StorageTek Automated Cartridge System Library Software

Installation Guide

Release 8.5
E96378-11
April 2022

StorageTek Automated Cartridge System Library Software Installation Guide, Release 8.5

E96378-11

Contents

Preface

1 Overview

2 Installing ACSLS on Solaris

3 Installing ACSLS on Linux

C Using the ACSLS Feature Card Availability Toolkit

D Configuring a Self-Signed Digital Certificate for HTTPS

E Maintenance and Troubleshooting Tasks for ACSLS on the Feature Card

F Servicing the Feature Card and Recovering ACSLS

Index

List of Figures

List of Tables

Preface

Automated Cartridge System Library Software (ACSLS) is Oracle's StorageTek server software that controls StorageTek automated tape libraries. The StorageTek ACS family of products consists of fully automated, tape cartridge-based data storage and retrieval systems. StorageTek ACSLS supports network access to different client systems that can range from workstations to mainframes to supercomputers running on a variety of operating systems.

This guide is for the individual responsible for administering StorageTek ACSLS. It is expected that you already have a working knowledge of the following:

- UNIX file and directory structure
- How to use UNIX commands and utilities for your platform
- UNIX system files
- How to do typical UNIX system administrator tasks, such as logging on as root and setting up user access to a UNIX application

Documentation Accessibility

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

Access to Oracle Support

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

1 Overview

Automated Cartridge System Library Software (ACSLS) is Oracle's StorageTek server software that controls StorageTek automated tape libraries. An Automated Cartridge System (ACS) is a group of tape libraries connected through pass-thru-ports (PTPs). ACSLS accesses and manages information stored in one or more ACSs through command processing across a network. The software includes a system administration component and interfaces to client system applications, and library management facilities. ACSLS 8.5 is bundled with WebLogic 10.3.6.

ACSLS 8.5 uses the relational database PostgreSQL. On Solaris 11, the PostgreSQL packages are available from the Oracle Software Delivery Cloud in the same location where you find the STKacsls package. Linux installation procedures described in this publication include the process of adding PostgreSQL packages from the Oracle `yum` repository after installing the Linux Product Pack.

Legal Notice

In addition to the Oracle Right to Use License for ACSLS, this product contains numerous third-party software components, each with its own license criteria. Read the `THIRDPARTYLICENSEREADME.txt` agreement located in the ACSLS 8.5 installation directory. For software components whose license requires re-distribution of the source code, you can find that source code under the initial package installation directory, `ACSLS_8.5.0` or `ACSLS_8.5.1` (typically under `/opt`). Look in the subdirectory, `acsls_thirdPartySoftware/`.

Software Requirements

ACSLS 8.5 has been developed and tested for the following operating system environments:

- ACSLS Release 8.5.1
  - Oracle's Sun SPARC and X86 platforms running Solaris 11, Update 3, or Solaris 11, Update 4:
    - For Solaris 11, Update 3, Support Repository Update (SRU) 35 or later is required.
    - For Solaris 11, Update 4, Support Repository Update (SRU) 8 or later is required.

    Oracle recommends using the latest SRU available. Visit the My Oracle Support page at http://support.oracle.com for more information.
  - Oracle Enterprise Linux releases 6.8, 6.10, 7.3, 7.6, 7.8 and 7.9
- ACSLS Release 8.5.0
  - Oracle's Sun SPARC and X86 platforms running Solaris 11, Update 3.

    Support Repository Update (SRU) 35 or later is required.

    Oracle recommends using the latest SRU available. Visit the My Oracle Support page at http://support.oracle.com for more information.

  - Oracle Enterprise Linux releases 6.8 and 7.3

Oracle Linux testing was performed in environments using Oracle's Unbreakable Enterprise Kernel. Other operating systems, including virtual environments, are not tested or supported.

> **Note:**
>
> Special device drivers are provided in ACSLS for use with logical libraries and with fibre-attached libraries, such as the SL500 and SL150. This is an issue for Solaris zoned environments. Because these device drivers are attached to the system kernel, they must reside in the global zone. In cases where these drivers are used, ACSLS cannot be installed in the local zoned environment. Logical libraries are not supported on the Linux operating system.

System Requirements

- Memory: 16GB minimum

  To show system memory:

  - Solaris:

    ```
    prtconf | grep Mem
    ```

  - Linux:

    ```
    grep MemTotal /proc/meminfo
    ```

- Swap space:

  Solaris and Linux systems should be equipped with a minimum of 16GB of memory and a minimum of 4GB of swap space. Server configuration and operating system can affect swap space requirements. Generally, no less than 30% of the physical memory is recommended.

  To check swap space, enter one of the following operating system commands:

  - Solaris:

    ```
    vmstat -S
    ```

    The result is expressed in kilobytes.

  - Linux:

    ```
    vmstat -s | grep total
    ```

    The result is expressed in kilobytes.

- File systems and required databases:

  ACSLS 8.5 enables you to install in any file system. You must define the following directories before installing ACSLS.

  - A base directory where the ACSLS components will be installed.
  - A default directory for ACSLS backups. It is recommended (but not required) to place the ACSLS backup directory in a separate file system from the ACSLS base directory.

Although you can install ACSLS in any directory, the default directories used for ACSLS are:

  - `/export/home` is the default ACSLS base directory.
  - `/export/backup` is the default ACSLS backup directory.

  The ACSLS base directory file system requires a minimum of 5GB free. Reserve an additional 5GB free for ACSLS backups. To view file system sizes, enter the following command:

  ```
  df -h
  ```

- The ACSLS creates and uses the `/var/tmp/acsls` directory for keeping required work files during execution. If you delete or move the contents of this directory while ACSLS is executing, the ACSLS will stop operating and require a restart.
- Fibre card is optional. A suitable HBA is required for Fibre Channel operations.
  - For target mode operation, supporting the Logical Library feature, this HBA must be a contemporary QLogic fibre card (4Gb or higher).
  - For initiator mode operation, supporting a fibre-connected library such as the SL500 or SL150, ACSLS 8.5 is fully tested and certified with QLogic and Emulex HBAs.

ACSLS GUI Requirements

If you plan to use the ACSLS GUI, ensure that you install the latest version of the Java Development Kit (JDK) on your ACSLS server before you install ACSLS. Refer to the *ACSLS Product Information* document for specific required Java versions.

The ACSLS 8.5 GUI can operate with most common browsers though formal testing has been limited to recent releases of FireFox, Chrome, and Internet Explorer. The Chrome browser and earlier versions of FireFox have tested well using the default settings for ACSLS in the WebLogic server. Internet Explorer 8 (and above) and FireFox 39 (and above) require configuration settings to provide a 2048-bit self-signed digital certificate for https. See "Configuring a Self-Signed Digital Certificate for HTTPS".

Co-Hosting

To ensure uninterrupted library service and to avoid unanticipated problems due to resource contention, it is generally recommended that ACSLS run in a standalone environment on a dedicated server. However, some systems are designed to allow multiple applications to run in co-hosted fashion as though they are completely isolated from one another. Solaris Containers and Oracle Solaris VM Server for SPARC enable conditional co-hosting possibilities for use with ACSLS.

The following list details the conditions and limitations associated with the various co-hosting options for an ACSLS application.

- Solaris Zones (containers)

  Solaris zones enable a system administrator to partition a standard, low cost server into four independent Solaris systems, each with its own isolated file system, and its own instance of Solaris. You can assign network resources to each zone and you can reboot any local (non-global) zone without affecting applications in other zones on the same platform.

However, the ability to share kernel resources, such as device drivers, across multiple zones is tenuous at best. Ideally, an application that requires kernel drivers would reside in the global zone. However, it is generally not good practice to install an application in the global zone since any fatal condition with the application could impact all other applications running in the other zones.

ACSLS 8.5 can reside in a Solaris zone only if it does not require drivers beyond the network interface. Any use of Logical Libraries requires a target-mode fibre-channel driver, and any connection to an SL500 or SL150 library requires an initiator-mode fibre-channel driver. Either of these configurations dictates that ACSLS must be installed in the global zone.

There is no version of ACSLS HA supported for use in Solaris zones.

- Oracle VM Server for SPARC

  Oracle VM Server for SPARC (formerly Logical Domains or LDOMs) technology offers significant advantages over Solaris Containers to the extent that each domain is in control of its own Solaris kernel.

  A Solaris administrator can partition hardware resources across the system, assigning a specific resource to a specific domain. Network resources on this virtual machine can easily be shared across any of up to 128 *guest domains* on the server. But applications that require access to I/O devices through the PCIe bus must be installed in special I/O domains. The number of I/O domains that you can create on the VM Server depends on the number of discrete PCIe buses on the SPARC platform. On a system with a single PCIe bus, you can have two I/O domains, and one of these must be the control domain.

  Any ACSLS application that relies solely on network connectivity to the library and for client applications can be installed in a guest domain on this server. The virtual network set-up procedure is described in the document, *Oracle VM Server for SPARC 2.1 Administration Guide* in the section, entitled "Using Virtual Networks".

  If your ACSLS 8.5 application is intended for use with logical libraries, or if you intend to connect to a fibre-channel library such as the SL500 or L700, then ACSLS must be installed in an I/O domain. Refer to the section "Setting up I/O Domains" in the *Oracle VM Server for SPARC 2.1 Administration Guide*.

2

Installing ACSLS on Solaris

This chapter describes how to install ACSLS Release 8.5 in a Solaris environment.

Topics include:

- Preparing for Installation
- Installing ACSLS
- Performing Post Installation Tasks

Preparing for Installation

Perform the following tasks to prepare for ACSLS installation. Once you have completed these tasks, you are ready to install ACSLS 8.5.

- Step 1: Export Existing Database and Control Files
- Step 2: Remove Previous ACSLS Versions
- Step 3: Ensure Solaris is Installed
- Step 4: Network Security Settings
- Step 5: Cron Administration
- Step 6: ACSLS Access Privileges
- Step 7: Download and Unzip the ACSLS 8.5 Package
- Step 8: Create User Accounts and Groups

Step 1: Export Existing Database and Control Files

If you are upgrading from a previous release and plan to use existing database and control files, you must export these files.

1. As user acsss, enter the following command:

   ```
   db_export.sh -f /path/myExport
   ```

   where *myExport* is the name of your export file.
2. Save both *myExport* and *myExport*.misc files to a non-volatile location.
3. If you are updating your operating system, then transfer these files to a remote machine for safe keeping.

For more information, refer to the “Database Administration" chapter in the *StorageTek ACSLS Administrator's Guide*.

Step 2: Remove Previous ACSLS Versions

Remove any previous version of ACSLS. If this is a new installation with no previous version of ACSLS, then skip this step.

1. Ensure that you have exported the database, using the db_export.sh utility command.
2. Log in as user acsss.
3. Shut down all ACSLS services:

   ```
   acsss shutdown
   ```

4. As root, go to the Package installation directory (typically /opt/ACSLS_*x.y.z*)

   To remove the package, follow the un-install instructions for the your specific installed release. For example, to remove the ACSLS release 8.4 package, execute the pkg_uninstall.sh script:

   ```
   # ./pkg_uninstall.sh
   ```

   The ACSLS user accounts still remain.
5. Remove ACSLS administrative accounts:

   ```
   # userdel acsss
   # userdel acsdb
   # userdel acssa
   # userdel postgres

   # groupdel acsls
   # groupdel postgres
   ```

6. Reboot.

Step 4: Network Security Settings

Your Solaris installation should "Enable remote services" to ensure that network client applications are able to communicate with the ACSLS server.

If you select the Solaris "Secure by Default" installation option, then it is necessary to alter a network configuration property for rpc-bind. To do this:

1. Check the property setting:

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

2. If the local_only property setting is *true*, you must set it to *false*.

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

Step 5: Cron Administration

Specific automated schedules known as *crontabs* are created for users acsss and acsdb when you run the install.sh utility. These crontabs are provided for ACSLS database maintenance backup activities.

An optional file, /etc/cron.allow (or /etc/cron.d/cron.allow on some systems) may exist on the system. This file controls which users are allowed to run the crontab command. If cron.allow exists, then user IDs for acsss and acsdb must be included in that file before you run install.sh. Otherwise, crontab creation for these users fails.

The file cron.deny exists by default on most systems. Any users listed in this file are explicitly denied access to the crontab command. Make sure that users acsss and acsdb are not contained in the cron.deny file.

Step 6: ACSLS Access Privileges

Note the following access privilege considerations:

- ACSLS 8.5 may be installed in any local file system. The ACSLS base directory and backup directories (for example, `/export/home` and `/export/backup`) must be mounted to allow `SETUID` so that user `acsss` can run as `root`. Super user access is required for scripts that start and stop ACSLS services and for scripts that collect diagnostic information for a support call.
- The `acsss umask` is set to `027` during installation.
- Network services, specifically `rpcbind`, must be enabled to allow ACSLS client communication unless the firewall security on ACSLS and all ACSAPI clients is configured without the need for the portmapper. For more information, refer to "Firewall Security" in the *StorageTek ACSLS Administrator's Guide*.

Step 7: Download and Unzip the ACSLS 8.5 Package

To download and unzip the ACSLS 8.5 package:

1. Start a web browser on the system and visit the Oracle Software Delivery Cloud:

   `https://edelivery.oracle.com`
2. Click **Sign In** and enter the user name and password provided by your Oracle Support representative.
3. In the search field, enter `acsls` and select **StorageTek Automated Cartridge System Library Software (ACSLS)**.
4. In the search results, select ACSLS release level 8.5.1.0.0 to add it to the cart.
5. Click **Selected Software** to view the cart.
6. On the Selected Software screen, select your desired platform and click **Continue**.
7. On the Oracle Terms and Restrictions screen, review and accept the terms of the licenses. Click **Continue**.
8. Click **Download** and save the zip file to a common installation directory, typically `/opt`.
9. Before extracting the ZIP file, remove any previously installed versions of ACSLS installation directories. For example:

   ```
   rm -rf /opt/ACSLS_8.4.0
   rm -rf /opt/ACSLS_8.5.0
   rm -rf /opt/ACSLS_8.5.1
   ```
10. Unzip the compressed file. The extracted package set is found in the resulting `ACSLS_8.5.1` subdirectory.

Step 8: Create User Accounts and Groups

Create the user accounts and associated groups described in the table below. For command examples, see Installation Command Examples.

ACSLS allows for a user-defined home directory for the ACSLS application. The parent directory of each user home directory is referenced by the variable, `$installDir`.

Note:

- It is your responsibility to define any required user account attributes such as passwords, based upon your specific configuration and processes.
- ACSLS user accounts (`acsss`, `acsdb`, and `acssa`) must execute `.profile` when logging in. In some instances, `.bash_profile` will override `.profile` for bash shell user accounts.
- If you use directories that cross external NFS or ZFS mount points, ensure that root level privileges extend across the mount points. Without these root level privileges, ACSLS installation may fail, or post-installation functionality issues may occur.

Table 2-1 Required ACSLS User Accounts (Solaris)

User Account	Group Assignment	Home Directory	Command Shell	Description
acsss	acsls	`$(installDir)/ACSSS` Default example: `/export/home/ACSSS` Ownership/Permissions: • Directory Owner: `acsss:acsls` • Minimum permissions: `rwxr-x---`	/bin/bash	ACSLS control user
acssa	acsls	`$(installDir)/ACSSA` Default example: `/export/home/ACSSA` Ownership/Permissions: • Directory Owner: `acssa:acsls` • Minimum permissions: `rwxr-x---`	/bin/bash	ACSLS SA user
acsdb	acsls	`$(installDir)/acsdb/ACSDB1.0` Default example: `/export/home/acsdb/ACSDB1.0` Ownership/Permissions: • Directory Owner: `acsdb:acsls` • Minimum permissions: `rwxr-x---`	/bin/bash	ACSLS DB user
postgres	postgres	`/usr/postgres/10-pgdg` Ownership/Permissions: • Directory Owner: `postgres:postgres` • Minimum. permissions: `rwxr-xr-x`	/bin/bash	postgres user

Table 2-1 (Cont.) Required ACSLS User Accounts (Solaris)

User Account	Group Assignment	Home Directory	Command Shell	Description
root	no requirement	standard root Ownership/Permissions: user defined	/bin/bash	root user

If the user accounts already exist and are locked, you must unlock each account before you install the package.

For example, to check if the acsss account is locked:

```
# passwd -S acsss
acsss LK
```

`LK` indicates that the account is locked. To unlock the account:

```
# passwd -u acsss
```

If these user accounts exist on an LDAP or NIS server and the `root` user on the local machine lacks `usermod` authority on the LDAP or NIS server, then manual intervention by the system administrator is required to complete the ACSLS installation. For example, if the `postgres` user already exists, you must change its home directory to `/usr/postgres/10-pgdg`. The user shell should be `/usr/bin/bash`.

Step 3: Ensure Solaris is Installed

Ensure that a compatible version of Solaris is installed.

- ACSLS Release 8.5.1 is designed to run on Oracle's Sun SPARC and X86 platforms running Solaris 11, Update 3, or Solaris 11, Update 4.
  - For Solaris 11, Update 3, Support Repository Update (SRU) 35 or later is required.
  - For Solaris 11, Update 4, Support Repository Update (SRU) 8 or later is required.
- ACSLS Release 8.5.0 is designed to run on Oracle's Sun SPARC and X86 platforms running Solaris 11, Update 3. Support Repository Update (SRU) 35 is required.

The Oracle Solaris Product Pack can be obtained from the Oracle Software Delivery Cloud:

`https://edelivery.oracle.com`

For installation procedures, refer to the Solaris installation publications.

> **Note:**
>
> Oracle recommends installing the entire Solaris distribution to ensure your installation includes the standard packages required for ACSLS operation.

Installing Missing Oracle Solaris Packages

If your installation is missing a standard Oracle Solaris package required for ACSLS operation, you must acquire and install the missing package.

Oracle Solaris packages are available at http://pkg.oracle.com.

For example, to find and install a missing `unixodbc` package:

1. Visit http://pkg.oracle.com.
2. In the search field, type `unixodbc` and click the Search button. To see more than the latest version of the package, use Advanced Search options.
   In the search results, the complete title of the package indicates the latest Solaris version,11.4:

   ```
   library/unixodbc@2.3.4,5.11-11.4.0.0.1.14.0:20180814T170705Z
   ```

   You can click the package name link to view version details including corresponding OS releases.
3. From your Solaris platform, click the Installation link to install.

Alternative installation tips:

Use the `pkg` command directly from the command line on the platform:

```
pkg install pkg://solaris/library
```

The release is displayed:

```
/unixodbc@2.3.4,5.11-11.4.0.0.1.14.0:20180814T170705Z solaris
```

If that version is disallowed, supply the package name without the version:

```
pkg install pkg://solaris/library/unixodbc
```

For more information, refer to "Adding and Updating Software in Oracle Solaris" in the Oracle Solaris Information Library.

Installing ACSLS

Perform the following tasks to install ACSLS:

1. Ensure that you have completed all pre-installation tasks described in "Preparing for Installation".
2. Log in as user `root`.
3. From the `ACSLS_8.5.0` or `ACSLS_8.5.1` directory, run the `pkg_install.sh` utility:

   ```
   ./pkg_install.sh
   ```

> **Note:**
>
> During installation, the pkgadd utility may generate warning messages regarding existing home directories and associated user shell-related files (for example, /export/home/ACSSS, .profile, and .bashrc).
> If you have previously cleared stale versions or files and set up home directories according to "Step 8: Create User Accounts and Groups", please safely ignore these warnings and proceed with installation.

4. The utility prompts you to enter the full path directory for the installation.

   Enter a desired directory path, or press **Enter** to accept the default path (/export/home). If the directory you specify does not exist, the script prompts for permission create the directory.

   > **Note:**
   >
   > Installation may take significant time based on network and server configuration settings.

5. Enter the following command to inherit the ACSLS environment:

   ```
   . /var/tmp/acsls/.acsls_env
   ```

6. As root, run the ACSLS install.sh utility:

   ```
   cd $ACS_HOME/install
   ./install.sh
   ```

7. The utility asks:

   ```
   Do you wish to host the ACSLS Graphical User Interface? (y/n)
   ```

   The ACSLS GUI is an optional feature. If you are co-hosting ACSLS with another application that uses WebLogic, enter **n** and then proceed with ACSLS installation.

   Otherwise, enter **y** to install the GUI.

   > **Note:**
   >
   > - Ensure that you have installed the latest version of the Java Development Kit (JDK) on your ACSLS server. See ACSLS GUI Requirements.
   > - If ACSLS installation fails during installation of the GUI, review the logs in ACSSS/log/sslm. These logs provide information as to why the GUI failed, in particular the weblogic.log.

   If this is a minor update or configuration change (not a new installation) your ACSLS GUI may already be installed.

   In this case, you have the option to re-install the GUI or to skip this section and retain the current ACSLS GUI domain.

   The utility asks:

```
The Acsls GUI Domain exists. Do you want to re-install it? (y/n)
```

- Enter **y** if you are installing a new ACSLS release.

  The WebLogic server package is extracted and the default GUI `admin` user account is created with the user name, `acsls_admin`.

  You are then asked to assign a password for the `admin` user. The password must be between eight and sixteen characters using both alpha and numeric characters.

  The installation procedure unpacks and deploys the ACSLS GUI application and then creates the `Acsls` user group. At a later time, you can add GUI users to this group using the administrative tool, `userAdmin.sh`.

- If you enter **n**, you have the option to remove the existing GUI configuration.

When you install WebLogic on your ACSLS server, a simple 512-bit public key is automatically available to support basic https exchanges with client browsers. Normally, no further configuration should be necessary. However, some browsers, notably the Microsoft Internet Explorer, require a lengthier key of no less than 1024 bits. See "Configuring a Self-Signed Digital Certificate for HTTPS" for a description of and procedures for configuring an SSL encryption key.

8. The utility asks:

   ```
   Which file system will be used to store database backups? [/export/backup]
   ```

   Enter a desired directory path where you intend for database backup files to reside, or press **Enter** to accept the default path.

   If your desired directory does not exist, you must first create it. The directory must be owned by root with permissions set to 755.

   > **Note:**
   >
   > Ensure that permissions for directory `/opt/oracle` are set to 755 to avoid ACSLS database installation failures related to postgreSQL.

9. The utility asks:

   ```
   Shall we install the mchanger driver for fibre-attached libraries? (y/n)
   ```

   Enter **`y`** if your library environment includes a fibre-attached library such as the SL500 or SL150 library. Otherwise, enter **`n`**.

   If you enter **`y`**, the routine scans the attached SAN environment, looking for any StorageTek library devices. It reports the devices it finds and asks whether any additional libraries are attached. If you have an older SCSI attached L700 or L180 library, respond **`y`** to the prompt.

   For SCSI attached libraries, simply enter the *`target:lun`* address for each library, separating them by a space. For example:

   ```
   ==> 4:0 5:0 5:1
   ```

10. ACSLS can present logical libraries to client applications over a fibre connection. Any portion of an attached physical library can be represented as a (SCSI) fibre-attached library with a fibre target port. To implement this capability, you must have

a QLogic fibre HBA. This step converts one or more QLogic HBA ports from their default initiator mode to target mode.

