



# **PRIMECLUSTER™**

Global Disk Services

Configuration and Administration Guide 4.1

(Linux)

## **Appendix**

January 2005



# Appendix A General Notes

## A.1 Rules

### A.1.1 Object Name

Users can name the following objects:

- Classes
- Disks (excluding shadow disk)
- Groups
- Volumes

The object name can contain a maximum of 32 alphanumeric characters, including the minus sign (-) and the underscore character (\_). The object name cannot start with the minus sign (-) or the underscore character (\_). Be sure to assign an alphanumeric character to the first character in the object name.

Class names must be unique within the entire cluster system and an attempt to create two or more classes with the same name will result in an error.

The other object names are unique within the class and an attempt to create more than one object with the same name in a class will result in an error.

GDS assigns slice names by combining the names of disks, groups, and volumes to which the slices belong. The slice naming conventions are as follows.

- When the slice is a mirror slice and exists in a disk that is connected directly to the highest level mirror group:  
***disk\_name.volume\_name***
- When the slice is a mirror slice and exists in a lower level group that is connected directly to the highest level mirror group:  
***lower\_level\_group\_name.volume\_name***
- When the slice belongs to a stripe volume:  
***the\_highest\_level\_stripe\_group\_name.volume\_name***
- When the slice belongs to a volume created within the highest level concatenation group:  
***the\_highest\_level\_concatenation\_group\_name.volume\_name***
- When the slice belongs to a switch volume:  
***active\_disk\_name.volume\_name***
- When the slice is a single slice:  
***single\_disk\_name.volume\_name***

Thus all object names can be identified uniquely within the entire cluster system by the combinations of the class names and the object names.



#### Note Shadow Disk Name

Name shadow disks according to the following rules:

- When the shadow disk is already registered with a class in another domain and managed as an SDX disk, assign the SDX disk name in said domain.
- When the shadow disk contains data copied from an SDX disk with the disk unit's copy function, assign the copy source SDX disk name.

## **A.1.2 Number of Classes**

---

There is no limit to the number of shared classes.

Separate classes according to the following guidelines:

- Register shared disks whose scope (groups of sharing nodes) is different with separate shared classes.
- For cluster applications to use shared disks, create one or more shared classes with respect to each cluster application.
- Divide the class if the number of created disks or volumes in a class exceeds the limit. For the numbers of disks and volumes, see "A.1.3 Number of Disks" and "A.1.5 Number of Volumes."
- In a large-scale system to which numerous disk units are connected, separating classes based on physical configurations and data contents of disks may bring higher manageability.
- When a disk unit is expanded, unless the rules above apply, register expanded disks to existing classes. Creating new classes is not required.

Do not separate classes more than necessary. Keeping the number of classes to a minimum will offer the following advantages.

- If a class includes more disks, the probability that GDS configuration information stored on disks is lost due to disk failure will be lower. Therefore, not separating classes more than necessary to increase the number of disks within one class will raise system reliability.
- If there are less shared classes, it takes shorter time to switch nodes.
- If there are fewer classes, less memory resources will be required.

## **A.1.3 Number of Disks**

---

The number of disks that can be registered with one class is limited to a maximum of 1024.

There are the following limits to the number of disks that can be connected to one group:

- To a mirror group, a maximum of 8 disks and lower level groups can be connected collectively. In other words, a maximum of eight-way multiplex mirroring is supported. However, be aware that the spare disk that will be automatically connected when a disk failure occurs is also included in the count.
- To a stripe group, a maximum of 64 disks and lower level groups can be connected collectively. In other words, a maximum of 64-column striping is supported.
- To a concatenation group, a maximum of 64 disks can be connected. In other words, a maximum of 64 disks can be concatenated.
- To a switch group, a maximum of 2 disks can be connected.

## **A.1.4 Number of Groups**

---

The number of groups that can be created in one class is limited to a maximum of 1024.

## **A.1.5 Number of Volumes**

---

There are the following limits to the number of volumes that can be created within a group or a single disk:

- The number of volumes with the physical slice attributes set to "on" is limited to a maximum of 4.
- The number of volumes with the physical slice attributes set to "on" or "off" is limited to a maximum of 224 collectively.
- Stripe groups and concatenation groups cannot contain volumes with the physical slice attributes set to "on".
- Shadow classes cannot contain volumes with the physical slice attributes set to "on" regardless whether the physical slices are registered with the partition table.

In addition, the following restrictions are applied.

- The number of volumes that can be created within a class is limited to a maximum of 224.
- The number of volumes that can be created within a node is limited to a maximum of 224.
- Nested groups can contain a maximum of 224 volumes and lower level groups collectively.

