StorageTek Automated Cartridge System Library Software

High Availability Cluster Installation, Configuration, and Operation

Release 8.4

E64905-01

December 2015

ORACLE®

StorageTek Automated Cartridge System Library Software High Availability Cluster Installation, Configuration, and Operation, Release 8.4

E64905-01

Contents

List of Figures

Preface

The guide contains guidelines and procedures for installing and configuring Oracle's StorageTek Automated Cartridge System Library Software High Availability (ACSLS HA) 8.4 Cluster software on both Solaris SPARC-based systems and x86-based systems.

ACSLS HA 8.4 is specifically designed to support ACSLS 8.4 on Solaris 11.2 with ZFS file systems. This version supports ACSLS software installation in any user-defined file system.

Audience

This document is intended for experienced UNIX System Administrators having a good understanding of the Solaris 11 operating system with ZFS, and a working knowledge of Solaris Cluster 4.*x*.

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 UNIX 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.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

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.

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 system integration.

There are multiple hardware and software components associated with an ACSLS HA system and the full installation procedure may take several days. For existing production library environments, customers are advised to install a simple, standalone ACSLS server to handle library production while the ACSLS HA installation is in progress.

The configuration is a two-node cluster. 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.

ACSLS HA leverages the monitor and failover features in Solaris Cluster and the multipath features in Solaris operating system to provide resilient library control operation with minimal downtime. Solaris offers IP multipathing to ensure uninterrupted network connectivity and Multipath disk I/O with RAID 1 to ensure uninterrupted access to system data. Solaris Cluster watches the health of system resources including the operating system, internal hardware and external I/O resources and it can manage a system switch over if needed. The ACSLS HA agent monitors the ACSLS application, its database, its file system, and connectivity to StorageTek library resources, invoking the Solaris Cluster 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 cluster framework and to the rest of the world. This identity is transferred automatically as needed between the cluster 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 here. If desired, Advanced Customer Services from Oracle can be arranged to advise, to assist, or to handle the entire installation.

For ACSLS documentation, refer to the Oracle Technical Network (OTN) at:

`http://docs.oracle.com/`

System Requirements

An ACSLS HA server configuration consists of two Solaris server nodes sharing an external RAID disk array

Client Options

ACSLS 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.

Logical libraries presented to fibre-channel clients using the SMCE (SCSI Media Changer Emulation) are not supported in this product.

Server Options

ACSLS HA 8.4 should run on any system that meets the minimum hardware requirements for Solaris Cluster 4.2. For specific server platform requirements, consult the document entitled *Oracle Solaris Cluster System Requirements*.

`http://www.oracle.com/technetwork/server-storage/solaris-cluster/documentation/sysreq-cluster-166689.pdf`

To avoid any single point of failure, each ACSLS HA server node must be configured with the following:

- Dual (Redundant) Power Supply
- Six 10/100/1000 base-T Ethernet ports
- Two fibre-channel ports (if using FC attached storage)
- Two SAS ports (if using SAS-attached storage)

If the system is intended for use with logical libraries, at least one additional dedicated fibre-channel port is required in configurations supporting SCSI client applications.

One additional fibre-channel port is required for any intended fibre-attached libraries, such as the SL500 or SL150.

Storage Array Options

For supported disk array subsystems, consult the document entitled *Oracle Solaris Cluster Storage Partner Program*.

`http://www.oracle.com/technetwork/server-storage/solaris-cluster/partnerprogram-cluster-168135.pdf`

Network Requirements

You should reserve a total of seven IP addresses.

1 - Logical Host (Cluster virtual IP (VIP)

2 - Node-1 IP address

3 - Node-2 IP address

4 - Library interface-1 source IP address (Node-1)

5 - Library interface-2 source IP address (Node-1)

6 - Library interface-1 source IP address (Node-2)

7 - Library interface-2 source IP address (Node-2)

Ideally, Library interface-1 should be on a different subnet from Library interface-2 (see Figure 2–1).

Software Requirements

ACSLS HA 8.4 requires the following software components:

- Oracle Solaris 11.2 (SPARC or X86)
- Oracle Solaris Cluster 4.2

Pre-Installation Checklist for ACSLS HA

Before installing a new ACSLS HA system or upgrading an ACSLS HA system to a new release, determine and record information about the customer environment in which ACSLS HA is installed.

Completing this pre-installation checklist eliminates risk. This checklist also ensures that your installation proceeds smoothly and is not delayed while you are waiting for details about the customer environment.

Oracle Support Personnel

- Who are the local Oracle people who support this customer?
- Do any of them have experience with Solaris system administration?
- Do any of them have experience with ACSLS?
- Do any of them have experience with ACSLS HA?

Customer Support Personnel

- Who are the customer's system administration staff who will support activities such as the ACSLS HA servers and the customer's network?
- Do any of them have experience with Solaris system administration?
- Do any of them have experience with ACSLS?
- Do any of them have experience with ACSLS HA?
- Who are the network administrators?

Hardware for the ACSLS HA System

- Oracle Sun servers - model?
- Solaris release and update level?
- Memory (10 GB minimum).
- Dual boot drives on each server are required to mirror the disks.
- What is the model of the shared disk? Is it supported by Solaris Cluster?
- SAS or Fibre HBAs used to connect each ACSLS server to the shared disk array.

- Six Ethernet ports required on each ACSLS server.
- If ACSLS is managing Fibre-attached libraries (SL500 or SL150) or presenting logical libraries using Fibre target mode ports, a Fibre HBA is required on each ACSLS server.
- What power cords are needed to connect the Solaris servers and the disk array to the customer's environment? For example, the plugs must match the power outlets at the customer site to avoid a delay in the HA installation.
- Before starting an HA installation, make sure the HA servers and shared disk array are configured correctly:
  - To satisfy the six-port requirement for Ethernet connections, each HA server must be equipped with an extra Network Interface Controller (NIC) card.
  - Be aware of the interface type for the external shared disk array and ensure that a compatible HBA is configured on each server.
  - If ACSLS is to communicate with Fibre-attached libraries, such as SL500s or SL150s, Fibre HBAs are required
  - If ACSLS presents logical libraries using Fibre target mode, Qlogic Fibre HBAs are required.

Network Information

Review the following network information.

IP Addresses and Host Names Assigned to the Two ACSLS HA Servers

- A total of nine IP addresses are needed for the installation:
  - Local address of ACSLS server node-1
  - Local address of ACSLS server node-2
  - Logical host address (virtual IP address shared between two HA nodes)
  - Library connection-a from node-1
  - Library connection-b from node-1 (for dual TCP/IP or multi-TCP/IP redundancy)
  - Library connection-a from node-2
  - Library connection-b from node-2 (for dual TCP/IP or multi-TCP/IP redundancy)
  - ILOM on node-1
  - ILOM on node-2
- Will ACSAPI client(s) communicate with ACSLS?
  - After a failover event, the active node assumes the virtual IP address. The virtual IP address is accessible to ACSLS clients, no mater which node is active.
  - Do you know how to set the ACSLS HA virtual IP address or hostname on the ACSAPI client(s)? This setting depends on the ISV application running on the client.
- The IP addresses that ACSLS uses to communicate with TCP/IP-attached libraries (SL8500s, SL3000s, and 9310s).

- To avoid a single point of failure, it is best to route each redundant library connection through its own separate subnet. To prevent problems due to high-volume network traffic, the subnet should be reserved for library communication and not subject to interference from general network chatter.
- The IP addresses and password(s) needed to access the service processor (for example, ILOM or ALOM) on each ACSLS HA node.

Communication with HLI Libraries

Is the communication between ACSLS and TCP/IP-attached libraries on a subnet(s) protected from broadcast traffic?

Firewalls

- Are there any firewalls between ACSAPI clients and the ACSLS HA system?
- Are there any firewalls between the ACSLS HA system and the libraries that it manages?

If any firewalls exist, refer to the “Firewall Security Option” appendix in the *ACSLS 8.4 Administrator's Guide* for details about how to configure ACSLS and ACSAPI clients to communicate through the firewalls.

SCSI Media Changer Using Fibre

- Does this ACSLS system present logical libraries to clients using Fibre Channel target mode ports? If so, this requires a QLogic Fibre HBA.

Installation Media

- Will there be direct or indirect access from the HA servers to the Internet to allow software downloads from the Oracle edelivery website, including Oracle Solaris Cluster, ACSLS, ACSLS HA, and other required packages or patches?

  If access is indirect, ensure that you can transfer files from the Internet to the HA servers.
- Assuming that software is to be downloaded directly from the Internet to the local machines, ensure that complete proxy information for browser configuration is available at install time.

Client (Backup or ILM) Applications that Communicate with ACSLS

- Will ACSAPI client(s) (such as backup or Information Lifecycle Management applications) communicate with ACSLS?
  - If ACSAPI clients will communicate with ACSLS, what are the client applications (such as NetBackup, Oracle SAM).
  - What versions of these clients are used?
  - If the client(s) run on Windows and use LibAttach, what version of LibAttach are they running?
- Do client applications communicate with ACSLS logical libraries using Fibre Channel target mode ports?
  - What are the client applications (such as NetBackup and Oracle SAM)?
  - What versions of these clients are used?

ACSLS User IDs and Group

ACSLS requires the following user IDs in the `acsls` group: `acsss`, `acsdb`, and `acssa`.

Can these user IDs and the `acsls` group be set locally on the ACSLS HA servers, or will they need to be integrated into the site's central user and password management system?

High Level Installation Procedure

A full ACSLS HA installation involves the following steps:

1. Install two Solaris platform servers attached to a common external fibre-channel or SAS2 disk array. Install Solaris 11.2 on each server.

   Refer to the document, *Installing Oracle Solaris 11 Systems*, available from the Oracle Technology Network library:

   `http://www.oracle.com/technetwork/documentation/solaris-11-192991.html`

2. Configure the basic Solaris system.

   - User access privileges
   - Multipath network access and disk I/O

   See "Configuring the Solaris System for ACSLS HA" on page 2-1.

3. Configure the ZFS file systems.

   - root storage pool
   - acsls storage pool

   See "File System Configuration with ZFS" on page 3-1.

4. Download software packages. See "Downloading Software Packages" on page 4-1.

   - ACSLS 8.4.0
   - Solaris Cluster 4.2
   - ACSLS HA 8.4.0

   See "Downloading Software Packages" on page 4-1.

5. Install ACSLS 8.4.0 and patch update (if any). See "Installing ACSLS 8.4" on page 5-1.

6. Install Solaris Cluster 4.2 and patch update (if any). See "Installing Solaris Cluster 4.2" on page 6-1.

7. Install ACSLS HA 8.4.0. See "ACSLS HA 8.4 Installation and Startup" on page 7-1.

8. Fine tune Cluster operation for ACSLS HA. See "Fine Tuning ACSLS HA" on page 8-1.

2

Configuring the Solaris System for ACSLS HA

Once the basic Solaris 11.2 system is installed, there are three contexts that entail specific configuration requirements for Solaris Cluster and ACSLS HA.

1. Configure `/etc/hosts`.
2. Configure access privileges for the `root` user.
3. Configure network interface for multipath internet access.
4. Configure multipath disk I/O.

Configuring /etc/hosts

Your `/etc/hosts` file on each node should contain entries for the localhost, the two cluster node names and their ip addresses, and the logical host.

```
127.0.0.1           localhost   loghost
129.99.99.101       thisNode.domain.com   thisNode
129.99.99.102       sisterNode
129.99.99.100       logicalHost.domain.com    logicalHost
```

Access Configuration for `root`

Solaris cluster monitoring and fail over operations are handled by the `root` user who must have control over the entire cluster from either node. The `root` user is established as a trusted user, providing private network access between nodes with secure shell (`ssh`) authentication

1. Enable direct root access to each node from an external system. Edit the file, `/etc/user_attr`, and comment out (or remove) the line that specifies a role for `root`.

   ```
   # root::::type=role
   ```

2. Allow `root` to log into the system from points other than the system console.

   Edit the file, `/etc/default/login`, and comment out the line that specifies console only access.

   ```
    # CONSOLE=/dev/console
   ```

3. Permit secure shell login access for `root`.

   Edit the file, `/etc/ssh/sshd_config`, and specify **yes** to `PermitRootLogin`.

   ```
   PermitRootLogin=yes
   ```

   Repeat steps 1-3 on the adjacent node.

4. Establish `root` as a Trusted User. This sets a login protocol for root to each node from the sister node for which the authentication is established without the need for a password.

   a. Create a public/private RSA key pair. To allow login from one node to the other without a password, do not enter a passphrase.