The utility asks:

```
Do you want to install support for Logical Libraries?(y/n)
```

Enter **y** if you are using logical libraries. Otherwise, enter **n**.

If you enter **y**, the utility asks:

```
The Logical Library features in ACSLS require target mode support.
- required action: pkg install system/storage/scsi-target-mode-framework
Install the target mode package now? (y or n)?
```

Enter **y** to install the target mode packages.

Next, the `install.sh` routine probes the system for qualified HBAs, and then lists the ports it finds.

Select the desired port by the corresponding number. The port you choose must be connected to a remote HBA.

ACSLS can present logical libraries to client applications over a fibre connection. Any portion of an attached physical library can be represented as a (SCSI) fibre-attached library with a fibre target port. To implement this capability, you must have a QLogic fibre HBA. This step converts one or more QLogic HBA ports from their default initiator mode to target mode.

11. If you choose not to install the GUI or logical library support features, then the utility asks:

    ```
    Shall we install the optional lib_cmd interface (y or n):
    ```

    This optional feature is a command-line interface that performs many of the same operations available in the ACSLS GUI. While many `lib_cmd` operations apply to logical library functions, this feature is also useful for displaying the status of physical libraries, volumes and drives.

    The `lib_cmd` feature installs automatically when you choose to install either the GUI or logical library support.

    Enter **y** if you wish to install this feature.

12. Depending on the set of features that you have selected in the above installation dialog, this final step installs Solaris SMF services to control the automatic start, stop, and status functions for each selected ACSLS feature.

    The service list includes any subset of the following:

    ```
    acsdb
    acsls
    smce
    rmi-registry
    surrogate
    stmf
    weblogic
    ```

> **Note:**
>
> - If install.sh encounters errors due to missing packages, you must acquire and install these packages. See Installing Missing Oracle Solaris Packages.
> - During installation, install.sh attempts to regenerate certificates used for secure communications between the user and the ACSLS web-based interface. Certificates are also required when ACSLS manages one or more SL4000 libraries. If an error is encountered during this step, install.sh presents the following prompt:
>
>   ```
>   Problems occurred while generating certificates, which
>   are required for GUI support and for SCI libraries.
>   Continue with installation? (This is NOT recommended) (y or n):
>   n
>   ACSLS installation is incomplete!
>   Please review the /tmp/install.log.
>   ```
>
>   Continuing with installation is only recommended for a non-GUI, legacy installation without SL4000 libraries. In this case, certificates are not required.

13. When the install.sh utility exits, **ACSLS installation is complete**.

Performing Post Installation Tasks

Once ACSLS is installed, you can perform the following post-installation tasks:

- Installing the XAPI Service
- Importing Database and Control Files
- Testing ACSLS Without a Library
- Verifying the ACSLS Installation

Installing the XAPI Service

The optional XML API (XAPI) service is an API that enables Enterprise level mainframe clients and servers to communicate using a common Enterprise Library Software (ELS) protocol over TCP/IP. ACSLS 8.5.0 and later releases can be configured with XAPI support.

To install the XAPI component:

1. Ensure you have installed the ACSLS package and run install.sh to finish the ACSLS installation.
2. Ensure you are logged in to the ACSLS server as root.
3. Source key ACSLS environment variables:

   ```
   . /var/tmp/acsls/.acsls_env
   ```

   (Note the required period and space before /var/tmp/acsls/.acsls_env).

4. Install the XAPI component:

   ```
   cd $ACS_HOME/install
   ./install_xapi.sh
   Installing the XAPI component for Oracle IBM mainframe clients. Continue? (y)
   ```

Importing Database and Control Files

Database and control files are customized files, user preferences, and local configuration files that are unique to your specific ACSLS environment.

If you exported existing database and control files before installing ACSLS 8.5, as described in "Step 1: Export Existing Database and Control Files", you can use the `db_import.sh` utility to import them once ACSLS 8.5 is installed.

Refer to the "Database Administration" chapter in the *StorageTek ACSLS Administrator's Guide* for this procedure.

Testing ACSLS Without a Library

After installing a new ACSLS release, you want to test it before using it to manage production libraries. If a test library environment is not available, this can be difficult because normally ACSLS must be configured to a library, and the library must be online for ACSLS to come up.

If you do not have a configured library or library partition available in a test environment, you can test a new ACSLS release in a limited way without having a test library for ACSLS to access. To do this:

1. Install the new ACSLS release on a separate server.
2. Export the database and control files from a production library environment using the `db_export.sh` utility. Refer to the *StorageTek ACSLS Administrator's Guide* for details.

   > **Note:**
   >
   > ACSLS must be down to export the database and control files.

3. Import the database and control files into your new ACSLS release using `db_import.sh`.
4. On your new ACSLS system, ensure that ACSLS does not try to connect to the imported library configuration. The ACSs and ports **must** stay offline to ACSLS.

   Otherwise, both the new ACSLS system and production system try to connect to the library, disconnecting the other system, and then in turn being disconnected by the other system. This repeats until one of the ACSLS systems is shut down.

   To keep all ACSs and port connections offline:

   - Modify the `acsls_startup_policy` file, in `$ACS_HOME/data/external/`.
   - Uncomment the line for each ACS that is configured in the imported database. Look at the comment header of `acsls_startup_policy` for details.

     For example, to prevent ACSLS from trying to bring ACS 0 online, change:

     # ACS0_desired_startup_state_is_offline

     to

     `ACS0_desired_startup_state_is_offline`

5. Test to ensure that ACSLS comes up and runs, exercising a limited set of commands.
   - Do **not** vary ports or ACSs online. If you do, you will halt library communication from your production ACSLS system.
   - Commands that send requests to the library will fail because the library is offline. However, ACSLS will continue to run and process requests.
   - Commands that do not rely on library resources work. These include submitting these commands using the ACSAPI from host applications:

     `query`

     `display`

     `define pool` and `delete pool`

     `idle` and `start`

     `lock` and `unlock`

     `set` commands, except for `set cap mode` which will fail because the library is offline.
   - Utilities that do not rely on library resources work. These include:

     `acsss` commands such as `acsss enable`, `acsss disable`, `acsss status`.

     `bdb.acsss` and `rdb.acsss`

     `db_export.sh` and `db_import.sh`

     > **Note:**
     >
     > `db_import.sh` overlays the `acsls_startup_policy` file. If this is a production system, this allows libraries to come online. Modify the `acsls_startup_policy` file before starting ACSLS.

     `dv_config`

     `drives_media.sh`

     `free_cells.sh`

     `userAdmin.sh`

     `volrpt`

     `watch_vols`
   - The ACSLS GUI will display library resources. However, commands such as `mount`, `dismount`, `enter`, and `eject` which requires library resources will fail.

Verifying the ACSLS Installation

To verify the ACSLS installation:

1. Ensure that your library is configured.

   Follow the instructions provided in the *ACSLS Administrator's Guide* to use `acsss_config` to configure ACSLS and create a database image of your library.

> **Note:**
>
> If you plan to use the SL4000 library, before running `acsss_config`, ensure that you have completed the following library configuration tasks using the SL4000 GUI:
>
> - Define an SL4000 library certificate, including the **Library Name (CN)**. This name must match that used in `acsss_config` and `config new acs`. If using a host name (DN), not an IP address, it must also resolve to the same exact name.
> - Define an SL4000 user that the ACSLS SCI interface can use to connect to the SL4000 library.
>
>   > **Note:**
>   >
>   > ACSLS SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.
>
> - Ensure that the SL4000 library is SCI capable, or has an SCI capable partition.
> - Ensure ACSLS server time and SL4000 library time are synced within a couple minutes of each other.
>
> Refer to the *ACSLS Administrator's Guide* for more information about these tasks.

2. Log in as user `acsss`.
3. Run the `acsss enable` command to start ACSLS.
4. Run `cmd_proc`.
5. From `cmd_proc`, query the server:
6. Verify that the following are online:

   ```
   query port all
   query acs all
   query lsm all
   query cap all
   query drive all
   ```

   At least one of each must be online. If necessary, use the vary command to bring them online.
7. Audit the library.

   Refer to "Auditing the Library" in the *StorageTek ACSLS Administrator's Guide*.
8. Do you have at least one cartridge in an LSM?
   - YES - Continue with the procedure.
   - NO - Enter a cartridge into an LSM.
9. List available volume and drive IDs.

```
query vol all
query drive all
```

10. Mount a volume:

    ```
    mount vol_id drive_id
    ```

    where `vol_id` is the volume ID and `drive_id` is the drive ID.

    Refer to the *StorageTek ACSLS Administrator's Guide* for more information.

11. Do you see a message indicating a successful mount?

    A successful mount message is:

    ```
    Mount: vol_id mounted on drive_id
    ```

    - YES - Procedure is complete.
    - NO - If an error message appears, run this verification procedure again, ensuring that you specified a valid, available drive and a library cartridge. If the mount continues to fail, contact Oracle Support for assistance.

12. Dismount the cartridge by entering:

    ```
    dismount vol_id drive_id force
    ```

    where `vol_id` is the volume and `drive_id` is the drive you mounted earlier in the procedure.

13. The verification procedure is complete.

3 Installing ACSLS on Linux

This chapter describes how to install ACSLS Release 8.5 in a Linux environment.

Topics include:

- Preparing for Installation
- Installing ACSLS
- Performing Post Installation Tasks

> **Note:**
>
> Logical libraries are not supported in the Linux environment.

Preparing for Installation

Perform the following tasks to prepare for ACSLS installation. Once you have completed these tasks, you are ready to install ACSLS 8.5.

- Step 1: Export Existing Database and Control Files
- Step 2: Remove Previous ACSLS Versions
- Step 3: Ensure Linux is Installed
- Step 4: SELinux Security Settings
- Step 5: Cron Administration
- Step 6: ACSLS Access Privileges
- Step 7: Adjust Linux Tuning Settings
- Step 8: Download and Unzip the ACSLS 8.5 Package
- Step 9: Configure YUM
- Step 10: Create User Accounts and Groups

Step 1: Export Existing Database and Control Files

If you are upgrading from a previous release and plan to use existing database and control files, you must export these files.

1. As user `acsss`, enter the following command:

   ```
   db_export.sh -f /path/myExport
   ```

   where *`myExport`* is the name of your export file.

2. Save both *`myExport`* and *`myExport`*`.misc` files to a non-volatile location.

3. If you are updating your operating system, then transfer these files to a remote machine for safe keeping.

For more information, refer to the “Database Administration" chapter in the *StorageTek ACSLS Administrator's Guide*.

Step 2: Remove Previous ACSLS Versions

Remove any previous version of ACSLS. If this is a new installation with no previous version of ACSLS, then skip this step.

1. Ensure that you have exported the database, using the `db_export.sh` utility command.
2. Log in as user `acsss`.
3. Shut down all ACSLS services:

   ```
   acsss shutdown
   ```

4. Remove any `acsss`, `acssa`, and `acsdb crontab` entries:
   - Login as user `acsss`; Execute a `crontab -r`; logout
   - Login as user `acssa`; Execute a `crontab -r`; logout
   - Login as user `acsdb`; Execute a `crontab -r`; logout
5. Remove the previous version of ACSLS for Linux:

   ```
   yum remove ACSLS.x86_64
   ```

6. Remove the postgreSQL database:

   ```
   yum remove PostgreSQL.x86_64
   ```

7. As user `root`, remove previously populated directories:

   ```
       rm -rf /export/home/ACSSS (or other directory where you installed ACSLS)
       rm -rf /export/home/SSLM (or other direcrtory)
       rm -rf /export/home/Oracle (or other directory)
       rm -rf /var/tmp/acsls
       rm -rf /opt/ACSLS_8.4.0
       rm -rf /opt/ACSLS_8.5.0
   rm -rf /opt/oracle/postgresql-10
   ```

8. Reboot.

Step 3: Ensure Linux is Installed

Ensure that a compatible version of Linux is installed.

- ACSLS Release 8.5.0 is designed to run under Oracle Enterprise Linux releases 6.8 and 7.3
- ACSLS Release 8.5.1 is designed to run under Oracle Enterprise Linux releases 6.8, 6.10, 7.3, 7.6, 7.8 and 7.9

The Oracle Enterprise Linux Product Pack can be obtained from the Oracle Software Delivery Cloud at:

edelivery.oracle.com

Oracle recommends installing the entire Linux distribution to ensure your installation includes the standard packages required for ACSLS operation.

> **Note:**
>
> If you perform an update to the Linux operating system after ACSLS is installed, ensure that you issue the `updatedb` command before rebooting the server. An operating system upgrade may impact ACSLS service operation during the reboot.

Before installing a new version of Linux, check with your IT system administrator to obtain the following information. The graphical installer requires the `kdelibs` package, which is included in the Oracle Enterprise Linux Product Pack.

- Hostname and IP address for the ACSLS server.
- Gateway IP address and netmask for your network, as well as the primary and secondary DNS.
- IP address.
- Network proxy information, if available.

During the installation, several key software components are installed:

- GNOME desktop environment.
- Internet support.
- X Windows.
- Resource Package Manager (RPM), Yellowdog Updater, and Modified (yum).
- Java 7 or 8. If you are installing the ACSLS GUI, use the latest Java JDK/SE version. Refer to the *ACSLS Product Information* document for specific required Java versions.

Do not install (or enable) the following:

- Software Development
- Web Server
- Database
- Dial-up network

Linux 7 Dependencies

Oracle recommends installing the entire Linux distribution to ensure your installation includes the standard packages required for ACSLS operation.

Additionally, Linux 7 installations include the following dependencies:

- glibc.i686
- pam
- pam.i686
- libstdc++
- libstdc++.i686
- libxml2

- libxml2.i686
- unixODBC.i686
- openssl
- openssl-libs.i686
- rpcbind
- libgssglue
- libcrypto.so.10
- bzip2
- mlocate

> **Note:**
>
> This represents a partial list. Additional packages may be required.

Installing Missing Oracle Linux Packages

If your installation is missing a standard Oracle Linux package required for ACSLS operation, use yum on the command line to acquire and install the missing package.

For example, to find and install a missing `bzip2` package:

1. Configure `/etc/yum.conf`.
2. Enter the following command:

   ```
   yum install bzip2
   ```

> **Note:**
>
> - If packages contain shared object libraries required by ACSLS, you must install 32-bit versions (for example, unixODBC).
> - If packages run a standalone process required by ACSLS, either 32-bit or 64-bit versions will work (for example, rpcbind).
>
> If a package is not working as expected, or causes faults, you may need to install a different version of the package. Examples include:
>
> - rpcbind (Some versions don't restart after reboot. For example, rpcbind.x86_64 on Oracle Linux 7.3 uses the version 0.2.0-48.el7.)
> - Java (ACSLS has specific minimum supported versions for this and other packages. Refer to the *ACSLS Product Information* document for specific required Java versions.)
> - unixODBC (may have installed the 64-bit version instead of the required 32-bit version)

Step 4: SELinux Security Settings

ACSLS 8.5 is designed to run in *optional* Security Enhanced Linux (SELinux) environments.

SELinux was merged into the Linux kernel in response to initiatives by the US National Security Agency. It provides access control to files, directories, and other system resources that go beyond the traditional protection found standard in UNIX environments. In addition to owner-group-public permission access, SELinux includes access control based on user role, domain, and context. The agent that enforces access control over all system resources is the Linux kernel.

To set SELinux enforcement:

1. As user `root`, use the `setenforce` command to enable or disable SELinux enforcement.

   ```
   setenforce [Enforcing | Permissive | 1 | 0 ]
   ```

   - Specify `Enforcing` or `1` to enable enforcement.
   - Specify `Permissive` or `0` to disable enforcement.

2. Verify the SELinux enforcement status:

   ```
   getenforce
   ```

> **Note:**
>
> - This command requires that SELinux is enabled. Use the command `sestatus` to view the status of SELinux.
> - To view the current system enforcement status, use the command `getenforce`.

Three SELinux policy modules are loaded into the kernel when you install ACSLS: `allowPostgr`, `acsdb`, and `acsdb1`. These modules provide the definitions and enforcement exceptions that are necessary for ACSLS to access its own database and other system resources while SELinux enforcement is active. With these modules installed, you should be able to run normal ACSLS operations, including database operations such as `bdb.acsss`, `rdb.acsss`, `db_export.sh` and `db_import.sh` without the need to disable SELinux enforcement.

If problems occur, you may need to disable SELinux or run in permissive mode. For more information, refer to the "Troubleshooting" appendix in the *StorageTek ACSLS Administrator's Guide*.

Step 5: Cron Administration

Specific automated schedules known as *crontabs* are created for users `acsss` and `acsdb` when you run the `install.sh` utility. These crontabs are provided for ACSLS database maintenance backup activities.

An optional file, `/etc/cron.allow` (or `/etc/cron.d/cron.allow` on some systems) may exist on the system. This file controls which users are allowed to run the `crontab` command. If `cron.allow` exists, then user IDs for `acsss` and `acsdb` must be included in that file before you run `install.sh`. Otherwise, `crontab` creation for these users fails.

The file cron.deny exists by default on most systems. Any users listed in this file are explicitly denied access to the crontab command. Make sure that users acsss and acsdb are not contained in the cron.deny file.

Step 6: ACSLS Access Privileges

Note the following access privilege considerations:

- ACSLS 8.5 may be installed in any local file system. The ACSLS base directory and backup directories (for example, /export/home and /export/backup) must be mounted to allow SETUID so that user acsss can run as root. Super user access is required for scripts that start and stop ACSLS services and for scripts that collect diagnostic information for a support call.
- The acsss umask is set to 027 during installation.
- Network services, specifically rpcbind, must be enabled to allow ACSLS client communication unless the firewall security on ACSLS and all ACSAPI clients is configured without the need for the portmapper. For more information, refer to "Firewall Security" in the *StorageTek ACSLS Administrator's Guide*.

Step 7: Adjust Linux Tuning Settings

Adjust Linux tuning settings for your configuration. See Linux and ACSLS Tuning Settings.

Step 8: Download and Unzip the ACSLS 8.5 Package

To download and unzip the ACSLS 8.5 package:

1. Start a web browser on the system and visit the Oracle Software Delivery Cloud:

   https://edelivery.oracle.com

2. Click **Sign In** and enter the user name and password provided by your Oracle Support representative.
3. In the search field, enter acsls and select **StorageTek Automated Cartridge System Library Software (ACSLS)**.
4. In the search results, select ACSLS release level 8.5.0.0.0 or 8.5.1.0.0 to add it to the cart.
5. Click **Selected Software** to view the cart.
6. On the Selected Software screen, select your desired platform and click **Continue**.
7. On the Oracle Terms and Restrictions screen, review and accept the terms of the licenses. Click **Continue**.
8. Click **Download** and save the zip file to a common installation directory, typically /opt.
9. Before extracting the ZIP file, remove any previously installed versions of ACSLS installation directories. For example:

   ```
   rm -rf /opt/ACSLS_8.4.0
   rm -rf /opt/ACSLS_8.5.0
   ```

10. Unzip the compressed file. The extracted package set is found in the resulting ACSLS_8.5.0 or `ACSLS_8.5.1` subdirectory.

Step 9: Configure YUM

After Linux installation, add specific packages required for ACSLS from the Oracle yum repository.

If your ACSLS server is behind a firewall, you may need to configure your ACSLS Linux system to use a local proxy server.

1. Edit `/etc/yum.conf` to update the local proxy server:

   ```
   yum/conf
   Proxy=http://your local proxy server
   http_caching=packages
   ```

2. Edit `/etc/wgetrc` to update proxy and caching parameters:

   ```
   wgetrc
   #You can set the default proxies for wget to use for http, https, and ftp.
   #They will override the value in the environment.
   http_proxy=http://your local proxy server

   # Remove the comment sign (#) from this line:
   #use_proxy=on
   ```

3. Configure `yum` to use the Oracle repository for the correct architecture.

   - Linux 6.8 or 6.10:

     Copy the provided `yum` repository file to `/etc/yum.repos.d/`.

     > **Note:**
     >
     > There should be only one file in this directory, `public-yum-ol6.repo`.

   - Linux 7.3, 7.6, 7.8 or 7.9:

     Copy the provided `yum` repository file to `/etc/yum.repos.d/`.

     > **Note:**
     >
     > There should be only one file in this directory, `public-yum-ol7.repo`.

4. Edit the file `/etc/yum/pluginconf.d/refresh-packagekit.conf` and set `enabled=0` to disable the yum packagekit refresh (Linux 6.8 or 6.10 only).

With these pre-requisites completed, you are now ready to install the ACSLS 8.5 package.

Step 10: Create User Accounts and Groups

Create the user accounts and associated groups described in the table below. For command examples, see Installation Command Examples.

ACSLS 8.5 allows for a user-defined home directory for the ACSLS application. The parent directory of each user home directory is referenced by the variable, `$installDir`.

Note:

- It is your responsibility to define any required user account attributes such as passwords, based upon your specific configuration and processes.
- ACSLS user accounts (`acsss`, `acsdb`, and `acssa`) must execute `.profile` when logging in. In some instances, `.bash_profile` will override `.profile` for bash shell user accounts.
- If you use directories that cross external NFS or ZFS mount points, ensure that root level privileges extend across the mount points. Without these root level privileges, ACSLS installation may fail, or post-installation functionality issues may occur.

Table 3-1 Required ACSLS User Accounts (Linux)

User Account	Group Assignment	Home Directory	Command Shell	Description
acsss	acsls	`$(installDir)/ACSSS` Default example: `/export/home/ACSSS` Ownership/Permissions: • Directory Owner: `acsss:acsls` • Minimum permissions: `rwxr-x---`	/bin/bash	ACSLS control user
acssa	acsls	`$(installDir)/ACSSA` Default example: `/export/home/ACSSA` Ownership/Permissions: • Directory Owner: `acssa:acsls` • Minimum permissions: `rwxr-x---`	/bin/bash	ACSLS SA user
acsdb	acsls	`$(installDir)/acsdb/ACSDB1.0` Default example: `/export/home/acsdb/ACSDB1.0` Ownership/Permissions: • Directory Owner: `acsdb:acsls` • Minimum permissions: `rwxr-x---`	/bin/bash	ACSLS DB user
postgres	postgres	`/opt/oracle/postgresql-10` Ownership/Permissions: • Directory Owner: `postgres:postgres` • Minimum. permissions: `rwxr-xr-x`	/bin/bash	postgres user

Table 3-1 (Cont.) Required ACSLS User Accounts (Linux)

User Account	Group Assignment	Home Directory	Command Shell	Description
root	no requirement	standard root Ownership/Permissions: user defined	/bin/bash	root user

If the user accounts already exist and are locked, you must unlock each account before you install the package.

