

# StorageTek Automated Cartridge System Library Software

## HA Linux Installation and Configuration Guide



Release 8.5.1

F28535-06

April 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

StorageTek Automated Cartridge System Library Software HA Linux Installation and Configuration Guide,  
Release 8.5.1

F28535-06

Copyright © 2019, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Documentation Accessibility	v
-----------------------------	---

## 1 Getting Started

---

System Requirements	1-1
Client Options	1-2
Server Options	1-2
NFS Options	1-2
Network Requirements	1-2
Software Requirements	1-2
High Level Installation Procedure	1-3

## 2 Configuring Linux for ACSLS HA

---

Configuring /etc/hosts	2-1
Multipath Bonded Network Configuration	2-2
Port Mapping	2-2
Configuring ACSLS HA Ethernet Interfaces	2-4
Network Interface Bonding	2-4

## 3 Configuring the File System with NFS

---

Configuring the ACSLSHA NFS Server	3-2
Mounting the NFS File System From the ACSLSHA Server	3-3

## 4 Downloading Software Packages

---

Downloading ACSLS 8.5.1 for Linux	4-1
Downloading ACSLS 8.5.1 HA for Linux	4-1
Accessing ACSLS Documentation	4-2

## 5 Installing ACSLS

---

Installing ACSLS on the First Node	5-1
Installing ACSLS on the Adjacent Node	5-2

## 6 Installing and Configuring ACSLS HA

---

Installing ACSLS HA	6-1
Running setup.py	6-2
Starting, Stopping and Statusing the acslsha Service	6-4
ACSLS HA Logging	6-5
Determining Which Node is Node 1	6-7
Performing a Forced Fail Over	6-7
Performing a Graceful Shutdown	6-7
Performing Maintenance on One or Both Nodes	6-7
Patching or Upgrading ACSLS 8.5.1	6-8
Recovering From a Corrupt ACSLS Database	6-8
Running setup.py While Any Node Is Currently Running ACSLSHA	6-8

## Index

---

# Preface

The guide contains guidelines and procedures for installing and configuring Oracle's StorageTek Automated Cartridge System Library Software High Availability (ACSLS HA) 8.5.1 Cluster software on LINUX-based systems.

This document is intended for experienced LINUX System Administrators having a good understanding of the LINUX operating system.

This document offers moderate background information for most of the technologies that are used and it provides guidance for the standard anticipated installation procedures. However this document alone does not replace an implied requirement for LINUX system familiarity and expertise.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

# 1

## Getting Started

ACSLS HA is a hardware and software configuration that provides dual-redundancy, automatic recovery and automatic failover recovery to ensure uninterrupted tape library control service if component or subsystem failures occur. This document explains the configuration, setup and testing procedures required to provide High Availability to ACSLS software.

### Note:

ACSLS HA Linux supports ACSLS 8.5.1 or later.

It is best to review the complete installation process before beginning the procedure. The process of installing a clustered application involves multiple steps requiring strict attention to detail. This procedure is normally undertaken by specialists in UNIX/LINUX system integration.

The configuration is a two-node system. It includes two complete subsystems (one active and one standby) with monitoring software capable of detecting serious system failures. It can switch control from the primary to the standby system for any non-recoverable subsystem failure. The configuration provides redundant power supplies, and redundant network and I/O interconnections that can recover subsystem communication failures instantly without the need for a general switch over.

The system leverages the monitor and failover features built in to ACSLS HA. The multipath features in the Linux operating system provide resilient library control operation with minimal downtime. Linux offers IP bonding to ensure uninterrupted network connectivity and multipaths to NFS disk I/O with RAID 1 to ensure uninterrupted access to system data. ACSLS HA monitors the health of system resources including the internal hardware and external I/O resources. It can also manage a system switch over if required.

The ACSLS HA agent monitors the ACSLS application, its database, its file system, and connectivity to StorageTek library resources, invoking the ACSLSHA failover service, if needed. In this redundant configuration, the ACSLS Library Control Server has a single logical host identity, which is always known within the framework and to the rest of the world. This identity is transferred automatically as needed between the nodes with minimal downtime during the transition.

Before embarking on the project, review the complete process of installing and configuring ACSLS HA as it is documented within this publication. If desired, Advanced Customer Services from Oracle can be arranged to advise, to assist, or to handle the entire installation.

## System Requirements

An ACSLS HA server configuration consists of two identical Linux server nodes sharing an external NFS file system. During configuration, ACSLS HA utilizes the same operating

system device name on each server. Therefore, any network interface cards installed in the servers must be identical.

## Client Options

ACSLA HA supports all ACSLS clients that use the Automated Cartridge System Application Programming Interface (ACSAPI) network interface. A single network IP address is shared between the two server nodes, allowing ACSAPI clients to address ACSLS using a common virtual host ID.

## Server Options

ACSLA HA 8.5.1 can run on any system that meets the minimum hardware requirements for Linux 7.3, 7.6, or 7.8.

Systems generally include four Ethernet ports on the motherboard. Additional network interface cards are required in order to accommodate a total of eight Ethernet ports per server. Best practice dictates that the two servers and the external ethernet cards are identical. The naming conventions used in this document for network ports, bond names and NFS assumes these best practices.