   ```
   # cd /root/.ssh
   # ssh-keygen -t rsa
   Enter file in which to save the key (//.ssh/id_rsa): ./id_rsa
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in ./id_rsa.
   Your public key has been saved in ./id_rsa.pub.
   The key fingerprint is:
   1a:1b:1c:1d:1e:1f:2a:2b:2c:2d:2e:2f:ea:3b:3c:3d root@node1
   ```

   This creates two files in the `/root/.ssh` directory: `id_rsa` and `id_rsa.pub`.

   b. Copy `id_rsa.pub` to the `/root/.ssh` directory on the sister node:

   ```
   # cat id_rsa.pub | ssh root@node2 \
     'cat >> /root/.ssh/authorized_keys'
   Password:
   ```

   c. With the authentication key in place, test the ability to assert commands remotely without a password.

   ```
   # hostname
   node1
   # ssh root@node2 hostname
   node2
   ```

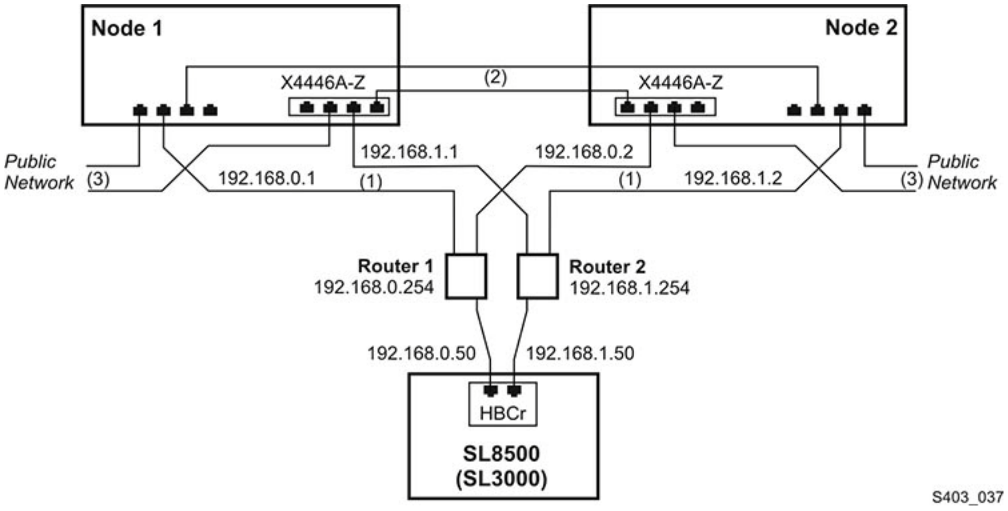
Multipath Network Configuration

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, using Internet Protocol Multi Pathing (IPMP) on Solaris. Internet Protocol Multi Pathing 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.

ACSLS HA requires redundant network connections for:

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

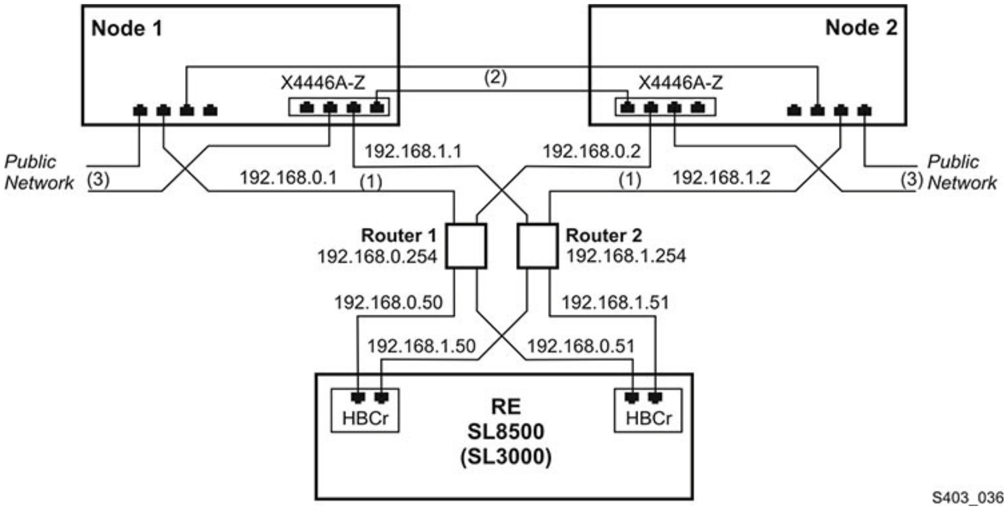
Figure 2–1 Single HBCr Library Interface Card Connected to Two Ethernet Ports on each Server Node



The figures in this section show eight Ethernet ports accessible with two separate controllers on each server. Six ports provide the three redundant connections. Two ports in this configuration remain unused. Despite the seeming complexity, there are only three dual-path Ethernet connections from each server:

- Server-library communication
- Server-to-server heartbeat exchange over a private network
- Server-to-client communication over a public network

Figure 2–2 Dual-HBC Configuration on a Library with Redundant Electronics



In a library with redundant electronics, there are two independent paths from each server node to each HBCr library controller. If communication to both ports on one HBCr interface should fail, ACSLS HA invokes an automatic switch to the alternate HBCr card. All of this is accomplished without the need to fail over to the alternate server node.

The Public Interface and IPMP

Solaris IPMP provides a mechanism for building redundant network interfaces to guard against failures with NICs, cables, switches or other networking hardware. When configuring IPMP on your Solaris host combine two or more physical network interfaces into a single IPMP group.

To view a mapping of network interface names to physical devices use the command, `dladm show-phys`:

Example:

```
# dladm show-phys
LINK       MEDIA         STATE      SPEED  DUPLEX    DEVICE
net2       Ethernet      up         100    full      ixgbe1
net3       Ethernet      up         10000  full      ixgbe3
net0       Ethernet      up         10000  full      ixgbe2
net1       Ethernet      up         1000   full      ixgbe0
```

To show the state of configured network interfaces, use `ipadm`.

Example:

```
# ipadm
NAME          CLASS/TYPE STATE  UNDER  ADDR
lo0           loopback   ok     --     --
   lo0/v4     static     ok     --     127.0.0.1/8
   lo0/v6     static     ok     --     ::1/128
net1          ip         ok     --     --
   net1/v4    static     ok     --     129.99.99.99/24
   net1/v6    addrconf   ok     --     fe99::999:999:ff23:ee02/10
net4          ip         ok     --     --
   net4/v4    static     ok     --     129.999.99.99/24
```

Two interfaces must be configured and assigned to a common group-id for ACSLS HA. Among these interfaces can be the primary interface of the server. In this case, since the `ip` address is already assigned to the interface (and not to the group), it is necessary to unconfigure this interface, then reconfigure it under the `ipmp` group.

Since network communication is interrupted during this operation, it is necessary to conduct the following steps from the server console.

To unconfigure the existing primary interface:

```
ipadm delete-addr <primary interface>
```

Example:

```
# ipadm delete-addr net0/v4
```

```
ipadm delete-ip <primary interface>
```

Example:

```
# ipadm delete-ip net0
```

To reconfigure the primary interface:

```
ipadm create-ip <primary interface>
```

Example:

```
# ipadm create-ip net0
```

To create a second primary interface:

```
ipadm create-ip <primary_interface>
```

Example:

```
# ipadm create-ip net5
```

To create the ipmp group:

```
ipadm create-ipmp <group_name>
```

Example:

```
# ipadm create-ipmp ipmp0
```

Note: The group name must be alpha-numeric.

To assign the host ip-address to the group:

```
ipadm create-addr -T static -a <ip-address> <group_name>
```

Example:

```
# ipadm create-addr -T static -a 129.99.99.9  ipmp0
```

To add the primary interface to the group:

```
ipadm add-ipmp -i <primary_interface> <group_name>
```

Example:

```
# ipadm add-ipmp -i net0 ipmp0
```

To add a second interface to the group:

```
ipadm add-ipmp -i <second_primary_interface> <group_name>
```

Example:

```
# ipadm add-ipmp -i net5 ipmp0
```

Verify your ipmp configuration using ipmp:

```
# ipadm
NAME                 CLASS/TYPE STATE        UNDER      ADDR
ipmp0                ipmp       ok           --         --
   ipmp0/v4          static     ok           --         123.45.67.89/8
lo0                  loopback   ok           --         --
   lo0/v4            static     ok           --         127.0.0.1/8
   lo0/v6            static     ok           --         ::1/128
net0                 ip         ok           ipmp0      --
net5                 ip         ok           ipmp0      --
```

Observe that two network interfaces were configured under the ipmp0 group. Observe that a version-4 ip address was assigned to the ipmp0 group.

You may need to reboot the system to commit these changes and to establish network communications under the new configuration

Repeat network configuration on the sister node.

The mapping between the ipmp group and the cluster public IP address is established when you start the cluster with start_acslsha.sh. See "ACSLS Cluster Operation" on page 9-1.

The Library Interface

Two remaining network interfaces are needed for the library configuration. For this example, net1 and net6 are used. Notice (in Figure 2–1 and Figure 2–2) that these two connections are routed over separate subnets to eliminate a router as single point of failure between the clustered server and the library.

1. Create the two network interfaces on each node.

```
# ipadm create-ip net1
# ipadm create-ip net6
```

2. Assign the ip-address for each interface.

```
# ipadm create-addr -T static -a <ip-address> net1/v4
# ipadm create-addr -T static -a <ip-address> net6/v4
```

   The address object is typically named using the interface and the protocol version: net1/v4

3. Verify your ipmp configuration:

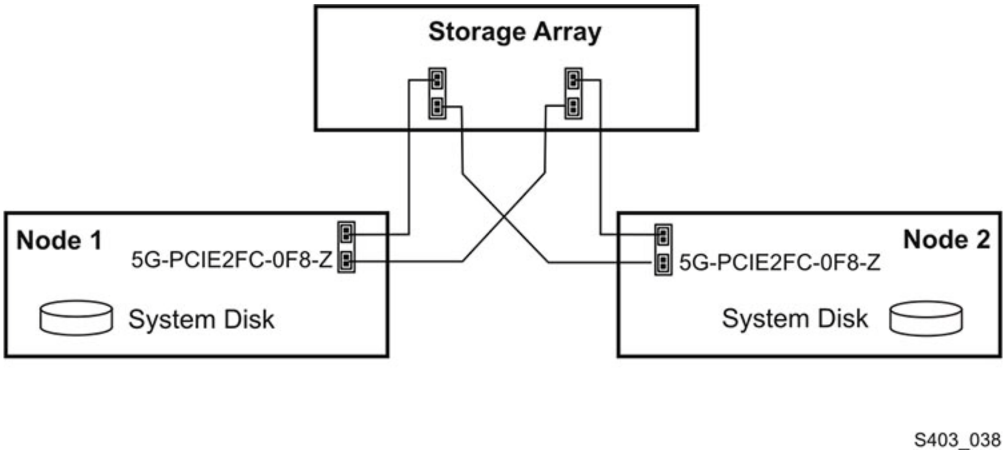
```
# ipadm
NAME              CLASS/TYPE STATE        UNDER      ADDR
ipmp0             ipmp       ok           --         --
   ipmp0/v4       static     ok           --         123.45.67.89/8
lo0               loopback   ok           --         --
   lo0/v4         static     ok           --         127.0.0.1/8
   lo0/v6         static     ok           --         ::1/128
net0              ip         ok           ipmp0      --
net1              ip         ok           --         --
   net1/v4        static     ok           --         192.168.0.1/8
net5              ip         ok           ipmp0      --
net6              ip         ok           --         --
   net6/v4        ip         ok           --         192.168.1.1/8
```

4. Reboot each node for these settings to take effect.

Multipath Disk Configuration

The external shared disk array connects to the two hosts, each with redundant fibre or SAS connections between the server and the disk array

Figure 2–3 Two Fibre Connections Per Server to External Shared Storage Array



The array should be setup to present two virtual drives to the attached host.

Solaris 11.2 takes care of multipath (MPXIO) settings automatically if it senses more than one path to a disk device. Verify that your Solaris system is properly configured with redundant connections to each virtual disk in the array.

```
# mpathadm list lu

/dev/rdsk/c0t600A0B800049EE1A0000840552D3E2F9d0s
       Total Path Count: 2
       Operational Path Count: 2
/dev/rdsk/c0t600A0B800049EDD600000DAD52D3DA90d0s2
       Total Path Count: 2
       Operational Path Count: 2
```

The `mpathadm` display should reveal two disk devices, each with an operational path count of 2.

If you run ACSLS HA on a SPARC system, `mpathadm` displays nothing until you have configured MPXIO. If the display reveals nothing, or reveals an improper configuration, proceed at this point to configure MPXIO using `stmsboot -e` on your Solaris 11.2 system. Refer to the document, *Oracle Solaris Administration: SAN Configuration and Multipathing* from the Oracle Technology Network library: http://docs.oracle.com/cd/E23824_01/html/E23097.

