StorageTek Automated Cartridge System Library Software

Installation Guide

Release 8.4

E62370-05

March 2018

StorageTek Automated Cartridge System Library Software Installation Guide, Release 8.4

E62370-05

Contents

Preface vii

Audience vii
Documentation Accessibility vii
Related Documents vii
Conventions viii

1 Overview

Software Requirements 1-1
System Requirements 1-2
Browser Requirements 1-3
Co-Hosting 1-3

2 Installing ACSLS on Solaris

Legal Notice 2-1
Exporting the Database and Control Files 2-2
Installing Solaris 2-2
Notes for the Solaris Administrator 2-2
Network Security 2-2
Access Privileges 2-3
User Administration 2-3
Cron Administration 2-4
Removing Previous ACSLS Version 2-4
Installing the ACSLS Package 2-5
Installing PostgreSQL 2-6
Running install.sh 2-7
Creating the Database 2-7
Installing an mchanger Driver 2-7
Installing Support for Logical Libraries 2-8
Installing the Graphical User Interface 2-8
Installing lib_cmd 2-9
Installing ACSLS Services 2-9
Adding Users of the ACSLS GUI 2-9
Installing the XAPI Service with ACSLS 2-10
Importing the Database and Control Files 2-10
Installing and Configuring your Library Hardware 2-10

Testing a New ACSLS Release Without a Library 2-11
Verifying ACSLS Installation 2-12
Auditing the Library 2-13
Uninstalling the XAPI Service 2-13
Uninstalling ACSLS 8.4 2-14
Uninstalling any SCSI Media Changer Drivers 2-15

3 Installing ACSLS on Linux

Preliminary Notes for the Linux Administrator 3-2
User Administration 3-2
Cron Administration 3-3
Access Privileges 3-3
SELinux Security 3-3
Removing Previous ACSLS Version 3-4
Before Installing Linux 3-4
Exporting the Database and Control Files 3-5
Installation Media 3-5
Downloading the Linux Media Pack 3-6
Installing Linux 3-6
Installing ACSLS Prerequisites 3-6
Adjusting Tuning Settings for Linux and ACSLS 3-7
Linux Installation Tips 3-8
Installing the ACSLS Package 3-8
Running install.sh 3-10
Creating the Database 3-11
Installing an mchanger Driver 3-11
Removing mchanger Device Links 3-12
Installing the Graphical User Interface 3-12
Installing lib_cmd 3-13
Installing ACSLS Services 3-13
Adding Users of the ACSLS GUI 3-13
Installing the XAPI Service with ACSLS 3-14
Importing the Database and Control Files 3-14
Installing and Configuring your Library Hardware 3-14
Testing a New ACSLS Release Without a Library 3-15
Verifying ACSLS Installation 3-15
Auditing Your Library 3-16
Uninstalling the XAPI Service 3-16
Uninstalling ACSLS 8.4 3-17

A Linux and ACSLS Tuning Settings

Linux Network and Operating System Tuning Settings A-1
Network Tuning Parameter Settings A-1
Operating System Tuning Parameter Settings A-2
ACSLS 8.4 Tuning Settings A-4
Verifying Tuning Settings A-5
Verifying Network Tuning Settings A-5

Verifying Operating System Tuning Settings A-5

B Configuring a Self-Signed Digital Certificate for HTTPS

Index

Preface

Automated Cartridge System Library Software (ACSLS) is Oracle's StorageTek's 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 accesses 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.

Related Documents

For more information, see the following documents in the ACSLS, Release 8.4. documentation set:

- *ACSLS 8.4 Security Guide*
- *ACSLS 8.4 Licensing Information User Manual*
- *ACSLS 8.4 Release Notes*

- *ACSLS 8.4 Product Information*
- *ACSLS 8.4 Administrator's Guide*
- *ACSLS 8.4 Messages*
- *ACSLS 8.4 Quick Reference*
- *ACSLS HA 8.4 Cluster Installation, Configuration, and Operation*
- *ACSLS 8.4 Interface Reference*
- *ACSLS SNMP Agent 2.4 Installation and User's Guide for Solaris*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 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 8.4 is bundled with WebLogic 10.3.6.

ACSLS 8.4 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. The Linux installation procedure described in this document explains the process of adding PostgreSQL packages from the Oracle `yum` repository after installing the Linux Product Pack.

Software Requirements

ACSLS 8.4 has been developed and tested for the following operating system environments:

- Oracle’s Sun SPARC and X86 platforms running Solaris 11, Update 2 or 3.
- Oracle Enterprise Linux 6.5 (ACSLS 8.4 base through patch 8.4.0-4)
- Oracle Enterprise Linux 6.8 (requires ACSLS 8.4 patch 8.4.0-4)

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.
- ACSLS 8.4 High Availability (HA) systems must be installed on their own dedicated platform pair.

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 4GB of memory and a minimum of 2GB 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 8.4 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
  ```

- Fibre card (optional) 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 8.4 is fully tested and certified with QLogic and Emulex HBAs.

Browser Requirements

The ACSLS 8.4 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 B-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 8.4 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 8.4 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*.

Solaris Cluster Software is supported on the Oracle VM Server for SPARC and this platform can be employed in an ACSLS HA application. Refer to the *Oracle Solaris Cluster Data Service for Oracle VM Server for SPARC Guide*.

2 Installing ACSLS on Solaris

ACSLS 8.4 has been developed and tested for the following Solaris environments:

- Oracle Solaris 11 Update 2 (ACSLS 8.4 base through patch 8.4.0-2)
- Oracle Solaris 11 Update 3 (requires ACSLS 8.4 patch 8.4.0-2 or later)

  For ACSLS patch installation instructions, refer to the README.txt file included with the patch.

> **Note:** Only run the `install.sh` script **after** the patch has been applied. This is true for both patches 8.4.0-2 and 8.4.0-3, which modify the installation requirements and behavior in that script.

This chapter includes the following topics:

- "Legal Notice" on page 2-1
- "Exporting the Database and Control Files" on page 2-2
- "Installing Solaris" on page 2-2
- "Installing the ACSLS Package" on page 2-5
- "Installing PostgreSQL" on page 2-6
- "Running install.sh" on page 2-7
- "Installing the XAPI Service with ACSLS" on page 2-10
- "Importing the Database and Control Files" on page 2-10
- "Installing and Configuring your Library Hardware" on page 2-10
- "Testing a New ACSLS Release Without a Library" on page 2-11
- "Verifying ACSLS Installation" on page 2-12
- "Auditing the Library" on page 2-13
- "Uninstalling the XAPI Service" on page 2-13
- "Uninstalling ACSLS 8.4" on page 2-14
- "Uninstalling any SCSI Media Changer Drivers" on page 2-15

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_8.4.0 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_8.4.0 (typically under `/opt`). Look in the subdirectory, `acsls_thirdPartySoftware/`.

Exporting the Database and Control Files

If you are upgrading from a prior release, you must export the database and control files. As user acsss, run the command:

