NetBackup™ Device Configuration Guide

UNIX, Windows, and Linux

Release 11.2.0.1

COHESITY

NetBackup Device Configuration Guide

Last updated: 2026-09-09

Legal Notice

Copyright © 2026 Cohesity, Inc. All rights reserved.

© 2026 Cohesity, Inc. All Rights Reserved. Cohesity, the Cohesity Logo and other Cohesity Marks are trademarks of Cohesity, Inc. in the US and/or internationally. The information supplied herein is the confidential and proprietary information of Cohesity and may only be used (a) by the intended recipients and (b) in conjunction with validly licensed Cohesity software and services. Find the terms of Cohesity licenses at www.cohesity.com/agreements.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. COHESITY SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Cohesity Support

Reach Cohesity Support

There are several ways to create a Cohesity support case.

- Go to Cohesity Support, to search in our knowledge base; or contact us by phone - United States and Canada: 1-855-9CO-HESI (926-4374), option 2.
- Log in to the Cohesity Support Portal to create a new case.
- Click the (**?**) icon on the Cohesity UI and select Support Portal.

Support/Service Assistance

First, contact the Service Provider that you have contracted for service and support. If you work directly with Cohesity and have a product warranty/entitlement, repair pricing, or technical support-related question, see your options below:

- To find solutions to your product issues or for suggestions or best practices, visit the Cohesity Knowledge Base.
- Log in to the Cohesity Support Portal to create a new case.
- To monitor your open cases, log in to the portal and click the **Cases** tab on the home page. This page should have all the case statuses and updates. You can also view individual case status.

Cohesity Software Running on Partner Hardware

For Cohesity software running on qualified third-party hardware, the following support workflow applies:

1. The customer may contact Cohesity Support first if the issue cannot be determined as a hardware issue.

   Note: Cohesity cannot process hardware replacement requests for partner hardware.

2. Cohesity Support triages the issue. If it is a software issue, Cohesity Support continues to work on it.
3. If it is a hardware/firmware issue or is suspected to be a hardware/firmware issue, Cohesity provides information about the issue to the customer and requests that the customer open a support ticket with the appropriate partner.
4. If needed, Cohesity Support can join a three-way call with the partner and the customer.
5. The customer informs Cohesity Support on the progress of the partner's case.

Contents

Chapter 1 Introducing device configuration 7

Using this guide 7
General device configuration sequence 8
Configuration cautions 8
About the NetBackup compatibility lists 9

Section 1 Operating systems 10

Chapter 2 Linux 11

Before you begin on Linux 11
About the required Linux SCSI drivers 12
About the st driver debug mode 13
Verifying the Linux drivers 13
About configuring robot and drive control for Linux 13
About the Linux robotic control device files 14
About the Linux tape drive device files 14
Verifying the device configuration on Linux 15
About SAN clients on Linux 15
About SCSI persistent bindings for Linux 16
About Emulex HBAs 16
Utilities to test SCSI devices 17
Linux command summary 17

Chapter 3 Solaris 18

Before you begin on Solaris 18
About the NetBackup sg driver 19
Determining if the NetBackup sg driver is installed 20
Special configuration for the StorEdge Network Foundation HBA driver 21
About binding Fibre Channel HBA drivers 21
Configuring Solaris 10 x86 for multiple drive paths 22
Installing/reinstalling the sg and the st drivers 22
st.conf file example 25
sg.conf file example 25

sg.links file example 26
Configuring 6 GB and larger SAS HBAs in Solaris 27
Preventing Solaris driver unloading 29
About Solaris robotic controls 30
About SCSI and FCP robotic controls on Solaris 30
Examples of SCSI and FCP robotic control device files on Solaris 31
About Solaris tape drive device files 31
About Berkeley-style close 33
About no rewind device files on Solaris 33
About fast-tape positioning (locate-block) on Solaris 33
About SPC-2 SCSI reserve on Solaris 33
Disabling SPC-2 SCSI reserve on Solaris 34
About nonstandard tape drives 34
Configuring Solaris SAN clients to recognize FT media servers 35
Adding the FT device entry to the st.conf file 35
Modifying the st.conf file so that Solaris discovers devices on two LUNS 36
Uninstalling the sg driver on Solaris 37
Solaris command summary 37

Chapter 4 Windows 38

Before you begin configuring NetBackup on Windows 38
About tape device drivers on Windows 39
Attaching devices to a Windows system 39

Section 2 Robotic storage devices 40

Chapter 5 Robot overview 41

NetBackup robot types 41
NetBackup robot attributes 42
ACS robots 42
TLD robots 43
Table-driven robotics 44
Robotic test utilities 45
Robotic processes 45
Processes by robot type 46
Robotic process example 47

Chapter 6 Oracle StorageTek ACSLS robots 49

About Oracle StorageTek ACSLS robots 50
Sample ACSLS configurations 50
Media requests for an ACS robot 54
About configuring ACS drives 54
Configuring shared ACS drives 56
Adding tapes to ACS robots 58
About removing tapes from ACS robots 58
Removing tapes from an ACS robot 59
Removing tapes using NetBackup 59
Robot inventory operations on ACS robots 59
Configuring a robot inventory filtering on ACS robots 61
NetBackup robotic control, communication, and logging 62
NetBackup robotic control, communication, and logging for Windows systems 62
NetBackup robotic control, communication, and logging for UNIX systems 62
ACS robotic test utility 67
acstest on Windows systems 67
acstest on UNIX systems 68
Changing your ACS robotic configuration 68
ACS configurations supported 68
Multiple ACS robots with one ACS library software host 69
Multiple ACS robots and ACS library software hosts 69
Oracle StorageTek ACSLS firewall configuration 70

Chapter 7 Device configuration examples 72

An ACS robot on a Windows server example 72
An ACS robot on a UNIX server example 75

Chapter 1

Introducing device configuration

This chapter includes the following topics:

- Using this guide
- General device configuration sequence
- About the NetBackup compatibility lists

Using this guide

Use this guide to help set up and configure the operating systems of the hosts you use for NetBackup servers. Also use this guide for help with storage devices. This guide provides guidance about NetBackup requirements; it does not replace the vendor documentation.

This guide is organized as follows:

- Information about operating systems.
- Information about robotic storage devices.

Read the "Before you start" sections (if applicable) of the chapters in this guide. These sections provide any important platform-specific instructions or may contain specific instructions or limitations for server types.

Cohesity tested the configuration file options in this guide; other configuration settings may also work.

To minimize configuration errors, you can copy and paste configuration details from a text file of the operating system chapters of this configuration guide. The format of this text file is similar to the printed version of the guide. Be sure to review the differences as explained at the beginning of the text file.

The `NetBackup_DeviceConfig_Guide.txt` file is installed with NetBackup server software in the following paths:

- `/usr/openv/volmgr` (UNIX)
- *`install_path`*`\Veritas\Volmgr` (Windows)

The Hardware Compatibility List contains information about supported devices.

See "About the NetBackup compatibility lists" on page 9.

General device configuration sequence

Use the following general sequence when you configure devices:

- Physically connect the storage devices to the media server. Perform any hardware configuration steps that the device vendor or the operating system vendor specifies.
- Create any required system device files for the drives and robotic control. Device files are created automatically on Windows and on some UNIX platforms. Explicit configuration of device files is required on some UNIX servers to make full use of NetBackup features.
  For SCSI-controlled libraries, NetBackup issues SCSI commands to the robotic devices. SCSI commands allow NetBackup to discover and configure devices automatically. You may have to configure the server operating system to allow device discovery.
- Add the storage devices to NetBackup and configure them.
  For instructions, see the NetBackup Web UI Administrator's Guide or the help. You can configure devices in NetBackup from the primary server to which the devices are attached (the device host).

Configuration cautions

Observe the following cautions:

- In multiple-initiator (multiple host bus adapter) environments, NetBackup uses SCSI reservation to avoid tape drive usage conflicts and possible data loss problems. SCSI reservation operates at the SCSI target level; the hardware that bridges Fibre Channel to SCSI must work correctly.
  By default, NetBackup uses SPC-2 SCSI reserve and release. Alternatively, you can use SCSI persistent reserve or disable SCSI reservation entirely.
  For information about the NetBackup use of SCSI reservation, see the following:
  - "Enable SCSI reserve" in the NetBackup Web UI Administrator's Guide, Volume I.

  - "How NetBackup reserves drives" in the NetBackup Web UI Administrator's Guide, Volume II.
- Cohesity does not recommend or support the use of single-ended to differential SCSI converters on NetBackup controlled devices. You may encounter problems if you use these converters.

About the NetBackup compatibility lists

Cohesity provides compatibility lists for the operating systems, peripherals, and software with which NetBackup works.

See the NetBackup compatibility lists at the following webpage:

https://support.cohesity.com/s/article/article-100040093

Section 1

Operating systems

- Chapter 2. Linux
- Chapter 3. Solaris
- Chapter 4. Windows

Chapter 2

Linux

This chapter includes the following topics:

- Before you begin on Linux
- About the required Linux SCSI drivers
- Verifying the Linux drivers
- About configuring robot and drive control for Linux
- Verifying the device configuration on Linux
- About SAN clients on Linux
- About SCSI persistent bindings for Linux
- About Emulex HBAs
- Utilities to test SCSI devices
- Linux command summary

Before you begin on Linux

Observe the following important points when you configure the operating system:

- Verify that NetBackup supports your server platform and devices. The Cohesity support Web site contains server platform compatibility information. For the compatibility information, see the NetBackup compatibility lists : https://support.cohesity.com/s/article/article-100040093
- For SCSI controlled libraries, NetBackup issues SCSI commands to the robotic devices. For NetBackup to function correctly, the properly named device files must exist. Information about how to configure device files is available.
  See “About configuring robot and drive control for Linux” on page 13.

- Verify that a SCSI low-level driver is installed for each HBA in your system, as follows:
  - Follow the HBA vendor's installation guide to install or load the driver in the kernel.
  - Configure the kernel for SCSI tape support and SCSI generic support.
  - Probe all LUNs on each SCSI device and enable the SCSI low-level driver for the HBA.
  - Enable multi-LUN support for the kernel according to the Linux documentation.

  For more information, refer to your HBA vendor documentation.
- Multipath configurations (multiple paths to robots and drives) are supported only with the following configurations:
  - Native path (`/dev/nstx`, `/dev/sgx`)
  - The `sysfs` file system that is mounted on `/sys`
  - Native udev rules for persistent device paths (`/dev/tape/by-path`)

After you configure the hardware, add the robots and the drives to NetBackup.

About the required Linux SCSI drivers

To use SCSI tape drives and robotic libraries, the following drivers must be configured in the kernel or loaded as modules:

- SCSI tape (st) driver.
- Standard SCSI driver.
- SCSI-adapter driver.
- Linux SCSI generic (sg) driver. This driver allows pass-through commands to SCSI tape drives and control of robotic devices.

  NetBackup and its processes use the pass-through driver as follows:
  - To scan or discover drives
  - For SCSI reservations
  - For SCSI locate-block operations
  - For SAN error recovery
  - For Quantum SDLT performance optimization
  - To collect robot and drive information

- To collect Tape Alert information from tape drives
- For WORM tape support
- For future features and enhancements

The standard Enterprise Linux releases have the sg and the st modules available for loading. The modules are loaded as needed. Also, you can load these modules if they are not in the kernel. Use the following commands:

```
/sbin/modprobe st
/sbin/modprobe sg
```

About the st driver debug mode

You can enable debug mode for the `st` tape driver. Debug mode echoes each command and its result to the system log. For details, see the Linux documentation.

Verifying the Linux drivers

NetBackup requires specific Linux drivers.

See "About the required Linux SCSI drivers" on page 12.

You can use the `/sbin/lsmod` command to verify that the st and the sg drivers are loaded in the kernel.

To verify that the drivers are installed and loaded in the kernel

- Invoke the `lsmod` command as follows:

```
lsmod
Module                  Size            Used by
sg                     14844              0
st                     24556              0
```

About configuring robot and drive control for Linux

NetBackup supports SCSI control and API control of robotic devices. SCSI control includes Fibre Channel Protocol (FCP), which is SCSI over Fibre Channel.

You must configure the control method, as follows:

- SCSI or Fibre Channel Protocol control.

NetBackup uses device files to configure control for SCSI tape devices, including robotic devices. (A robotic device in a library moves the media between storage slots and the drives in the library.)

- API control over a LAN.
  See the "Oracle StorageTek ACSLS robots" topic of this guide.

About the Linux robotic control device files

For robotic devices, NetBackup uses the `/dev/sgx` device files, where *x* is a decimal number from 0 to 255. Linux should create the device files automatically. If the device files do not exist, see the Linux documentation for information about how to create them.

If you use device discovery, NetBackup looks for /dev/sgx robotic control device files. NetBackup discovers the robotic control device files (and hence the devices) automatically. Alternatively, if you add a robot manually in NetBackup, you must enter the pathname to the device file for that robotic device.

About the Linux tape drive device files

For tape drive device files, NetBackup uses the `/dev/tape/by-path/xxxx-nst` symbolic link files (`-nst` indicates the no rewind device file). The `/dev/tape/by-path` files are symbolic links to `/dev/nstx` device files. The Linux `udev` system creates the `/dev/tape/by-path` symlinks. These are persistent paths that always point to the same device. The`/dev/nstx` files can change associated devices, without updating NetBackup. Therefore, the `/dev/nstx` paths should not be used.

The Linux driver should create the `/dev/nstx` device files automatically. The Linux `udev` device management system should create the `/dev/tape/by-path` symbolic link files automatically. If the device files do not exist, see the Linux documentation for information about how to create them.

If you use device discovery in NetBackup, NetBackup looks for `/dev/tape/by-path/xxxx-nst` symbolic link files. NetBackup discovers the device files (and hence the devices) automatically. Alternatively, if you add a drive manually in NetBackup, you should enter the `/dev/tape/by-path/xxxx-nst` symbolic link pathname as the device file for that drive. If the `/dev/nstx` device paths are configured, restarting the NetBackup Device Manager (`ltid`) converts the paths to `/dev/tape/by-path` persistent paths.

The NetBackup `avrd` daemon establishes a default tape driver operating mode. If you change the default mode, NetBackup may not read and write tapes correctly, which results in data loss.

Verifying the device configuration on Linux

The `/proc/scsi/scsi` file shows all of the devices that the SCSI driver detects.

If the operating system detects the SCSI devices, NetBackup can discover them.

