lun=<lun_list>:<permissions>]
```

```
target:
    TargetName=iqn.2006-02.com.chelsio.diskarray.san1
    TargetDevice=/dev/sda

    PortalGroup=1@102.50.50.25:3260
    PortalGroup=2@102.60.60.25:3260

    # initiator "iqn.2006-02.com.chelsio.san1" is allowed
    # full read-write access to this target
    ACL=iname=iqn.2006-02.com.chelsio.san1

    # any initiator from IP address 102.50.50.101 is allowed full
    # read-write access of this target
    ACL=sip=102.50.50.101

    # any initiator connected via the target portal 102.60.60.25
    # is allowed full read-write access to this target
    ACL=dip=102.60.60.25

    # initiator "iqn.2005-09.com.chelsio.san2" from 102.50.50.22
    # and connected via the target portal 102.50.50.25 is allowed
    # read only access of this target
    ACL=iname=iqn.2006-
02.com.chelsio.san2;sip=102.50.50.22;dip=102.50.50.25;lun=ALL:R
```

ACL Enforcement

To toggle ACL enforcement on a per-target base, a Chelsio keyword "ACL_Enable" is provided:

- Setting "ACL_Enable=Yes" enables the target to perform initiator authorization checking for all the initiators during login phase. And in addition, once the initiator has been authorized to access the target, the access rights will be checked for each individual LU the initiator trying to access.
- Setting "ACL_Enable=No" disable the target to perform initiator authorization checking.

When a target device is marked as read-only (RO), it takes precedence over ACL's write permission (i.e., all of ACL write permission of an initiator is ignored).

TARGET STORAGE DEVICE CONFIGURATION

An iSCSI Target can support one or more storage devices. The storage device can either be the built-in RAM disk or an actual backend storage.

Configuration of the storage is done through the Chelsio configuration file via the key-value pair TargetDevice.

When option `NULLRW` is specified, on writes the data is dropped without being copied to the storage device, and on reads the data is not actually read from the storage device but instead random data is used. This option is useful for measuring network performance.

The details of the parameters for the key `TargetDevice` are found in the table of Chelsio Entity Settings section earlier in this document.

RAM Disk Details

For the built-in RAM disk:

- The minimum size of the RAM disk is 1 Megabyte (MB) and the maximum is limited by system memory.
- To use a RAM disk with a Windows Initiator, it is recommended to set the size ≥ 16 MB.

To configure an ramdisk specify `MEM` as the device mode:

```
TargetDevice=<name>,MEM,size=xMB
```

Where: `<name>` Is a unique name given to the RAM Disk. This name identifies this particular ramdisk. If multiple RAM Disks are configured for the same target, the name must be unique for each RAM Disk.

`x` Is the size of the RAM Disk in MB. It's an integer between 1 - max, where max is limited by system memory. If this value is not specified the default value is 16 MB.

```
target:
#<snip>
# 16 Megabytes RAM Disk named ramdisk1
TargetDevice=ramdisk1,MEM,size=16MB
#<snip>
```

FILE Mode Storage Device Details

The FILE mode storage device is the most common and versatile mode to access the actual storage attached to the target system:

- The FILE mode can accommodate both block devices and virtual block devices.
- The device is accessed in the exclusive mode. The device should not be accessed (or active) in any way on the target system.
- Each device should be used for one and only one iSCSI target.
- "SYNC" can be used with FILE mode to make sure the data is flushed to the storage device before the Target responds back to the Initiator.

To configure a FILE storage device specify `FILE` as the device mode:

```
TargetDevice=<path to the storage device>[,FILE][,SYNC]
```


Where: `<path>` Is the path to the actual storage device, such as `/dev/sdb` for a block device or `/dev/md0` for a software RAID. The path must exist in the system.

`SYNC` When specified, the Target will flush all the data in the system cache to the storage driver before sending response back to the Initiator.

Example Configuration of FILE Mode Storage

Below is an example:

```
target:
#<snip>
# software raid /dev/md0 is accessed in FILE mode
TargetDevice=/dev/md0,FILE
#<snip>
```

BLK Mode Storage Device Details

The BLK mode storage device is suitable for high-speed storage attached to the target system:

- The BLK mode can accommodate only block devices.
- The device is accessed in the exclusive mode. The device should not be accessed (or active) in any way on the target system.
- Each device should be used for one and only one iSCSI target.

To configure a block storage device specify `BLK` as the device mode:

```
TargetDevice=<path to the storage device>,BLK
```

Where: `<path>` Is the path to the actual storage device, such as `/dev/sdb`. The path must exist in the system.