```
db_export.sh -f /path/to/my/export/file
```

In the example above, *`myExport`* is the name you assign to your export file. A second file with a `.misc` extension is also created. You should save *`myExport`* and *`myExport.misc`* to a non-volatile location. If you are updating your OS, then transfer these files to a remote machine for safe keeping.

For more information and procedures, refer to Exporting the Database in the “Database Administration” chapter of the *StorageTek ACSLS 8.4 Administrator’s Guide*.

If you have created additional ACSLS GUI users on ACSLS 8.1 or later releases, record those user IDs so you can re-add them after installing the new version of ACSLS. To do this:

1. As user `acsss`:

   ```
   cd $ACS_HOME/install
   ```

2. Type `su root`.

   Do not type `su - root` to retain your `acsss` environment.

3. Run `./userAdmin.sh` to get a list of existing users of the ACSLS GUI:

   Select the List Users option and then the Exit option when you have finished.

4. Record the user IDs so you can re-add them later as described in "Adding Users of the ACSLS GUI" on page 2-9.

Installing Solaris

This section describes the steps to install Solaris.

Notes for the Solaris Administrator

For installation procedures refer to the Solaris Installation instructions.

ACSLS 8.4 was tested using the Entire Distribution selection for the Solaris install. Oracle does not provide a minimum list of required packages for ACSLS, but the Entire Distribution is recommended.

Network Security

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
```

Access Privileges

- ACSLS 8.4 may be installed in any file system. The ACSLS base and the ACSLS backup directories (for example, `/export/home` and `/export/backup`) must be mounted to allow `SETUID` so 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, see the *ACSLS Administrator's Guide*, "Firewall Security Option" for details.

User Administration

- Three ACSLS user accounts, (`acsss`, `acssa`, `acsdb`) are added automatically when you install the ACSLS package.
- The package install creates an `acsls` group and assigns all three users to this group. It also adds `root` to the `acsls` group.
- If user accounts for the three acsls users already exist, the user home directory and group id will be adjusted automatically (if necessary) by the package install routine.
- ACSLS 8.4 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`. The user home directories for ACSLS are:

  ```
  acsss   $installDir/ACSSS
  acssa   $installDir/ACSSA
  acsdb   $installDir/acsdb/ACSDB1.0
  ```

- If user accounts already exist for these users and you are changing the `$installDir`, then these users must be logged out of the system during the installation since their home directory will change.
- If the user accounts already exist and they are locked, they must be unlocked before you install the package.

  To check if the `acsss` account is locked:

  ```
  # passwd -s acsss
  acsss LK
  ```

  The "LK" tells you that the account is locked. To unlock the account:

  ```
  # passwd -u acsss
  ```

  Do this for each user account.

- 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 needed to complete the ACSLS installation. Make sure the users are re assigned to the `acsls` group and their home directories conform as stated in the fourth bullet. The user shell should be: /bin/bash

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.d/cron.allow 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 fail.

  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 acsss and acsdb are not contained in the cron.deny file.

Removing Previous ACSLS Version

1. If you are upgrading from an earlier version of ACSLS be sure to preserve your existing database which includes the library configuration and the tape cartridge locations inside the library. As user acsss:

   ```
   $ acsss db
   $ db_export.sh -f /path/to/a/safe/location
   ```

   A typical ACSLS installation may involve an OS upgrade. Ensure the two export files are stored to a remote location where they are preserved during the upgrade

2. Remove the ACSLS package.

   a. As user acsss, bring down ACSLS.

   ```
   $ acsss shutdown
   ```

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

   On Solaris, to remove the package, execute the un-install script.

   ```
   # ./pkg_unistall.sh
   ```

   On Linux, use rpm to remove the package.

   ```
   # rpm -e ACSLS
   ```

   For ease of re-installation, not all ACSLS files are removed. The ACSLS user accounts and some directories still remain.

   c. To remove the GUI user accounts and other files left behind, use:

   ```
   # cd $installDir
   # rm -rf ACSSS ACSSA acsdb SSLM
   ```

   d. To remove ACSLS administrative accounts:

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

Installing the ACSLS Package

ACSLS 8.4 installs in any directory. Determine the base install directory where the ACSLS application should reside. If that directory does not exist, you should create it before installing the STKacsls package. The directory must be owned by root with permissions set to 755.

> **Note:** Unless otherwise specified by the user, ACSLS will be installed in /export/home.

To follow the default installation procedure:

1. Go to the Oracle Software Delivery Cloud, and find the ACSLS_8.4.0 software bundle available for both SPARC and X86 platforms.

   a. Download the appropriate software bundle to a common installation directory, typically /opt, and unzip the compressed file. The extracted package set is found in the resulting ACSLS_8.4.0 subdirectory.

   b. PostgreSQL 8.3 installs automatically when you install ACSLS 8.4. If you prefer to install PostgreSQL 8.4 (see "Installing PostgreSQL" on page 2-6), download the postgresql.xxx.bz2 file to your install directory (typically /opt) before running the package install script in the next step.

   c. Go into the ACSLS_8.4.0 directory and run the command:

   ```
   ./pkg_install.sh
   ```

   The pkg_install script first prompts you to confirm your intent to install ACSLS 8.4.

   A number of packages are included with the ACSLS 8.4 installation bundle. They include STKacsls and five postgres packages that support PostgreSQL.

   On Solaris 11 the specific postgres packages are not already installed, so they are installed automatically when you run the pkg_install script. Review the license for each package and respond **y** to accept the package.

   d. Before installing the STKacsls package, the script prompts you (y/n) whether to accept /export/home as the default base directory for the ACSLS application.

   If you answer **n**, the script asks you to enter the desired path to the package base directory. If the directory you specify does not exist, the script prompts for permission create the directory.

   When the package installation is complete, you find that the packages in ACSLS_8.4.0 have been moved to /var/spool/pkg. They remain there for ease of re-installation until they are manually removed. What remains in ACSLS_8.4.0 is pkg_install.sh, pkg_uninstall.sh, and README.txt. You can use these scripts to uninstall or re-install ACSLS at any time. Any SUNWpostgr 8.3 packages that were not installed will also remain in this directory.

2. The package installation utility creates user and group IDs for the following users: acsss, acssa, and acsdb. It assigns home directories for these users and places them in the acsls group. The root user is also added to the acsls group.

   When upgrading from a prior ACSLS version, determine whether you intend to change the installation directory. ACSLS users should be logged out whenever their home directory is likely to change.