## NFS Options

Any network file system server is valid provided that its NFS4 options are set according to the installation instructions in this guide. NFS version 4 must be utilized.

The *ACSLA Installation Guide* instructs you to create users and groups on the ACSLSA server. You must also create these users and groups on the NFS server.

## Network Requirements

You must reserve a total of seven IP addresses:

- One Logical Host IP address
- Two NFS IP addresses
- Two Library IP addresses
- Two Node Interconnect IP addresses

Ideally, library interfaces reside on different subnets and attach to separate Host Library Interface cards. Best practice dictates that the ethernet cards installed in both systems be identical.

## Software Requirements

ACSLA HA 8.5.1 requires the following software components:

- Oracle Linux 7.3, 7.6, or 7.8
- ACSLS 8.5.1
- ACSLS HA 8.5.1

## High Level Installation Procedure

ACSLS HA installation involves the following steps:

1. Install two Linux platform servers. Refer to the document, *Installing Oracle Linux 7.6 Systems*, available from the Oracle Technology Network library. The root file system on the internal disks must be protected with some form of RAID or other redundancy.
2. Configure the basic Linux system including the cabling for seven network interface ports on each of two nodes.
3. Define and configure the bonded IP addresses for the “node to node” interconnect and NFS.
4. Configure the NFS file system on the NFS server and mount it from each node.
5. Download software packages ACSLS 8.5.1 and ACSLS HA 8.5.1.
6. Install and configure ACSLS 8.5.1 and patch update (if any) on both nodes.
7. Install and configure ACSLS HA 8.5.1 on both nodes.
8. Start ACSLS HA on both servers and test the Logical IP and client connections to ACSLS.

# 2

## Configuring Linux for ACSLS HA

This chapter describes how to prepare the Linux system to support ACSLS HA.

Topics include:

- [Configuring /etc/hosts](#)
- [Multipath Bonded Network Configuration](#)
- [Configuring ACSLS HA Ethernet Interfaces](#)

### Configuring /etc/hosts

Your `/etc/hosts` file on each node must contain entries for the local host, the two Linux node names, their IP addresses and the logical host. You can create this file on both nodes even if these IP addresses are not yet configured.

Define the local host 127.0.0.1 as shown in the example below.

The “public network interface” is that which you defined when you first installed the operating system. In the following example, it is mapped to physical device `NET0` (logical device `eno1`).

The `/etc/hosts` file must contain entries for the ACSLS HA interconnects as shown below. They must be named `localnode` and `remotenode`. You will configure the actual interfaces later in the configuration process. For now, add them to the `/etc/hosts` file.

```
# localhost
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6
localhost6.localdomain6

# Public Network
10.80.25.113 hostname1.Domain-name.com hostname1
# ACSLS-HA Logical Host
10.80.25.65 hostname1.Domain-name.com hostname1
# ACSLS-HA Interconnects
192.168.84.1 localnode
192.168.84.2 remotenode

# localhost
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomainlocalhost6
localhost6.localdomain6

# Public Network
10.80.25.131 hostname2.Domain-name.com hostname2
```

```
# ACSLS-HA Logical Host
10.80.25.65      hostname2.Domain-name.com      hostname2
# ACSLS-HA Interconnects
192.168.84.1    remotenode
192.168.84.2    localnode
```

## Multipath Bonded Network Configuration

Redundancy is the overall scheme for high-availability computing. Redundancy applies not only to the servers, but to each communication interface on each server. For the public interface, use Internet Protocol Bonding on Linux. Internet Protocol Bonding provides instant NIC recovery for failing network communications without the need for a general system failover. For the library interface, this means using a dual TCP/IP connection with two network interfaces across two independent routes. If any element in one route should fail, ACSLS continues to communicate over the alternate interface. Note that if both paths to the library interfaces fail at the same time, ACSLS HA will not fail over to the other node. This behavior protects the integrity of other ACSs that may also be controlled by the instance of ACSLS under the control of ACSLS HA.

ACSLS HA requires redundant network connections for the following:

- Public and client communications
- Library communications
- Private intra-node cluster communications
- NFS communications

## Port Mapping

[Figure 2-1](#) shows eight Ethernet ports on each server, accessible from two separate Network Interface Controllers (four ports on each). A total of eight ports on each node are used.

Connect cables for eight network interface ports:

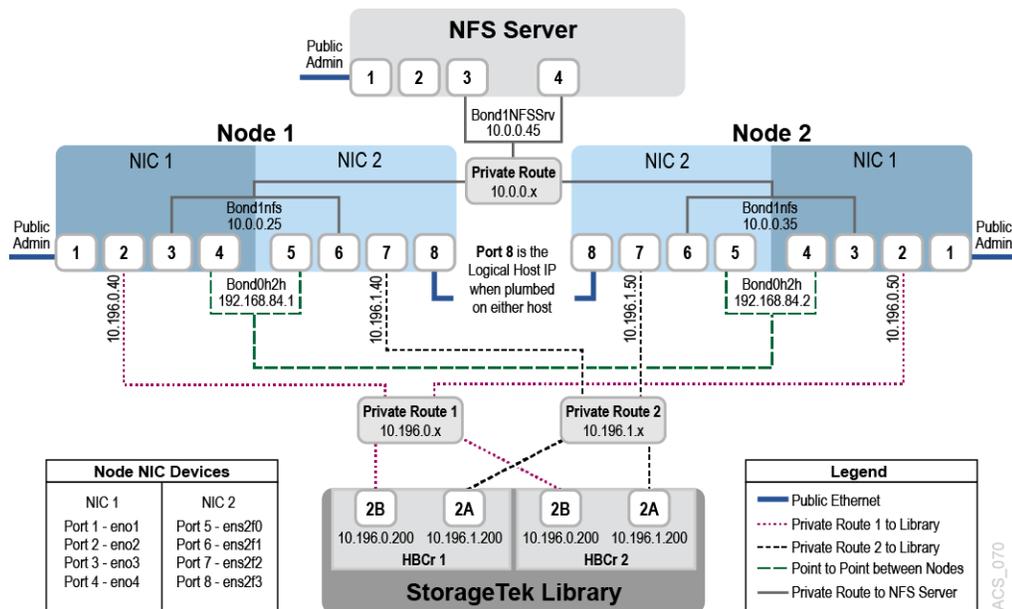
- One cable for ACSLS-HA Logical Host connection for client access.
- Two cables for Library communications
- Two cables for direct Intra-node communications
- Two cables for NFS communications
- One cable for Public administration access.

Use [Figure 2-1](#) as a guide.

 **Note:**

- Each port of each pair for the direct intra-node and NFS connections must reside on a different network interface card (NIC). This ensures that each NIC is not a single point of failure when the connections are bonded.
- The naming conventions used for the ports (eno1, ens2fo) in this guide may differ from your environment if your servers are of a different brand or model than used by Oracle and/or if you have installed external ethernet cards that use a different naming convention. If they differ then you will have adjust your configuration commands accordingly. It is strongly recommended that your ethernet cards and motherboard ports are named (device name) exactly the same.
- The IP addresses, bondings, and other networking configuration parameters in the examples provided in this publication coincide with [Figure 2-1](#) and are used as if you are configuring the environment in this diagram. You must adjust accordingly for the private routes, IP addresses and other networking parameters in your environment.

**Figure 2-1 Port Mapping Example**



This figure shows two Network interface cards with four ports each.

- Each network device (port) has an associated configuration file named `ifcfg-interface` in the `/etc/sysconfig/network-scripts` directory, where X-interface is the name of the interface. The names of your ports may differ from this if you are using a different server brand. You must have identical ethernet cards installed in each system.
- Configuration files `ifcfg-eno1` through `ifcfg-eno4` belong to the first NIC card on the motherboard and `ifcfg-ens2f0` through `ifcfg-ens2f3` belong to the installed NIC card.

When configuring network connections and bonding with command line tools (`nmcli`), use the name of the interface, the portion of the configuration file name that follows `ifcfg-`. For example, the interface name for `ifcfg-eno1` is `eno1`.

- Port 1 on each host in the diagram is the interface that you defined when you first installed the operating system.

**Note:**

Refer to the *Oracle Linux 7 Administration Guide* for more information about network configuration.

## Configuring ACSLS HA Ethernet Interfaces

Perform the procedures in this section to configure ACSLS HA Ethernet interfaces.

### Network Interface Bonding

Linux Bonding provides a mechanism for building redundant network interfaces to guard against failures with NICs, cables, switches or other networking hardware. When configuring Bonding on your Linux host, combine two or more physical network interfaces into a single Bond. The following examples illustrate the creation of Node to Node and NFS Bonding.

**Note:**

Oracle recommends that you use the “balance-rr” mode for bonding which is the default mode in Oracle Linux. It provides both load balancing and fault tolerance.

If the NetworkManager service is running, you can use the `nmcli` command to display the state of the system's physical network interfaces. This will help you recognize the device names for each interface. You will use these device names to create the IP addresses and then add them to the networking manager.

Run the `nmcli device status` command to view the Ethernet ports:

```
# nmcli device status

DEVICE TYPE      STATE      CONNECTION
eno1  ethernet  disconnected  --
eno2  ethernet  disconnected  --
eno3  ethernet  disconnected  --
eno4  ethernet  disconnected  --
ens2f0 ethernet  disconnected  --
ens2f1 ethernet  disconnected  --
ens2f2 ethernet  disconnected  --
ens2f3 ethernet  disconnected  --
```

From the output above, note the four ports on each Network Interface Controller (NIC):

- NIC1: (eno1,eno2,eno3,eno4)
- NIC2: (ens2f0,ens2f1,ens2f2,ens2f3)

In the steps below, you build and assign the following network connections on each Node using the nmcli interface. Assign a name for each connection to identify its function.

- (eno1) – Public Administration connection
- (eno2) – ACSLS Library connection 1
- (eno4,ens2f0) – Bonded connection for Node to Node (N2N) communication
- (eno3,ens2f1) – Bonded connection for NFS communication
- (ens2f2) – ACSLS Library connection 2
- (ens2f3) – ACSLSHA Logical Host connection for ACSLS

The following steps illustrate the process used to build the library connections and bonds.

# 3

## Configuring the File System with NFS

This chapter demonstrates how to set up your NFS server using Linux 7.3, 7.6, or 7.8.

