HP-UX iSCSI Software Initiator Support Guide

HP-UX 11i v1 & 11i v2

Edition 5

Manufacturing Part Number : T1452-90011

E0705

United States

Legal Notices

The information contained herein is subject to change without notice.

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.

U.S. Government License

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.

Trademark Notices

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

Contents

7. The iSCSI Software Interface Driver

A. Login Key Configuration

B. Transport Statistics

C. Diagnostic Messages

D. iSCSI Software Interface Driver Statistics

E. iSCSI Software Interface Driver Diagnostic Messages

F. Glossary

Tables

Figures

About This Document

This document describes how to install, configure, and troubleshoot the HP-UX iSCSI Software Initiator on HP-UX 11i v1 and HP-UX 11i v2 platforms.

The document manufacturing part number (T1452-90011) and publication time frame (E0705), provide a unique identifier for this document and indicate when it was published. The manufacturing part number will change when a new edition is released.

Document updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

The latest version of this document can be found on line at docs.hp.com.

Intended Audience

This document is intended for system and network administrators responsible for installing, configuring, and managing the HP-UX iSCSI Software Initiator. Administrators are expected to have knowledge of operating system concepts, commands, and configuration.

Experience with Transmission Control Protocol/Internet Protocol (TCP/IP), networking concepts, and network configuration is assumed.

This document is not a tutorial.

New and Changed Documentation in This Edition

- This edition has been updated to provide configuration information for the "Start on Boot" feature of the islpd (iSCSI Service Location Protocol) daemon.
- This edition has been updated to provide further clarification on how to troubleshoot issues with iswd daemon.

Publishing History

Table 1 iSCSI Software Initiator Support Guide Publishing History

Document Manufacturing Part Number	Operating Systems Supported	Supported Product Versions	Publication Date
5187-4536	11i v2	B.11.23.03	November 2003
5990-7213	11i v1 11i v2	B.11.11.03a B.11.23.03a	March 2004
T1452-90003	11i v1 11i v2	B.11.11.03b B.11.23.03b B.11.23.03c	May 2004
T1452-90008	11i v1 11i v2	B.11.11.03d B.11.23.03d	November 2004
T1452-90011	11i v1 11i v2	B.11.11.03e B.11.23.03e	July 2005

Related Documents

The *HP-UX iSCSI Software Initiator Release Notes* and the *HP-UX iSCSI Software Initiator Support Matrix* can be accessed at `http://www.docs.hp.com` in the *Networking and Communications* collection under the *iSCSI* category.

HP Encourages Your Comments

HP is committed to providing documentation that meets your needs.

Please let us know if you have suggestions or find any errors.

Please include the document title and manufacturing part number, and send comments to:
netinfo_feedback@cup.hp.com

1 iSCSI Overview

This chapter contains information on:

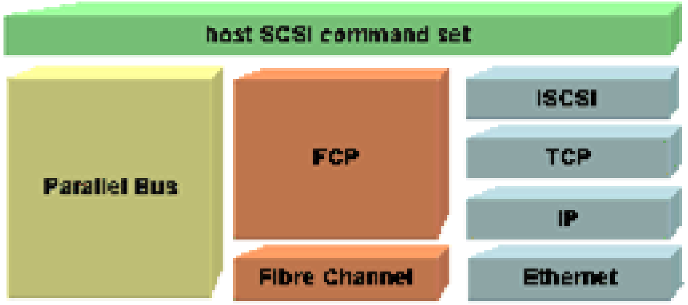
- “iSCSI Protocol Overview” on page 13
- “The iSCSI PDU” on page 14
- “iSCSI Layering” on page 15

- “iSCSI Session and TCP Connections” on page 16
- “iSCSI Login” on page 17
- “iSCSI Concepts: Network Entities, Portals, and Nodes” on page 19

iSCSI Protocol Overview

NOTE This chapter provides a brief, high level, overview of the iSCSI Protocol as defined by RFC 3720. For comprehensive information on the iSCSI Protocol specification, consult RFC 3720 at: `http://www.ietf.org/rfc/rfc3720.txt`

Figure 1-1 iSCSI: A Transport Protocol Alternative that Operates Over TCP/IP



SCSI (Small Computer Systems Interface) is a widely implemented family of protocols used for communication with I/O devices, particularly storage devices.

SCSI is a client-server architecture. Clients of a SCSI interface are called “initiators”. Initiators issue SCSI “commands” to request services from “targets”. Targets are typically components, or logical units, on a server.

A “SCSI transport” maps the client-server SCSI protocol to a specific interconnect. Initiators are one endpoint of a SCSI transport and targets are the other endpoint. The SCSI protocol has been mapped over various transports, including Parallel SCSI and Fibre Channel.

iSCSI is a transport protocol for SCSI, operating at the same level as Parallel SCSI and Fibre Channel.

iSCSI is a storage transport protocol developed by the Internet Engineering Task Force (IETF) for transporting SCSI packets over TCP/IP.

iSCSI provides an interoperable solution that takes advantage of existing Internet infrastructure and Internet management facilities.

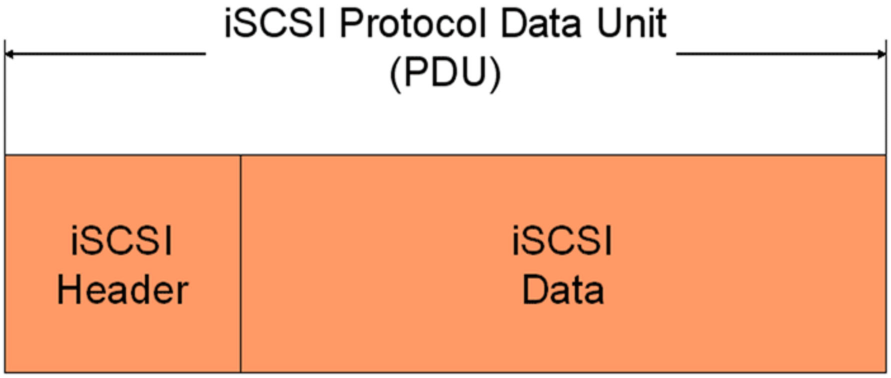
iSCSI does not have the distance limitations associated with the Fibre Channel storage transport.

The iSCSI protocol enables the transport of Block I/O over IP Networks. It operates on top of TCP by encapsulating SCSI commands in a TCP/IP stream.

The iSCSI PDU

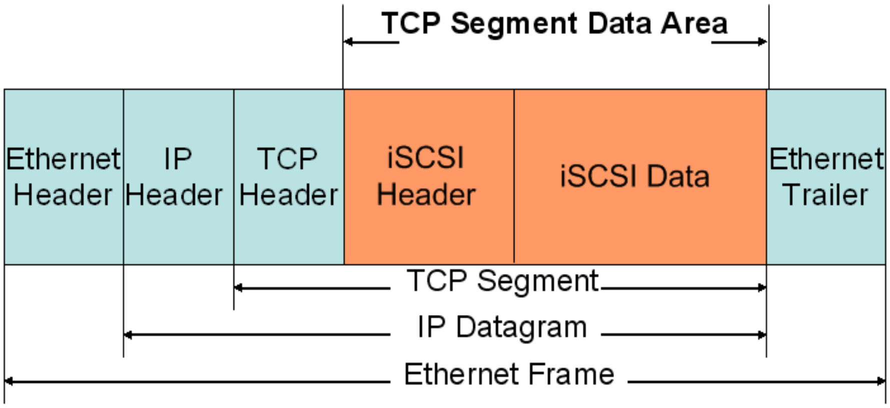
iSCSI initiators and targets communicate with messages known as “iSCSI Protocol Data Units”. An iSCSI PDU has a Header and an optional Data Section.

Figure 1-2 iSCSI Protocol Data Unit (PDU)



iSCSI PDUs are transported in the TCP Segment Data Area of Ethernet Frames. The size of an iSCSI PDU is not dictated by the capacity of the TCP segment data area and an iSCSI PDU does not need to begin at a specific offset within a TCP segment data area. An iSCSI PDU can span multiple TCP segment data areas, or several iSCSI PDUs can be contained in a single TCP segment data area.

Figure 1-3 TCP Segment Data Area of an Ethernet Frame



iSCSI Layering

iSCSI constructs Protocol Data Units (PDUs) consisting of SCSI commands, SCSI data and SCSI responses. iSCSI PDUs are inserted in the TCP segment data area of IP datagrams. The IP datagrams are then moved through the TCP/IP stack and transported over the network, between the SCSI services layer of host systems and the SCSI services layer of target storage devices.

The iSCSI protocol works seamlessly over TCP/IP networks, without requiring any changes to TCP/IP protocols.

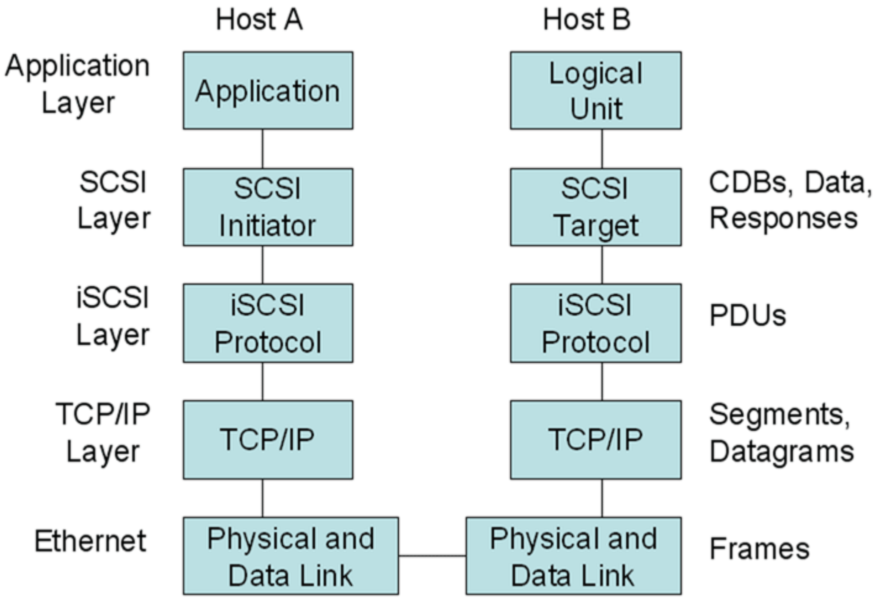
In the outbound direction (Initiator to Target):

1. The SCSI layer builds SCSI Command Descriptor Blocks (CDBs) and passes them to the iSCSI layer (along with the rest of the command execution parameters).
2. The iSCSI layer builds iSCSI PDUs and relays them to one or more TCP connections.
3. The TCP connection(s) form an initiator-target “session” (I_T Nexus).

In the inbound direction (Target to Initiator):

1. The iSCSI layer receives iSCSI PDUs on one or more TCP connections in a TCP/IP stream.
2. The iSCSI layer extracts the SCSI CDBs from the iSCSI PDUs and passes them to the SCSI layer.

Figure 1-4 iSCSI Layering



iSCSI Session and TCP Connections

In an iSCSI session, communication between an initiator and a target occurs over one or more TCP connections. The TCP connections carry control messages, data digests, SCSI commands, parameters, and data, all encapsulated in iSCSI Protocol Data Units (iSCSI PDUs).

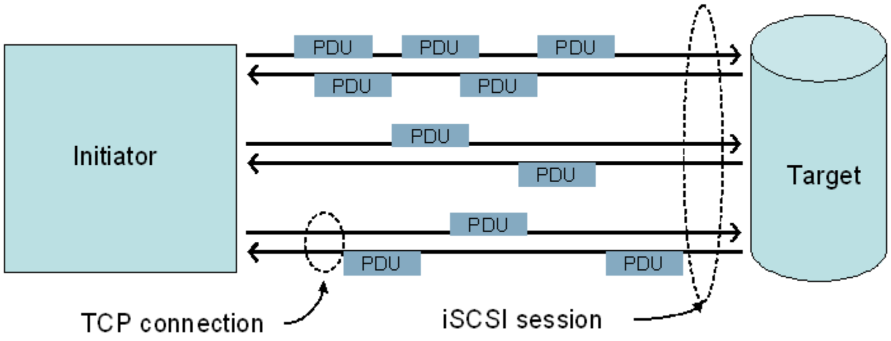
The TCP connections that link an initiator with a target, forming an iSCSI session, are comparable to a SCSI I_T nexus.

An initiator sees one “target image” across all of the connections in a session. A target also sees one “initiator image” across all of the connections in a session.

iSCSI defines two types of sessions:

- **Normal Operational Session -** a session in which SCSI commands, data and responses can be transferred between an iSCSI initiator and an iSCSI target.
- **Discovery Session -** a session only opened for target discovery.

Figure 1-5 An iSCSI Session



iSCSI Login

The iSCSI login enables:

- A TCP connection for iSCSI use
- Authentication of the parties
- Negotiation of the session’s parameters
- Marking the connection as belonging to an iSCSI session

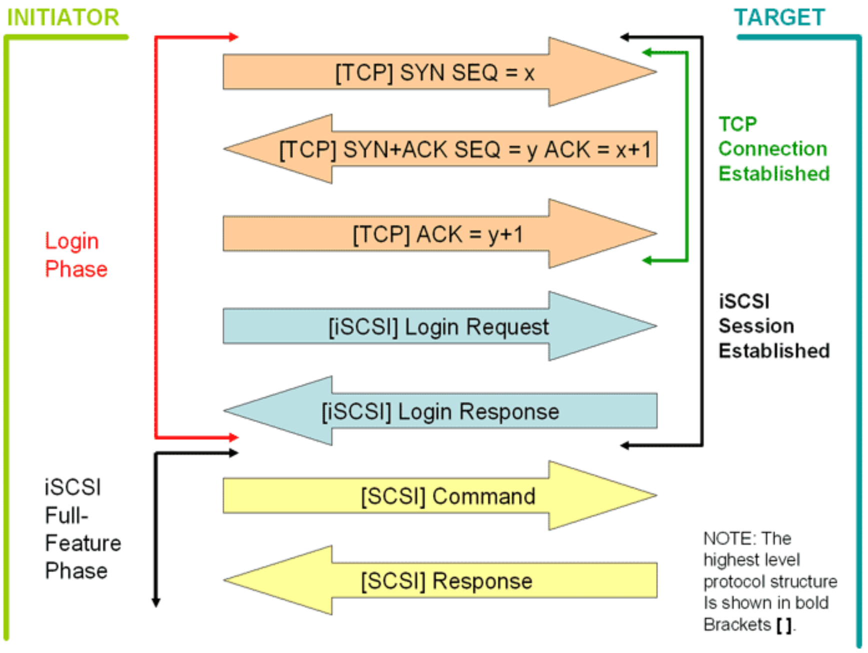
An iSCSI session is established to identify all of the connections between an initiator and a target belonging to the same I_T nexus.

Targets listen on a well-known TCP port (3260, as defined in the iSCSI Protocol Specification), or on a user configured TCP port, for incoming connections. The initiator begins the login process by connecting to one of these TCP ports.

An iSCSI Session has two phases:

- Login Phase
- Full Featured Phase

Figure 1-6 iSCSI Session Establishment and Phases


INITIATOR
TARGET
[TCP] SYN SEQ = x
[TCP] SYN+ACK SEQ = y ACK = x+1
[TCP] ACK = y+1
TCP Connection Established
Login Phase
[iSCSI] Login Request
[iSCSI] Login Response
iSCSI Session Established
iSCSI Full-Feature Phase
[SCSI] Command
[SCSI] Response
NOTE: The highest level protocol structure Is shown in bold Brackets **[ ]**.

Login Phase

The iSCSI Login Phase consists of Login requests and responses. Once authentication has occurred and operational parameters have been set, the session transitions to the Full Feature Phase and the initiator begins performing SCSI I/Os. NOTE: Using authentication is optional.

iSCSI parameters are negotiated using Login Requests and Responses, during session establishment. During the Full Feature Phase, iSCSI parameters are negotiated using Text Requests and Responses. In both cases the mechanism is an exchange of iSCSI-text-key=value pairs (also referred to as key=value pairs).

The Login Phase proceeds in two stages:

- Security/Authentication Stage

  This stage consists of text exchanges using IDs, Certificates, etc., using key=value pairs.

  One of the keys that is negotiated in this stage of the Login Phase is AuthMethod. For example:

key=value	AuthMethod=CHAP

  AuthMethod defines the authentication method.

- Operational Parameters Negotiation Stage

  This stage consists of text string negotiation of operating parameters using key=value pairs of login parameter exchanges.

  Two of the many login keys that are negotiated in the Operational Parameters Negotiation stage of the Login Phase are MaxRecvDataSegmentLength and FirstBurstLength. For example:

key=value	MaxRecvDataSegmentLength=<numerical-value>

  MaxRecvDataSegmentLength defines the maximum data segment length an initiator or target can receive in an iSCSI PDU (in bytes).

key=value	FirstBurstLength=<numerical-value>

  FirstBurstLength defines the maximum amount of unsolicited data the initiator can send to the target during the execution of a single SCSI command (in bytes).

NOTE For a complete list of iSCSI login keys, consult RFC 3720 at: `http://www.ietf.org/rfc/rfc3720.txt`

iSCSI Full Feature Phase

After successfully completing the Login Phase on the first (leading) connection of the session, a session is in Full Feature Phase.