> **Note:** Secure administration practices recommend that you to set initial passwords for these users immediately after the package installation.

3. Once the ACSLS packages are installed, `root` needs to inherit the ACSLS environmental attributes. To do this, log out and log back in, or simply `su -` to inherit the `acsls` group identity. Verify with the `groups` command.

   ```
   su -
   # groups
      root acsls
   ```
   (other groups may be listed)

4. To set your shell to the ACSLS installation environment, source the `.acsls_env` file:

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

   This step enables you to refer to `$ACS_HOME` during subsequent installation operations.

5. Proceed to "Running install.sh" on page 2-7.

Installing PostgreSQL

PostgreSQL 8.3 installs by default with ACSLS 8.4. Run `pkg_install.sh` and no further action is necessary. PostgreSQL 8.3 is completely compatible with ACSLS 8.4.

ACSLS 8.4 is also compatible with PostgreSQL 8.4. If you prefer to install a recent update of PostgreSQL 8.4, simply download the bz2 bundle (`postgresql-8.4.xx-S11.<`*`platform`*`>-32.tar.bz2`) from the PostgreSQL website to the install directory (typically `/opt`) where you downloaded the ACSLS 8.4 zip bundle. The ACSLS install script (`pkg_install.sh`) recognizes what you have downloaded and installs it automatically.

To obtain a recent update to PostgreSQL 8.4, go to the following website:

`http://www.postgresql.org/ftp/binary/`

1. From this URL you get a listing of all PostgreSQL releases. You should select the latest maintenance level for version 8.4. Be sure to get the 32-bit version that is compatible with your server architecture. Navigate to the download file and select in the following order:

   ```
   v8.4.xx
   binary
   solaris
   solaris11
   sparc or i386
   postgresql-8.4.xx-S11.<platform>-32.tar.bz2
   ```

2. Be sure to select the 32-bit version.

3. Move the `postgresql bz2` file to the parent directory of this package install directory (typically `/opt`). The ACSLS_8.4.0 installation script, `pkg_install.sh`, automatically installs the compressed PostgreSQL file that you downloaded, and moves it to the proper file system directory, `/usr/postgres/8.4`.

4. If the `PostgreSQL-8.4 tar` or `bz2` file is not found, and if PostgreSQL 8.3 or 8.4 is not already installed in `/usr/postgres/`, the `pkg_install.sh` script installs the

four SUNWpostgr-83 packages included in this directory. It then moves them to /var/spool/pkg where they are installed using pkgadd.

Running install.sh

The install.sh utility enables you to select from the extracted ACSLS 8.4 package the specific features required for your unique Oracle StorageTek library environment. Flexibility has been added in ACSLS 8.4, allowing you to choose whether to install options including the Graphical User Interface (GUI) and fibre library support. You can run this utility to install the entire product, any portion of the product, or to alter an already-installed product without the need for a full installation.

While you are still logged in as root, run the commands:

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

Creating the Database

Database creation is first step in the install.sh routine. This step is necessary if you are installing the package for the first time. If your ACSLS database already exists and you do not want to rebuild it, then you have the option to skip this step. This step creates a new database under PostgreSQL and establishes an automated schedule for database backups.

Determine the directory where you intend for the database to reside. If that directory does not exist, then you must first create the directory. The directory must be owned by root with permissions set to 755. Unless you specify otherwise, the backup directory will be placed directly under your base directory. See step-1(c) in "Installing the ACSLS Package" on page 2-5.

The install.sh routine asks:

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

Click **Return** to select the suggested directory, or specify a different directory. If you assign a relative path, it is placed directly under the desired path that you assigned in step-2 in the section, "Installing the ACSLS Package" on page 2-5.

The install routine proceeds to load policy modules. These allow the ACSLS application to freely access its PostgreSQL database.

Installing an mchanger Driver

The mchanger driver is relevant only to fibre-attached or SCSI-attached library configurations. The install.sh routine asks:

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

Respond with **y** or **n** whether your library environment includes a fibre-attached library such as the SL500 or SL150 library.

If you enter **y**, the routine scans the attached SAN environment, looking for any StorageTek library devices. It reports the devices it finds and prompts 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
```

Installing Support for Logical Libraries

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 `install.sh` routine probes the system for qualified HBAs, and then lists the ports it finds with the following prompt:

```
Please select the HBA port you intend for Target-mode operation:
1) HBA Port WWN xxxyyyzzz       Not connected
2) HBA Port WWN aaabbbccc   Connected to a remote HBA
```

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

Installing the Graphical User Interface

The Graphical User Interface (GUI) is an option.

If you are co-hosting ACSLS with another application that uses WebLogic, do not install the ACSLS GUI. To install the GUI:

1. Enter **y** at the following prompt:

   ```
   Do you want to install the ACSLS Graphical User Interface? (y/n)
   ```

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

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

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

3. Select one of the following:

   - 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 install 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 (y/n) whether to remove the existing GUI configuration.

When you install WebLogic on your ACSLS server, a 512-bit public key is automatically available to support basic https exchanges with client browsers. Normally, no further configuration should be necessary. However, more recent browsers, notably Internet Explorer 8 and above and Firefox 39 and above, require a lengthier key of no less than 1024 bits. Refer to "Configuring a Self-Signed Digital Certificate for HTTPS" on page B-1 for a description of and procedures for configuring an SSL encryption key.

Installing lib_cmd

The `lib_cmd` feature is a command-line interface that performs many of the same operations that can be performed in the ACSLS GUI. This tool installs automatically if you choose to install either the GUI or logical library support.

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. If you do choose to install neither the GUI nor logical library support, you are given the option to install `lib_cmd`.

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

Installing ACSLS Services

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
```

Adding Users of the ACSLS GUI

During install.sh, you created the acsls_admin user.   This user can now create accounts and assign passwords for other users of the ACSLS Web-based GUI application. You can refer to the list of GUI users that you saved earlier. To add a user, follow this procedure:

1. As root, go to the `/export/home/ACSSS/install` directory.
2. Run `./userAdmin.sh`
3. Enter the `acsls_admin` password that you assigned in "Installing the Graphical User Interface" on page 2-8.
4. From the menu, select (1) to add a new user.
5. Enter the ID of the user you want to add.
6. Assign a password for that user.

Passwords must contain eight characters with a combination of alpha and numeric or special characters.

You can use the `userAdmin.sh` utility at any time to add or delete users or to change passwords for all ACSLS GUI users. See `userAdmin.sh` in the Utilities chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

Installing the XAPI Service with ACSLS

The XML API (XAPI) is an API that allows StorageTek clients and servers to communicate using a common ELS protocol over TCP/IP. ACSLS 8.4 and later releases can be configured with XAPI support.