```
target:
#<snip>
# /dev/sdb is accessed in BLK mode
TargetDevice=/dev/sdb,BLK
#<snip>
```

Multi-path Support

To support multi-path from the initiator, it is highly recommended that the following options to be specified:

- `[,ID=xxxxxx]`: SCSI ID, a twenty-four (24) bytes alpha-numeric string
- `[,WWN=xxxxxxxxxx]`: SCSI World Wide Name (WWN), a sixteen (16) bytes alpha-numeric string
- `[,SN= xxxxxx]`: SCSI SN, a sixteen (15) bytes alpha-numeric string.

The user should make sure the three values listed above are the same for the target luns that involved in the multipath.

TARGET REDIRECTION SUPPORT

An iSCSI Target can redirect an initiator to use a different IP address and port (often called a portal) instead of the current one to connect to the target. The redirected target portal can either be on the same machine, or a different one.

ShadowMode for Local vs. Remote Redirection

The ShadowMode setting specifies whether the Redirected portal groups should be present on the same machine or not. If ShadowMode is enabled, the redirected portal groups are on a different system. If it is disabled then the redirected portal groups must be present on the same system otherwise the target would fail to start.

Below is an example with ShadowMode enabled:

```
target:
#<snip>
    # any login requests received on 10.193.184.81:3260 will be
    # redirected to 10.193.184.85:3261.

    PortalGroup=1@10.193.184.81:3260,[2]
    PortalGroup=2@10.193.184.85:3261

    # the PortalGroup "2" is NOT presented on the same system.
    ShadowMode=Yes

#<snip>
```

Below is an example with ShadowMode disabled:

```
target:
#<snip>
    # any login requests received on 10.193.184.81:3260 will be
    # redirected to 10.193.184.85:3261

    PortalGroup=1@10.193.184.81:3260,[2]
    PortalGroup=2@10.193.184.85:3261

    # the PortalGroup "2" IS present on the same system
    ShadowMode=No

#<snip>
```

Redirecting to Multiple Portal Groups

The Chelsio iSCSI Target Redirection allows redirecting all login requests received on a particular portal group to multiple portal groups in a round robin manner.

Below is an example Redirection to Multiple Portal Groups:

```
target:
  #<snip>
    # any login requests received on 10.193.184.81:3260 will be
    # redirected to 10.193.184.85:3261 and 10.193.184.85:3262 in a
    # Round Robin Manner.

    PortalGroup=1@10.193.184.81:3260,[2,3]
    PortalGroup=2@10.193.184.85:3261
    PortalGroup=3@10.193.184.85:3262
    ShadowMode=No
  #<snip>
```

THE COMMAND LINE INTERFACE TOOLS “ISCSICTL” AND “CHISNS”

iscsictl

`iscsictl` is the tool Chelsio provides for controlling the iSCSI target. It is a Command Line Interface (CLI) that is invoked from the console. Its usage is as follows:

```
iscsictl <options> <mandatory parameters> [optional parameters]
```

The mandatory and optional parameters are the **key=value** pair(s) defined in RFC3720, or the **var=const** pair(s) defined for Chelsio iSCSI driver implementation. In this document, the key=value is referred to as “pair”, and var=const is referred to as “parameter” to clarify between iSCSI protocol’s pair value(s), and Chelsio iSCSI driver’s parameter value(s). Note that all **value** and **const** are case sensitive.

chisns

`chisns` is the command line tool for controlling the iSNS client. This is a simple tool that starts the iSNS client with a client and server parameter.

iscsictl options

Options	Mandatory Parameters	Optional Parameters	Descriptions
-h			Display the help messages.
-v			Display the version.
-f	<[path/] filename>		<p>Specifies a pre-written iSCSI configuration text file, used to start, write, save, or reload the iSCSI node(s).</p> <p>This option must be specified with one of the following other options: “-S”, “-W”. For the “-S” option “-f” must be specified first. All other options will ignore this “-f” option.</p> <p>If the “-f” option is not specified with the commands above the default configuration file will be used. It’s name and location is:</p> <pre>/etc/chelsio-iscsi/chiscsi.conf</pre> <p>The configuration file path and filename must conform to Linux standards.</p> <p>For the format of the iSCSI configuration file, please see “Format of The iSCSI Configuration File” section earlier in this document.</p>
-k	<key> [=<val>]		Specifies an iSCSI Entity or Chelsio Entity parameter. This option can be specified after “-c” option to retrieve a parameter setting..
-c		target=<name> [, name2 . . . , <nameN>	Display the Chelsio iSCSI target configuration. target=<name> parameter: Where name is the name of the node whose information will be returned. name can be one or more string of names, separated by a comma, <name1[,name2,...,nameN] ALL>

Options	Mandatory Parameters	Optional Parameters	Descriptions
]	<p>A name of ALL returns information on all targets. ALL is a reserved string that must be uppercase.</p> <p>Example: iscsictl -c target=iqn.com.cc.it1 iscsictl -c target=iqn.com.cc.target1 -k TargetAlias</p> <p>The <name> parameter can also be specified as one or more parameter on the same command line, separated by a comma, target=<name1>, <name2>, ... ,<nameN></p> <p>The target=<name> parameter(s) are optional and if not specified all active Chelsio iSCSI targets(s) configuration(s) will be displayed.</p> <p>If target=ALL is specified or no parameters are specified the output will be abbreviated. Specify specific targets to get detailed configuration data.</p> <p>If the target=<name> option is specified, the -k <key> option can optionally be specified along with this option to display only the selected entity parameter setting.</p> <p>Example: iscsictl -c target=iqn.com.cc.target1 -k HeaderDigest</p>
-F		target=<name> -k lun=<value>	<p>Flush the cached data to the target disk(s).</p> <p>target=<name> parameter: Where name is the name of the target to be flushed. name can be one or more string of names, separated by a comma, <name1[,name2,...,nameN] ALL></p> <p>A name of ALL will cause all the target data to be flushed. ALL is a reserved string that must be uppercase.</p> <p>The target=name parameter is optional. If no target=name parameter is specified, it is the same as specifying target=ALL.</p> <p>The -k lun=<value> option is optional. It can be used to further specify a particular lun to be flushed.</p> <p>Example: To flush all the targets in the system: iscsictl -F To flush a particular target: iscsictl -F target=iqn.com.cc.it1 To flush only the lun 0 of a particular target: iscsictl -F target=iqn.com.cc.it1 -k lun=0</p>
-g			Display the Chelsio iSCSI Global Entity settings.
-G	<var=const>		Set the Chelsio iSCSI Global Entity settings.

Options	Mandatory Parameters	Optional Parameters	Descriptions
			<p>var=const parameter: Where var=const can be anyone listed under Chelsio Global Entity Settings</p> <p>Example: iscsictl -G iscsi_auth_order=ACL</p> <p>The var=const parameter(s) are mandatory.</p> <p>If the var=const parameter is not specified, the command will be denied.</p> <p>If any of the specified var=const parameter is invalid, the command will reject only the invalid parameters, but will continue on and complete all other valid parameters if any others are specified.</p>
-s	target=<name>		<p>Stop the specified active iSCSI targets.</p> <p>target=<name> parameter: See the description of option -c for the target=<name> parameter definition.</p> <p>The target=<name> parameter is mandatory. If no target=<name> parameter is specified, the command will be denied.</p> <p>If the target=<name> parameter is specified, only the specified targets from the target=<name> parameters will be stopped.</p> <p>If target=ALL is specified, all active targets will stop.</p>