To verify that the operating system can see the devices

- Run the following command from a terminal window:

  ```
  cat /proc/scsi/scsi
  ```

  The output that is displayed should be similar to the following:

  ```
  Attached devices:
  Host: scsi8 Channel: 00 Id: 05 Lun: 00
    Vendor: IBM      Model: ULT3580-HH8      Rev: HB81
    Type:   Sequential-Access                ANSI  SCSI revision: 06
  Host: scsi8 Channel: 00 Id: 05 Lun: 01
    Vendor: IBM      Model: 3573-TL          Rev: 1110
    Type:   Medium Changer                   ANSI  SCSI revision: 05
  Host: scsi2 Channel: 00 Id: 01 Lun: 00
    Vendor: IBM      Model: ULT3580-HH7      Rev: H9E3
    Type:   Sequential-Access                ANSI  SCSI revision: 06
  ```

About SAN clients on Linux

NetBackup SAN clients on Linux hosts require the SCSI Generic (sg) driver pass-through tape drive device files for traffic to NetBackup FT media servers. The media server FT devices appear as `ARCHIVE Python` tape devices during SCSI inquiry from the SAN client. (However, they are not tape devices and do not appear as tape devices in NetBackup device discovery.)

You should verify that you have the correct driver and device files.

See "Verifying the Linux drivers" on page 13.

If your Linux operating system does not add all of the SCSI device files automatically, you can do so manually. The following is an example of code you can include in the `/etc/rc.local` file to add LUN 1, targets 0-7 on Controllers 0-2. Note that the last line is the `MAKEDEV` command, which makes the required device files. The code you include in your `/etc/rc.local` file depends on how your hardware environment.

```
# Add the troublesome device on LUN 1 for the FT server
echo "scsi add-single-device 0 0 0 1"    > /proc/scsi/scsi
echo "scsi add-single-device 0 0 1 1"    > /proc/scsi/scsi
echo "scsi add-single-device 0 0 2 1"    > /proc/scsi/scsi
```

```
echo "scsi add-single-device 0 0 3 1"    > /proc/scsi/scsi
echo "scsi add-single-device 0 0 4 1"    > /proc/scsi/scsi
echo "scsi add-single-device 0 0 5 1"    > /proc/scsi/scsi
echo "scsi add-single-device 0 0 6 1"    > /proc/scsi/scsi
echo "scsi add-single-device 0 0 7 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 0 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 1 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 2 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 3 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 4 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 5 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 6 1"    > /proc/scsi/scsi
echo "scsi add-single-device 1 0 7 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 0 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 1 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 2 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 3 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 4 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 5 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 6 1"    > /proc/scsi/scsi
echo "scsi add-single-device 2 0 7 1"    > /proc/scsi/scsi
/dev/MAKEDEV sg
```

About SCSI persistent bindings for Linux

Cohesity recommends that you use persistent bindings to lock the mappings between the SCSI targets that are reported to Linux and the specific devices. The Linux kernel device manager udev creates the `/dev/tape/by-path` symbolic links to `/dev/nst`*x* device paths that NetBackup uses to communicate with tape drives. The udev system creates the persistent paths using the `/dev/tape/by-path` symbolic links. Do not change the default udev rules that create these paths.

If you cannot use binding with the HBA in your configuration, add an `ENABLE_AUTO_PATH_CORRECTION` entry in the `/usr/openv/volmgr/vm.conf` file on all Linux media servers.

About Emulex HBAs

If you use a `/usr/openv/volmgr/AVRD_DEBUG` touch file on a NetBackup media server with an Emulex HBA driver, the system log may contain entries similar to the following:

```
Unknown drive error on DRIVENAME (device N, PATH) sense[0] = 0x70,
sense[1] = 0x0, sensekey = 0x5
```

You can ignore these messages.

Utilities to test SCSI devices

You can manipulate tape devices with the operating system `mt` command. For more information, see the `mt`(1) man page.

You can use theNetBackup `robtest` utility to test robots. The `robtest` utility resides in `/usr/openv/volmgr/bin`.

A set of SCSI utilities are available from the Linux SCSI Generic (`sg`) driver home page.

Linux command summary

The following is a summary of commands that were used in this topic:

- `/sbin/lsmod`
  Lists the modules that are loaded.
- `/sbin/modprobe`
  Installs loadable kernel modules.
- `/usr/sbin/reboot`
  Stops and restarts the system.
- `/bin/mknod /dev/sg`*`x`*` c 21 `*`N`*
  Creates SCSI generic device files; *x* is a decimal number from 0 to 255.

Chapter 3

Solaris

This chapter includes the following topics:

- Before you begin on Solaris
- About the NetBackup sg driver
- Determining if the NetBackup sg driver is installed
- Special configuration for the StorEdge Network Foundation HBA driver
- About binding Fibre Channel HBA drivers
- Configuring Solaris 10 x86 for multiple drive paths
- Installing/reinstalling the sg and the st drivers
- Configuring 6 GB and larger SAS HBAs in Solaris
- Preventing Solaris driver unloading
- About Solaris robotic controls
- About Solaris tape drive device files
- Configuring Solaris SAN clients to recognize FT media servers
- Uninstalling the sg driver on Solaris
- Solaris command summary

Before you begin on Solaris

Observe the following points when you configure the operating system:

- Verify that NetBackup supports your server platform and devices. Download the NetBackup hardware and the operating system compatibility lists.

https://support.cohesity.com/s/article/article-100040093

- For SCSI controlled libraries, NetBackup issues SCSI commands to the robotic devices.

  For NetBackup to function correctly, the properly named device files must exist, as follows:

  - NetBackup installs its own pass-through driver, the SCSI generic `sg` driver. You must configure this driver properly to create device files for any device NetBackup uses.
  - The Solaris tape and disk driver interfaces also create a device file for each tape drive device. These device files must exist for all read or write I/O capability.

  See "About Solaris robotic controls" on page 30.
- Verify that the Solaris `st` driver is installed.
- Verify that the devices are configured correctly. To do so, use the Solaris `mt` command and the NetBackup `/usr/openv/volmgr/bin/sgscan` utility.

  For the tape drives that you want to share among NetBackup hosts, ensure that the operating system detects the devices on the SAN.
- When you configure devices, you should attach all peripherals and restart the system with the reconfigure option (`boot -r` or `reboot -- -r`).
- If you remove or replace adapter cards, remove all device files that are associated with that adapter card.
- If you use the Automated Cartridge System (ACS) robotic software, you must ensure that the Solaris Source Compatibility Package is installed. The package is required so that the ACS software can use the shared libraries in `/usr/ucblib`.
- Oracle systems with parallel SCSI host bus adapters do not support 16-byte SCSI commands on any devices that are attached to these HBAs. Therefore, those HBAs do not support WORM media. To override this limitation, create a touch file as follows:

  ```
  touch /usr/openv/volmgr/database/SIXTEEN_BYTE_CDB
  ```

After you configure the hardware, add the robots and the drives to NetBackup.

About the NetBackup sg driver

NetBackup provides its own SCSI pass-through driver to communicate with SCSI-controlled robotic peripherals. This driver is called the SCSA (generic SCSI pass-through driver), also referred to as the `sg` driver.

For full feature support, NetBackup requires the `sg` driver and SCSI pass-through device paths.

Install the NetBackup `sg` driver on each Solaris NetBackup media server that hosts tape devices. Each time you add or remove a device, you should reinstall the `sg` driver again.

If you do not use a pass-through driver, performance suffers.

NetBackup uses the pass-through driver for the following:

- By `avrd` and robotic processes to scan drives.
- By NetBackup to position tapes by using the locate-block method.
- By NetBackup for SAN error recovery.
- By NetBackup for Quantum SDLT performance optimization.
- By NetBackup for SCSI reservations.
- By NetBackup device configuration to collect robot and drive information.
- To collect Tape Alert information from tape devices allowing support of functions such as tape drive cleaning.
- For WORM tape support.
- Future NetBackup features and enhancements

Note: Because NetBackup uses its own pass-through driver, NetBackup does not support the Solaris `sgen` SCSI pass-through driver.

See "Installing/reinstalling the sg and the st drivers" on page 22.

Determining if the NetBackup sg driver is installed

Use the following procedure to determine if the `sg` driver is installed and loaded.

More information about the driver is available.

See "About the NetBackup sg driver" on page 19.

To determine if the sg driver is installed and loaded

- Invoke the following command:

  ```
  /usr/sbin/modinfo | grep sg
  ```

  If the driver is loaded, output includes a line similar to the following:

  ```
  57 113d1d00 3760 316 1 sg (SCSA Generic Revision: 3.7a)
  ```

Special configuration for the StorEdge Network Foundation HBA driver

When you configure the `sg` driver, it binds the StorEdge Network Foundation host bus adapter World Wide Port Names for use by the `sg` driver.

See "Installing/reinstalling the sg and the st drivers" on page 22.

The configuration process uses the Solaris `luxadm` command to probe for HBAs that are installed in the system. Ensure that the `luxadm` command is installed and in the shell path. For Solaris 11 and later, NetBackup uses the Solaris `sasinfo` command to probe for SAS attached devices.

To determine if a host contains a StorEdge Network Foundation HBA, you can run the following command:

```
/usr/openv/volmgr/bin/sgscan
```

If the script detects a StorEdge Network Foundation HBA, it produces output similar to the following example:

```
#WARNING: detected StorEdge Network Foundation connected devices not
        in sg configuration file:
#
#    Device World Wide Port Name 21000090a50001c8
#
#    See /usr/openv/volmgr/NetBackup_DeviceConfig_Guide.txt topic
#    "Special configuration for Sun StorEdge Network Foundation
#    HBA/Driver" for information on how to use sg.build and
#    sg.install to configure these devices
```

Each time you add or remove a device, you should configure the NetBackup `sg` driver and the Sun `st` driver again.

See "About the NetBackup sg driver" on page 19.

For 6 GB and larger serial attached SCSI (SAS) HBAs, also configure class 08 and 0101 for the `sg` driver.

See "Configuring 6 GB and larger SAS HBAs in Solaris" on page 27.

About binding Fibre Channel HBA drivers

For Fibre Channel HBAs other than StorEdge Network Foundation, you must bind the devices to specific target IDs on the NetBackup host. When you bind devices to targets, the target ID does not change after a system reboot or a Fibre Channel configuration change.

In some instances, Cohesity products are configured to use a specific target ID. If you change the ID, the products fail until you configure the ID correctly.

How you bind devices to targets is vendor and product specific. For information about how to modify the HBA configuration files to bind devices to targets, see the documentation for the HBA.

The binding may be based on the following:

- Fibre Channel World Wide Port Name (WWPN)
- World Wide Node Name (WWNN)
- The destination target ID and LUN

After you bind the devices to target IDs, continue with the Solaris configuration in the same manner as for parallel SCSI installations.

See "Installing/reinstalling the sg and the st drivers" on page 22.

Each time you add or remove a device, you must update the bindings and then configure the `sg` and the `st` drivers again.

Configuring Solaris 10 x86 for multiple drive paths

To use multiple paths to the same tape drive, NetBackup requires that Solaris Multiplexed I/O (MPxIO) be disabled. MPxIO is enabled by default on Solaris 10 x86 systems.

Use the following procedure to disable MPxIO.

To disable MPxIO

1. Use a text editor to open the following file:

   `/kernel/drv/fp.conf`

2. Change the `mpxio-disable` value from **no** to **yes**. After the change, the line in the file should appear as follows:

   `mpxio-disable="yes"`

3. Save the changes and exit from the text editor.

Installing/reinstalling the sg and the st drivers

You must install the NetBackup `sg` driver and the Sun `st` driver on each Solaris NetBackup media server that hosts tape devices.

Each time you add or remove a device, you should configure the NetBackup `sg` driver and the Sun `st` driver again. For 6 GB and larger serial-attached SCSI (SAS) HBAs, also configure class 08 and 0101 for the `sg` driver.

See "Configuring 6 GB and larger SAS HBAs in Solaris" on page 27.

Before you configure the `sg` and the `st` drivers, ensure that all devices are turned on and connected to the HBA.

See "About the NetBackup sg driver" on page 19.

The `sg.build` command uses the Solaris `sasinfo` command to probe for SAS attached device paths. This command is only available on Solaris 11 and later. On Solaris 10 and earlier, you must configure the `sg` driver manually.

To install and configure the sg and the st drivers

1 Invoke the following two commands to run the NetBackup `sg.build` script:

```
cd /usr/openv/volmgr/bin/driver
/usr/openv/volmgr/bin/sg.build all -mt target -ml lun
```

The following describes the options:

- The `all` option creates the following files and populates them with the appropriate entries:
  - `/usr/openv/volmgr/bin/driver/st.conf`
    See "st.conf file example" on page 25.
  - `/usr/openv/volmgr/bin/driver/sg.conf`
    See "sg.conf file example" on page 25.
  - `/usr/openv/volmgr/bin/driver/sg.links`
    See "sg.links file example" on page 26.
- The `-mt` *`target`* option and argument specify the maximum target ID that is in use on the SCSI bus (or bound to an FCP HBA). The maximum value is 126. By default, the SCSI initiator target ID of the adapter is 7, so the script does not create entries for target ID 7.

- The `-ml` *`lun`* option and argument specify the maximum number of LUNs that are in use on the SCSI bus (or by an FCP HBA). The maximum value is 255.

2 Replace the following seven entries in the `/kernel/drv/st.conf` file with all of the entries from the `/usr/openv/volmgr/bin/driver/st.conf` file:

```
name="st" class="scsi" target=0 lun=0;
name="st" class="scsi" target=1 lun=0;
name="st" class="scsi" target=2 lun=0;
name="st" class="scsi" target=3 lun=0;
name="st" class="scsi" target=4 lun=0;
name="st" class="scsi" target=5 lun=0;
name="st" class="scsi" target=6 lun=0;
```

You should make a backup copy of the `/kernel/drv/st.conf` file before you modify it.

3 Reboot the system with the reconfigure option (`boot -r` or `reboot -- -r`).

During the boot process, the system probes all targets in the `st.conf` file for devices. It should create device files for all of the devices it discovers.

4 Verify that Solaris created the device nodes for all the tape devices by using the following command:

```
ls -l /dev/rmt/*cbn
```

5 Install the new `sg` driver configuration by invoking the following two commands:

```
/usr/bin/rm -f /kernel/drv/sg.conf
/usr/openv/volmgr/bin/driver/sg.install
```

The NetBackup `sg.install` script does the following:

- Installs and loads the `sg` driver.
- Copies the `/usr/openv/volmgr/bin/driver/sg.conf` file to `/kernel/drv/sg.conf`.
- Creates the `/dev/sg` directory and nodes.
- Appends the `/usr/openv/volmgr/bin/driver/sg.links` file to the `/etc/devlink.tab` file.