In the Full Feature Phase, the initiator sends SCSI commands and data to the target by encapsulating them in iSCSI PDUs that go over the iSCSI session (transport). The initiator receives SCSI responses embedded in iSCSI PDUs, from the target. SCSI I/O only occurs after the Full Feature Phase begins.

iSCSI Concepts: Network Entities, Portals, and Nodes

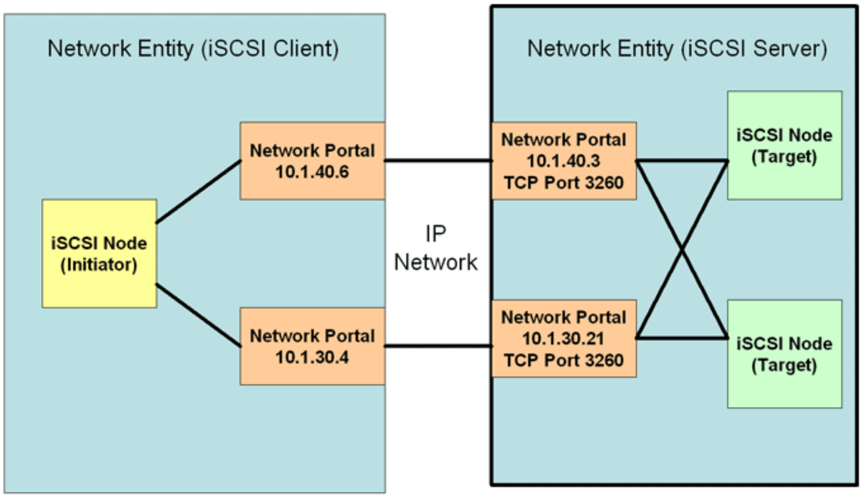
- **Network Entity -** a device or gateway that is accessible from the IP network. A network entity must have one or more network portals. Each network portal can be used by one or more iSCSI nodes within that network entity to gain access to the IP network.
- **Network Portal -** a component of a network entity that has a TCP/IP network address and may be used by an iSCSI node within that network entity for the connection(s) within one of its iSCSI sessions.

  A network portal in an initiator is identified by its IP address.

  A network portal in a target is identified by its IP address and its listening TCP port.
- **iSCSI Node -** a single iSCSI initiator or iSCSI target. There can be one or more iSCSI nodes within a network entity.

Figure 1-7 provides an example of two iSCSI nodes (targets in this case) sharing network portals within the same network entity.

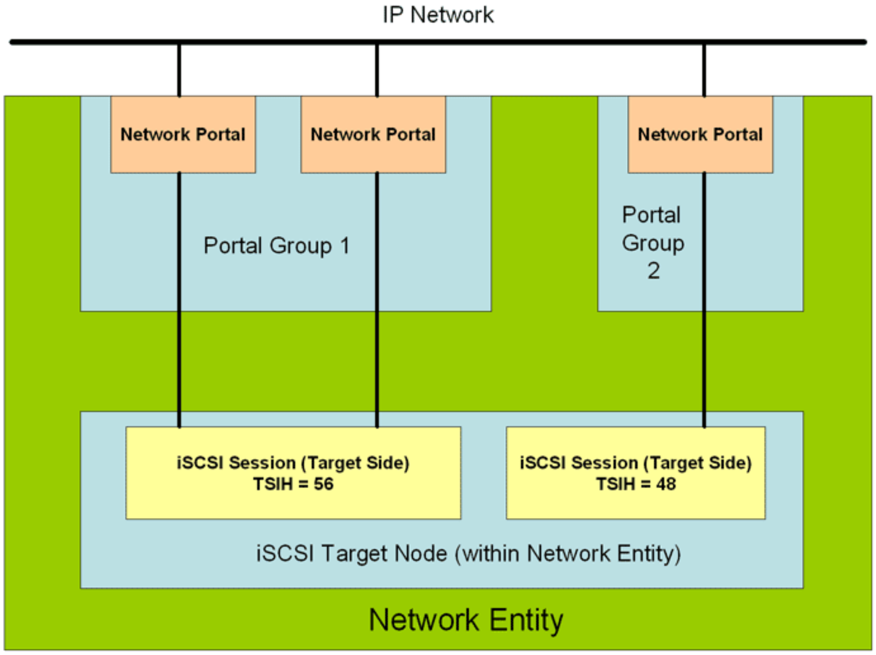
Figure 1-7 Network Entities, Portals and Nodes


Network Entity (iSCSI Client)
Network Entity (iSCSI Server)
Network Portal 10.1.40.6
iSCSI Node (Initiator)
Network Portal 10.1.30.4
IP Network
Network Portal 10.1.40.3 TCP Port 3260
Network Portal 10.1.30.21 TCP Port 3260
iSCSI Node (Target)
iSCSI Node (Target)

- **Portal Groups -** a set of network portals within a network entity that share network connections and can collectively coordinate an iSCSI session.
- **Target Portal Group (TPG) -** Although iSCSI initiators and iSCSI targets use portal groups to coordinate iSCSI sessions, only target portal groups are used directly in the iSCSI protocol.
- **Target Portal Group Tag (TPGT) -** iSCSI portal groups that are associated with target nodes are identified by a numerical target portal group tag ranging from 0 to 65535.
- **Target Session Identifying Handle (TSIH) -** a tag generated by an iSCSI target to identify an iSCSI session with a specific iSCSI initiator. The TSIH is generated during session establishment.

Figure 1-8 shows an iSCSI Target sharing two Target Portal Groups to conduct multiple iSCSI sessions.

Figure 1-8 iSCSI Target Sharing Two Target Portal Groups


IP Network
Network Portal
Network Portal
Network Portal
Portal Group 1
Portal Group 2
iSCSI Session (Target Side) TSIH = 56
iSCSI Session (Target Side) TSIH = 48
iSCSI Target Node (within Network Entity)
Network Entity

2 HP-UX iSCSI Software Initiator Product Overview

This chapter contains information on:

HP-UX iSCSI Software Initiator Features

- iSCSI Specification Compliance

  The iSCSI Software Initiator is based on the iSCSI Protocol Specification (RFC 3720). No attempt has been made to support any preliminary version of the iSCSI specification. For details, consult RFC 3720 at: `http://www.ietf.org/rfc/rfc3720.txt`

- SCSI Specification Compatibility

  The iSCSI Software Initiator is compatible with the current SCSI user API. Customers using the existing interface to SCSI will experience no compatibility issues with the iSCSI Software Initiator.

- Application Compatibility

  The iSCSI Software Initiator requires no changes to applications that access mass storage devices.

- Target Interoperability

  The iSCSI Software Initiator is interoperable with all iSCSI Protocol Specification (RFC 3720) compliant targets. No component of the iSCSI Software Initiator implementation is, in any way, HP proprietary. No component of the iSCSI Software Initiator requires any feature of iSCSI that is not mandatory for iSCSI Protocol Specification (RFC 3720) to operate properly.

- Driver Tracing and Logging

  The iSCSI Software Initiator supports driver tracing and the logging of events.

  HP-UX Event Monitoring Services (EMS) are supported by the iSCSI Software Initiator.

- Connections per Session

  The iSCSI Software Initiator supports one connection per session (an iSCSI session is equivalent to a SCSI I-T nexus).

- HP-UX Integration

  The iSCSI Software Initiator has been fully integrated into HP-UX. Startup, shutdown, volume managers, file systems, SAM, and STM, have been modified to support iSCSI.

  The iSCSI Software Initiator will also work with PVLinks.

  Existing applications will run unchanged on iSCSI volumes. See “System Startup” on page 72, for additional information.

- Challenge-Handshake Authentication Protocol (CHAP)

  The iSCSI Software Initiator supports CHAP for target authentication of initiators and initiator authentication of targets.

- Multiprocessor Compatible

  The iSCSI Software Initiator is usable in a multi-processor environment.

- New Management Tool

  The iSCSI Software Initiator introduces the “`iscsiutil`” management tool to the system. The `iscsiutil` tool is installed with the iSCSI Software Initiator.

- Available Online

  The iSCSI Software Initiator can be downloaded from the HP Software Depot and installed using the Software Distributor (SD) installation software. See “System Startup” on page 72 for details.

HP-UX iSCSI Software Initiator Components

The iSCSI Software Initiator has two kernel components:

- **iSCSI Transport Driver:** the session management and transport module.
- **iSCSI Software Interface Driver:** this module interfaces with the iSCSI Transport Driver and the host based TCP/IP stack.

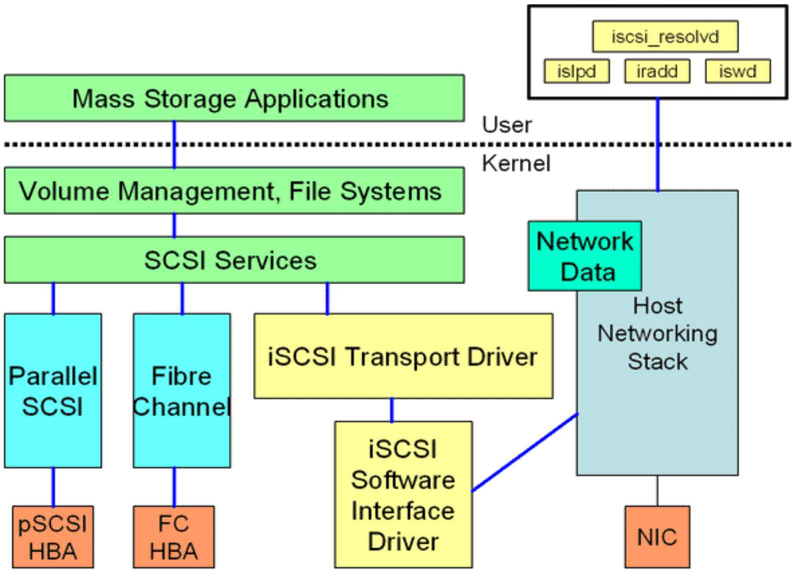
The iSCSI Software Initiator has four user space components:

- **iswd:** A connection management daemon. It opens and terminates connections for the iSCSI Software Interface Driver.
- **iscsi_resolvd:** This daemon resolves addresses for the iSCSI Software Transport Driver. It resolves host names, provided the domain name can be determined from the host name.
- **islpd:** This daemon is the SLP User Agent for iSCSI. It interfaces with the iSCSI Software Transport Driver to update iSCSI persistent information with discovered iSCSI Targets.
- **iradd:** This daemon implements the interface between the iSCSI Software Transport driver and a RADIUS server. It authenticates targets during iSCSI login using the Challenge Handshake Authentication Protocol (CHAP).

NOTE For more information on the iswd daemon, see “iSCSI Software Interface Driver Daemon (iswd)” on page 79.

For more information on the iscsi_resolvd, islpd, and iradd daemons, see “iSCSI Software Initiator Daemons” on page 60.

Figure 2-1 HP-UX iSCSI Software Initiator Block Diagram



Targets

The iSCSI Software Initiator is interoperable with all iSCSI Protocol Specification (RFC 3720) compliant targets. No component of the iSCSI Software Initiator implementation is, in any way, HP proprietary.

The iSCSI Software Initiator supports the static discovery of targets and, optionally, the dynamic discovery of targets.

Hardware Path Representation

The iSCSI virtual Host Bus Adapter (HBA) is defined in the IO tree with respect to a virtual root node, which will always have a value of 255. The hardware path for iSCSI targets is built off of this virtual HBA node, instead of being built off of a physical HBA node, as is currently done with Fibre Channel and parallel SCSI targets. The interaction between the iSCSI session and a physical HBA is abstracted in the virtual HBA.

The external representation of the hardware path for an iSCSI LUN is:

255/<iscsi_virtual_HBA>/<session_instance>.<virtual_bus>.<virtual_target>.<lun>

where:

- **iscsi_virtual_HBA** represents a virtual HBA off of the virtual root node. The value of **iscsi_virtual_HBA** is typically zero (0).
- **session_instance** is the iSCSI session instance number. The iSCSI session defines a SCSI I-T nexus for transactions between the iSCSI initiator and the iSCSI target.
- **virtual_bus** is an instance number (ext_bus) of a SCSI-2 bus.
- **virtual_target** is a virtual representation of a SCSI-2 target.
- **lun** is the representation of a SCSI-2 logical unit number.

  Because iSCSI targets define internal storage in SCSI-3 format, a mapping is required between the SCSI-3 LUN and:

  — The HP-UX SCSI-2 bus

  — The HP-UX SCSI-2 target

  — The HP-UX SCSI-2 LUN representation

  The mapping used is equivalent to the mapping used by the HP-UX Fibre Channel driver.

The hardware path for iSCSI targets will be persistent across reboots. It is maintained in the **/etc/ioconfig** file and used during system startup to reconstruct the IO tree.

Device Discovery

The implementation of iSCSI on HP-UX uses a static discovery algorithm as the default means of identifying iSCSI targets (devices). As an option, dynamic discovery of targets is available using Service Location Protocol (SLP).

Static Discovery

Static discovery requires the system administrator to identify all iSCSI discovery targets that will be accessible to an HP-UX host before an `ioscan` is executed. The configuration is done using the `iscsiutil` tool (see "The iscsiutil tool" on page 55, for details). Targets are defined using either their IP addresses or their DNS host names, combined with:

— the TCP port number on the iSCSI target used for iSCSI access

— the target portal group tag

The iSCSI target information that is entered through the `iscsiutil` tool will be maintained in a persistent area of storage called the kernel registry. It is only necessary to enter the target data once, because the kernel registry data will persist across reboots and upgrades.

When an `ioscan` is initiated, the iSCSI Software Initiator performs a probe by obtaining the target data from the kernel registry and then attempting to establish a session with the iSCSI discovery target. If an iSCSI session is established, a successful discovery login with the iSCSI discovery target is implied. A successful discovery login will result in operational targets (reported behind a discovery target) being registered into the kernel registry. A successful discovery session will be closed when the probe is complete.

An iSCSI normal session is established to each operational target registered in the kernel registry. An iSCSI normal session is identified by a session instance identifier. Once the normal session has been successfully established, the HP-UX iSCSI Software Initiator will determine the number of LUNs behind the target. Any LUNs found are used to define SCSI-2 virtual busses that will later be used for SCSI-2 target and LUN probing by the SCSI Services layer.

An iSCSI session will be used to define one or more SCSI-2 virtual busses. The virtual busses are necessary because iSCSI target storage is defined using SAM-2 (SCSI-3), but HP-UX supports SCSI-2.

SCSI-3 LUNs behind a target will typically be defined sequentially starting at LUN 0. Because the SCSI-2 implementation only permits 128 LUNs per bus, the bus mapping will define a virtual bus for each 128 LUN grouping. Thus, if one or more LUNs exist in the range 0-127, then the iSCSI Software Initiator creates virtual bus 0. If one or more LUNs exist in the range 128-255, then the iSCSI Software Initiator creates virtual bus 1, etc. This process is repeated until all SCSI-3 LUNs on the iSCSI target are mapped to HP-UX iSCSI virtual busses. Next, the HP-UX SCSI Services will probe the virtual SCSI-2 busses and define SCSI-2 virtual targets and LUNs.

NOTE Traditionally, HP-UX has used the `ioscan` tool to dynamically discover all possible targets and LUNs accessible by a host. Dynamic discovery is only available for iSCSI targets that support Service Location Protocol (SLP), provided SLP is available to the HP-UX host.

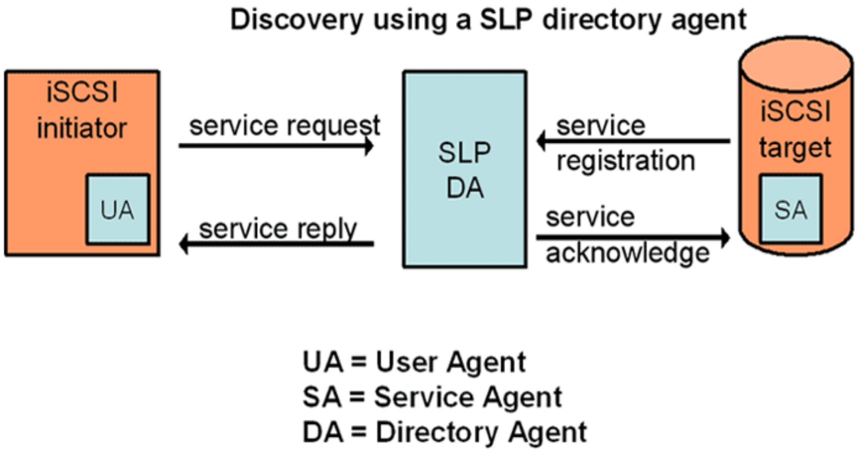
Service Location Protocol Based Dynamic Discovery

Service Location Protocol (SLP) is used for iSCSI dynamic discovery. The **islpd** daemon is a user space daemon that implements the SLP User Agent (UA) and the interface to the iSCSI transport driver.

The HP-UX SLP components must be separately installed on the system. See Table 2-1, "SLP Server Installation Information and Related Documents," on page 27, for detailed information.

The Directory Agent (DA) can be on the same system, or on any other system in the same subnet. Targets must be on the same subnet as the DA to be discovered by the DA. At least 1 DA must be present on the subnet. Dynamic scanning based on SLP is triggered when the `ioscan` command is executed. The islpd queries the DA(s) on the subnet for all of the iSCSI targets. Only targets that support SLP (and have been configured to use SLP) will be registered with the DA. Each target address supplied by the SLP DA is entered into the kernel registry as an operational target.

Figure 2-2 Service Location Protocol



NOTE Once the operational targets are registered, the process used for static discovery of targets applies to them (starting from normal session establishment to operational targets), see “Static Discovery” on page 26.

Table 2-1 SLP Server Installation Information and Related Documents

Description	url
Service Location Protocol, version 2 (RFC 2608)	http://www.ietf.org/rfc/rfc2608.txt
An API for Service Location (RFC 2614)	http://www.ietf.org/rfc/rfc2614.txt
HP Website to download SLP for HP-UX	http://www.software.hp.com (search for SLP)
Open source SLP code and references	http://www.openslp.org
Finding iSCSI targets and Name Servers using SLP (RFC 4018)	http://www.ietf.org/rfc/rfc4018.txt

3 Installation

This chapter contains information on:

- “Verifying the Installation” on page 35

Locating and Installing the iSCSI Software Initiator

The iSCSI Software Initiator is located at the HP Software Depot.

1. Go to http://www.software.hp.com.
2. Enter “iSCSI Software Initiator” in the “search” box.
3. When the search results show “iSCSI Software Initiator”, click on “Receive for Free”.
4. In the “Software Specification” window, highlight the HP-UX version that you want to install the HP-UX iSCSI Software Initiator on, then complete the required fields and click on “Next >>”.
5. Look for "Download Software", then click on the iSCSI Software Initiator version that you would like to download.
6. In the “Document” column (next to the “Download Software” column), click on “Installation Instruction” to download instructions for using the Software Distributor tool to install the iSCSI Software Initiator.

