Script command structure

SANtricity commands

NetApp
October 14, 2025

Table of Contents

Script command structure

Learn about the structure of a SANtricity CLI script commands

To ensure successful execution, script commands must be written in the proper structure.

All script commands have the following structure:

```
*command operand-data* (*statement-data*)
```

- `command` identifies the action to be performed.
- `operand-data` represents the objects associated with a storage array that you want to configure or manage.
- `statement-data` provides the information needed to perform the command.

The syntax for *`operand-data`* has the following structure:

```
(*object-type* | all *object-types* | [*qualifier*] (*object-type*
[*identifier*] (*object-type* [*identifier*] | *object-types*
[*identifier-list*])))
```

An object can be identified in four ways:

- Object type — Use when the command is not referencing a specific object.
- **`all`** parameter prefix — Use when the command is referencing all of the objects of the specified type in the storage array (for example, **`allVolumes`**).
- Square brackets — Use when performing a command on a specific object to identify the object (for example, **`volume [engineering]`**).
- A list of identifiers — Use to specify a subset of objects. Enclose the object identifiers in square brackets (for example, **`volumes [sales engineering marketing]`**).

A qualifier is required if you want to include additional information to describe the objects.

The object type and the identifiers that are associated with each object type are listed in this table.

Object Type	Identifier
`controller`	a or b
`drive`	For enclosures with drawers, use tray ID, drawer ID, and slot ID. Alternatively, just tray ID and slot ID.

Object Type	Identifier
replacementDrive	For enclosures with drawers, use tray ID, drawer ID, and slot ID. Alternatively, just tray ID and slot ID.
driveChannel	Drive channel identifier
host	User label
hostChannel	Host channel identifier
hostGroup	User label
hostPort	User label
iscsiInitiator	User label or iSCSI Qualified Name (IQN)
iscsiTarget	User label or IQN
storageArray	Not applicable
tray	Tray ID
volume	Volume user label or volume World Wide Identifier (WWID) (`set` command only)
volumeCopy	Target volume user label and, optionally, the source volume user label
volumeGroup	User label Valid characters are alphanumeric, a hyphen, and an underscore.

Statement data is in the form of:

- Parameter = value (such as **raidLevel=5**)
- Parameter-name (such as **batteryInstallDate**)
- Operation-name (such as **redundancyCheck**)

A user-defined entry (such as user label) is called a variable. In the syntax, it is shown in italic (such as `trayID` or `volumeGroupName`).

Learn about SANtricity CLI script commands

Because you can use the script commands to define and manage the different aspects of a storage array (such as host topology, drive configuration, controller configuration, volume definitions, and volume group definitions), the actual number of commands is extensive.

The commands fall into general categories that are reused when you apply the commands to configure or maintain a storage array. The following table lists the general form of the script commands and a definition of each command.

Syntax	Description
`accept object {statement-data}`	Performs the pending operation.
`activate object {statement-data}`	Sets up the environment so that an operation can take place or performs the operation if the environment is already set up correctly.
`autoConfigure storageArray {statement-data}`	Automatically creates a configuration that is based on the parameters that are specified in the command.
`check object {statement-data}`	Starts an operation to report on errors in the object, which is a synchronous operation.
`clear object {statement-data}`	Discards the contents of some attributes of an object. This operation is destructive and cannot be reversed.
`create object {statement-data}`	Creates an object of the specified type.
`deactivate object {statement-data}`	Removes the environment for an operation.

Syntax	Description
`delete object`	Deletes a previously created object.
`diagnose object {statement-data}`	Runs a test and shows the results.
`disable object {statement-data}`	Prevents a feature from operating.
`download object {statement-data}`	Transfers data to the storage array or to the hardware that is associated with the storage array.
`enable object {statement-data}`	Sets a feature to operate.
`load object {statement-data}`	Transfers data to the storage array or to the hardware that is associated with the storage array. This command is functionally similar to the `download` command.
`recopy object {statement-data}`	Restarts a volume copy operation by using an existing volume copy pair. You can change the parameters before the operation is restarted.
`recover object {statement-data}`	Re-creates an object from saved configuration data and the statement parameters. (This command is similar to the `create` command.)
`remove object {statement-data}`	Removes a relationship between objects.
`repair object {statement-data}`	Repairs errors found by the `check` command.