-S	target=<name>		<p>Start or reload the iSCSI targets.</p> <p>target=<name> parameter: Where name is the name of the target(s) that will be started or reloaded.</p> <p>The target=<name> parameter can be specified as one or more parameter on the same command line, separated by a space,</p> <p>target=<name1> target=<name2> ... target=<nameN></p> <p>The target=<name> parameter can also be, target=ALL</p> <p>A name of ALL starts or reloads all targets specified in the configuration file. ALL is a reserved string that must be uppercase.</p> <p>The target=<name> parameter is optional.</p> <p>If this command line option is specified without the -f option, the default configuration file <code>/etc/chelsio-iscsi/chiscsi.conf</code> will be used.</p>

Options	Mandatory Parameters	Optional Parameters	Descriptions
			<p>Rules,</p> <ol style="list-style-type: none"> 1. If the target=<name> parameter is specified, only the targets from the list will be started or reloaded. 2. If target=ALL is specified, all targets specified from the iSCSI configuration file will be started or reloaded. 3. If the target=<name> parameter is not specified, all active targets configurations will be reloaded from the configuration file while those targets are running. All non-active targets specified will not be loaded / started. <p>For Rules 1-3, if the specified targets are currently active (running), they will get reloaded.</p> <p>For Rules 1 & 2, if the specified targets are not currently active, they will be started.</p> <p>For Rules 2 & 3, please note the differences – they are not the same!</p> <p>The global settings are also reloaded from the configuration file with this option.</p>
-r	target=<name>	-k initiator=<name>	<p>Retrieve active iSCSI sessions under a target.</p> <p>target=<name> parameter: Where name must be a single target name.</p> <p>If target=<name> parameter is specified as target=<name>, the sessions can be further filtered based on the remote node name with optional -k initiator=<name> option.</p> <p>Examples: iscsictl -r target=iqn.com.cc.it1 iscsictl -r target=iqn.com.cc.it1 -k initiator=iqn.com.cc.ii1</p> <p>The first target=<name> parameter is mandatory. If it is not specified, the command will be denied.</p>
-D	<Session handle in hex>		<p>Drop initiator session.</p> <p>This option should be specified with the handle of the session (in hex) that needs to be dropped. The session handle can be retrieved using the previous mentioned iscsictl option (-r used to retrieve active iSCSI sessions under a target).</p>
-W			<p>Overwrite the specified iSCSI configuration file with ONLY the current iSCSI global settings and the active iSCSI targets' configuration to the specified iSCSI configuration file.</p> <p><i>Will delete any non-active targets' configuration from the specified file.</i></p> <p>The -f option MUST be specified along with this option.</p>

chisns options

<i>Options</i>	<i>Mandatory Parameters</i>	<i>Optional Parameters</i>	<i>Descriptions</i>
-h			Display the help messages.
	server=<IP address>[:<port>]	id=<isns entity id> query=<query interval>	<p>Start the Chelsio iSNS client.</p> <p>server=<IP address>[:<port>] where server is the iSNS server address. The port is optional and if it's not specified it defaults to 3205. The server with the ip address is mandatory and if it's not specified the, the command will be denied.</p> <p>id=<isns entity id> where id is the iSNS entity ID used to register with the server. It defaults to <hostname>.</p> <p>query=<query interval> where query is the initiator query interval (in seconds). It defaults to 60 seconds.</p> <p>Examples:</p> <p>chisns server=192.0.2.10 chisns server=192.0.2.10:3205 id=isnsc1n2 query=30</p> <p>In the first example the minimum command set is given where the IP address of the iSNS server is specified.</p> <p>In the second example a fully qualified command is specified by also setting three optional parameters. Here, the mandatory IP address and the corresponding optional port number are specified. Also set is the iSNS entity ID to 'isnsc1n2' as well as the query interval to 30 seconds.</p>

RULES OF TARGET RELOAD (I.E. "ON THE FLY" CHANGES)

After a target has been started its settings can be modified via reloading of the configuration file (i.e., `iscsictl -S`).

The following parameters **cannot** be changed once the target is up and running otherwise the target reload would fail:

- TargetName
- TargetSessionMaxCmd
- ACL_Enable