Topics include:

- [Configuring the ACSLSHA NFS Server](#)
- [Mounting the NFS File System From the ACSLSHA Server](#)

### Note:

You may choose to use another operating system for NFS (such as Solaris or Windows) provided that you understand how to implement the commands, permissions, groups and user access required for ACSLS 8.5.1 users and groups as outlined in the *ACSL 8.5.1 Installation Guide*. NFS version 4 (NFS4) must be utilized for the NFS file system.

In this section we will create the NFS4 file system on the NFS server. Special attention must be given to the options as this will enable ACSLS to share its database (`/export/home`) across the two nodes.

The `/export/home` directory will be created under each Linux server's root file system and then be mounted to the `<folder-to-share-to-nodes>` directory on the NFS server by the ACSLSHA application.

### Note:

The `<folder-to-share-to-nodes>` directory can be any directory on the NFS Server. However, it is highly recommended that the directory be empty and named something meaningful so as to easily identify what the share is being used for. In the examples below, the share folder that will be created will be named `/node1-node2-acslsha-share`.

ACSLSHA utilizes a small directory on the NFS server's file system. It is named `/export/acslsha`. You need not create or concern yourself with this directory. It is defined here only to point out that it should not be touched by any user. It is created by ACSLSHA and it updates two files, `heartbeat1` and `heartbeat2` as ACSLSHA monitors the system. ACSLSHA utilizes these files in order to ensure that Node1 knows that Node2 is healthy and visa versa. It is also used during ACSLSHA startup time in order to prevent both nodes from becoming the "primary" in the event of a startup tie between the two nodes.

## Configuring the ACSLSHA NFS Server

Perform the following steps to configure the NFS server:

1. Build a network bond between two NIC ports for load balancing and redundancy (see [Figure 2-1](#)).

```
# nmcli device status

DEVICE     TYPE      STATE      CONNECTION
eno1       ethernet  connected  eno1
eno2       ethernet  disconnected --
eno3       ethernet  disconnected --
eno4       ethernet  disconnected --

# nmcli connection add type bond con-name bond1NFSSrv ifname
bond1NFSSrv mode blance-rr ip4 10.0.0.45/24 autoconnect yes

# nmcli connection add type bond-slave con-name bond1NFSSrv-con1
ifname eno3 master bond1NFSSrv

# nmcli connection add type bond-slave con-name bond1NFSSrv-con2
ifname eno4 master bond1NFSSrv

# nmcli connection up bond1NFSSrv
# systemctl restart network
# nmcli device status

DEVICE     TYPE      STATE      CONNECTION
bond1NFSSrv bond      connected  bond1NFSSrv
eno1       ethernet  connected  eno1
eno2       ethernet  disconnected --
eno3       ethernet  connected  bond1NFSSrv-con1
eno4       ethernet  connected  bond1NFSSrv-con2
```

2. Create ACSLS user IDs and groups according to the instructions provided in the *ACSLs Installation Guide*.

ACSLs requires specific users and groups to be created as part of its installation. These users and groups are also required on the NFS Server. Refer to the ACSLS Installation Guide for instructions on how to create the required users and groups on the NFS Server. Ownership/permissions must be applied to the NFS share directory. In this example, the ownership/permission requirements must be applied to the `/node1-node2-acslsha-share` directory on the NFS server).

3. Install the `nfs-utils` package:

```
# yum install nfs-utils
```

4. Edit the `/etc/exports` file to define the directories that the server will make available for clients to mount. This directory will contain the ACSLS installation for both nodes. Each entry consists of the local path to the exported directory, followed by a list of clients (nodes 1 and 2) that can mount the directory with client-specific mount options in parentheses.

For example:

```
/node1-node2-acslsha-share 10.0.0.25(rw,sync,no_root_squash,no_all_squash)
/node1-node2-acslsha-share 10.0.0.35(rw,sync,no_root_squash,no_all_squash)
```

 **Note:**

There is no space between a client specifier and the parenthesized list of options.

For more information, refer the exports(5) manual page.

5. Start the nfs-server service, and configure the service to start following a system reboot:

```
# systemctl start nfs-server
# systemctl enable nfs-server
```

 **Note:**

It has been determined that using DNS in order to reference the IPs on both the NFS server and the local nodes can cause errors if DNS is not accessible or is excessively slow. Please use explicate IPs.

## Mounting the NFS File System From the ACSLSHA Server

Use the `mount` command to test your NFS connections.

The NFS file system must remain mounted from both nodes throughout the entire ACSLS and ACSLS installation and configuration processes.

```
# mkdir -p /export/home

# mount -t nfs -o rw,suid,soft 10.0.0.45: /node1-node2-acslsha-share /export/
home
```

where 10.0.0.45 is the IP address of the NFS server.

 **Note:**

Using DNS in order to reference the IPs on both the NFS server and the local nodes can cause errors if DNS is not accessible or is excessively slow. To avoid these issues, use explicate IPs.

# 4

## Downloading Software Packages

This chapter describes how to install ACSLS and ACSLS HA software.

Topics include:

- [Downloading ACSLS 8.5.1 for Linux](#)
- [Downloading ACSLS 8.5.1 HA for Linux](#)
- [Accessing ACSLS Documentation](#)

You must download the software packages to each server node. Place the packages in the `/opt` directory.