You install the XAPI component separately from ACSLS and after ACSLS has been installed.

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 as `root` to the ACSLS server.
3. Source key ACSLS environment variables:

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

   (There is a 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 the Database and Control Files

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

If you have exported the database and control files, you now need to import them. The *control files* include those files in the `data/external` directory that have been customized to your specific environment.

If you are migrating to ACSLS 8.4 from a previous release and have customized your dynamic or static variables, you must import them. For information, refer to Importing the Database in the “Database Administration” chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

Installing and Configuring your Library Hardware

If you are configuring ACSLS with an actual library follow this procedure. If you are installing a new ACSLS release and do not have a test library to use for configuring and testing ACSLS, see "Testing a New ACSLS Release Without a Library" on page 2-11".

1. Verify the server system hardware is properly configured, connected, powered on, and ready.
2. Verify each of the physical connections (example: Ethernet, fibre, SCSI) connections between the server and the library hardware.
3. Before configuring ACSLS to your library complex, make sure all libraries, rails, and CAPs are fully configured, powered on, and ready.
4. Create or import the Library Configuration. Refer to the *ACSLS 8.4 Administrator's Guide* for details.

   To import the configuration from an earlier ACSLS release, see the section *Importing the Database* in the “Database Administration” chapter.

To create a new library configuration, see the section *Configuring or Reconfiguring Library Hardware* in the “Installing and Configuring Your Library Hardware” chapter.

5. If you are using logical libraries to support SCSI clients over Fibre Channel, set up the FC connections between any client HBA ports and suitable HBA ports on the ACSLS server. Fibre connections to logical library client machines should be active when you install ACSLS.

6. For help with connectivity problems, refer to the 'Troubleshooting” chapter in the *ACSLS 8.4 Administrator's Guide*.

7. Refer to the “Installing and Configuring Your Library Hardware” chapter in the *ACSLS 8.4 Administrator's Guide*. See the section “Using acsss_config to Configure Your Library Hardware”.

Testing a New ACSLS Release 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 library or library partition to use as a test environment, it is possible to test a new ACSLS release in a limited way without having a test library for ACSLS to access. Follow this procedure:

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. See the *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 come ups 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 ACSLS Installation

Use the following procedure to verify ACSLS. You should be logged in as `acsss`. This procedure mounts or dismounts a cartridge.

To start ACSLS Software, log in as user `acsss` and run the `acsss enable` command. Refer to `acsss` in the “Utility” chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

For instructions on using `cmd_proc`, refer to "Using a cmd_proc" in the *StorageTek ACSLS 8.4 Administrator's Guide*.

1. Query the server from the `cmd_proc` by entering

   ```
   query server
   ```

If messages are displayed indicating that the server is in waiting mode, wait for a message indicating that the server is running.

2. Verify the following are online. You must have at least one of each online. If not, bring them online with the `vary` command.

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

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

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

4. Mount a volume by entering:

   ```
   mount vol_id drive_id
   ```

   Use the `query drive` command to get the ID of an available drive and the `query volume` command to get the ID of a library cartridge. Refer to the "Installing and Configuring Your Library Hardware" chapter in the *StorageTek ACSLS 8.4 Administrator's Guide*.

5. Did 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/dismount still fails, call Oracle Support for assistance.

6. Dismount the cartridge by entering:

   ```
   dismount vol_id drive_id force
   ```

   In the above command, *`vol_id`* is the volume and *`drive_id`* is the drive you specified in step-4.

Auditing the Library

The last step of your installation is auditing your libraries. You also need to audit your libraries if:

- This is a new installation.
- You are adding new libraries to an existing configuration.

Refer to "Auditing the Library" in the Library Management chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

Uninstalling the XAPI Service

The XAPI component can be removed without uninstalling ACSLS. To do this:

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

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

(There is a period and space before `/var/tmp/acsls/.acsls_env`).

3. Uninstall the XAPI component.

```
cd $ACS_HOME/install
./remove_xapi.sh
Do you wish to remove the xapi service? (y)
```

Uninstalling ACSLS 8.4

> **Note:** If you are upgrading to another release of ACSLS, make sure to export your ACSLS database by using the `db_export.sh` utility command discussed in the "Utility" chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

To uninstall ACSLS:

1. Log in as `acsss`.
2. Enter `acsss shutdown`.
3. Remove the package. To do this:

   a. Log in as `root`.

   b. Go to the ACSLS_8.4.0 package installation directory (typically `/opt/ACSLS_8.4.0`) and 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.

   On Solaris 11, the `pkg_uninstall` utility prompts you whether to uninstall the PostgreSQL packages. You would normally answer "*n*" to this prompt unless you are permanently removing the ACSLS application.

4. To remove the contents of the ACSLS database backup directory:

   ```
   rm -rf $ACSDB_BKUP
   ```

5. 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 ACSLS from your system, perform the following steps:

   ```
   cd $installDir
   rm -rf Oracle, SSLM
   userdel acsss
   userdel acssa
   ```

```
userdel acsdb
```

6. Reboot.

Uninstalling any SCSI Media Changer Drivers

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
   ```

3

Installing ACSLS on Linux

ACSLS 8.4 has been developed and tested for the following Linux environments:

- Oracle Enterprise Linux 6.5 (ACSLS 8.4 base through patch 8.4.0-4)
- Oracle Enterprise Linux 6.8 (requires ACSLS 8.4 patch 8.4.0-4)

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

`https://edelivery.oracle.com`

The procedures described in this chapter conform to Linux environments that include the specific packages detailed in the section, "Installing Linux" on page 3-6. By closely following that section and "Linux Installation Tips" on page 3-8, your installation should proceed smoothly and you can avoid issues of software incompatibility or unresolved package dependencies. Deviation from the prescribed set of packages is recommended only for experienced Linux administrators.

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
```

This chapter includes the following topics:

- "Preliminary Notes for the Linux Administrator" on page 3-2
- "Exporting the Database and Control Files" on page 3-5
- "Installation Media" on page 3-5
- "Installing Linux" on page 3-6
- "Installing the ACSLS Package" on page 3-8
- "Running install.sh" on page 3-10

- "Installing the XAPI Service with ACSLS" on page 3-14
- "Importing the Database and Control Files" on page 3-14
- "Installing and Configuring your Library Hardware" on page 3-14
- "Testing a New ACSLS Release Without a Library" on page 3-15
- "Verifying ACSLS Installation" on page 3-15
- "Auditing Your Library" on page 3-16
- "Uninstalling the XAPI Service" on page 3-16
- "Uninstalling ACSLS 8.4" on page 3-17

Preliminary Notes for the Linux Administrator

This section includes the following topics:

- "User Administration" on page 3-2
- "Cron Administration" on page 3-3
- "Access Privileges" on page 3-3
- "SELinux Security" on page 3-3
- "Before Installing Linux" on page 3-4

User Administration

- Three ACSLS user accounts, (acsss, acssa, acsdb) are added automatically when you install the ACSLS package.
- The package install creates an acsls group and assigns all three users to this group. It also adds root to the acsls group.
- If user accounts for the three acsls users already exist, the user home directory and group id will be adjusted automatically (if necessary) by the package install routine.
- ACSLS 8.4 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. The user home directories for ACSLS are:

```
acsss   $installDir/ACSSS
acssa   $installDir/ACSSA
acsdb   $installDir/acsdb/ACSDB1.0
```

- If user accounts already exist for these users and you are changing the $installDir, then these users must be logged out of the system during the installation since their home directory will change.
- If the user accounts already exist and they are locked, they must be unlocked before you install the package. To check if the acsss account is locked:

```
# passwd -S acsss
acsss L
```

  The "L" indicates that the account is locked. To unlock the account:

```
# passwd -u acsss
```

  Do this for each user account.

- 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 needed to complete the ACSLS installation. Make sure the users are re assigned to the acsls group and their home directories conform to the guidelines in this section. The user shell should be /bin/bash.

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 certain machines) 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 fail.

  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 acsss and acsdb are not contained in the cron.deny file.

Access Privileges

- ACSLS 8.4 may be installed in any file system. The ACSLS base and the ACSLS backup directories (example: /export/home and /export/backup) must be mounted to allow SETUID so 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, see the *ACSLS Administrator's Guide*, "Firewall Security Option" for details.

SELinux Security

ACSLS 8.4 is designed to run in *optional* Security Enhanced Linux environments. SELinux was merged into the Linux 2.6.0 kernel in 2003 in response to initiatives by the US National Security Agency. It provides access control to files, directories, and other system resources that go beyond the traditional protection found standard in UNIX environments. In addition to owner-group-public permission access, SELinux includes access control based on user role, domain, and context. The agent that enforces access control over all system resources is the Linux kernel.

The root user on a Linux system can set enforcement on or off with the setenforce command.

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

Use Enforcing or 1 to put SELinux in enforcing mode. Use Permissive or 0 to put SELinux in permissive mode.

Note:

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

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

For more information, refer to the "Troubleshooting" Appendix in the *StorageTek ACSLS 8.4 Administrator's Guide*.

Removing Previous ACSLS Version

1. Is this a new installation?
   - YES - go to "Installing ACSLS Prerequisites" on page 3-6.
   - NO - make sure you exported the database by using the `db_export.sh` utility command.
2. Shut down ACSLS:

   You must be logged in as `acsss` to do this.

   ```
   acsss shutdown
   ```

   Remove any `acsss` and `acsdb` crontab entries:

   ```
   crontab -r acsss
   crontab -r acsdb
   ```
3. Remove ACSLS for Linux 8.3:

   ```
   rpm -e ACSLS-8.3.0.i686
   ```
4. Verify that no database processes are running before you begin the install. If in doubt, reboot.

Before Installing Linux

Before you begin installing Linux, check with your IT system administrator to obtain the following information. The graphical installer requires the kdelibs package, which is included in the Linux Installer Media 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.

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

Exporting the Database and Control Files

If you are porting an earlier release of ACSLS from a Solaris or AIX machine, be sure to export your database and control files from that machine. The control files include those files in the `data/external` directory that have been customized to your local library environment. If you are moving the database and control files to the ACSLS 8.4 Linux platform, you must export the database and control files. As user `acsss`, run the command:

```
db_export.sh -f myExport
```

In the example above, *`myExport`* is the name you assign to your export file. You should save *`myExport`* and *`myExport.misc`* to a non-volatile location. If you are updating your OS, then transfer these files to a remote machine for safe keeping.

For more information and procedures, refer to Exporting the Database in the "Database Administration" chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

If you have created additional ACSLS GUI users on ACSLS 8.1 or later releases, record those user IDs so you can re-add them after installing the new version of ACSLS. To do this:

1. Login in as `root` and source the ACSLS operating environment:

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

2. Go to the `$ACS_HOME/install` directory.
3. Run `./userAdmin.sh` to get a list of existing users of the ACSLS GUI:

   Select the List Users option and then the Exit option when you have finished.
4. Record the user IDs so you can re-add them later as described in "Adding Users of the ACSLS GUI" on page 3-13.

Installation Media

You may install Oracle Linux from DVD media, from a jump-start server, or from an ISO image that resides on a remote server. Most contemporary Oracle Sun X86 servers are equipped with an advanced service processor using Integrated Lights Out Manager (ILOM). The ILOM enables you to install the Linux operating system on the same machine using remotely mounted media. Consult your Sun server documentation for details how to use the Oracle ILOM.

Downloading the Linux Media Pack

Use the following procedure to download the Linux installer media pack from the Oracle Software Deliver Cloud. The media pack is delivered as a compressed ISO image file which you can extract and write to portable media of your choice.

1. Start a web browser on the system and navigate to the Oracle Software Delivery Cloud at the following URL.

   `https://edelivery.oracle.com`

2. Click Sign In.
3. Enter the username and password provided by your Oracle support representative.
4. Click **Accept** the on Export Restrictions screen.
5. Enter **acsls** for the Product and select StorageTek Automated Cartridge System Library Software (ACSLS).
6. Click **Select Platform** and check **Linux x86-64**. Click **Select**.
7. On the Selected Products screen, click **Continue**.
8. For Available Releases, check the box next to the Automated Cartridge System Library Software 8.4.0.0.0 for Linux x86-64 and click **Continue**.
9. On the Oracle Terms and Restrictions screen, review and accept the terms of the licenses. Click **Continue**.
10. Click **Download** and save the zip file to the location of your choice.

Installing Linux

ACSLS 8.4 has been tested and verified on Linux releases 6.5 and 6.8, with **required** changes to system settings. See"Adjusting Tuning Settings for Linux and ACSLS" on page 3-7 for more information.

It is recommended that PostgreSQL packages are **not** included with the base installation of Linux. These packages are installed in the following section.