## A.1.6 Creating Group Hierarchy

---

The following eight kinds of group hierarchical structures, including a nonhierarchical structure, are available.

higher level group <-----> lower level group  
mirror group (\*1)  
mirror group (\*1) - stripe group (\*3)  
mirror group (\*1) - stripe group (\*3) - concatenation group (\*8)  
mirror group (\*1) - concatenation group (\*6)  
stripe group (\*2)  
stripe group (\*2) - concatenation group (\*7)  
concatenation group (\*4)  
switch group (\*5)

Possible operations on groups at each hierarchical level that change the structure are as follows.

### (\*1) The highest level mirror group

- Disks, lower level stripe groups, and lower level concatenation groups can be connected or disconnected. However, disconnection is impossible if it can change the volume configuration or status.
- If no volume exists, the group itself can be removed.
- Volumes can be created or removed.

### (\*2) The highest level stripe group

- If no volume exists, disks and lower level concatenation groups can be connected or disconnected.
- If no volume exists, this type group can be connected to a mirror group.
- If no volume exists, the group itself can be removed.
- If more than one disks or lower level concatenation groups are connected to, volumes can be created or removed.

### (\*3) The lower level stripe group

- This type group can be disconnected from the highest level mirror group. However, disconnection is impossible if it can change the volume configuration or status.

### (\*4) The highest level concatenation group

- Disks can be connected.
- If no volume area exists on the disk that was connected last, that disk can be disconnected.
- If no volume exists, this type group can be connected to a mirror group or a stripe group.
- If no volume exists, the group itself can be removed.
- Volumes can be created or removed.

### (\*5) The highest level switch group

- Disks can be connected.
- If no volume or inactive disk exists, the active disk can be disconnected.
- The inactive disk can be disconnected.
- If no volume exists, the group itself can be removed.
- Volumes can be created or removed.

### (\*6) Lower level concatenation group connected to the highest level mirror group

- Disks can be connected.
- If more than one disk is connected and no volume area exists on the disk that was connected last, that disk can be disconnected.
- This type group can be disconnected from the highest level mirror group. However, disconnection is impossible if it can change the volume configuration or status.

### **(\*7) Lower level concatenation group connected to the highest level stripe group**

- If no volume exists within the highest level stripe group, disks can be connected.
- If more than one disk is connected and no volume exists within the highest level group, the disk that was connected last can be disconnected.
- If no volume exists within the highest level stripe group, this type group can be disconnected from the highest level stripe group.

### **(\*8) Lower level concatenation group connected to a lower level stripe group**

- None.

## **A.1.7 Exception to Proxy Configuration**

---

Although a pair of volumes or a pair of groups can be related as the master and the proxy, there are the following exceptions:

- Objects that belong to different classes cannot be related as the master and the proxy.
- Objects that belong to the same group cannot be related as the master and the proxy.
- Stripe type objects cannot be related as the master and the proxy. However, a mirror group to which a lower level stripe group is connected, or a mirror volume within that mirror group, can be related as the master and the proxy. For example, to stripe a master volume, connect a stripe group to a mirror group and create the master volume in the mirror group.
- Concatenation type objects cannot be related as the master and the proxy. However, a mirror group to which a lower level concatenation group is connected, or a mirror volume within that mirror group, can be related as the master and the proxy. For example, to concatenate disks to create a large-sized master group, connect a concatenation group to a mirror group and create the mirror group the master group.
- For a proxy that is already related to a master, another proxy cannot be related.
- A master to which a proxy is already related cannot be related as a proxy to another master.
- Shadow class objects cannot be related as the master and the proxy.

## **A.1.8 Number of Proxy Volumes**

---

Although more than one proxy volume can be related to one master volume, the number of proxy volumes that can be related is limited to meet the following conditions:

- The total number of slices composing one master volume and slices composing any proxy volumes that are related to the master volume cannot exceed 32.

For example, if the master volume and the proxy volumes all consist of single volumes, a maximum of 31 proxy volumes can be related to the master volume.

## **A.1.9 Proxy Volume Size**

---

The size of a proxy volume must be equal to that of the master volume to which it is related.

## **A.1.10 Proxy Group Size**

---

The size of a proxy group must be larger than the last block number of volumes within the master group to which the proxy is related.

## A.2 Important Points

### A.2.1 Managing System Disks

---

Do not use volumes in a shared class as any one of the following file systems or a swap area.  
/, /usr, /var, swap area

### A.2.2 Restraining Access to Physical Special File

---

After registration of disks with GDS is complete, accessing the disks using physical special files as below becomes impossible. GDS prevents unintentional access by the users to physical special files in order to protect mirroring statuses.

```
/dev/sd[a-z]*[1-4]*
```

For example, if you execute the dd command to write to the disk with physical special file, the following error message will be displayed.

```
# dd if=/dev/zero of=/dev/sda1  
dd: /dev/sda1: open: Device busy
```

Disk list output by the fdisk(8) command does not display the disk registered with GDS.