### Downloading ACSLS 8.5.1 for Linux

Perform the following steps to download ACSLS to both nodes:

1. Start a web browser on the system and navigate to the Oracle Software Delivery Cloud website at the following URL:  
<https://edelivery.oracle.com>
2. Click **Sign In** and enter the user name and password provided by your Oracle support representative.
3. Read and accept the export restrictions.
4. In the search field, enter `acsls` and select `StorageTek Automated Cartridge System Library Software (ACSL)`.
5. Locate the ACSLS 8.5.1 release and click **Add to Cart**.
6. Click `View Cart`. Verify the selected software, and then click `Checkout`.
7. Under `Selected software`, click the `Select Platform/Languages` menu and select the Linux platform. Click `Continue`.
8. Read and accept the copyright licenses and click `Continue`.
9. Verify ACSLS for your platform and click `Continue`.
10. Select the `VXXXX-xx` package and save the zip file to the location of your choice.
11. Repeat this procedure to download the software to the second node.

### Downloading ACSLS 8.5.1 HA for Linux

Perform the following steps to download ACSLS HA to both nodes:

1. Start a web browser on the system and navigate to the Oracle Software Delivery Cloud website at the following URL:  
<https://edelivery.oracle.com>

2. Click `Sign In` and enter the user name and password provided by your Oracle support representative.
3. Read and accept the export restrictions.
4. In the search field, enter `acsls` and select `StorageTek Automated Cartridge System Library Software (ACSL) High-Availability Agent (HA)`.
5. Click `Add to Cart`.
6. Click `View Cart`. Verify the selected software, and then click `Checkout`.
7. Under `Selected software`, click the `Select Platform/Languages` menu and select the `Linux` platform. Click `Continue`.
8. Read and accept the copyright licenses and click `Continue`.
9. Verify ACSLS HA for your platform and click `Continue`.
10. Select the zip file and click `Download`.
11. Select the `VXXXX-xx` package and save the zip file to the location of your choice.
12. Repeat this procedure to download the software to the second node.

## Accessing ACSLS Documentation

To access ACSLS documentation:

1. Start a web browser on the system and navigate to the Oracle Help Center website at the following URL:  
<https://docs.oracle.com>
2. Select `Hardware`.
3. Select `Storage Documentation`.
4. Select `StorageTek ACSLS Manager documentation`.
5. Select `Automated Cartridge System Library Software 8.5`.

# 5

## Installing ACSLS

This chapter describes ACSLS installation in an ACSLS HA Linux configuration.

Topics include:

- [Installing ACSLS on the First Node](#)
- [Installing ACSLS on the Adjacent Node](#)

### Installing ACSLS on the First Node

Ensure that `/export/home` is mounted from both nodes to the NFS file system server while performing the installation and configuration of both ACSLS and ACSLS HA. When you are finished installing and configuring both applications on both nodes, then you must ensure that `/export/home` has been unmounted from both nodes. ACSLS HA will mount and dismount `/export/home` as needed during startup and shutdown.

A portion of ACSLS is installed in the `/opt/oracle` directory and another portion of ACSLS is installed in the `/export/home` mounted NFS directory. When you install and configure the second node, it will overwrite some of the files in `/export/home` that were written by node 1. This is acceptable. However, you must change ownership of the `/export/home/ACSSS/log/acsdb_install.log` file while installing ACSLS on node 2 after running `pkg_install`, but before running `install.sh`.

#### Note:

ACSLs installation will prompt you for the location of your backups. It is critical that you use the `/export/home/backup` directory that is mounted to the NFS file system.

Refer to the *ACSLs Installation Guide* to completely install ACSLS 8.5.1 and then configure and audit the library on the first node. During installation, you will be instructed to create ACSLS users and groups on each node. You must also create these users and groups on the NFS server.

To ensure that ACSLSHA has access to ACSLS start and status functions, you must change the group of the `/etc/init.d/acsls` file to `acsls`. For example:

```
chgrp acsls /etc/init.d/acsls
```

#### Note:

This action must be performed on both nodes.

After installing, configuring and testing ACSLS 8.5.1 on the first node, shut down ACSLS on the first node:

```
#su - acsss  
$acsss shutdown
```

 **Note:**

Never attempt to start ACSLS on a node if it is already running on the other node. This will corrupt the ACSLS database.

## Installing ACSLS on the Adjacent Node

Log in to the second node and begin to install ACSLS according to the instructions provided in the *ACSLs Installation Guide*.

### Important Note - Additional Step for Node 2 Installation:

When installing ACSLS on Node 2, you must first run `pkg_install.sh` which installs the ACSLS rpm. However, before running the `/export/home/ACSSS/install/install.sh` script (which completes the installation), you must change ownership on the file `/export/home/ACSSS/log/acbdb_install.log` to `acbdb`. To do this, open a new terminal on Node 2 and run the following command with super user permissions:

```
# chown acbdb /export/home/ACSSS/log/acbdb_install.log
```

To ensure that ACSLSHA has access to ACSLS start and status functions, you must change the group of the `/etc/init.d/acsls` file to `acsls`. For example:

```
chgrp acsls /etc/init.d/acsls
```

 **Note:**

This action must be performed on both nodes.

After installing, configuring, and testing ACSLS on both nodes, shut down ACSLS. Leave the `/export/home` directory mounted to both nodes at this time as it is required for ACSLS HA installation.