NOTE The Software Distributor is used for software installations on HP-UX systems. It also provides an interface for removing software from HP-UX systems.

iSCSI Software Initiator Components

The iSCSI Software Initiator is composed of several files that will be copied to the appropriate directories on the host system during installation.

The Software Distributor will add all of the following iSCSI files to HP-UX revision 11i v1 and HP-UX revision 11i v2 host configurations:

- **/usr/conf/lib/libiscsi.a**

  This is the iSCSI library of 64 bit object modules that will be linked into the HP-UX kernel.
- **/opt/iscsi/bin/iscsiutil**

  This is the iSCSI utility tool.
- **/opt/iscsi/bin/iswd**

  This is the iSCSI Software Interface Driver daemon.
- **/opt/iscsi/bin/iscsi_resolvd**

  This is the iSCSI hostname resolution daemon.
- **/opt/iscsi/bin/iradd**

  This is the iSCSI RADIUS server daemon for CHAP.
- **/opt/iscsi/bin/islpd**

  This is the iSCSI Service Location Protocol (SLP) daemon.
- **/sbin/rc2.d/S532iscsi**

  This is the iSCSI startup script, which is symbolically linked to the **/sbin/init.d/iscsi** file.
- **/usr/share/man/man7.Z/iscsi.7**

  /usr/share/man/man1m.Z/iscsiutil.1m

  /usr/share/man/man1m.Z/iscsi-daemons.1m

  /usr/share/man/man1m.Z/iswd.1m

  /usr/share/man/man1m.Z/islpd.1m

  /usr/share/man/man1m.Z/iradd.1m

  /usr/share/man/man1m.Z/iscsi_resolvd.1m

  These are the iSCSI man pages.
- **/opt/iscsi/bin/iscsidiag**

  This is a diagnostic tool for debugging the iSCSI Software Initiator.
- **/opt/iscsi/tools/iscsi.pl**

  /opt/iscsi/tools/isw.pl

  These are iSCSI Perl scripts used during Q4 dump analysis of the iSCSI Software Initiator.

In addition, the Software Distributor will add the following iSCSI files exclusively to HP-UX revision 11i v1 host configurations:

- **/usr/conf/lib/iscsi_dbg.o**

This is the iSCSI 64 bit object module containing debug information for Q4 dump analysis of the iSCSI Software Initiator. This module will be linked into the HP-UX kernel.

- **/usr/conf/master.d/iscsi**

  This is the iSCSI master file.

Kernel Build

The SD tool will add the following iSCSI references to an HP-UX host configuration:

- **`iscsial`**

  The `iscsial` statement in the /stand/system file results in the inclusion of the iSCSI adaptation layer in the kernel build. The adaptation layer is required for any iSCSI operation to be executed.

- **`iscsi`**

  The `iscsi` statement in the /stand/system file results in the inclusion of the iSCSI transport layer in the kernel build. The iSCSI transport layer is required for any iSCSI operation to be executed.

- **`isw`**

  The `isw` statement in the /stand/system file results in inclusion of the iSCSI Software Interface Driver in the kernel build, and execution of the `iswd` daemon during system startup.

NOTE After the /stand/system file has been updated and the iSCSI Software Initiator has been successfully installed on the system, the kernel will be rebuilt and the system will be rebooted by the SD tool.

NOTE The iSCSI Software Initiator defines no kernel tunable variables.

Verifying the Installation

After the system reboots, verify that the installation was successful by following these steps:

Step 1. Issue the **swlist** command as follows:

```
# swlist iSCSI-00
```

If the HP-UX 11iv1 iSCSI Software Initiator is installed correctly, the generated output will look similar to this:

```
# Initializing...

# Contacting target "myhost"...

#

# Target:  myhost:/

#


# iSCSI-00               B.11.11.03e   HP-UX iSCSI Software Initiator
  iSCSI-00.ISCSI-SWD     B.11.11.03e   HP-UX iSCSI Software Initiator
```

If the HP-UX 11iv2 iSCSI Software Initiator is installed correctly, the generated output will look similar to this:

```
# Initializing...

# Contacting target "myhost"...

#

# Target:  myhost:/

#


# iSCSI-00                 B.11.23.03e   HP-UX iSCSI Software Initiator

  iSCSI-00.ISCSI-SWD        B.11.23.03e   HP-UX iSCSI Software Initiator
```

Step 2. Issue the **ioscan** command as follows:

```
# ioscan -kfnC iscsi
```

If the software is installed correctly, the generated output will look similar to this:

```
Class     I  H/W Path  Driver   S/W State   H/W Type     Description
=====================================================================
iscsi     0  255/0     iscsi    CLAIMED     VIRTBUS      iSCSI Virtual Node
```

If the software is not installed correctly, see "Troubleshooting the iSCSI Software Initiator Installation" on page 63.

4 Configuration

This chapter contains information on:

- “Configuring the iSCSI Software Initiator” on page 39
- “Challenge-Handshake Authentication Protocol (CHAP) Configuration” on page 43
- “Configuring CHAP Authentication Uni-directional” on page 44

- “Configuring CHAP Authentication Bi-directional” on page 46
- “Starting the iradd (iSCSI CHAP) Daemon” on page 48
- “Configuring iSCSI Service Location Protocol (SLP) Scope” on page 50

Configuring the iSCSI Software Initiator

After the iSCSI Software Initiator has been installed and the system has been rebooted, the following iSCSI-specific tasks (in addition to network setup) must be done manually to complete the system setup:

1. Add the path for iscsiutil and other iSCSI executables to the root path as:

   # PATH=$PATH:/opt/iscsi/bin

   See "The iscsiutil tool" on page 55, for more information on `iscsiutil`.

2. Configure the iSCSI initiator name.

   The iSCSI protocol mandates an initiator name for the host iSCSI node.

   iSCSI initiator names (iSCSI names) are defined in the iSCSI Qualified Name (iqn) or IEEE EUI-64 (eui) format.

   The iSCSI Software Initiator configures a default initiator name in the iqn format.

NOTE To display the iSCSI initiator name that has been configured by default, enter:

# iscsiutil -l

If the default iSCSI initiator name configured by the iSCSI Software Initiator meets your requirements, skip ahead to item 4.

If you want to change the iSCSI initiator name, read the following overview of the iqn and eui naming formats. For further details, consult the iSCSI protocol specification (RFC 3720) at: `http://www.ietf.org/rfc/rfc3720.txt`

iSCSI Qualified Name (iqn)

A default iSCSI iqn initiator name appears in the following example:

iqn.1986-03.com.hp:hpfcs214.2000853943

The string **iqn.** identifies this iSCSI initiator name as an iSCSI Qualified Name to distinguish it from an iSCSI initiator name in the "eui." format.

1986-03. is a date code in yyyy-mm format followed by a dot. This date MUST be a date during which the naming authority owned the domain name used in the iqn formated iSCSI initiator name.

com.hp is the reversed domain name of the naming authority (person or organization) that created this iSCSI initiator name.

:hpfcs214 is an optional string that must comply with a character set and length boundaries that the owner of the domain name deems appropriate. The optional string must be preceeded by a colon. The optional string may contain product types, serial numbers, host identifiers, or software keys (e.g, it may include colons to separate organization boundaries). The string following the colon (**:**) in the example above depicts the hostname (**hpfcs214**) followed by the partition identifier (**2000853943**).

With the exception of the colon prefix, the owner of the domain name can designate the content of the optional string. It is the responsibility of the naming authority to ensure the iSCSI names it assigns are unique worldwide.

For example, if the Hewlett-Packard Company owned the domain name "stor.hp.com", registered in 2001, the iSCSI qualified names that might be generated by the Hewlett Packard Company appear in the following example:

```
              Naming     String defined by
 Type  Date    Auth      "stor.hp.com" naming authority
+--++-----+ +---------+ +--------------------------------+
|  ||     | |         | |                                |

iqn.2001-04.com.hp.stor:initiator:master-host-ae12345

iqn.2001-04.com.hp.stor:storage.disk2.sys1.xyz

iqn.2001-04.com.hp.stor:storage:diskarrays-sn-a8675309

iqn.2001-04.com.hp.stor
```

IEEE EUI-64 Name (eui)

An iSCSI initiator name in the eui format appears in the following example:

```
Type  EUI-64 identifier (ASCII-encoded hexadecimal)
+--++--------------+
|  ||              |
eui.02004567A425678D
```

The format is "eui." followed by an EUI-64 identifier (16 ASCII-encoded hexadecimal digits). Any leading zeroes among the16 ASCII-encoded hexadecimal digits, must be specified.

The IEEE Registration Authority provides a service for assigning globally unique identifiers.

The IEEE EUI-64 naming format might be used when a manufacturer is already registered with the IEEE Registration Authority and uses EUI-64 formatted worldwide unique names for it's products.

Now configure the iSCSI initiator name using the following command:

```
# iscsiutil [iscsi-device-file] -i -N <iSCSI-initiator-name>
```

where

[iscsi-device-file] is the iSCSI device filepath, /dev/iscsi. It's use is optional when other options such as -i and -N are included in the command.

-i configures iSCSI initiator information.

-N is the initiator name option. When preceded by **-i**, it requires the iSCSI Initiator Name as an argument. The first 256 characters of the name string will be stored in the iSCSI persistent information.

<iSCSI-initiator-name> is the initiator name you have chosen, in iqn or eui format.

for example:

```
# iscsiutil -i -N <initiator name in iqn or eui format>
```

3. To display the initiator name for confirmation, enter:

   ```
   # iscsiutil -l
   ```

4. For each iSCSI target device that is to be statically identified, store the target device information in the kernel registry.

   Add one (or several) discovery target(s):

```
# iscsiutil [/dev/iscsi] -a -I <ip-address> [-P <tcp-port>] [-M <portal-grp-tag>]
```

where

-a adds a discovery target address into iSCSI persistent information. Only discovery target addresses can be added using this option.

-I is the option that requires the IP Address or the Hostname of the discovery target portal address as an argument.

<ip-address> is the IP Address or Hostname component of the target network portal.

[-P <tcp-port>] is the listening TCP port component of the discovery target network portal (optional). The default iSCSI TCP port number is 3260.

[-M <portal-grp-tag>] is the target portal group tag (optional). The default target portal group tag for discovery targets is 1.

for example:

```
# iscsiutil -a -I 192.1.1.110
```

or, if the Hostname of the target portal access is used:

```
# iscsiutil -a -I target.hp.com
```

If an iSCSI TCP port of the network portal used by a discovery target is different than the default iSCSI port (3260), the TCP port of the network portal used by the discovery target must be specified, for example:

```
# iscsiutil -a -I 192.1.1.110 -P 5001
```

or

```
# iscsiutil -a -I target.hp.com -P 5001
```

5. To display the discovery target(s) that have been configured, enter:

   ```
   # iscsiutil -p -D
   ```

6. To discover the operational target devices, enter:

   ```
   # /usr/sbin/ioscan -H 255
   ```

7. To create the device files for the targets, enter:

   ```
   # /usr/sbin/insf -H 255
   ```

8. To display operational targets, enter:

   ```
   # iscsiutil -p -O
   ```

All of the iSCSI login keys configured by default by the iSCSI Software Initiator apply to all of the targets connected to the iSCSI host node. Currently, there are three iSCSI login keys that may be configured by the user on a per target basis.

The three user configurable login keys are:

```
HeaderDigest
DataDigest
```

`AuthMethod` (Authentication Method)

The default value for `HeaderDigest` is:

```
None,CRC32C
```

The default value for `DataDigest` is:

```
None,CRC32C
```

The default value for `AuthMethod` is:

```
None
```

If you choose to configure any of the three user configurable login keys on a specific target, see Appendix A, “Login Key Configuration,” on page 81, for details on configuring the login keys.

Challenge-Handshake Authentication Protocol (CHAP) Configuration

Challenge-Handshake Authentication Protocol (CHAP) is an authentication protocol that defines a methodology for authenticating initiators and targets. If you do not intend to use CHAP for authentication, this aspect of the iSCSI Software Intitator configuration is not necessary and can be ignored.

The iSCSI Software Initiator has visible system administration interactions with the Challenge-Handshake Authentication Protocol (CHAP). The iSCSI Software Initiator running on HP-UX can use CHAP optionally, for authentication. The user is expected to understand the CHAP authentication method prior to its use. CHAP software is not part of the iSCSI Software Initiator.

The configuration of a RADIUS server and CHAP configuration on an iSCSI Target, is beyond the scope of this document. However, the following documentation will help you to understand the CHAP protocol and the RADIUS server installation.

Table 4-1 CHAP and RADIUS Server Documentation

Description	url
CHAP information (RFC 1994)	http://www.ietf.org/rfc/rfc1994.txt
RADIUS server documentation information (RFC 2865)	http://www.ietf.org/rfc/rfc2865.txt
RADIUS server installation information	http://www.software.hp.com -click on “security and manageability” - click on “HP-UX aaa server”

NOTE CHAP is currently the only authentication method supported by the iSCSI Software Initiator.

Configure the `AuthMethod` key with `"CHAP,None"` as the value for all Targets:

```
# iscsiutil -t authmethod CHAP None
```

During the next login negotiation, the iSCSI Software Initiator proposes `"CHAP,None"` (in its order of preference) to the iSCSI target for the `AuthMethod` login key.

The target MUST respond with the first value that it supports. The target is expected to respond to the initiator with `"CHAP"` for the `AuthMethod` login key (provided CHAP is configured properly on the target). If the target responds with `"CHAP"`, CHAP will be chosen as the authentication method. If the target responds with `"None"`, authentication will not be performed.

NOTE Currently, AuthMethod is one of the three iSCSI login keys that may be configured by the user on a per target basis. The default value for AuthMethod is “None”. If you want to configure AuthMethod on a per target basis, see “Authentication Method Configuration Examples” on page 84.

Two authentication options are available if CHAP is chosen as the authentication method:

- **Uni-directional** CHAP method:

  The target uses CHAP to authenticate the initiator. The initiator does not authenticate the target.

  The Uni-directional CHAP method does not require the use of the iradd daemon (iSCSI CHAP daemon). It also does not require configuration of a RADIUS server on the host (initiator) side.

  The default CHAP method is Uni-directional.

- **Bi-directional** CHAP method:

  The target uses CHAP to authenticate the initiator. The initiator uses CHAP to authenticate the target.

  The Bi-directional CHAP method requires the use of the iradd daemon (iSCSI CHAP daemon), as well as the configuration of a RADIUS server on the host (initiator) side.

The initiator authentication method and related attributes are configured using iscsiutil and stored persistently across reboots.

Configuring CHAP Authentication Uni-directional

The following examples illustrate configuration of CHAP once it has been selected as the authentication method that will be used.

(1) Configure for the Uni-directional authentication method:

```
# iscsiutil -u -H <chap-authentication-type> [-T <target-name>] [-I <ip-address>]   [-P <tcp-port>] [-M
<portal-grp-tag>]
```

To configure Uni-directional authentication on a global basis:

```
# iscsiutil -u -H CHAP_UNI
```

To configure Uni-directional authentication for a particular Discovery Target Address:

```
# iscsiutil -u -H CHAP_UNI -I 192.1.1.10 -M 3
```

To configure Uni-directional authentication for a particular Operational Target:

```
# iscsiutil -u -H CHAP_UNI -T iqn.2003-11.com.hp.stor:iSCSI
```

To configure Uni-directional authentication for a particular Operational Target Address:

```
# iscsiutil -u -H CHAP_UNI -T iqn.2003-11.com.hp.stor:iSCSI -I 192.1.1.1 -P 5000   -M 1
```

(2) Configure the CHAP initiator username:

```
# iscsiutil -u -N <chap-initiator-name> [-T <target-name>] [-I <ip-address>]   [-P <tcp-port>] [-M
<portal-grp-tag>]
```

If the CHAP initiator name is not configured, the iSCSI initiator name will be used instead.

To configure the CHAP initiator name on a global basis:

```
# iscsiutil -u -N mychapusername
```

To configure the CHAP initiator username for a specific Discovery Target Address:

```
# iscsiutil -u -N mychapusername -I 192.1.1.25 -M 2
```

To configure the CHAP initiator username for a specific Operational Target:

```
# iscsiutil -u -N mychapusername -T iqn.2003-11.com.hp.stor:iSCSI
```

To configure the CHAP initiator username for a specific Operational Target Address:

```
# iscsiutil -u -N mychapusername -T iqn.2003-11.com.hp.stor:iSCSI -I 192.1.1.1   -P 5000 -M 1
```

(3) Configure the initiator CHAP secret:

```
# iscsiutil -u -W <chap-initiator-secret> [-T <target-name>] [-I <ip-address>]   [-P <tcp-port>] [-M
<portal-grp-tag>]
```

The secret can be entered in two forms, ASCII and hexadecimal. Note that in the hexadecimal form, the number of hex digits must be even.

To configure the CHAP secret on a global basis:

```
# iscsiutil -u -W mychapsecret
```

or

```
# iscsiutil -u -W 0xed345ba678dfffe54e35666fa2c3c3
```

To configure the CHAP secret for a specific Discovery Target Address:

```
# iscsiutil -u -W mychapsecret -I 192.1.1.34 -M 1
```

To configure the CHAP secret for a particular Operational Target:

```
# iscsiutil -u -W mychapsecret -T iqn.2003-11.com.hp.stor:iSCSI
```

To configure the CHAP secret for a particular Operational Target Address:

```
# iscsiutil -u -W mychapsecret -T iqn.2003-11.com.hp.stor:iSCSI -I 192.1.1.1   -P 5000 -M 1
```

(4) Verification of the configured parameters:

To display authentication parameters common to all targets:

```
# iscsiutil -l
```

To display authentication parameters for all Discovery Targets:

```
# iscsiutil -pD
```

To display authentication parameters for all Operational Targets:

```
# iscsiutil -pO
```

To display authentication parameters for all Sessions:

```
# iscsiutil -pS
```

To display authentication parameters for a particular Operational Target identified by its Target Name:

```
# iscsiutil -p -T <target-name>
```

NOTE If authentication parameters are configured on a per target basis, the parameters displayed by "iscsiutil -l" are overridden by the parameters displayed by the other display commands.

