StorageTek Automated Cartridge System Library Software

Installation Guide

Release 8.5

E96378-04

April 2019

ORACLE

StorageTek Automated Cartridge System Library Software Installation Guide, Release 8.5

E96378-04

Contents

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.

Audience

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_8.5.0` 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` (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:

- Oracle's Sun SPARC and X86 platforms running Solaris 11, Update 3.

  Support Repository Update (SRU) 35 is required. Visit the My Oracle Support page at `https://www.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 such device drivers are attached to the system kernel, they must reside in the global zone. In cases where such 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: 4GB 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. When system memory exceeds 6GB, provide swap space that is no less than 30% of physical memory. 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.

Browser Requirements

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" on page C-1.

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

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

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 `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 3: Ensure Solaris is Installed

Ensure that a compatible version of Solaris is installed.

ACSLS 8.5 has been designed and tested to run on Oracle's Sun SPARC and X86 platforms running Solaris 11, Update 3.

Support Repository Update (SRU) 35 is required. Visit the My Oracle Support page at https://www.support.oracle.com for more information.

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.

ACSLS 8.5 was tested using the Entire Distribution selection for the Solaris installation. Oracle does not provide a minimum list of required packages for ACSLS, but the Entire Distribution is recommended.

If the Entire Distribution is not used, the Solaris installation may be missing a standard Solaris package required for correct ACSLS operation. If this occurs, acquire and install the missing package. Solaris packages can be obtained from 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.

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 8.5.0.0.0 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` subdirectory.

Step 8: Create User Accounts and Groups

Create the user accounts and associated groups described in Table 2–1. For command examples, see Appendix B.

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.

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
root	acsls	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.

Installing ACSLS

Perform the following tasks to install ACSLS:

1. Ensure that you have completed all pre-installation tasks described in "Preparing for Installation" on page 2-1.
2. Log in as root.
3. From the ACSLS_8.5.0 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 Table 2–1, 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.

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" on page C-1 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.

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
    ```

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 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" on page 2-1, 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 Administration 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)**.
   - Define an SL4000 user that the ACSLS SCI interface can use to connect to the SL4000 library.
   - Ensure that the SL4000 library is SCI capable, or has an SCI capable partition.

   Refer to Chapter 5, "Installing and Configuring Your Library Hardware" in the *ACSLS Administration 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 "Installing and Configuring Your Library Hardware" chapter in the *StorageTek ACSLS Administrator's Guide*.

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.

A known issue currently prevents use of the ACSLS GUI and the `lib_cmd` CLI (both are optional components). When running the `install.sh` script, reply **`No`** when asked about installing those components.

The output of `acsss status` should appear as follows (legacy mode):

```
-bash-4.4$ acsss status
Copyright 1989, 2018 Oracle and/or its affiliates. All Rights
Reserved.

    acsdb: online
    acsls: online
```

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: SE Linux 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 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 `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 8.5 has been designed and tested to run under Oracle Enterprise Linux releases 6.8 and 7.3.

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

https://edelivery.oracle.com

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:

In this procedure, you install key software components, including the following:

- GNOME desktop environment.
- Internet support.
- X Windows.
- Resource Package Manager (RPM), Yellowdog Updater, and Modified (yum).
- Java.

Do not install (or enable) the following:

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

If your Oracle Linux install is missing a standard Oracle Linux package required for correct ACSLS operation, please acquire and install that package. Solaris packages can be obtained from https://yum.oracle.com.

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

1. Visit https://yum.oracle.com.
2. Select the link corresponding to your Oracle Linux release. For example, `Oracle Linux 6`.
3. Select `Latest i386` (for 32-bit ACSLS compliant packages), or other appropriate link to get a list of packages.
4. Type `Ctrl-F` to search the page, and then type `unixodbc` in the Search field.
5. Click the package name link to download an RPM of that package.
6. Install the RPM package.

7. 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.

8. From your Solaris platform, click the Installation link to install.

Note: There are alternative ways to acquire and install packages using yum. Make sure you acquire 32-bit versions if the packages contain shared object libraries required by ACSLS.