Be aware that Solaris I/O multipathing-enabled devices receive new names that indicate that they are under Solaris I/O multipathing control. A device has a different name from its original name.

```
Original device name:    c1t0d0
Name After stmds boot:   c0t600A0B800049EDD600000C9952CAA03Ed0
```

3

File System Configuration with ZFS

Solaris 11.2 is based on a ZFS file system. Disk I/O, disk partitioning, and disk mirroring (or RAID) are handled entirely by ZFS. Consequently, there should be no need to partition the disk (as was typically done with UFS file systems). The whole system disk should be presented as a single partition.

Your storage array is already configured with RAID, so it is not essential to configure an additional level of RAID using ZFS for your ACSLS file system. ZFS RAID is essential if you are using simple JBOD disks, but additional RAID is optional if you employ a qualified disk array.   The examples below will illustrate either approach.

Creating a Mirrored Root File System

1. Your Solaris platform should be configured with two physical disk drives. Partition the system disk and its mirror drive for optimal ZFS performance.

   On a new system before the operating system installation, each of the system disk drives can be partitioned so that partition-0 contains most (if not all) of the entire disk space. ZFS operates faster and more reliably if it has access to the whole disk. Ensure that the partition defined for ZFS on the second disk is the same size as that defined on the primary disk.

   On a system where Solaris 11.2 is already installed, use `format` or `fdisk` on the primary system disk to view the size of the `root` partition. Format the second system disk with a partition of equal size. Label the disk when the format is complete.

2. When the system is up, verify the `rpool` with the command, `zpool status`.

   ```
   # zpool status
     pool: rpool
     state: ONLINE
     scan: none requested
     config:
        NAME                      STATE     READ WRITE CKSUM
        rpool                     ONLINE       0     0     0
         c0t5000C5000EA48903d0s0  ONLINE       0     0     0
   ```

3. Identify the second system disk and determine its device-id.

   ```
   # echo | format
   AVAILABLE DISK SELECTIONS:
     0. c0t5000C5000EA48893d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
       /scsi_vhci/disk@g5000c5000ea48893
     1. c0t5000C5000EA48903d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
       /scsi_vhci/disk@g5000c5000ea48903
   ```

Choose the alternate device having close to the same size as the device revealed in step 2. In this example, the second disk-id is c0t5000C5000EA48893d0s

4. Add the second disk to the `rpool`.

```
# zpool attach -f rpool \
       c0t5000C5000EA48903d0 \
       c0t5000C5000EA48893d0
```

The system begins resilvering the mirrored drive, copying the contents of the boot drive to the second drive. This operation takes several minutes and should not be interrupted by a reboot.

You can monitor the progress using:

```
zpool status -v
```

Note 1: Until resilvering is complete, any status display shows the disk to be in a degraded mode. The disk remains in a degraded state while information is being copied from the primary disk to the mirror

Note 2: If the `zpool attach` fails because the disk is labeled as an EFI disk, then follow the process described in the document, *Solaris Admin: Devices and File Systems:* `http://docs.oracle.com/cd/E23824_01/pdf/821-1459.pdf`. This process converts the EFI disk to SMI is as follows:

```
# format -e
  (select the drive to serve as the rpool mirror).
  format> partition
  partition> print
  partition> label
    (specify label type "0")
    Ready to label? y
  partition> modify
    (select "1" All free Hog)
    Do you wish to continue ... yes
    Free Hog Partition[6]? (specify partition "0")
    (Specify a size of "0" to the remaining partitions)
    Okay to make this current partition table? yes
    Enter table name: "c1t1d0"
    Ready to label disk? y
 partition> quit
 format> quit
```

5. Confirm the mirrored `rpool` configuration.

```
# zpool status
  pool: rpool
 state: ONLINE
  scan: resilvered 6.89G in 0h3m with 0 errors
config:
        NAME                       STATE     READ WRITE CKSUM
        rpool                      ONLINE       0     0     0
          mirror-0                 ONLINE       0     0     0
            c0t5000C5000EA48903d0  ONLINE       0     0     0
            c0t5000C5000EA48893d0  ONLINE       0     0     0
```

Repeat this operation on the adjacent node.

Creating a Mirrored File System for the ACSLS Application

The ACSLS file system resides in a `zpool` on the external shared storage array. The examples below employ a simple mirrored array (RAID 1) using only two disks. These may be real drives, but are most likely virtual devices presented as discrete drives from the attached storage array.

The storage array is already configured with RAID, so it is not essential to configure an additional level of RAID using ZFS for your ACSLS file system. ZFS RAID is essential if using simple JBOD disks, but additional RAID is optional if employing a qualified disk array. The examples below illustrate either approach.

1. Prepare the shared storage array.

   In standard configurations, use a single virtual drive from the disk array. Otherwise, a ZFS RAID mirroring configuration uses two virtual drives of equal size. The admin tool can be used with the disk array or the Solaris format utility to partition the two virtual drives so they are of equal size.

2. Determine the intended base directory for the ACSLS installation.

   ACSLS 8.4 is installable in any file system. The base file system chosen should not already exist in the system `rpool`. If it already exists, the existing file system should be destroyed before creating it under the new `zpool`.

   If the default `/export/home` base directory is used for ACSLS, it is necessary to destroy the `/expor`t file system from the default `root` pool in Solaris 11.2.

   To confirm whether `/export/home` is attached to the `rpool`, run the command:

   ```
   # zfs list
   ```

   To detach `/export/home` from `rpool`, first save any files or directories to be preserved. Ensure that no users' home directories are currently active in `/export/home`. Then use `zfs destroy` to remove everything under `/export`:

   ```
   # zfs destroy -r rpool/export
   ```

   Repeat this step to detach `rpool/export` on the adjacent node.

3. Use `format` to identify the device names of the drives on the attached disk array:

   ```
   # echo | format
   AVAILABLE DISK SELECTIONS:
      0. c0t5000C5000EA48893d0 <FUJITSU-MAY2073RCSUN72G-0501-68.37GB>
      /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@0,0
      /dev/chassis/SYS/HD0/disk
   1. c0t5000C5000EA48893d0 <FUJITSU-MAY2073RCSUN72G-0501-68.37GB>
      /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd@1,0
      /dev/chassis/SYS/HD1/disk
   3. c0t600A0B800049EDD600000C9952CAA03Ed0 <SUN-LCSM100_F-50.00GB>
      /scsi_vhci/disk@g600a0b800049edd600000c9952caa03e
   4. c0t600A0B800049EE1A0000832652CAA899d0 <SUN-LCSM100_F-50.00GB>
      /scsi_vhci/disk@g600a0b800049ee1a0000832652caa899
   ```

   In this example, there are two system disks and the two virtual disks presented from the disk array having device names beginning with `c0t600A...`

4. Create the `acslspool`.

   For standard configurations using a qualified disk array, create the `acslspool` as follows:

   ```
   # zpool create -m /export/home acslspool\
   ```

```
  /dev/dsk/c0t600A0B800049EDD600000C9952CAA03Ed0
```

If ZFS RAID is added as suggested in step 1, create a mirrored configuration as follows:

```
# zpool create -m /export/home acslspool mirror \
   /dev/dsk/c0t600A0B800049EDD600000C9952CAA03Ed0 \
   /dev/dsk/c0t600A0B800049EE1A0000832652CAA899d0
```

5. Verify the new `acslspool`.

```
# zpool status acslspool
  pool: acslspool
 state: ONLINE
  scan: none requested
config:
  NAME                                       STATE   READ WRITE CKSUM
  acslspool                                  ONLINE   0     0     0
    mirror-0                                 ONLINE   0     0     0
      c0t600A0B800049EDD600000C9952CAA03Ed0  ONLINE   0     0     0
      c0t600A0B800049EE1A0000832652CAA899d0  ONLINE   0     0     0
```

Note: When using a RAID disk array, the mirrored ZFS configuration is optional.

6. Create a test file in the new pool and verify.

```
# cd /export/home
# date > test
# ls
test
# cat test
Tue Jan  7 11:48:05 MST 2015
```

7. Export the pool.

```
# cd /
# zpool export acslspool
```

8. Log in to the adjacent node (which is referred to as the new current node).

9. From the new current node, confirm that `/export/home` (or the intended file system for ACSLS) is not mounted anywhere in the `root` pool.

```
# zfs list
```

If the file system exists in the `rpool`, repeat step 2 (above) on this current node.

10. From the new current node, import the `acslspool` and verify that `acslspool` is present on this node.

```
# zpool import acslspool
# zpool status
  pool: acslspool
 state: ONLINE
  scan: none requested
config:
NAME                                       STATE   READ WRITE CKSUM
acslspool                                  ONLINE   0     0     0
  mirror-0                                 ONLINE   0     0     0
```

```
    c0t600A0B800049EDD600000C9952CAA03Ed0  ONLINE   0     0     0
    c0t600A0B800049EE1A0000832652CAA899d0  ONLINE   0     0     0
```

If `zpool import` failed, you can attempt the operation with `zpool import -f`.

Note: When using a RAID disk array, the mirrored ZFS configuration is optional.

11. Verify the test file is present on the new current node.

```
# cd /export/home
# ls
test
# cat test
Tue Jan  7 11:48:05 MST 2015
```

4

Downloading Software Packages

This chapter describes the procedures for downloading the major software packages and patches required for ACSLS HA, as well the product documentation.

Downloading Software Packages

Three major software package are required for ACSLS HA.

- ACSLS 8.4
- Oracle Cluster 4.2
- ACSLS HA 8.4

You must download the software packages to each server node. It is recommend the packages be placed in the `/opt` directory.

Packages are available from the Oracle Software Delivery Cloud.

`https://edelivery.oracle.com/`

Downloading ACSLS 8.4

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. Sign in with the username and password provided by your Oracle support representative.
3. Read and accept the export restrictions.
4. Check the Filter Products By Programs box.
5. Enter **acsls** for the Product and select StorageTek Automated Cartridge System Library Software (ACSLS).
6. Click **Select Platform**, then your Solaris platform (Solaris or X86). Click **Select**.
7. On the Selected Products screen, click **Continue**.
8. Under Available Releases, check the box for the ACSLS 8.4.0.0.0 release for your Solaris platform and click **Continue**.
9. On the Oracle Terms and Restrictions screen, review and accept the terms of the licenses. Click **Continue**.
10. Select the `V77685-`*xx* package and save the zip file to the location of your choice.

See "Installing ACSLS 8.4" on page 5-1 for procedures.

Downloading PostgreSQL 8.4 (optional)

The packages for PostgreSQL 8.3 are included when you download ACSLS 8.4. These work fine and are installed automatically when ACSLS 8.4 is installed. However, if you prefer to install PostgreSQL 8.4, download the `postgres 8.4 bz2` file from `http://www.postgresql.org/` to the `/opt` directory on each node. When ACSLS is installed, it finds the `.bz` file in `/opt` and installs PostgreSQL 8.4 automatically. For detailed instructions, refer to the section "INSTALLING POSTGRESQL" in the README.txt file included with the ACSLS 8.4 package.

Downloading Oracle Cluster 4.2

Two steps are required when downloading Oracle Cluster 4.2:

- Downloading the Solaris Cluster base image.
- Determining if you need the Solaris Cluster patch update.

See "Installing Solaris Cluster 4.2" on page 6-1 for procedures.

Downloading the Solaris Cluster Base Image

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. Sign in with the username and password provided by your Oracle support representative.
3. Read and accept the export restrictions.
4. Check the Filter Products By Programs box.
5. Enter **Oracle Solaris Cluster** and select the **Enterprise Edition**.
6. Click **Select Platform** (SPARC or x86), then click **Select**.
7. On the Selected Products screen, click **Continue**.
8. Under Available Releases, select **Alternate Release** and the **4.2.0.0.0 Enterprise Edition** for your Solaris platform. Click **Continue**.
9. Read and accept the copyright licenses and click **Continue**.
10. Select the `V46190-`*`xx`* package and save the zip file to the location of your choice.

Determining if You Need the Solaris Patch Update

Observe your currently installed Solaris version.

```
# pkg info entire | grep Version
```

- If your Solaris version is 11.2.12 or lower, the Oracle Solaris Cluster patch update is not required.
- If your Solaris version is 11.2.13 or higher, the Cluster patch update is required.
  1. Go to the Oracle Support website.

     `http://support.oracle.com`

  2. Click **Sign In** and enter the username and password provided by your Oracle support representative.
  3. Click **Patches & Updates**.