6 Verify that the <command>sg</command> driver finds all of the robots and tape drives.

st.conf file example

The following `/usr/openv/volmgr/bin/driver/st.conf` file example shows targets 0-15 and LUNs 0-7.

```
name="st" class="scsi" target=0 lun=0;
name="st" class="scsi" target=0 lun=1;
name="st" class="scsi" target=0 lun=2;
name="st" class="scsi" target=0 lun=3;
name="st" class="scsi" target=0 lun=4;
name="st" class="scsi" target=0 lun=5;
name="st" class="scsi" target=0 lun=6;
name="st" class="scsi" target=0 lun=7;
name="st" class="scsi" target=1 lun=0;
name="st" class="scsi" target=1 lun=1;
name="st" class="scsi" target=1 lun=2;
.
<entries omitted for brevity>
.
name="st" class="scsi" target=15 lun=5;
name="st" class="scsi" target=15 lun=6;
name="st" class="scsi" target=15 lun=7;
```

sg.conf file example

The following /usr/openv/volmgr/bin/driver/sg.conf file example shows targets 0-15 and LUNs 0-8. It also includes target entries for three StorEdge Network Foundation HBA ports.

The `sg.build -mt` option does not affect FCP targets, but the -ml option does. The Solaris luxadm command detected three ports (identified by their World Wide Names). Therefore, the `sg.build` script created entries for LUNs 0 through 7 for those three ports.

```
name="sg" class="scsi" target=0 lun=0;
name="sg" class="scsi" target=0 lun=1;
name="sg" class="scsi" target=0 lun=2;
name="sg" class="scsi" target=0 lun=3;
name="sg" class="scsi" target=0 lun=4;
name="sg" class="scsi" target=0 lun=5;
name="sg" class="scsi" target=0 lun=6;
name="sg" class="scsi" target=0 lun=7;
name="sg" class="scsi" target=1 lun=0;
name="sg" class="scsi" target=1 lun=1;
```

```
name="sg" class="scsi" target=1 lun=2;
...
<entries omitted for brevity>
...
name="sg" class="scsi" target=15 lun=5;
name="sg" class="scsi" target=15 lun=6;
name="sg" class="scsi" target=15 lun=7;
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53c3";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53c3";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53c6";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53c6";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53c9";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53c9";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53cc";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53cc";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53b9";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53b9";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53c3";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53c3";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53c6";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53c6";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53c9";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53c9";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53cc";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53cc";
name="sg" parent="fp" target=0 lun=0 fc-port-wwn="500104f0008d53b9";
name="sg" parent="fp" target=0 lun=1 fc-port-wwn="500104f0008d53b
```

sg.links file example

The following `/usr/openv/volmgr/bin/driver/sg.links` file example shows targets 0-15 and LUNs 0-7. It also includes entries for three StorEdge Network Foundation HBA ports.

The `sg.build -mt` option does not affect FCP targets, but the -ml option does. The Solaris luxadm command detected three ports (identified by their World Wide Names). Therefore, the `sg.build` script created entries for LUNs 0 through 7 for those three ports.

The field separator between the `addr=`*`x`*`, `*`y`*`;` field and the `sg/` field is a tab. The `addr=` field uses hexadecimal notation, and the `sg/` field uses decimal values.

```
# begin SCSA Generic devlinks file - creates nodes in /dev/sg
type=ddi_pseudo;name=sg;addr=0,0;       sg/c\N0t0l0
```

```
type=ddi_pseudo;name=sg;addr=0,1;       sg/c\N0t0l1
type=ddi_pseudo;name=sg;addr=0,2;       sg/c\N0t0l2
type=ddi_pseudo;name=sg;addr=0,3;       sg/c\N0t0l3
type=ddi_pseudo;name=sg;addr=0,4;       sg/c\N0t0l4
type=ddi_pseudo;name=sg;addr=0,5;       sg/c\N0t0l5
type=ddi_pseudo;name=sg;addr=0,6;       sg/c\N0t0l6
type=ddi_pseudo;name=sg;addr=0,7;       sg/c\N0t0l7
type=ddi_pseudo;name=sg;addr=1,0;       sg/c\N0t1l0
type=ddi_pseudo;name=sg;addr=1,1;       sg/c\N0t1l1
...
<entries omitted for brevity>
...
type=ddi_pseudo;name=sg;addr=f,5;       sg/c\N0t15l5
type=ddi_pseudo;name=sg;addr=f,6;       sg/c\N0t15l6
type=ddi_pseudo;name=sg;addr=f,7;       sg/c\N0t15l7
type=ddi_pseudo;name=sg;addr=w500104f0008d53c3,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53c3,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53c6,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53c6,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53c9,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53c9,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53cc,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53cc,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53b9,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53b9,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53c3,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53c3,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53c6,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53c6,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53c9,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53c9,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53cc,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53cc,1;       sg/c\N0t\A1l1
type=ddi_pseudo;name=sg;addr=w500104f0008d53b9,0;       sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f0008d53b9,1;       sg/c\N0t\A1l1
# end SCSA devlinks
```

Configuring 6 GB and larger SAS HBAs in Solaris

Use the procedure in this topic to configure the NetBackup `sg` driver for Oracle 6 GB and larger SAS HBAs on Solaris.

A separate topic describes how to install the NetBackup `sg` and Sun `st` drivers.

See "Installing/reinstalling the sg and the st drivers" on page 22.

Note: Support for Solaris 6 GB serial-attached SCSI (SAS) HBAs for tape devices requires a specific Solaris patch level. Ensure that you install the required patches. For supported Solaris versions, see the Oracle Support website.

To configure 6 GB and larger SAS HBAs in Solaris

1. Verify that the 6 GB SAS tape device path exists by running the following command in a shell window:

   ```
   ls -l /dev/rmt | grep cbn
   ```

   6 GB SAS tape devices should have `iport@` in the name path. The following is an example of the output (the tape drive address is highlighted):

```
1cbn -> ../../devices/pci@400/pci@0/pci@9/LSI,sas@0/iport@8/tape@w500104f000ba856a,0:cbn
```

2. Edit the `/etc/devlink.tab` file

   Include the following lines for every 6 GB SAS tape drive in the `/etc/devlink.tab` file. Replace *drive_address* with the tape drive address; see the output from step 1 for the tape drive addresses.

```
type=ddi_pseudo;name=sg;addr=wdrive_address,0,1;      sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=wdrive_address,1,1;      sg/c\N0t\A1l1
```

   Include the following lines for every 6 GB SAS robotic library in the `/etc/devlink.tab` file. Replace *drive_address* with the tape drive address; see the output from step 1 for the tape drive address.

```
type=ddi_pseudo;name=medium-changer;addr=wdrive_address,0;    sg/c\N0t\A1l0
type=ddi_pseudo;name=medium-changer;addr=wdrive_address,1;    sg/c\N0t\A1l1
```

   The following are example entries for the `devlink.tab` file:

```
# SCSA devlinks for SAS-2 drives:
type=ddi_pseudo;name=sg;addr=w500104f000ba856a,0,1;      sg/c\N0t\A1l0
type=ddi_pseudo;name=sg;addr=w500104f000ba856a,1,1;      sg/c\N0t\A1l1
# SCSA devlinks for SAS-2 libraries:
type=ddi_pseudo;name=medium-changer;addr=w500104f000ba856a,0;    sg/c\N0t\A1l0
type=ddi_pseudo;name=medium-changer;addr=w500104f000ba856a,1;    sg/c\N0t\A1l1
```

3. Verify that the sg driver SCSI classes are 08 and 0101 by running the following command:

   ```
   grep sg /etc/driver_aliases
   ```

   The following is an example of the output:

   ```
   sg "scsiclass,0101"
   sg "scsiclass,08"
   ```

4. If the `sg` driver SCSI classes are not 08 and 0101, reinstall the `sg` driver by using the following commands:

   ```
   rem_drv sg
   update_drv -d -i "scsiclass,08" sgen
   add_drv -m '* 0600 root root' -i '"scsiclass,0101" "scsiclass,08"' sg
   ```

5. Restart the host.
6. Verify that the `sg` drivers exist by running the following command:

   ```
   ls -l /dev/sg
   ```

   The following is an example of the output (the output was modified to fit on the page):

```
c0tw500104f000ba856al0 ->
 ../../devices/pci@400/pci@0/pci@9/LSI,sas@0/iport@8/sg@w500104f000ba856a,0,1:raw
c0tw500104f000ba856al1 ->
 ../../devices/pci@400/pci@0/pci@9/LSI,sas@0/iport@8/medium-changer@w500104f000ba856a,1:raw
```

7. Verify that the NetBackup `sgscan` utility recognizes the tape devices by entering the following command:

   ```
   /usr/openv/volmgr/bin/sgscan
   ```

   The following is an example of the output:

```
/dev/sg/c0tw500104f000ba856al0: Tape (/dev/rmt/1): "HP      Ultrium 5-SCSI"
/dev/sg/c0tw500104f000ba856al1: Changer: "STK     SL500"
```

Preventing Solaris driver unloading

When system memory is limited, Solaris unloads unused drivers from memory and reloads drivers as needed. Tape drivers are often unloaded because they are used less often than disk drivers.

The drivers NetBackup uses are the st driver (from Sun), the `sg` driver (from Cohesity), and Fibre Channel drivers. Problems may occur depending on when the driver loads and unloads. These problems can range from a SCSI bus not able to detect a device to system panics.

Cohesity recommends that you prevent Solaris from unloading the drivers from memory.

The following procedures describe how to prevent Solaris from unloading the drivers from memory.

To prevent Solaris from unloading the drivers from memory

- Add the following `forceload` statements to the `/etc/system` file:

```
forceload: drv/st
forceload: drv/sg
```

To prevent Solaris from unloading the Fibre Channel drivers from memory

- Add an appropriate `forceload` statement to the `/etc/system` file.

  Which driver you force to load depends on your Fibre Channel adapter. The following is an example for a Sun Fibre Channel driver (SunFC FCP v20100509-1.143):

```
forceload: drv/fcp
```

About Solaris robotic controls

NetBackup supports SCSI control and API control of robotic devices. A robotic device in a library moves the media between the storage slots and the drives in the library.

Robotic control varies, as follows:

- SCSI or Fibre Channel Protocol control.
  See “About SCSI and FCP robotic controls on Solaris” on page 30.
- API control over a LAN.
  See the "Oracle StorageTek ACSLS robots" topic in this guide.

About SCSI and FCP robotic controls on Solaris

When you configure the NetBackup `sg` driver, a NetBackup script creates the device files for the attached robotic devices.

See “About the NetBackup sg driver” on page 19.

If you use device discovery in NetBackup, NetBackup discovers the robotic control device files in the `/dev/sg` directory (and hence the devices) automatically. If you add a robot manually in NetBackup, you must enter the pathname to the device file.

To display the device files that the `sg` driver can use, use the NetBackup `sgscan` command with the `all` parameter. The word "Changer" in the `sgscan` output identifies robotic control device files.

Examples are available.

See "Examples of SCSI and FCP robotic control device files on Solaris" on page 31.

Examples of SCSI and FCP robotic control device files on Solaris

The following is an example of `sgscan all` output from a host, to which the examples refer:

```
# /usr/openv/volmgr/bin/sgscan all
/dev/sg/c0t6l0: Cdrom: "TOSHIBA XM-5401TASUN4XCD"
/dev/sg/c1tw500104f0008d53b9l0: Changer: "STK     SL500"
/dev/sg/c1tw500104f0008d53c3l0: Tape (/dev/rmt/0): "HP      Ultrium 3-SCSI"
/dev/sg/c1tw500104f0008d53c6l0: Tape (/dev/rmt/1): "HP      Ultrium 3-SCSI"
/dev/sg/c1tw500104f0008d53c9l0: Tape (/dev/rmt/2): "IBM     ULTRIUM-TD3"
/dev/sg/c1tw500104f0008d53ccl0: Tape (/dev/rmt/3): "IBM     ULTRIUM-TD3"
/dev/sg/c2t1l0: Changer: "STK     SL500"
/dev/sg/c2t2l0: Tape (/dev/rmt/22): "HP      Ultrium 3-SCSI"
/dev/sg/c2t3l0: Tape (/dev/rmt/10): "HP      Ultrium 3-SCSI"
/dev/sg/c2tal0: Tape (/dev/rmt/18): "IBM      ULTRIUM-TD3"
/dev/sg/c2tbl0: Tape (/dev/rmt/19): "IBM      ULTRIUM-TD3"
/dev/sg/c3t0l0: Disk (/dev/rdsk/c1t0d0): "FUJITSU MAV2073RCSUN72G"
/dev/sg/c3t3l0: Disk (/dev/rdsk/c1t3d0): "FUJITSU MAV2073RCSUN72G"
```

You can filter the `sgscan` output for device types by using other sgscan options. The following is the `sgscan` usage statement:

```
sgscan [all|basic|changer|disk|tape] [conf] [-v]
```

About Solaris tape drive device files

NetBackup uses the tape drive device files that support compression, no rewind on close, and Berkeley style close.

When you configure the Solaris `st` driver, Solaris creates the device files for the attached tape devices

See "Installing/reinstalling the sg and the st drivers" on page 22.

The device files are in the `/dev/rmt` directory, and they have the following format:

`/dev/rmt/`*`ID`*`cbn`

The following describe the device file names:

- *`ID`* is the logical drive number as shown by the NetBackup `sgscan` command.
- `c` indicates compression.
- `b` indicates Berkeley-style close.
- `n` indicates no rewind on close.

If you use device discovery in NetBackup, NetBackup discovers the device files and hence the devices. If you add a tape drive to a NetBackup configuration manually, you must specify the pathname to the device file. NetBackup requires compression, no rewind on close, and Berkeley-style close device files.