# 6

## Installing and Configuring ACSLS HA

This chapter describes how to install and configure ACSLS HA for Linux.

Topics include:

- [Installing ACSLS HA](#)
- [Running setup.py](#)
- [Starting, Stopping and Staturing the acslsha Service](#)
- [ACSLs HA Logging](#)
- [Determining Which Node is Node 1](#)
- [Performing a Forced Fail Over](#)
- [Performing a Graceful Shutdown](#)
- [Performing Maintenance on One or Both Nodes](#)
- [Patching or Upgrading ACSLS 8.5.1](#)
- [Recovering From a Corrupt ACSLS Database](#)
- [Running setup.py While Any Node Is Currently Running ACSLSHA](#)

### Installing ACSLS HA

Ensure that `/export/home` is mounted to the NFS file system from both nodes while performing the installation and configuration of ACSLS HA. When you are completely finished installing and configuring ACSLS HA on both nodes, you will be instructed to unmount `/export/home` from both nodes. ACSLS HA will mount and dismount `/export/home` as needed during startup and shutdown.

The ACSLS HA Linux rpm is named `ACSLs-HA-8.5.1-X.XXX.x86_64.rpm` where X indicates version levels. Earlier, you were instructed to download this file to the `/opt` directory. The following examples will use this directory.

Perform the following steps:

1. From Node 1, CD to the `/opt` directory:

```
# cd /opt
```

2. Install the rpm:

```
# rpm -ivh ACSLS-HA-8.5.1-0.00X.x86_64.rpm
```

3. To ensure that ACSLS HA is registered with the Linux system services, use the following command to reload the system daemon:

```
#systemctl daemon-reload
```

4. From Node 2, repeat the above steps to install the ACSLS HA rpm on the other node.
5. Prepare to run the ACSLS HA `setup.py` command on Node 1.

## Running setup.py

The following example illustrates all of the `setup.py` options. You must run `setup.py` on both nodes, one at a time starting with Node 1. Note that when you run `setup.py` on Node1, it writes the same response data to Node 2 with the exception of setting up the SSH keys. When you run `setup.py` on Node2 you may choose to only run option 1 (Configure SSH keys between the nodes) followed by option 2 to verify that the configuration was correctly written to Node 2 when Node 1 was configured. (Optional) Enter contextual information here, including the purpose of the task.

If you select Action 2 (Display current configuration) while running `setup.py` for the first time, the configuration entries will be displayed as `None`.

(Optional) Enter task prerequisites here.

1. On Node 1, run `setup.py`:

```
[root@axid ~]# /opt/oracle/acslsha/setup.py
```

```
Validating local node.
```

```
Validating remote node.
```

```
Reading config file.
```

```
It is highly recommended that you execute each menu item in order starting with 1. If you choose not to set up the ssh keys between the nodes, you will need to enter the password for the remote node when prompted.
```

```
Building the menu
```

```
1) Configure SSH keys between the nodes  
2) Display current configuration  
3) Configure Logical Host for connecting to ACSLS  
4) Configure FileSystem  
q) Quit
```

2. Select Action 1.

```
Select action: 1
```

```
Please enter root password for remote node when prompted.
```

```
root@remotenode's password:
```

Respond with the root password of the remote node.

### 3. Select Action 2.

```
Select action: 2
LogicalHostDevice : None
StorageFilesystemType : None
LogicalHostIp : None
NodeId : None
StorageFilesystem : None
StorageMountPoint : None
StorageOptions : None
```

### 4. Select Action 3.

ACSL S HA must know the logical host address and device to access ACSLS. ACSLS HA will move this IP address between the nodes as necessary. The first step is to enter the IP address for the logical host used to access ACSLS. The format of the address is a dot delimited quad and a slash, followed by the subnet mask. For example, 10.80.25.81/23.

```
Enter the IP Address: 10.80.25.81/23 (enter your IP address)
Enter the device: eno1 (enter your device)
```

Successfully configured the logical host.

### 5. Select Action 4.

ACSL S HA must know the location of the file system containing the ACSLS installation. ACSLS HA will move this file system between the nodes as necessary.

```
The file system is currently set to None
Would you like to change this filesystem (y/Y/n/N/yes/no): Y
Enter filesystem: 10.0.0.123:/export/home (Enter the IP and name of your
NFS file system)
```

```
The Mount Point is currently set to None
Would you like to change this mount point (y/Y/n/N/yes/no): Y
Enter mount point: /export/home (Enter your local mount point directory)
```

```
The filesystem type is currently set to None
Would you like to change this type (y/Y/n/N/yes/no): Y
```

```
Enter filesystem type: nfs
The file system options are currently set to None
Would you like to change the options (y/Y/n/N/yes/no): Y
```

```
Enter options: rw,suid,soft (Note that no spaces are allowed in this
response)
```

Successfully configured the file system.