Syntax	Description
`replace object {statement-data}`	The specified object replaces an existing object in the storage array.
`reset object {statement-data}`	Returns the hardware or an object to an initial state.
`resume object`	Starts a suspended operation. The operation starts where it left off when it was suspended.
`revive object`	Forces the object from the Failed state to the Optimal state. Use this command only as part of an error recovery procedure.
`save object {statement-data}`	Writes information about the object to a file.
`set object {statement-data}`	Changes object attributes. All changes are completed when the command returns.
`show object {statement-data}`	Shows information about the object.
`start object {statement-data}`	Starts an asynchronous operation. You can stop some operations after they have started. You can query the progress of some operations.
`stop object {statement-data}`	Stops an asynchronous operation.
`suspend object {statement-data}`	Stops an operation. You can then restart the suspended operation, and it continues from the point where it was suspended.

Syntax	Description
`validate object {statement-data}`	Validates a security key.

Learn about recurring SANtricity CLI syntax elements

Recurring syntax elements are a general category of parameters and options that you can use in the script commands. The following table lists the conventions used in the recurring syntax elements.

Convention	Definition
a | b	Alternative ("a" or "b")
`italicized-words`	Needs user input to fulfill a parameter (a response to a variable)
`[ ... ]` (square brackets)	Zero or one occurrence (square brackets are also used as a delimiter for some command parameters)
`{ ... }` (curly braces)	Zero or more occurrences
(**a | b | c**)	Choose only one of the alternatives

The following table lists the recurring syntax parameters and the values that you can use with the recurring syntax parameters.

Recurring Syntax	Syntax Value
autoconfigure-vols-attr- value-list	*autoconfigure-vols-attr-value-pair* {*autoconfigure-vols-attr-value-pair*}

Recurring Syntax	Syntax Value
autoconfigure-vols-attr- value-pair	driveType=drive-type | driveMediaType=drive-media-type | raidLevel=raid-level | volumeGroupWidth=integer-literal | volumeGroupCount=integer-literal | volumesPerGroupCount=integer-literal6| hotSpareCount=integer-literal | segmentSize=segment-size-spec | cacheReadPrefetch=(TRUE | FALSE) securityType=(none | capable | enabled)7| dataAssurance=(none | enabled)5
boolean	(TRUE | FALSE)
cache-flush-modifier- setting	immediate, 0, .25, .5, .75, 1, 1.5, 2, 5, 10, 20, 60, 120, 300, 1200, 3600, infinite
capacity-spec	*integer-literal* [KB | MB | GB | TB | Bytes]
count-based-repository- spec	repositoryRAIDLevel =repository-raid-level repositoryDriveCount=integer-literal [repositoryVolumeGroupUserLabel =user-label] [driveType=drive-type4] [trayLossProtect=(TRUE | FALSE)1] | [drawerLossProtect=(TRUE | FALSE)2] | [dataAssurance=(none | enabled)5] |

Recurring Syntax	Syntax Value
create-raid-vol-attr- value-list	*create-raid-volume-attribute-value-pair* {*create-raid-volume-attribute-value-pair*}
create-raid-volume- attribute-value-pair	`capacity=capacity-spec | owner=(a | b) | cacheReadPrefetch=(TRUE | FALSE) | segmentSize=integer-literal | usageHint=usage-hint-spec`
create-volume-copy-attr- value-list	*create-volume-copy-attr-value-pair* {*create-volume-copy-attr-value-pair*}
create-volume-copy-attr- value-pair	`copyPriority=(highest | high | medium | low | lowest) | targetReadOnlyEnabled=(TRUE | FALSE) | copyType=(offline | online) | repositoryPercentOfBase=(20 | 40 | 60 | 120 | default) | repositoryGroupPreference=(sameAsSource | otherThanSource | default)`
drive-media-type	`(HDD | SSD | unknown | allMedia)` *HDD* means hard disk drive. *SSD* means solid state disk.
drive-spec	*trayID*,*slotID* or trayID,drawerID,slotID A drive is defined as two or three integer literal values separated by a comma. Low-density trays require two values. High-density trays, those trays that have drawers, require three values.
drive-spec-list	*drive-spec drive-spec*
drive-type	`(fibre | SATA | SAS)` Only SAS drives are supported for firmware versions 7.86 and later.
error-action	`(stop | continue)`

Recurring Syntax	Syntax Value
ethernet-port-options	`enableIPv4=(TRUE | FALSE) | enableIPv6=(TRUE | FALSE) | IPv6LocalAddress=ipv6-address | IPv6RoutableAddress=ipv6-address | IPv6RouterAddress=ipv6-address | IPv4Address=ip-address | IPv4ConfigurationMethod= (static | dhcp) | IPv4GatewayIP=ip-address | IPv4SubnetMask=ip-address | duplexMode=(TRUE | FALSE) | portSpeed=(autoNegotiate | 10 | 100 | 1000)`
feature-identifier	All features in SANtricity 11.40 are enabled by default.
filename	*string-literal*
gid	*string-literal*
hex-literal	A literal in the range of 0x00 - 0xFF.
host-card-identifier	`(1 | 2 | 3 | 4)`
host-type	`string-literal|integer-literal`