Among the various authentication parameters displayed by the verification commands described above, the parameters of interest for the "Uni-directional" CHAP method are:

- Authentication Method
- CHAP Method
- Initiator CHAP Name
- CHAP Secret

NOTE CHAP Method is only valid if Authentication Method is set. The values displayed by the verification commands for the Authentication Method parameters are the values proposed by the iSCSI Software Initiator to the iSCSI target, in order of preference. The target MUST respond with the first value that it supports.

Configuring CHAP Authentication Bi-directional

(1) Configure the CHAP username and secret the same way as for the Uni-directional authentication method.

(2) Configure the NAS and RADIUS server parameters.

```
# iscsiutil -u -R <nas-hostname> <nas-secret> <radius-server-hostname>
```

where:

<nas-hostname> is the IP address or hostname of the Network Access Server (NAS). NAS operates as a client of a RADIUS server (this is the host that runs the iradd daemon). This IP address or hostname is embedded in the "Access Request" messages. The IP address may be different from the source IP address of the UDP packets sent by iradd.

<nas-secret> is the secret for the iradd daemon. This secret must be configured as the NAS secret of iradd on the RADIUS server. It is used by iradd to authenticate the RADIUS server.

<radius-server-hostname> is the IP address or hostname of the RADIUS server.

(3) Configure for the Bi-directional authentication method as follows:

```
# iscsiutil -u -H <chap-authentication-type> [-T <target-name>] [-I <ip-address>]   [-P <tcp-port>] [-M
<portal-grp-tag>]
```

To configure Bi-directional authentication on a global basis:

```
# iscsiutil -u -H CHAP_BI
```

To configure Bi-directional authentication for a particular Discovery Target Address:

```
# iscsiutil -u -H CHAP_BI -I 192.1.1.10 -M 3
```

To configure Bi-directional authentication for a particular Operational Target:

```
# iscsiutil -u -H CHAP_BI -T iqn.2003-11.com.hp.stor:iSCSI
```

To configure Bi-directional authentication for a particular Operational Target Address:

```
# iscsiutil -u -H CHAP_BI -T iqn.200-1.com.hp.stor:iSCSI -I 192.1.1.1 -P 5000   -M 1
```

(4) Verification of the configured parameters:

NOTE CHAP Method is only valid if Authentication Method is set. The values displayed by the verification commands for the Authentication Method parameters are the values proposed by the iSCSI Software Initiator to the iSCSI target, in order of preference. The target MUST respond with the first value that it supports.

To display authentication parameters common to all targets:

```
# iscsiutil -l
```

To display authentication parameters for all Discovery Targets:

```
# iscsiutil -pD
```

To display authentication parameters for all Operational Targets:

```
# iscsiutil -pO
```

To display authentication parameters for all Sessions:

```
# iscsiutil -pS
```

To display authentication parameters for a particular Operational Target identified by its Target Name:

```
# iscsiutil -p -T <target-name>
```

NOTE If authentication parameters are configured on a per target basis, the parameters displayed by "iscsiutil -l" are overridden by the parameters displayed by the other display commands.

Among the various authentication parameters displayed by the verification commands described above, the parameters of interest for the "Bi-directional" CHAP method are:

- Authentication Method
- CHAP Method
- Initiator CHAP Name
- CHAP Secret
- NAS Hostname
- NAS Secret
- RADIUS Server Hostname

Starting the iradd (iSCSI CHAP) Daemon

NOTE The Bi-directional CHAP method requires the use of the `iradd` daemon (iSCSI CHAP daemon). The Uni-directional CHAP method does not require the use of the `iradd` daemon (iSCSI CHAP daemon).

To start the `iradd` daemon:

```
# iradd
```

Once the `iradd` daemon has been started, the `iradd` daemon will be restarted automatically each time the system reboots.

NOTE For more information on `iradd`, see “iSCSI Challenge-Handshake Authentication Protocol Daemon (iradd)” on page 60.

Configuring iSCSI Service Location Protocol Daemon "Start on Boot"

The iSCSI Software Initiator uses the iSCSI Service Location Protocol daemon (islpd) for SLPv2 based dynamic target discovery. By default, the islpd daemon is not started on boot. The user may configure the "Start on Boot" feature for islpd either to enable auto-start or disable auto-start of islpd daemon on boot.

To configure the islpd "Start on Boot", use the following options:

1. To enable auto-start of the `islpd` daemon on boot

   # /opt/iscsi/bin/islpd -a

2. To disable auto-start of the `islpd` daemon on boot

   # /opt/iscsi/bin/islpd -r

3. To check the configuration of the "Start on Boot" feature

   # /opt/iscsi/bin/islpd -g

NOTE Enabling the auto-start of `islpd` daemon on boot will not automatically start the `islpd` daemon. Explicit invocation of the islpd daemon is required to start the islpd daemon.

4. To enable auto-start of the islpd daemon for future boots and to start the `islpd` daemon for the current boot, execute the following commands:

   # /opt/iscsi/bin/islpd -a

   # /opt/iscsi/bin/islpd

Configuring iSCSI Service Location Protocol (SLP) Scope

The iSCSI Software Initiator uses SLP for dynamic Target Discovery. The SLP scope is used to control the availability of service advertisements. The iSCSI SLP User Agent (UA) can be configured with one or more scope strings. If no specific scope string is configured, the scope string will be "default".

To configure the iSCSI SLP Scope List:

1. Enter:

   # iscsiutil -g -F <iscsi_slp_scope_string_list>

   If more than one scope string is to be configured, a comma (,) is used to separate individual scope strings. For example:

   # iscsiutil -g -F iscsi-scope1, iscsi-scope2 , ...

2. To display the iSCSI SLP Scope List that has been configured, enter:

   # iscsiutil -l

5 Management

This chapter contains information on:

- “iSCSI Transport Statistics” on page 58
- “Diagnostic Messages” on page 59
- “iSCSI Software Initiator Daemons” on page 60
- “iSCSI Software Initiator Name Resolution Daemon (iscsi_resolvd)” on page 60
- “iSCSI Challenge-Handshake Authentication Protocol Daemon (iradd)” on page 60
- “iSCSI Service Location Protocol Daemon (islpd)” on page 60

SAM

NOTE For detailed information about SAM, see *Using System Administration Manager (SAM)*. This document can be viewed or downloaded at http://www.docs.hp.com, or a hard copy can be ordered from HP.

The HP-UX System Administration Manager (SAM) has been modified to support the following iSCSI functionality:

- Recognition of the iSCSI Software Initiator

  SAM will recognize the iSCSI virtual bus and display it under Peripheral Devices -> Device List.

- Inclusion and removal of the iSCSI Software Initiator in the /stand/system file

  The HP-UX iSCSI Software Initiator may be included in the HP-UX kernel and is configurable through SAM. System Administrators can add/remove iSCSI Software Initiator components (`iscsi`, `isw`, `iscsial`) from the kernel using SAM -> Kernel Configuration -> Drivers.

- Addition and deletion of static iSCSI targets

  The HP-UX iSCSI Software Initiator supports static discovery of iSCSI targets. For HP-UX static scans, a set of devices called iSCSI discovery targets is defined in the kernel registry (a persistent store) and that set of devices is scanned for existence of storage accessible by the host.

  SAM provides system administrators with a user interface to view statically configured iSCSI targets, add new iSCSI discovery targets and delete existing iSCSI discovery targets from the kernel registry. This functionality can be accessed from SAM -> Peripheral Devices -> iSCSI.

- Storage Management

  Storage accessible by the HP-UX iSCSI Software Initiator can be managed using the HP-UX SAM tool. All storage and alternate paths in different subnets are accessible and configurable through SAM in a manner similar to the access and configuration of existing storage.

ioscan

After the iSCSI Software Initiator is installed, a virtual node will appear in the ioscan output. This virtual node will appear as follows:

```
                      iscsi   0       255/0
```

When data is available for valid iSCSI targets, the output of the ioscan command for iSCSI targets will be similar to the following example:

```
iscsi   0  255/0         iscsi    CLAIMED  VIRTBUS   iSCSI  VirtualNode
ext_bus 2  255/0/0.0     iscsial  CLAIMED  INTERFACE iSCSI-SCSIProtocolInterface
target  5  255/0/0.0.0   tgt      CLAIMED  DEVICE
disk    2  255/0/0.0.0.0 sdisk    CLAIMED  DEVICE    <<a disk description>>
disk    3  255/0/0.0.0.1 sdisk    CLAIMED  DEVICE    <<a disk description>>
```

The first line of the sample ioscan output displays the iSCSI virtual node. This is the root node for all iSCSI storage and will occur only once in the ioscan output. The iSCSI transport driver claims the iSCSI root node.

The second line of the sample ioscan output displays the initiator session identifier instance (ISID) and the SCSI-2 virtual bus. This implies that the ioscan operation was able to successfully establish a discovery session (session instance is 0) with the iSCSI target identified in the registry. It also implies that storage was defined behind the iSCSI target. The storage behind the target was defined in the SCSI-3 range of LUNs 0-127; therefore, virtual bus 0 was created. The driver iscsial (iSCSI adaptation layer) claimed the bus as an iSCSI virtual bus, and the iscsial driver component will control operations to this bus.

The third line of the sample ioscan output displays a SCSI-2 target. SCSI-2 permits 16 targets per bus, therefore, every eighth LUN on the iSCSI target (using SCSI-3) maps to a new SCSI-2 target.

The fourth and fifth lines of the sample ioscan output display SCSI-2 LUNs. SCSI-2 defines 8 LUNs per target, therefore, every eighth LUN on the iSCSI target will map to LUN 0 for a new SCSI-2 target. The SCSI class drivers, in this case the sdisk class drivers, claim the disk LUNs.

The iscsiutil tool

The `iscsiutil` command is a management and diagnostic tool used with the iSCSI Software Initiator.

The `iscsiutil` tool provides a command line interface to:

— Configure the iSCSI Software Initiator related parameters.

— Display statistics for the interface driver, connection processing, session processing, and the discovery sequence.

— Execute diagnostic functionality.

Device files are automatically created to access the iSCSI transport and interface drivers.

The device file to interface to the iSCSI transport driver is `/dev/iscsi`, which is created during installation.

The iSCSI transport driver defines iSCSI session management common to all iSCSI interface drivers.

The iSCSI transport driver defines interfaces with the SCSI services layer.

NOTE For a complete list of options supported by the `iscsiutil` tool, review the `iscsiutil` manpage on an HP-UX system that has the iSCSI Software Initiator installed on it. From the command line enter:

```
# man iscsiutil
```

iscsiutil Command Utilization

Most command sequences require root or super user permission.

NOTE The parameters enclosed within [ ] are optional. The parameters enclosed within < > are mandatory.

Configuration with iscsiutil

To add an iSCSI discovery target IP address or a target portal group tag to the kernel registry, use:

```
# iscsiutil [/dev/iscsi] -a -I <ip-address> [-P <tcp-port>][-M <portal-grp-tag>]
```

The iSCSI target will be probed during `ioscan` processing using the static scanning technique described in “Device Discovery” on page 25.

`<ip-address>` The ip_address specified can be an IPv4 formatted address or a DNS host name. The bracketed [ ] parameters are optional.

`<tcp-port>` A TCP port may be specified for the iSCSI target. If no port is specified, the default iSCSI port (3260) will be used.

`<portal-grp-tag>` The target portal group tag may be specified (a value from 0 to 65535 inclusive); otherwise, a default value of 1 will be used.

To Delete an iSCSI discovery target IP address from the kernel registry, use:

```
# iscsiutil [/dev/iscsi] -d -I <ip-address> [-P <tcp-port>][-M <portal-grp-tag>]
```

<ip-address> The ip_address specified can be an IPv4 formatted address or a DNS name

<tcp-port> A TCP port may be specified for the iSCSI target. If no port is specified, the default iSCSI target port (3260) will be used.

<portal-grp-tag> The target portal group tag may be specified (a value from 0 to 65535 inclusive); otherwise a default value of 1 will be used.

To add an initiator name and optionally add an initiator alias, use:

```
# iscsiutil  [/dev/iscsi] -i  -N  <initiator-name> [-A <initiator-alias>]
```

(*names must be in "iqn" or "eui" format*)

See "Configuring the iSCSI Software Initiator" on page 39, for more information on iqn and eui naming formats.

Management with iscsiutil

To display statistics from the iSCSI Software Initiator, use:

```
# iscsiutil [/dev/iscsi] -s [-G] [-S [<ssn_inst> [-C [<cid>]]]]
```

where:

-G displays the global statistics.

-S displays the session statistics for all sessions; if an initiator session identifier instance (ssn_inst) is specified, then statistics will be displayed for that session only.

-C displays statistics for all connections of the specified session instance (ssn_inst); if a connection identifier (cid) is specified, then statistics will be displayed for that connection.

To display the iSCSI name and alias for the iSCSI initiator node, as well as the current authentication method and login key information, use:

```
# iscsiutil [/dev/iscsi] -l
```

To display the transport driver name and version, use:

```
# iscsiutil [/dev/iscsi]
```

To display information about current sessions, targets, and connections, use:

```
# iscsiutil [/dev/iscsi] -p [-O | -D | [-T <target-name>]]
```

or

```
# iscsiutil [/dev/iscsi] -p [-S [<ssn_inst> [-C [<cid>]] [-V]]]
```

where:

-D displays all current discovery target information.

-O displays all current operational targets and related information.

-T displays information for the iSCSI target-name specified.

-S displays the session information for all sessions; if an initiator session identifier instance (`ssn_inst`) is specified, then information will be displayed for that session only.

Session information displayed includes:

- The session instance number.
- The target iSCSI name.
- The target IP address and port.
- The target portal group tag.

-C displays statistics for all connections of the specified session instance (`ssn_inst`); if a connection identifier (`cid`) is specified, only the statistics for that connection will be displayed.

-V displays negotiated login key information.

To display iscsiutil usage, use:

```
# iscsiutil
```

To issue an iSCSI NOP-OUT indicating the destination session and connection, use:

```
# iscsiutil [/dev/iscsi] -n  -S  <ssn_inst> [-C <cid> ] [-L <data-size>]
```

where:

-S provides an initiator session identifier instance (`ssn_inst`) on which the iSCSI NOP-OUT is to be sent

-C provides a specific connection identifier (`cid`) for the iSCSI NOP-OUT

-L is the <data-size>, or number of bytes, to be used for the NOP-OUT. The valid range for the data-size is 0 to 4096 bytes. The default data-size is 64 bytes.

iSCSI Transport Statistics

The iSCSI Software Initiator maintains a variety of transport statistics, which are accessible through the `iscsiutil` tool. For a detailed listing and explanation of the iSCSI transport statistics see Appendix B, “Transport Statistics,” on page 87.

Diagnostic Messages

The HP-UX iSCSI Software Initiator works with HP-UX Event Monitoring Services (EMS) and the Support Tools Manager (STM). By default, the iSCSI Software Initiator logs all diagnostic messages to the STM log files. STM can be used to view the diagnostic messages logged in the STM log files.

EMS can be configured to automatically notify the system administrator when diagnostic messages are logged by the iSCSI Software Initiator.

In addition to logging all diagnostic messages to the STM log files, the iSCSI Software Initiator logs some of the diagnostic messages to the /var/adm/syslog/syslog.log file.

For a detailed listing and explanation of messages that the iSCSI Software Initiator can generate, see Appendix C, “Diagnostic Messages,” on page 95.

iSCSI Software Initiator Daemons

iSCSI Software Initiator Name Resolution Daemon (iscsi_resolvd)

iscsi_resolvd is a user level daemon responsible for resolving a hostname to an IP address. This daemon operates by receiving hostnames from the iSCSI Software Initiator and returning the corresponding IP addresses to the iSCSI Software Initiator. This daemon is automatically started from the iSCSI Software Initiator start-up script during HP-UX system start-up.

iSCSI Challenge-Handshake Authentication Protocol Daemon (iradd)

CHAP authentication for iSCSI uses the iradd daemon. The iradd daemon allows the iSCSI Software Transport driver to communicate with a RADIUS server.

The iradd daemon helps the initiator authenticate targets. It is not used for Uni-directional CHAP authentication, where the initiator is authenticated by the target.

The iradd daemon receives authentication requests from the iSCSI Software Transport driver. It forwards the authentication requests to the RADIUS server via a RADIUS message "Access Request". The RADIUS server responds to iradd with a RADIUS message, "Access Accept" (if the authentication of the target succeeded), or with "Access Reject" (if the authentication failed). The iradd daemon then passes the result to the iSCSI Software Transport driver.

The iradd daemon has the following command line, which is executed in the iSCSI Software Initiator startup script, once CHAP has been requested through iscsiutil:

```
# /opt/iscsi/bin/iradd [-r <retries>] [-t <timeout>] [-i <id>]
```

where:

-r retries is the number of retries iradd will perform until a reply is received from the RADIUS server. The unreliable UDP protocol is used between iradd and the RADIUS server. The default value is 10.

-t timeout is the timeout in seconds between retry attempts. The default value is 3.

-i id is the starting message identifier (number) used in the RADIUS messages between iradd and the RADIUS server. The default value is the process id of iradd.

NOTE For more information on CHAP see “Challenge-Handshake Authentication Protocol (CHAP) Configuration” on page 43.

iSCSI Service Location Protocol Daemon (islpd)

The islpd daemon is a user space daemon that implements the Service Location Protocol (SLP) User Agent (UA) and the interface to the iSCSI transport driver. The iSCSI transport driver adds discovered iSCSI targets to the persistent information maintained in the kernel registry. Dynamic scanning based on SLP is triggered when the ioscan command is executed. The UA queries the DA to obtain a list of all available iSCSI targets on the network that are known to the DA. Each target address discovered is registered in the kernel registry on the host by the iSCSI transport driver. For more information on SLP see “Service Location Protocol Based Dynamic Discovery” on page 26.