For example, to check if the acsss account is locked:

```
# passwd -S acsss
acsss LK
```

`LK` indicates that the account is locked. To unlock the account:

```
# passwd -u acsss
```

If these user accounts exist on an LDAP or NIS server and the `root` user on the local machine lacks `usermod` authority on the LDAP or NIS server, then manual intervention by the system administrator is required to complete the ACSLS installation. Make sure the users are reassigned to the `acsls` group and their home directories conform as stated above. The user shell should be `bin/bash`.

Installing ACSLS

Perform the following tasks to install ACSLS:

1. Ensure that you have completed all pre-installation tasks described in "Preparing for Installation".
2. Log in as user `root`.
3. From the `/opt/ACSLS/ACSLS_8.5.0` or `/opt/ACSLS/ACSLS_8.5.1` directory, run the `pkg_install.sh` utility:

   ```
   ./pkg_install.sh
   ```

4. The utility prompts you to enter the full path directory for the installation.
   Enter a desired directory path, or press **Enter** to accept the default path (`/export/home`).
   If the directory you specify does not exist, the script prompts for permission create the directory.
5. The utility lists additional packages required by ACSLS and asks:

   ```
   OK to install (y/n):
   ```

   Enter **`y`** to install the additional packages and continue with installation, or **`n`** to terminate the installation.

   When you enter **`y`**, installation begins. Progress is displayed on screen. Installation may take significant time based on network and server configuration settings. `pkg_install.sh` relies on yum to install ACSLS and various dependencies. In addition to installing additional required packages, the utility also verifies the required user accounts and groups.

6. Once pkg_install.sh has completed, as user root, enter the updatedb command to ensure that any newly-added Linux packages are available in remaining installation steps, including install.sh script processing.

> **Note:**
>
> updatedb is also recommended when performing an operating system update after ACSLS installation. Enter this command before rebooting the server. An operating system upgrade may impact ACSLS service operation during the reboot.

7. Enter the following command to inherit the ACSLS environment:

```
. /var/tmp/acsls/.acsls.env
```

8. As user root, enter the following commands to run the ACSLS install.sh utility:

```
cd $ACS_HOME/install
./install.sh
```

9. The install.sh utility asks:

```
Core dump files help diagnose issues when they occur.

To do this the following will be modified:
- File permissions /var/crash will be changed
- core_pipe_list,core_uses_pid, core_pattern, suid_dumpable will be modified
- sysctl.conf will be modified, original one stored as .orig
- limits.conf will be modified, original one stored as .orig
- ulimit core updated
- service abrtd will be started

Can we make the above changes to enable core dump files on your server?
(y or n):
```

Enter y to enable the core dump feature. with this feature enabled, ACSLS processes that encounter a SEGV fault will generate a core dump and place it in the /var/crash directory. These core dump files are helpful in diagnosing issues with ACSLS. Provide these files to Oracle Support when they become available.

If you enter n, core dumps will not be generated.

To disable the core dump feature, enter the following commands:

```
ulimit -c 0
cp /etc/security/limits.conf.orig /etc/security/limits.conf
cp /etc/sysctl.conf.orig /etc/sysctl.conf
```

To enable the core dump feature at a later time, re-run install.sh or use the following procedure:

a. Log in as user root.

b. Enter the following commands:

```
. /var/tmp/acsls/.acsls_env

cd /export/home/ACSSS/bin
./enableLinuxDumps.sh
```

> **Note:**
>
> If you choose to enable the core dump feature, you must regularly monitor and manage your core dump files to ensure that they do not consume all available disk space on the ACSLS server.

10. The utility asks:

```
Do you wish to host the ACSLS Graphical User Interface? (y/n)
```

The ACSLS GUI is an optional feature. If you are co-hosting ACSLS with another application that uses WebLogic, enter **n** and then proceed with ACSLS installation.

Otherwise, enter **y** to install the GUI.

> **Note:**
>
> - Ensure that you have installed the latest version of the Java Development Kit (JDK) on your ACSLS server. See "ACSLS GUI Requirements".
> - If ACSLS installation fails during installation of the GUI, review the logs in `ACSSS/log/sslm`. These logs provide information as to why the GUI failed, in particular the `weblogic.log`.

If this is a minor update or configuration change (not a new installation) your ACSLS GUI may already be installed.

In this case, you have the option to re-install the GUI or to skip this section and retain the current ACSLS GUI domain.

The utility asks:

```
The Acsls GUI Domain exists. Do you want to re-install it? (y/n)
```

- Enter **y** if you are installing a new ACSLS release.
  The WebLogic server package is extracted and the default GUI `admin` user account is created with the user name, `acsls_admin`.

  You are then asked to assign a password for the `admin` user. The password must be between eight and sixteen characters using both alpha and numeric characters.

  The installation procedure unpacks and deploys the ACSLS GUI application and then creates the `Acsls` user group. At a later time, you can add GUI users to this group using the administrative tool, `userAdmin.sh`.
- If you enter **n**, you have the option to remove the existing GUI configuration.

When you install WebLogic on your ACSLS server, a simple 512-bit public key is automatically available to support basic https exchanges with client browsers. Normally, no further configuration should be necessary. However, some browsers, notably the Microsoft Internet Explorer, require a lengthier key of no less than 1024 bits. See "Configuring a Self-Signed Digital Certificate for HTTPS" for a description of and procedures for configuring an SSL encryption key.

11. The utility asks:

```
Which file system will be used to store database backups? [/export/backup]
```

Enter a desired directory path where you intend for database backup files to reside, or press **Enter** to accept the default path.

If your desired directory does not exist, you must first create it. The directory must be owned by root with permissions set to 755.

> **Note:**
>
> Ensure that permissions for directory `/opt/oracle` are set to 755 to avoid ACSLS database installation failures related to postgreSQL.

12. The utility asks:

    ```
    Shall we install the mchanger driver for fibre-attached libraries? (y/n)
    ```

    Enter **`y`** if your library environment includes a fibre-attached library such as the SL500 or SL150 library. Otherwise, enter **`n`**.

    If you enter **`y`**, the routine scans the attached SAN environment, looking for any StorageTek library devices. It reports the devices it finds and asks whether any additional libraries are attached. If you have an older SCSI attached L700 or L180 library, respond **`y`** to the prompt.

    For SCSI attached libraries, simply enter the *`target:lun`* address for each library, separating them by a space. For example:

    ```
    ==> 4:0 5:0 5:1
    ```

13. If you choose not to install the GUI or logical library support features, then the utility asks:

    ```
    Shall we install the optional lib_cmd interface (y or n):
    ```

    This optional feature is a command-line interface that performs many of the same operations available in the ACSLS GUI. While many `lib_cmd` operations apply to logical library functions, this feature is also useful for displaying the status of physical libraries, volumes and drives.

    The `lib_cmd` feature installs automatically when you choose to install either the GUI or logical library support.

    Enter **`y`** if you wish to install this feature.

14. Depending on the set of features that you have selected in the above installation dialog, this final step installs Linux init.d services to control the automatic start, stop, and status functions for each selected ACSLS feature.
    The service list includes any subset of the following:

    ```
    acsdb
    acsls
    rmi-registry
    surrogate
    weblogic
    ```

> **Note:**
>
> - If install.sh encounters errors due to missing packages, you must acquire and install these packages. SeeInstalling Missing Oracle Linux Packages.
> - During installation, install.sh attempts to regenerate certificates used for secure communications between the user and the ACSLS web-based interface. Certificates are also required when ACSLS manages one or more SL4000 libraries. If an error is encountered during this step, install.sh presents the following prompt:
>
> ```
> Problems occurred while generating certificates, which
> are required for GUI support and for SCI libraries.
> Continue with installation? (This is NOT recommended) (y or n): n
> ACSLS installation is incomplete!
> Please review the /tmp/install.log.
> ```
>
> Continuing with installation is only recommended for a non-GUI legacy installation without SL4000 libraries. In this case, certificates are not required.

15. When the install.sh utility exits, **ACSLS installation is complete**.

Performing Post Installation Tasks

Once ACSLS is installed, you can perform the following post-installation tasks:

- Adjusting ACSLS Tuning Settings
- Installing the XAPI Service
- Importing Database and Control Files
- Testing ACSLS Without a Library
- Verifying the ACSLS Installation

Adjusting ACSLS Tuning Settings

Set recommended ACSLS tuning settings for your configuration. See "ACSLS Tuning Settings".

Installing the XAPI Service

The optional XML API (XAPI) service is an API that enables Enterprise level mainframe clients and servers to communicate using a common Enterprise Library Software (ELS) protocol over TCP/IP. ACSLS 8.5 and later releases can be configured with XAPI support.

To install the XAPI component:

1. Ensure you have installed the ACSLS package and run install.sh to finish the ACSLS installation.
2. Ensure you are logged in to the ACSLS server as root.

3. Source key ACSLS environment variables:

   ```
   . /var/tmp/acsls/.acsls_env
   ```

   (Note the required period and space before `/var/tmp/acsls/.acsls_env`).

4. Install the XAPI component:

   ```
   cd $ACS_HOME/install
   ./install_xapi.sh
   Installing the XAPI component for Oracle IBM mainframe clients. Continue? (y)
   ```

Importing Database and Control Files

Database and control files are customized files, user preferences, and local configuration files that are unique to your specific ACSLS environment.

If you exported existing database and control files before installing ACSLS 8.5, as described in "Step 1: Export Existing Database and Control Files", you can use the `db_import.sh` utility to import them once ACSLS 8.5 is installed.

Refer to the "Database Administration" chapter in the *StorageTek ACSLS Administrator's Guide* for this procedure.

Testing ACSLS Without a Library

After installing a new ACSLS release, you want to test it before using it to manage production libraries. If a test library environment is not available, this can be difficult because normally ACSLS must be configured to a library, and the library must be online for ACSLS to come up.

If you do not have a configured library or library partition available in a test environment, you can test a new ACSLS release in a limited way without having a test library for ACSLS to access. To do this:

1. Install the new ACSLS release on a separate server.
2. Export the database and control files from a production library environment using the `db_export.sh` utility. Refer to the *StorageTek ACSLS Administrator's Guide* for details.

   > **Note:**
   >
   > ACSLS must be down to export the database and control files.

3. Import the database and control files into your new ACSLS release using `db_import.sh`.
4. On your new ACSLS system, ensure that ACSLS does not try to connect to the imported library configuration. The ACSs and ports **must** stay offline to ACSLS.

   Otherwise, both the new ACSLS system and production system try to connect to the library, disconnecting the other system, and then in turn being disconnected by the other system. This repeats until one of the ACSLS systems is shut down.

   To keep all ACSs and port connections offline:

   - Modify the `acsls_startup_policy` file, in `$ACS_HOME/data/external/`.

- Uncomment the line for each ACS that is configured in the imported database. Look at the comment header of `acsls_startup_policy` for details.

  For example, to prevent ACSLS from trying to bring ACS 0 online, change:

  # ACS0_desired_startup_state_is_offline

  to

  `ACS0_desired_startup_state_is_offline`

5. Test to ensure that ACSLS comes up and runs, exercising a limited set of commands.

   - Do **not** vary ports or ACSs online. If you do, you will halt library communication from your production ACSLS system.
   - Commands that send requests to the library will fail because the library is offline. However, ACSLS will continue to run and process requests.
   - Commands that do not rely on library resources work. These include submitting these commands using the ACSAPI from host applications:

     `query`

     `display`

     `define pool` and `delete pool`

     `idle` and `start`

     `lock` and `unlock`

     `set` commands, except for `set cap mode` which will fail because the library is offline.

   - Utilities that do not rely on library resources work. These include:

     `acsss` commands such as `acsss enable`, `acsss disable`, `acsss status`.

     `bdb.acsss` and `rdb.acsss`

     `db_export.sh` and `db_import.sh`

     > **Note:**
     >
     > `db_import.sh` overlays the `acsls_startup_policy` file. If this is a production system, this allows libraries to come online. Modify the `acsls_startup_policy` file before starting ACSLS.

     `dv_config`

     `drives_media.sh`

     `free_cells.sh`

     `userAdmin.sh`

     `volrpt`

     `watch_vols`

   - The ACSLS GUI will display library resources. However, commands such as `mount`, `dismount`, `enter`, and `eject` which requires library resources will fail.

Verifying the ACSLS Installation

To verify the ACSLS installation:

1. Ensure that your library is configured.

   Follow the instructions provided in the *ACSLS Administrator's Guide* to use `acsss_config` to configure ACSLS and create a database image of your library.

   > **Note:**
   >
   > If you plan to use the SL4000 library, before running `acsss_config`, ensure that you have completed the following library configuration tasks using the SL4000 GUI:
   >
   > - Define an SL4000 library certificate, including the **Library Name (CN)**. This name must match that used in `acsss_config` and `config new acs`. If using a host name (DN), not an IP address, it must also resolve to the same exact name.
   > - Define an SL4000 user that the ACSLS SCI interface can use to connect to the SL4000 library.
   >
   >   > **Note:**
   >   >
   >   > ACSLS SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.
   >
   > - Ensure that the SL4000 library is SCI capable, or has an SCI capable partition.
   > - Ensure ACSLS server time and SL4000 library time are synced within a couple minutes of each other.
   >
   > Refer to the *ACSLS Administrator's Guide* for more information about these tasks.

2. Log in as user `acsss`.
3. Run the `acsss enable` command to start ACSLS.
4. Run `cmd_proc`.
5. From `cmd_proc`, query the server:
6. Verify that the following are online:

   ```
   query port all
   query acs all
   query lsm all
   query cap all
   query drive all
   ```

At least one of each must be online. If necessary, use the vary command to bring them online.

7. Audit the library.

   Refer to "Auditing the Library" in the *StorageTek ACSLS Administrator's Guide*.

8. Do you have at least one cartridge in an LSM?

   - YES - Continue with the procedure.
   - NO - Enter a cartridge into an LSM.

9. List available volume and drive IDs.

   ```
   query vol all
   query drive all
   ```

10. Mount a volume:

    ```
    mount vol_id drive_id
    ```

    where `vol_id` is the volume ID and `drive_id` is the drive ID.

    Refer to the *StorageTek ACSLS Administrator's Guide* for more information.

11. Do you see a message indicating a successful mount?

    A successful mount message is:

    ```
    Mount: vol_id mounted on drive_id
    ```

    - YES - Procedure is complete.
    - NO - If an error message appears, run this verification procedure again, ensuring that you specified a valid, available drive and a library cartridge. If the mount continues to fail, contact Oracle Support for assistance.

12. Dismount the cartridge by entering:

    ```
    dismount vol_id drive_id force
    ```

    where `vol_id` is the volume and `drive_id` is the drive you mounted earlier in the procedure.

13. The verification procedure is complete.

4

Installing ACSLS on the SL4000 Feature Card

This chapter describes ACSLS support for the SL4000 feature card.

Topics include:

Overview

Beginning with ACSLS 8.5, you can install the ACSLS server on a feature card inserted into the Base Card Cage of the SL4000 library. This upgrade provides a fully functional Oracle Enterprise Linux environment with ACSLS installed in a secure RAID-1 file system.

The feature card upgrade kit is an ordered option for the SL4000 library. It includes the following:

- One library controller card. This will be converted into a feature card.
- Two library controller storage cards, each card containing a hard drive and local power for the drive. These will be used by the feature card.
- One DC power converter.

The SL4000 Base Card Cage can accommodate two feature card upgrade kits.

- For ACSLS 8.5.0, only one feature card upgrade kit is supported.
- For ACSLS 8.5.1, one or two feature card upgrade kits are supported.

The feature card is shipped as a generic library controller card. Its character as an ACSLS feature card is established when the card is inserted into its designated position within the SL4000 library frame.

> **Note:**
>
> ACSLS on the feature card **does not** support multiple library connections.
>
> There is a one-to-one correspondence between the instance of ACSLS running on the feature card and the SL4000 library it supports. Accordingly, this instance of ACSLS is only used to manage the library where the feature card is installed and running. This also applies in a dual feature card configuration, even if both feature cards have ACSLS installed. If you have two feature card kits installed in the same library, even if both feature cards have ACSLS installed, you can still only use ACSLS to manage the single library where the cards are installed and running.
>
> Refer to Using the ACSLS Feature Card Availability Toolkit for more information on how to use ACSLS in a dual feature card configuration.

Installation Options

Feature card configuration is dependent on the ACSLS release level.

Feature Card with ACSLS 8.5.1

For ACSLS 8.5.1, one or two feature card upgrade kits are supported. You can now use a second feature card with its corresponding two feature storage cards for a backup copy of ACSLS or SDP2. Additionally, the ACSLS 8.5.1 Feature Card Availability Toolkit (FCAT) enables you to run ACSLS on dual feature cards to provide new availability capability for ACSLS on the feature card.

For this release, **the feature card requires Oracle Support for hardware related tasks. Once the hardware is configured, you can install and manage ACSLS using the information in this guide.** Alternatively, you can purchase Oracle Advanced Customer Services support to assist with installation.

ACSLS 8.5.1 supports SL4000 firmware version 1.0.2.75 or later. Verify the current version on the Feature Card at `/etc/version.txt`.

If you already have SDP2 installed on a feature card with its own associated storage cards, then you can now install ACSLS 8.5.1 on the second feature card, as long as you have upgraded your SL4000 firmware and both feature cards. You must upgrade all feature cards to the same firmware version as your library. Plan for the need to adjust, upgrade, or reinstall your SDP2 or ACSLS applications after a library firmware upgrade. Work with your Oracle Support representative to plan next steps.

For ACSLS 8.5.1, the application only makes use of the two feature storage cards located directly above the feature card. Although you can install a maximum of four feature storage cards, if only one application (ACSLS or SDP2) is installed on a single feature card, then only the two feature storage cards located directly above the feature card are used. The remaining feature storage cards are not used unless an additional feature card is installed and the application itself makes use of the feature storage cards. In that case, both feature cards and all four feature storage cards are used.

Feature Card with ACSLS 8.5.0

For ACSLS 8.5.0, only one feature card upgrade kit is supported. You **cannot** use a second feature card and associated feature storage cards for any application.

For this release, **the feature card and ACSLS are installed by Oracle Support**. Alternatively, you can purchase both feature card and ACSLS installation by Oracle Advanced Customer Services.

If you are interested in using the SL4000 feature card to run ACSLS 8.5.0, you must contact Oracle Support for an analysis of your tape storage environment, including planned and required usage. Oracle Support uses this analysis to determine whether the feature card can be used in your environment, and can then proceed with installation and configuration. The procedures to install ACSLS 8.5.0 **are not** the same as those for ACSLS 8.5.1. **Do not** follow the procedures outlined in this document to install with ACSLS 8.5.0.

ACSLS 8.5.0 supports SL4000 firmware version 1.0.1.69.30201 only.

If you already have SDP2 installed on a feature card with associated feature storage cards, then you cannot install ACSLS 8.5.0 at this time. Work with your Oracle Support representative to determine possible next steps.

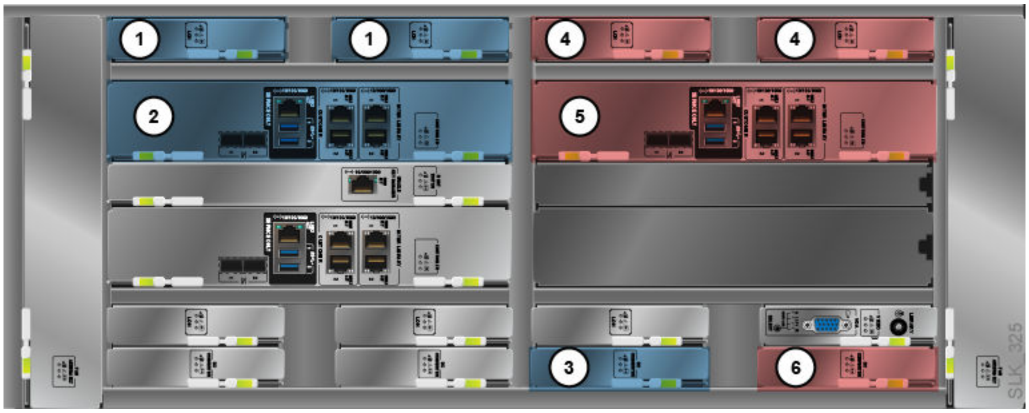
> **Note:**
>
> Oracle highly recommends that you upgrade the SL4000 library firmware to 1.0.2.75 or later and ACSLS to release 8.5.1 in order to take advantage of valuable enhancements.

Feature Card Locations

The rear of the SL4000 Base Module houses the Card Cage, which contains controller cards, disk storage, cooling fans, switches and power converters.

The following figure shows the feature card upgrade kit locations within the SL4000 Base Card Cage.

Figure 4-1 Base Card Cage Showing Feature Card Upgrade Locations



Feature Card Kit 1 (Side-A in FCAT configuration)

1. Feature Storage Cards (LOH)

2. Feature Card (LOC)

3. DC Power Converter (LOY)

Feature Card Kit 2 (Side-B in FCAT configuration)

4. Feature Storage Cards (LOH)

5. Feature Card (LOC)

6. DC Power Converter (LOY)

If you are using ACSLS with one feature card upgrade kit, Oracle recommends that the feature card is installed in the Feature Card Kit 1 (Side-A) location.

If you are using two feature card upgrade kits, and already have SDP2 installed on a feature card in the Feature Card Kit 1 (Side-A) location, ACSLS can be installed on the feature card in the Feature Card Kit 2 (Side-B) location.

Feature storage cards for each feature card are positioned directly above their respective feature cards.

Note:

- The feature card automatically boots and performs a base initialization the first time it is powered on. Therefore, only install the feature card when you are ready to install ACSLS. Ensure that your feature card contains the proper SL4000 firmware version. If you need to update firmware to version 1.0.2.75 or later, contact Oracle support.
- If you intend to install an additional feature card for SDP2, contact Oracle Support for assistance.
- The feature card's position is permanent and must remain consistent. An initialized feature card can only function when it resides in its permanent slot. This also applies to the feature storage cards associated with the feature card.

Pre-Installation Requirements for ACSLS on the Feature Card

Ensure that the following equipment and system information are in place prior to installation.

Note:

All hardware installation tasks are to be completed by Oracle Support.