4. Under Patch Search, click **Product or Family (Advanced)**.
5. In the Product is box, enter **Solaris Cluster**.
6. In the Release is box, click the down arrow and check **Solaris Cluster 4.2.0**. Click **Search**.
7. Look for ORACLE SOLARIS CLUSTER 4.2.5.*x.x* REPO ISO image matching your platform (SPARC or X86) and select that patch. Click **Download**.
8. Under Patch Search, verify the information and click **Download**.
9. Select the patch package and save the zip file to the location of your choice.

Downloading ACSLS HA 8.4

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. Sign in with the username and password provided by your Oracle support representative.
3. Read and accept the Export Restrictions.
4. Check Filter Products By **Programs**.
5. Enter **acsls** and select StorageTek Automated Cartridge System Library Software (ACSLS) High-Availability Agent (HA).
6. Select Platform (SPARC or x86), then click **Select**.
7. On the Selected Products screen, click **Continue**.
8. Verify ACSLS HA for your platform and click **Continue**.
9. Read and accept the copyright licenses and click **Continue**.
10. Select the zip file and click **Download**.
11. Read and accept the copyright licenses and click **Continue**.
12. Select the `V75269-`*xx* package and save the zip file to the location of your choice.

See "ACSLS HA 8.4 Installation and Startup" on page 7-1 for procedures.

Downloading Patches

Patches may be available for ACSLS 8.4, Solaris Cluster 4,2, and ACSLS HA 8.4. Check for patch updates from the Oracle Support Site:

`https://support.oracle.com`

1. Sign in with your Oracle ID and password.
2. Select the **Patches and Updates** tab.
3. In the **Search** tab, click **Product or Family (Advanced)**.
4. Follow the procedures as described above for the software package.

Downloading Product Documentation

To download related product documentation:

`http://docs.oracle.com`

For ACSLS:

1. Locate Storage and select Storage Software.
2. Select StorageTek ACSLS Manager documentation, then Automated Cartridge System Library Software 8.4.
3. Select View Library.

For Solaris Cluster:

1. Locate Operating Systems and select Operating Systems.
2. Under Oracle Solaris Cluster, select Oracle Solaris Cluster 4.2.
3. Select the desired language for Cluster 4.2 and the document.

5

Installing ACSLS 8.4

ACSLS 8.4 installation is covered in detail in the *StorageTek Automated Cartridge System Library Software 8.4 Installation Guide*. To install ACSLS 8.4, follow this high level procedure.

Installing on the First Node

1. Download ACSLS 8.4 to the /opt directory on each server.
2. Unzip the downloaded zip file.
3. Confirm that the acslspool is mounted on the current node

   ```
   # zfs list
   NAME                     USED  AVAIL  REFER  MOUNTPOINT
   acslspool               1.60G  47.4G  1.60G  /export/home
   rpool                   6.97G  60.0G  4.58M  /rpool
   rpool/ROOT              4.39G  60.0G    31K  legacy
   rpool/ROOT/solaris      4.39G  60.0G  3.17G  /
   rpool/ROOT/solaris/var  1.22G  60.0G  1.21G  /var
   rpool/VARSHARE          95.5K  60.0G  95.5K  /var/share
   rpool/dump              1.55G  60.0G  1.50G  -
   rpool/swap              1.03G  60.0G  1.00G  -
   node2:# clrg resume acsls-rg
   ```

4. Go into the ACSLS installation directory and run the package installation script:

   ```
   # cd /opt/ACSLS_8.4.0
   # ./pkg_install.sh
   ```

5. Observe /etc/passwd on this node. Take note of the user and group id numbers assigned to users acsss, acssa, and acsdb.

   ```
   # tail -3 /etc/passwd
   # grep acsls /etc/group
   ```

   When installing on the adjacent node, confirm that the assigned user id numbers on the second node agree with the corresponding id numbers listed here.

6. Source the ACSLS environment and run the package installation script.

   ```
   # . /var/tmp/acsls/.acsls_env
   # cd $ACS_HOME/install
   # ./install.sh
   ```

Note: When the install script prompts for the database backup directory, specify a directory that is mounted to the shared disk array. Use any path under the ACSLS installation directory (`$installDir`). For example, if installing ACSLS in `/export/home`, use `/export/home/backup` for the database backup files.

7. Confirm that the attached library is connected.

```
# su - acsss
$ testlmutcp <library ip address>
```

8. Run the library configuration routine.

```
$ acsss_config
```

9. Check for any patch update to ACSLS 8.4. If a patch exists, download and install it as directed.

10. Export the `acslspool` from the current node.

```
$ exit
# cd /
# zpool export acslspool
```

This operation fails if any users or operations are currently active in the ACSLS file system.

Installing on the Adjacent Node

While most of the files in the `STKacsls` package are extracted to the shared disk array (where these files have already been installed), it remains necessary to install ACSLS 8.4 on the adjacent node since ACSLS users are added to each node and numerous system files (such as SMF start/stop scripts, cron jobs) must be installed on each node.

1. Log in to the adjacent node and import the acslspool.

```
# zpool import acslspool
```

If this operation fails, try `zpool import -f acslspool`.

2. Go into the ACSLS package installation directory and install the package.

```
# cd /opt/ACSLS_8.4.0
# ./pkg_install.sh
```

The install routine advises you that some ACSLS files already exist in the installation directory (on the shared drive). You can choose to over-write them.

3. Observe the user and group ids that were assigned on this node for users `acsss`, `acssa`, and `acsdb`.

```
# tail -3 /etc/passwd
# grep acsls /etc/group
```

Confirm that these numbers agree with those assigned on the first node. By default the GID should be 100, but this number will change if GID 100 is already in use.

It is important that the UID and GID numbers agree between the two nodes. If these numbers are different on the second node, edit the /etc/passwd file so that the UID and GID numbers on the second node agree with those on the first node.

Now confirm that the group id of the files in $ACS_HOME have group ownership of acsls. If they do not, then it is necessary to uninstall and reinstall the STKacsls package.

```
# pkgrm STKacsls
# pkgadd STKacsls
```

4. Inherit the ACSLS environment and run the install shell script.

```
# . /var/tmp/acsls/.acsls_env
# cd $ACS_HOME/install
# ./install.sh
```

Note 1: When the install script prompts for the database backup directory, specify a directory that is mounted to the shared disk array. Use any path under the ACSLS installation directory (`$installDir`). For example, if installing ACSLS in `/export/home`, use `/export/home/backup` for the database backup files.

Note 2: When running `install.sh` on the second node, it is not necessary to reinstall the ACSLS GUI. When the install script advises that the `Acsls GUI` Domain already exists for the GUI, simply say **no** to the prompt to reinstall the GUI, and be sure to say **no** to the prompt to remove the GUI.

5. If a patch to ACSLS 8.4 was added on the first node, repeat the patch install on this node.

6. Confirm that the attached library is connected.

```
# su - acsss
$ testlmutcp <library ip address>
```

7. Run the library configuration routine.

```
$ acsss_config
```

To enable redundancy between the HA server and the library, refer to Figure 2–1, "Single HBCr Library Interface Card Connected to Two Ethernet Ports on each Server Node" or Figure 2–2, "Dual-HBC Configuration on a Library with Redundant Electronics". When running `acsss_config` (option-8), be sure to define two connections to each ACS and enter the library IP address for each connection.

6 Installing Solaris Cluster 4.2

Solaris Cluster Installation is covered in detail in the *Oracle Solaris Cluster Software Installation Guide*, available from the Oracle technetwork site (see "Downloading Software Packages" on page 4-1 of this document).

ACSLSHA 8.4 is supported on Solaris 11.2 with Oracle Solaris Cluster 4.2.

Cluster Package Installation

Follow this procedure to install Cluster software.

1. Create a directory, /opt/OSC.

   ```
   # mkdir /opt/OSC
   ```

2. You may have downloaded one or two iso images for the Solaris Cluster package, depending on the operation system version that you determined in "Downloading Oracle Cluster 4.2" on page 4-2. Move each Cluster package downloaded to the /opt/OSC directory.

3. Unzip the packages. Identify an iso image for each unzipped package.

4. Create a pseudo device from each iso image.

   ```
   #  /usr/sbin/lofiadm -a /opt/OSC/V46190-01.iso

   #  /usr/sbin/lofiadm -a /opt/OSC/osc-4_2_5_1_0-repo-incr.iso
   ```

   Keep track of the /dev/lofi instance numbers you created for each of these.

5. In the /opt/OSC directory, create a mount point for each pseudo device.

   ```
   # mkdir mnt

   # mkdir mnt1
   ```

6. Mount the pseudo devices to these mount points.

   ```
   # mount -F hsfs -o ro /dev/lofi/1 /opt/OSC/mnt

   # mount -F hsfs -o ro /dev/lofi/2 /opt/OSC/mnt1
   ```

7. Select one of the following:

   - If your Solaris version is 11.2.13 or higher, proceed to step 8.
   - If your Solaris version is 11.2.12 or below and you downloaded only the base Cluster image, you can now publish the repository for that base image.

     ```
     #  pkg set publisher -G  '*'  -g  file:/opt/OSC/mnt/repo  ha-cluster
     ```

Proceed to the step 13 in this section to install the package.

8. In the following steps, you copy the OSC base package into a read/write file system to merge the patch update with the base package.

   Create a read/write file system for merging the OSC packages.

   ```
   # cd /opt/OSC

   # mkdir merged_iso
   ```

9. Copy the base OSC image repository into the directory created.

   ```
   # cp -r  mnt/repo  merged_iso
   ```

10. Synchronize the two images together into the merged directory.

    ```
    # rsync -aP mnt1/repo  merged_iso
    ```

11. Rebuild the search indexes for the repository

    ```
    # pkgrepo rebuild -s  merged_iso/repo
    ```

12. Publish the `ha-cluster` repository of the merged packages.

    ```
    # pkg set-publisher -g file:/opt/OSC/merged_iso/repo ha-cluster
    ```

13. Install the Oracle Solaris Cluster package.

    ```
    # pkg install --accept ha-cluster-full
    ```

Repeat steps 1-13 on the adjacent node.

The scinstall Routine

The Solaris Cluster installation routine makes a series of checks between the two nodes to ensure that it can monitor system operation from both servers and can control startup and failover actions.

Preliminary Steps:

1. Before running `scinstall`, it is helpful to establish an environment for `root` which includes the path to the cluster utilities that have just been installed. Edit the file `/root/.profile`. Change the path statement to include `/usr/cluster/bin`.

   ```
   export PATH=/usr/cluster/bin:/usr/bin:/usr/sbin
   ```

   Be sure to make this change on each node. To inherit the new path, log out and log back in, or simply `su -`.

2. Confirm that the `config/local_only` property for `rpc/bind` is *false*

   ```
   # svccfg -s network/rpc/bind listprop config/local_only
   ```

   If this property returns *true*, then you must set it to *false*.

   ```
   # svccfg -s network/rpc/bind setprop config/local_only=false
   ```

   Now confirm:

   ```
   # svccfg -s network/rpc/bind listprop config/local_only
   ```

3. An essential hardware setup requirement for Cluster software is the existence of two private network connections, reserved to ensure uninterrupted communication for cluster operation between the two nodes.

   Figure 2–1, "Single HBCr Library Interface Card Connected to Two Ethernet Ports on each Server Node" shows these physical connections, labeled as (2). Each connection originates from a separate network adapter to ensure that no single point of failure can interrupt Cluster's internal communication. The scinstall routine checks each of the two connections to verify that no other network traffic is seen on the wire. Finally, scinstall verifies that communication is functional between the two lines. Once the physical connection is verified, the routine plumbs each interface to a private internal address beginning with 172.16.

   Before running scinstall, verify the assigned network device ID for the two network ports on each server set up for this private connection. Run dladm show-phys to view the interface assignments.

   ```
   # dladm show-phys
   ```

4. A Logical Host Name and IP address must be established to represent the cluster from either node. This logical host reliably responds to network communication whether the active host would be running from node1 or node2.

   Update the /etc/hosts file on both nodes to include the logical hostname and logical ip address. This host becomes active when you start ACSLS HA ("Configuring ACSLS HA" on page 7-2).

5. For a successful cluster installation, have the Solaris Common Agent Container enabled. Verify that the agent container is enabled.

   ```
   # cacaoadm status
   ```

   If the status response indicates that the agent container is DISABLED at system startup, then enable it as follows:

   ```
   # cacaoadm enable
   ```

Run scinstall

From one of the two nodes, run the command scinstall, and then follow this procedure:

1. From the main menu, select **Create a new cluster**.
2. From the sub menu, select **Create a new cluster**.
3. Accept initial defaults.
4. Select **Typical** install.
5. Assign a name for the cluster, such as acsls_cluster.
6. At the Cluster Nodes prompt, enter the hostname of the adjacent node. Accept the node list if it is correct
7. Define the two private node interconnections identified for this purpose. Allow the install routine to plumb TCP links to the physical connections.
8. Follow the prompts to create the cluster. Unless you have identified a specific device to serve as a quorum device, allow the scinstall routine to select the quorum device(s).