6 Troubleshooting

This chapter contains information on:

- “Troubleshooting the iSCSI Software Initiator Installation” on page 63
- “Troubleshooting Undetected Target Devices” on page 64
- “Troubleshooting issues with iswd daemon” on page 67

- “Diagnostics” on page 68

Troubleshooting the iSCSI Software Initiator Installation

If you have attempted to install the iSCSI Software Initiator, but you suspect there is a problem, first review "Verifying the Installation" on page 35, to ensure the HP-UX iSCSI Software Initiator has been installed successfully. If the HP-UX iSCSI Software Initiator installation has failed, follow the techniques below to troubleshoot the issue.

1. Verify that the HP-UX system you are attempting to install the iSCSI Software Initiator on is running an OS release that is supported by the iSCSI Software Initiator.
2. Review the *HP-UX iSCSI Software Initiator Release Notes* to obtain a list of required patches.
3. Review "Locating and Installing the iSCSI Software Initiator" on page 31, to ensure the iSCSI Software Initiator has been successfully downloaded and the installation instructions have been followed.
4. If there is an issue with locating the iSCSI Software Initiator, or downloading it from the HP Software Depot, see Software Depot frequently asked questions at:

   `http://www.software.hp.com/portal/swdepot/faqcategory.do`

   or contact SW Depot Customer Service at:

   `http://www.software.hp.com/portal/swdepot/feedback.do`

   for further assistance.
5. Review the logfile `/var/adm/sw/swagent.log` for errors encountered during the installation. Typical causes of installation errors are:

   — Lack of disk space on the `"/"` (root), `"/var"` (if /var is a separate file system) and `"/stand"` (boot) file systems. Make sure the disk space estimated for the iSCSI Software Initiator (displayed in the `/var/adm/sw/swagent.log` file) is available before retrying the installation.

   — A solution for some errors may be suggested in the `/var/adm/sw/swagent.log` file. If so, follow those instructions to rectify the problem.
6. If there are kernel build errors, ensure that the steps (1) and (2) above have been followed. One of the causes for build failure is a lack of disk space in the `"/stand"` (boot) filesystem. Increase the disk space in `the "/stand"` (boot) filesystem and retry the installation. If you still have build errors, contact HP support for assistance.
7. Review the `/etc/rc.log` file and search for "HP-UX iSCSI" to find the output of the iSCSI start-up script: `/sbin/rc2.d/S532iscsi`. Make sure there were no errors, while the iSCSI Software Initiator was initializing, during system start-up. If the following errors are seen, the iSCSI Software Initiator installation has not completed successfully:

   ```
   iscsiutil: iSCSI stack is not loaded

   iscsiutil: (/dev/iscsi) Wrong device file

   iscsiutil: (/dev/isw) Wrong device file
   ```
8. Verify that the entries listed in "Kernel Build" on page 34, are present in the `/stand/system` file.
9. Verify that the files listed in "iSCSI Software Initiator Components" on page 32, are installed in the correct directories on the HP-UX system.
10. Verify that there are no kernel build errors (see step 5 for details).
11. If the problem is still not resolved, try installing the iSCSI Software Initiator again.

Troubleshooting Undetected Target Devices

1. Verify that all of the required patches or superseding patches for the iSCSI Software Initiator have been installed. Review the *HP-UX iSCSI Software Initiator Release Notes* to obtain a list of required patches.
2. If a target device cannot be seen from the HP-UX host, verify that the problem is not an infrastructure connectivity issue by executing the `/usr/sbin/ping` command from the HP-UX host, specifying the IP address or qualified domain name of the iSCSI target. If ping is not successful, verify that the target device is available, powered on, and properly connected to the network.
3. If the iSCSI discovery target does not appear to respond:

— Verify that the IP address or domain name of the discovery target was entered correctly by executing: **`iscsiutil -p -D`**

— Verify that the iSCSI Name is valid.

The iSCSI Initiator Name or iSCSI Target Name is referred to as an iSCSI Name. The iSCSI Software Initiator configures a default Initiator Name in the "iqn" format. If you have changed the Initiator Name from its default using `iscsiutil` and the Initiator Name is in an invalid format, iSCSI login negotiation could fail leaving target devices inaccessible from the HP-UX host.

The iSCSI Software Initiator validates iSCSI Names. If a target reports a Target Name in an "iqn", "eui" or "naa" format, but it does not comply with the iSCSI Name format rules, or if a Target Name contains a prohibited character, iSCSI Login negotiation will fail and the target will not be seen by the HP-UX host.

NOTE The iSCSI Name format rules are detailed in “Configuring the iSCSI Software Initiator” on page 39. Prohibited characters are described in section 6.2 of RFC 3722, "String Profile for Internet Small Computer Systems Interface (iSCSI) Names". Consult RFC 3722 at: `http://www.ietf.org/rfc/rfc3722.txt`

The following characters MUST NOT be used in iSCSI names:

0000-002C; [ASCII CONTROL CHARACTERS and SPACE through ,]

002F; [ASCII /]

003B-0040; [ASCII ; through @]

005B-0060; [ASCII [through ‘]

007B-007F; [ASCII { through DEL]

You will need to work with the target vendor representative to correct an invalid Target Name format.

— Verify that the fully qualified domain name, or the hostname, of the discovery target is known (if used instead of IP address) using `nslookup`.

— Confirm the TCP port and Target Portal Group Tag are correct for the iSCSI discovery target in question. The values entered, or used by default by the iSCSI Software Initiator, can be seen by executing: **`iscsiutil -p -D`**

4. Verify that the steps listed in “Configuring the iSCSI Software Initiator” on page 39, have been followed correctly and completed successfully.

5. Issue the `iscsiutil -sG` command and look at the following statistics:

   Number of Discovery session open failures

   If there is a non-zero value for this statistic, determine the cause of failure by either looking at the message logged in the `/var/adm/syslog/syslog.log` file or by monitoring the `EMS/STM` log files. The most common cause for failure would be an incorrect configuration of the components of the iSCSI target address:

   `<ip-address>, <tcp-port> and <portal-grp-tag>` or an iSCSI Login Negotiation failure.

   Number of Normal session open failures

   If there is a non-zero value for this statistic, determine the cause of failure by either looking at the message logged in the `/var/adm/syslog/syslog.log` file or by monitoring the `EMS/STM` log files. The most common cause for failure would be an incorrect configuration of the authentication and digest methods. Refer to Appendix A, “Login Key Configuration,” on page 81, for configuration steps.

   Number of SCSI INQUIRY commands issued that failed

   If there is a non-zero value for this statistic, determine the cause of failure by either looking at the message logged in the `/var/adm/syslog/syslog.log` file or by monitoring the `EMS/STM` log file. Some of the causes for failure are:

   — The iSCSI session and/or connection is not in an online state, resulting in the `SCSI Inquiry` command not being sent to the target.

   — The target device terminated the `SCSI Inquiry` command with a `Check Condition`, `Busy`, or `Reservation Conflict` status.

   — The target device did not respond to the `SCSI Inquiry` command sent by the initiator within a stipulated amount of time.

   — The target failed to execute the `SCSI Inquiry` command.

   — The `SCSI Inquiry` command could not be issued due to memory resource constraints.

   Review the detailed "Cause and Action" messages logged in the `EMS/STM` log file and take the necessary action.

6. Target devices will not be seen if there are `iSCSI Login Negotiation` failures. Some of the causes for login failures are; protocol violation by the target, initiator login errors, or lack of memory on the initiator.

   Determine the cause of login failures by either looking at the message logged in the `/var/adm/syslog/syslog.log file` or by monitoring the `EMS/STM` log file.

   The detailed “`Cause and Action`” messages logged in the `EMS/STM` log file provide guidance on the necessary action to take. If `EMS/STM` is not set up and the only source of message logging is `syslog.log`, review the Appendix , “Diagnostic Messages,” on page 96, to determine the action to be taken.

7. iSCSI uses `ext_bus` instances from the same pool as Fibre Channel and Parallel SCSI. HP-UX is limited to 256 total allocations of `ext_bus` instances. The addition of iSCSI busses is a use of this resource. If you see the “`Attempt to allocate more than the maximum number of sessions`” message in the `/var/adm/syslog/syslog.log` file, the number of targets attached to the HP-UX system may need to be

reduced, or if there are unused bus instance entries, it may be necessary to change (compress) bus instance assignments. This should only be done with the assistance of an HP support representative. Contact HP support for further assistance.

8. When authentication is desired, if some targets are not seen due to authentication failure:

   - Verify that the iradd daemon is running (if Bi-Directional CHAP authentication is desired).
   - Verify that the configuration steps in “Challenge-Handshake Authentication Protocol (CHAP) Configuration” on page 43, are followed.

9. When SLP based discovery is used and some targets are not seen in spite of the iSCSI islpd daemon starting successfully (as per the `/etc/rc.log and /var/adm/syslog/syslog.log` files), please verify the SLP configuration on the iSCSI target.

Troubleshooting issues with iswd daemon

The iswd daemon is involved in the connection management for the iSCSI Software Interface Driver. It opens and terminates TCP connections interfacing with the iSCSI Software Interface Driver.

CAUTION The iswd deamon is required to access the iSCSI devices. To avoid upredictable results, the deamon must not be terminated.

If a target device cannot be seen from the HP-UX host, verify that the iswd daemon is alive by executing

```
# ps -ef | grep iswd
```

If the iswd daemon is not alive, restart the daemon and re-issue an ioscan by executing

```
# /opt/iscsi/bin/iswd
# /usr/sbin/ioscan -H 255
```

If the iswd daemon is accidentally terminated the behavior displayed depends on the state of the iswd daemon. The most likely scenario would be:

1. An attempt to terminate (kill) the iswd daemon will have no effect initially as the daemon thread is sleeping in the kernel waiting for connection open requests. The active outstanding connections will not be affected as long as the daemon thread is in kernel space.
2. The signal sent by the kill(1) command is queued and serviced once the daemon thread goes to user space upon an enqueue of a connection open request, causing the iswd daemon to be terminated. A Connection open request is enqueued when a command requiring iSCSI target access is executed.

The less likely scenario would be

1. An attempt to terminate (kill) the iswd daemon when the thread is in user space will cause the daemon to be terminated immediately.

ACTION: Restart the iswd daemon in both the above scenerios

The following message will be displayed during execution of shutdown(1m) or reboot(1m).

```
reboot: CAUTION: some process(es) wouldn't die
```

This message is the result of the iswd daemon being kept alive to complete various tasks during reboot. When iSCSI is configured in a system, this message may be ignored.

Diagnostics

The HP-UX Support Tool Manager (STM) has been modified so a user can select the iSCSI virtual node entries in the STM map. However, when iSCSI devices are selected, all of the tools are grayed-out (not available).

The HP-UX iSCSI Software Initiator works with HP-UX Event Monitoring Services (EMS) and the Support Tools Manager (STM). By default, the iSCSI Software Initiator logs all diagnostic messages to the STM log files. STM can be used to view the diagnostic messages logged in the STM log files.

EMS can be configured to automatically notify the system administrator when diagnostic messages are logged by the iSCSI Software Initiator.

In addition to logging all diagnostic messages to the STM log files, the iSCSI Software Initiator logs some of the diagnostic messages to the /var/adm/syslog/syslog.log file. For more details on diagnostic messages, see Appendix C, “Diagnostic Messages,” on page 95.

If a problem cannot be resolved using the troubleshooting techniques listed in this chapter, provide the log file generated by the **iscsidiag** tool to HP Support.

iscsidiag is an iSCSI Software Initiator debug information gathering tool.

Execute **/opt/iscsi/bin/iscsidiag** to capture the debug information.

The logfile will be placed in the **/tmp/iscsidiag** directory with a filename of: **iscsidiag.<pid>.log**

If you want to place the log file in an alternate directory, invoke **iscsidiag** as: **/opt/iscsi/bin/iscsidiag -t <directory_name>**

Provide the log file **iscsidiag.<pid>.log** to HP Support for further assistance. The log file will be located at the default location (/tmp/iscsidiag) or at an alternate location, if an alternate location has been specified using the **-t** option.

7 The iSCSI Software Interface Driver

This chapter contains information on:

iSCSI Software Interface Driver Overview

The HP-UX iSCSI Software Interface Driver (SWD) is a host based implementation of the iSCSI protocol that uses standard Network Interface Cards (NICs). There are no special network infrastructure requirements.

The iSCSI SWD interfaces with the iSCSI transport layer for session management and SCSI support. It also interfaces with the network stack above the TCP layer. iSCSI Protocol Data Units (PDUs) are sent and received by the iSCSI SWD on the TCP stream.

The iSCSI SWD receives SCSI commands and data from the iSCSI transport layer and sends SCSI responses to the iSCSI transport layer.

iSCSI Software Interface Driver Technical Overview

System Startup

In the standard HP-UX system startup sequence, access to mass storage is established before networking is initialized. This order of events is in conflict with the iSCSI SWD, because iSCSI target devices cannot be accessed prior to networking initialization.

When the iSCSI SWD attempts to:

— discover iSCSI targets

— activate volume groups with physical volumes on iSCSI targets

— access file systems on iSCSI targets

before network initialization is completed, the ENETUNREACH errno is returned to the caller. The Logical Volume Manager (LVM) will not generate an error message when it sees this errno.

Administrative commands that access iSCSI targets will also return the ENETUNREACH errno until networking is initialized. Upon seeing the errno, the administrative commands will not generate an error message.

Once the first attempt at network initialization is complete, the ENETUNREACH errno will no longer be returned. Any errno that is returned will be the same as if iSCSI were not installed in the system.

The iSCSI Software Intitiator startup script has been modified to perform iSCSI target access operations a *second* time after networking has been initialized.

The resolution of the ordering problems described above has placed limitations on the iSCSI SWD. Because network initialization is performed using the /var directory, the /var directory cannot be on an iSCSI target. Also, the boot, root, primary swap, and dump file systems are not supported on iSCSI volumes.

iSCSI Software Interface Driver Technical Specifications

Features

- The iSCSI SWD conforms to the iSCSI Protocol Specification (RFC 3720).
- The iSCSI SWD will function over standard NICs (Network Interface Cards).
- The iSCSI SWD will be a compute-intensive driver.

Limitations

- The iSCSI SWD does not support the definition of the boot, root, primary swap, dump, or var volumes on iSCSI logical units.
- The iSCSI SWD supports only IPv4 addresses.

iSCSI Software Interface Driver Configuration

Existing networking commands must be used to establish routing information and LAN configuration for host based networking.

The user is expected to understand the Service Location Protocol (SLP) dynamic scanning technique in order to use it. SLP is a software component separate from the iSCSI SWD. SLP is used optionally by iSCSI for maintaining a dynamic name server (DNS) capability. See Table 2-1, “SLP Server Installation Information and Related Documents,” on page 27, for more SLP information.

Kernel Build

The isw statement in the /stand/system file results in the inclusion of the iSCSI Software Interface Driver module, as well as the execution of the iswd daemon, during system startup.

iSCSI Software Interface Driver Management

If the `iscsiutil` command is directed to the `/dev/isw` device file, the command is explicitly directed to the iSCSI SWD. Most command sequences require root or super user permission. The `iscsiutil` command can be used to:

- Display the iSCSI Software Interface Driver statistics:

  ```
  # iscsiutil /dev/isw -s -G

  -G displays the iSCSI Software Interface Driver global statistics.
  ```

- Clear the iSCSI Software Interface Driver statistics:

  ```
  # iscsiutil   /dev/isw   -c  -G
  ```

- Display the iSCSI Software Interface Driver name and version:

  ```
  # iscsiutil   /dev/isw
  ```

The `iscsiutil` tool requires the `/dev/isw` device file for its access to the iSCSI Software Interface Driver. This file will be created when the iSCSI Software Interface Driver is installed.

Device files are created to access the iSCSI transport and Software Interface Drivers:

> The iSCSI Software Interface Driver provides a low level interface to the network stack.
>
> The `/dev/isw device` file, which is created during installation, provides an interface to the iSCSI Software Interface Driver.

iSCSI Software Interface Driver Statistics

Statistics are maintained in the iSCSI Software Interface Driver (SWD). For a detailed listing and explanation of the iSCSI Software Interface Driver statistics see Appendix D, “iSCSI Software Interface Driver Statistics,” on page 105.

iSCSI Software Interface Driver Diagnostic Messages

The HP-UX iSCSI Software Interface Driver works with HP-UX Event Monitoring Services (EMS) and the Support Tools Manager (STM). By default, the iSCSI Software Interface Driver logs all diagnostic messages to the STM log files. STM can be used to view the diagnostic messages logged in the STM log files.

EMS can be configured to automatically notify the system administrator when diagnostic messages are logged by the iSCSI Software Interface Driver.

In addition to logging all diagnostic messages to the STM log files, the iSCSI Software Interface Driver logs some of the diagnostic messages to the /var/adm/syslog/syslog.log file.

For a detailed listing and explanation of messages that the iSCSI Software Interface Driver can generate, see Appendix E, “iSCSI Software Interface Driver Diagnostic Messages,” on page 117.

iSCSI Software Interface Driver Daemons

iSCSI Software Interface Driver Daemon (iswd)

The iSCSI SWD daemon, `iswd`, is a user level process that communicates with the iSCSI SWD component and the network sockets interface. The iSCSI SWD daemon will receive requests to open or close a connection, then it will use the existing application network socket interface to perform the connection open/close operation. Once a connection has been established, the iSCSI SWD daemon will use a system call to inform the iSCSI SWD that the connection is available (on open), the connection attempt failed (on open), or the connection was successfully closed. Threads forked from the daemon will perform individual connection establishment.

When the iSCSI SWD daemon initializes, it will bind a module into the connection path in the kernel. Requests to and from the TCP stack will pass through the kernel component. The kernel component will be provided with an entry point into the iSCSI SWD, and, on successful open, the iSCSI SWD will be provided with an entry point into the kernel module. Once the open is complete, data transfers can be made between the iSCSI SWD and kernel module without iSCSI SWD daemon involvement, thus eliminating copies between user and kernel space.