User Equipment

Equipment Option 1:

- USB keyboard
- USB mouse
- External monitor and VGA cable
- Ethernet cable with internet access via SL4000
- Optional USB flash drive (to store the ACSLS software bundle after download)

Equipment Option 2:

- Laptop computer and Ethernet cable
- Optional USB flash drive (with ACSLS software bundle stored on it)
- WinSCP, PuTTY or similar software for file transfer and emulation

Library Equipment and Feature Card Information

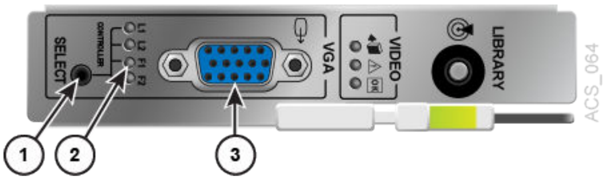
- One factory fresh library controller card with SL4000 firmware version 1.0.2.75 or later (this will become the feature card)
- Two library controller storage cards (feature storage cards)
- One DC Power Converter
- Site Domain Name, DNS Service IP(s), Search Domain, SL4000 IP address, SL4000 Hostname (Contact your System Administrator)
- Feature Card IP Address, Feature Card Hostname, Network Gateway Address, Netmask (Contact your System Administrator). If you are using the FCAT for a dual feature card configuration, you may require two IP addresses and two hostnames, depending upon your chosen configuration option. See Using the ACSLS Feature Card Availability Toolkit for more information.
- Feature Card access credentials (i.e. the user ID `root` and its password). Contact Oracle Support for these credentials as necessary.
- Location of local Yum repository or Yum Network Proxy and Network Proxy credentials
- ACSLS and SL4000 publications

Video, Network, and USB Connections

If you are using an external monitor, connect it to the VGA input in the video card located in the bottom of the SL4000 frame. Use the SELECT switch to set the video controller setting as shown in the figure below.

- Select Position 3 (F1) for the feature card in the left (Feature Card Kit 1) position within the SL4000 Base Card Cage.
- Select Position 4 (F2) for the feature card in the right (Feature Card Kit 2) position within the SL4000 Base Card Cage.

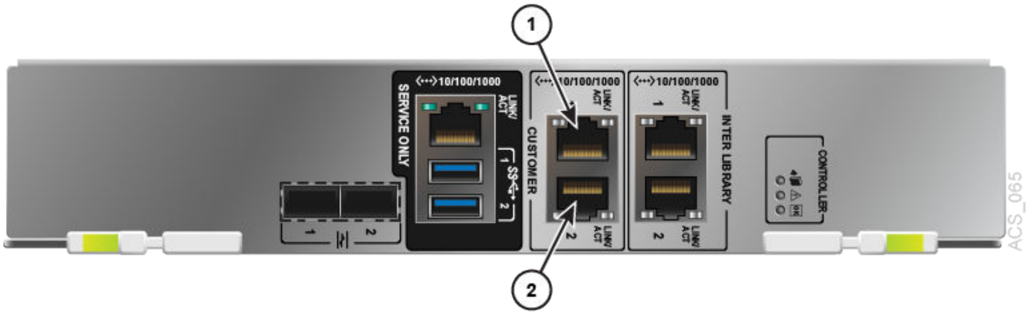
Figure 4-2 SL4000 Video Card



1. SELECT switch
2. Video controller setting
3. VGA video input

For the network connection, connect a CAT5 Ethernet cable to the top network access point in position 1 of the two CUSTOMER connectors on the feature card, as shown in the figure below. This position corresponds to `ifconfig` device id `p4p3`. The IP ports are not preconfigured by default. See Step 2: Establish a Temporary Connection to the External Network.

Figure 4-3 Feature Card Network Connection

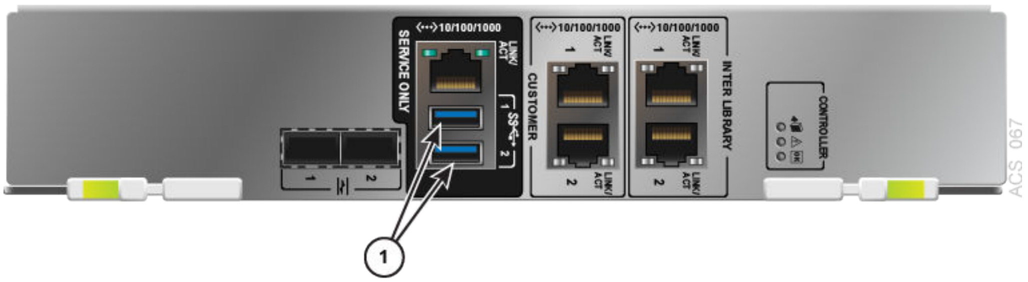


1. CUSTOMER network input 1
2. CUSTOMER network input 2

> **Note:**
>
> While the Feature Card can operate with a single network connection, the Feature Card initialization scripts detailed below establish an IP bond3 network to allow for redundant network connectivity.

If you are using an optional USB flash drive to upgrade the feature card's SL4000 firmware or ACSLS release, or to install any version of ACSLS, insert the flash drive into one of the USB ports on the feature card, as shown in the figure below. Note that USB ports are to be used by Oracle Support only.

Figure 4-4   Feature Card USB Ports



1. USB ports

Initializing the Feature Card and Storage Cards

> **Note:**
>
> All hardware installation tasks referenced in the following sections are to be completed by Oracle Support.

The factory shipped library controller card contains a base image of Linux 6.8 that is specific to the SL4000 feature card. The initialization service is used to establish the card as a feature card.

If the feature card has already been newly inserted into the library for ACSLS use, you can skip this step. Otherwise, perform the following tasks to initialize and verify your feature card in preparation for ACSLS feature card configuration and installation:

- Step 1: Insert and Install the DC Power Converter
- Step 2: Insert and Initialize the Feature Card
- Step 3: Verify Feature Card Initialization
- Step 4: Insert, Initialize, and Verify the Second Feature Card if Using Dual ACSLS Feature Cards
- Step 5: Insert and Initialize All Feature Storage Cards

Step 1: Insert and Install the DC Power Converter

Insert the DC power converter into the SL4000 library as shown in Feature Card Locations.

Step 2: Insert and Initialize the Feature Card

To initialize the feature card using the Feature Card Kit 1 (Side-A) location:

1. Insert and seat the factory fresh library controller card directly above the upper left feature card slot in the SL4000 library as shown in Feature Card Locations.

> **Note:**
>
> If using the Feature Card Kit 2 location, use the upper right feature card slot instead.

2. The library controller card automatically boots and runs an initialization service. A series of screens are displayed as the root file system is established on the feature card. This process establishes the designated library controller card as a feature card.

> **WARNING:**
>
> The feature card's position is permanent and must remain consistent. An initialized feature card can only function when it resides in its permanent slot. This also applies to the feature storage cards associated with the feature card.

3. Log in to the feature card as user `root`.

Step 3: Verify Feature Card Initialization

To verify that an individual feature card is initialized, log in to the feature card as user `root` and verify that the following are configured:

Use `df` to verify file system setup. Output should appear similar to the following:

```
# df
Filesystem      1K-blocks      Used  Available  Use%  Mounted on
/dev/sdX3        16382888   6446788    9080856   42%  /
tmpfs             8037332       224    8037108    1%  /dev/shm
/dev/sdX1          499656     74716     388244   17%  /boot
/dev/sdX2        64376668  10257336   50826148   17%  /u01
/dev/sdX6        25451616    306876   23828800    2%  /var
#
```

Note that `sdX` may be `sdc`, `sde`, or some other name.

Step 4: Insert, Initialize, and Verify the Second Feature Card if Using Dual ACSLS Feature Cards

If you are using the ACSLS Feature Card Availability Toolkit (FCAT) in a dual feature card configuration, perform the following steps for the second feature card:

1. For the second feature card, repeat the installation and initialization steps described in Step 2: Insert and Initialize the Feature Card.

2. For the second feature card, repeat the verification steps described in Step 3: Verify Feature Card Initialization.

Step 5: Insert and Initialize All Feature Storage Cards

Perform the following steps to initialize your feature storage cards:

1. For each feature card, verify that two associated feature storage cards are seated in their proper locations within the SL4000 Base Card Cage, directly above the feature card. If you are using the ACSLS Feature Card Availability Toolkit (FCAT), ensure that all four feature storage cards are installed. See Feature Card Locations for proper locations.
2. Perform a soft boot on each feature card and ensure that all feature cards are running. See Performing a Soft Boot of the Feature Card.

Configuring the Feature Card and Preparing for ACSLS Installation

> **Note:**
>
> - All ACSLS software tasks referenced in the following sections are to be completed by the customer or Oracle Advanced Customer Services, if purchased separately.
> - Feature card configuration scripts provide baseline configuration of the feature card to run ACSLS. ACSLS must be the only application running on the feature card. Other applications may interfere with ACSLS operations.
> - In the steps below that require you to manually edit specific configuration files, you must make a restorable copy of the originals, which will be restored in a later step.

Perform the following tasks to prepare the initialized feature card and its associated feature storage cards for use by ACSLS:

- Step 1: Connect User Equipment
- Step 2: Establish a Temporary Connection to the External Network
- Step 3: Download the ACSLS Software Bundle
- Step 4: Extract the ACSLS Feature Card Scripts
- Step 5: Configure the Feature Card for Bond3 Network Connectivity
- Step 6: Configure the Feature Card with DNS Servers and Other Required Settings
- Step 7: Configure the Feature Card Host Name
- Step 8: Initialize and Configure the Storage Cards
- Step 9: Verify Storage Card Configuration
- Step 10: Extract the ACSLS Software Bundle

Step 1: Connect User Equipment

Connect your user equipment to the library using one of the options described in User Equipment.

- If you are using User Equipment Option 1 or have not yet downloaded the ACSLS software bundle onto a flash drive, then proceed with Step 2: Establish a Temporary Connection to the External Network below.

- If you are using User Equipment Option 2, ensure that you have downloaded the ACSLS software bundle to your laptop or flash drive. Then proceed with Step 4: Extract the ACSLS Feature Card Scripts.

Step 2: Establish a Temporary Connection to the External Network

To establish the network connection:

1. If you have not already done so, log in to the feature card as user `root`.
2. Gather the following details for your newly initialized feature card. These are defined and assigned by your System Administrator, and will be associated with the feature card in these steps.
   - IP address `<IP_BOND3>`

     > **Note:**
     >
     > This is the Customer Network Interface IP (i.e. "public") address.

   - Netmask `<NM>`
   - Gateway address `<IP_GW>`
3. Using vi or your favorite text editor, edit and update the Internet configuration file for port `/etc/sysconfig/network-scripts/ifcfg-p4p3` as follows, and then save your updated settings.

   > **Note:**
   >
   > Remember to make a copy of the original file first.

   ```
   DEVICE=p4p3
   TYPE=Ethernet
   USERCTL=no
   NM_CONTROLLED=no
   BOOTPROTO=static
   IPV6INIT=no
   ONBOOT=yes
   IPADDR=<IP_BOND3>
   NETMASK=<NM>
   GATEWAY=<IP_GW>
   ```

4. Enable the `p4p3` communication port:

   ```
   ifconfig p4p3 <IP_BOND3> netmask <NM> up; route add default gw <IP_GW> p4p3
   ```

5. Using vi or your favorite text editor, edit and update the `/etc/resolv.conf` file as follows, and then save your updated settings.

   > **Note:**
   >
   > Remember to make a copy of the original file first.

```
domain <currentDomain>                      search <searchDomain>
<currentDomain>     nameserver <dns1_IP>     nameserver <dns2_IP>
nameserver <dns3_IP>
```

The `<currentDomain>` is your site's domain suffix, which may already be included in the file. Otherwise, you must ask your local System Administrator. The second and third `nameserver` entries are optional depending on whether your site provides multiple nameservers.

6. Ping the feature card to verify that the IP address is successfully configured. If there is no response then the connection has not been successfully established.

   ```
   ping <IP_BOND3>
   ```

Step 3: Download the ACSLS Software Bundle

On the feature card, download the latest bundle `ACSLS_8.5.1-X.Y_Linux.zip` from the Oracle Software Delivery Cloud website.

This step may require the Firefox browser (already installed on the feature card) or another terminal window to allow the download process to be monitored and guided.

Use the Firefox browser on the feature card to download ACSLS software.

1. From your login shell, enter the command `firefox &`.
   The Firefox browser is displayed on your desktop machine.
2. Once Firefox launches, configure your local network proxy preferences:
   a. Click the **open menu** icon at the top right of the browser, then select **preferences**.
   b. From the menu in the left-hand frame, select **Advanced**, then **Network**, then **Connection Settings**.
   c. From the pop-up menu, select or define the appropriate proxy for your local network environment.
3. Set the download location for the ACSLS software bundle to `/root/Downloads`.
   a. Click the **open menu** icon at the top right of the browser, then select **preferences**.
   b. From the menu in the left-hand frame, select **General.**
   c. Under **Downloads** in the right frame, make sure the **Save files to** radio button is selected.
   d. To the right of the radio button, click **Browse** and select **File System** from the left-hand frame.
   e. From the right-hand frame, select the `root` folder, then select `Downloads`, and click the **open** button at the bottom of the frame.

   The display under Downloads should show the selection, `Save files to /root/Downloads`.
4. In the Firefox browser, navigate to the Oracle Software Delivery Cloud. If the Oracle Cloud is not displayed, navigate to the Oracle Software Delivery Cloud at the following URL and follow the steps listed below.
   `https://edelivery.oracle.com`
   a. Click **Sign In**.
   b. Enter the user name and password provided by Oracle support.

   c. If offered, click **Accept** on the Export Restrictions screen.
   d. Select **Download Package** from the menu.
   e. Type `ACSLS` in the text box and click **Search**.
   f. Select the resulting product version you desire, and it will be added to your cart.
   g. Select **Checkout** from the upper right to display the Checkout screen.
   h. Under Platform/Languages, click the menu and select **Linux x86-64**. Then click **Continue**.
   i. On the Oracle Terms and Restrictions Screen, review and accept the terms of the licenses by checking the box and clicking **Continue**.
   j. Download your individual packages by clicking on each name.

5. Once the ACSLS software download is complete, you must perform the following steps to close the temporary connection and to place the system in the correct state for the remaining configuration steps to succeed:
   a. Enter the following command to disable the port:

      ```
      ifdown p4p3
      ```

   b. Restore the original `/etc/resolv.conf` file and remove all copies or backups of it from that directory.
   c. Restore the original `/etc/sysconfig/network-scripts/ifcfg-p4p3` file, and remove all copies or backups of it from that directory.

Step 4: Extract the ACSLS Feature Card Scripts

To extract the ACSLS software and associated scripts:

1. Clean up any previous downloads and installs to ensure the latest release is used. To do this, remove any previously downloaded versions of ACSLS 8.5.0 or 8.5.1 from the `/tmp` directory:

   ```
   cd /tmp
   rm -rf ACSLS_8.5.0*
   rm -rf ACSLS_8_5_1*
   ```

   Additionally, if the directory `/opt/ACSLS` exists, remove any previously downloaded versions of ACSLS 8.5.0 or 8.5.1:

   ```
   cd /opt/ACSLS
   rm -rf ACSLS_8.5.0*
   rm -rf ACSLS_8_5_1*
   ```

2. Copy the ACSLS download bundle from your download location of `/root/Downloads` or from your USB flash drive, into the feature card `/tmp` directory.

   For example, to copy the bundle from the `Downloads` directory to the `/tmp` directory:

   ```
   cp /root/Downloads/<ACSLS_bundle_name>.zip /tmp
   ```

> **Note:**
>
> If you are using a USB flash drive for the download, remove the flash drive from the library once the download is complete.

3. Once the ACSLS download bundle is in `/tmp`, unzip the bundle to extract the file `fc_config_scripts.zip`. This file contains the ACSLS feature card scripts.

   ```
   unzip -x <ACSLS_bundle_name>.zip ACSLS_8.5.1/fc_config_scripts.zip
   ```

   > **Note:**
   >
   > `fc` indicates "feature card" and should not be confused with "fibre channel".

   Ensure that the file `fc_config_scripts.zip` has been extracted to the directory `/tmp/ACSLS_8.5.1`.

   ```
   ls -alt /tmp/ACSLS_8.5.1
   ```

   The file `fc_config_scripts.zip` should appear in the directory listing for `/tmp/ACSLS_8.5.1`.

4. Extract the file `featureCard_unzipper.sh` from the `fc_config_scripts.zip` file in the directory `/tmp/ACSLS_8.5.1`:

   ```
   cd /tmp/ACSLS_8.5.1unzip -x fc_config_scripts.zip featureCard_unzipper.sh
   ```

   The following files should now appear in a directory listing for `/tmp/ACSLS_8.5.1`:

   ```
   fc_config_scripts.zipfeatureCard_unzipper.sh
   ```

5. As user `root`, execute the `featureCard_unzipper.sh` utility to automatically transfer the extracted files to their final locations on the ACSLS feature card. This utility also validates that you are installing on a supported SL4000 firmware version of the feature card.

   ```
   ./featureCard_unzipper.sh
   ```

   You should see a message similar to the following:

   ```
   The version of SL4000 Firmware hosted on feature card is '1.0.2.X.Y'.ACSLS can be
   installed on this feature card.
   ```

6. After running `featureCard_unzipper.sh`, verify the presence of the extracted files in the directory `/usr/local/bin`.

   ```
   ls -alt /usr/local/bin
   ```

   The following files should appear, among others:

   ```
   featureCard_acslsStorageManager.shfeatureCard_bond3.shfeatureCard_hostname.shfeatur
   eCard_resolvconf.sh
   ```

7. Verify the presence of the extracted files in the directory `/etc/init.d`.

   ```
   ls -alt /etc/init.d/
   ```

   The following files should appear:

   ```
   bootCounterFCmountFileSystemsFCstartupCheckFC
   ```

`FC` indicates that these services support ACSLS running on the feature card. These services are designed to run in the background.

Step 5: Configure the Feature Card for Bond3 Network Connectivity

> **Note:**
>
> The bond3 network is the default network configuration for ACSLS on the feature card. Optionally, you can customize the network configuration settings to suit your network infrastructure.

To configure the feature card with the bond3 Customer Network Interface as user `root`:

1. Go to the repository directory of the feature card configuration scripts.

   ```
   cd /usr/local/bin
   ```

2. Use the `ifconfig` command to verify that the bond3 network does not already exist.

   If it exists, then disable the bond3 network interface of the feature card using the following command:

   ```
   ./featureCard_bond3.sh -d
   ```

   This disables the ports for the remaining steps, and should not be required on a newly initialized feature card.

   > **Note:**
   >
   > Stop any applications currently running on the feature card.

3. Enter the following command to establish the `ifcfg` files for port `p4p1`, port `p4p3`, and `bond3`:

   ```
   ./featureCard_bond3 -c <IP_BOND3> <IP_GW> [NM]
   ```

   where:

   - IP address `<IP_BOND3>`
   - Gateway address `<IP_GW>`
   - Netmask `<NM>`

4. Verify the contents of directory `/etc/sysconfig/network-scripts` by listing the `ifcfg` file in the directory:

   ```
   ls -alt /etc/sysconfig/network-scripts
   ```

   Verify that the following files appear:

   ```
   ifcfg-p4p3
   ifcfg-p4p1
   ```

> **Note:**
>
> The file ifcfg-bond3 does not appear, as it is currently disabled.

5. Verify that configuration entries exist for ports p4p1 and p4p3, and the bond3 network interface on the feature card. This will show both enabled and disabled ports.

   ```
   ifconfig -a
   ```

6. The script featureCard_bond3.sh sets up the bond3 external network interface on the feature card which ACSLS will use for its operation. Next, enable the bond3 network interface:

   ```
   ./featureCard_bond3.sh -e
   ```

7. Verify that the feature card can now be pinged from outside the library or from the Library Controller card.

   ```
   ping <IP_BOND3>
   ```

Step 6: Configure the Feature Card with DNS Servers and Other Required Settings

To configure the feature card with your DNS servers:

1. Contact your System Administrator to obtain DNS server IP addresses, a local domain name, and the search list. This information will be used to update /etc/resolv.conf with your new feature card settings.

   You must have a minimum of one DNS server IP address. The maximum is three.

2. The script featureCard_resolvconf.sh sets up the following for the feature card:

   - DNS servers, <DNS1_IP> [DNS2_IP] [DNS3_IP]
   - Domain name, <DOMAINNAME>
   - One or more search domains, <SEARCHDomain1> …[SEARCHDomainN]

   The feature card configuration scripts are found in directory /usr/local/bin.

3. Set up and verify the DNS servers (domain name servers) for the feature card.

   ```
   ./featureCard_resolvconf.sh -dns <DNS1_IP> [DNS2_IP] [DNS3_IP]
   ```

   If not satisfied, then revert the change, as follows, before making another change.

   ```
   ./featureCard_resolvconf.sh -revert
   ```

   Note that you can only revert the very last change.

4. Once you are satisfied with the DNS server settings, then set up and verify the Domain Name <domainName> setting for /etc/resolv.conf on the feature card.

   ```
   ./featureCard_resolvconf.sh -domain <domainName>
   ```

   If not satisfied, then revert the change, as follows, before making another change.

   ```
   ./featureCard_resolvconf.sh -revert
   ```

   Note that you can only revert the very last change.

5. Once you are satisfied with the Domain Name setting, then set up the Search Domain list, `<searchDomain1>` … `[searchDomainN]`, for `/etc/resolv.conf` on the feature card.

   ```
   ./featureCard_resolvconf.sh -search <searchDomain1> … [searchDomainN]
   ```

   If not satisfied, then revert the change, as follows, before making another change.

   ```
   ./featureCard_resolvconf.sh -revert
   ```

   Note that you can only revert the very last change.

   > **Note:**
   >
   > Once the DNS servers and other required settings are configured, the resolv.conf file will be made immutable to ensure that the settings are not accidentally overwritten. If you need to update this file, use the same procedure as outlined in this section. Otherwise, to make changes to this file you may need to remove and re-enable the immutable attribute of the file: Use the following commands:
   >
   > ```
   > chattr -i /etc/resolv.conf
   > <make changes using your favorite editor>
   > chattr +i /etc/resolv.conf
   > ```
   >
   > When finished, ensure that you make the file immutable again.

Step 7: Configure the Feature Card Host Name

The script `featureCard_hostname.sh` changes the default host name of the feature card so that host applications can identify through the host name and communicate with ACSLS running on the feature card.

1. Assign a new host name to the feature card. This host name should be registered with DNS lookup or LDAP for your organization so that the feature card can be accessed by the ACSLS host applications as proper server.

   ```
   ./featureCard_hostname.sh <FC_HOSTNAME>
   ```

   where `FC` indicates the feature card.

   `FC_HOSTNAME` must be comprised of standard characters:

   - a-z
   - A-Z
   - 0-9
   - - (hyphen)
   - . (period)

   An _ (underscore) and other non-standard characters **cannot** be used.

2. After running `featureCard_hostname.sh`, verify that the `/etc/hosts` file contains the following.

   ```
   <IP_BOND3>      <FC_HOSTNAME>
   ```

3. Verify that the feature card can now be pinged by `<FC_HOSTNAME>` from outside the library or from the Library Controller card:

   ```
   ping <FC_HOSTNAME>
   ```

Step 8: Initialize and Configure the Storage Cards

The factory shipped library storage cards require initialization and configuration to be used in conjunction with the feature card. Configuration is specific to the application that is to be installed on the feature card. Do not use an SDP2 storage card configuration script for the ACSLS application, or vice versa. The configuration scripts are unique to the application and will cause faults if used otherwise.To initialize and configure the storage cards using the Feature Card Kit 1 location:

1. Ensure that the feature storage cards have been installed by Oracle Support and are running. See Step 5: Insert and Initialize All Feature Storage Cards.
2. From the feature card, as user `root`, run the following command to configure the storage for ACSLS on the feature card:

   ```
   cd /usr/local/bin./featureCard_acslsStorageManager.sh
   ```

3. As part of this process, RAID storage is mirrored. Watch the output on the screen to see it complete with a success message.