To display the tape device files that are configured on your system, use the `sgscan` command with the `tape` parameter, as follows:

```
# /usr/openv/volmgr/bin/sgscan tape
/dev/sg/c1tw500104f0008d53c3l0: Tape (/dev/rmt/0): "HP      Ultrium 3-SCSI"
/dev/sg/c1tw500104f0008d53c6l0: Tape (/dev/rmt/1): "HP      Ultrium 3-SCSI"
/dev/sg/c1tw500104f0008d53c9l0: Tape (/dev/rmt/2): "IBM     ULTRIUM-TD3"
/dev/sg/c1tw500104f0008d53ccl0: Tape (/dev/rmt/3): "IBM     ULTRIUM-TD3"
/dev/sg/c2t2l0: Tape (/dev/rmt/22): "HP      Ultrium 3-SCSI"
/dev/sg/c2t3l0: Tape (/dev/rmt/10): "HP      Ultrium 3-SCSI"
/dev/sg/c2tal0: Tape (/dev/rmt/18): "IBM      ULTRIUM-TD3"
/dev/sg/c2tbl0: Tape (/dev/rmt/19): "IBM      ULTRIUM-TD3"
```

The following are examples of no-rewind, compression, Berkeley-style close device files from the preceding `sgscan` example output:

- For the Ultrium3 SCSI drive at LUN 0 of World Wide Node Name (WWNN) 500104f0008d53c3, the device file pathname is:

  `/dev/rmt/0cbn`

- For the HP Ultrium3 SCSI drive at SCSI ID 2 of adapter 2, the device file pathname is:

  `/dev/rmt/22cbn`

You can show all device types by using the all option. The output can help you associate tape devices with other SCSI devices that may be configured on the same adapter. The following is the `sgscan` usage statement:

```
sgscan [all|basic|changer|disk|tape] [conf] [-v]
```

About Berkeley-style close

NetBackup requires Berkeley-style close for tape drive device files. The letter `b` in the file name indicates Berkeley-style close device files.

In Berkeley-style close, the tape position remains unchanged by a device close operation. (Conversely, in AT&T-style close, the drive advances the tape to just after the next end-of-file (EOF) marker.) To establish the correct position for the next tape operation, applications must assume the tape's position after a close. NetBackup assumes Berkeley-style close on Solaris systems.

About no rewind device files on Solaris

NetBackup requires no rewind on close device files for tape drives.

With no rewind on close, the tape is not rewound after a close operation. It remains in position for the next write operation.

The letter `n` in the device file names in the `/dev/rmt` directory specifies no rewind on close.

About fast-tape positioning (locate-block) on Solaris

Applies to AIT, DLT, Exabyte, DTF, and half-inch tape drives.

To position a tape to a specific block, NetBackup supports the SCSI `locate-block` command. It requires the NetBackup `sg` driver.

NetBackup uses the locate-block command by default.

Cohesity recommends that you do not disable locate-block positioning. If you need to disable it, execute the following command:

```
touch /usr/openv/volmgr/database/NO_LOCATEBLOCK
```

If locate-block positioning is disabled, NetBackup uses the forward-space-file/record method.

About SPC-2 SCSI reserve on Solaris

By default, NetBackup uses SPC-2 SCSI reserve and release for tape drive reservations in shared drive environments. The NetBackup Shared Storage Option provides shared drive functionality in NetBackup.

Alternatively, you can use SCSI persistent reserve for shared tape drive reservations in NetBackup, as follows:

- For the tape drives that support SPC-3 Compatible Reservation Handling (CRH), you can use SCSI persistent reserve by enabling it in NetBackup. No special configuration in Solaris is required.
- For the tape drives that do not support CRH, you must disable SPC-2 SCSI reserve in Solaris for those drives. After you disable SPC-2 SCSI reserve, you can use persistent reserve by enabling it in NetBackup. If the drive does not support CRH and you do not disable SPC-2 SCSI reserve, access attempts to the drive fail.
  See "Disabling SPC-2 SCSI reserve on Solaris" on page 34.

For more information about NetBackup and SCSI reservations, see the following:

- The description of the **Enable SCSI Reserve Media** host property in the NetBackup Web UI Administrator's Guide, Volume I.
- The "How NetBackup reserves drives" topic in the NetBackup Administrator's Guide, Volume II.

Disabling SPC-2 SCSI reserve on Solaris

Use the following procedure to disable SPC-2 SCSI reserve.

More information about reservations is available.

See "About SPC-2 SCSI reserve on Solaris" on page 33.

To disable SPC-2 SCSI reserve

- Modify the Solaris `st.conf` file on the NetBackup media server. In the `tape-config-list` section of the `st.conf` file, set the `ST_NO_RESERVE_RELEASE` configuration value (0x20000) in the appropriate `data-property-name` entry.

  For example, the following entry disables SCSI reserve and release for all tape drives that use the `DLT7k-data` configuration values:

  ```
  DLT7k-data = 1,0x38,0,0x20000,4,0x82,0x83,0x84,0x85,2;
  ```

  For more information about the `st.conf` file, see the Solaris `st`(7D) man page.

About nonstandard tape drives

Solaris includes the device drivers that support most standard devices.

To receive the most current support for devices, you should install the latest Solaris patch for the `st` driver.

However, if you have a device that Solaris does not support, the device manufacturer should provide the software to install and administer the device properly. In addition, the device vendor should contact Oracle to add support for the device to Solaris.

For more information about what you need for unsupported devices, contact the device vendor. Also see the Solaris devices and file systems documentation.

Configuring Solaris SAN clients to recognize FT media servers

NetBackup SAN clients use tape drivers and SCSI pass-through methods for Fibre Transport traffic to NetBackup FT media servers. The media server FT devices appear as `ARCHIVE Python` tape devices during SCSI inquiry on the SAN client. However, they are not tape devices and do not appear as tape devices in NetBackup device discovery.

Cohesity owns the `ARCHIVE` brand name and `Python` product name. Therefore, `st.conf` file changes to `ARCHIVE Python` do not affect an existing tape drive product.

Table 3-1 is an overview of procedures to configure the Solaris operating system so that it recognizes the NetBackup FT devices on the NetBackup media servers.

Table 3-1 Configuring SAN clients to recognize FT media servers

Step	Task	Procedure
1	Add the Fibre Transport device entry to the `st.conf` file	See "Adding the FT device entry to the st.conf file" on page 35.
2	Modify the `st.conf` file so that Solaris discovers devices on two LUNS	See "Modifying the st.conf file so that Solaris discovers devices on two LUNS" on page 36.

Adding the FT device entry to the st.conf file

The following procedure describes how to add the FT device entry to the `st.conf` file.

To add the FT device entry to the st.conf file

1. In the `/kernel/drv/st.conf` file, find the `tape-config-list=` section or create it if it does not exist.
2. Examine the `tape-config-list=` section for a line that begins with `ARCHIVE Python` and contains `ARCH_04106`. If such a line exists, ensure that it begins with a comment character (`#`).

3 Add the following line to the `tape-config-list=` section:

```
"ARCHIVE Python", "FT Pipe", "ARCH_04106";
```

4 Find the line that begins with `ARCH_04106`, copy it, and paste it after the `tape-config-list=` line. Delete the comment character (`#`) from the beginning of the line. The following is an example of the line:

```
ARCH_04106 = 1, 0x2C, 0, 0x09639, 4, 0x00, 0x8C, 0x8c, 0x8C, 3;
```

Modifying the st.conf file so that Solaris discovers devices on two LUNS

The following procedure describes how to modify the `st.conf` file so that Solaris discovers devices on two LUNS.

To modify the st.conf file so that Solaris discovers devices on two LUNS

1 Find the following line in the `st.conf` file:

```
name="st" class="scsi" target=0 lun=0;
```

2 Replace that line and the following lines through target 5 with the following. Doing so modifies the `st.conf` file to include searches on non-zero LUNs.

```
name="st" class="scsi" target=0 lun=0;
name="st" class="scsi" target=0 lun=1;
name="st" class="scsi" target=1 lun=0;
name="st" class="scsi" target=1 lun=1;
name="st" class="scsi" target=2 lun=0;
name="st" class="scsi" target=2 lun=1;
name="st" class="scsi" target=3 lun=0;
name="st" class="scsi" target=3 lun=1;
name="st" class="scsi" target=4 lun=0;
name="st" class="scsi" target=4 lun=1;
name="st" class="scsi" target=5 lun=0;
name="st" class="scsi" target=5 lun=1;
name="st" parent="fp" target=0;
name="st" parent="fp" target=1;
name="st" parent="fp" target=2;
name="st" parent="fp" target=3;
name="st" parent="fp" target=4;
name="st" parent="fp" target=5;
name="st" parent="fp" target=6;
```

Uninstalling the sg driver on Solaris

You can uninstall the `sg` driver. If you do, NetBackup performance suffers. The following procedure describes how to uninstall the `sg` driver.

To uninstall the sg driver

- Invoke the following command:

  ```
  /usr/sbin/rem_drv sg
  ```

Solaris command summary

The following is a summary of commands that may be useful when you configure and verify devices:

- `/usr/sbin/modinfo | grep sg`
  Displays whether or not the `sg` driver is installed.
- `/usr/openv/volmgr/bin/driver/sg.install`
  Installs the `sg` driver or updates the `sg` driver.
- `/usr/sbin/rem_drv sg`
  Uninstalls the `sg` driver. This command usually is not necessary because `sg.install` uninstalls the old driver before it upgrades a driver.
- `/usr/openv/volmgr/bin/sg.build all -mt` *`max_target`* `-ml` *`max_lun`*
  Updates `st.conf`, `sg.conf`, and `sg.links`, and generates SCSI Target IDs with multiple LUNs.
- `/usr/openv/volmgr/bin/sgscan all`
  Scans all connected devices with an SCSI inquiry and provides correlation between physical and the logical devices that use all device files in `/dev/sg`.
  Also checks for the devices that are connected to the StorEdge Network Foundation HBA that are not configured for use by Cohesity products.
- `boot -r or reboot -- -r`
  Reboot the system with the reconfigure option (-r). The kernel's SCSI disk (sd) driver then recognizes the drive as a disk drive during system initialization.

See the procedures in this chapter for examples of their usage.

Chapter 4

Windows

This chapter includes the following topics:

- Before you begin configuring NetBackup on Windows
- About tape device drivers on Windows
- Attaching devices to a Windows system

Before you begin configuring NetBackup on Windows

Observe the following points when performing the configurations described in this chapter:

- Verify that NetBackup supports your server platform and devices. Download the NetBackup hardware and operating system compatibility lists: https://support.cohesity.com/s/article/article-100040093
- For NetBackup to recognize and communicate with connected devices and for device discovery to discover devices, NetBackup issues SCSI pass-through commands to the devices in a configuration.
  A tape driver must exist for each tape device. Attached devices appear in the registry.
- Use the Microsoft Windows device applications to verify that the devices are configured correctly. The device applications available on your server may differ depending on your Windows operating system. Make sure that Windows detects the devices on the SAN before you configure the NetBackup Shared Storage Option.
- If you have multiple devices connected to a fibre bridge, Windows may only see one LUN. This will normally be the device with the lowest-ordered LUN.

This limitation occurs because of the default install settings for the device driver for some fibre channel HBAs. See your vendor documentation to verify the settings.

- Information about how to configure API robot control over a LAN is available See the "Oracle StorageTek ACSLS robots" topic in this guide.

After configuring the hardware, add the drives and robots to NetBackup.

About tape device drivers on Windows

Cohesity does not provide device drivers for Windows hosts. If you require drivers, contact Microsoft or the tape drive vendor.

Attaching devices to a Windows system

The following procedure describes a general method for attaching devices to a Windows computer. The Microsoft Windows device applications available on the server that you use in these steps may differ depending on your Windows operating system.

To attach devices to a Windows system

1. Use the appropriate Windows application to obtain information on any currently attached SCSI devices.
2. If you attach a new robotic library or drive to a NetBackup media server, follow the vendor's instructions for attaching the device.

   Shut down the server and physically attach the supported device. Ensure that SCSI targets and termination settings are consistent with adapter card and peripheral vendor recommendations.
3. Reboot the server and answer the prompts for adapter card peripheral configuration options. Watch the display to ensure that the adapter card recognizes the attached peripherals.
4. If you add drives, install the tape drivers and use the appropriate Windows application to verify that the drive was recognized.

Section 2

Robotic storage devices

- Chapter 5. Robot overview
- Chapter 6. Oracle StorageTek ACSLS robots
- Chapter 7. Device configuration examples

Chapter 5

Robot overview

This chapter includes the following topics:

- NetBackup robot types
- NetBackup robot attributes
- Table-driven robotics
- Robotic test utilities
- Robotic processes

NetBackup robot types

A robot is a peripheral device that moves tape volumes into and out of tape drives. NetBackup uses robotic control software to communicate with the robot firmware.

NetBackup classifies robots according to one or more of the following characteristics:

- The communication method the robotic control software uses; SCSI and API are the two main methods.
- The physical characteristics of the robot. Library refers to a large robot, in terms of slot capacity or number of drives.
- The media type commonly used by that class of robots. HCART (1/2-inch cartridge tape) is an example of a media type.

The table lists the NetBackup robot types that are supported in release 11.2.0.1, with drive and slot limits for each type.

To determine which robot type applies to the model of robot that you use, see the NetBackup Enterprise Server and Server - Hardware and Cloud Storage Compatibility List for your release.

Table 5-1 NetBackup robot types in release 11.2.0.1

Robot type	Description	Drive limits	Slot limits	Note
ACS	Automated Cartridge System	1680	No limit	API control. The ACS library software host determines the drive limit.
TLD	Tape library DLT	No limit	32000	SCSI control.

Note: The user interface for NetBackup may show configuration options for the peripheral devices that are not supported in that release. Those devices may be supported in an earlier release, and a NetBackup primary server can manage the hosts that run earlier NetBackup versions. Therefore, the configuration information for such devices must appear in the user interface. The NetBackup documentation may also describe the configuration information for such devices. To determine which versions of NetBackup support which peripheral devices, see the NetBackup Enterprise Server and Server - Hardware and Cloud Storage Compatibility List.

NetBackup robot attributes

NetBackup configures and controls robots differently depending on the robot type. The following tables list the attributes that dictate how these robot types differ.

For more detailed information about supported devices, firmware levels, and platforms, see the hardware compatibility list for your NetBackup version:

https://support.cohesity.com/s/article/article-100040093

See “NetBackup robot types” on page 41.

ACS robots

Unlike other robot types, NetBackup does not track slot locations for the media in ACS robots. The ACS library software tracks slot locations and reports them to NetBackup.

The following table describes the ACS robot attributes.

Table 5-2 ACS robot attributes

Attribute	NetBackup server
API robot	Yes
SCSI control	No

Table 5-2 ACS robot attributes *(continued)*

Attribute	NetBackup server
LAN control	Yes
Remote Robot control	No. Each host that has ACS drives that are attached to it has robotic control.
NDMP support	Yes
Shared drives support	Yes
Drive cleaning support	No. The ACS library software manges drive cleaning.
Media access port support	Yes, for eject only.
NetBackup tracks slots	No
Media type support	DLT, DLT2, DLT3, HCART, HCART2, and HCART3.
Hosts Supported	Windows, UNIX, and Linux. Windows servers require STK LibAttach software. See the Cohesity support web site for the latest compatibility information and obtain the appropriate LibAttach software from STK.
Barcode Support	Yes. Depends on ACS library software to obtain NetBackup media IDs. Barcodes must be the same as the media ID (1 to 6 characters).
Robot Examples	Oracle SL500, Oracle SL3000, and Oracle SL8500

TLD robots

The following table describes the tape library DLT attributes.