A Login Key Configuration

Configuring iSCSI Login Keys

In accordance with the iSCSI protocol, an iSCSI initiator must negotiate iSCSI login keys with each iSCSI target to:

— enable an iSCSI connection

— authenticate the parties

— negotiate the session’s parameters

— mark the connection as belonging to an iSCSI session

All of the iSCSI login keys configured by default by the iSCSI Software Initiator apply to all of the targets connected to the iSCSI host node. Currently, there are three iSCSI login keys that may be configured by the user on a per target basis.

The three user configurable login keys are:

`HeaderDigest`

`DataDigest`

`AuthMethod` (Authentication Method)

The default value for `HeaderDigest` is:

`None,CRC32C`

The default value for `DataDigest` is:

`None,CRC32C`

The default value for `AuthMethod` is:

`None`

The user configurable login keys may be configured in one of the following ways:

— For all targets on a global basis

— For a particular Operational target.

— For a particular Discovery or Operational target address.

NOTE A login key configured for all targets, applies to all targets and all target addresses that are not already configured for that login key.

A login key configured for a particular Operational target will apply to all Operational target addresses that are not already configured for that login key.

Unique iSCSI Sessions are opened for each Operational target address. The login keys configured for a particular Operational target address correspond to the unique iSCSI Session opened on that Operational target address.

The usage to configure the login values for the user configurable login keys is as follows:

```
# iscsiutil [iscsi-device-file] -t <login-key> <login-val-1> <login-val-2> ...
<login-val-N> [-T <target-name> [-I <ip-address>] [-P <tcp-port>]
[-M <portal-grp-tag>]]
```

The order in which the login values are listed in this command defines the order in which the iSCSI initiator proposes them to the target. The login keys and values are not case-sensitive.

The configurable login keys currently available are:

- **HeaderDigest**

  A header digest that can be negotiated during iSCSI login. The list of parameters that the `HeaderDigest` key can accept are:

  `None`

  `CRC32C`

- **DataDigest**

  A data digest that can be negotiated during iSCSI login. The list of values that the `DataDigest` key can accept are:

  `None`

  `CRC32C`

- **AuthMethod**

  An authentication method that can be negotiated during iSCSI login. The list of values that the `AuthMethod` key can accept are:

  `None`

  `CHAP`

The order of precedence of login key configuration is as follows:

(a) iSCSI Target Address level

(b) iSCSI Target level

(c) Global level

Configuration of login keys at the Discovery target level is not possible, because Discovery targets are not identified by iSCSI target names.

Configuration of login keys for Discovery target addresses does not involve defining the iSCSI target names. For Operational targets (addresses), the target names should be defined.

NOTE One or two login key values may be specified for each of the login keys listed above, however, no more than two login key values can be specified for a login key.

Header and Data Digest Configuration Examples

Configure the `HeaderDigest` key with "`None,CRC32C`" as the value for all targets:

```
# iscsiutil -t headerdigest None CRC32C
```

Configure the `DataDigest` key with "`CRC32C,None`" as the value for a particular Discovery Target Address which is already configured:

```
# iscsiutil -t datadigest CRC32C None -I 192.1.1.58 -M 2
```

Configure the `DataDigest` key with "`CRC32C,None`" as the value for an Operational Target:

```
# iscsiutil -t datadigest CRC32C None -T iqn.2003-11.com. hp.stor:iSCSI.Storage
```

Configure the HeaderDigest key with "CRC32C" as the value for an Operational Target Address:

```
# iscsiutil -t headerdigest CRC32C -I 192.1.1.58 -M 2 -T
iqn.2003-11.com.hp.stor:iSCSI.Storage
```

Configure both the HeaderDigest and the DataDigest keys with "CRC32C" as the value for an Operational Target Address:

```
# iscsiutil -t bothdigest CRC32C -I 192.1.1.58 -M 2 -T iqn.
2003-11.com.hp.stor:iSCSI.Storage
```

Authentication Method Configuration Examples

Configure the AuthMethod key with "CHAP,None" as the value for all Targets:

```
# iscsiutil -t authmethod CHAP None
```

Configure the AuthMethod key with "CHAP" as the value for a particular Discovery Target Address, which is already configured:

```
# iscsiutil -t authmethod CHAP -I 192.1.1.58 -M 2
```

Configure the AuthMethod key with "None,CHAP" as the value for an Operational Target:

```
# iscsiutil -t authmethod None CHAP -T iqn.2003-11.com.hp. stor:iSCSI.Storage
```

Configure the AuthMethod key with "CHAP" as the Authentication Method for an Operational Target Address:

```
# iscsiutil -t authmethod CHAP -I 192.1.1.58 -M 2 -T iqn.2003-11.com.hp.stor:iSCSI.Storage
```

Displaying Login Keys

NOTE The values displayed for the login keys by the following commands are the values proposed by the iSCSI Software Initiator to the iSCSI target, in order of preference. The target MUST respond with the first value that it supports. If the login keys are configured on a per target basis, the login keys displayed by "iscsiutil -l" are overridden by the login keys displayed by the other display commands.

To verify the login key configuration, execute any of the following login key display commands.

To display login keys common to all Targets:

```
# iscsiutil -l
```

To display login keys of all Discovery Targets:

```
# iscsiutil -pD
```

To display login keys of all Operational Targets:

```
# iscsiutil -pO
```

To display login keys of all Sessions:

```
# iscsiutil -pS
```

To display login keys of a particular Operational Target identified by its Target Name:

```
# iscsiutil -p -T <target-name>
```

Among the various login keys displayed by the commands described above, the parameters of interest are:

- `Authentication Method`
- `Header Digest`
- `Data Digest`

Displaying Negotiated Login Key Values

After executing the "ioscan" command, which initiates a login negotiation, the following command can be executed to display the negotiated login keys for all sessions (provided login negotiation is successful):

```
# iscsiutil -pVS
```

B Transport Statistics

Transport Statistics

Transport statistics are explained in Table B-1, “Transport Statistics,” on page 88.

The Class column (CL) provides message classification. Messages can be informational (I), target errors (T), transient driver errors (D), or connectivity problems (C).

Informational Messages are counters for driver events. They are not an indication of an error, but should an error occur, they may provide some profiling information.

Target Errors are detected at the initiator and should be reported to HP and/or the target vendor. Not all target errors are reported on the host side, and it is the responsibility of the system administrator to monitor any device specific logs for target issues.

Transient Driver Errors typically occur when some resource, for example, memory, is in short supply, or something is not configured correctly. The error is considered transient, because a retry of the operation, or a correct re-configuration, is typically successful. I/Os that experience transient errors are retried, so no data is lost. Control operations such as an application open, or a task management command, may not be retried (the determination to retry is left to the application or to the administrator). If the system resource load is increased, a small value for a transient driver error statistic may be an indication of problems. Larger values for the transient driver error statistic will start to impact performance.

Connectivity Problems are typically network or target availability problems. Connectivity problems are transient in the sense that a network infrastructure engineer can resolve the problem and I/O traffic will resume as before.

Table B-1 Transport Statistics

iscsiutil Statistic	CL	Description of Field
Transport Global Statistics		
Number of `ioscan`'s issued	I	The total number of times the iSCSI subtree of nodes has been scanned.
Number of Discovery sessions opened	I	The number of discovery sessions that were opened to discover new targets. A discovery session is opened for each iSCSI target port during an `ioscan`. A single `ioscan` would therefore result in a separate discovery session for each target port (portal group).

Table B-1 **Transport Statistics (Continued)**

iscsiutil Statistic	CL	Description of Field
Number of Discovery session open failures	D/C	The number of attempted discovery sessions that failed. The failure is due to an inability to open a connection. Some of the reasons why a connection open might fail are: — Initiator/host name not configured. — Hostname could not be resolved (confirm that iscsi_resolvd daemon is running). — Target is unreachable (network or target problems). — Resource allocation failures (memory, ISID, target).
Number of Normal sessions opened	I	The number of normal sessions that were opened. Normal sessions are opened to perform I/O. In general, normal sessions are any session other than discovery sessions.
Number of Normal session open failures	D/C	The number of attempted normal/operational sessions that failed. The failure is due to an inability to open a connection. Some of the reasons why a connection open might fail are: — Hostname could not be resolved (confirm that iscsi_resolvd daemon is running). — Target is unreachable (network or target problems). — Resource allocation failures (memory, ISID, target).
Number of SCSI-3 REPORT LUNs commands issued	I	The number of SCSI-3 REPORT_LUNS commands issued to iSCSI targets during ioscans.
Number of SCSI-3 REPORT LUNs commands issued that failed	D	The number of attempted SCSI-3 REPORT LUNS commands that failed. The failure is due to an incomplete I/O operation or no support for the command by the target. Lack of support for this command is not an error, although it is included in this counter.
Number of SCSI INQUIRY commands issued	I	The number of SCSI INQUIRY commands issued to iSCSI targets during ioscans.
Number of SCSI INQUIRY commands issued that failed	D/C/T	The number of attempted SCSI INQUIRY commands that failed. The failure is due to an incomplete I/O operation.

Table B-1 Transport Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of SendTargets commands issued	I	The number of iSCSI SendTargets commands sent to iSCSI targets. The iSCSI SendTargets command is used in discovery sessions to determine normal targets behind a target port (portal group). Normal sessions can then be established with the normal targets for I/O operations.
Number of SendTargets commands issued that failed	D	The number of iSCSI SendTargets command attempts that failed. The failure is due to an operation time-out which is seen as an incomplete I/O operation.
Number of SendTargets response parse failures	T	The number of iSCSI SendTargets command attempts that failed as a result of a received key parse error. This failure would result from invalid keys received from the target device.
Number of unclaimed LUNs	I	The number of LUNs that will not be seen in an `ioscan` operation. LUNS are reported to the initiator through the SCSI-3 REPORT LUNS command. All LUNs are viewed, and a primary addressing mode is determined. All LUNs behind a target with a mode that is not the primary addressing mode will not be seen in the `ioscan` output. Also, LUNs using a multi-level LUN address are not supported and are included in this count.
Number of I/Os that failed due to session being offline	D/C	The total number of I/O requests that failed as a result of a session being offline. This is a global total over all sessions.
Time when statistics were last cleared	I	The time that the statistics were last cleared. This defines the period of time to which the statistics can be applied, and therefore can be used for averaging the statistics. Because each system is different, separate statistic rates can be determined on a per-system basis and used to identify load changes.
Transport Session Statistics		
Number of session opens from upper layers	I	The number of session open requests received in the transport layer from upper layer protocols (SCSI).
Number of session closes from upper layers	I	The number of session close requests received in the transport layer from upper layer protocols (SCSI).

Table B-1 Transport Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of session reopens	I	The number of time the ioconfigd (I/O Configuration Daemon) initiates a session reopen.
Number of session opens in the reopen context	I	The number of session open requests received in the transport layer while in the reopen context. A reopen is a component of session error recovery.
Number of session opens aborted in the reopen context	I	The number of session open requests received, while in a reopen context, that were aborted because the session is already open.
Number of session closes aborted in the reopen context	I	The number of session close requests received, while in a reopen context, that were aborted because the session is already closed.
Number of connection opens that failed	C	The number of session open requests received that failed. The failure could have been due to unavailability of resources, or problems with the connection to the target.
Number of successful login redirections	I	The number of times logins were successfully redirected to different target addresses.
Number of async events received for dead connections	C	The number of TCP connections that have been unexpectedly dropped. This is typically a result of connectivity problems. If a problem is persistent, a network infrastructure engineer should be contacted.
Number of destination hostname resolution failures	C	The number of hostname resolution failures. Hostname resolution is performed via the iscsi_resolvd daemon. A hostname resolution failure indicates a network configuration problem.
Number of destination address routing failures	C	The number of destination address routing failures. Destination address routing is performed through host networking. An address routing failure indicates a network configuration problem.
Number of session logouts	I	The number of session logouts performed by the initiator.
Number of session state machine transitions to online state	I	The number of session state machine transitions to the online state.

Table B-1 Transport Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of session state machine transitions to transient state	I	The number of session state machine transitions to the transient state. A session enters the transient state when: — The current active connection dies. — Task Management cold reset is in progress. — Session logout has occurred. — Flow control has been enabled.
Number of session state machine transitions to offline state	I	The number of session state machine transitions to the offline state.
Number of requests to enable session flow control	I	The number of requests from the network interface driver to enable session flow control for I/Os. This might be the result of excessive I/Os through a session that impacts general network performance.
Number of requests to disable session flow control	I	The number of requests from the network interface driver to disable session flow control for I/Os.
Number of async events "target will drop all connections" received	I	The number of asynchronous requests from the target to perform a session logout.
Number of SendTargets command send failures	D/I/T	The number of PDU send attempts that failed. This could be the result of resource allocation problems or target connectivity issues.
Number of unexpected events received in session online state	D	The number of session state machine unexpected events received while in the online state. The unexpected events will be ignored, and will not impact operations. If this value is non-zero, an investigation into the cause of the unexpected events should be initiated.
Number of unexpected events received in session transient state	D	The number of session state machine unexpected events received while in the transient state. The unexpected events will be ignored, and will not impact operations. If this value is non-zero, an investigation into the cause of the unexpected events should be initiated.

Table B-1 Transport Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of unexpected events received in session offline state	D	The number of session state machine unexpected events received while in the offline state. The unexpected events will be ignored, and will not impact operations. If this value is non-zero, an investigation into the cause of the unexpected events should be initiated.
Number of Task Management commands sent	I	The number of Task Management commands sent by all sessions.
Number of Task Management commands failed	D/T	The number of Task Management commands that failed. The failures could be due to unavailable resources or target connectivity problems.
Number of login failures due to not configuring CHAP secret	D	The number of CHAP authentication configuration errors. CHAP authentication was requested and the initiator secret had not been configured, or had not been correctly configured, by the administrator.
Number of times there was no connection associated with an authentication response	D/C	The number of times a connection has been closed, or logged out, for which the iradd daemon has responded with an authentication response using CHAP.
Number of times the connection signature stamp did not match with the authentication response	I	The number of times a connection has been closed, or logged out, but the associated structures are not freed, for which the iradd daemon has responded with an authentication response using CHAP.
Number of times the target sent a "bad next stage" during login negotiation	T	The number of times the target sent a "bad next stage" during login negotiation. A non-zero value indicates problems were experienced during a login attempt to an iSCSI target. These would typically indicate protocol violations and would not be seen with conforming targets.
Number of login context allocation failures	D	The number of times the iSCSI login context allocation has failed. The failures could be due to unavailable resources.
Number of unsolicited NOP-INs sent by the target	I	The total number of unsolicited NOP-IN PDUs received from the target.

C Diagnostic Messages

Diagnostic Messages

The HP-UX iSCSI Software Initiator works with HP-UX Event Monitoring Services (EMS) and the Support Tools Manager (STM). By default, the iSCSI Software Initiator logs all diagnostic messages to the STM log files. STM can be used to view the diagnostic messages logged in the STM log files.

EMS can be configured to automatically notify the system administrator when diagnostic messages are logged by the iSCSI Software Initiator.

In addition to logging all diagnostic messages to the STM log files, the iSCSI Software Initiator logs some of the diagnostic messages to the `/var/adm/syslog/syslog.log` file.

The messages that can be logged to the `syslog.log` file are defined at 8 levels. By default, messages with numerical values less than or equal to LVL_ERR (Level 3) will be logged. See Table C-1, “Logging Levels for the syslog.log File,” below, for detailed descriptions of logging levels.

Table C-1 Logging Levels for the `syslog.log` File

Logging Level	Level Number	Description
LVL_PANIC	0	The system is unusable and a panic will follow. The message will provide additional data about the panic.
LVL_ALERT	1	The system has entered a state that requires an action by the administrator. The message describes the condition, and the action that must be taken
LVL_CRIT	2	A critical event has occurred. The system is still usable. The message will provide additional details about the event.
LVL_ERR	3	An error condition has occurred. The system is still usable. The message will provide additional details about the error.
LVL_WARN	4	An event has occurred that may indicate some future action is necessary. The system is still usable, and the current system operation has not been affected. The message will provide additional details about the warning.
LVL_NOTE	5	An event has occurred that fits the normal profile of operation, however, the event is called out so that it may be viewed in the context of other things occurring in the system. The system is still usable, and the current system operation has not been affected. The message will provide additional details about the noted event.
LVL_INFO	6	An informational message regarding some unusual, but acceptable, event that occurred during system operation. The system is usable, and the information can be ignored. The message will provide additional details.
LVL_DEBUG	7	A debug message used internally by the HP lab to assist in debugging an exception event. Customers should not attempt to interpret messages at this level.

To change the level of event logging, execute the following command:

```
iscsiutil [/dev/iscsi] -b <dbg_level>
```

where `dbg_level` is one of the level numbers listed in Table C-1, “Logging Levels for the syslog.log File,” on page 96.

Logging of events is always turned on. By default, messages at the LVL_ERR and lower levels are displayed in the syslog.log file. Level zero reporting cannot be turned off.

The messages detailed in the following “Message Descriptions” section, may be generated when the iSCSI Software Initiator is in use.

The first value in the first line of each message listing is defined in an iSCSI header file and can be used to reference a specific message. The second value is the level of the event.

The second entry in each message listing is the message string presented in C language format with print format characters present.

The third entry in each message listing is a brief description of the message.

It is necessary to have a familiarity with the iSCSI specification and the driver implementation to completely understand the messages. See Table C-1, “Logging Levels for the syslog.log File,” on page 96 for a detailed explanation of logging levels.