   To see more detail during, or after this process, use the following command:

   ```
   cat /proc/mdstat
   ```

   This command displays the configuration definition of the RAID mirror created by the `featureCard_acslsStorageManager.sh` command.

Step 9: Verify Storage Card Configuration

Once the `featureCard_acslsStorageManager.sh` script completes, the storage cards are configured. Verify this as follows:

Use `df` and the specific `ls` command below to verify file system configuration. Output should appear similar to the following:

```
# df
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/sdX3       16382888  6755128   8772516  44% /
tmpfs            8037336       76   8037260   1% /dev/shm
/dev/sdX1         499656    74812    388148  17% /boot
/dev/sdX2       65924860 10267712  52285324  17% /u01
/dev/sdX6       16382888   467952  15059692   4% /var
/dev/sdY1       47929224  1271936  44215920   3% /bkupa
/dev/sdZ1       47929224  1271936  44215920   3% /bkupb
/dev/mdN       191986276  3929464 178297736   3% /export
#
# ls -al /bkupc
lrwxrwxrwx 1 root root 10 Sep 24 20:51 /bkupc -> /u01/bkupc
#
```

Note:

- sdX, sdY, and sdZ, may be sda, sdb, sdc, sdd, or some other name, as determined by Linux. sdX refers to the SSD on the feature card; and /bkupc is also located on the SSD. sdY and sdZ refer to the hard drives on the storage cards.
- mdN may be md1 or md2, depending upon whether you have installed the feature card in the Side-A (md1) or Side-B (md2) location within the SL4000 Base Card Cage.

Alternatively you might use lsblk to readily see the partitions and mirrors. For example:

```
# lsblk
NAME    MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sdX       8:0    0 111.8G  0 disk
  sdX1    8:1    0   512M  0 part  /boot
  sdX2    8:2    0    64G  0 part  /u01
  sdX3    8:3    0    16G  0 part  /
  sdX4    8:4    0     1K  0 part
  sdX5    8:5    0     8G  0 part  [SWAP]
  sdX6    8:6    0    16G  0 part  /var
sdY       8:32   0 558.9G  0 disk
  sdY1    8:33   0  46.6G  0 part  /bkupa
  sdY2    8:34   0 186.3G  0 part
    mdN   9:2    0 186.1G  0 raid1 /export
sdZ       8:48   0 558.9G  0 disk
  sdZ1    8:49   0  46.6G  0 part  /bkupb
  sdZ2    8:50   0 186.3G  0 part
    mdN   9:2    0 186.1G  0 raid1 /export
```

Note:

In some cases, the featureCard_acslsStorageManager.sh script may need to reconfigure or otherwise perform cleanup, and ask you to reboot. Please reboot as directed. Once the system reboots, open a terminal window and run that script again so it can complete its work. See Step 8: Initialize and Configure the Storage Cards.

The configuration setup leverages the two attached disk drives (/dev/sdY and /dev/sdZ in the example above) to do the following:

- Formats both storage cards, each with a single 200GB partition for ACSLS data.
- Pairs the two disks into a RAID-1 configuration as device /dev/md1 (or /dev/md2).
- Mounts /dev/md1 (or /dev/md2) to /export.
- Mounts the RAID device.
- Creates two 50GB partitions for assorted backup files, sdY2 and sdZ2, including a copy of ACSLS_8.5.1.X.Y_Linux.zip.

- Creates the directories `/export/home`, `/export/backup`, `/export/PACKAGE`, and `/opt/ACSLS`, if not already present.

Oracle recommends that you label the feature card and the two feature storage cards using an external label. If you remove these components for servicing or other reasons, ensure that you replace them in the locations from which they were removed. Example labels might be:

- For Feature Card Kit 1: Feature Card Left –Side-A, Storage Card Left 1, Storage Card Left 2
- For Feature Card Kit 2: Feature Card Right –Side-B, Storage Card Right 1, Storage Card Right 2

Step 10: Extract the ACSLS Software Bundle

To extract the ACSLS software:

1. If you have previously attempted to install ACSLS on the feature card, you may have extraneous files remaining. Ensure that no previously downloaded versions of ACSLS are present.

   You may not have directories `ACSLS_8.5.0` or `ACSLS_8.5.1`, depending on whether any previous ACSLS installations were performed on the feature card and not subsequently removed.

   To remove previous versions of ACSLS:

   ```
   cd /opt/ACSLS
   rm -rf ACSLS_8.5.0*
   rm -rf ACSLS_8_5_1*
   ```

2. Copy the ACSLS download bundle from the feature card directory of `/tmp` to the ACSLS installation area. For example, to copy from the location `/tmp` to `/opt/ACSLS`:

   ```
   cp /tmp/ACSLS_8.5.1.X.Y.Linux.zip /opt/ACSLS
   ```

3. Unzip the ACSLS bundle:

   ```
   unzip ACSLS_8.5.1.X.Y.Linux.zip
   ```

   The ACSLS 8.5.1 files are extracted to the directory `/opt/ACSLS/ACSLS_8.5.1`.

4. Verify the presence of the extracted files in the directory `/opt/ACSLS/ACSLS_8.5.1`.

   ```
   ls /opt/ACSLS/ACSLS_8.5.1
   ```

   The following files should appear, amongst others:

   ```
   ACSLS-8.5.1-X.Y.x86_64.rpm
   fc_config_scripts.zip
   pkg_install.sh
   pkg_uninstall.sh
   PostgreSQL-10.5-0-OL6.x86_64.rpm
   PostgreSQL-10.5-0-OL7.x86_64.rpm
   public-yum-ol6.repo
   public-yum-ol7.repo
   README.txt
   wlUpdate
   ```

5. Once you have confirmed that all files are readily available, you can then remove any copies of the `ACSLS_8.5.1-X.Y.Linux.zip` from `/tmp` and `/root/Downloads`.

Installing, Configuring, and Running ACSLS on the Feature Card

Once you have initialized and configured the feature card, you must install and configure the ACSLS software on the card.

Step 1: Installing ACSLS

Installation of ACSLS 8.5 and later differs significantly from previous ACSLS releases.

Before installing ACSLS 8.5 and later on the feature card, ensure that you have completed all initialization and configuration tasks described earlier in this chapter.

Additionally, ensure that you have completed the pre-installation tasks described in Installing ACSLS on Linux. Reference the sections on configuring YUM, creating user accounts and groups, and installing ACSLS.

> **Note:**
>
> - ACSLS must be installed in the default directories on the feature card. User-defined directories are not supported.
> - ACSLS 8.5 and later uses the StorageTek Library Control Interface (SCI) protocol to connect with and operate the library. It does not use direct SCSI communications with the SL4000.
>
>   > **Note:**
>   >
>   > ACSLS SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.
>
> - ACSLS 8.5 and later on a feature card does not support the ACSLS High Availability package on Linux. Instead, use the ACSLS Feature Card Availability Toolkit (FCAT).
> - ACSLS 8.5 and later on a feature card does not support the ACSLS SNMP Agent.
> - You must follow all steps for configuring the SL4000 to ACSLS as described in the *ACSLS Administrator's Guide*.

Once installation is complete, ACSLS resides on a RAID-1 disk pair under the `/export` file system.

Three redundant backup directories, `/bkupa`, `/bkupb`, and`/bkupc`, store the downloaded ACSLS package and the customized Linux system files. Copies of unexpired ACSLS database backup files are also maintained in these locations.

You may need to manage these RAID-1 disk and backup directories when troubleshooting or addressing system faults associated with the feature card.

Step 2: Configuring and Running ACSLS

Follow the instructions provided in the *ACSLS Administrator's Guide* to use `acsss_config` to configure ACSLS and create a database image of your library. Although local backups of the database on the feature card will be created, it is highly recommended that you also establish periodic backups of your database to tape media or a storage server outside of the library as part of your own organization's disaster recovery processes.

Before running `acsss_config`, ensure that you have completed the following library configuration tasks using the SL4000 GUI:

- Define an SL4000 library certificate, including the **Library Name (CN)**. This name must match that used in `acsss_config` and `config new acs`. If using a host name (DN), not an IP address, it must also resolve to the same exact name.
- Define an SL4000 user that the ACSLS SCI interface can use to connect to the SL4000 library.

> **Note:**
>
> ACSLS SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.

- Ensure that the SL4000 library is SCI capable, or has an SCI capable partition.
- Ensure ACSLS server time and SL4000 library time are synced within a couple minutes of each other.

Refer to the *ACSLS Administrator's Guide* for more information about these tasks.

> **Note:**
>
> ACSLS on the feature card does **not** support multiple library connections.
> There is a one-to-one correspondence between an instance of ACSLS running on the feature card to the SL4000 library that it supports. Accordingly, ACSLS, when running on the feature card, should be used only to manage the SL4000 within which the feature card is installed. It should not be used to manage other libraries within your organization.

Once configured, enter the following command to enable `acsss` and begin operations:

```
acsss enable
```

This command is only valid if at least one ACS is configured.

For any additional operations, refer to your ACSLS publications, as all ACSLS operations run on the feature card as they do on a standalone server.

5

Un-Installing ACSLS

This chapter describes how to un-install ACSLS Release 8.5.

Topics include:

- Un-installing ACSLS on Solaris
- Un-Installing ACSLS on Linux

> **Note:**
>
> If you are upgrading from ACSLS 8.4 to ACSLS 8.5, refer to the *ACSLS 8.4 Installation Guide* for ACSLS 8.4 un-installation instructions.

Un-installing ACSLS on Solaris

This section describes how to un-install ACSLS 8.5 on Solaris, and optionally remove the XAPI and media changer components without installing the ACSLS software.

Removing the XAPI Service

Optionally, you can remove the ACSLS XAPI service ***without*** uninstalling ACSLS. This procedure is the same for both Solaris and Oracle Enterprise Linux platforms.

1. Log in as user `root` to the ACSLS server.
2. Source key ACSLS environment variables:

   ```
   . /var/tmp/acsls/.acsls_env
   ```

   Note the required period and space before `/var/tmp/acsls/.acsls_env`.
3. Uninstall the XAPI service:

   ```
   cd $ACS_HOME/install
   ./remove_xapi.sh

   Do you wish to remove the xapi service? (y)
   ```

Removing SCSI Media Changer (mchanger) Device Links

SCSI media changer (mchanger) drivers and device links are automatically removed when you uninstall the ACSLS software. However, you can optionally remove them ***without*** uninstalling ACSLS.

1. Log in as user `root`.
2. Remove the SCSI Media Changer (mchanger) drivers.

   ```
   # rem_drv mchanger
   ```

3. Remove `mchanger.conf`.

   ```
   # rm /usr/kernel/drv/mchanger.conf
   ```

4. Remove any mchanger device links.

   ```
   # rm /dev/mchanger*
   ```

5. Remove package directories.

   ```
   # rm -rf /opt/STKchanger
   ```

Uninstalling the ACSLS Software on Solaris

To un-install the ACSLS 8.5 software:

1. Log in as user `acsss`.
2. Shut down all ACSLS services:

   ```
   acsss shutdown
   ```

3. Log in as user `root`.
4. Go to the ACSLS_8.5.0 or ACSLS_8.5.1 package installation directory (typically `/opt/ACSLS_8.5.`*`x`*)
5. Run `pkg_uninstall.sh`.

   The `pkg_uninstall` script removes many, but not all ACSLS file systems and it keeps the user accounts in place for `acsss`, `acssa`, and `acsdb`. This approach allows for faster upgrades of ACSLS.
6. The `pkg_uninstall` script prompts you whether to uninstall the PostgreSQL packages.

   Enter **`N`** at this prompt unless you are permanently removing the ACSLS application.
7. Remove the contents of the ACSLS database backup directory:

   ```
   rm -rf $ACSDB_BKUP
   ```

8. WebLogic and the ACSLS GUI are not removed automatically during a package uninstall for the following reasons:

   - Upgrading ACSLS may not require an upgrade of WebLogic or the ACSLS GUI.
   - Uninstalling WebLogic and the ACSLS GUI removes ACSLS GUI users and their passwords.
   - Uninstalling WebLogic and the ACSLS GUI removes any custom SSL keystore that may have been configured for the ACSLS GUI.
   - Reinstalling WebLogic takes time (five minutes or more) to complete.

   To completely remove all remaining ACSLS components:

   ```
   cd $installDir
   ```

   ```
   rm -rf Oracle, SSLM
   userdel acsss
   userdel acssa
   userdel acsdb
   userdel postgres
   ```

```
groupdel acsls
groupdel postgres
```

9. Reboot.

   ACSLS is now uninstalled.

Un-Installing ACSLS on Linux

This section describes how to un-install ACSLS 8.5 on Oracle Enterprise Linux, and optionally remove the XAPI and media changer components without installing the ACSLS software.

Removing the XAPI Service

Optionally, you can remove the ACSLS XAPI service ***without*** uninstalling ACSLS. This procedure is the same for both Solaris and Oracle Enterprise Linux platforms.

1. Log in as user `root` to the ACSLS server.
2. Source key ACSLS environment variables:

   ```
   . /var/tmp/acsls/.acsls_env
   ```

   (Note the period and space before `/var/tmp/acsls/.acsls_env`).
3. Uninstall the XAPI service:

   ```
   cd $ACS_HOME/install
   ./remove_xapi.sh

   Do you wish to remove the xapi service? (y)
   ```

Removing SCSI Media Changer (mchanger) Drivers and Device Links

In Linux, `/dev/mchanger*` is a symbolic link to the standard SCSI Generic *sg* driver used when controlling fibre-attached libraries such as the SL150.

These mchanger device links are automatically removed when you uninstall the ACSLS software. However, you can optionally remove them ***without*** uninstalling ACSLS.

1. Remove the device links for mchanger in `/dev`.

   ```
   # cd /dev
   # rm mchanger*
   ```
2. Remove the rules that created the device links you removed in step 1.

   ```
   # cd /etc/udev/rules.d
   # rm persistent-storage-tape-acsls.rules
   ```

Uninstalling the ACSLS Software on Linux

To un-install the ACSLS 8.5 software:

1. Log in as user `acsss`.
2. Shut down all ACSLS services:

   ```
   acsss shutdown
   ```

3. Log in as user `root`.
4. Go to the ACSLS_8.5.0 or ACSLS_8.5.1 package installation directory (typically `/opt/ACSLS_8.5.x`)
5. Run `pkg_uninstall.sh`, if it exists.

   The `pkg_uninstall` script removes many, but not all ACSLS file systems and it keeps the user accounts in place for `acsss`, `acssa`, and `acsdb`. This approach allows for faster upgrades of ACSLS.

   Not all versions of ACSLS for Linux include this script. If the `pkg_uninstall` script does not exist, see Uninstalling ACSLS on Linux Without the pkg_uninstall Script.
6. The `pkg_uninstall` script prompts you whether to uninstall the PostgreSQL packages.

   Enter **`N`** at this prompt unless you are permanently removing the ACSLS application.
7. Remove the contents of the ACSLS database backup directory:

   ```
   rm -rf $ACSDB_BKUP
   ```
8. WebLogic and the ACSLS GUI are not removed automatically during a package uninstall for the following reasons:

   - Upgrading ACSLS may not require an upgrade of WebLogic or the ACSLS GUI.
   - Uninstalling WebLogic and the ACSLS GUI removes ACSLS GUI users and their passwords.
   - Uninstalling WebLogic and the ACSLS GUI removes any custom SSL keystore that may have been configured for the ACSLS GUI.
   - Reinstalling WebLogic takes time (five minutes or more) to complete.

   To completely remove all remaining ACSLS components:

   ```
   cd $installDir

   rm -rf Oracle, SSLM
   userdel acsss
   userdel acssa
   userdel acsdb
   userdel postgres
   groupdel acsls
   groupdel postgres
   ```
9. Reboot.

   ACSLS is now uninstalled.

Uninstalling ACSLS on Linux Without the pkg_uninstall Script

Not all versions of ACSLS for Linux include the `pkg_uninstall` script. To un-install the ACSLS 8.5 software without the `pkg_uninstall.sh` script:

1. As `root`, verify the ACSLS package that is currently installed:

   ```
   yum list installed ACSLS
   ```

   Example of an installed ACSLS:

```
yum list installed ACSLS

Loaded plugins: aliases, changelog, kabi, langpacks, tmprepo, ulninfo, verify, 
versionlock
Loading support for kernel ABI
Installed Packages
ACSLS.x86_64
8.5.1-22                                                       installed
```

2. As `root`, enter the command to remove the package:

   ```
   # yum remove ACSLS
   ```

   > **Note:**
   >
   > Ensure that no `acsss` owned processes are running on the Linux server when you enter this command.

3. Remove PostgreSQL:

   a. List all postgres-related packages:

      ```
      # yum list installed | grep -i postgres
      ```

   b. Remove all listed packages using the `yum remove <`*`pkg-name`*`>` command. For example:

      ```
      # yum remove PostgreSQL.x86_64
            <… output from remove operation…>

      # yum remove postgresql-libs.i686
            <… output from remove operation…>
      ```

      All packages associated with PostgreSQL are removed.

4. Reboot.
   ACSLS is now uninstalled.

A

Installation Command Examples

This appendix provides examples of network commands that can be issued by a system administrator during the installation process. Some of these commands are referenced in the installation chapters.

This listing is provided only as an example. These commands are dependent upon many factors, including server security configuration (LDAP, NIS, files, NSS services) and company policies and procedures governing creation, assignment, and removal of group and user accounts including UID and GID assignments. Consult with your IT administrator as there are many ways your server may be configured to handle authentication and management of users and groups.

```
#
# Verify User and Group Accounts for ACSLS and PostgreSQL Group
# and User accounts for users: acsss, acssa and acsdb and group acsls
#

# Verify Group account for acsls
getent group acsls

# Create acsls group if none is present
groupadd acsls

# Verify user account for acsss
getent passwd acsss

# Example output of existing acsss user:
> getent passwd acsss
acsss:x:505:516:ACSLS control login:/export/home/ACSSS:/bin/bash

# Create acsss user if none is present using default ACSLS
# install directory, adjust for user defined installation directory path
useradd -d /export/home/ACSSS -g acsls -s /bin/bash -c 'ACSLS control login'  acsss

# Verify user account for acssa
getent passwd acssa

# Example output of existing acssa user:
> getent passwd acssa
acssa:x:506:516:ACSLS SA login:/export/home/ACSSA:/bin/bash

# Create acssa user if none is present using default ACSLS install
# directory, adjust for user defined installation directory path
useradd -d /export/home/ACSSA -g acsls -s /bin/bash -c 'ACSLS SA login' acssa

# Verify user account for acsdb
getent passwd acsdb

# Example output of existing acsdb user:
> getent passwd acsdb
acsdb:x:507:516:ACSLS Database Owner:/export/home/acsdb/ACSDB1.0:/bin/bash

# Create acsdb user if none is present using default ACSLS install
# directory, adjust for user defined installation directory path
useradd -d /export/home/acsdb/ACSDB1.0 -g acsls -c 'ACSLS Database Owner' acsdb

#
# Group and User accounts for users: postgres and group postgres
#

# Verify Group account for postgres
getent group postgres

# Example output of existing postgres group:
> getent group postgres
postgres:x:26:

# Create postgres group if none is present
groupadd postgres

# Verify user account for postgres
getent passwd postgres

# Linux Example output of existing postgres user:
> getent passwd postgres
postgres:x:26:26:PostgreSQL Server:/opt/oracle/postgresql-10:/bin/bash

# Create postgres user if none is present
# using Linux Postgres install directory
useradd -d /opt/oracle/postgresql-10 -g postgres -c 'ACSLS Database' postgres
```

B

Linux and ACSLS Tuning Settings

This appendix describes tuning settings required in an environment running ACSLS 8.5 on Linux.

Topics include:

Linux Network Settings

For Linux 6.8, 6.10, 7.3, 7.6, 7.8 or 7.9, use the following guidelines to apply the appropriate settings based on the size of your system:

- Small system: 64 GB RAM or less
- Medium system: 64 GB to 128 GB RAM
- Large system: Greater than 128 GB RAM

Specify settings in the file `/etc/sysctl.conf`.