Step 4: SE Linux Security Settings

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

SE Linux 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, SE Linux 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 SE Linux enforcement:

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

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

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

2. Verify the SE Linux enforcement status:

   ```
   getenforce
   ```

Note:

- This command requires that SE Linux is enabled. Use the command `sestatus` to view the status of SE Linux.
- To view the current system enforcement status, use the command `getenforce`.

Three SE Linux 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 SE Linux 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 SE Linux enforcement.

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" on page A-1.

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 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` 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:

     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:

     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 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 Table 3–1. For command examples, see Appendix B.

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.

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

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

User Account	Group Assignment	Home Directory	Command Shell	Description

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
root	acsls	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" on page 3-1.
2. Log in as `root`.
3. From the `/opt/ACSLS/ACSLS_8.5.0` 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.
- Enter **n** to terminate the installation.

Note: Installation may take significant time based on network and server configuration settings. This utility 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. Enter the following command to inherit the ACSLS environment:

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

7. As `root`, run the ACSLS `install.sh` utility:

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

8. 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.

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" on page C-1 for a description of and procedures for configuring an SSL encryption key.

9. 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.

10. 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
    ```

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 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
    ```

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:

- Adjusting ACSLS Tuning Settings
- Installing the XAPI Service

Adjusting ACSLS Tuning Settings

Set recommended ACSLS tuning settings for your configuration. See "ACSLS Tuning Settings" on page A-15.

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" on page 3-2, 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 Administration 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)**.
   - Define an SL4000 user that the ACSLS SCI interface can use to connect to the SL4000 library.
   - Ensure that the SL4000 library is SCI capable, or has an SCI capable partition.

   Refer to Chapter 5, "Installing and Configuring Your Library Hardware" in the *ACSLS Administration 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 "Installing and Configuring Your Library Hardware" chapter in the *StorageTek ACSLS Administrator's Guide*.

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 8.5 support for the SL4000 feature card.

Topics include:

Overview

Beginning with ACSLS 8.5, the ACSLS server can be installed on a feature card directly attached to the SL4000 library. This option provides a fully functional Oracle Enterprise Linux 6.8 environment with ACSLS installed in a secure RAID-1 file system.

The feature card 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 600GB hard drive and local power for the drive. These will be used by the feature card.
- One DC power converter.

ACSLS 8.5 supports SL4000 Keystone firmware version 1.0.1.69.30201.

The SL4000 library can accommodate two feature card kits. However, you can only configure one kit for ACSLS 8.5.

Installation Options

The feature card is installed by Oracle Support. Optionally, you can purchase both feature card and ACSLS installation by Oracle Advanced Customer Services.

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.

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.

If you are interested in using the SL4000 feature card to run ACSLS, 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.

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 `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 `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

To un-install the ACSLS 8.5 software:

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

   ```
   acsss shutdown
   ```

3. Log in as `root`.
4. Go to the ACSLS_8.5.0 package installation directory (typically `/opt/ACSLS_8.5.0`)
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 `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

To un-install the ACSLS 8.5 software:

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

   ```
   acsss shutdown
   ```

3. 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                              8.5.0-22
   installed
   ```

4. 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.

5. 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.

6. Reboot.

   ACSLS is now uninstalled.

A

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 or Linux 7.3, 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 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 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 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 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 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 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 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 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) 0
stack size              (kbytes, -s) 16000
cpu time               (seconds, -t) unlimited
max user processes              (-u) 30000
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) 30000

-bash-4.1$ ulimit -aH
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) unlimited
max memory size         (kbytes, -m) unlimited
open files                      (-n) 65568
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200

real-time priority              (-r) 0
stack size              (kbytes, -s) unlimited
cpu time               (seconds, -t) unlimited
max user processes              (-u) 65568
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) 65568
```

B

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

# Example output of existing acsls group:
> getent group acsls
acsls:x:516:root

# 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
```

C

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.

Index

X

Y