9. Don't be alarmed if the utility reports that the cluster check *failed on both nodes*. A failure is reported even for minor warnings. Review the report for each node, and look for any serious errors or violations that may be returned. The routine displays the path to a log file which reports details surrounding any errors or warnings encountered during the operation. Review the log file and correct any severe or moderately severe problems that were identified.

   The `scinstall` routine is run from one node and installs Solaris Cluster on both nodes. Observe the routine configuring one node, rebooting that node, then configuring the second node, and rebooting that second node.

Verify Cluster Configuration

1. Verify that both nodes are included in the cluster.

```
# clnode list -v
Node                Type
----                ----
node1               cluster
node2               cluster
```

2. View the list of devices available to Solaris Cluster.

```
# cldevice list -v
DID Device   Full Device Path
d1           node1:/dev/rdsk/c0t600A0B800049EDD600000C9952CAA03Ed0
d1           node2:/dev/rdsk/c0t600A0B800049EDD600000C9952CAA03Ed0
d2           node1:/dev/rdsk/c0t600A0B800049EE1A0000832652CAA899d0
d2           node2:/dev/rdsk/c0t600A0B800049EE1A0000832652CAA899d0
d3           node1:/dev/rdsk/c1t0d0
d4           node1:/dev/rdsk/c1t1d0
d5           node2:/dev/rdsk/c1t0d0
d6           node2:/dev/rdsk/c1t1d0
```

   In this example, the shared disk devices are d1 and d2 while d3 and d4 are the node1 boot devices and d5 and d6 are the node2 boot devices. Notice that d1 and d2 are accessible from either node.

3. A quorum consists of three or more devices. It is used during startup events to determine which node is to become the active node.

   Confirm that a full quorum has been configured.

```
# clquorum list -v
Quorum              Type
------              ----
d1                  shared_disk
node1               node
node2               node
```

   A second shared_disk can be optionally added to the list of quorum devices.

```
# clquorum add d2
# clquorum list -v
Quorum              Type
------              ----
d1                  shared_disk
d2                  shared_disk
node1               node
node2               node
```

If the shared disk devices are not listed, determine their device id's and then add them to the quorum.

a. Identify the device id for each shared disk.

```
# cldevice list -v
```

b. Run `clsetup` to add the quorum devices.

```
# clsetup

Select '1' for quorum.
Select '1' to dd a quorum device.
Select 'yes' to continue.
Select 'Directly attached shared disk'
Select 'yes' to continue.
Enter the device id (d<n>) for the first shared drive.
Answer 'yes' to add another quorum device.
Enter the device id for the second shared drive.
```

c. Run `clquorum show` to confirm the quorum membership.

```
# clquorum show
```

4. Review overall cluster configuration.

```
# cluster check -v | egrep -v "not applicable|passed"
```

Look for any violated instances in the list.

5. Verify the list of registered resource types.

```
# clrt list
SUNW.LogicalHostname:4
SUNW.SharedAddress:2
SUNW.gds:6
```

If `SUNW.gds` is not listed, register it.

```
# clrt register SUNW.gds
```

Confirm with `clrt list`.

7

ACSLS HA 8.4 Installation and Startup

The SUNWscacsls package contains ACSLS agent software that communicates with Oracle Solaris Cluster. It includes special configuration files and patches that insure proper operation between ACSLS and Solaris Cluster.

Basic Installation Procedure

1. Unzip the downloaded SUNWscacsls.zip file in /opt.

   ```
   # cd /opt
   # unzip SUNWscacsls.zip
   ```

2. Install the SUNWscacsls package.

   ```
   # pkgadd -d .
   ```

3. Repeat steps 1 and 2 on the adjacent node.
4. Verify that the acslspool remains mounted on one of the two nodes.

   ```
   # zpool status acslspool
   ```

   If the acslspool is not mounted, check the other node.

   If the acslspool is not mounted to either node, then import it to the current node as follows:

   ```
   # zpool import -f acslspool
   ```

   Verify with zpool status.

5. Go into the /opt/ACSLSHA/util directory on the node that owns the acslspool and run the copyUtils.sh script. This operation updates or copies essential files to appropriate locations on both nodes. There is no need to repeat this operation on the adjacent node.

   ```
   # cd /opt/ACSLSHA/util
   # ./copyUtils.sh
   ```

6. On the node where the acslspool is active, as user acsss, start the ACSLS application (acsss enable) and verify that it is operational. Resolve any issues encountered. Major issues may be resolved by removing and reinstalling the STKacsls package on the node.

   If the STKacsls package must be reinstalled, run the /opt/ACSLSHA/util/copyUtils.sh script after installing the package

7. Shutdown acsls.

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

8. Export the `acslspool` from the active node.

```
# zpool export acslspool
```

> **Note:** This operation fails if user `acsss` is logged in, if a user shell is active anywhere in the `acslspool`, or if any `acsss` service remains enabled.

9. Import the `acslspool` from the adjacent node.

```
# zpool import acslspool
```

10. Startup the ACSLS application on this node and verify successful library operation. Resolve any issues encountered. Major issues may be resolved by removing and reinstalling the STKacsls package on the node.

    If the STKacsls package must be reinstalled, run the `/opt/ACSLSHA/util/copyUtils.sh` script after installing the package.

Configuring ACSLS HA

This step creates three ACSLS resources that are managed and controlled by Solaris Cluster:

- `acsls-rs` is the ACSLS application itself.
- `acsls-storage` is the ZFS file system on which ACSLS resides.
- `<logical host>` is the virtual IP (the network identity that is common to both nodes). See "Configuring /etc/hosts" on page 2-1.

Once these resource handles are created, they are assigned to a common resource group under the name `acsls-rg`.

To configure these resources, first verify that the `acslspool` is mounted (`zpool list`), then go to the `/opt/ACSLSHA/util` directory and run `acsAgt configure`:

```
# cd /opt/ACSLSHA/util
# ./acsAgt configure
```

This utility prompts for the logical host name. Ensure that the logical host is defined in the `/etc/hosts` file, and that the corresponding i.p. address maps to the `ipmp` group defined in the chapter, "Configuring the Solaris System for ACSLS HA" on page 2-1. In addition, before running `acsAgt` configure, use `zpool list` to confirm that the `acslspool` is mounted to your current server node.

This configuration step may take a minute or more to complete. Once the resource handles have been created, the operation attempts to start the ACSLS application.

Monitoring ACSLS Cluster Operation

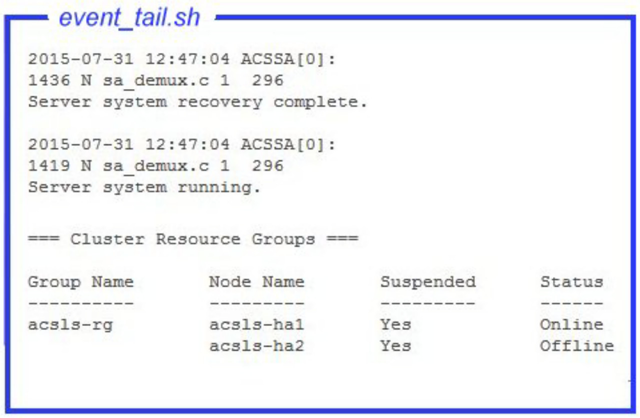
There are multiple vantage points from which to view the operation of the ACSLS Cluster. When Solaris Cluster probes the ACSLS application once each minute, it is helpful to view the results of probes as they happen. When a probe returns a status to trigger a node switch over event, it is helpful to view the shutdown activity on one node and the start-up activity on the adjacent node. It is generally helpful to have a view into the operational health of the ACSLS application over time.

The primary operational vantage point is from the perspective of ACSLS. The tail of the acsss_event.log can provide the best indication of overall system health from moment to moment.

The tool event_tail.sh in the /opt/ACSLSHA/util/ directory provides direct access to the acsss_event.log from either node. The view provided from this tool remains active, even as control switches from one node to the other. In addition to normal ACSLS activity, this tool dynamically tracks each status change of the ACSLS Cluster resource group (acsls-rg) allowing a real time view when one node is going offline and the other coming online. Assert this tool from the shell as follows:

```
# /opt/ACSLSHA/util/event_tail.sh
```

Figure 7–1 Example of event_tail.sh



```
2015-07-31 12:47:04 ACSSA[0]:
1436 N sa_demux.c 1  296
Server system recovery complete.

2015-07-31 12:47:04 ACSSA[0]:
1419 N sa_demux.c 1  296
Server system running.

=== Cluster Resource Groups ===

Group Name       Node Name       Suspended      Status
----------       ---------       ---------      ------
acsls-rg         acsls-ha1       Yes            Online
                 acsls-ha2       Yes            Offline
```

To view start and stop activity from the perspective of a single node, view the start_stop_log from that node as follows:

```
# tail -f /opt/ACSLSHA/log/start_stop_log
```

To view the results of each periodic probe on the active node:

```
# tail -f /opt/ACSLSHA/log/probe_log
```

Solaris Cluster and the ACSLS Cluster agent sends details of significant events to the Solaris System log (var/adm/messages). To view the system log on a given node, a link is provided in the directory, /opt/ACSLSHA/log:

```
# tail -f /opt/ACSLSHA/log/messages
```

The ha_console.sh Utility

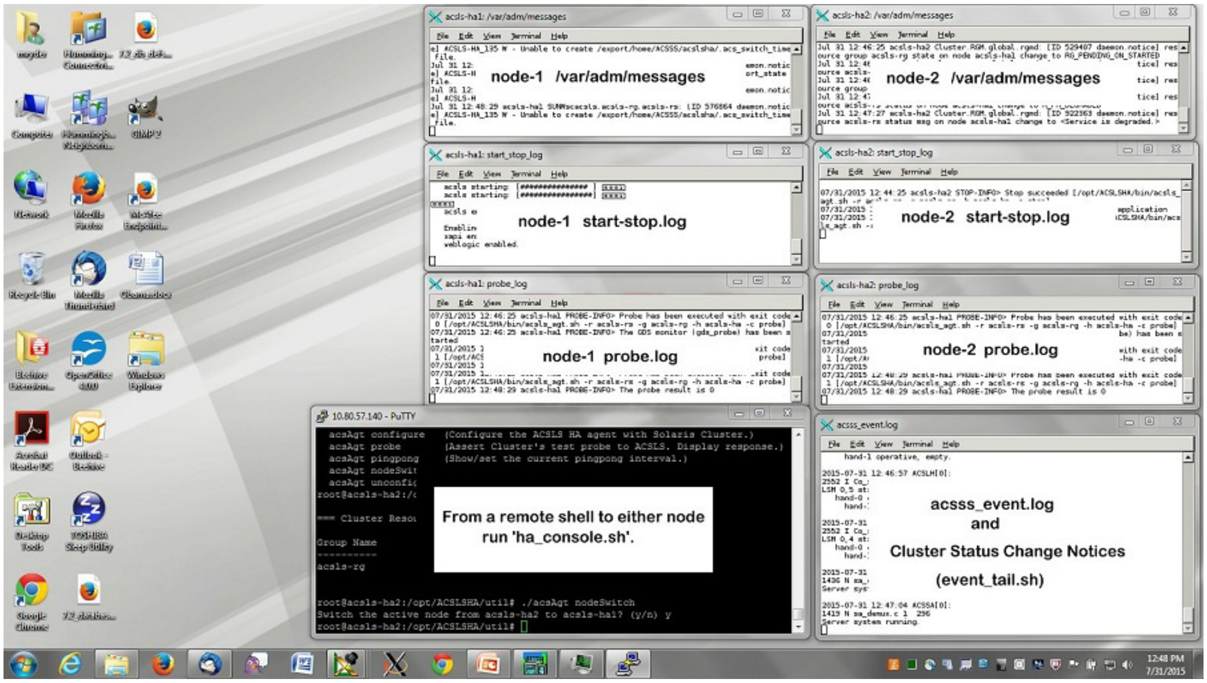
With multiple vantage points in a clustered configuration, and with migration of cluster control from one node to the other over time, it can be a challenge to follow the operational activity of the system from moment to moment from a single perspective. The ha_console.sh utility makes it easier to provide a comprehensive view.