Table 5-3 TLD robot attributes

Attribute	NetBackup Server	NetBackup Enterprise Server
API robot	No	No
SCSI control	Yes	Yes
LAN control	Not Applicable	No

Table 5-3 TLD robot attributes *(continued)*

Attribute	NetBackup Server	NetBackup Enterprise Server
Remote robot control	Not Applicable	Yes
NDMP support	Yes	Yes
Shared drives support	Not Applicable	Yes
Drive cleaning support	Yes	Yes
Media access port support	Yes	Yes
NetBackup tracks slots	Yes	Yes
Hosts supported	Windows, UNIX, and Linux.	Windows, UNIX, and Linux.
Media type support	DLT, DLT2, DLT3 HCART, HCART2, HCART3	DLT, DLT2, DLT3, HCART, HCART2, HCART3
Barcode support	Yes. Barcodes can be from 1 to 16 characters in length. The Media Manager media ID is six or fewer characters.	Yes. Barcodes can be from 1 to 16 characters in length. The Media Manager media ID is six or fewer characters.
Robot examples	HPE MSL, Fujitsu FibreCAT TX48, IBM TotalStorage3583, Spectra Logic T680, Sun/Oracle SL3000	HPE MSL, Fujitsu FibreCAT TX48, IBM TotalStorage3583, Spectra Logic T680, Sun/Oracle SL3000

Table-driven robotics

Table-driven robotics provides support for new robotic library devices without the need to modify any library control binary files. This feature uses a device mapping file for supported robots and drives.

You may be able to add support for new or upgraded devices without waiting for a maintenance patch from Cohesity. The device mapping file includes the information that relates to the operation and control of libraries. Therefore, you can download an updated mapping file to obtain support for newly NetBackup-certified devices.

For the device mappings file downloads, see the following URL:

https://support.cohesity.com/s/article/article-100040093

See “NetBackup robot types” on page 41.

Robotic test utilities

You can use robotic test utilities for testing robots already configured in NetBackup.

Invoke the test utilities as follows:

- `/usr/openv/volmgr/bin/robtest` (UNIX and Linux)
- *`install_path`*`\Veritas\Volmgr\bin\robtest.exe` (Windows)

From each test utility, you can obtain a list of available test commands by entering a question mark (?).

Use the `drstat` command to determine the drive addressing parameters for the ACS robot type. This command is available in the robotic test utilities for these robot types.

NetBackup addresses drives as follows:

- For ACS robot types, by ACS, LSM, Panel, and Drive number
- For other robot types, by the robot drive number

See "NetBackup robot types" on page 41.

Robotic processes

A NetBackup robotic process and possibly a robotic control process exist on a NetBackup media server for each robot that you install, as follows:

- Every media server that has a drive in a robotic library has a robotic process for that robotic library. The robotic process receives requests from the NetBackup Device Manager (`ltid`) and sends necessary information directly to the robotics or to a robotic control process.
- Robotic control processes exist only for the robot types that support library sharing (or robot sharing).

When the NetBackup Device Manager starts, it starts the robotic processes and the robotic control processes for all of the configured robots on that host. When the Device Manager stops, the robotic processes and the robotic control processes stop. (On UNIX, the name is Media Manager Device daemon.)

You can start and stop the Device Manager manually from the **NetBackup web UI** in one of the following ways:

- On the left, click **Activity Monitor** and then click the **Daemons** tab. Select `ltid` and then click **Start** or **Stop**.

- On the left, click **Storage > Media servers** and then click the **Media servers** tab. Select the media server, then click **Stop/Restart media manager device daemon**.

In addition, the NetBackup Commands Reference Guide describes commands to control the robotic processes that run on Windows media servers.

You can determine if a robotic process or robotic control process is active by in the **Processes** tab of the **Activity monitor**.

You can determine the control state of a device in the **Device monitor**. On the left click **Storage > Tape storage** and click on the **Device monitor** tab. If the value in the **Control** column for a drive shows the control mode, the robotic process is running and the drive is usable. For example, for a TLD robot the control mode is **TLD**.

Other values such as **AVR** or **DOWN** may indicate that the drive is unusable.

See "Processes by robot type" on page 46.

See "Robotic process example" on page 47.

See "NetBackup robot types" on page 41.

Processes by robot type

The following table describes the robotic processes and robotic control processes for each robot type.

Table 5-4 Robotic processes and robotic control processes

Robot type	Process	Description
Automated Cartridge System (ACS)	acsd	The NetBackup ACS daemon `acsd` provides robotic control to mount and dismount volumes. It also requests inventories of the volumes that are under the control of ACS library software.
	acssel	The NetBackup ACS storage server interface (SSI) event logger `acssel` logs events. UNIX and Linux only.
	acsssi	The NetBackup ACS storage server interface (SSI) `acsssi` communicates with the ACS library software host. `acsssi` processes all RPC communications from `acsd` or from the ACS robotic test utility that are intended for the ACS library software. UNIX and Linux only.
Tape library DLT (TLD)	tldd	The tape library DLT daemon `tldd` runs on a NetBackup server that has a drive in the tape library DLT. This process receives NetBackup Device Manager requests to mount and unmount volumes, and sends these requests to the robotic-control process, `tldcd`.

Table 5-4 Robotic processes and robotic control processes *(continued)*

Robot type	Process	Description
	tldcd	The tape library DLT Control daemon `tldcd` communicates with the tape library DLT robotics through a SCSI interface. For library sharing, `tldcd` runs on the NetBackup server that has the robotic control.

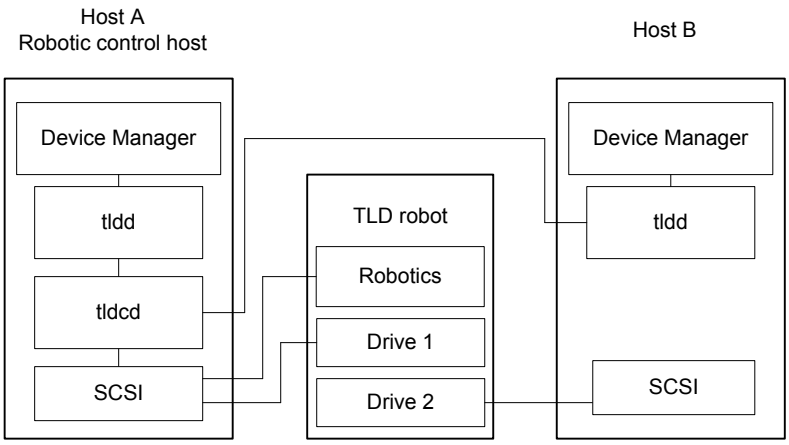
See "NetBackup robot types" on page 41.

Robotic process example

Each drive in a tape library DLT (TLD) robot can be attached to a different host, and a `tldd` process runs on each host. However, only one host controls the robotics, and the `tldcd` robotic control process runs on that host only. To mount a tape, the `tldd` process on the host to which the drive is attached sends control information to the `tldcd` process on the robotic control host.

The following figure shows the processes and where they run for a TLD robot.

Figure 5-1 TLD robot control process example



The following describes this example:

- Each host connects to one drive, and a `tldd` robotic process runs on each host.
- The robotic control and therefore the robotic control process, `tldcd`, is on host A.

The NetBackup Device Manager services on host A and B start `tldd`. The `tldd` process on host A also starts `tldcd`. Requests to mount tapes from host B go to `tldd` on host B, which then sends the robotic command to `tldcd` on host A.

See “NetBackup robot types” on page 41.

Chapter 6

Oracle StorageTek ACSLS robots

This chapter includes the following topics:

- About Oracle StorageTek ACSLS robots
- Sample ACSLS configurations
- Media requests for an ACS robot
- About configuring ACS drives
- Configuring shared ACS drives
- Adding tapes to ACS robots
- About removing tapes from ACS robots
- Robot inventory operations on ACS robots
- NetBackup robotic control, communication, and logging
- ACS robotic test utility
- Changing your ACS robotic configuration
- ACS configurations supported
- Oracle StorageTek ACSLS firewall configuration

About Oracle StorageTek ACSLS robots

Note: If you use the access control feature of Oracle StorageTek ACSLS controlled robots and the NetBackup media sharing feature, do the following: ensure that all servers in the NetBackup media server share group have the same ACSLS permissions to all the same ACSLS media and ACSLS drives. Any mismatches can cause failed jobs and stranded tapes in drives.

Oracle StorageTek Automated Cartridge System Library Software controlled robots are NetBackup robot type ACS.

ACS robots are API robots (a NetBackup robot category in which the robot manages its own media).

Unlike other robot types, NetBackup does not track slot locations for the media in ACS robots. The Automated Cartridge System Library Software tracks slot locations and reports them to NetBackup.

The term automated cartridge system (ACS) can refer to any of the following:

- A type of NetBackup robotic control.
- The Oracle StorageTek system for robotic control.
- The highest-level component of the Oracle StorageTek ACSLS. It refers to one robotic library or to multiple libraries that are connected with a media pass-through mechanism.

The ACS library software component can be either of the following Oracle StorageTek products:

- Oracle StorageTek Automated Cartridge System Library Software (ACSLS)
- Oracle StorageTek Library Station

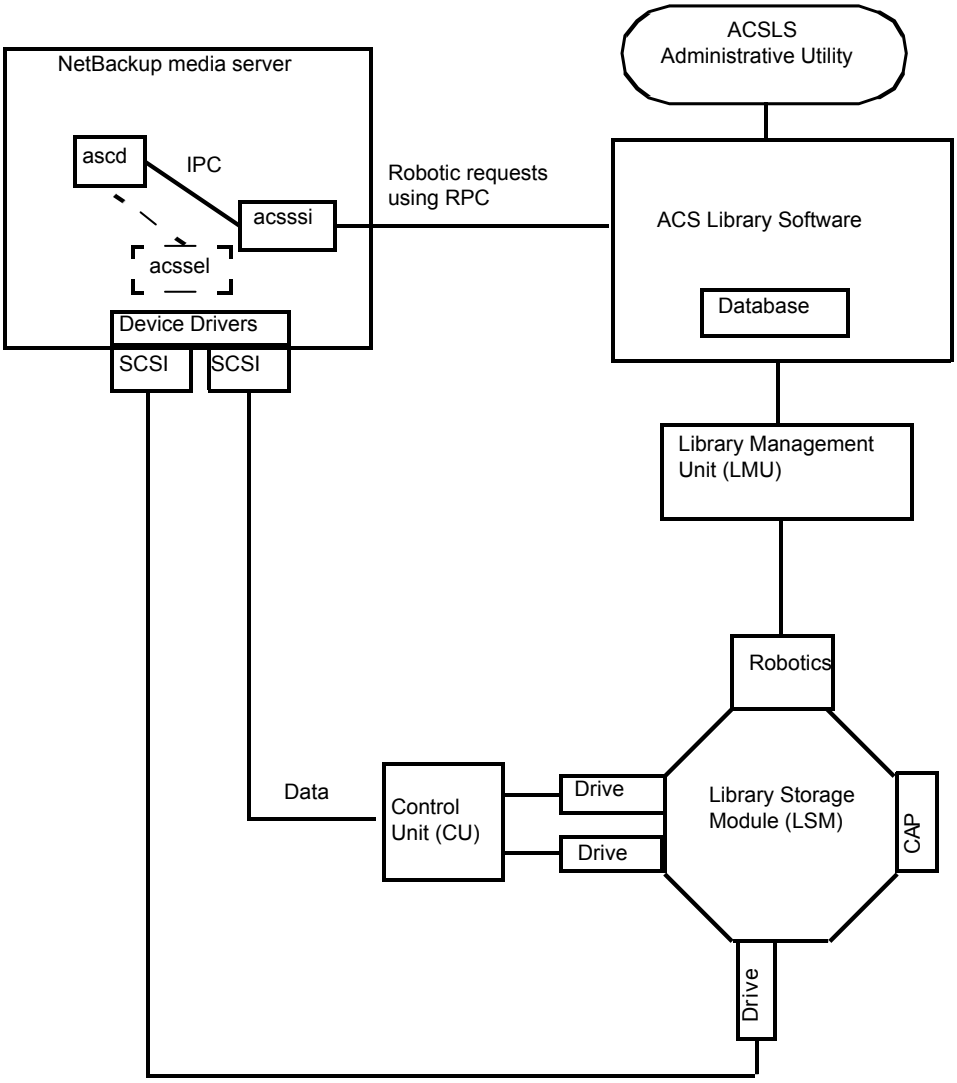
Sample ACSLS configurations

The sample ACSLS configurations show the following:

- A typical UNIX ACSLS configuration.
  See Figure 6-1 on page 51.
- A typical Windows ACSLS configuration.
  See Figure 6-2 on page 52.
- The major components in typical configurations.
  See Table 6-1 on page 53.