Installing ACSLS Prerequisites

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` and `/etc/wgetrc` to update proxy and caching parameters:

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

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
```

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

   a. Obtain the repository list from the Oracle yum server.

   ```
   # cd /etc/yum.repos.d
   # wget http://public-yum.oracle.com/public-yum-ol6.repo
   ```

   b. Edit the file, `public-yum-ol6.repo`, to include i686 packages. Add the following lines to the bottom of this file.

   ```
   [ol6_latest_i386]
   name=Oracle Linux 6 Latest 32-bit (i386)
   baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL6/latest/i386/
   gpgkey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-ol6
   gpgcheck=1
   enabled=1
   ```

3. Disable the yum packagekit refresh.

   a. Edit the file:

   ```
   /etc/yum/pluginconf.d/refresh-packagekit.conf
   ```

   b. Set enabled=0.

4. Install the pre-requisite packages for ACSLS.

   ```
   yum install -y glibc.i686 pam pam.i686
   yum install -y libstdc++ libstdc++.i686
   yum install -y libxml2 libxml2.i686
   yum install -y postgresql-libs-8.4.20-6.el6.i686
   yum install -y postgresql-8.4.20-6.el6.i686
   yum install -y postgresql-server-8.4.20-6.el6.i686
   yum install -y unixODBC.i686 postgresql-odbc.i686
   ```

5. Verify that a newer version of Postgresql is not installed on the Linux machine, as newer versions include pathing changes that cause errors for ACSLS:

   ```
   rpm -qa | egrep "^postgresql*" | egrep "8.4.2[1-9]"
   ```

   If new versions are not returned, proceed with installation.

   If versions newer than 8.4.20 are returned, for example:

   ```
   postgresql84-libs-8.4.22-1PGDG.rhel6.i686
   postgresql84-8.4.22-1PGDG.rhel6.i686
   postgresql84-server-8.4.22-1PGDG.rhel6.i686
   ```

   Then remove the newer Postgresql versions:

   ```
   rpm -e -nodeps postgresql84-libs-8.4.22-1PGDG.rhel6.i686
   rpm -e -nodeps postgresql84-8.4.22-1PGDG.rhel6.i686
   rpm -e -nodeps postgresql84-server-8.4.22-1PGDG.rhel6.i686
   ```

   When you are finished, repeat the initial yum installation steps at the beginning of Step 6.

Once pre-requisite packages are installed, you are now ready to install the ACSLS 8.4 package.

Adjusting Tuning Settings for Linux and ACSLS

If you plan to run ACSLS in a Linux 6.5 or 6.8 environment, you must perform Linux operating system and ACSLS tuning tasks designed to optimize ACSLS performance and stability. See Appendix A, "Linux and ACSLS Tuning Settings" for complete instructions.

Linux Installation Tips

Your Linux installation may not contain all of the packages required for ACSLS. When you install the ACSLS package with `rpm -ivh ACSLS-8.4.0*.rpm`, this operation may return an error indicating specific missing package dependencies. For each dependency that is identified, your can install that package with `yum`. Simply copy the package name from the dependency error message and submit that package name to a `yum install` command.

```
# yum install <package name>
```

To get a list of any dependencies in the ACSLS package before you attempt to install it, run the `rpm install` command with the --test option.

```
# rpm -ivh --test ACSLS-8.4.0*.rpm
```

If there are still missing dependencies that remain on your Linux system, those dependencies are displayed with this test option.

Installing the ACSLS Package

Note: ACSLS patch 8.4.0-4 is required in a Linux 6.8 environment. For 8.4.0-4 patch installation instructions, refer to the `README.txt` file included with the patch.

Visit the Oracle Software Delivery Cloud, and locate the ACSLS_8.4.0 software bundle available for the Linux platform. You will typically download the file to the `/opt` directory on your ACSLS server. Unzip the file and go into the `ACSLS_8.4.0` directory to access the `rpm` package, `ACSLS-8.4.0*.rpm`.

ACSLS 8.4 installs in any directory. Determine the base install directory where the ACSLS application should reside. If that directory does not exist, then you must create it before installing the STKacsls package. The directory must be owned by root with permissions set to `755`.

Note: Unless otherwise specified by the user, ACSLS will be installed in `/export/home`.

1. As `root`, go into the `ACSLS_8.4.0` package installation directory and install the ACSLS package.

   ```
   pkg=ACSLS-8.4.0*.rpm
   rpm -i $pkg
   ```

   This method places the ACSLS application under the default install directory, `/export/home`. To place the application under a different directory, use the following method to extract the package:

   ```
   path=/my/desired/path
   ```
   (the file system where you intend for the ACSLS application to reside)

   ```
   rpm -Uvh --prefix $path ACSLS-8.4.0*.rpm
   ```

The rpm utility creates user and group IDs for users acsss, acssa, and acsdb. It assigns home directories for these users and places them in the acsls group. The root user is also added to the acsls group.

When upgrading from a prior ACSLS version, determine whether you intend to change the installation directory. ACSLS users should be logged out whenever their home directory is likely to change.

> **Note:** Secure administration practices recommend that you to set initial passwords for these users immediately after the package installation.

2. Once the package installation is complete, you should inherit the new ACSLS environment to your current shell. To inherit the acsls group identity, you must log out and log back in (or simply su -). Verify with the groups command.

```
su -
# groups
   root acsls
```

(other groups may be listed)

> **Note:** To ensure a consistent environment when installing ACSLS 8.4, it is recommended that you complete the installation in a single shell session, logged in as user 'root', in which you have verified root membership in the acsls group.
>
> Problems caused by incorrect group membership can occur if multiple shell sessions are used during installation (a stale shell session logged in as user 'root', for example, may not have the same group access).

To verify ACSLS and root accounts for proper Group (GID) association:

a. Verify ACSLS accounts:

Submit the following command to verify that all ACSLS accounts (acsss, acssa, and acsdb) are associated with the same 'acsls' Group ID (GID), as defined in the /etc/group file for group named 'acsls':

```
egrep "acsss|acssa|acsdb" /etc/passwd
```

Account information is displayed. For example:

```
acsss:x:505:511:ACSLS control login:/export/home/ACSSS:/bin/bash
acssa:x:506:511:ACSLS SA login:/export/home/ACSSA:/bin/bash
acsdb:x:507:511:ACSLS Database Owner:/export/home/acsdb/ACSDB1.0:/bin/bash
```

Notice that all ACLS accounts are associated with the same Group ID (511 in this example).

If the ACSLS accounts are not associated with the same Group ID, submit the following commands as user 'root' to correct the associations:

```
usermod -g acsls acsss
usermod -g acsls acssa
usermod -g acsls acsdb
```

When you are finished, re-submit the initial command to verify that all accounts are associated with the same Group ID:

```
egrep "acsss|acssa|acsdb" /etc/passwd
```

b. Verify root account:

Submit the following command to verify that the root user is part of the 'acsls' group:

```
egrep "acsls" /etc/group
```

Account information is displayed. For example:

```
acsls:x:511:root
```

Notice that the root user is listed for the 'acsls' group.

If root is not listed, submit the following command to add a supplementary 'acsls' group:

```
usermod -G acsls root
```

When you are finished, re-submit the initial command to verify that the root user is part of the 'acsls' group:

```
egrep "acsls" /etc/group
```

3. To set your shell to the ACSLS installation environment, source the `.acsls_env` file.

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

This step lets you refer to `$ACS_HOME` during subsequent installation operations.

Running install.sh

The `install.sh` utility lets you select from the extracted ACSLS 8.4 package the specific features required for your unique Oracle StorageTek library environment. Flexibility has been added in ACSLS 8.4, allowing you to choose whether to install options including the Graphical User Interface (GUI) and fibre library support. You can run this utility to install the entire product, any portion of the product, or to alter an already-installed product without the need for a full installation.

While you are still logged in as `root`, run the following commands:

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

The software looks for a fully qualified host name in the `/etc/hosts` file. Some Postgres utilities require the fully qualified host name. Add the fully qualified hostname to your `/etc/hosts` file after this installation

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
```

Creating the Database

Database creation is first step in the `install.sh` routine. This step is necessary if you are installing the package for the first time. If your ACSLS database already exists and you do not want to rebuild it, then you have the option to skip this step.

This step creates a new database under PostgreSQL and establishes an automated schedule for database backups. For Linux 6, it loads SE Linux policy modules that provide ACSLS with secure but unrestrained access to the PostgreSQL database engine.

Determine the directory where you intend for database backup files to reside. If that directory does not exist, then you must first create the directory. The directory must be owned by `root` with permissions set to `755`. Unless you specify otherwise, database backups are placed under `/export/backup`.

The `install.sh` routine asks:

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

Click **Return** to select the suggested directory, or specify a different directory. If you assign a relative path, it is placed directly under the desired path that you assigned in step-1 in the previous section, "Installing the ACSLS Package" on page 3-8. If yours is an HA configuration, ensure that the backup directory that you choose resides in a file system on the shared disk.

The install routine proceeds to load SE Linux policy modules. Unless you specify otherwise, the routine places the directory for database backups directly under the desired path that you assigned in step-1.

Installing an mchanger Driver

The mchanger driver is relevant only to fibre-attached or SCSI-attached library configurations. The `install.sh` routine asks:

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

Respond with **y** or **n** whether your library environment includes a fibre-attached library such as the SL500 or SL150 library.

If you entered **y**, the routine scans the attached SAN environment, looking for any StorageTek library devices. It reports the devices it finds and prompts 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
```

Removing mchanger Device Links

When controlling fibre-attached libraries such as the SL500 or SL150, *mchanger* is the name that ACSLS uses when referring to the device driver for the SCSI media changer device. On Linux, `/dev/mchanger*` is a symbolic link to the SCSI Generic *sg* driver. It is not our purpose to remove *sg* since *sg* is a standard device driver on Linux. We simply remove the device links between *mchanger* and *sg*. This task is normally accomplished when you remove the ACSLS rpm package. To remove *mchanger* without removing the ACSLS package, use the following procedure:

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

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

2. Remove the rules that created the device links that you removed in step 1.

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

To re-create the mchanger device links:

1. As user `acsss`:

   ```
   cd $ACS_HOME/install
   ```

2. Type `su root`.

   Do not type `su - root` as this causes a loss of your `acsss` environment.

3. Source the ACSLS environment:

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

4. Create the mchanger device links.

   ```
   # $ACS_HOME/install/install_scsi_Linux.sh
   ```

Installing the Graphical User Interface

The Graphical User Interface (GUI) is an option.

If you are co-hosting ACSLS with another application that uses WebLogic, do not install the ACSLS GUI.

To install the GUI:

1. Enter **y** at the following prompt:

   ```
   Do you want to install the ACSLS Graphical User Interface? (y/n)
   ```

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

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

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

3. Select one of the following:

   - 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 install 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 (y/n) whether 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. Refer to "Configuring a Self-Signed Digital Certificate for HTTPS" on page B-1 for a description of and procedures for configuring an SSL encryption key.

Installing lib_cmd

The `lib_cmd` feature is a command-line interface that performs many of the same operations that can be performed in the ACSLS GUI. This tool installs automatically if you choose to install either the GUI or logical library support.

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. If you do choose to install neither the GUI nor logical library support, you are given the option to install `lib_cmd`.

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

Installing ACSLS Services

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
```

Adding Users of the ACSLS GUI

During `install.sh`, you created the `acsls_admin` user. This user can now create accounts and assign passwords for other users of the ACSLS Web-based GUI application. You can refer to the list of GUI users that you saved earlier. To add a user, follow this procedure:

1. As root, go to the `/export/home/ACSSS/install` directory.
2. Run `./userAdmin.sh`
3. Enter the `acsls_admin` password that you assigned in "Installing the Graphical User Interface" on page 3-12.
4. From the menu, select (1) to add a new user.
5. Enter the ID of the user you want to add.
6. Assign a password for that user.

Passwords must contain eight characters with a combination of alpha and numeric or special characters.

You can use the `userAdmin.sh` utility at any time to add or delete users or to change passwords for all ACSLS GUI users. See `userAdmin.sh` in the Utilities chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

Installing the XAPI Service with ACSLS

The XML API (XAPI) is an API that allows Enterprise level mainframe clients and servers to communicate using a common Enterprise Library Software (ELS) protocol over TCP/IP. ACSLS 8.4 and later releases can be configured with XAPI support.

The XAPI component is install separately from ACSLS and after ACSLS has been installed.

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 as `root` to the ACSLS server.
3. Source key ACSLS environment variables:

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

   (There is a 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 the Database and Control Files

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

If you have exported the database and control files, you now need to import them. For information, refer to Importing the Database in the "Database Administration" chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

Installing and Configuring your Library Hardware

1. Verify the server system hardware is properly configured, connected, powered on, and ready.

2. Verify each of the physical connections (example: Ethernet, fibre, SCSI) connections between the server and the library hardware.
3. Before configuring ACSLS to your library complex, make sure all libraries, rails, and CAPs are fully configured, powered on, and ready.
4. Create or import the Library Configuration. Refer to the *ACSLS 8.4 Administrator's Guide* for details.

   To import the configuration from an earlier ACSLS release, see the section *Importing the Database* in the "Database Administration" chapter.

   To create a new library configuration, see the section *Configuring or Reconfiguring Library Hardware* in the "Installing and Configuring Your Library Hardware" chapter.
5. For help with connectivity problems, refer to the "Troubleshooting" chapter in the *StorageTek ACSLS 8.4 Administrator's Guide*.
6. Refer to the "Installing and Configuring Your Library Hardware" chapter in the *StorageTek ACSLS 8.4 Administrator's Guide*. See the section "Using acsss_config to Configure Your Library Hardware".