Log in to either node on the ACSLS HA system from the remote desktop and run `ha_console.sh`. This utility looks at your login identity (`who am i`) to determine where to send the `DISPLAY`. Log in directly from the local console or desktop system to the HA node to see the display. If problems are encountered, look for messages in `gnome-terminal.log` in the `/opt/ACSLSHA/log` directory.

```
# /opt/ACSLSHA/util/ha_console.sh
```

This utility monitors all of the logs mentioned in this section from both nodes. It launches seven gnome-terminal windows on the local console screen. It can be helpful to organize the windows on the screen in the following manner:

Figure 7–2 Organizing your gnome-terminal Windows



node-1 /var/adm/messages

node-2 /var/adm/messages

node-1 start-stop.log

node-2 start-stop.log

node-1 probe.log

node-2 probe.log

From a remote shell to either node run 'ha_console.sh'.

acsss_event.log and Cluster Status Change Notices (event_tail.sh)

```
  acsAgt configure   (Configure the ACSLS HA agent with Solaris Cluster.)
  acsAgt probe       (Assert Cluster's test probe to ACSLS. Display response.)
  acsAgt pingpong    (Show/set the current pingpong interval.)
  acsAgt nodeSwit
  acsAgt unconfi
root@acsls-ha2:/

=== Cluster Reso

Group Name
----------
acsls-rg

root@acsls-ha2:/opt/ACSLSHA/util# ./acsAgt nodeSwitch
Switch the active node from acsls-ha2 to acsls-ha1? (y/n) y
root@acsls-ha2:/opt/ACSLSHA/util#
```

From a single terminal display, a comprehensive view of the entire ACSLS cluster complex is displayed.

Because the remote system sends display data to the local screen, yopen X-11 access on your local system. On UNIX systems, the command to do this is `xhost +`. On a Windows system, X-11 client software such as `xming` or `exceed` must be installed.

If you have difficulty using `ha_console.sh`, multiple login sessions can be opened from the local system to each node to view the various logs mentioned in this section.

Verifying Cluster Operation

1. Once `acslsha` has started and is registered with Solaris Cluster, use cluster commands to check status of the ACSLS resource group and its associated resources.

```
# clrg status
   === Cluster Resource Groups ===
   Group Name        Node Name        Suspended      Status
   ----------        ---------        ---------      ------
   acsls-rg         node1           No             Online
                    node2           No             Offline

# clrs status
   === Cluster Resources ===
   Resource Name       Node Name      State        Status Message
   -------------       ---------      -----        --------------
   acsls-rs            node1          Online       Online
                       node2          Offline      Offline
   acsls-storage       node1          Online       Online
                       node2          Offline      Offline
   <logical host>      node1          Online       Online
                       node2          Offline      Offline
```

2. Temporarily suspend cluster failover readiness to facilitate initial testing.

```
# clrg suspend acsls-rg
# clrg status
```

3. Test cluster switch operation from the active node to the standby.

```
# cd /opt/ACSLSHA/util
# ./acsAgt nodeSwitch
```

   Switch over activity can be monitored from multiple perspectives using the procedures described in the previous section.

4. Verify network connectivity from an ACSLS client system using the logical hostname of the ACSLS server.

```
# ping acsls_logical_host
# ssh root@acsls_logical_host hostname
passwd:
```

   This operation should return the hostname of the active node.

5. Verify ACSLS operation.

```
# su acsss
$ acsss status
```

6. Repeat steps 3, 4, and 5 from the opposite node.

7. Resume cluster failover readiness

```
# clrg resume acsls-rg
# clrg status
```

8. The following series of tests involve verification of node failover behavior.

   To perform multiple failover scenarios in sequence, lower the default pingpong interval from twenty minutes to five minutes. (See the chapter, "Fine Tuning ACSLS HA" on page 8-1 for details.) For testing purposes, it is handy to lower the default setting.

   To change the pingpong interval, go to the `/opt/ACSLSHA/util` directory and run `acsAgt pingpong`.

```
# ./acsAgt pingpong
Pingpong_interval
   current value:  1200 seconds.
   desired value: [1200] 300
Pingpong_interval : 300 seconds
```

9. Reboot the active node and monitor the operation from the two system consoles and from the viewpoints suggested in "Monitoring ACSLS Cluster Operation" on page 7-2. Verify automatic failover operation to the standby node.

10. Verify network access to the logical host from a client system as suggested in step 4.

11. Once ACSLS operation is active on the new node, reboot this node and observe failover action to the opposite node.

    If monitoring the operation using `ha_console.sh`, the windows associated with the rebooting node can be seen going away. Once that node is up again, run the command `ha_console.sh` once more on either node to restore the windows from the newly rebooted node.

12. Repeat network verification as suggested in step 4.

"ACSLS Cluster Operation" on page 9-1 provides a complete set of failover scenarios. Any number of these scenarios can be tested before placing the ACSLS HA system into production. Before returning the system to production, restore the recommended pingpong interval setting to avoid constant failover repetition.

8

Fine Tuning ACSLS HA

This chapter explains how to set up an optimal failover policy in a library complex, how to adjust the default pingpong interval to avoid unwanted fail-back events, and how to register for email notification of failover events.

Defining a Failover Policy for Library Communications

The ACSLS HA agent constantly monitors communication between ACSLS and the attached libraries. Such communication is critical for continuous ACSLS operation. But what action, if any, should be taken in a failed library communication depends upon a policy that is determined by the local ACSLS HA administrator.

A policy table, `$ACS_HOME/acslsha/ha_acs_list.txt`, allows the local administrator to define the desired failover action for any ACS that requires HA recovery. In a library communication failure, and depending on the administrator's directive, the ACSLS HA agent fails over to the alternate node if successful ACS communication has been confirmed on that node.

In multiple ACS environments, it may be desirable for the ACSLS HA system to fail over when communication with any single ACS has failed. But since any fail over action disrupts production on all attached libraries, the administrator may prefer to limit general fail over action to the more critical ACS (or ACSs) in the data center. A policy record is created in `ha_acs_list.txt` for each ACS for which cluster fail over action is required when library communication is lost. Each record has two fields:

```
ACS Number   Fail-over Action (true or false)
```

The first field is the ACS ID and the second field is the Boolean value of *true* or *false*. The logic of the policy settings is as follows:

- When the second field is *false*, the ACSLS HA agent does not initiate cluster failover action to the alternate node, even though communication to the ACS has failed and cannot be restored.
- When the second field is *true*, the ACSLS HA agent asserts cluster failover action after every attempt to reestablish communication from the primary node has failed. The system fails over only if library contact has been confirmed on the alternate node.

The default action is *false* for any ACS that is not listed in this file.

Libraries with Redundant Electronics (RE)

For libraries with Redundant Electronics (RE), the ACSLS HA agent attempts to switch communication to the alternate RE path before resorting to cluster failover action. This

RE switch action applies only to a single SL8500, an SL3000, or an older 9310 with dual LMUs. Automatic RE switching is not attempted on any partitioned library.

Setting the Failover Pingpong_interval

The Solaris Cluster Pingpong_interval is a timeout property that prevents repeated failover action if full recovery cannot be restored after the first cluster failover event.

This is a user-modifiable property for the ACSLS resource group. The default value is set to 20 minutes. With this setting, the first failover event occurs immediately when failover action is requested by the ACSLS HA agent. But if the condition which might trigger failover action is not cleared on the new cluster node, then subsequent failover action is delayed until the defined pingpong interval has expired. This prevents needless thrashing of control between one cluster node and the other until the root problem has been resolved.

To change the default setting of this property, modify the default number defined in the file, $ACS_HOME/acslsha/pingpong_interval. That number is expressed in seconds.

The default setting of 1200 seconds is a reasonable setting for most medium to large library configurations. An optimal timeout value for this property depends upon the actual number of LSMs and tape drives that exist in the library configuration. Larger library configurations take longer to recover after a failover event and so this number should be set to a longer interval for systems configured with more than ten LSMs or forty drives, or both.

A setting of 1800 (30 minutes) would be recommended for a forty-LSM configuration, while a setting of 900 (15 minutes) is recommended for smaller libraries configured with one to four LSMs.

Changes you make here do take effect until you reconfigure ACSLS HA with the command, acsAgt configure.

```
# cd /opt/ACSLSHA/util
# ./acsAgt configure
```

This command may be asserted even if the acsls-rg resource group is already active. It registers the new default setting without impacting normal HA operation.

The pingpong_interval setting can be dynamically changed for testing purposes using acsAgt pingpong. The value set with this command remains in effect until you restart the resource group with acsAgt configure.

Registering for Email Notification of System Events

Users with administrative duties may register for automatic email notification of system events, including system boot events and ACSLS HA cluster failover events.

To register for such events, users must add their email address in the respective files under the directory:

```
$ACS_HOME/data/external/email_notification/
   boot_notification
   ha_failover_notification
```

Place the email address of each intended recipient on a single line under the header remarks. Thereafter, every time the system boots or the HA cluster fails over to the standby node, each registered user is notified by email.

This capability assumes that the sendmail service has been enabled on the ACSLS server, and that network firewall constraints allow for email communication from the data center.

9

ACSLS Cluster Operation

Solaris Cluster is designed to achieve automatic system recovery under severe failure scenarios by transferring operational control from one server node to the next. But most failures in a Solaris system do not require full system switch over action to recover.

- Failures involving network communication are handled quickly and quietly by Solaris IPMP.
- System disk failures are handled silently and automatically by Solaris ZFS.
- Failures with any single disk drive in the attached storage array are recovered automatically by the storage array firmware. And where the storage array might lack the ability to recover from disk failure, Solaris ZFS is in control to provide uninterrupted disk I/O to the alternate drive in the mirrored configuration.
- If an HBA port to the shared array should fail, Solaris automatically switches to an alternate port. Similarly, if a controller module on the shared array should fail or an interconnecting cable is disconnected, Solaris instantly reverts to the alternate path that connects to the disk resource.
- Failure in a library communication path is recovered automatically by dual TCP/IP logic in ACSLS. And operations from a failed library controller card are recovered automatically by ACSLS HA logic associated with library Redundant Electronics (RE).
- If any of the multiple running processes in ACSLS should fail, the ACSLS daemon instantly reboots the failed process.
- Should the ACSLS daemon itself fail, or should any of the remaining ACSLS services stop running, the Solaris Service Management Facility (SMF) is there to instantly restart the failed service.

All of these scenarios are handled quickly and automatically without the involvement of Solaris Cluster. But if any other severe fault should impact ACSLS operation on the active server node, ACSLS HA instructs Solaris Cluster to switch control over to the alternate node.

Once it is started, ACSLS HA probes the system once every minute, watching for any of the following events to occur:

- Loss of communication to an attached library.
- Loss of network contact to the ACSLS logical host.
- Loss of contact to the RPC listener port for client calls.
- Loss of access to the ACSLS file system.
- Unrecoverable maintenance state of the acsls SMF service.

Any of these events triggers a Cluster fail over. Solaris Cluster also knows to fail over if any fatal system conditions on the active server node occurs.

Starting Cluster Control of ACSLS

To activate Cluster failover control:

```
# cd /opt/ACSLSHA/util
# ./acsAgt configure
```

The utility prompts you for the logical host name. Ensure that the logical host is defined in the /etc/hosts file, and that the corresponding i.p. address maps to the ipmp group defined in the chapter, "Configuring the Solaris System for ACSLS HA" on page 2-1. Before running acsAgt configure, use zpool list to confirm that acslspool is mounted to the current server node.

This action initiates Cluster control of ACSLS. Solaris Cluster monitors the system, probing once each minute to verify the health of ACSLS specifically and the Solaris system in general. Any condition that is deemed fatal initiates an action on the alternate node.

To check cluster status of the ACSLS resource group:

```
# clrg status
```

The display:

- Reveals the status of each node.
- Identifies which node is the active node.
- Reveals whether failover action is suspended.

Setting the Failover Policy for acsls-storage

It is advisable to set a policy in the acsls-storage resource to reboot the active node whenever communication is lost between that node and the shared RAID disk device. This action causes the active node to relinquish control when it cannot connect to the disk, allowing Solaris Cluster to pass control to the alternate node. By setting the Failover_mode from SOFT to HARD, this ensures a reboot of the active node whenever communication has been lost to the shared storage device.

To view the existing Failover_mode, run the following command:

```
#  clrs show -v acsls-storage | grep Failover
```

The Failover_mode should be set to HARD as follows:

```
# clrs set -p Failover_mode=HARD  acsls-storage
```

ACSLS Operation and Maintenance Under Cluster Control

Once cluster control has been activated, ACSLS can be operated in normal fashion. Start and stop ACSLS using the standard acsss control utility. Under cluster control, a user starts and stops ACSLS services in the same fashion as starting and stopping the application on a standalone ACSLS server. Operation is administered with these standard acsss commands:

```
acsss enable
acsss disable
acsss db
```

Manually starting or stopping `acsss` services with these commands in no way causes Solaris Cluster to intervene with failover action. Nor will the use of the Solaris SMF commands (such as `svcadm`) cause Cluster to intervene. Whenever `acsss` services are aborted or interrupted, it is SMF, not Cluster, that is primarily responsible for restarting these services.