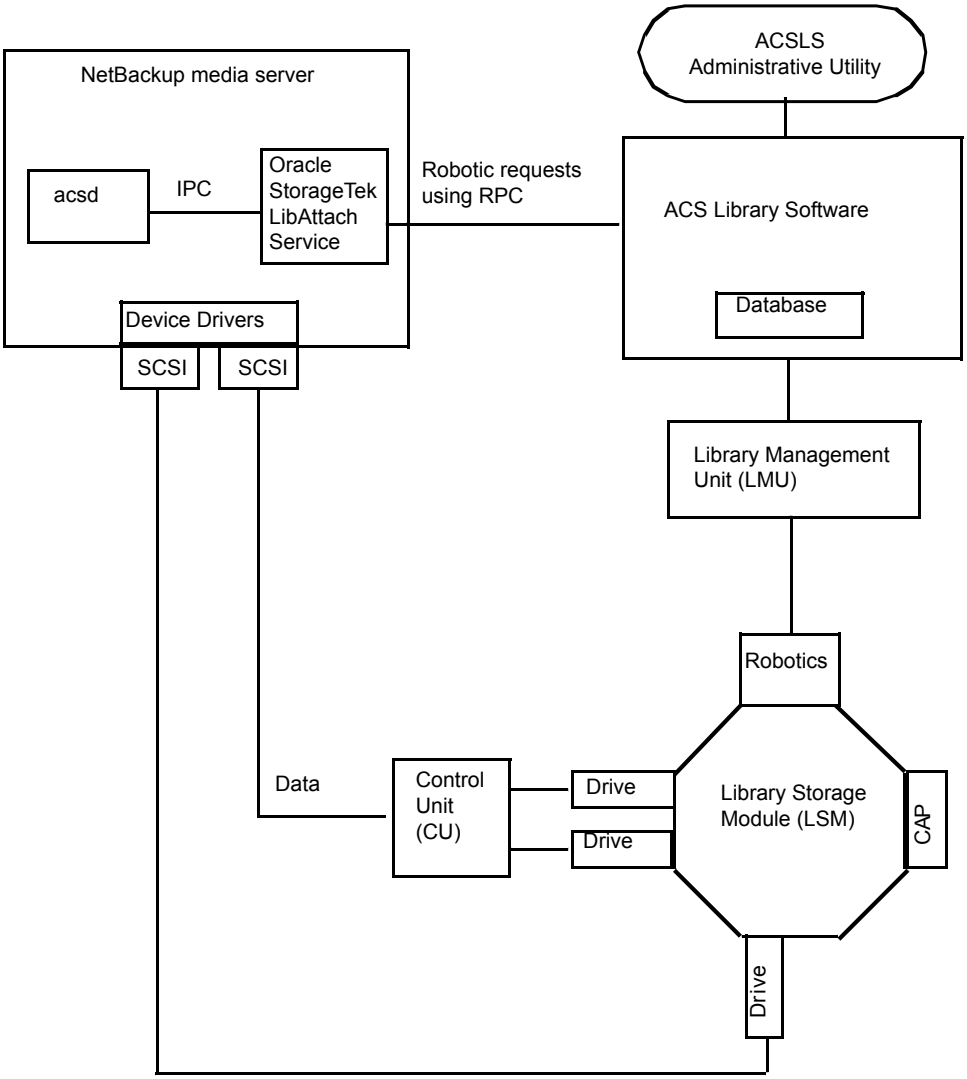
The following figure shows a typical UNIX ACSLS configuration.

Figure 6-1 Typical ACSLS configuration on UNIX



The following figure shows a typical Windows ACSLS configuration.

Figure 6-2 Typical ACSLS configuration on Windows



The following table describes the components of the ACSLS configuration.

Table 6-1 ACSLS configuration component description

Component	Description
NetBackup media server	Specifies a host that has NetBackup media server software and is a client to the ACS library software host. The NetBackup ACS robotic daemon (`acsd`) formulates requests for mounts, unmounts, and inventories. An API then uses IPC communication to routes these requests to: ■ (UNIX) The NetBackup ACS storage server interface (`acsssi`). The requests are converted into RPC-based communications and sent to the ACS library software. ■ (Windows) the Oracle StorageTek LibAttach service. This service sends the requests to the ACS library software.
Oracle StorageTek LibAttach Service Windows computers only	Specifies that Library Attach for Windows, an ACS library software client application, enables Windows servers to use the StorageTek Nearline enterprise storage libraries. LibAttach provides the connection between Windows and ACS library software through a TCP/IP network. Obtain the appropriate LibAttach software from Oracle. See the Cohesity support Web site for the latest compatibility information.
The following ACS library software: ■ Automated Cartridge System Library Software (ACSLS) ■ Oracle StorageTek Library Station	Receives the robotic requests from NetBackup and uses the Library Management Unit to find and mount or unmount the correct cartridge on media management requests. On compatible host platforms, you may be able to configure ACS library software and NetBackup media server software on the same host.
Library Management Unit (LMU)	Provides the interface between the ACS library software and the robot. A single LMU can control multiple ACSLS robots.
Library Storage Module (LSM)	Contains the robot, drives, or media.
Control Unit (CU)	Specifies that the NetBackup media server connects to the drives through device drivers and a control unit (tape controller). The control unit may have an interface to multiple drives. Some control units also allow multiple hosts to share these drives. Most drives do not require a separate control unit. In these cases, the media server connects directly to the drives.
CAP	Specifies the Cartridge Access Port.

Media requests for an ACS robot

The following is the sequence of events for a media request for an ACS robot:

- The Media Manager device daemon (UNIX) or NetBackup Device Manager service (Windows) `ltid` receives the request from `bptm`.
- `ltid` sends a mount request to the NetBackup ACS process `acsd`.
- `acsd` formulates the request.

  An API then uses Internal Process Communications (IPC) to send the request on the following systems:

  - UNIX. The NetBackup ACS storage server interface `acsssi`. The request is then converted into RPC-based communications and sent to the ACS library software.
  - Windows. The Oracle StorageTek LibAttach service. This service sends the request to the ACS library software.
- If the Library Storage Module (LSM) in which the media resides is offline, the ACS library software reports this offline status to NetBackup. NetBackup assigns the request a pending status. NetBackup retries the request hourly until the LSM is online and the ACS library software can satisfy the media request.
- The ACS library software locates the media and sends the necessary information to the Library Management Unit (LMU).
- The LMU directs the robotics to mount the media in the drive. When the LibAttach service (Windows) or `acsssi` (UNIX) receives a successful response from the ACS library software, it returns the status to `acsd`.
- The `acsd` child process (that is associated with the mount request) scans the drive. When the drive is ready, `acsd` sends a message to `ltid` that completes the mount request. NetBackup then begins to send data to or read data from the drive.

About configuring ACS drives

An ACS robot supports DLT or 1/2-inch cartridge tape drives. If an ACS robot contains more than one type of DLT or 1/2-inch cartridge tape drive, you can configure an alternate drive type. Therefore, there can be up to three different DLT and three different 1/2-inch cartridge drive types in the same robot. If you use alternate drive types, configure the volumes by using the same alternate media type. Six drive types are possible: DLT, DLT2, DLT3, HCART, HCART2, and HCART3.

Before you configure drives in NetBackup, configure the operating system tape drivers and device files for those drives. For information about how to do so, refer to the operating system documentation. For guidance about the NetBackup requirements, see the information about the host operating system in this guide

Use the same methods to create or identify device files for these drives as for other drives. If the drives are SCSI and connect to the robot through a shared control unit, the drives share the same SCSI ID. Therefore, you must specify the same logical unit number (LUN) for each drive.

When you configure ACS drives as robotic in NetBackup, you must include the ACS drive coordinate information.

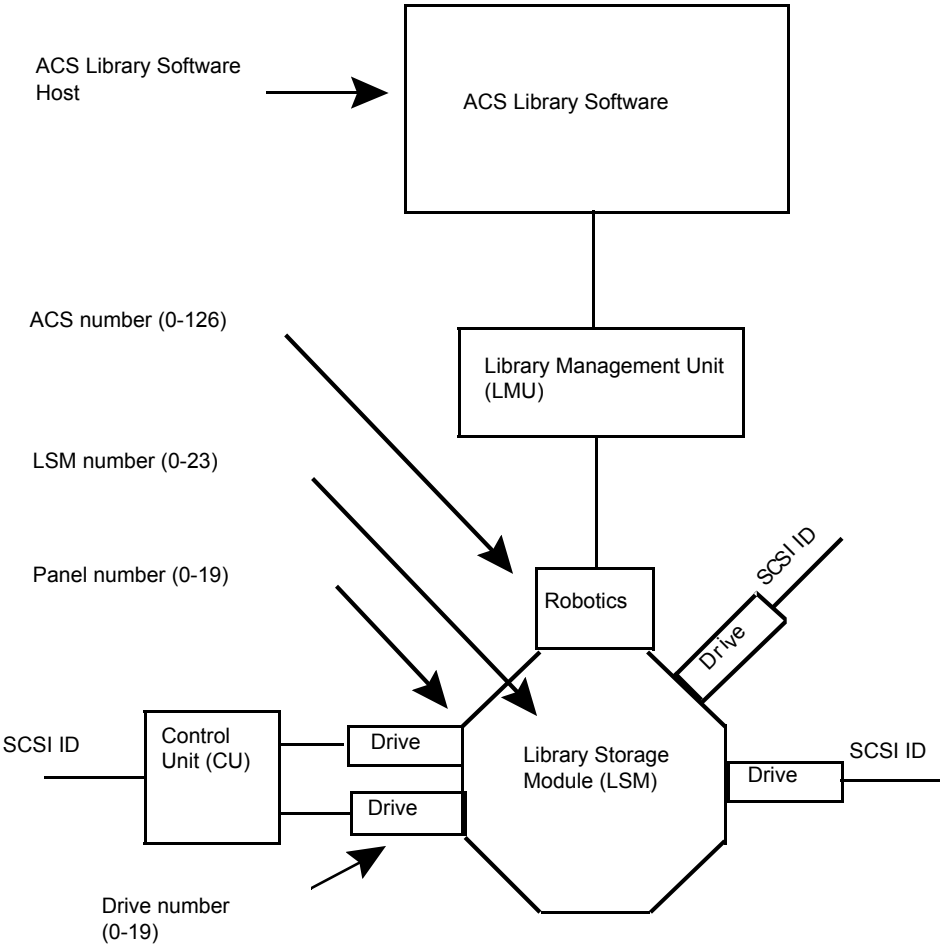
The following table shows the ACS drive coordinates.

Table 6-2 ACS drive coordinates

ACS drive coordinate	Description
ACS number	Specifies the index, in ACS library software terms, that identifies the robot that has this drive.
LSM number	Specifies the Library Storage Module that has this drive.
Panel number	Specifies the panel where the drive is located.
Drive number	Specifies the physical number of the drive in ACS library software terms.

The following figure shows the location of this information in a typical ACS robot.

Figure 6-3 ACSLS robot and drive configuration information



Configuring shared ACS drives

If the ACSLS server does not support serialization, use the following procedure to configure shared drives. Shared drives require the NetBackup Shared Storage Option license. Oracle StorageTek ACSLS versions before 6.1 do not support serialization.) If the server supports serialization, use the NetBackup **Configure drives and robots** wizard to configure shared drives.

This procedure can significantly reduce the amount of manual configuration that is required in an SSO environment. For example, for 20 drives that 30 hosts share,

these configuration steps require that you configure only 20 device paths rather than 600 device paths.

During the setup phase, the NetBackup **Configure drives and robots** wizard tries to discover the tape drives available. The wizard also tries to discover the positions of the drives within the library (if the robot supports serialization).

A SAN (including switches rather than direct connection) can increase the possibility of errors. If errors occur, you can define the tape drive configuration manually by using the NetBackup web UI or the NetBackup commands.

Take care to avoid any errors. With shared drives, the device paths must be correct for each server. Also, ensure that the drives are defined correctly to avoid errors. (A common error is to define a drive as ACS index number 9 rather than ACS index 0.)

Use the following procedure to configure shared drives in a nonserialized configuration.

To configure shared drives in a nonserialized configuration

1. Run the NetBackup **Configure drives and robots** wizard on one of the hosts to which drives in an ACS-controlled library are attached. Allow the drives to be added as standalone drives.
2. Add the ACS robot definition and update each drive to indicate its position in the robot. Make each drive robotic and add the ACS, LSM, Panel, and Drive information.

   Information is available about how to determine the correct drive addresses and how to verify the drive paths. See "Correlating device files to physical drives" in the NetBackup Web UI Administrator's Guide.
3. After you verify the drive paths on one host, run the **Configure drives and robots** wizard again. Scan all hosts that have ACS drives in the library.

   The wizard adds the ACS robot definition and the drives to the other hosts and uses the correct device paths.

   For this process to work correctly, the following must be true:

   - The wizard discovered the devices and their serial numbers successfully the first time.
   - You configured the drive paths correctly on the first host.

Adding tapes to ACS robots

ACS robotic control software supports the following characters in a volume ID that are not valid NetBackup media ID characters. (Volume ID is the ACS term for media ID).

Therefore, do not use any of the following characters when you configure ACS volumes:

- Dollar sign ($)
- Pound sign (#)
- The yen symbol
- Leading and trailing spaces

The following table is an overview of how to add tapes to an ACS robot and then add those tapes to NetBackup.

Table 6-3 Adding tapes to ACS robots process

Task	Description
Add barcode labels to the media and insert the media into the robot by using the media access port.	The Library Manager reads the barcodes and classifies the media by media type. A category is assigned to each volume. Some volume categories restrict application access to certain volumes. The Library Manager tracks volume locations.
Define the media in NetBackup by using the ACS volume IDs as media IDs.	To define the media, do one of the following: ■ Update the volume configuration by using the **Inventory robot** function. ■ Add new volumes. See the NetBackup Web UI Administrator's Guide. Because the ACS volume IDs and barcodes are the same, NetBackup has a record of the barcodes for the media. Note that you do not enter slot locations because the ACS library software manages slot locations.
Verify the volume configuration.	Use **Show contents** and **Compare contents with volume configuration** options in the **Inventory robot** function.

About removing tapes from ACS robots

You can remove tapes by using the Oracle StorageTek utility or by using NetBackup.

See “Removing tapes from an ACS robot” on page 59.

See “Removing tapes using NetBackup” on page 59.

Removing tapes from an ACS robot

If you remove media from an ACS robot, you must move the media logically to standalone in NetBackup.

If you do not move media logically, NetBackup does not know that the media were moved. NetBackup may issue mount requests for it, which causes a misplaced tape error.

However, you can move media from one location to another within the robot. The ACS library software finds the requested media if its database is current.

To remove tapes from an ACS robot

- Perform one of the following actions:
  - Update the volume configuration by using the **Inventory robot** function.
  - Move the volumes.

  See the NetBackup Web UI Administrator's Guide for details.

Removing tapes using NetBackup

Both of these methods perform the logical move and the physical move.

To remove tapes using NetBackup

- Use one of the following methods:
  - On the left, select **Storage > Tape storage > Volumes**. Locate the volume and select **Eject from robot**.
  - Use the NetBackup `vmchange` command.
    NetBackup Commands Reference Guide

Robot inventory operations on ACS robots

If the ACS library software host is a Oracle StorageTek Library Station, an Inventory Robot Filter (`INVENTORY_FILTER`) entry may be required in the `vm.conf` file. Old versions of Library Station do not support queries of all volumes in an ACS robot.

In NetBackup, the ACS robot type supports bar codes.

The following sequence of events occurs when you inventory an ACS robot in NetBackup:

- NetBackup requests volume information from the ACS library software.
- The ACS library software provides a listing of the volume IDs, media types, ACS location, and LSM location from its database.

See Table 6-4 on page 60.

- NetBackup maps the volume IDs into media IDs and bar codes. For example in the previous table, volume ID 100011 becomes media ID 100011 and the barcode for that media ID is also 100011.
- If the operation does not require a volume configuration update, NetBackup uses the media type defaults for ACS robots when it creates its report.
- If the operation requires a volume configuration update, NetBackup does the following:
  - Maps the ACS media types to the default NetBackup media types.
  - Adds the ACS and the LSM locations for new volumes to the EMM database. This location information is used for media and drive selection.