Testing a New ACSLS Release Without a Library

After installing a new ACSLS release, you want to test it before using it to manage production libraries. If you do not have a library or library partition to use as a test environment, it's possible to test a new ACSLS release in a limited way without having a test library for ACSLS to access.

See "Testing a New ACSLS Release Without a Library" on page 2-11 for more information and procedures.

Verifying ACSLS Installation

Use the following procedure to verify ACSLS. You should be logged in as `acsss`. This procedure mounts or dismounts a cartridge.

To start ACSLS Software, log in as user `acsss` and run the `acsss enable` command. Refer to `acsss` in the "Utility" chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

For instructions on using `cmd_proc`, refer to "Using a cmd_proc" in the *StorageTek ACSLS 8.4 Administrator's Guide*.

1. Query the server from the `cmd_proc` by entering

   ```
   query server
   ```

   If messages are displayed indicating that the server is in waiting mode, wait for a message indicating that the server is running.
2. Verify the following are online. You must have at least one of each online. If not, bring them online with the `vary` command.

   ```
   query port all
   query acs all
   query lsm all
   query drive all
   ```
3. Do you have at least one cartridge in an LSM?

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

4. Mount a volume by entering:

```
mount vol_id drive_id
```

Use the query drive command to get the ID of an available drive and the query volume command to get the ID of a library cartridge. Refer to the "Installing and Configuring Your Library Hardware" chapter in the *ACSLS Administrator's Guide*.

5. Did 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/dismount still fails, call Oracle Support for assistance.

6. Dismount the cartridge by entering:

```
dismount vol_id drive_id force
```

where `vol_id` is the volume and `drive_id` is the drive you specified in Step 4.

Auditing Your Library

The last step of your installation is auditing your libraries. You must audit your libraries if:

- This is a new installation.
- You are adding new libraries to an existing configuration.

Refer to "Auditing the Library" in the Library Management chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

Uninstalling the XAPI Service

The XAPI component can be removed without uninstalling ACSLS. To do this:

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

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

(There is a period and space before `/var/tmp/acsls/.acsls_env`).

3. Uninstall the XAPI component.

```
cd $ACS_HOME/install
./remove_xapi.sh
Do you wish to remove the xapi service? (y)
```

Uninstalling ACSLS 8.4

> **Note:** If you are upgrading to another release of ACSLS, make sure to export your ACSLS database by using the `db_export.sh` utility command discussed in the "Utility" chapter of the *StorageTek ACSLS 8.4 Administrator's Guide*.

To uninstall ACSLS:

1. Log in as `acsss`.
2. Enter `acsss shutdown`.
3. Remove package:
   - Log in as `root`.
   - Enter `rpm -e ACSLS`

   The rpm 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.
4. To remove the contents of the ACSLS database backup directory:

   ```
   rm -rf <previous ACSLS install directory>
   ```

5. 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 ACSLS from your system, perform the following steps:

   ```
   cd <previous ACSLS database backup directory>
   rm -rf Oracle SSLM
   userdel acsss
   userdel acssa
   userdel acsdb
   ```

6. Reboot.

A

Linux and ACSLS Tuning Settings

This appendix describes tuning settings required in an environment running ACSLS 8.4 on Oracle Enterprise Linux release 6.5 or 6.8. It includes the following topics:

- "Linux Network and Operating System Tuning Settings" on page A-1
- "ACSLS 8.4 Tuning Settings" on page A-4
- "Verifying Tuning Settings" on page A-5

Linux Network and Operating System Tuning Settings

The following Linux tuning parameter values are required to accommodate the size and complexity of ACSLS. These tuning parameters are a combination of Network operating system related parameters configured in the file `/etc/sysctl.conf`, and ACSLS User operating system limits as set 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.

Network Tuning Parameter Settings

The following tunable options must be persisted to the file `/etc/sysctl.conf`:

```
# 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 = 1

# Disable netfilter on bridges.
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 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 shared segment size, in bytes
kernel.shmmax = 68719476736

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296

# Set Read/Write Max Buffers size to 8MB
net.core.wmem_max=8388608
net.core.rmem_max=8388608

# Set Read/Write Default Buffers size to 8MB
net.core.wmem_default=8388608
net.core.rmem_default=8388608

# Set TCP listen backlog queue depth
net.core.somaxconn=2048

# Allow Setuid programs to drop core
fs.suid_dumpable=1

# Set the core pattern to drop in /var/crash
kernel.core_pattern=/var/crash/core.%u.%e.%p
```

Operating System Tuning Parameter Settings

The following tunable options must be persisted to the file `/etc/security/limits.d/90-nproc.conf`:

```
#
# 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   8192000


# 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   8192000

# 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 8.4 Tuning Settings

This section describes required changes to ACSLS product behavior variable settings.

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**.

   ```
   **** 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`.

Verifying Tuning Settings

After rebooting the ACSLS server using the `reboot -p` command, verify your tuning parameter changes.

Verifying Network Tuning Settings

Use the following commands to display the associated tuning values:

```
sysctl net.core.wmem_max
sysctl net.core.rmem_max
sysctl net.core.wmem_default
sysctl net.core.rmem_default
sysctl net.core.somaxconn
```

Verify that the following values are returned:

```
net.core.wmem_max=8388608
net.core.rmem_max=8388608
net.core.wmem_default=8388608
net.core.rmem_default=8388608
net.core.somaxconn=2048
```

Verifying Operating System Tuning Settings

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

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 1-b.

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 1-b 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

A

ACSLS
- de-installing, 2-14
- uninstalling, 2-14
- verifying installation, 2-12

B

browser requirements, 1-3

C

co-hosting, 1-3

E

Exporting the database, 2-2

I

Installing Solaris, 2-2

L

Linux
- auditing the library, 3-16
- exporting database and control files, 3-5
- importing database and control files, 3-14
- installation media, 3-5
- installing ACSLS, 3-8
- installing and configuring library hardware, 3-14
- installing Linux, 3-6
- preliminary notes, 3-2
- running install.sh, 3-10
- testing a new ACSLS release without a library, 3-15
- uninstalling ACSLS, 3-17
- uninstalling the XAPI service, 3-16
- verifying ACSLS installation, 3-15

M

mchanger
- uninstalling, 2-15

S

self-signed digital certificate for https, B-1
software requirements, 1-1
Solaris
- auditing the library, 2-13
- configuring library hardware, 2-10
- exporting the database and control files, 2-2
- importing the database, 2-10
- installing ACSLS package, 2-5
- installing PostgreSQL, 2-6
- removing previous version, 2-4
- testing a new ACSLS release without a library, 2-11
- uninstalling, 2-14

system requirements, 1-2

V

verifying ACSLS, 2-12