Solaris Cluster only intervenes to restore control on the adjacent node under the following circumstances:

- Lost communication with the ACSLS filesystem.
- Lost communication with all redundant public Ethernet ports.
- Lost and unrecoverable communication with a specified library.

Suspending Cluster Control

If it is suspected that the maintenance activity might trigger an unwanted cluster failover event, suspend cluster control of the `acsls` resource group.

To suspend Cluster control:

```
# clrg suspend acsls-rg
```

While the resource group is suspended, Solaris Cluster makes no attempt to switch control to the adjacent node, no matter what conditions might otherwise trigger such action.

This suspension enables you to make more invasive repairs to the system, even while library production may be in full operation.

If the active node happens to reboot while in suspended mode, it does not mount the `acslspool` after the reboot, and ACSLS operation is halted. To clear this condition, resume Cluster control.

To resume Cluster control:

```
# clrg resume acsls-rg
```

If the shared disk resource is mounted to the current node, then normal operation resumes. But if Solaris Cluster discovers upon activation that the `zpool` is not mounted, it immediately switches control to the adjacent node. If the adjacent node is not accessible, control switches back to the current node. Cluster attempts to mount the `acslspool` and start ACSLS services on this node.

Powering Down the ACSLS HA Cluster

The following procedure provides for a safe power down sequence if it is necessary to power down the ACSLS HA System.

1. Determine the active node in the cluster.

   ```
   # clrg status
   ```

   Look for the online node.

2. Log in as `root` to the active node and halt Solaris Cluster control of the ACSLS resource group.

   ```
   # clrg suspend acsls-rg
   ```

3. Switch to user acsss and shutdown the acsss services:

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

4. Log out as acsss and gracefully power down the node.

```
$ exit
# init 5
```

5. Log in to the alternate node and power it down with init 5.

6. Power down the shared disk array using the physical power switch.

Powering Up a Suspended ACSLS Cluster System

To restore ACSLS operation on the node that was active before a controlled shutdown:

1. Power on both nodes locally using the physical power switch or remotely using the Sun Integrated Lights Out Manager.

2. Power on the shared disk array

3. Log in to either node as root.

4. If you attempt to login as acsss or to list the $ACS_HOME directory, you find that the shared disk resource is not mounted to either node. To resume cluster monitoring, run the following command:

```
# clrg resume acsls-rg
```

   With this action, Solaris Cluster mounts the shared disk to the node that was active when the system was brought down. This action should also automatically reboot the acsss services and resume normal operations.

Creating a Single Node Cluster

There may be occasions where ACSLS must continue operation from a standalone server environment on one node while the other node is being serviced. This would apply in situations of hardware maintenance, an operating system upgrade, or an upgrade to Solaris Cluster.

Use the following procedures to create a standalone ACSLS server.

1. Reboot the desired node in a non-cluster mode.

```
# reboot -- -x
```

   To boot into non-cluster mode from the Open Boot Prom (OBP) on SPARC servers:

```
ok: boot -x
```

   On X86 Servers, it is necessary to edit the GRUB boot menu.

   a. Power on the system.

   b. When the GRUB boot menu appears, press **e** (edit).

   c. From the submenu, using the arrow keys, select **kernel /platform/i86pc/multiboot**. When this is selected, press **e**.

   d. In the edit mode, add -x to the multiboot option kernel /platform/i86pc/multiboot -x and click **return**.

   e. With the multiboot -x option selected, press **b** to boot with that option.

2. Once the boot cycle is complete, log in as root and import the ACSLS Z-pool.

   ```
   # zpool import acslspool
   ```

   Use the -f (force) option if necessary when the disk resource remains tied to another node.

   ```
   # zpool import -f acslspool
   ```

3. Bring up the acsss services.

   ```
   # su - acsss
   $ acsss enable
   ```

10

Installing, Upgrading, and Removing Software Components

This chapter describes procedures to follow when upgrading or removing the various components associated with ACSLS HA.

Installing Patches for ACSLS

To install patches for the STKacsls package:

1. Suspend cluster control.

   ```
   # clrg suspend acsls-rg
   ```

2. Download the patch to the `/opt` directory and unzip the package.
3. Go into the `/opt/ACSLS_8.x.x` directory and follow the instructions in the patch `README.txt` file.
4. Disable ACSLS operation and switch control to the adjacent node.

   ```
   # su - acsss
   $ acsss shutdown
   $ exit
   # cd /opt/ACSLSHA/util
   # acsAgt nodeSwitch
   ```

5. Install the ACSLS patch on the new node.
6. Apply ACSLS HA changes to the updated ACSLS structure.

   ```
   # cd /opt/ACSLSHA/util
   # ./copyUtils.sh
   ```

7. Enable the ACSLS services

   ```
   # su - acsss
   # acsss enable
   ```

8. Resume cluster control of the `acsls-rg` resource group.

   ```
   # clrg resume acsls-rg
   ```

Removing the ACSLS Package

Removal of the ACSLS package may be necessary in cases of an ACSLS upgrade. To do so, it is necessary to disable cluster control, halt ACSLS services on both nodes, then remove the package on each node. Use the following procedure:

1. Suspend Cluster control.

   ```
   node1:# clrg suspend acsls-rg
   ```

2. On the active node, shutdown ACSLS.

   ```
   node1:# su - acsss
   node1:$ acsss shutdown
   node1:$ exit
   node1:#
   ```

3. Export the file system on the shared disk array.

   ```
   node1:# cd /
   node1:# zpool export acslspool
   ```

   This operation fails when logged in as user `acsss`.

4. Log in to the alternate node and import the shared disk array.

   ```
   node1:# ssh <alternate node>
   node2:# zpool import acslspool
   ```

5. Shutdown ACSLS

   ```
   node2:# su - acsss
   node2:$ acsss shutdown
   node2:$ exit
   node2:#
   ```

6. Remove the `STKacsls` package.

   ```
   node2:# pkgrm STKacsls
   ```

7. Return to the original node and remove the `STKacsls` package.

   ```
   node2:# exit
   node1:# pkgrm STKacsls
   ```

Installing Upgrade Releases of ACSLS

It is necessary to remove the `STKacsls` package on both nodes before installing a new release of ACSLS. Refer to the procedure detailed in the section above. To install a new package, follow this procedure:

1. Download the STKacsls package to the `/opt` directory and unzip the package. Repeat this step on the alternate node.

2. With Solaris Cluster suspended, ensure that the shared disk array (`acslspool`) is mounted to the current node.

   ```
   node1:# zpool list
   ```

   If the `acslspool` is not mounted, login to the alternate node. If it is not mounted to either node, import the `acslspool`.

3. Go into the `/opt/ACSLS_8.x.x` directory and follow the instructions in the `README.txt` file.

4. Export the `acslspool`.

   ```
   node1:# zpool export acslspool
   ```

   This operation fails if logged in as user `acsss`.

5. Log in to the alternate node and repeat steps 1 through 3.
6. Go to the `/opt/ACSLSHA/util` directory and run `copyUtils.sh`.

   ```
   node2:# cd /opt/ACSLSHA/util
   node2:# ./copyUtils.sh
   ```

7. Start up ACSLS library control.

   ```
   node2:# su - acsss
   node2:$ acsss enable
   node2:$ exit
   node2:#
   ```

8. Resume cluster control of the `acsls` resource group.

   ```
   node2:# clrg resume acsls-rg
   ```

Reinstalling ACSLS HA or Installing Upgrades

Reinstalling the `SUNWscacsls` package requires the halting of ACSLS library operation and taking the `acsls-rg` resource group offline.

1. Shutdown ACSLS.

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

2. Save any customized changes to `ha_list.txt` and `pingpong_interval`.

   ```
   # cd $ACS_HOME/acslsha
   # cp ha_list.txt ha_list.save
   # cp pingpong_interval pingpong_interval.save
   ```

3. Bring the `acsls-rg` resource group offline.

   ```
   # clrg offline acsls-rg
   ```

   This action unmounts the `acslspool` from the ZFS file system

4. Unconfigure the `acsls-rg` resource group.

   ```
   # cd /opt/ACSLSHA/util
   # ./acsAgt unconfigure
   ```

5. Download and unzip the updated `SUNWscacsls` package to both nodes.
6. Remount the acslspool to either of the two nodes.

   ```
   # zpool import -f acslspool
   ```

7. Start ACSLS to verify proper functionality.

   ```
   # su - acsss
   $ acsss enable
   $ exit
   #
   ```

8. Run the `copyUtils.sh` utility.

   ```
   # cd /opt/ACSLSHA/uti
   # ./copyUtils.sh
   ```

9. Restore any customized files you saved in step 2.

   ```
   # cd $ACS_HOME/acslsha
   # cp ha_list.save ha_list.txt
   # cp pingpong_interval.save pingpong_interval
   ```

10. Configure the `acsls-rg` resource group.

    ```
    # cd /opt/ACSLSHA/util
    # ./acsAgt configure
    ```

    This action stops and restarts ACSLS

Within minutes, the command `clrg status` should reveal that the `acsls-rg` resource group is back online and ACSLS is operational.

Upgrading Solaris Cluster

Before removing or upgrading Solaris Cluster, bring down ACSLS and unconfigure the `acsls-rg` resource group.

```
# su - acsss
$ acsss shutdown
$ exit
# cd /opt/ACSLSHA/util
# ./acsAgt unconfigure
```

Consult the current Solaris Cluster documentation for specific upgrade procedures. The general command to upgrade Solaris Cluster is:

```
# scinstall -u
```

Once your Solaris Cluster installation has been updated, configure the `acsls-rg` resource group:

```
# cd /opt/ACSLSHA/util
# ./acsAgt configure
```

Removing Solaris Cluster

1. Bring down ACSLS and the `acsls-rg` resource group.

   ```
   # su - acsss
   $ acsss shutdown
   $ exit
   # cd /opt/ACSLSHA/util
   # ./acsAgt unconfigure
   ```

2. Reboot both nodes into a non-cluster mode.

   ```
   # reboot -- -x
   ```

3. When both nodes are up, log in from either node and remove the Solaris Cluster package.

   ```
   # scinstall -r
   ```

11

Cluster Logging, Diagnostics, and Testing

This chapter describes the various resources available for testing ACSLS HA installation, and for diagnosing issues and troubleshooting problems that may emerge on the system.

Monitoring Overall Cluster Operation

The activities that transpire during a startup or switch over event are widely distributed across the two nodes. Consequently, the vantage point that chosen to observe the overall operation during testing can greatly determine the ability to see the unfolding events as they transpire. "The ha_console.sh Utility" on page 7-3 describes procedures for setting up a comprehensive view

A recommended dashboard configuration for observing the overall HA behavior during testing would include eight shell windows, four windows from each node.

1. A command shell for `root` should be reserved on each node to assert various commands as needed
2. Set up a window on each node to display the tail of the system `/var/adm/messages` file.

   ```
   # tail -f /var/adm/messages
   ```

   Solaris Cluster prints all informational messages to this log file.
3. Set up another window on each node to display the tail of the `acsls-rs` resource `start_stop log`.

   ```
   # tail -f /var/cluster/logs/DS/acsls-rg/acsls-rs/start_stop_log.txt
   ```

   All messages posted by the `acsls_agt.sh` start script are displayed here.
4. A third window on each node should be set up to display the tail of the `acsls-rs` probe log.

   ```
   # tail -f /var/cluster/logs/DS/acsls-rg/acsls-rs/probe_log.txt
   ```

   Once the application has started, Solaris Cluster probes the ACSLS resource once every minute. A numeric code is returned to Cluster from each probe, and the results are printed to the file, `probe_log.txt`. With each probe, any of five standard return values are seen posted to this log:

   ```
     0 -  The probe found that ACSLS is healthy and functioning normally.
     1 -  The probe may not have completed due to a functional error.
     2 -  The probe reports that ACSLS is in a transitional state.
     3 -  The ACSLS application has been intentionally placed offline.
   201 -  A condition was detected that requires fail-over action.
   ```

It is only in response to code 201 that Solaris Cluster initiates failover action. The conditions that prompt such action are listed in the chapter, "ACSLS Cluster Operation" on page 9-1. All other return codes from the Cluster probe are considered informational and no Cluster responsive actions are asserted.