Network Settings - Small System

Specify the following for a system consisting of 64 GB RAM or less:

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 0

# Controls the default maxmimum size of a mesage queue
kernel.msgmnb = 65536
```

```
# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum number of shared memory in bytes
kernel.shmmax = 429494272

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 104857
```

Network Settings - Medium System

Specify the following for a system consisting of 64 GB to 128 GB RAM:

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 0

# Controls the default maxmimum size of a mesage queue
kernel.msgmnb = 65536

# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum number of shared memory in bytes
kernel.shmmax = 8589934592

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 2097152
```

Network Settings - Large System

Specify the following for a system consisting of greater than 128 GB RAM:

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0
```

```
# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 0

# Controls the default maxmimum size of a mesage queue
kernel.msgmnb = 65536

# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum number of shared memory in bytes
kernel.shmmax = 17179869184

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4194304
```

Linux 6.8 or 6.10 Operating System Settings

The following settings are recommended to accommodate the size and complexity of ACSLS. Use the following guidelines to apply the appropriate settings based on the size of your system:

- Small system: 64 GB RAM or less
- Medium system: 64 GB to 128 GB RAM
- Large system: Greater than 128 GB RAM

Specify settings in the file `/etc/security/limits.d/90-nproc.conf`.

Once these values are set, you must reboot the ACSLS server using the `reboot -p` command.

Linux 6.8 or 6.10 Operating System Settings - Small System

Specify the following for a system consisting of 64 GB RAM or less:

```
#
# ACSSS user limits
#

# Max core file size
acsss      hard    core   unlimited
acsss      soft    core   unlimited

# Max number of processes
acsss      hard    nproc   65568
acsss      soft    nproc   30000

# Max number of files open
acsss      hard    nofile   65568
acsss      soft    nofile   30000

# Max CPU usage
acsss      hard    cpu   unlimited
acsss      soft    cpu   unlimited
```

```
# Max number of locks open
acsss      hard    locks   65568
acsss      soft    locks   30000

# Max number data size
acsss      hard    data   unlimited
acsss      soft    data   unlimited

# Max number stack size
acsss      hard    stack   unlimited
acsss      soft    stack   16000

# Max number rss size
acsss      hard    rss   unlimited
acsss      soft    rss   1819200

# Max number address size
acsss      hard    as   unlimited
acsss      soft    as   unlimited

# Max size for memory locked
acsss      hard    memlock   unlimited
acsss      soft    memlock   2900000

# Max number stack size
acsss      hard    pipe   16000
acsss      soft    pipe   8192

# Max number of pending signals
acsss      hard    sigpending   257359
acsss      soft    sigpending   257359

#
# ACSDB user limits
#

# Max core file size
acsdb      hard    core   unlimited
acsdb      soft    core   unlimited

# Max number of processes
acsdb      hard    nproc   65568
acsdb      soft    nproc   30000

# Max number of files open
acsdb      hard    nofile   65568
acsdb      soft    nofile   30000

# Max CPU usage
acsdb      hard    cpu   unlimited
acsdb      soft    cpu   unlimited

# Max number of locks open
acsdb      hard    locks   65568
acsdb      soft    locks   30000

# Max number data size
acsdb      hard    data   unlimited
acsdb      soft    data   unlimited
```

```
# Max number stack size
acsdb      hard    stack   unlimited
acsdb      soft    stack   16000

# Max number rss size
acsdb      hard    rss   unlimited
acsdb      soft    rss   1819200

# Max number address size
acsdb      hard    as   unlimited
acsdb      soft    as   unlimited

# Max size for memory locked
acsdb      hard    memlock   unlimited
acsdb      soft    memlock   2900000

# Max number stack size
acsdb      hard    pipe   16000
acsdb      soft    pipe   8192

# Max number of pending signals
acsdb      hard    sigpending   257359
acsdb      soft    sigpending   257359
```

Linux 6.8 or 6.10 Operating System Settings - Medium System

Specify the following for a system consisting of 64 GB to 128 GB RAM:

```
#
# ACSSS user limits
#

# Max core file size
acsss      hard    core   unlimited
acsss      soft    core   unlimited

# Max number of processes
acsss      hard    nproc   65568
acsss      soft    nproc   30000

# Max number of files open
acsss      hard    nofile   65568
acsss      soft    nofile   30000

# Max CPU usage
acsss      hard    cpu   unlimited
acsss      soft    cpu   unlimited

# Max number of locks open
acsss      hard    locks   65568
acsss      soft    locks   30000

# Max number data size
acsss      hard    data   unlimited
acsss      soft    data   unlimited

# Max number stack size
acsss      hard    stack   unlimited
acsss      soft    stack   16000
```

```
# Max number rss size
acsss      hard    rss   unlimited
acsss      soft    rss   3638400

# Max number address size
acsss      hard    as   unlimited
acsss      soft    as   unlimited

# Max size for memory locked
acsss      hard    memlock   unlimited
acsss      soft    memlock   3900000

# Max number stack size
acsss      hard    pipe   16000
acsss      soft    pipe   8192

# Max number of pending signals
acsss      hard    sigpending   257359
acsss      soft    sigpending   257359

#
# ACSDB user limits
#

# Max core file size
acsdb      hard    core   unlimited
acsdb      soft    core   unlimited

# Max number of processes
acsdb      hard    nproc   65568
acsdb      soft    nproc   30000

# Max number of files open
acsdb      hard    nofile   65568
acsdb      soft    nofile   30000

# Max CPU usage
acsdb      hard    cpu   unlimited
acsdb      soft    cpu   unlimited

# Max number of locks open
acsdb      hard    locks   65568
acsdb      soft    locks   30000

# Max number data size
acsdb      hard    data   unlimited
acsdb      soft    data   unlimited

# Max number stack size
acsdb      hard    stack   unlimited
acsdb      soft    stack   16000

# Max number rss size
acsdb      hard    rss   unlimited
acsdb      soft    rss   3638400

# Max number address size
acsdb      hard    as   unlimited
acsdb      soft    as   unlimited

# Max size for memory locked
```

```
acsdb      hard    memlock   unlimited
acsdb      soft    memlock   3900000

# Max number stack size
acsdb      hard    pipe   16000
acsdb      soft    pipe   8192

# Max number of pending signals
acsdb      hard    sigpending   257359
acsdb      soft    sigpending   257359
```

Linux 6.8 or 6.10 Operating System Settings - Large System

Specify the following for a system consisting of greater than 128 GB RAM:

```
#
# ACSSS user limits
#

# Max core file size
acsss      hard    core   unlimited
acsss      soft    core   unlimited

# Max number of processes
acsss      hard    nproc   65568
acsss      soft    nproc   30000

# Max number of files open
acsss      hard    nofile   65568
acsss      soft    nofile   30000

# Max CPU usage
acsss      hard    cpu   unlimited
acsss      soft    cpu   unlimited

# Max number of locks open
acsss      hard    locks   65568
acsss      soft    locks   30000

# Max number data size
acsss      hard    data   unlimited
acsss      soft    data   unlimited

# Max number stack size
acsss      hard    stack   unlimited
acsss      soft    stack   16000

# Max number rss size
acsss      hard    rss   unlimited
acsss      soft    rss   3900000

# Max number address size
acsss      hard    as   unlimited
acsss      soft    as   unlimited

# Max size for memory locked
acsss      hard    memlock   unlimited
acsss      soft    memlock   3900000

# Max number stack size
```

```
acsss      hard    pipe   16000
acsss      soft    pipe   8192

# Max number of pending signals
acsss      hard    sigpending   257359
acsss      soft    sigpending   257359

#
# ACSDB user limits
#

# Max core file size
acsdb      hard    core   unlimited
acsdb      soft    core   unlimited

# Max number of processes
acsdb      hard    nproc   65568
acsdb      soft    nproc   30000

# Max number of files open
acsdb      hard    nofile   65568
acsdb      soft    nofile   30000

# Max CPU usage
acsdb      hard    cpu   unlimited
acsdb      soft    cpu   unlimited

# Max number of locks open
acsdb      hard    locks   65568
acsdb      soft    locks   30000

# Max number data size
acsdb      hard    data   unlimited
acsdb      soft    data   unlimited

# Max number stack size
acsdb      hard    stack   unlimited
acsdb      soft    stack   16000

# Max number rss size
acsdb      hard    rss   unlimited
acsdb      soft    rss   3900000

# Max number address size
acsdb      hard    as   unlimited
acsdb      soft    as   unlimited

# Max size for memory locked
acsdb      hard    memlock   unlimited
acsdb      soft    memlock   3900000

# Max number stack size
acsdb      hard    pipe   16000
acsdb      soft    pipe   8192

# Max number of pending signals
acsdb      hard    sigpending   257359
acsdb      soft    sigpending   257359
```

Linux 7.3, 7.6, 7.8, or 7.9 Operating System Settings

The following settings are recommended to accommodate the size and complexity of ACSLS. Use the following guidelines to apply the appropriate settings based on the size of your system:

- Small system: 64 GB RAM or less
- Medium system: 64 GB to 128 GB RAM
- Large system: Greater than 128 GB RAM

Specify settings in the file `/etc/security/limits.d/20-nproc.conf`.

Once these values are set, you must reboot the ACSLS server using the `reboot -p` command.

Linux 7.3, 7.6, 7.8, or 7.9 Operating System Settings - Small System

Specify the following for a system consisting of 64 GB RAM or less:

```
#
# ACSSS user limits
#

# Max core file size
acsss      hard    core   unlimited
acsss      soft    core   unlimited

# Max number of processes
acsss      hard    nproc   65568
acsss      soft    nproc   30000

# Max number of files open
acsss      hard    nofile   65568
acsss      soft    nofile   30000

# Max CPU usage
acsss      hard    cpu   unlimited
acsss      soft    cpu   unlimited

# Max number of locks open
acsss      hard    locks   65568
acsss      soft    locks   30000

# Max number data size
acsss      hard    data   unlimited
acsss      soft    data   unlimited

# Max number stack size
acsss      hard    stack   unlimited
acsss      soft    stack   8192

# Max number rss size
acsss      hard    rss   unlimited
acsss      soft    rss   1819200

# Max number address size
acsss      hard    as   unlimited
acsss      soft    as   unlimited

# Max size for memory locked
acsss      hard    memlock   unlimited
acsss      soft    memlock   2900000

# Max number stack size
acsss      hard    pipe   16000
acsss      soft    pipe   8192

# Max number of pending signals
acsss      hard    sigpending   257359
acsss      soft    sigpending   257359

#
# ACSDB user limits
#

# Max core file size
acsdb      hard    core   unlimited
acsdb      soft    core   unlimited

# Max number of processes
acsdb      hard    nproc   65568
acsdb      soft    nproc   30000

# Max number of files open
acsdb      hard    nofile   65568
acsdb      soft    nofile   30000

# Max CPU usage
acsdb      hard    cpu   unlimited
acsdb      soft    cpu   unlimited

# Max number of locks open
acsdb      hard    locks   65568
acsdb      soft    locks   30000

# Max number data size
acsdb      hard    data   unlimited
acsdb      soft    data   unlimited

# Max number stack size
acsdb      hard    stack   unlimited
acsdb      soft    stack   16000

# Max number rss size
acsdb      hard    rss   unlimited
acsdb      soft    rss   1819200

# Max number address size
acsdb      hard    as   unlimited
acsdb      soft    as   unlimited

# Max size for memory locked
acsdb      hard    memlock   unlimited
acsdb      soft    memlock   2900000

# Max number stack size
acsdb      hard    pipe   16000
acsdb      soft    pipe   8192
```

```
# Max number of pending signals
acsdb      hard    sigpending   257359
acsdb      soft    sigpending   257359
```

Linux 7.3, 7.6, 7.8, or 7.9 Operating System Settings - Medium System

Specify the following for a system consisting of 64 GB to 128 GB RAM:

```
#
# ACSSS user limits
#

# Max core file size
acsss      hard    core   unlimited
acsss      soft    core   unlimited

# Max number of processes
acsss      hard    nproc   65568
acsss      soft    nproc   30000

# Max number of files open
acsss      hard    nofile   65568
acsss      soft    nofile   30000

# Max CPU usage
acsss      hard    cpu   unlimited
acsss      soft    cpu   unlimited

# Max number of locks open
acsss      hard    locks   65568
acsss      soft    locks   30000

# Max number data size
acsss      hard    data   unlimited
acsss      soft    data   unlimited

# Max number stack size
acsss      hard    stack   unlimited
acsss      soft    stack   16000

# Max number rss size
acsss      hard    rss   unlimited
acsss      soft    rss   3638400

# Max number address size
acsss      hard    as   unlimited
acsss      soft    as   unlimited

# Max size for memory locked
acsss      hard    memlock   unlimited
acsss      soft    memlock   3900000

# Max number stack size
acsss      hard    pipe   16000
acsss      soft    pipe   8192

# Max number of pending signals
acsss      hard    sigpending   257359
acsss      soft    sigpending   257359
```

```
#
# ACSDB user limits
#

# Max core file size
acsdb      hard    core   unlimited
acsdb      soft    core   unlimited

# Max number of processes
acsdb      hard    nproc   65568
acsdb      soft    nproc   30000

# Max number of files open
acsdb      hard    nofile   65568
acsdb      soft    nofile   30000

# Max CPU usage
acsdb      hard    cpu   unlimited
acsdb      soft    cpu   unlimited

# Max number of locks open
acsdb      hard    locks   65568
acsdb      soft    locks   30000

# Max number data size
acsdb      hard    data   unlimited
acsdb      soft    data   unlimited

# Max number stack size
acsdb      hard    stack   unlimited
acsdb      soft    stack   8192

# Max number rss size
acsdb      hard    rss   unlimited
acsdb      soft    rss   3900000

# Max number address size
acsdb      hard    as   unlimited
acsdb      soft    as   unlimited

# Max size for memory locked
acsdb      hard    memlock   unlimited
acsdb      soft    memlock   3900000

# Max number stack size
acsdb      hard    pipe   16000
acsdb      soft    pipe   8192

# Max number of pending signals
acsdb      hard    sigpending   257359
acsdb      soft    sigpending   257359
```

Linux 7.3, 7.6, 7.8, or 7.9 Operating System Settings - Large System

Specify the following for a system consisting of greater than 128 GB RAM:

```
#
# ACSSS user limits
#
```

```
# Max core file size
acsss      hard    core   unlimited
acsss      soft    core   unlimited

# Max number of processes
acsss      hard    nproc   65568
acsss      soft    nproc   30000

# Max number of files open
acsss      hard    nofile   65568
acsss      soft    nofile   30000

# Max CPU usage
acsss      hard    cpu   unlimited
acsss      soft    cpu   unlimited

# Max number of locks open
acsss      hard    locks   65568
acsss      soft    locks   30000

# Max number data size
acsss      hard    data   unlimited
acsss      soft    data   unlimited

# Max number stack size
acsss      hard    stack   unlimited
acsss      soft    stack   16000

# Max number rss size
acsss      hard    rss   unlimited
acsss      soft    rss   3900000

# Max number address size
acsss      hard    as   unlimited
acsss      soft    as   unlimited

# Max size for memory locked
acsss      hard    memlock   unlimited
acsss      soft    memlock   3900000

# Max number stack size
acsss      hard    pipe   16000
acsss      soft    pipe   8192

# Max number of pending signals
acsss      hard    sigpending   257359
acsss      soft    sigpending   257359

#
# ACSDB user limits
#

# Max core file size
acsdb      hard    core   unlimited
acsdb      soft    core   unlimited

# Max number of processes
acsdb      hard    nproc   65568
acsdb      soft    nproc   30000
```

```
# Max number of files open
acsdb      hard    nofile   65568
acsdb      soft    nofile   30000

# Max CPU usage
acsdb      hard    cpu   unlimited
acsdb      soft    cpu   unlimited

# Max number of locks open
acsdb      hard    locks   65568
acsdb      soft    locks   30000

# Max number data size
acsdb      hard    data   unlimited
acsdb      soft    data   unlimited

# Max number stack size
acsdb      hard    stack   unlimited
acsdb      soft    stack   16000

# Max number rss size
acsdb      hard    rss   unlimited
acsdb      soft    rss   3900000

# Max number address size
acsdb      hard    as   unlimited
acsdb      soft    as   unlimited

# Max size for memory locked
acsdb      hard    memlock   unlimited
acsdb      soft    memlock   3900000

# Max number stack size
acsdb      hard    pipe   16000
acsdb      soft    pipe   8192

# Max number of pending signals
acsdb      hard    sigpending   257359
acsdb      soft    sigpending   257359
```

ACSLS Tuning Settings

This section provides specific details about how to reply to certain questions when running ACSLS `install.sh` and `acsss_config`. These details determine the settings for specific parameters, as well as controlling behavior of specific components within ACSLS.

Do the following:

1. Run ACSLS `acsss_config`
   IMPORTANT: Do this after running install.sh, and after any import of control files from ACSLS 7.3.1.
2. Select option 3: **Set general product behavior variables**
3. Increase the number of ACSMT (performs mounts/dismounts requests) processes from a default of 2 to the max of 5.

```
Changes to the number of mount processes ACSLS supports will not take effect until
the product is restarted.
Number of mount processes [2]: 5
```

4. Increase the number of ACSQY (performs various query requests) processes from a default of 2 to the max of 5.

   ```
   Changes to the number of query processes ACSLS supports will not take effect
   until the product is restarted.
   Number of query processes [2]: 5
   ```

5. Increase the number of concurrent ACSLS processes to 70.

   ```
   Changes to the maximum number of ACSLS processes will not take effect
   until the product is restarted.
   Maximum number of ACSLS processes [8]: 70
   ```

6. Turn off the ACSLM TCP/IP INET socket. You will be asked about the value for ENABLE_INET_ACSLM. Set it to **FALSE**, unless you have installed the ACSLS GUI or are using logical libraries.

   ```
   **** ENABLE_INET_ACSLM Must be TRUE ****
   This variable must be TRUE to allow the GUI and logical
   libraries to communicate with legacy ACSLS processes. [TRUE]: FALSE
   ```

   You may also do this using `dv_config` if it becomes necessary at any time in the future, using the command `dv_config -p ENABLE_INET_ACSLM`.

   > **WARNING:**
   >
   > DO NOT set `ENABLE_INET_ACSLM` to `FALSE` if you have installed the ACSLS GUI or are using logical libraries. In these cases, set this parameter to `TRUE` in order to avoid resource issues such as failed fork().

Verifying Tuning Settings

After rebooting the ACSLS server using the `reboot -p` command, verify your tuning parameter changes.

To verify operating system tuning settings:

1. Login in as user `root`.
2. Change user to `acsss` using the command `su - acsss`.
3. Perform Soft and Hard limit checks using the following commands:

   ```
   ulimit -aS
   ulimit -aH
   ```

4. Change back to user `root` using the command `exit`.
5. Change user to `acsdb` using the command `su - acsdb`.
6. Perform Soft and Hard limit checks using the following commands:

   ```
   ulimit -aS
   ulimit -aH
   ```

Examples:

```
-bash-4.1$ ulimit -aS
core file size          (blocks, -c) unlimited
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 257359
max locked memory       (kbytes, -l) 3900000
max memory size         (kbytes, -m) 8192000
open files                      (-n) 30000
pipe size            (512 bytes, -p) 8
POSIX message queues (bytes, -q)     819200
real-time priority  (-r)
0stack size (kbytes, -s)
16000cpu time (seconds, -t)
unlimitedmax user processes
(-u) 30000virtual memory
(kbytes, -v) unlimitedfile locks
(-x) 30000-bash-4.1$ ulimit -aHcore file size
(blocks, -c) unlimiteddata seg size
(kbytes, -d) unlimitedscheduling priority
(-e) 0file size
(blocks, -f) unlimitedpending signals
(-i) 257359max locked memory
(kbytes, -l) unlimitedmax memory size
(kbytes, -m) unlimitedopen files
(-n) 65568pipe size
(512 bytes, -p) 8POSIX message queues
(bytes, -q) 819200real-time priority
(-r) 0stack size
(kbytes, -s) unlimitedcpu time
(seconds, -t) unlimitedmax user processes
(-u) 65568virtual memory
(kbytes, -v) unlimitedfile locks
(-x) 65568
```

C

Using the ACSLS Feature Card Availability Toolkit

This appendix describes how to use the ACSLS Feature Card Availability Toolkit (FCAT) in a dual feature card configuration with ACSLS 8.5.1.

Topics include:

- Overview
- Configuring the FCAT Environment

Overview

Beginning with ACSLS 8.5.1, you can install and run ACSLS on dual feature cards within the SL4000 library.

Depending on your available hardware, application installation, and configuration, different forms of failover are supported by ACSLS to enable continuous operation for both ACSLS and attached client applications. The ACSLS Feature Card Availability Toolkit (FCAT) is a suite of tools and capabilities that have been both expanded and newly developed to provide this application support on the feature card.

FCAT scripts are included in the bundle of scripts you extracted during initial feature card configuration. See "Step 4: Extract the ACSLS Feature Card Scripts".

Pre-Installation Requirements for FCAT

Ensure that the following equipment and system information are in place prior to installation:

> **Note:**
>
> All hardware installation tasks are to be completed by Oracle Support.

- An additional FCAT Client Server, running a version of Linux supported by ACSLS. This can be:
  - A standalone server
  - A server in use with a separate Linux partition
  - A server with the space and network connectivity available to run the ACSLS FCAT monitoring and switch utilities.
- FCAT client server credentials
- User and library equipment as described in "Pre-Installation Requirements for ACSLS on the Feature Card".

- A separate ACSLS license for each feature card. Work with your Oracle Support and Sales representatives as appropriate.
- Both feature cards must be inserted and active for all of the configuration tasks described in this appendix. However, the bond3 (public) network interface must be inactive on both feature cards. Do not insert both feature cards simultaneously unless you have deactivated the bond3 network. To do this, enter the following command on each feature card separately:

  ```
  ./featureCard_bond3.sh  -d
  ```

Configuring the FCAT Environment

Perform the following tasks to configure the FCAT environment:

> **Note:**
>
> In the steps below that require you to manually edit specific configuration files, please make a restorable copy of the original.

- Step 1: Install and Configure ACSLS on Both Feature Cards
- Step 2: Sync Time Between Feature Cards and the SL4000
- Step 3: Extract and Install the Feature Card Availability Toolkit
- Step 4: Shut Down ACSLS on Both Feature Cards
- Step 5: Cluster the Feature Cards and Establish Trust
- Step 6: Copy Select Feature Card Scripts to a Separate FCAT Linux Client
- Step 7: Remove Existing Trusts Between the FCAT Client and Feature Cards
- Step 8: Configure the FCAT Client to Recognize Both Side-A and Side-B Feature Cards
- Step 9: Monitor ACSLS Availability from the FCAT Client
- Step 10: Enable ACSLS on the Side-A feature Card
- Step 11: Trigger Failover from One Feature Card to the Other
- Step 12: Start ACSLS (with optional Database Restore)

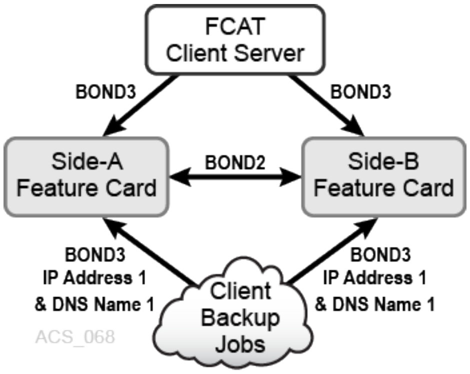
Step 1: Install and Configure ACSLS on Both Feature Cards

To begin, choose one of the following configuration options, based on whether you want to configure both feature cards with the same IP Address and DNS name, or with two different IP addresses and DNS names:

- Option 1

  In this option, both feature cards use the same IP address and DNS name, as shown in the following figure:

Figure C-1 FCAT Configuration Option 1



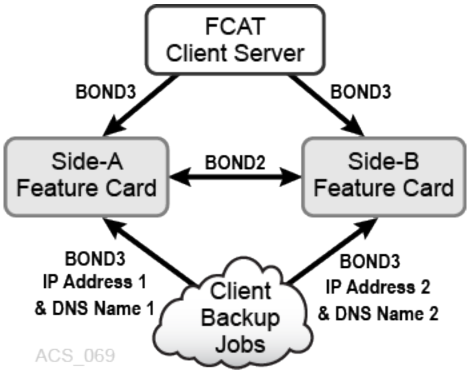
For clients attached to ACSLS through the feature card, all client addressing uses the same IP address and DNS name, regardless of which feature card you are using at a given time. bond2 is the network interface used only for internal library communications. bond3 is the customer network interface used to communicate with the feature cards.

This is the default option.

- Option 2

  In this option, each feature card uses a different IP address and DNS name, as shown in the following figure:

Figure C-2 FCAT Configuration Option 2



For clients attached to ACSLS through the feature card, client addressing uses distinct IP addresses and DNS names for each feature card. This option allows you to switch connected clients to run against a specific instance of ACSLS. bond2 is the network interface used only for internal library communications. bond3 is the customer network interface used to communicate with the feature cards.

Once you have chosen your option, install and configure ACSLS on each feature card as described in Installing ACSLS on the SL4000 Feature Card .

> **WARNING:**
>
> When installing ACSLS, ensure that you have configured both feature card platforms and installed instances of ACSLS identically. Otherwise, ACSLS failover, startup, and other operations may not function properly.

Step 2: Sync Time Between Feature Cards and the SL4000

As noted in the *ACSLS Administrator's Guide*, for proper operation of ACSLS with the SL4000, the ACSLS server clock must be in sync with the SL4000 internal clock. This is important on a standalone ACSLS server running against the SL4000 as well as when using ACSLS on the SL4000 feature cards. This synchronization ensures that database backups are current and correct.

Synchronize the following:

- The time between both feature cards and the SL4000.
- The time between both feature cards.

  If you do not already have access to an NTP (Network Time Protocol) server, establish one to keep the information replicated between the two feature cards in sync.

Step 3: Extract and Install the Feature Card Availability Toolkit

The Feature Card Availability Tool kit is bundled with the initial feature card scripts extracted during feature card installation, as described in Configuring the Feature Card and Preparing for ACSLS Installation.

The following FCAT scripts are included:

- `fcatServer_clusterizeFeatureCards.sh`
- `fcatServer_featureCardStatusPayload.sh`
- `fcatServer_switchActivePassiveCard.sh`
- `fcatClient_featureCardStatus.sh`
- `fcatClient_triggerSwitch.sh`

Ensure that these scripts are available under `/usr/local/bin` on both the Side-A and Side-B feature card, and reload them if necessary.

- The Side-A feature card is installed in the Feature Card Kit 1 location.
- The Side-B feature card is installed in the Feature Card Kit 2 location.

See Feature Card Locations for more information.

Step 4: Shut Down ACSLS on Both Feature Cards

To shut down the ACSLS application on both the Side-A and Side-B feature card:

1. Log in to the Side-A feature card as user `acsss` and enter the following command to shut down the ACSLS application:

```
cd $ACS_HOME/bin
cmd_proc_shell idle
acsss shutdown
```

2. Verify that ACSLS is completely shut down.

   ```
   acsss status
   ```

3. Repeat this procedure for the Side-B feature card.

Step 5: Cluster the Feature Cards and Establish Trust

You must cluster the feature cards to establish trust between them. This enables ACSLS to keep both feature cards current with their database updates and ensures that ACSLS can be quickly resumed.

To establish trust between both feature cards, you must establish trust from the Side-A to Side-B feature card, and from the Side-B feature card to Side-A feature card, as follows:

1. Establish Trust from Side-A to Side-B feature card for user `acsss`.

   a. On the Side-A feature card, as user `acsss`, access the directory `/usr/local/bin`:

      ```
      cd /usr/local/bin
      ```

   b. Run the following command:

      ```
      fcatServer_clusterizeFeatureCards.sh
      ```

      The following message appears:

      ```
      =============================================================Trust between two
      feature cards plugged in the same libraryform the backbone for ACSLS
      availability solution on featurecards as it enables the exchange of data and
      files betweenthose two feature cards.ACSLS availability solution for feature
      card is *not*possible without trust between the two feature
      cards.=============================================================Do you want
      to create the trust [acsss:Side-A => Side-B](yes/no)?
      ```

   c. At the prompt, enter `yes` to create the trust.

   d. At the password prompt, enter the password for user `acsss` on the Side-B feature card.
      Trust is established and a confirmation message appears:

      ```
      root: fcatServer_clusterizeFeatureCards.sh: INFO: A simplex trust
      [acsss : Side-A => Side-B] is created successfully.

      =============================================================A simplex trust
      is created between feature cards from Side-Ato Side-B for 'acsss'. In order to
      break this trust, executerunuser -l acsss -c
      'fcatServer_clusterizeFeatureCards.sh -d'on <server_name> from command
      line.=============================================================
      ```

   e. As user `acsss`, exit the session:

      ```
      exit
      ```

2. Establish trust from the Side-A to Side-B feature card for user `root`.

   a. On the Side-A feature card, as user `root`, access the directory `/usr/local/bin`:

```
cd /usr/local/bin
```

b. Run the following command:

```
fcatServer_clusterizeFeatureCards.sh
```

The following message appears:

```
============================================================Trust
between two feature cards plugged in the same libraryform the backbone
for ACSLS availability solution on featurecards as it enables the
exchange of data and files betweenthose two feature cards.ACSLS
availability solution for feature card is *not*possible without trust
between the two feature
cards.============================================================
Do you want to create the trust [root:Side-A => Side-B](yes/no)?
```

c. At the prompt, enter `yes` to create the trust.

d. At the password prompt, enter the password for user `root` on the Side-B feature card.
Trust is established and a confirmation message appears:

```
root: fcatServer_clusterizeFeatureCards.sh: INFO: A simplex trust
[root : Side-A => Side-B] is created successfully.

============================================================A simplex
trust is created between feature cards from Side-Ato Side-B for 'root'.
In order to break this trust, executerunuser -l root -c
'fcatServer_clusterizeFeatureCards.sh -d'on <server_name> from command
line.============================================================
```

e. As user `root`, exit the session:

```
exit
```

3. Repeat steps 1 and 2 from the Side-B feature card to establish trust from the Side-B to Side-A feature card for both user `acsss` and user `root`.
Once trust is fully established, information can flow between the two feature cards and you can use the ACSLS Feature Card Availability Toolkit tools to monitor the feature cards for any issues that may be encountered.

Step 6: Copy Select Feature Card Scripts to a Separate FCAT Linux Client

To enable the ability to monitor ACSLS and trigger a switch from one feature card to another, copy the following FCAT scripts to a dedicated Linux Client server (Client host):

- `fcatClient_featureCardStatus.sh`
- `fcatClient_triggerSwitch.sh`

These scripts can be copied from the `/usr/local/bin` directory on either the Side-A or Side-B feature card, whichever has an enabled bond3 IP address. Although these client scripts are agnostic to the directory in which they will reside on the client host, it is recommended that you dedicate a directory for these scripts.

When you are finished, verify that the files are listed in the destination directory.

Step 7: Remove Existing Trusts Between the FCAT Client and Feature Cards

In preparation for configuring the FCAT client to recognize your feature cards, you must first ensure that any older, existing trusts are removed.

1. On the FCAT client, as user `root`, open the following file in the text editor of your choice:

   ```
   /root/.ssh/known_hosts
   ```

   This file includes entries for various known hosts. For example:

   ```
   <FC_HOSTNAME>,<IP_BOND3> ssh-rsa
   AAB3NzaC1yc2EAABIwAAQEA3EMv/fPWJoa9ZAVWYrdr5yfs5N2G/AsBSN/Mu/GI79KFELw
   6qfFCxagQaf7f/w0taer+Rzbovog3Tp2NGikdstdCX02/ucpcDbpp2CNcF8imnEsL5H76I
   1y8CMEQ1t3xDNZz5WXuPeCDT17Nq3KXtRt7CO0iNgPQhQB210jG02S/Nt9AJK7xiaTh8OM
   FwiaBrCowQugCGPHanZo7NP1X9ZT1VP5RGnqIyfYyZSDZzkUBS73GxGcGiEmARS0BODjFS
   kKrqOKpdhc/Z7EYsw==
   ```

   Where:

   - `<FC_HOSTNAME>` is the feature card name for Side-A or Side-B.
   - `<IP_BOND3>` is the feature card IP address for Side-A or Side-B.

2. Review this file to locate any entries associated with the two feature cards. Delete these outdated entries for IP addresses and associated host names.
3. Save your changes and close the file.

Step 8: Configure the FCAT Client to Recognize Both Side-A and Side-B Feature Cards

Complete the following steps to establish Side-A and Side-B feature card keys in the `known_hosts` file. This allows net operations between the platforms.

1. Proceed according to your chosen configuration option:

   - If you are using Option 1, using one IP address and DNS name for both feature cards, proceed with step 2.
   - If you are using Option 2, with distinct IP addresses and different DNS names for each feature card, skip to Step 5.

2. Ensure that the Side-B feature card is disabled on the bond3 public network. Use the following commands if necessary:

   ```
   cd /usr/local/bin
   featureCard_bond3.sh -d
   ```

3. Ensure that the Side-A feature card is enabled on the bond3 public network. Use the following commands if necessary:

   ```
   cd /usr/local/bin
   featureCard_bond3.sh -e
   ```

4. Perform the following steps to update the `known_hosts` file on your FCAT Client, adding the feature card IP address and DNS name for the Side-A feature card.

a. On the FCAT Client, as user `root`, run the following command using the `FC_HOSTNAME` for the Side-A feature card:

```
ssh <FC_HOSTNAME>
```

The following message appears:

```
The authenticity of host '<FC_HOSTNAME> (FC_BOND3_IP)' can't be
established.
RSA key fingerprint is 85:44:72:86:3f:e1:6d:44:42:8c:6d:31:5d:b4:97:5c.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '<FC_HOSTNAME>, <IP_BOND3_IP>' (RSA) to the
list of known hosts.

root@<FC_HOSTNAME>'s password:
```

b. At the prompt, enter `yes` to continue the connection.
A password prompt appears.

c. At the password prompt, enter the `root` user password for the feature card you are attempting to connect to.
A Login message appears, indicating that you are logged into the targeted feature card.

d. Exit the session to return to your FCAT Client. The key for the Side-A feature card is now in the `/root/.ssh/known_hosts` file.

5. You have now successfully established the key for the Side-A feature card. If you are using an additional (Side-B) feature card, complete the following steps:

   a. On the FCAT client, as user `root`, open the following file in the text editor of your choice:
   `/root/.ssh/known_hosts`

   This file includes entries for various known hosts.

   b. Edit the file, placing a `#` sign at the beginning of the entry for the Side-A feature card to comment out this entry.

   c. Save your changes and close the file.

   d. Return to step 1 of this procedure and perform steps 1-4 to establish the key for the Side-B feature card, just as you did for Side-A.

   e. Update the `known_hosts` file again by removing the `#` sign to make the Side-A feature card active.

   f. On the Side-A feature card, as user `root`, enable the bond3 public network:

```
cd /usr/local/bin
featureCard_bond3.sh -e
```

   g. On the Side-B feature card, as user `root`, disable the bond3 public network:

```
cd /usr/local/bin
featureCard_bond3.sh -d
```

6. You have now successfully established the keys for both Side-A and Side-B feature cards, allowing the FCAT Client to recognize them.

Step 9: Monitor ACSLS Availability from the FCAT Client

To monitor the status for both feature cards:

1. On the FCAT Client, locate the directory where you copied the following files:
   - `fcatClient_featureCardStatus.sh`
   - `fcatClient_triggerSwitch.sh`
2. Run one of the following commands:
   - If you are using Option 1, with both feature cards using the same IP address and DNS name, enter one of the following commands:

     ```
     fcatClient_featureCardStatus.sh <HN-1>

     fcatClient_featureCardStatus.sh <IP_BOND3>
     ```

     where `HN-1` is the host name used for both feature cards.
   - If you are using Option 2, with each feature card using a distinct IP address and DNS name, enter one of the following commands:

     ```
     fcatClient_featureCardStatus.sh <HN-1> [<HN-2>]

     fcatClient_featureCardStatus.sh <IP_BOND3_AorB> [<IP_BOND3_AorB>]
     ```

     where:
     - `HN-1` and `HN-2` are the distinct host names for your feature cards.
     - AorB indicates the feature card location, Side-A or Side-B, as appropriate.
3. You may be prompted to enter a password if the trust between the FCAT Client and feature cards have been established, or conditions for the previous trust have changed.
4. Once the `fcatClient_featureCardStatus.sh` command is processed, monitoring is enabled for ACSLS on both feature cards. The monitoring screen is refreshed every 30-35 seconds, and displays the following:
   - Active or passive status for each feature card
   - Whether trust is established and active (Peer status is `Reachable`).
   - Payload ID (increments with every screen refresh)
   - Additional information, including the status of the ACSLS application

The following is an example of the ACSLS Availability Status screen at initial startup:

```
=======================================================================

                ACSLS Availability Status

Payload ID      : 1
Request Time    : Wed May 22 13:46:20 MDT 2019
Report Time     : Wed May 22 13:46:22 MDT 2019

*** Feature cards need to synchronize their
clocks.----------------------------------------------------------------------

Side                : Side-A
Node                : FC_hostname_A [ <IP_BOND3_A> ]
```

```
Availability Status : Active since 2019-05-22 19:33:14
Peer Status (root)  : Reachable [ EXTERNAL ]
Peer Status (acsss) : Reachable [ EXTERNAL ]
Hardware Status     : up 2:08, 6 users, load average: 0.00, 0.02, 0.05
ACSLS Uptime        : 0-00:00:00 <days-hrs:min:sec>

   rmi-registry is offline
   surrogate is offline
   acsdb is offline
   acsls is offline

----------------------------------------------------------------------
----------------------------------------------------------------------

Side                : Side-B
Node                : FC_hostname_B [ <IP_BOND3_B> ]
Availability Status : Passive
Peer Status (root)  : Reachable [ EXTERNAL ]
Peer Status (acsss) : Reachable [ EXTERNAL ]
Hardware Status     : up 1:51, 3 users, load average: 0.08, 0.03, 0.05
ACSLS Uptime        : 0-00:00:00 <days-hrs:min:sec>

   rmi-registry is offline
   surrogate is offline
   acsdb is offline
   acsls is offline

----------------------------------------------------------------------

======================================================================
```

This display is based on Option 2, with each feature card using a distinct IP address and different DNS name.

- Status is displayed for both feature cards, as they are both available.
- `[EXTERNAL]` on the Peer status lines indicates Option 2. `[INTERNAL]` would indicate Option 1.
- ACSLS is offline as it is not yet enabled.
- The display alerts you that the clocks must be synchronized between both feature cards.

Step 10: Enable ACSLS on the Side-A feature Card

On the Side-A feature card, enter the following command to enable ACSLS:

```
acsss enable
```

Only one instance of ACSLS on the feature card can be enabled at one time.

Step 11: Trigger Failover from One Feature Card to the Other

In the event that failover is required, you must trigger a switch between the two feature cards from the FCAT Client. The active feature card becomes passive and the passive feature card becomes active.

Note:

- Before triggering a switch, quiesce client jobs and other operations to ensure a smooth transition to the newly active feature card.
- While the switchover activates the other card, it does not automatically activate ACSLS on that card. You will need to do so manually after the card switchover has occurred.

To trigger a feature card switch:

1. On the FCAT Client, locate the directory where you copied the `fcatClient_triggerSwitch.sh` file:
2. Run one of the following commands:

   ```
   fcatClient_triggerSwitch.sh <HN>
   fcatClient_triggerSwitch.sh <IP_BOND3>
   ```

   where `HN` is the host name of the feature card to perform the switchover.
3. You may be prompted to enter a password if the trust between the FCAT Client and feature cards have been established, or conditions for the previous trust have changed.
4. Once the `fcatClient_triggerSwitch.sh` command is entered, an updated ACSLS Availability Status screen appears. This screen is refreshed every 30-35 seconds.

   Switch processing may take several minutes. The following message:

   ```
   Recovery (PID=number) is in progress...
   ```

   indicates that the switch is still in process. This screen continues to refresh every 30-35 seconds until the switch is complete. PID is the process ID number.

   Once the feature card switchover is complete, the Status screen is updated.

   - The previously active feature card is now passive and is disabled (offline).
   - The previously passive feature card is now active. ACSLS can now be manually enabled on that card.

   For example:

   ```
   =======================================================================

                   ACSLS Availability Status

   Payload ID      : 55
   Request Time    : Wed May 22 14:59:46 MDT 2019
   Report Time     : Wed May 22 14:59:48 MDT 2019

   ----------------------------------------------------------------------

   Side                : Side-A
   Node                : FC_hostname_A [ <IP_bond3_A> ]
   Availability Status : Passive
   Peer Status (root)  : Reachable [ INTERNAL ]
   Peer Status (acsss) : Reachable [ INTERNAL ]
   Hardware Status     : up 3:22, 8 users, load average: 0.10, 0.26, 0.16
   ACSLS Uptime        : 04:19 <days-hrs:min:sec>
   ```

```
   rmi-registry is offline
   surrogate is offline
   acsdb is online
   acsls is offline

----------------------------------------------------------------------
----------------------------------------------------------------------

Side                : Side-B
Node                : FC_hostname_B [ <IP_bond3_B> ]
Availability Status : Active since 2019-05-22 20:41:11
Peer Status (root)  : Reachable [ INTERNAL ]
Peer Status (acsss) : Reachable [ INTERNAL ]
Hardware Status     : up 3:04, 4 users, load average: 0.08, 0.03, 0.05
ACSLS Uptime        : 0-00:00:00 <days-hrs:min:sec>

   Recovery (PID=12698) is in progress. Please try later...

----------------------------------------------------------------------

=======================================================================
```

Note:

- In this example, there was no true hardware failure. Therefore, status is displayed for both feature cards. If a true hardware failure occurs and a feature card is unreachable, you may see an error or incomplete status screen.
- If the time delta between the two feature cards is significant, a warning message is displayed and the two feature cards must be resynchronized.

Step 12: Start ACSLS (with optional Database Restore)

After completing failover from one card to the other, ACSLS must be restarted manually on the newly active card. You can choose to do a database restore before starting ACSLS, or simply start ACSLS and it will use the last automated snapshot, which is synchronized between the paired Feature Cards automatically.

To simply start ACSLS and use the latest database snapshot, on the newly active feature card, perform the following steps:

1. Ensure ACSLS and the database are in a known and synchronized state by shutting down:

   ```
   su - acsss
   acsss shutdown
   ```

2. Start ACSLS for continued operation:

   ```
   acsss enable
   ```

If you wish to restore another database (other than the one synchronized between the Feature Cards), perform the following steps:

1. As user acsss, restore the ACSLS database from the most recent backup or an accessible backup location:

   Run db_restore.sh:

   ```
   su - acsss
   cd /bin
   ./db_restore.sh latest
   or
   ./db_restore.sh/bkupb/<DB_SNAPSHOT_FILENAME>
   ```

   Substitute the specific file name of your backup for *latest* and <DB_SNAPSHOT_FILENAME>.

   Redundant copies of recent backup files reside under /bkupa, /bkupb or /bkupc. If no ACSLS database backups are available to you locally, you may want to pull the latest database backup from your remote backup server. If you do not want to use an existing database backup, and would rather create a new one, then run accss_config instead.

2. Enable ACSLS as user acsss:

   ```
   su - acsss
   acsss enable
   ```

   Once ACSLS is enabled, you can set your client jobs to run and resume normal operations.

D

Configuring a Self-Signed Digital Certificate for HTTPS

This appendix explains how to create a custom SSL encryption certificate for the AcslsDomain in your WebLogic server. This procedure is required if you intend to create a self-signed digital certificate for use with browsers that do not accept the demo certificate provided by default with the ACSLS GUI.

Internet Explorer 8 (and above) and FireFox Version 39 (and above) requires this WebLogic set-up procedure for use with HTTPS servers that do not employ certificates verified by a third-party digital signing authority.

1. Generate a keystore database of cryptographic keys.

   a. As `root` user, source the basic `acsls` environmental variables.

   ```
   . /var/tmp/acsls/.acsls_env
   ```

   b. Define keyStore parameters:

   ```
   keyPath=$installDir/Oracle/Middleware/wlserver_10.3/server/lib
   identStore=acslsIdent.jks
   trustStore=acslsTrust.jks
   keyPass=<password>
   storPass=<password>
   ```

   c. Generate the public/private encryption key pair and digital certificate. Place them in the keyStore.

   ```
   keytool -genkeypair -alias selfsigned -keyalg RSA -keysize 2048 \
   -validity 365 -keypass $keyPass -storepass $storPass \
   -keystore $keyPath/$identStore
   ```

   This produces a certificate valid for 365 days with encryption key that is 2048 bits in length. The keytool prompts you with the following questions. The answers you give are written to a certificate that can be displayed on a remote browser any time the ACSLS GUI user is asked to confirm the authenticity of the HTTPS connection.

   ```
   What is your first and last name?
   [Unknown]:  ACSLS Library Server

   What is the name of your organizational unit?
   [Unknown]:  Tape Library Services

   What is the name of your organization?
   [Unknown]:  Our Organization

   What is the name of your City or Locality?
   [Unknown]:  Our Town

   What is the name of your State or Province?
   [Unknown]:  Our Province?

   What is the two-letter country code for this unit?
   [Unknown]:  XY
   ```

When prompted for a password, click **Return** to use the value for $identPass that you set in step 1b.

The tool summarizes the parameters you submitted and asks you to confirm (**yes**/**no**) that the parameters are correct.

d. Export the ident certificate and import it to the trust certificate.