Information about the default media type mappings and how to configure media type mappings is available.

See the NetBackup Administrator's Guide, Volume I.

The following table shows an example of the ACS drive coordinates that NetBackup receives.

Table 6-4 ACS drive coordinates

ACS volume ID	ACS media type	ACS	LSM
100011	DLTIV	0	0
200201	DD3A	0	0
412840	STK1R	0	1
412999	STK1U	0	1
521212	JLABEL	0	0
521433	STK2P	0	1
521455	STK2W	0	1
770000	LTO_100G	0	0
775500	SDLT	0	0
900100	EECART	0	0
900200	UNKNOWN	0	0

Configuring a robot inventory filtering on ACS robots

If you want NetBackup to use only a subset of the volumes under ACS library control, you can filter the volume information from the library. To do so, you use the ACSLS administrative interface to assign the volumes you want to use to a scratch pool or pools. Then you configure NetBackup to use only the volumes in those scratch pools.

A NetBackup robot inventory includes the volumes that exist in the ACS scratch pool. The ACS library software moves each volume from the scratch pool after it is mounted.

A partial inventory also includes those volumes that NetBackup can validate exist in the robotic library, including volumes not in the ACS scratch pool. To prevent losing track of previously mounted volumes, the library reports the complete list of volumes that exist in the robotic library.

The following procedure is an example of how to configure an inventory filter.

To configure an inventory filter (example)

1. Use the ACSLS administrative interface (ACSSA) command to create a scratch pool. Assign ID 4 and 0 to 500 as the range for the number of volumes, as follows:

   ```
   ACSSA> define pool 0 500 4
   ```

2. Use the ACSLS administrative interface (ACSSA) command to define the volumes in scratch pool 4:

   ```
   ACSSA> set scratch 4 600000-999999
   ```

3. On the NetBackup media server from which you invoke the inventory operation, add an INVENTORY_FILTER entry to the vm.conf file. The following is the usage statement:

   ```
   INVENTORY_FILTER = ACS robot_number BY_ACS_POOL acs_scratch_pool1
   [acs_scratch_pool2 ...]
   ```

   The following define the options and arguments:

   - *robot_number* is the number of the robot in NetBackup.
   - *acs_scratch_pool1* is the scratch pool ID as configured in the ACS library software.
   - *acs_scratch_pool2* is a second scratch pool ID (up to 10 scratch pools are allowed).

For example, the following entry forces ACS robot number 0 to query scratch volumes from Oracle StorageTek pool IDs 4 and 5.

```
INVENTORY_FILTER = ACS 0 BY_ACS_POOL 4 5
```

NetBackup robotic control, communication, and logging

How NetBackup uses robotic control, communication, and logging during tape operations depends on the operating system type as follows:

- Windows systems
  See "NetBackup robotic control, communication, and logging for Windows systems" on page 62.
- UNIX systems
  See "NetBackup robotic control, communication, and logging for UNIX systems" on page 62.

NetBackup robotic control, communication, and logging for Windows systems

The NetBackup `acsd` process provides robotic control to mount and dismount volumes. It also requests inventories of the volumes that are under the control of ACS library software. The NetBackup Device Manager service `ltid` starts the `acsd` process and communicates with it.

The `acsd` process requests SCSI tape unloads through the device host's tape driver before it uses the ACS API to request that tape dismounts. This request process accommodates the configurations that have SCSI multiplexors. Loaded tapes are not ejected forcibly when a dismount operation occurs.

NetBackup robotic control, communication, and logging for UNIX systems

On UNIX systems, several NetBackup daemons and processes provide robotic control, communication, and logging.

NetBackup ACS daemon (acsd)

The NetBackup ACS daemon `acsd` provides robotic control to mount and dismount volumes. It also requests inventories of the volumes that are under the control of

ACS library software. the Media Manager device daemon `ltid` starts the `acsd` daemon and communicates with it. If `ltid` is active already, you can start `acsd` manually.

The `acsd` daemon requests SCSI tape unloads through the device host's tape driver before it uses the ACS API to request that tape dismounts. This control process accommodates the configurations that have SCSI multiplexors. Loaded tapes are not ejected forcibly when a dismount operation occurs.

When `acsd` starts, it first starts the NetBackup `acssel` process and then starts the `acsssi` process. When it starts `acsssi`, `acsd` passes the ACS library software host name to `acsssi`. One copy of `acsssi` starts for each ACS library software host that appears in the NetBackup device configuration for the media server. If multiple media servers share drives in an ACS robot, `acsssi` must be active on each media server.

NetBackup ACS SSI event logger (acssel)

The NetBackup ACS storage server interface (SSI) event logger `acssel` is modeled after the Oracle StorageTek `mini_el` event logger. Therefore, its functional model differs from other NetBackup robotic controls.

The NetBackup `acsd` daemon starts `acssel` automatically. You also can start it manually. Event messages are logged to the following file:

```
/usr/openv/volmgr/debug/acsssi/event.log
```

Note: Cohesity recommends that `acssel` run continuously because it tries to connect on the event logger's socket for its message logging. If `acsssi` cannot connect to `acssel`, NetBackup cannot process requests immediately. Therefore, retry and error recovery situations can occur.

On UNIX systems, only the `kill` command stops `acssel`. The NetBackup `bp.kill_all` utility (UNIX) stops the `acssel` process. On Windows systems, the `bpdown.exe` program stops the `acssel` process.

The full path to the event logger is `/usr/openv/volmgr/bin/acssel`. The usage format is as follows:

```
acssel [-d] -s socket_name
```

The following describes the options:

- -d displays debug messages (by default, debug messages are disabled).
- *socket_name* is the socket name (or IP port) to listen on for messages.

Using acssel with a different socket name

If the `vm.conf` file does not contain an `ACS_SEL_SOCKET` entry, `acssel` listens on socket name 13740 by default.

You can change this default by using one of the following methods:

- Modify the `vm.conf` configuration file.
  See To change the default by modifying the vm.conf configuration file.
- Add environment variables. This method assumes that one ACS robot is configured and that the SSI default socket name has not been changed. (The `vm.conf ACS_SEL_SOCKET` entry can change the default).
  See To change the default by adding environment variables.
  `acssel` also has a command line option to specify the socket name. However, because `acsssi` needs to know the event logger socket name, setting an environment variable is preferred.

To change the default by modifying the vm.conf configuration file

1. Edit the `vm.conf` file and add an `ACS_SEL_SOCKET` entry. The following is an example:

   ```
   ACS_SEL_SOCKET = 13799
   ```

2. Stop the `acsd`, `acsssi`, and `acssel` processes by invoking the following script. (This script stops all NetBackup processes.)

   ```
   /usr/openv/NetBackup/bin/bp.kill_all
   ```

3. Restart the NetBackup daemons and processes by invoking the following script:

   ```
   /usr/openv/NetBackup/bin/bp.start_all
   ```

To change the default by adding environment variables

1. Stop the `acsd`, `acsssi`, and `acssel` processes by invoking the following script. (This script stops all NetBackup processes.)

   ```
   /usr/openv/NetBackup/bin/bp.kill_all
   ```

2. Set the wanted socket name in an environment variable and export it. The following is an example:

   ```
   ACS_SEL_SOCKET = 13799
   export ACS_SEL_SOCKET
   ```

3. Start the event logger in the background.

   ```
   /usr/openv/volmgr/bin/acssel &
   ```

4 Set the ACS library software host name for `acsssi` in an environment variable.

```
CSI_HOSTNAME = einstein
export CSI_HOSTNAME
```

5 Start `acsssi` as follows:

```
/usr/openv/volmgr/bin/acsssi 13741 &
```

6 Optionally, start `acstest` by using the `robtest` utility or by using the following command:

```
/usr/openv/volmgr/bin/acstest -r einstein -s 13741
```

If you request SCSI unloads, you also must specify drive paths on the `acstest` command line.

See "ACS robotic test utility" on page 67.

The `robtest` utility specifies drive paths automatically if ACS drives have been configured.

7 Start `ltid` as follows, which starts `acsd`. You can use the `-v` option for verbose message output.

```
/usr/openv/volmgr/bin/ltid
```

During initialization, `acsd` obtains the SSI Event Logger socket name from `vm.conf` and sets `ACS_SEL_SOCKET` in the environment before it starts `acssel`. If `acsssi` is started manually, it has to use (listen on) the same SSI socket that `acsd` uses to send data.

NetBackup ACS storage server interface (acsssi)

The NetBackup ACS storage server interface (SSI) `acsssi` communicates with the ACS library software host. `acsssi` processes all RPC communications from `acsd` or from the ACS robotic test utility that are intended for the ACS library software.

One copy of `acsssi` must run for each unique ACS library software host that is configured on a NetBackup media server. `acsd` tries to start copies of `acsssi` for each host. However, if an `acsssi` process for a specific ACS library software host exists already, the new `acsssi` processes for that host fails during initialization.

In normal operations, `acsssi` runs in the background and sends log messages to `acssel`.

You can specify the socket name (IP port) used by `acsssi` in any of the following ways:

- On the command line when you start `acsssi`.

- By using an environment variable (ACS_SSI_SOCKET).
- Through the default value.

If you configure acsssi to use a nondefault socket name, you also must configure the ACS daemon and ACS test utility to use the same socket name.

The ACS library software host name is passed to acsssi by using the CSI_HOSTNAME environment variable.

acsssi is based on the Oracle StorageTek storage server interface. Therefore, it supports environment variables to control most aspects of operational behavior.

See "Optional environment variables" on page 67.

About the ACS_SSI_SOCKET configuration option

By default, acsssi listens on unique, consecutive socket names; the socket names begin at 13741. To specify socket names on an ACS library software host basis, you can add a configuration entry in the NetBackup vm.conf file.

Use the following format:

```
ACS_SSI_SOCKET = ACS_library_software_hostname socket_name
```

The following is an example entry (do not use the IP address of the ACS library host for this parameter):

```
ACS_SSI_SOCKET = einstein 13750
```

Starting acsssi manually

This method is not the recommended method to start acsssi. Normally, acsd starts acsssi.

Before you can start acsssi manually, you must configure the CSI_HOSTNAME environment variable. The following is a Bourne shell example:

```
CSI_HOSTNAME=einstein
export CSI_HOSTNAME
/usr/openv/volmgr/bin/acsssi 13741 &
```

Use the following procedure to start acsssi.

To start acsssi

1. Start the event logger, acssel.
2. Start acsssi. The format is acsssi *socket_name*.

Optional environment variables

If you want individual NetBackup `acsssi` processes to operate differently, you can set environment variables before the `acsssi` processes are started.

The following table describes the optional environment variables.

Table 6-5 Optional environment variables

Environment variable	Description
`SSI_HOSTNAME`	Specifies the name of the host where ACS library software RPC return packets are routed for ACS network communications. By default, the local host name is used.
`CSI_RETRY_TIMEOUT`	Set this variable to a small positive integer. The default is 2 seconds.
`CSI_RETRY_TRIES`	Set this variable to a small positive integer. The default is five retries.
`CSI_CONNECT_AGETIME`	Set this variable to a value between 600 seconds and 31536000 seconds. The default is 172800 seconds.

ACS robotic test utility

The `acstest` utility lets you verify ACS communications and provides a remote system administrative interface to an ACS robot. It can also be used to query, enter, eject, mount, unload, and dismount volumes. In addition, `acstest` lets you define, delete, and populate ACS library software scratch pools.

While `acsd` services requests, do not use `acstest`. Communication problems may occur if `acsd` and `acstest` process ACS requests at the same time.

acstest on Windows systems

`acstest` depends on the Oracle StorageTek LibAttach service being started successfully. You can verify that this service is started by using the Services tool available in administrative tools in the Windows control panel. `acstest` attempts to communicate with ACS library software by using the `LibAttach` service.

The usage format follows:

`acstest -r` *`ACS_library_software_hostname`* `[-d` *`device_name ACS, LSM, panel, drive`*`] ... [-C` *`sub_cmd`*`]`

The following example assumes that the `LibAttach` service started:

`install_path``\Volmgr\bin\acstest -r einstein -d Tape0 0,0,2,1`

acstest on UNIX systems

`acstest` depends on `acsssi` being started successfully. You can use the UNIX `netstat -a` command to verify that a process listens on the SSI socket. `acstest` attempts to communicate with ACS library software using `acsssi` and connects on an existing socket.

The usage format follows. You can pass the socket name on the command line. Otherwise, the default socket name (13741) is used.

```
acstest -r ACS_library_software_hostname [-s socket_name] [-d
drive_path ACS, LSM, panel, drive] ... [-C sub_cmd]
```

The following example assumes that the `acsssi` process has been started by using socket 13741:

```
/usr/openv/volmgr/bin/acstest -r einstein -s 13741
```

Changing your ACS robotic configuration

UNIX and Linux systems only.

If you change your ACS robot configuration, you should update NetBackup so that `acsssi` can successfully communicate with `acsd`, `acstest`, and ACS library software.

Any `acsssi` processes must be canceled after your changes are made and before the Media Manager device daemon `ltid` is restarted. Also, for the `acstest` utility to function, `acsssi` for the selected robot must be running.

Use the following procedure to update NetBackup after you change your configuration.

To update NetBackup after you change your configuration

1. Make your configuration changes.
2. Use `/usr/openv/NetBackup/bin/bp.kill_all` to stop all running processes.
3. Restart the NetBackup daemons and processes by invoking the following script:

   `/usr/openv/NetBackup/bin/bp.start_all`

ACS configurations supported

UNIX and Linux systems only.

NetBackup supports the following ACS configurations:

- Multiple robots that are controlled from a single ACS host
  See "Multiple ACS robots with one ACS library software host" on page 69.

- Multiple robots that are controlled from multiple ACS hosts
  See "Multiple ACS robots and ACS library software hosts" on page 69.

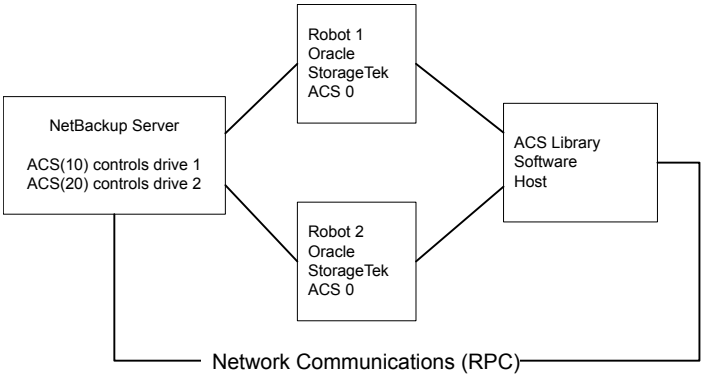
Multiple ACS robots with one ACS library software host

NetBackup supports the following configuration:

- A NetBackup server is connected to drives in multiple ACS robots.
- The robots are controlled from a single ACS library software host.