### 6. To display the current configuration, select Action 2:

- 1) Configure SSH keys between the nodes
- 2) Display current configuration
- 3) Configure Logical Host for connecting to ACSLS
- 4) Configure FileSystem

q) Quit

Select action: 2

```
LogicalHostDevice : eno1
StorageFilesystemType : nfs
LogicalHostIp : 10.80.25.81/23
NodeId : 1
StorageFilesystem : 10.80.25.124:/acslsha_Straub-Tooheys
StorageMountPoint : /export/home
StorageOptions : rw,suid,soft
```

**7. Repeat this entire procedure to run setup.py on Node 2.**

When `setup.py` runs, it will mount and unmount the NFS file system. At this time, ensure that it is unmounted from both nodes using the mount command first. If you see that the NFS file system is mounted then unmount it using the umount command. For example:

```
#mount
```

If the NFS file system is shown, then enter the following command:

```
#umount /export/home/
```

## Starting, Stopping and Statusing the acslsha Service

ACSL HA uses a Linux 7.3, 7.6, or 7.8 service for control. This service is called `acslsha`.

### Starting the acslsha Service

To start the service, issue the following command on both nodes:

```
#systemctl start acslsha
```

The node that you start first becomes the primary node. Start ACSLS and render the Logical Host IP to which ACSLS clients will attach.

The node that you start second becomes the secondary node. This node monitors the primary node and remains in standby until a failover occurs.

### Stopping the acslsha Service

To stop `acslsha`, first ensure that there is no activity or outstanding operations in ACSLS. Then enter, the following command:

```
#systemctl stop acslsha
```

 **Note:**

- If you stop the primary node, the product will fail over to the secondary. If you wish to shut down `acslsha` gracefully, stop the secondary node first.
- It may take several minutes before `acslsha` completely stops, as it must first shut down ACSLS.

**Statusing the acslsha Service**

To check status, enter the following command:

```
#systemctl status acslsha
```

Typical `acslsha` status from a node that is running:

```
# systemctl status acslsha
?oe acslsha.service - The Oracle ACSLSHA Service
Loaded: loaded (/usr/lib/systemd/system/acslsha.service; disabled; vendor
preset:
disabled)
Active: active (running) since Wed 2020-01-29 14:17:34 MST; 2 days ago
Main PID: 7244 (bash)
CGroup: /system.slice/acslsha.service
       7244 /bin/bash -c TERM=xterm /opt/oracle/acslsha/bin/AcslsHa.py >&1 |
/opt/oracle/acslsha/bin...
       7246 /usr/bin/python -u /opt/oracle/acslsha/bin/AcslsHa.py
       7247 /usr/bin/python -u /opt/oracle/acslsha/bin/logger.py -l 100000 -
g 10
-f /var/log/acslsha...
       63487 /usr/bin/python -u /opt/oracle/acslsha/bin/AcslsHa.py
       63488 /usr/bin/python -u /opt/oracle/acslsha/bin/AcslsHa.py
       63490 /usr/bin/python -u /opt/oracle/acslsha/bin/AcslsHa.py
       63492 /usr/bin/python -u /opt/oracle/acslsha/bin/AcslsHa.py
```

Typical `acslsha` status from a node that is not running:

```
acslsha.service - The Oracle ACSLSHA Service
Loaded: loaded (/usr/lib/systemd/system/acslsha.service; disabled; vendor
preset:
disabled)
Active: inactive (dead)
```

## ACSLs HA Logging

The state of each node can be determined by following the current `AcslsHa.log` on each node. You must be aware of the time stamps as a node running as primary may have previously been a secondary.

A node in Primary state (running ACSLS) will repeatedly log the following:

```
2020/01/07 07:31:48.611329 INFO - Monitoring with primary = True:
2020/01/07 07:31:48.611375 DEBUG - System state changed to :
MONITORING PRIMARY
2020/01/07 07:31:48.611454 DEBUG - AcslsHa: Updating node status with
primary = True and status = MONITORING PRIMARY
```

A node in Secondary state will repeatedly log the following:

```
2020/01/06 13:36:37.383299 INFO - Monitoring with primary = False:
2020/01/06 13:36:37.383333 DEBUG - System state changed to :
MONITORING SECONDARY
2020/01/06 13:36:37.383364 DEBUG - AcslsHa: Updating node status with
primary = False and status = MONITORING SECONDARY
```

A running primary node that has not yet been a secondary node will contain the following logs:

Directory `/var/log/acslsha` - Contains the current logs.

Logs are restarted and archived when they reach 100,000 lines or when `acslsha` is restarted on that node (whichever happens first). Archived logs reside in directories located under `/var/log/` named `acslsha.0`, `acslsha.1`, and up to `acslsha.9`, where `acslsha.0` is the most recent archive. The `acslsha` directory (with no “dot” number) is always the current running set of logs.

Log files in the `/var/log/acslsha` directory or any `acslsha.#` directory include the following:

- `AcslsHa.log`:  
The main currently running ACSLS HA log.
- `acslshaResourceAcsls.log`:  
Contains Information about ACSLS HA's starting and stopping of ACSLS.
- `acslsResource.log`:  
Contains information about the current status of ACSLS.
- `acslshaResourceLogicalHost.log`:  
Contains information about the Logical Host IP. Initially, this log will indicate that the Logical host IP has been started.
- `acslshaResourceRemoteNode.log`:  
Contains information that the primary node logs about the remote node (the secondary). When viewed on the secondary, this log contains information that the secondary logs about it's remote (the primary).
- `acslshaResourceStorage.log`:  
Contains the startup and the name of the storage resource (NFS file system mount). NFS errors or a loss of network connection to the NFS file server are logged here.
- `storageResource.log`:  
This log remains empty until a storage resource issue occurs.
- `setup.log`:

Contains the responses to the questions asked when `setup.py` was run. Note that you may need to review any archived set of logs in order to locate the most recently updated version of this file as setup is typically only run once and will move with the archives during a restart of ACSLS HA.