```
keytool -exportcert -alias selfsigned -file $keyPath/root.cer \
-keystore $keyPath/$identStore -storepass $storPass

keytool -importcert -alias selfsigned -file $keyPath/root.cer \
-keystore  $keyPath/$trustStore -storepass $storPass
```

Answer **yes** to the prompt to confirm.

e. Copy the files, $keyPath/acslsIdent.jks and $keyPath/acslsTrust.jks, to the $SSLM_HOME/AcslsDomain/ directory.

2. Configure WebLogic to use the newly-generated keyStore.

   a. Logon to the WebLogic console as acsls_admin using the acsls_admin password.
   http://<acsls_server>:7001/console

   b. From the main page top-left corner of the console page, click **Lock & Edit**.

   c. Just below the Lock and Edit button, you see 'Domain Structure'. Select **Environment** under the AcslsDomain.

   d. From the Summary of Environment frame, click **Servers**.

   e. From the Summary of Servers frame, select the Configuration tab and click **AdminServer(admin)** from the Servers table.

   f. From the Settings for AdminServer frame, select the **Keystores** tab.

   g. Under the Keystores tab, click **Change** and select **Custom Identity and Custom Trust**. Click **Save**.

   h. In the Custom Identity Keystore text box, enter the path to the acslsIdent.jks file using the $keyPath/$identStore values that you defined in step 1b above. In the Custom Identity Keystore Type box, enter **jks**.

   i. In the Custom Identity Keystore Passphrase text box, enter the password that you defined as $storPass in step 1-b above. Confirm the Custom Identity Keystore Passphrase in the next text box.

   j. In the Custom Trust Keystore text box, enter the full path to the acslsTrust.jks file using the $keyPath/$trustStore values that you defined in step 1-b. In the Custom Trust Keystore Type text box, enter **jks**.

   k. In the Custom Trust Keystore Passphrase text box, enter the password you defined for $storPass in step 1-b. Enter confirmation of that password in the remaining text box.

   l. Click **Save**. Observe the verification message at the top of the page.

   m. Select **SSL** tab in the Settings for Administrator frame.

   n. In Identity and Trust Locations ensure that **Keystores** is selected. If necessary, click **Change** to correct the setting.

   o. In the Private Key Alias text box, enter **selfsigned**.

p. In the Private Key Passphrase text box, enter the same password you defined as `$keyPass` in step 1-b above. Confirm it using the same password in the remaining text box.

q. Click **Save**. Look for the green verification message at the top of the page.

r. Click the **Advanced** field under the SSL tab. Set Hostname Verification to **none**. Select the check box for **Use JSEE SSL**.

s. Click **Save**. Look for the green verification message at the top of the page.

t. Click **Activate Changes** in the top-left corner of the page. Observe the verification message at the top of the page.

u. Restart the `weblogic` service.

E

Maintenance and Troubleshooting Tasks for ACSLS on the Feature Card

This appendix provides maintenance and troubleshooting information for configurations with ACSLS on the SL4000 feature card.

Topics include:

- General Feature Card Maintenance and Troubleshooting Tasks
- FCAT Related Maintenance and Troubleshooting Tasks

> **Note:**
>
> All hardware-related tasks described in this appendix are to be performed by Oracle Support. These tasks are combined with ACSLS software recovery tasks, which can be performed by Oracle Support or by the customer, in conjunction with Oracle Support.

General Feature Card Maintenance and Troubleshooting Tasks

The following tasks apply to configurations with ACSLS on the feature card.

Performing a Soft Boot of the Feature Card

In certain situations, you may be required to perform a soft reboot of the feature card. If possible, shut down the applications running on the card before performing a soft boot.

To perform the soft boot, log in as user `root` and issue the following command:

```
reboot -f
```

Monitoring the Feature Card

To monitor your feature card hardware performance, follow the directions outlined for viewing the SL4000 GUI **Hardware** tab. Refer to the *SL40000 Library Guide* for more information.

Updating Your IP Address, Host Name, or Other Network Related Items

If you need to change your IP address, host name, or other network-related information for the feature card, complete the following steps described in Configuring the Feature Card and Preparing for ACSLS Installation.

See:

- Step 5: Configure the Feature Card for Bond3 Network Connectivity

- Step 6: Configure the Feature Card with DNS Servers and Other Required Settings
- Step 7: Configure the Feature Card Host Name

Not Enough Disk Space in the root Partition on the Feature Card

Depending on where you store your ACSLS diagnostic bundles, you may run out of disk space in the root partition on the feature card. You may need to manage this space to maintain a reasonable size or create a soft link to point to the logs elsewhere. The following is an example of the commands you may perform as user root to enable more space, if required:

```
mkdir /u01/tmp
ln -s /u01/tmp/diags /tmp/diags
chmod 777 /u01/tmp /tmp/diags /u01/tmp/diags
```

Establishing Mount Points after Removing and Re-inserting a Feature Storage Card

When you remove and re-insert a feature storage card in the SL4000 library, a new mount point is created and the previous mount point may be retained. To eliminate confusion, use one of the following methods to remove the previous mount point if it is also displayed:

- Automatic unmount (preferred)

  Reboot the impacted feature card following the procedure described in "Performing a Soft Boot of the Feature Card".

- Manual unmount

  1. Manually unmount the devices associated with '/bkupa' and '/bkupb' using the following operating system command:

     ```
     umount <filesystemName>
     ```

     where <filesystemName> is the /bkupa or /bkupb name.

  2. Run the following script:

     ```
     /usr/local/bin/featureCard_acslsStorageManager.sh
     ```

Troubleshooting and Recovering from Unexpected ACSLS Storage Problems

If you encounter issues where disk layout or mount points have been lost, you may need to repair the file system and storage setup for ACSLS. Depending on the issue, you may need to complete the following steps one or more times to return the system to a normal state:

1. Before running any of the commands detailed in the following steps, log in as user acsss and enter the following command to shut down the ACSLS application:

   ```
   acsss shutdown
   ```

2. You may be required to rerun `featureCard_acslsStorageManager.sh` as user `root`. This command assumes that you have shut down the ACSLS application. To repair ACSLS storage-related issues, issue the following command:

   ```
   cd /usr/local/bin./featureCard_acslsStorageManager.sh
   ```

3. A soft reboot may be required as part of the command sequence above, to ensure that all settings have been updated correctly. To do this, log in as user `root` and issue the following command:

   ```
   reboot
   ```

4. An issue may require you to mirror RAID storage. Monitor the mirroring process and ensure that it has been completed before continuing to enable ACSLS. As user `root`, you can use the following command to monitor the rebuilding of the mirror.

   Open a separate terminal window and enter:

   ```
   watch cat /proc/mdstat
   ```

   Monitor the `mdstat` display to determine when rebuilding of the mirrored drive is complete. Within minutes, `[UU]` is displayed, indicating that the rebuilding operation has completed.After `[UU]` is displayed, you can proceed with enabling ACSLS.

FCAT Related Maintenance and Troubleshooting Tasks

The following tasks apply to configurations using the ACSLS Feature Card Availability Toolkit (FCAT) in a dual feature card configuration with ACSLS 8.5.1.

Breaking Trust Between Two Feature Cards

Complete the following steps:

1. Enter the following command:

   ```
   fcatServer_clusterizeFeatureCards.sh -d
   ```

   The following message is displayed:

   ```
   ==============================================================

   Trust between two feature cards plugged in the same library
   form the backbone for ACSLS availability solution on
   feature cards as it enables the exchange of data and files between
   those two feature cards.

   ACSLS availability solution for feature card is *not*
   possible without trust between the two feature cards.

   ==============================================================
   Do you want to break the trust [root : Side-A => Side-B] ? [Y/N] y
   ```

2. At the prompt, enter `Y`.

   The following message is displayed:

   ```
   root: fcatServer_clusterizeFeatureCards.sh: INFO: A simplex trust [root : Side-A
   => Side-B] is destroyed.
   ```

To ensure that all trusts are removed, run the above command on both Side-A and Side-B feature cards, first as user `acsss` and then as user `root`.

Breaking Trust Between the FCAT Client and Feature Cards

Reference the configuration tasks as described in "Configuring the FCAT Environment":

See "Step 7: Remove Existing Trusts Between the FCAT Client and Feature Cards".

Updating Passwords Associated with Feature Cards

Complete the following steps:

1. Login to the Side-A feature card as user `acsss` and shut down the feature card:

   ```
   acsss shutdown
   ```

2. Repeat the above step for the Side-B feature card.
3. Remove existing trust between the FCAT Client and feature cards, as described in Using the ACSLS Feature Card Availability Toolkit:

   See Step 7: Remove Existing Trusts Between the FCAT Client and Feature Cards.

4. Remove existing trust between the two feature cards for both user `acsss` and user `root`, as described in Breaking Trust Between Two Feature Cards.
5. Perform your password changes.
6. Re-create trust between the feature cards, as described in Using the ACSLS Feature Card Availability Toolkit.

   See Step 5: Cluster the Feature Cards and Establish Trust.

7. Re-create trust between the FCAT client and feature cards, as described in Using the ACSLS Feature Card Availability Toolkit:

   See Step 8: Configure the FCAT Client to Recognize Both Side-A and Side-B Feature Cards.

8. Enable ACSLS as user `acsss` to begin operations:

   ```
   acsss enable
   ```

Verifying that Trusts are Established Between Feature Cards

Use the `fcatClient_featureCardStatus.sh` command to verify that trusts are established between the two feature cards installed in the upper two LOC card slots within the same SL4000 library.

If one of the feature cards indicates Unreachable for a given user, then attempt to establish trust again using the `fcatServer_clusterizeFeatureCards.sh` utility.

If you cannot view both feature cards on the ACSLS Availability Status screen, then verify that you have correctly established trusts in your `/root/.ssh/known_hosts` file.

Verifying that Database Snapshots are Successfully Replicated

When the FCAT tools are properly configured, the database is automatically replicated between the two feature cards and all four feature storage cards when an ACSLS change results in a database update.

To verify that the backup files and all four feature storage cards are in sync, review the directory `/export/backup/` and look for files in the form `2019-05-22_18:13:31.tar`. You should see the same set of files for each feature card. Copies are also located on all of the feature storage cards.

Note:

Ensure that the current time for both feature cards is in sync. See Step 2: Sync Time Between Feature Cards and the SL4000.

Troubleshooting Feature Card Tools

All of the `fcatClient` scripts are certified to run on the OBI Linux release 6.8, 6.10, 7.3, 7.6, 7.8 and 7.9 platforms. Other versions of Unix may require standard porting methods.

F

Servicing the Feature Card and Recovering ACSLS

This appendix provides procedures used to service the SL4000 feature card and recover the ACSLS application running on the feature card.

Topics include:

- Performing a Soft Boot of the Feature Card
- Monitoring the Feature Card
- Shutting Down the Feature Card and ACSLS for Service
- Replacing a Faulty Feature Card
- Replacing a Single Feature Storage Card
- Replacing Two Feature Storage Cards
- Upgrading ACSLS on the SL4000 Feature Card

> **Note:**
>
> All hardware-related tasks described in this appendix are to be performed by Oracle Support. These tasks are combined with ACSLS software recovery tasks, which can be performed by Oracle Support or by the customer, in conjunction with Oracle Support.

Performing a Soft Boot of the Feature Card

In certain situations, you may be required to perform a soft reboot of the feature card. If possible, shut down the applications running on the card before performing a soft boot.

To perform the soft boot, log in as user `root` and issue the following command:

```
reboot -f
```

Monitoring the Feature Card

To monitor your feature card hardware performance, follow the directions outlined for viewing the SL4000 GUI **Hardware** tab. Refer to the *SL40000 Library Guide* for more information.

Shutting Down the Feature Card and ACSLS for Service

When it is necessary to service the SL4000 library or its components, you must shut down the card and any applications running on it. Once the card is rebooted, you can then restart the applications. Example commands follow:

1. Log in as user `acsss` and enter the following command to shut down the ACSLS application.

   ```
   acsss shutdown
   ```

2. Log in as user `root` and enter the following command to shut down the feature card.

   ```
   shutdown -h now
   ```

3. If hardware service is required, physically remove the feature card, perform required service, and then re-insert the feature card in its original slot to power it up. As a reminder, once a feature card is initially inserted in a specific slot within the SL4000 Base Card Cage, you must always replace it in the same slot.
4. Start the ACSLS software.

   ```
   acsss enable
   ```

Replacing a Faulty Feature Card

> **Note:**
>
> This procedure applies to a faulty feature card. Do not use this method to upgrade the feature card or the ACSLS application.

To replace a faulty feature card with ACSLS, perform the following steps:

1. If the hardware is up and ACSLS is running on the feature card, you must shut down ACSLS. Log in as user `acsss` and enter the following command to shut down the ACSLS application:

   ```
   acsss shutdown
   ```

2. If the feature card is up and running, you must shut down the feature card. Log in as user `root` and enter the following command to shut down the feature card and safely remove it:

   ```
   shutdown -h now
   ```

3. Physically remove the faulty feature card and unseat the associated feature storage cards.
4. Use one of the following options to initialize the replacement feature card:
   - Place the new replacement feature card in the slot that the faulty feature card was removed from, and then boot it. Then insert the feature storage cards.
   - Place the new replacement feature card into the previously unused slot and boot it. Then insert the feature storage cards and perform a soft boot. You must remove the feature storage cards from their locations above the original feature card slot and place them directly into the new feature storage card slots, directly above the new feature card slot. Then proceed with the ACSLS setup and installation process as noted below.

> **Note:**
>
> Apply new external labels to the feature card and feature storage cards. Include appropriate names, to represent changes made.

You have now effectively initialized the feature card, just as you did in the initial installation of the feature card.

5. Log in as user `root` and verify this initialization as described in Installing ACSLS on the SL4000 Feature Card :
   See "Step 3: Verify Feature Card Initialization".

6. Connect your equipment and establish a network connection as described in Installing ACSLS on the SL4000 Feature Card :
   See:

   - "Step 1: Connect User Equipment"
   - "Step 2: Establish a Temporary Connection to the External Network"

7. Download the ACSLS bundle as described in Installing ACSLS on the SL4000 Feature Card :
   See "Step 3: Download the ACSLS Software Bundle".

   If there was a previously installed and running version of ACSLS on the feature card, you can use an alternative approach to download the ACSLS 8.5 software bundle from one of the current feature storage cards, assuming they are running and accessible. To use this approach, perform the following steps:

   a. As user `root`, run the command `lsblk -f` to determine the Linux device name of one of the two feature storage cards. Output should appear similar to the following:

   ```
   # lsblk -f
   NAME FSTYPE LABEL UUID MOUNTPOINT
   sdX
   sdY
   sdZ
   sdZ1 ext4   Aboot f0bf63fa-3460-40dd-9a91-83022631277f /boot
   sdZ2 ext4   Au01  5f73feature cardb5-627d-4ce2-afeature card6-11a16d7310f0 /u01
   sdZ3 ext4   Aslash 60aa9dd0-e629-4f40-8e03-74b6db53d3dd /
   sdZ4
   sdZ5 swap   Aswap  65f20091-01a7-49b6-b212-985a6ebcf549 [SWAP]
   sdZ6 ext4   Avar   6036e949-5403-48db-a522-035db9f60e88 /var
   #
   ```

   > **Note:**
   >
   > - `sdX`, `sdY`, and `sdZ`, may be `sda`, `sdb`, `sdc`, `sdd`, or some other name, as determined by Linux. `sdZ` refers to the SSD on the feature card; and `/bkupc` is also located on the SSD. `sdX` and `sdY` refer to the hard drives on the feature storage cards.
   > - `mdN` may be `md1` or `md2`, depending upon whether you have installed the feature card into the left side (`md1`) or the right side (`md2`).

b. Identify the Linux device names for the two feature storage cards in the listing from `lsblk`. Select one of these to be temporarily mounted, from which you can access the needed ACSLS software bundle. Assume one of these is named `sdX1`. As user `root`, mount the feature storage card device to a temporary mount point and copy the previously downloaded version of ACSLS as follows:

```
mkdir /tmpMount
mount /dev/sdX1 /tmpMount
cp /tmpMount/ACSLS_8.5.1-X.Y.Linux.zip /tmp
umount /tmpMount
rmdir /tmpMount
```

c. If you were able to access a viable copy of the ACSLS software bundle, you can now find it in `/tmp`. If a viable copy was not available at `/dev/sdX1`, try this same command sequence with the other feature storage card device named `/dev/sdY1`. If no viable copy was found on all available feature storage cards, then you will need to use one of the methods described earlier (via browser and eDelivery or via a pre-loaded flash drive) to download the bundle.

8. Extract the feature card scripts as described in :
   See "Step 4: Extract the ACSLS Feature Card Scripts".

9. Ensure network, host name, DNS, and other network-related settings are established.
   The ACSLS recovery process maintains as many settings as possible. However, depending upon the nature of the fault, you may need to redo one or all of your previous configuration steps. It is recommended that you verify the settings established previously, including bond3 network connectivity and ensure that your IP address, host name, and related items are correct using the `hostname` command, `ping` command, and others as required. If you encounter issues, then review applicable steps as described in Installing ACSLS on the SL4000 Feature Card :

   See "Configuring the Feature Card and Preparing for ACSLS Installation".

   > **Note:**
   >
   > If you need to change your IP address, host name, or other network-related information as part of this task for feature card replacement, you must redo these steps.

10. Verify that the feature storage cards are initialized as described in Installing ACSLS on the SL4000 Feature Card :
    See "Step 5: Insert and Initialize All Feature Storage Cards".

11. As user root, issue the following commands to remove remnants from the previous ACSLS installation:

```
rm -rf /opt/ACSLS/ACSLS_8.5.1
userdel acsss
userdel acssa
userdel acsdb
userdel postgres
yum remove ACSLS
yum remove PostgreSQL
```

You may need to kill errant processes if they prohibit successful execution of yum removals. It is important that ACSLS and PostgreSQL are successfully removed.

12. Extract the ACSLS software bundle as described in Installing ACSLS on the SL4000 Feature Card :
    See "Step 10: Extract the ACSLS Software Bundle".

13. Perform the steps described in Installing, Configuring, and Running ACSLS on the Feature Card.

14. Restore the ACSLS database from the most recent backup or an accessible backup location:

    a. As user `acsss`, run `db_restore.sh`:

       ```
       su - acsss
       cd /bin
       ./db_restore.sh latest
       or
       ./db_restore.sh/bkupb/<DB_SNAPSHOT_FILENAME>
       ```

       Substitute the specific file name of your backup for *`latest`* and `<DB_SNAPSHOT_FILENAME>`.

       Redundant copies of recent backup files reside under `/bkupa`, /`bkupb` or /`bkupc`. If no ACSLS database backups are available to you locally, you may want to pull the latest database backup from your remote backup server. If you do not want to use an existing database backup, and would rather create a new one, then run `accss_config` instead.

    b. Once you have restored the database or created a new database version, re-start ACSLS as user `acsss`:

       ```
       su - acsss
       ```

Replacing a Single Feature Storage Card

To replace a single feature storage card with ACSLS, perform the following steps:

1. Log in as user `acsss` and enter the following command to shut down the ACSLS application running on the feature card:

   ```
   acsss shutdown
   ```

2. Physically remove the feature storage card to be replaced.

3. Insert the replacement feature storage card into that same location.

4. Apply a new external label to the replacement feature storage card. Include an appropriate name that reflects the changes made.

5. As user `root`, run the `featureCard_acslsStorageManager.sh` to establish the necessary ACSLS system. This command assumes that you have shut down the ACSLS application.

   ```
   cd /usr/local/bin
   ./featureCard_acslsStorageManager.sh
   ```

   As part of this process, RAID storage must be mirrored. Accordingly, you should monitor the mirroring process and ensure that it has been completed before continuing to enable

ACSLS. As user `root`, open a separate terminal window and enter the following command to monitor the rebuilding of the mirror:

```
watch cat /proc/mdstat
```

Monitor the `mdstat` display to determine when rebuilding of the mirrored drive is complete. Within minutes, `[UU]` is displayed, indicating that the rebuilding operation has completed.After `[UU]` is displayed, you can proceed with the next step.

Replacing Two Feature Storage Cards

To replace two feature storage cards with ACSLS, perform the following steps:

1. Log in as user `acsss` and enter the following command to shut down the ACSLS application running on the feature card:

   ```
   acsss shutdown
   ```

2. Physically remove the feature storage cards to be replaced.
3. Insert the replacement feature storage cards into the same locations.
4. Apply a new external label to the replacement feature storage cards. Include an appropriate name that reflects the changes made.
5. Extract the feature card scripts as described in Installing ACSLS on the SL4000 Feature Card :
   See "Step 4: Extract the ACSLS Feature Card Scripts".
6. Ensure network, host name, DNS, and other network-related settings are established.
   The ACSLS recovery process maintains as many settings as possible. Verify bond3 network connectivity and ensure that your IP address, host name, and related items are correct using the `hostname` command, `ping` command, and others as required. If you encounter issues, then review applicable steps as described in Installing ACSLS on the SL4000 Feature Card :

   See "Configuring the Feature Card and Preparing for ACSLS Installation".

   > **Note:**
   >
   > If you need to change your IP address, host name, or other network-related information as part of this task for feature card replacement, you must redo these steps.

7. Verify that the feature storage cards are initialized as described in Installing ACSLS on the SL4000 Feature Card :
   See "Step 5: Insert and Initialize All Feature Storage Cards".
8. As user root, issue the following commands to remove remnants from the previous ACSLS installation:

   ```
   rm -rf /opt/ACSLS/ACSLS_8.5.1
   userdel acsss
   userdel acssa
   userdel acsdb
   userdel postgres
   ```

```
yum remove ACSLS
yum remove PostgreSQL
```

You may need to kill errant processes if they prohibit successful execution of yum removals. It is important that ACSLS and PostgreSQL are successfully removed.

9. Extract the ACSLS software bundle as described in Installing ACSLS on the SL4000 Feature Card :
   See "Step 10: Extract the ACSLS Software Bundle".
10. Perform the steps described in Installing, Configuring, and Running ACSLS on the Feature Card.
11. Restore the ACSLS database from the most recent backup or an accessible backup location:
    a. As user `acsss`, run `db_restore.sh`:

       ```
       su - acsss
       cd /bin
       ./db_restore.sh latest
       or
       ./db_restore.sh/bkupb/<DB_SNAPSHOT_FILENAME>
       ```

       Substitute the specific file name of your backup for *`latest`* and `<DB_SNAPSHOT_FILENAME>`.

       Redundant copies of recent backup files reside under `/bkupa`, /`bkupb` or /`bkupc`. If no ACSLS database backups are available to you locally, you may want to pull the latest database backup from your remote backup server. If you do not want to use an existing database backup, and would rather create a new one, then run `accss_config` instead.
    b. Once you have restored the database or created a new database version, re-start ACSLS as user `acsss`:

       ```
       su - acsss
       ```

Upgrading ACSLS on the SL4000 Feature Card

If you have an existing SL4000 feature card running ACSLS, contact Oracle Support for guidance and required materials to perform the feature card upgrade.

- ACSLS 8.5.1 supports SL4000 firmware version 1.0.2.75 or later.
- ACSLS 8.5.0 supports SL4000 firmware version 1.0.1.69.30201 only. For this release, **the feature card and ACSLS are installed by Oracle Support**.

> **Note:**
>
> If you are currently running SL4000 Library firmware 1.0.1.69.30201 and ACSLS 8.5.0, Oracle highly recommends that you upgrade the SL4000 library firmware to 1.0.2.75 or later and ACSLS to release 8.5.1 in order to take advantage of valuable enhancements.

If you only need to upgrade an instance of ACSLS (but not the library firmware) running on the feature card, refer to the ACSLS upgrade procedures included in your ACSLS publications. Downgrades of ACSLS below the 8.5.1 release are not supported on the feature card.

The SL4000 library firmware must be upgraded before ACSLS can be upgraded:

1. Ensure that you have addressed all SL4000 and ACSLS pre-installation requirements included in this document. For the SL4000, work with Oracle Support and refer to the *SL4000 Modular Library System Library Guide* as appropriate.
2. Upgrade SL4000 library firmware using procedures included in the *SL4000 Library Guide*.
3. Work with Oracle Support to upgrade the feature cards to the same firmware version as used for the SL4000 library firmware.

   Once you have completed the hardware tasks to upgrade the SL4000 library and feature cards, and have inserted the associated hard drives, you can then install the latest ACSLS software.

   Log in to the feature card as user `root` and perform the steps described in Installing ACSLS on the SL4000 Feature Card beginning with Initializing the Feature Card and Storage Cards.

   Follow the remaining steps to install and configure ACSLS 8.5.1 on the feature card.Once ACSLS installation is complete, you can either begin anew or restore the ACSLS database from the latest (most recent) backup as user `acsss`.

   To restore a previous version of the ACSLS database, load the database from a remote server onto the feature card into the `/bkupc` directory from a flash drive or other acceptable file transfer approach within your organization. Then run `db_restore.sh` using the backup's specific file name and directory location as shown in the following example:

   ```
   su - acssscd /bin./db_restore.sh /bkupc/<DB_SNAPSHOT_FILENAME>
   ```

   Once you have restored a database, re-start ACSLS as user `acsss`:

   ```
   su - acsssacsss enable
   ```

   If you do not want to use an existing database backup, then run `accss_config` to create a new one.

Index