Message Descriptions

1. `ISCSI_LOG_SCSI_INQ_FAILED                              LVL_ERR`

```
"SCSI Inquiry Command from the initiator to the target failed \n"
"with cdb_status 0x%x, io_status 0x%x and return value of 0x%x."
```

```
There is something wrong with the target nexus. An interpretation of the status
is necessary to determine what has happened.
```

2. `ISCSI_LOG_REPORTLUNS_FAILED                            LVL_DEBUG`

```
"SCSI Report LUNs Command from the initiator to the target \n"
"failed with cdb_status 0x%x, io_status 0x%x and return value of 0x%x."
```

```
Either the target does not support the REPORT_LUNS SCSI command,or there is
something wrong with the target nexus. An interpretation of the status is
necessary to determine what has happened.
```

3. `ISCSI_LOG_INV_LUN_ADDR_TYPE                            LVL_ERR`

```
"Unsupported LUN Addressing type recognized."
```

```
The target has attempted to use invalid LUN addressing.
```

4. `ISCSI_LOG_NO_LUNS_CONFIGURED                           LVL_WARN`

```
"No LUNs configured on the target."
```

```
There are no LUNs configured on the target. Please configure some LUNs and
re-issue the "ioscan" command.
```

5. ISCSI_LOG_PROTOCOL_MISSING_TGT_KEY_PART   LVL_ERR

```
"The target did not send the remaining portion of the <key=value> \n"
"pair. The partial <key=value> pair sent in the previous login response \n"
"was (%s)."
```

A login key returned by the target is incomplete and, as such, cannot be processed. The specific target login key is included. This information should be provided to HP for analysis.

6. ISCSI_LOG_PROTOCOL_KEY_ACROSS_TOO_LONG   LVL_ERR

```
"The target device sent a <key=value> pair spanning across login \n"
"responses of size %d bytes. The maximum initiator supported <key=value> \n"
"pair size is %d bytes. The partial <key=value> pair in the login response \n"
"payload is (%s)."
```

A target login key was returned and the key is defined across responses and is too long. The size of the target key and the specific target key name are included. This information should be provided to the support team for analysis.

7. ISCSI_LOG_PROTOCOL_ALLOC_ACROSS_FAIL   LVL_ERR

```
"Failure to allocate memory for a login response buffer."
```

A login key does not completely fit into the remaining space of a response buffer. It was necessary to attempt allocation of an additional response buffer to contain the rest of the login key. The allocation of the additional buffer failed.

8. ISCSI_LOG_PROTOCOL_NO_EQUAL_AFTER_NAME   LVL_ERR

```
"An equal sign was missing after a target login key \n"
"(%s)."
```

The target login key that is specified, was received, and a terminating null character was found, before an equal sign was identified. This information should be provided to the support team for analysis.

9. ISCSI_LOG_PROTOCOL_NO_0_AFTER_VAL   LVL_ERR

```
"The null character was missing after the <key=value> pair \n"
"(%s) sent by the target."
```

The target login key that is specified was received, and a second equal character was found, before a null character was identified. This information should be provided to the support team for analysis.

10. ISCSI_LOG_PROTOCOL_INIT_KEY_NOTUNDERSTD   LVL_ERR

```
"Initiator login key (%s) was not understood by \n"
"the target."
```

The initiator login key supplied to the target was not recognized or understood by the iSCSI target.This information should be provided to the support team for analysis.

11. ISCSI_LOG_PROTOCOL_INIT_KEY_REJECTED LVL_ERR

"Initiator login key (%s) was rejected by the target."

The initiator login key supplied to the iSCSI target was recognized by the iSCSItarget and
then rejected. This information should be provided to the supportteam for analysis.

12. ISCSI_LOG_PROTOCOL_NOTUNDER_ALLOC_FAIL LVL_ERR

"Failure to allocate memory to hold a login key (%s) \n"
"not understood by the target."

An attempt to allocate a buffer failed. The buffer was to be used internally by the iSCSI Software Initiator to hold an initiator login key that had not yet been recognized by the target.

13. ISCSI_LOG_PROTOCOL_TGT_KEY_NOTUNDERSTD LVL_INFO

"Target login key (%s) was not understood by \n"
"the initiator."

The target sent the specified login key (which was not recognized by the initiator). This information should be provided to the target vendor for resolution. The initiator will respond to the target in the next login text PDU with a "not understood" indicator.

14. ISCSI_LOG_PROTOCOL_UNEXPT_TGT_INIT_NEGO LVL_ERR

"Target initiated an unexpected negotiation of a \n"
"login key (%s)."

This is a case of the target violating the iSCSI protocol by wrongly proposing a key which it should have only responded to in response to that particular key being proposed from an initiator. This case would occur if a particular key which is to be proposed by an initiator only, has not been proposed yet, but the target goes ahead and proposes that particular key.

15. ISCSI_LOG_PROTOCOL_BAD_IKEY_VAL LVL_ERR

"Target returned a value (%d) which is out of (%d - %d) \n"
"range or is an unexpected value. The value proposed \n"
"by the initiator is %d. The login key is (%s)."

The target returned a numerical key value which is either out of range or is an unexpected value. The login failed as a result of the protocol violation. In the case of an unexpected value, the protocol violation could be because the value does not comply with the result function for that key. If the result function is Minimum, the value responded should be less than or equal to the value proposed by the initiator. If the result function is Maximum, the value responded should be greater than or equal to the value proposed by the initiator.

16. ISCSI_LOG_PROTOCOL_BAD_CKEY_VAL LVL_ERR

```
"Target returned an invalid list response for a login key. \n"
"The <key=value> pair is (%s). The value proposed by the \n"
"initiator is (%s)."
```

The target returned a string key value that is invalid. The login failed as a result of protocol violation. This could be because the value does not comply with the result function for that key. If the result function is Boolean "AND", the value responded by the target should be the boolean "AND" of the initiator proposed value and the value selected by the target.

17. ISCSI_LOG_PROTOCOL_IKEY_NO_RSP LVL_ERR

```
"No response received from the target for integer \n"
"login key (%s)."
```

The target failed to respond to or recognize, a numerical key. The login failed as a result of the protocol violation.

18. ISCSI_LOG_PROTOCOL_CKEY_NO_RSP LVL_ERR

```
"No response received from the target for character \n"
"login key (%s)."
```

The target failed to respond to or recognize, a string key. The login failed as a result of the protocol violation.

19. ISCSI_LOG_PROTOCOL_TOO_MANY_LOGIN_EXCH LVL_ERR

```
"The maximum number of exchanges (%d) for a login stage has been \n"
"exceeded."
```

The initiator and target have performed exchanges in an attempt to successfully complete the login phase. The number of exchanges has exceeded the HP-UX implementation maximum indicated in the message. To prevent the possibility of an infinite loop, the login attempt has been terminated.

20. ISCSI_LOG_PROTOCOL_BAD_TGT_NSG LVL_ERR

```
"The next login stage of the target (%d) is invalid with \n"
"respect to the current initiator stage (%d) as well as the next \n"
"initiator stage (%d)."
```

The target returned a value for the next login stage that is invalid. The login failed as a result of the protocol violation. This information should be presented to the target vendor for resolution.

21. ISCSI_LOG_PROTOCOL_ALLOC_KEYS_CB LVL_ERR

```
"Memory allocation failure of a data structure (size %d bytes) \n"
"required for a login attempt to proceed successfully."
```

An attempt to allocate a connection key structure has failed. The login failed as a result. This could be a transient condition. The login can be retried by the application. A recommendation is to monitor the statistics using the iscsiutil tool for allocation failures. Allocation failures are typically a symptom of insufficient memory.

22. ISCSI_LOG_PROTOCOL_BAD_CHAP_A_VAL LVL_WARN

"Invalid value (%s) for the key CHAP_A."

An invalid value was returned by the target for the CHAP negotiation key "CHAP_A".

23. ISCSI_LOG_PROTOCOL_BAD_CHAP_I_VAL LVL_WARN

"Invalid value (%s) for the key CHAP_I."

An invalid value was returned by the target for the CHAP negotiation key "CHAP_I".

24. ISCSI_LOG_PROTOCOL_BAD_CHAP_C_VAL LVL_WARN

"Invalid value (%s) for the key CHAP_C."

An invalid value was returned by the target for the CHAP negotiation key "CHAP_C".

25. ISCSI_LOG_PROTOCOL_BAD_CHAP_N_VAL LVL_WARN

"Invalid value (%s) for the key CHAP_N."

An invalid value was returned by the target for the CHAP negotiation key "CHAP_N".

26. ISCSI_LOG_PROTOCOL_BAD_CHAP_R_VAL LVL_WARN

"Invalid value (%s) for the key CHAP_R."

An invalid value was returned by the target for the CHAP negotiation key "CHAP_R".

27. ISCSI_LOG_PROTOCOL_BAD_CHAP_AIC LVL_WARN

"Target sent an invalid key (%s) when the initiator was \n"
"waiting for CHAP_A, CHAP_I or CHAP_C key."

An invalid value was returned by the target instead of a CHAP_A, CHAP_I or CHAP_C key.

28. ISCSI_LOG_PROTOCOL_BAD_CHAP_NR LVL_WARN

"Target sent an invalid key (%s) when the initiator was \n"
"waiting for a CHAP_N or CHAP_R key."

An invalid value was returned by the target instead of a CHAP_N or CHAP_R key.

29. ISCSI_LOG_SSN_RESOLV_FAIL LVL_ERR

"Unable to resolve the Target Hostname (%s)."

The specified Hostname is not a Qualified Domain Name. This will cause the target port
hostname resolution to fail. Check if the hostname specified for the target port is a Qualified
Domain Name. If the problem persists, please contact your network administrator for further assistance.

30. ISCSI_LOG_SSN_ROUTE_FAIL LVL_ERR

```
"No route found for IP address (%s)."
```

The initiator is not able to reach the specified target address. The specified IP address for the target is changed resulting in no route to the target. Check if the connection between initiator and target is physically secure and established. Check if the target has the same IP address as displayed in the log message. If the problem persists, please contact your network administrator for further assistance.

31. ISCSI_LOG_INCOR_KEY LVL_ERR

```
"Invalid text key (%s) received from the target in \n"
"SendTargets response."
```

An invalid text key was received from the target in response to a SendTargets request. The login failed as a result of this protocol violation. This information should be presented to the target vendor for resolution.

32. ISCSI_LOG_INCOR_TGT_ADDR LVL_ERR

```
"Invalid key value (%s) received from the target for \n"
"TargetAddress key."
```

The specified invalid key value was received from the target for theTargetAddress. The login failed as a result of this protocol violation. The information should be presented to the target vendor for resolution.

33. ISCSI_LOG_MAX_SSNS_OVERFLOW LVL_CRIT

```
"Attempt to allocate more than the maximum number of sessions."
```

An attempt was made to allocate more than the maximum number of sessions. This is an HP-UX limitation that cannot be exceeded. The number of targets attached to the initiator may need to be reduced, or if there are unused ext_bus entries, it may be necessary to reassign (compress) ext_bus assignments. This should only be done with the assistance of HP support.

34. ISCSI_LOG_LOGIN_NO_CHAP_SECRET LVL_WARN

```
"The initiator CHAP secret is not configured."
```

CHAP authentication failed as the CHAP initiator secret is not configured. Use the iscsiutil command to configure the CHAP initiator secret.

35. ISCSI_LOG_BAD_ISCSI_NAME LVL_ERR

Initiator received an iSCSI name from a target that was in an invalid format. The target cannot be recognized by the iSCSI initiator as a result.

36. ISCSI_LOG_INCOR_KEY_REDI LVL_ERR

```
"Invalid text key (%s) received from the target in"
"a login redirection response."
```

A key that wasn’t a TargetAddress key was received while processing a login redirection request.

37. `ISCSI_LOG_TEMP_REDIRECT_REQUESTED` LVL_INFO

```
"Login response with status code (0x%x) to"
"TargetAddress %s indicated a"
"temporary redirection request to %s."
```

During login,an iSCSI target device requested the initiator to perform a temporary login redirection to a new TargetAddress.

38. `ISCSI_LOG_PERM_REDIRECT_REQUESTED` LVL_INFO

```
"Login response with status code (0x%x) to"
"TargetAddress %s indicated a"
"permanent redirection request to %s."
```

During login,an iSCSI target device requested the initiator to perform a permanent login redirection to a new TargetAddress.

39. `ISCSI_LOG_REDIRECT_OCCURRED` LVL_INFO

```
"Login to TargetAddress %s was redirected"
"successfully to %s."
```

The target requested the login to be redirected to a new target address. The request was granted and the redirected login succeeded.

40. `ISCSI_LOG_INCOR_TGT_ADDR_REDI` LVL_ERR

```
"Invalid key value (%s) received from the target for\n"
"TargetAddress key."
```

An invalid key value was sent by the target within a login redirection response for TargetAddress text key. The discovery of target devices would fail as a result.

D iSCSI Software Interface Driver Statistics

iSCSI Software Interface Driver Statistics

Statistics are maintained in the iSCSI Software Interface Driver (SWD). These statistics are explained in Table D-1, “Software Interface Driver Statistics,” on page 107.

The Class column (CL) provides message classification. Messages can be informational (I), target errors (T), transient driver errors (D), or connectivity problems (C).

Informational Messages are counters for driver events. They are not an indication of an error, but should an error occur, they may provide some profiling information.

Target Errors are detected at the initiator and should be reported to HP and/or the target vendor. Not all target errors are reported on the host side. It is the responsibility of the system administrator to monitor any device specific logs for target issues.

Transient Driver Errors will typically occur when some resource, for example, memory, is in short supply, or something is not configured correctly. The error is considered transient, because a retry of the operation, or a correct re-configuration, would typically be successful. I/Os that experience transient errors will be retried, so no data will be lost. Control operations such as an application open, or a task management command, may not be retried (the determination to retry is left to the application or to the administrator). If the system resource load is increased, a small value for a transient driver error statistic may be an indication of problems . Larger values for the transient driver error statistic will start to impact performance.

Connectivity Problems will typically be network or target availability problems. Connectivity problems are transient in the sense that a network infrastructure engineer can resolve the problem and I/O traffic will resume as before.

Table D-1 Software Interface Driver Statistics

iscsiutil Statistic	CL	Description of Field
Software Interface Driver Global Interface Statistics		
Number of connection opens	I	The number of TCP connection opens initiated. The statistic is incremented when a call to open a connection is made.
Number of connection closes	I	The number of TCP connection closes initiated. The statistic is incremented when a call to close a connection is made.
Software Interface Driver Connection Statistics		
Number of times the login failed	I/T	The number of login failures due to incorrect text / key values and formatting. If this is a transient problem at the target end, the initiator would recover on a successive retry attempt.
Number of exception status class values returned by target	D	The number of iSCSI login phase failure. The reasons for login failure can be determined by looking at the STM/syslog.log logs to find more detailed information. The class of login failures enumerated here consists of interoperability (potential protocol violations) issues between the host and the target.
Number of PDU headers with Protocol errors received by initiator	T	The number of login failures due to protocol violations by the target. The protocol violation occurs when a target sends a PDU login response header and the initiator determines that the response is not protocol compliant.
Number of times iswd daemon failed to open a connection	D/C	The number of times the `iswd` daemon failed to open a connection to the requested target. The failure can be the result of resource allocation failures, incorrect target configuration, or network infrastructure problems.The exact reason for the failed will be logged in `syslog.log`.
Number of failures to send a login command due to kernel memory allocation failure	D	The number of attempts to send the Login command that failed due to memory allocation failures. The upper level driver recovery may retry the session open, resulting in a re-attempt to send the Login command. If the memory allocation request succeeds, the Login command will transmit successfully.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of asynchronous failures waiting for a login response	D	The number of asynchronous failures experienced while waiting for a login response. The asynchronous failure might be due to a PDU exchange timeout/abort or lack of memory resources. The upper level driver recovery may retry the session open, resulting in a re-attempt to send the Login command. If the memory allocation request succeeds, the Login command will transmit successfully.
Number of asynchronous failures waiting for a logout response	D/C	The number of asynchronous failures experienced while waiting for a logout response. The asynchronous failure might be due to a PDU exchange timeout/abort or lack of memory resources. One additional attempt to complete the Logout is made by requesting the iswd daemon to close the TCP connection and tear down the stream.
Number of unexpected TCP closes in the active state	C	The number of unexpected TCP close events received during the connection ready state. As part of recovery, all the I/Os on this connection are aborted, the connection is closed, and a session reopen is triggered.
Number of timeouts on FIN after sending a logout command	C	The number of timeouts on FIN after the target has sent a logout response. A close is triggered by requesting the iswd daemon to close the TCP connection and tear down the stream.
Number of TCP connection open timeouts	C	The number of TCP connection open timeout occurs. The timeout will trigger the freeing of resources. The upper level driver recovery may retry the session open, resulting in a re-attempt to send the Login command. If the memory allocation request succeeds, the Login command will transmit successfully.
Number of unexpected connection closes after a login command	C	The number of unexpected TCP close events received while waiting for a login response. As part of the recovery mechanism, resources are freed, the connection is closed, and a session reopen is triggered.
Number of unexpected connection closes after a logout command	C	The number of unexpected TCP close events received while waiting for a logout response. As part of the recovery mechanism, the Logout PDU is aborted, resources are freed, and the TCP connection is closed.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of target authentication timeouts	I/C	The number target authentication timeouts that occurred during communication with the userspace iradd daemon.
Number of target authentication failures	I/C	The number of target authentication failures for CHAP. Either the target's CHAP information is not configured in the RADIUS server, or the CHAP information provided by the target is incorrect.
Number of temporary redirection requests	I	Number of temporary login redirections requested by a target device.
Number of permanent redirection requests	I	Number of permanent login redirections requested by a target device.
Number of kernel memory allocation failures	D	The number of memory allocation attempts for a kernel PDU structure that failed. A failure to allocate a PDU structure means that some outbound command, or a NOP-OUT in response to a NOP-IN, could not be completed. As a result: — The regular occurrence of this event will have a negative impact on performance. — Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI), however, there is no retry policy for native iSCSI commands.
Number of streams message allocation failures	D	The number of streams message memory allocation attempts that failed. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of PDU transmission failures due to an offline connection	D	The number of transmission attempts of native iSCSI commands that failed, because the connection on which the I/O was attempted, was offline. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).