The following figure shows multiple ACS robots that are controlled from a single ACS library software host.

Figure 6-4 Multiple ACS robots, one ACS library software host



Inventory requests include: the volumes that are configured on the ACS library software host that resides on the ACS robot that is designated in the drive address.

In this example, assume the following about drive 1:

- Has an ACS drive address (ACS, LSM, panel, drive) of 0,0,1,1 in the NetBackup device configuration
- Is under control of robot number 10 (ACS(10)).

If any other robot ACS(10) drives have a different ACS drive address (for example, 1,0,1,0), the configuration is invalid.

NetBackup supports configurations of multiple LSMs in a single ACS robot if a pass-through port exists.

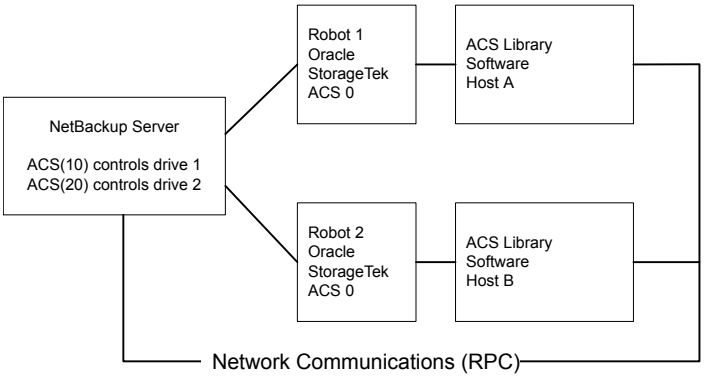
Multiple ACS robots and ACS library software hosts

NetBackup supports the following configuration:

- A NetBackup server is connected to drives in multiple ACS robots.
- The robots are controlled from separate ACS library software hosts.

The following figure shows multiple ACS robots that are controlled from multiple ACS library software hosts.

Figure 6-5 Multiple ACS robots, multiple ACS library software hosts



Inventory requests include the volumes that are configured on the ACS library software hosts (Host A for Robot 1 and Host B for Robot 2). The software hosts reside on the robot (ACS 0 for each) that is designated in the Oracle StorageTek drive address.

In this example, assume the following about drive 1:

- Has an ACS drive address (ACS, LSM, panel, drive) of 0,0,1,1 in the NetBackup device configuration
- Is under control of robot number 10 (ACS(10))

If any other robot ACS(10) drives have a different ACS drive address (for example, 1,0,1,0), the configuration is invalid.

NetBackup supports configurations of multiple LSMs in a single ACS robot if a pass-through port exists.

Oracle StorageTek ACSLS firewall configuration

To configure an ACS robot in an Oracle StorageTek ACSLS firewall environment, use the following NetBackup `vm.conf` file configuration entries to designate TCP port connections:

- `ACS_CSI_HOSTPORT`

- `ACS_SSI_INET_PORT`
- `ACS_TCP_RPCSERVICE`

More information about `vm.conf` entries is available.

See the NetBackup Administrator's Guide, Volume I.

The Oracle StorageTek ACSLS server configuration options must match the entries in the `vm.conf` file. For example, in a typical ACSLS firewall configuration, you would change the following settings as shown:

- `Changes to alter use of TCP protocol...`
  Set to TRUE - Firewall-secure ACSLS runs across TCP.
- `Changes to alter use of UDP protocol...`
  Set to FALSE - Firewall-secure ACSLS runs across TCP.
- `Changes to alter use of the portmapper...`
  Set to NEVER - Ensures that the ACSLS server does not query the portmapper on the client platform.
- Enable CSI to be used behind a firewall...
  Set to TRUE - Allows specification of a single port for the ACSLS server.
- `Port number used by the CSI...`
  The port that the user chooses. The 30031 default value is used most often.
  This port number must match the port number that you specify in the NetBackup vm.conf file.

For complete information about setting up a firewall-secure ACSLS server, refer to your vendor documentation.

Chapter 7

Device configuration examples

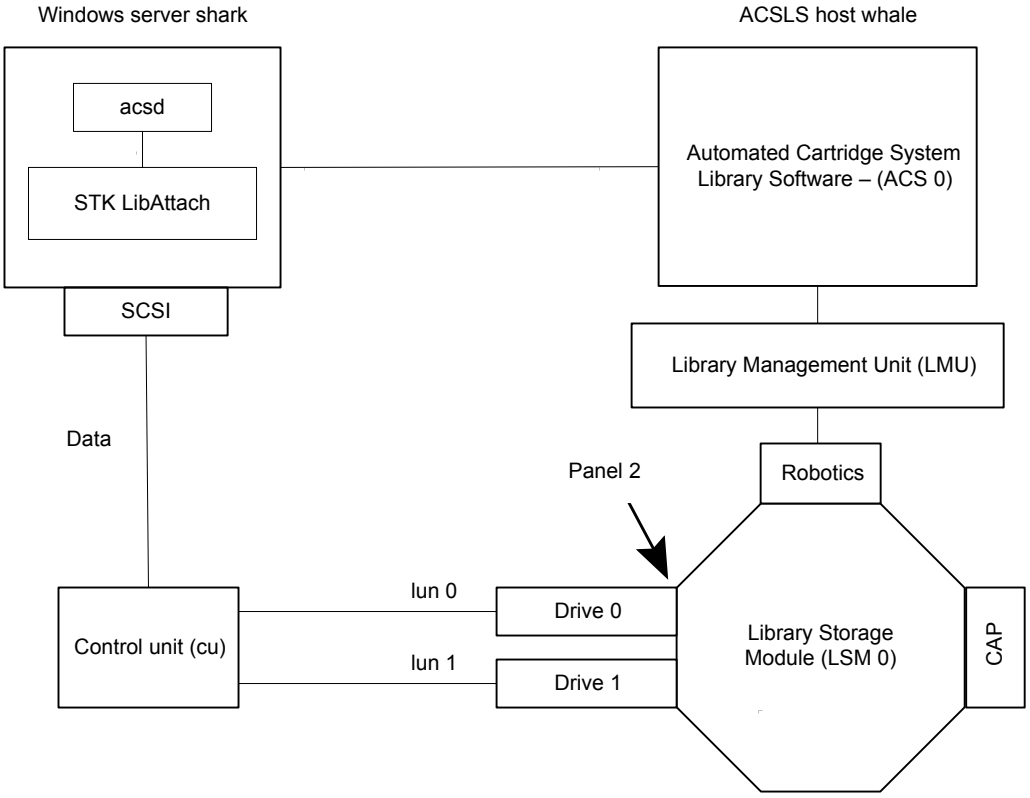
This chapter includes the following topics:

- An ACS robot on a Windows server example
- An ACS robot on a UNIX server example

An ACS robot on a Windows server example

The following figure shows a Windows server and ACS robot configuration.

Figure 7-1 Windows server and ACS robot configuration example



This configuration uses an Automated Cartridge System (ACS) robot for storage. Server shark can be a Windows NetBackup primary server or media server.

The following are items to note when you review this example:

- The Oracle StorageTek ACSLS host (in the Add Robot dialog) is host whale, where the ACS library software resides. In this example, Automated Cartridge System Library Software (ACSLS) is installed as the ACS library software.
  On some server platforms, you can run NetBackup media server software and ACS library software on the same server. Therefore, you need only one server.
- The ACS, LSM, PANEL, and DRIVE numbers are part of the ACS library software configuration and must be obtained from the administrator of that host.
- Robot number and ACS number are different terms. Robot number is the robot identifier used in NetBackup. ACS number is the robot identifier in ACS library software. These numbers can be different, although they both default to zero.

- If you connnect the drives through an independent control unit, you must use the correct Logical Unit Numbers (LUNs) so that the correct tape name is used.
- The Add Robot dialog entries include an ACSLS Host entry so that the ACS library software host communicates by using STK LibAttach software. This software must be installed on each Windows server that has the ACS drives attached to it.

The following table shows the robot attributes for the remote host shark.

Table 7-1 Add Robot dialog entries (remote host)

Dialog box field	Value
Device Host	shark
Robot Type	ACS (Automated Cartridge System)
Robot Number	0
Robot control is handled by a remote host	Set (cannot be changed for this robot type)
ACSLS Host	whale

The following table shows the drive 0 attributes.

Table 7-2 Add Drive dialog entries (drive 0)

Dialog box field	Value
Device Host	shark
Drive Type	1/2" Cartridge (hcart)
Drive Name	shark_drive_0
Path Information	[5,0,1,0]
Drive is in a Robotic Library	Yes
Robotic Library	ACS(0) - whale
ACS	ACS: 0 LSM: 0 PANEL: 2 DRIVE: 0

The following table shows the drive attributes for drive 1.

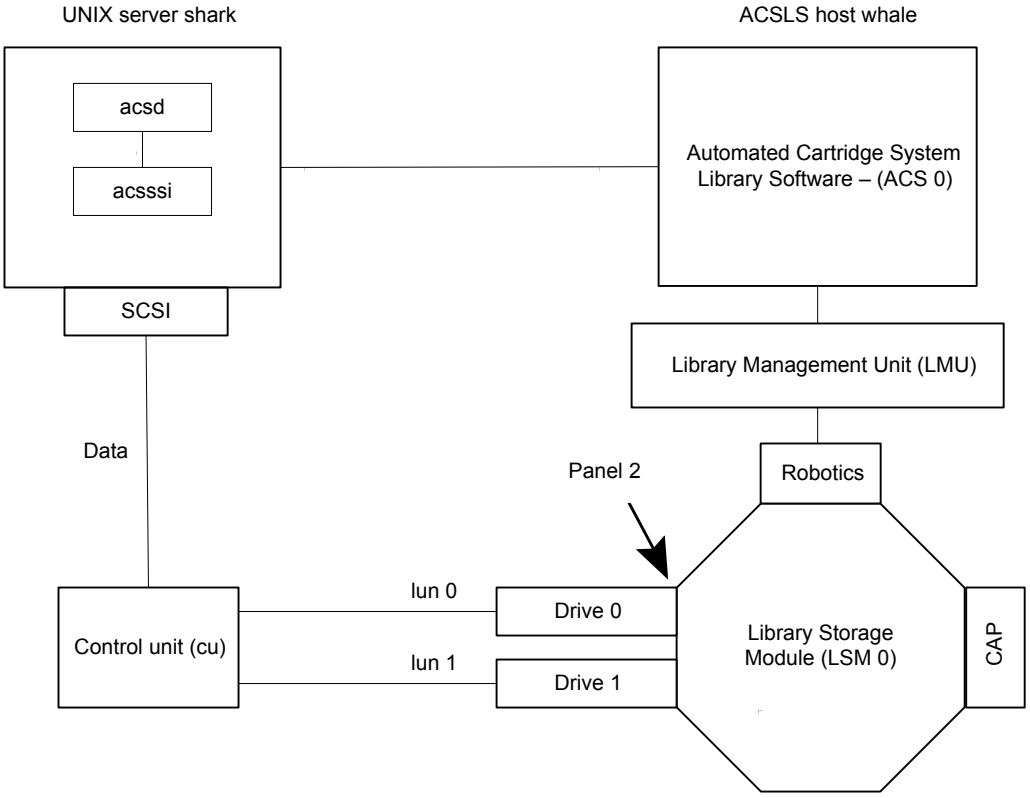
Table 7-3 Add Drive dialog entries (drive 1)

Dialog box field	Value
Device Host	shark
Drive Type	1/2" Cartridge (hcart)
Drive Name	shark_drive_1
Path Information	[4,0,1,1]
Drive is in a Robotic Library	Yes
Robotic Library	ACS(0) - whale
ACS	ACS: 0 LSM: 0 PANEL: 2 DRIVE: 1

An ACS robot on a UNIX server example

The following figure shows a UNIX server and ACS robot configuration.

Figure 7-2 UNIX server and ACS robot configuration example



This configuration uses an Automated Cartridge System (ACS) robot for storage. Host shark can be a UNIX NetBackup primary server or media server.

The following are some items to note when you review this example:

- The ACSLS Host (in the Add Robot dialog) is server whale, where the ACS library software resides. In this example, Automated Cartridge System Library Software (ACSLS) is installed as the ACS library software.
  On some server platforms, you can run NetBackup media server software and ACS library software on the same server. Therefore, you need only one server.
- The ACS, PANEL, LSM, and DRIVE numbers are part of the ACS library software configuration and must be obtained from that system.
- Robot number and ACS number are different terms. Robot number is the robot identifier used in NetBackup. ACS number is the robot identifier in ACS library software. These numbers can be different, although they both default to zero.

- If you connnect the drives through an independent control unit, you must use the correct Logical Unit Numbers (LUNs) so that the correct tape name is used.
- The Add Robot dialog entries include an ACSLS Host entry. That entry configures NetBackup to use the ACS Storage Server Interface (acsssi) to communicate with the ACS library software host.

The following table shows the robot attributes.

Table 7-4 Add Robot dialog entries (remote host)

Dialog box field	Value
Device Host	shark
Robot Type	ACS (Automated Cartridge System)
Robot Number	0
Robot control is handled by a remote host	Set (cannot be changed for this robot type)
ACSLS Host	whale

The following table shows the drive 0 attributes.

Table 7-5 Add Drive dialog entries (drive 0)

Dialog box field	Value
Device Host	shark
Drive Name	shark_drive_0
Drive Type	1/2" Cartridge (hcart)
No Rewind Device	/dev/rmt1.1
Drive is in a Robotic Library	Yes
Robotic Library	ACS(0) - whale
ACS	ACS Number: 0 LSM Number: 2 PANEL Number: 0 DRIVE Number: 0

The following table shows the drive 1 attributes.

Table 7-6 Add Drive dialog entries (drive 1)

Dialog box field	Value
Device Host	shark
Drive Name	shark_drive_1
Drive Type	1/2" Cartridge (hcart)
No Rewind Device	/dev/rmt1.1
Drive is in a Robotic Library	Yes
Robotic Library	ACS(0) - whale
ACS	ACS Number: 0 LSM Number: 2 PANEL Number: 0 DRIVE Number: 1