Sample probes for testing can be asserted at any time from the command line. Use the command, acsAgt probe:

```
#/opt/ACSLSHA/util/acsAgt probe
```

All of the logs mentioned above reflect a system view as seen by Solaris Cluster. Two additional logs in the $ACS_HOME/log/ directory provide a view from the ACSLS application level. The acsss_event.log reports all significant events encountered by ACSLS from the moment the application was started. And any ACSLS startup difficulties encountered by SMF are logged in acsls_start.log.

Cluster Monitoring Utilities

Solaris Cluster utilities are found in the /usr/cluster/bin directory.

- To view the current state of the ACSLS resource group: `clrg list -v`
- To view the current status of the two cluster nodes: `clrg status`
- To view the status of the resource groups: `clrs status`
- To get verbose status on the nodes, the quorum devices, and cluster resources: `cluster status`
- For a detailed component list in the cluster configuration: `cluster show`
- To view the status of each Ethernet node in the resource group: `clnode status -m`
- To view the status of the various acsls-rg resources on each note: `scstat -g`
- To view the health of the heartbeat network links: `clintr status`
- To view IPMP status: `scstat -i`
- To view node status: `scstat -n`
- To view quorum configuration and status: `scstat -q` or `clq status`
- To show detailed cluster resources, including timeout values: `clresource show -v`

Recovery and Failover Testing

This section discusses the conditions, monitoring, and testing for recovery and failover testing.

Recovery Conditions

There are numerous fatal system conditions that can be recovered without the need of a system fail over event. For example, with IPMP, one Ethernet connection in each group may fail for whatever reason, but communication should resume uninterrupted through the alternate path.

The shared disk array should be connected to the servers with two distinct ports on each server. If one path is interrupted, disk I/O operation should resume without interruption over the alternate path.

ACSLS consists of several software services that are monitored by the Solaris Service Management Facility (SMF). As user acsss, list each of the acsss services with the command acsss status. Among these services are the PostgreSQL database, the WebLogic Web application server, and the ACSLS application software. If any given service fails on a Solaris system, SMF should automatically reboot that service without the need for a system failover.

The acsls service itself consists of numerous child processes that are monitored by the parent, acsss_daemon. To list the ACSLS sub-processes, use the command, psacs (as user acsss). If any of the child processes is aborted for any reason, the parent should immediately reboot that child and recover normal operation.

Recovery Monitoring

The best location to view recovery of system resources (such as disk I/O and Ethernet connections), is the system log, /var/adm/messages.

SMF maintains a specific log for each software service that it monitors. This log displays startup, restart, and shutdown events. To get the full path to the service log, run the command, svcs -l *service-name* ACSLS services can be listed using the acsss command: $ acsss status. Subprocesses can be listed with the command, $ acsss p-status.

To view recovery of any ACSLS sub-process, you can monitor the acsss_event.log ($ACS_HOME/ACSSS/log/acsss_event.log). This log displays all recovery events involving any of the ACSLS sub-processes.

Recovery Tests

Redundant network connections should be restarted automatically by the Solaris IPMP logic. Any interrupted data connection to the shared disk array should be restarted automatically by Solaris on the redundant data path. Services under control of Solaris Service Management Facility should be restarted automatically by SMF.

For tests that involve an actual failover event, be aware of the property setting defined in the file: $ACS_HOME/acslsha/pingpong_interval. Despite the conditions which may trigger a failover event, Solaris Cluster does not initiate failover action if a prior failover event occurred within the specified pingpong_interval.

To view or to dynamically change the pingpong interval, go to the /opt/ACSLSHA/util directory and run acsAgt pingpong:

```
# ./acsAgt pingpong
Pingpong_interval
   current value:  1200 seconds.
   desired value: [1200] 300
Pingpong_interval : 300 seconds.
```

Use any or all of the following techniques to evaluate the resilience of HA installation:

1. While ACSLS is operational, disconnect one Ethernet connection from each IPMP group on the active node. Monitor the status using # scstat -i.

   Observe the reaction in /var/adm/messages. ACSLS operation should not be interrupted by this procedure.

2. Ensure that Cluster Failover_mode is set to **HARD**. While ACSLS is operational, disconnect one fibre or SAS connection from the active server to the shared disk resource.

Observe the reaction in /var/adm/messages. ACSLS operation should not be interrupted by this procedure.

Repeat this test with each of the redundant I/O connections.

3. Abruptly terminate ACSLS by killing the acsss_daemon. Use pkill acsss_daemon.

   Run svcs -l acsls to locate the service log.

   View the tail of this log as the acsss_daemon is stopped.Observe that the service is restarted automatically by SMF. Similar action should be seen if stopping acsls with acsls shutdown.

4. Using SMF, disable the acsls service.

   This can be done as root with svcadm disable acsls or it can be done as user acsss with acsss disable.

   Because SMF is in charge of this shutdown event, there is no attempt to restart the acsls service. This is the desired behavior. The acsls service must be restarted under SMF. As root, use the command, svcadm enable acsls. Or, as user acsss, use the command, acsss-enable.

5. Bring down the acsdb service.

   As user acsdb, source the .acsls_env file.

   ```
   $ su acsdb
   $ . /var/tmp/acsls/.acsls_env
   ```

   Now, abruptly disable the PostgreSQL database with the following command:

   ```
   pg_ctl stop \
        -D $installDir/acsdb/ACSDB1.0/data \
        -m immediate
   ```

   This action should bring down the database and also cause the acsls processes to come down. Run svcs -l acsdb to locate the acsdb service log.

   View the tail of both the acsdb service log and the acsls service log the database is brought down. Observe that when the acsdb service goes down, it also brings down the acsls service. Both services should be restarted automatically by SMF.

6. While ACSLS is operational, run psacs as user acsss to get a list of subprocesses running under the acsss_daemon.

   Stop any one of these subprocesses. Observe the acsss_event.log to confirm that the subprocess is restarted and a recovery procedure is invoked.

Failover Conditions

Solaris Cluster Software monitors the Solaris system, looking for fatal conditions that would necessitate a system failover event. Among these would be a user initiated fail over (acsAgt nodeSwitch or clrg switch -n), a system reboot of the active node, or any system hang, fatal memory fault, or unrecoverable I/O communications on the active node. Solaris Cluster also monitors HA agents that are designed for specific applications. The ACSLS HA Agent requests a system failover event under any of the following conditions:

- TCP/IP communication is lost between the active node and the logical host.
- The $ACS_HOME file system is not mounted.
- The database backup file system ($ACS_HOME/.../backup) is not mounted.

- Communication is lost to the library corresponding to a specified ACS in the file `$ACS_HOME/acslsha/ha_acs_list.txt` whose desired state is online and where a `switch lmu` is not otherwise possible or successful.

Failover Monitoring

From moment to moment, the failover status of the respective nodes can be monitored using the command `# clrg status`.

Fail over activity can also be monitored by observing the tail of the `start_stop_log`:

```
# tail -f /var/cluster/logs/DS/acsls-rg/acsls-rs/start_stop_log.txt
```

It can be useful to view (`tail -f`) the `/var/adm/messages` file on both nodes as you perform diagnostic failover operations. See "Monitoring ACSLS Cluster Operation" on page 7-2.

Failover Tests

1. The simple command to initiate a Cluster failover event is `acsAgt nodeSwitch`.

   ```
   # acsAgt nodeSwitch
   ```

   Or, use the equivalent Cluster command:

   ```
   # clrg switch -n <node name> acsls_rg
   ```

   This action should bring down the ACSLS application and switch operation from the active server to the standby system. The options `-M -e` instructs the cluster server to enable SMF services on the new node. See "Monitoring ACSLS Cluster Operation" on page 7-2.

2. A system reboot on the active node should initiate an immediate HA switch to the alternate node.

   This operation should conclude with ACSLS running on the new active node. On the standby node, watch the tail of the `/var/adm/messages` file as the standby system assumes its new role as the active node. The command, `# clrg status`, can also be periodically run.

3. Using `init 5`, power down the active server node and verify system failover.

4. Unplug both data lines between the active server node and the shared disk Storage Array and verify a system switch to the standby node.

5. Assuming that a given library is listed in the policy file, `ha_acs_list.txt`, disconnect both Ethernet communication lines between the active server node and that library.

   Verify system failover to the standby node.

Additional Tests

If the mirrored boot drives are hot-pluggable, disable one of the boot drives and confirm that the system remains fully operational. With one boot drive disabled, reboot the system to verify that the node comes up from the alternate boot drive. Repeat this action for each of the boot drives on each of the two nodes.

Remove any single power supply from the active node and the system should remain fully operational with the alternate power supply.

12

Troubleshooting Tips

ACSLS HA 8.4 is the integration of the ACSLS application operating on a two-node system under Solaris 11.2 with IPMP and ZFS under the control of Solaris Cluster 4.2.

Verifying that ACSLS is Running

To verify that ACSLS services run on the active node, use the following command as user `acsss`:

```
# su - acsss
$ acsss status
```

If one or more services are disabled, enable them with `$ acsss enable`.

If the status display reveals that one or more of the ACSLS services is in maintenance mode, run the command `$ acsss l-status`.

Look for the path to the log file of the faulty service and view that log for hints that might explain why the service was placed in maintenance mode.

If one or more of the acsls services is in maintenance mode, they can be cleared by disabling then enabling them with the `acsss` command.

```
$ acsss shutdown
$ acsss enable
```

As `root`, use `# svcadm clear <service name>` to clear an individual service.

The service is not cleared until the underlying fault has been corrected.

Specific operational logs should also be reviewed as a means to reveal the source of a problem. Most of these are found in the `$ACS_HOME/log` directory.

The primary log to review is the `acsss_event.log`. This log records most events surrounding the overall operation of ACSLS.

If the problem has to do with the ACSLS GUI or with logical library operation, the relevant logs are found in the `$ACS_HOME/log/sslm` directory.

For the ACSLS GUI and WebLogic, look for the `AcslsDomain.log`, the `AdminServer.log`, and the `gui_trace.logs`.

Installation problems surrounding WebLogic are found in the `weblogic.log`.

For logical library issues, once a logical library has been configured, consult the `slim_event.logs`, and the `smce_stderr.log`.

Addressing Connection to the Shared Disk Resource

1. Verify that the `acsls-storage` resource is online to the active cluster node.

   ```
   # clrs status acsls-storage
   ```

2. If the `acsls-storage` resource is not online, verify if the resource is mounted to ZFS on the active node:

   ```
   # zpool status
   ```

   If the `acslspool` is not mounted on the active node, verify if it is mounted on the standby node

   ```
   # ssh standby hostname zpool status
   ```

   If the shared disk resource is mounted on the standby node, then switch cluster control to that node.

   ```
   # clrg switch -n standby hostname acsls-rg
   ```

3. If the `acslspool` is not mounted on the active node, and the `acsls-storage` resource is offline, verify that the `acslspool` is visible to the active node.

   ```
   # zpool import (no argument)
   ```

   > **Note:** This operation works only if `acsls-storage` is offline. To bring it offline, use the command `clrs disable acsls-storage`.

   If the `acslspool` is visible to the active node, attempt to import it:

   ```
   # zpool import -f acslspool
   ```

   If the `import` operation succeeds, bring the `acsls-storage` resource online to Solaris Cluster:

   ```
   # clrs enable acsls-storage
   ```

   If the `acslspool` is not visible to the active node, it is be necessary to troubleshoot the physical connection to the shared drive.

When the Logical Host Cannot be pinged

1. Verify that the logical hostname is registered with Solaris Cluster.

   ```
   # clrslh list
   ```

2. Determine the active node:

   ```
   # clrg status | grep -i Online
   ```

3. Verify the active node can be pinged.

   ```
   # ping <node name>
   ```

4. Verify that the `logical-host` name resource is online to the active node.

   ```
   # clrslh status
   ```

   If the logical host is not online, then enable it.

```
# clrs enable <logical host>
```

5. Verify the state of the IP interfaces assigned to the public group.

   ```
   # ipadm
   ```

   In the output display, verify the *ok* state of each member of the public ipmp group.

6. For each interface in the public group (ipmp0), verify its physical state.

   ```
   # dladm show-phys
   ```

7. Verify that the logical host is plumbed to one or the other of the two interfaces in the public ipmp group (revealed in step-5)

   ```
   # arp <logical-hostname>
   # ifconfig net0
   # ifconfig net4
   ```

   This example assumes that net0 and net4 were assigned to the public ipmp group.

   The MAC address of one of the two interfaces should agree with the MAC address assigned to the logical hostname.

Checking the Interconnection Between Nodes

If it is suspected that cluster control fails because of lost communication for Cluster between the two nodes, check the private interconnection for Cluster as follows:

```
# cluster status -t interconnect
```

Index