iSCSI Software Interface Driver Statistics

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of streams message duplication failures	D	The number of streams message duplication operations that failed. As a result of the failure, the iSCSI Software Initiator will not transmit the related iSCSI commands. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of PDU exchange timeouts	D/C/T	The number of iSCSI PDUs successfully transmitted to the target for which no response was received within a specified period of time. If a timeout occurs, the iSCSI Software Initiator will initiate a logout for the session as part of the recovery, and eventually will attempt to login again with the target. The problem could be a network infrastructure problem or a target congestion issue. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of PDU exchanges aborted	C/T	The number of iSCSI PDUs that were aborted by the initiator while waiting for a response from a target. The abort occurs if a response to the iSCSI PDU has not been received at the initiator within a preset timeout period. The failure can also be due to an unexpected close of the TCP connection. The problem could be a network infrastructure problem or a target congestion issue.
Number of PDU exchanges abandoned	D/T	The number of exchanges between an iSCSI initiator and target that were abandoned. This will typically occur when the number of exchanges to complete a negotiation goes beyond a predetermined limit, usually indicating an infinite loop.
Number of I/Os issued on this connection	I	The number of SCSI I/Os that were sent to the networking stack by the iSCSI Software Initiator.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of I/O timeouts	D/T	The number of SCSI I/Os that did not complete within a time period preset by an upper level protocol (SCSI). The driver will recover from SCSI I/O timeouts using session level error recovery, for example, tearing down the session and starting over. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of kernel memory allocation failures	D	The number of kernel memory allocation failures for SCSI I/O related data structures. Failure to allocate will result in the I/O not being processed and returned to the SCSI layer. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of streams message allocation failures	D	The number of memory allocation attempts for a kernel driver structure that failed. A failure to allocate the driver SCSI structure means that some SCSI I/O could not be completed. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of I/Os delayed while waiting for resources	D	The number of memory allocation attempts for a kernel driver structure that failed. The iSCSI Software Initiator will retry the allocation request a set number of times. The ultimate failure to allocate the driver structure means that some SCSI I/O could not be completed. The regular occurrence of this event will have a negative impact on performance. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of I/Os failed due to an offline connection	D	The number of SCSI I/Os that were aborted as a result of retrying memory resource allocation when the connection went offline. As a result of this event, the corresponding SCSI I/O will be aborted. This could be a network infrastructure problem. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of I/Os failed due to memory resource constraints	D	The number of SCSI I/Os that failed to acquire memory resources within the maximum number of retries. As a result of the memory allocation failures, the SCSI I/O was failed backed to SCSI. Failed I/Os will be retried according to existing SCSI retry policies (same as Fibre Channel and parallel SCSI).
Number of invalid Data-In PDUs received	T	The total number of invalid Data-In PDUs sent by the target. A non-zero value for this stat indicates that the target is not adhering to the iSCSI protocol by not sending Data-In PDUs with: Buffer Offset in increasing offset order with non-overlapping ranges. DataSN in increasing order.
Number of I/O underruns	I	The number of I/Os on a connection where the number of bytes that were received by the initiator did not match the number of bytes that the target sent.
Number of I/O underflows	I	The number of I/Os on a connection where the target sent less data than what was requested by the initiator. A large value for this statistic is normal.
Number of I/O overflows	I	The total number of I/Os on a connection where the target sent more data than what was requested by the initiator. This indicates that the Expected Data Transfer Length in the I/O request was not sufficient.
Number of I/O failures due to response code errors	T	The total number of I/Os that failed due to a response code error in the SCSI response PDU. This means that the target failed to execute the I/Os. A large value here could indicate that the target is not functioning properly.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of Data-In PDUs received without data	I	The total number of empty Data-In PDUs received by the initiator.
Number of invalid R2T PDUs received	T	The total number of Request to Transfer (R2T) PDUs from the target that had an incorrect buffer offset or Data Transfer Length in them. A non-zero value for this stat indicates that the target is not adhering to the error recovery policy negotiated by the initiator.
Number of I/Os that failed to respond to an R2T due to kernel memory allocation failures	D	The total number of Request to Transfer (R2T) PDUs that could not be honored by the initiator due to resource constraints at the initiator. A non-zero value for this stat indicates that the initiator is intermittently running out of resources while handling iSCSI I/O traffic.
Number of unexpected R2T PDUs received during a Read I/O	T	The number of Request to Transfer (R2T) requests sent by the target for a Read operation. This is an unexpected behavior.
Number of unexpected Data-In PDUs received during a Write I/O	T	The total number of data-in PDUs received by the initiator while the initiator was doing a write operation to the target. This is an unexpected behavior.
Number of Data-In PDUs with incorrect residual count	T	The total number of I/O requests where the target had an incorrect residual count value and where the status for the I/O was sent as part of the last Data-In PDU. This might happen when the target indicated an underflow condition but the residual count value did not match the expected residual count.
Number of SCSI Response PDUs with incorrect residual count	T	The total number of I/O requests where the target had an incorrect residual count value set. This might happen when the target indicated an underflow condition but the residual count value did not match the expected residual count. The response PDU in this case was sent as a separate PDU by the target.
Number of I/O underruns	I	The number of I/Os on a connection where the number of bytes that were received by the initiator did not match the number of bytes that the target sent.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of I/O underflows	I	The number of I/Os on a connection where the target sent less data than what was requested by the initiator. A large value for this statistic is normal.
Number of I/O overflows	I	The total number of I/Os on a connection where the target sent more data than what was requested by the initiator. This indicates that the Expected Data Transfer Length in the I/O request was not sufficient.
Number of I/O failures due to response code errors	T	The total number of I/Os that failed due to a response code error in the SCSI response PDU. This means that the target failed to execute the I/Os. A large value here could indicate that the target is not functioning properly.
Number of Data-In PDUs received without data	I	The total number of empty Data-In PDUs received by the initiator.
Number of invalid R2T PDUs received	T	The total number of Request to Transfer (R2T) PDUs from the target that had an incorrect buffer offset or Data Transfer Length in them. A non-zero value for this stat indicates that the target is not adhering to the error recovery policy negotiated by the initiator.
Number of I/Os that failed to respond to an R2T due to kernel memory allocation failures	D	The total number of Request to Transfer (R2T) PDUs that could not be honored by the initiator due to resource constraints at the initiator. A non-zero value for this stat indicates that the initiator is intermittently running out of resources while handling iSCSI I/O traffic.
Number of I/O failures due to streams message concatenation memory allocation failures	D	The total number of I/O failures in the inbound path resulting from a failure of `msgpullup` call.
Number of holes seen in the status sequencing	D/T	The total number of PDUs that were received where the status sequence number of the PDU does not match the expected status sequence number. For error recovery level zero, this will cause the initiator to do a Session level logout.
Number of SCSI Async events received	I	The number of asynchronous events received by the initiator.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of "target requests logout" Async events received	I	The total number of times the target sent an asynchronous event with the AsyncEvent set as "target requests logout".
Number of "target will drop connection" Async events received	I	The total number of times the target sent an asynchronous event with the AsyncEvent set as "target will drop connection".
Number of "target requests parameter negotiation" Async events received	I	The total number of times the target sent an asynchronous event with the AsyncEvent set as "target request parameter negotiation".
Number of "vendor specific" Async events received	I	The total number of times the target sent an asynchronous event with the AsyncEvent set as "vendor specific Async event".
Number of Reject PDUs due to Data Digest errors	C	The total number of Reject PDUs sent by the target that had the Reason set as "Data (payload) Digest Error".
Number of Reject PDUs due to SNACK rejects	I	The total number of Reject PDUs sent by the target that had the reason set as "SNACK Reject".
Number of Reject PDUs due to Protocol Errors	D	The total number of Reject PDUs sent by the target that had the Reason set as "Protocol Error ".
Number of Reject PDUs due to excessive Immediate Commands	T	The total number of Reject PDUs sent by the target that had the Reason set as "Immediate Command Reject". This typically happens if the target has too many outstanding immediate commands.
Number of Reject PDUs due to "Task in progress"	T	The total number of Reject PDUs sent by the target that had the Reason set as "Task in Progress".
Number of Reject PDUs due to Invalid Data ACK	D	The total number of Reject PDUs sent by the target that had the Reason set as "Invalid Data ACK".
Number of Reject PDUs due to Invalid PDU field	D	The total number of Reject PDUs sent by the target that had the Reason set as "Invalid PDU Field".
Number of Reject PDUs due to Lack of Target Resources	T	The total number of Reject PDUs sent by the target that had the Reason set as "Long Operations Reject". A large value indicates that the target is frequently running out of resources.

Table D-1 Software Interface Driver Statistics (Continued)

iscsiutil Statistic	CL	Description of Field
Number of Reject PDUs due to Negotiation Resets	I	The total number of Reject PDUs sent by the target that had the Reason set as "Negotiation Reset".
Number of Reject PDUs due to target Waiting for Logout	I	The total number of Reject PDUs sent by the target that had the Reason set as "Waiting for Logout".
Number of Reject PDUs due to Miscellaneous reasons	T	The total number of Reject PDUs sent by the target for which the Reason code does not match any of the currently defined codes.
Time when statistics were last cleared	I	The time that the statistics were last cleared. This provides an indication of the period of time to which the statistics can be applied, and therefore can be used for averaging the statistics. Because each system is different, separate statistic rates can be determined on a per-system basis and used to determine load changes.

E iSCSI Software Interface Driver Diagnostic Messages

iSCSI Software Interface Driver Diagnostic Messages

The HP-UX iSCSI Software Interface Driver works with HP-UX Event Monitoring Services (EMS) and the Support Tools Manager (STM). By default, the iSCSI Software Interface Driver logs all diagnostic messages to the STM log files. STM can be used to view the diagnostic messages logged in the STM log files.

EMS can be configured to automatically notify the system administrator when diagnostic messages are logged by the iSCSI Software Interface Driver.

In addition to logging all diagnostic messages to the STM log files, the iSCSI Software Interface Driver logs some of the diagnostic messages to the /var/adm/syslog/syslog.log file.

The messages listed below may be generated when the iSCSI Software Interface Driver is in use.

The first value in the first line of each message listing is defined in an iSCSI header file and can be used to reference a specific message. The second value is the level of the event.

The second entry in each message listing is the message string presented in C language format with print format characters present.

The third entry in each message listing is a brief description of the message.

It is necessary to have a familiarity with the iSCSI specification and the driver implementation to completely understand the messages. See Table C-1, “Logging Levels for the syslog.log File,” on page 96, for a detailed explanation of logging levels.

1. ISW_LOG_CLOSED_ON_LOGIN   LVL_WARN

```
"Unexpected connection close while awaiting a login response."

An unexpected close of the TCP connection occurred while a login was being
attempted. The login failed as a result.
```

2. ISW_LOG_BAD_STATUS_CLASS   LVL_WARN

```
"Login failed with response of status class (0x%x) \n"
"and status code (0x%x) to TargetAddress %s."

A bad status class was returned in the login response buffer. The login failed as
a result.
```

3. ISW_LOG_BAD_PROTO_HDR   LVL_ERR

```
"Login response with a protocol error received."

A protocol header violation from the target has occurred.
```

4. ISW_LOG_BAD_HDR_DIGEST   LVL_ERR

```
"Incorrect Header Digest. Expected/Received = (0x%x)/(0x%x)."

The initiator received an incorrect checksum (CRC) value for the header segment
of the iSCSI PDU sent from the target.
```

5. ISW_LOG_BAD_DATA_DIGEST   LVL_ERR

```
"Incorrect Data Digest. Expected/Received = (0x%x)/(0x%x)."

The initiator received an incorrect checksum (CRC) value for the data segment
of the iSCSI PDU sent from the target.
```

F Glossary

This chapter contains definitions of terminology and acronyms used throughout this document.

Terminology

Directory Agent - A process which collects service advertisements. There can only be one DA present per given host.

Discovery Session - Initiated with a Discovery Target to discover Operational Targets in a Network Entity.

Gigabit Ethernet - An ethernet infrastructure with the capability of transferring data at the rate of 1 gigabit per second in either the transmit or receive direction (abbreviated GigE).

iSCSI Adaptation Layer - A module which interfaces with the SCSI-2 services layer.

Interface Driver - A driver that interfaces to a host bus adapter or to another software subsystem that controls a host bus adapter.

iSCSI Infrastructure - Components in a network supporting the iSCSI protocol that do not perform as iSCSI targets or initiators; same as network infrastructure.

iSCSI Initiator - A network port with the capability of performing as a SCSI initiator as defined by SAM-2.

iSCSI Name - is permanent, globally unique, and location or address independent. Both targets and initiators require names for the purpose of identification. In addition, names enable iSCSI storage resources to be managed regardless of location (address). The iSCSI name is the principal object used in authentication of targets to initiators and initiators to targets.

iSCSI Node - represents a single iSCSI initiator or iSCSI target. There are one or more iSCSI Nodes within a Network Entity. The iSCSI Node is accessible via one or more Network Portals. An iSCSI Node is identified by its iSCSI Name.

iSCSI Target - A network port with the capability of performing as a SCSI target as defined by SAM-2.

iSCSI Transport Layer - A session management module, which also interfaces with the iSCSI Adaptation Layer and the interface driver.

Login Redirection - When an iSCSI login to a target is attempted, the target responds by redirecting the login attempt to a different target address. Redirected addresses are provided as part of the login response to the initial login attempt.

Network Entity - represents a device or gateway that is accessible from the IP network. A Network Entity must have one or more Network Portals, each of which can be used by some iSCSI Nodes contained in that Network Entity to gain access to the IP network.

Network Portal - a component of a Network Entity that has a TCP/IP network address and can be used by an iSCSI Node within that Network Entity for it's iSCSI session connection(s). In an initiator, the Network Portal is identified by its IP address. In a target, the Network Portal is identified by its IP address and its listening TCP port.

Normal Session - An unrestricted session used to transfer SCSI commands, data, and responses, between an iSCSI initiator and an iSCSI Operational Target.

Persistent Storage - An area of storage where stored data will persist across power outages.

Portal Group - A Portal Group defines a set of Network Portals within an iSCSI Node that collectively support the capability of coordinating a session with connections that span these portals.

Portal Group Tag - This 16-bit quantity identifies a Portal Group within an iSCSI Node. All Network Portals in a given iSCSI Node that have the same portal group tag, are in the same Portal Group.

Service Agent - A process working on the behalf of one or more services to advertise the services.

Session - The iSCSI equivalent of a SCSI I-T nexus. A session must be established between an iSCSI initiator and an iSCSI target prior to any communication.

Supported - A feature in a release for which implementation and testing for that release have been completed.

Target Address - consists of three components, the IP address of the network portal the target uses, its TCP port number, and its target portal group tag.

Transport Driver - A protocol specific layer combining aspects of the iSCSI Adaption Layer and the iSCSI Transport Layer, which defines a transport technique for SCSI block IO.

Unsupported - A feature in a release for which implementation and testing of that feature is incomplete, or a feature in a release that has been declared “unsupported” by Hewlett-Packard.

User Agent - A process working on the user’s behalf to establish contact with some service. The UA retrieves service information from the Service Agents or Directory Agents.

Acronyms

CHAP - Challenge-Handshake Authentication Protocol - A technique to authenticate initiators and targets.

DA - Directory Agent - an SLP component.

Fibre Channel - an encapsulation protocol used primarily for SCSI on HP-UX.

GigE - Gigabit Ethernet - an ethernet network that functions at Gigabit speeds.

HA - High Availability - a configuration component designed to maximize hardware and software availability through various component fail-over techniques.

HBA - Host Bus Adapter - a computer hardware component that provides outside connectivity from a computer system using a specific protocol across an internal bus.

HP-UX - HP-Unix - HP version of the UNIX Operating System.

IETF - Internet Engineering Task Force - a standards preparation group.

IP - Internet Protocol - a network transmission protocol.

IPSec - Internet Protocol Security - an internet protocol security system.

iSCSI - SCSI over IP - an IETF encapsulation protocol for use of SCSI block IO over an IP network.

ISID - Initiator Session Identifier - The iSCSI session identifier defined on the host. See the iSCSI specification for further details.

KRS - Kernel Registry Services - An HP-UX kernel specific mechanism that facilitates the maintenance of structured data. Subsystems wihtin the kernel can use KRS to maintain data. This data can either be volatile, or persistent, across system reboots.

LU - Logical Unit - as defined in SAM-2: an end device such as a stand-alone disk or tape, or a piece of logical storage in a disk array.

LUN - Logical Unit Number - as defined in SAM-2: the path to a LU, or the 64-bit address of the LU.

LVM - Logical Volume Manager - HP-UX software subsystem for volume management.

NAS - Network Access Server - Operates as a client of a RADIUS server. This is the host which runs the “iradd” daemon.

OLA/R/D - OLA (Online Addition), OLR (Online Replacement), OLD (Online Deletion), and OLAR (Online Addition and Replacement) - the HP-UX ability to add, delete, or replace a PCI/PCI-X HBA while the system remains up and running.

PCI - Peripheral Component Interconnect - a system IO bus widely used in computer systems; the IO bus standard for new HP servers.

PCI-X -Peripheral Component Interconnect Extended - the follow-on higher speed version of PCI.

pSCSI - parallel SCSI - transport layer support for SCSI parallel busses.

PVLinks - Physical Volume Links - an HP-UX component of LVM that permits the fail-over from a primary path to a LU to an alternate path to the LU.

RADIUS - Remote Authentication Dial In User Service - A RADIUS server is used in CHAP authentication of initiators and targets.

SA - Service Agent

SAM - System Administration Manager - an HP-UX GUI tool for system level administration.

SAM-2 - SCSI Architecture Model (2) - a T10 standard.

SAN - Storage Area Network - a network with emphasis on storage with a defined protocol for the infrastructure.

SCSI - Small Computer Systems Interface - a mass storage data transmission protocol.

SD - Software Distributor - an HP-UX tool for installation and distribution of software.

SG - Serviceguard - an HA product for HP-UX.

SLP - Service Location Protocol - a standard for local service delivery in an intranet.

STM - Support Tool Manager - an HP-UX diagnostic tool subsystem.

SWD - Software Interface Driver - an interface driver for the iSCSI Software Initiator.

TPGT - Target Portal Group Tag - a number between 0 and 65535 that identifies a target portal group.

UA - User Agent - an SLP component.

Index

Index