Recurring Syntax	Syntax Value
instance-based- repository-spec	`(repositoryRAIDLevel =repository-raid-level repositoryDrives= (drive-spec-list) [repositoryVolumeGroupUserLabel =user-label] [trayLossProtect=(TRUE | FALSE)1]) | [drawerLossProtect=(TRUE | FALSE)2]) | (repositoryVolumeGroup=user-label [freeCapacityArea=integer-literal3])` Specify the **repositoryRAIDLevel** parameter with the **repositoryDrives** parameter. Do not specify the RAID level or the drives with the volume group. Do not set a value for the **trayLossProtect** parameter when you specify a volume group.
ip-address	**(0-255).(0-255).(0-255).(0-255)**
ipv6-address	**(0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF): (0-FFFF):(0-FFFF):(0-FFFF):(0-FFFF)** You must enter all 32 hexadecimal characters.
iscsi-host-port	`(1 | 2 | 3 | 4)` The host port number might be 2, 3, or 4 depending on the type of controller you are using.

Recurring Syntax	Syntax Value
iscsi-host-port-options	`IPv4Address=ip-address | IPv6LocalAddress=ipv6-address | IPv6RoutableAddress=ipv6-address | IPv6RouterAddress=ipv6-address | enableIPv4=(TRUE | FALSE) | enableIPv6=(TRUE | FALSE) | enableIPv4Priority=(TRUE | FALSE) | enableIPv6Priority=(TRUE | FALSE) | IPv4ConfigurationMethod=(static | dhcp) | IPv6ConfigurationMethod= (static | auto) | IPv4GatewayIP=ip-address | IPv6HopLimit=integer | IPv6NdDetectDuplicateAddress=integer | IPv6NdReachableTime=time-interval | IPv6NdRetransmitTime=time-interval | IPv6NdTimeOut=time-interval | IPv4Priority=integer | IPv6Priority=integer | IPv4SubnetMask=ip-address | IPv4VlanId=integer | IPv6VlanId=integer | maxFramePayload=integer | tcpListeningPort=tcp-port-id | portSpeed=(autoNegotiate | 1 | 10)`
iscsiSession	`[session-identifier]`
nvsram-offset	*hex-literal*
nvsramBitSetting	*nvsram-mask, nvsram-value* = *0xhexadecimal, 0xhexadecimal* | *integer-literal* The *0xhexadecimal* value is typically a value from 0x00 to 0xFF.

Recurring Syntax	Syntax Value
nvsramByteSetting	*nvsram-value* = *0xhexadecimal* | *integer-literal* The `0xhexadecimal` value is typically a value from 0x00 to 0xFF.
portID	`(0-127)`
raid-level	`(0 | 1 | 3 | 5 | 6)`
recover-raid-volume-attr- value-list	*recover-raid-volume-attr-value-pair* {*recover-raid-volume-attr-value-pair*}

recover-raid-volume-attr- value-pair	`owner=(a|b)`
	`|cacheReadPrefetch=(TRUE | FALSE)`
	`| dataAssurance=(none | enabled)`

repository-raid-level	`(1 | 3 | 5 | 6)`
repository-spec	`instance-based-repository-spec` | `count-based-repository-spec`
segment-size-spec	*integer-literal* - all capacities are in base-2.
serial-number	`string-literal`

Recurring Syntax	Syntax Value

slotID	For high-capacity drive trays, specify the tray ID value, the drawer ID value, and the slot ID value for the drive. For low-capacity drive trays, specify the tray ID value and the slot ID value for the drive. Tray ID values are `0` to `99`. Drawer ID values are `1` to `5`.
	
	All slot ID maximums are 24. Slot ID values either begin at 0 or 1, depending on the tray model.
	
	Enclose the tray ID value, the drawer ID value, and the slot ID value in square brackets ([ ]).
	