- ACL

The following parameters **can** be changed by reloading of the configuration file. The new value will become effective **immediately** for all connections and sessions:

- TargetDevice
- PortalGroup

The following parameter **can** be changed by reloading of the configuration file. The new value will **NOT** affect any connections and sessions that already completed login phase:

- TargetAlias
- MaxConnections
- InitialR2T

- ImmediateData
- FirstBurstLength
- MaxBurstLength
- MaxOutstandingR2T
- HeaderDigest
- DataDigest
- MaxRecvDataSegmentLength
- AuthMethod
- Auth_CHAP_Initiator
- Auth_CHAP_Target
- Auth_CHAP_ChallengeLength
- Auth_CHAP_Policy

The following parameters **should not** be changed because only one valid value is supported:

- DataPDUInOrder (support only "Yes")
- DataSequenceInOrder (support only "Yes")
- ErrorRecoveryLevel (support only "0")
- OFMarker (support only "No")
- IFMarker (support only "No")

The following parameters **can be** changed but would **not** have any effect because they are either not supported or they are irrelevant:

- DefaultTime2Wait (not supported)
- DefaultTime2Retain (not supported)
- OFMarkInt (irrelevant because OFMarker=No)
- IFMarkInt (irrelevant because IFMarker=No)

SYSTEM WIDE PARAMETERS

The Chelsio Global Entity Settings are system wide parameters that can be controlled through the configuration file or the use of the command line "`iscsiictl -G`". The finer points of some of these parameters are described in detail here:

iscsi_login_complete_time

Options: An integer value between 0 and 3600 (seconds), defaults is 300 (seconds)

This is the login timeout check. The value controls how long in seconds the initiator must complete the login phase. If an connection has been in the login phase longer than the set value, the target will drop the connection.

Value zero turns off this login timeout check.

iscsi_auth_order

Options: "ACL" or "CHAP", defaults to "CHAP"

On an iSCSI target when `ACL_Enable` is set to "Yes", `iscsi_auth_order` decides whether to perform CHAP first then ACL or perform ACL then CHAP.

When setting `iscsi_auth_order=ACL`, initiator authorization will be performed at the start of the login phase for a iSCSI normal session: upon receiving the first `iscsi_login_request`, the target will

check its ACL, if this iscsi connection does not match any ACL provisioned, the login attempt will be terminated.

When setting `iscsi_auth_order=CHAP`, initiator authorization will be performed at the end of the login phase for an iSCSI normal session: before going to the full feature phase, the target will check its ACL, if this iscsi connection does not match any ACL provisioned, the login attempt will be terminated.

NOTE: `iscsi_auth_order` has no meaning when `ACL_Enable` is set to "No" on a target.

iscsi_target_vendor_id

Options: A string of maximum of 8 characters, defaults to "CHISCSI".

The `iscsi_target_vendor_id` is part of the device identification sent by an iSCSI target in response of a SCSI Inquiry request.

iscsi_chelsio_ini_idstr

Options: A string of maximum of 255 characters, defaults to "cxgb4i".

For an iscsi connection, more optimization can be done when both initiator and target are running Chelsio adapters and drivers.

This string is used to compare with the initiator name received to identify if the initiator is running Chelsio drivers: if the initiator name contains the same substring as `iscsi_chelsio_ini_idstr` it is assumed the initiator is running with the Chelsio iscsi initiator driver and additional offload optimization is performed.

6 SOFTWARE COMPLIANCE

The Chelsio iSCSI software was designed and implemented to compliance with the following iSCSI RFCs:

- RFC-3720 – Internet Small Computer Systems Interface (iSCSI)
- RFC-3721 – Internet Small Computer Systems Interface (iSCSI) Naming and Discovery
- RFC-3722 – String Profile for Internet Small Computer Systems Interface (iSCSI) Names
- RFC-3980 – T11 Network Address Authority (NAA) Naming Format for iSCSI Node Names
- RFC-1994 – PPP Challenge Handshake Authentication Protocol (CHAP)
- RFC-4171 – Internet Storage Name Service (iSNS)

These published RFCs can be found at the IETF's web site at: www.ietf.org.

7 CUSTOMER SUPPORT

For any distribution specific problems, please check ERRATA included in the chiscsi release for possible workaround.

If you have problems with the software or hardware, please contact our customer support team via email at support@chelsio.com or check our website at www.chelsio.com.

Chelsio Communications, Inc.
370 San Aleso Ave.
Sunnyvale, CA 94085
U.S.A.

www.chelsio.com

Copyright © 2007-2015 by Chelsio Communications, Inc.
All rights reserved.