A node running as the secondary that has never been a primary will only contain the following logs:

- `acslshaResourceRemoteNode.log`
- `AcslsHa.log`
- `acslshaResourceRemoteNode.log`

## Determining Which Node is Node 1

Run `/opt/oracle/acslsha/setup.py` and select Action 2.

Note that this is not necessarily the “primary” node running ACSLS. In order to check the primary node, switch users to `acsss` and run the `acsss status` command. The node running ACSLS is the primary. Also see the section on logging to determine which node is currently the primary.

## Performing a Forced Fail Over

In order to force a failover, simply stop the `acslsha` service on the primary (active node). It will fail over to the secondary, which will then become the new primary.

Note that the secondary will reboot the original primary in order to ensure that all `acslsha` services are not active. At this time, you may start `acslsha` as the secondary on the inactive node.

## Performing a Graceful Shutdown

In order to gracefully shut down `acslsha` and all of its resources (ACSLs, the storage monitor, and the LogicalHostIP) stop the secondary node first. You can then stop the primary as follows, and both nodes will remain idle and not rebooted:

```
#systemctl stop acslsha
```

## Performing Maintenance on One or Both Nodes

To perform maintenance:

1. Perform a graceful shutdown on node 2:

```
systemctl stop acslsha
```

2. Perform a graceful shutdown on node 1:

```
systemctl stop acslsha
```

## Patching or Upgrading ACSLS 8.5.1

To patch or upgrade:

1. Stop ACSLS HA on both nodes (see [Performing a Graceful Shutdown](#)).
2. Follow the ACSLS patching or upgrade procedures outlined in the *ACSLs Administrator's Guide*. Ensure that the NFS file system is mounted to `/export/home` on the node that you are currently updating.

## Recovering From a Corrupt ACSLS Database

To recover from a corrupt database:

1. Ensure that ACSLS HA is stopped on both nodes (see [Performing a Graceful Shutdown](#)).
2. Manually mount the NFS file system to `/export/home` on one node only.
3. Start ACSLS on the node from which you mounted `/export/home`.
4. ACSLS automatically enters recovery mode and rebuilds the database.
5. Shut down ACSLS when the recovery operation is complete.
6. Unmount the NSF file system.
7. Start ACSLS HA on both nodes.

## Running `setup.py` While Any Node Is Currently Running ACSLSHA

**Do not** run `setup.py` while any node is currently running ACSLS HA. This will cause ACSLS HA to stop and failover.

# Index

## A

---

### ACSL S

- database recovery, [6-8](#)
- installing on the adjacent node, [5-2](#)
- installing on the first node, [5-1](#)
- patching or upgrading, [6-8](#)

### ACSL S HA

- description, [1-1](#)
- high level installation, [1-3](#)
- installing, [6-1](#)
- system requirements, [1-1](#)

### acslsha service

- logging, [6-5](#)
- starting and stopping, [6-4](#)

## B

---

bonding, [2-4](#)

## C

---

### client options

- ACSL S HA, [1-2](#)

### configuring

- ACSL S HA Ethernet, [2-4](#)
- ACSL SHA NFS server, [3-2](#)
- etc hosts file, [2-1](#)
- multipath bonded network, [2-2](#)
- port mapping, [2-2](#)

## D

---

database recovery, [6-8](#)

determining node 1, [6-7](#)

### documentation

- accessing, [4-2](#)

### downloading

- ACSL S 8.5.1 for Linux, [4-1](#)
- ACSL S 8.5.1 HA for Linux, [4-1](#)

## E

---

### etc hosts file

- configuring, [2-1](#)

### Ethernet

- configuring, [2-4](#)

## F

---

### failover

- performing, [6-7](#)

## I

---

### installing

- ACSL S HA, [6-1](#)
- ACSL S on adjacent node, [5-2](#)
- ACSL S on first node, [5-1](#)

## L

---

logging acslsha service, [6-5](#)

## M

---

### maintenance

- nodes, [6-7](#)

mounting NFS file system, [3-3](#)

multipath bonded network configuration, [2-2](#)

## N

---

network interface bonding, [2-4](#)

### NFS options

- ACSL S HA, [1-2](#)

### NFS server

- configuring for ACSLSHA, [3-2](#)

### node 1

- determining, [6-7](#)

## P

---

patching ACSLS 8.5.1, [6-8](#)

port mapping example, [2-2](#)

## R

---

requirements  
ACSLS HA, [1-1](#)  
client options, [1-2](#)  
NFS options, [1-2](#)  
server options, [1-2](#)  
software, [1-2](#)

## S

---

server options  
ACSLS HA, [1-2](#)

server options (*continued*)

setup.py  
  running, [6-2](#)  
shutdown  
  performing, [6-7](#)  
software requirements  
  ACSLS HA, [1-2](#)  
starting acslsha service, [6-4](#)

## U

---

upgrading ACSLS 8.5.1, [6-8](#)