	`(drive=\(trayID,[drawerID,]slotID\)|`
	`drives=\(trayID1,[drawerID1,]slotID1 ...`
	`trayIDn,[drawerIDn,]slotIDn\) )`

test-devices	`controller=(a|b)` `esms=(esm-spec-list)drives=(drive-spec-list)`
test-devices-list	*test-devices*{*test-devices*}
time-zone-spec	`(GMT+HH:MM | GMT-HH:MM)` `[dayLightSaving=HH:MM]`
trayID-list	*trayID*{*trayID*}
usage-hint-spec	`usageHint=(multiMedia | database | fileSystem)` The usage hint, or expected I/O characteristics, of the volume are used by the controller to indicate an appropriate default volume segment size and dynamic cache read prefetch. For file system and database, a 128 KB segment size is used. For multimedia, a 256 KB segment size is used. All three usage hints enable dynamic cache read prefetch.

Recurring Syntax	Syntax Value
user-label	*string-literal* Valid characters are alphanumeric, the dash, and the underscore.
user-label-list	*user-label* {*user-label*}
volumeGroup-number	*integer-literal*
wwID	*string-literal*

1For tray loss protection to work, your configuration must adhere to the following guidelines:

Level	Criteria for Tray Loss Protection	Minimum number of trays required
Disk Pool	The disk pool contains no more than two drives in a single tray.	6
RAID 6	The volume group contains no more than two drives in a single tray.	3
RAID 3 or RAID 5	Each drive in the volume group is located in a separate tray.	3
RAID 1	Each drive in a RAID 1 pair must be located in a separate tray.	2
RAID 0	Cannot achieve Tray Loss Protection.	Not applicable

2For drawer loss protection to work, your configuration must adhere to the following guidelines:

Level	Criteria for drawer loss protection	Minimum number of drawers required
Disk Pool	The pool includes drives from all five drawers and there are an equal number of drives in each drawer. A 60-drive tray can achieve Drawer Loss Protection when the disk pool contains 15, 20, 25, 30, 35, 40, 45, 50, 55, or 60 drives.	5

Level	Criteria for drawer loss protection	Minimum number of drawers required
RAID 6	The volume group contains no more than two drives in a single drawer.	3
RAID 3 or RAID 5	Each drive in the volume group is located in a separate drawer.	3
RAID 1	Each drive in a mirrored pair must be located in a separate drawer.	2
RAID 0	Cannot achieve Drawer Loss Protection.	Not applicable

If you have a storage array configuration in which a volume group spans several trays, you must make sure that the setting for drawer loss protection works with the setting for tray loss protection. You can have drawer loss protection without tray loss protection. You cannot have tray loss protection without drawer loss protection. If the **`trayLossProtect`** parameter and the **`drawerLossProtect`** parameter are not set to the same value, the storage array returns an error message and a storage array configuration will not be created.

3 To determine if a free capacity area exists, run the `show volumeGroup` command.

4 The default drive (drive type) is `SAS`.

The **`driveType`** parameter is not required if only one type of drive is in the storage array. If you use the **`driveType`** parameter, you also must use the **`hotSpareCount`** parameter and the **`volumeGroupWidth`** parameter.

5 The **`dataAssurance`** parameter relates to the Data Assurance (DA) feature.

The Data Assurance (DA) feature increases data integrity across the entire storage system. DA enables the storage array to check for errors that might occur when data is moved between the hosts and the drives. When this feature is enabled, the storage array appends error-checking codes (also known as cyclic redundancy checks or CRCs) to each block of data in the volume. After a data block is moved, the storage array uses these CRC codes to determine if any errors occurred during transmission. Potentially corrupted data is neither written to disk nor returned to the host.

If you want to use the DA feature, start with a pool or volume group that includes only drives that support DA. Then, create DA-capable volumes. Finally, map these DA-capable volumes to the host using an I/O interface that is capable of DA. I/O interfaces that are capable of DA include Fibre Channel, SAS, and iSER over InfiniBand (iSCSI Extensions for RDMA/IB). DA is not supported by iSCSI over Ethernet, or by the SRP over InfiniBand.

When all the required hardware and the I/O interface is DA-capable, you can set the **`dataAssurance`** parameter to `enabled` and then use DA with certain operations. For example, you can create a volume group that includes DA-capable drives, and then create a volume within that volume group that is DA-enabled. Other operations that use a DA-enabled volume have options to support the DA feature.

6 The **`volumesPerGroupCount`** parameter is the number of equal-capacity volumes per volume group.

7 The **`securityType`** parameter enables you to specify the security setting for a volume group that you are creating. All of the volumes are also set to the security setting that you choose. Available options for setting the security setting include:

- `none` — The volume group is not secure.
- `capable` — The volume group is security capable, but security has not been enabled.
- `enabled` — The volume group is security enabled.

A storage array security key must already be created for the storage array if you want to set **`securityType=enabled`**. (To create a storage array security key, use the `create storageArray securityKey` command.)

Copyright information

Trademark information