This access protection is cancelled when a disk is removed from a class. It will also be cancelled when an object is used for disk swap. The access protection function will turn back on if disk recovery operation is performed on the object.

Access to physical special files is prevented on the following nodes:

- For disks that are registered with disk classes  
All of the nodes in the relevant domains
- For disks that are registered with shadow classes  
Nodes where the relevant shadow classes reside

For the following disks, access to physical special files is not prevented. Be careful not to access these physical special files.

- Disks that are registered with classes in other domains
- Disks that are registered with shadow classes on other nodes

### A.2.3 Booting from a CD-ROM Device

---

If the system cannot be booted, for example, it may be required recovering the system booting from a CD-ROM device.

However, there are the following concerns when booting from a CD-ROM device.

- An operation mistake can easily be caused since circumstances when booting with this method and when booting from a boot disk may alter the correlation between the disk and the device special file (/dev/sd[a-z]\*[1-4]\*).
- Only partial mirrored disks can be written in since access to the device special file of a physical slice is not restrained. Therefore, the mirroring state could collapse.

For these reasons, avoid booting from a CD-ROM device unless directed in this manual.

When mounting a file system by booting from a CD-ROM device in any procedure not described in this manual from necessity, it should be mounted as read-only.

## A.2.4 Initializing Disk

---

When a physical disk is registered with a class (excluding a shadow class), all data on the physical disk will be lost since GDS automatically reformats the disk. The same applies when swapping and restoring the physical disk.

When using the GDS Management View, a dialog box reminding you that all data will be lost appears. However, please be aware that there will be no reconfirmation when actually performing the command.

Also, when registering a disk containing data with a class (excluding a shadow class), the disk will be initialized at the time of the registration with a class (excluding a shadow class). For this reason, be sure to back up data before the registration and restore the data after creating the volume.

Be aware that when a physical special file name is set for `/etc/fstab`, you will have to change it to a special file name for mirror volume.

```
/dev/sfdsk/class_name/rdisk/volume_name (character special file)
/dev/sfdsk/class_name/dsk/volume_name (block special file)
```

## A.2.5 Disk Size

---

Out of the physical disk area, the capacity available for creating volumes equals the physical disk size rounded down to the cylinder boundary, minus the private slice size. This size is called the available size of disk.

Private slice is an area that GDS reserves on each disk for storing data, such as configuration information and the JRM log.

When a physical disk is registered with a class, a private slice is reserved. Disks within the same class will all have the same private slice size. The size is determined by the size of the disk that is first registered with the class.

The size of a private slice can be estimated as below. The private slice size never exceeds the following estimated values.

- When the size of the disk first registered with the class is 10GB and below:  
32MB
- When the size of the disk first registered with the class is over 10GB:  
32MB + (0.1% of the disk capacity to the cylinder boundary, rounded-up)

The size of a log for JRM is determined by the size of the volume. As a result, the log area for JRM may not be reserved because the private slice size becomes insufficient when a mirror volume is created registering a disk larger in size than the first registered disk with the class. In such a case, the volume is incapable of using JRM. Therefore you are recommended to first register a disk of the maximum size with the class.

When physical disks are registered with shadow classes, the sizes of private slices are determined based on those saved in the partition table of the physical disks. For this reason, regardless of the order of registering physical disks with shadow classes, the sizes of private slices of the shadow classes become consistent with those of the corresponding disk classes. In other words, the order of registering physical disks with shadow classes is not a cause for concern.



Note

### Cylinder Size

The cylinder size of disks registered with shared classes is 32,768 blocks (= 16 MB). To calculate the size of disks, assume that the cylinder size is 16 MB.



## A.2.6 Volume Size

The size of a volume is automatically adjusted conforming to the following conditions.

- When creating a volume into a stripe group  
A volume is created in the size calculated by rounding-up the size specified when creating the volume to a common multiple of (stripe width) \* (stripe column number) and the cylinder size (32,768 blocks = 16 MB).  
**(Example)**  
If the stripe width is 32, the number of the stripe columns is 3 and 20 MB (= 40,960 blocks) is specified as the size of a volume when it is created, the size is rounded to a common multiple of 96 (= 32 \* 3) blocks and 32,768 blocks, and a volume with 98,304 blocks (= 48 MB) is created.
- When creating other volumes  
A volume is created in the size calculate by rounding-up the size specified when creating the volume to a multiple of the cylinder size (32,768 blocks = 16 MB).  
**(Example)**  
If 20 MB (= 40,960 blocks) is specified as the size of a volume when it is created, the size is rounded to a multiple of 32,768 blocks, and a volume with 65,536 blocks (= 32 MB) is created.

## A.2.7 Hot Spare

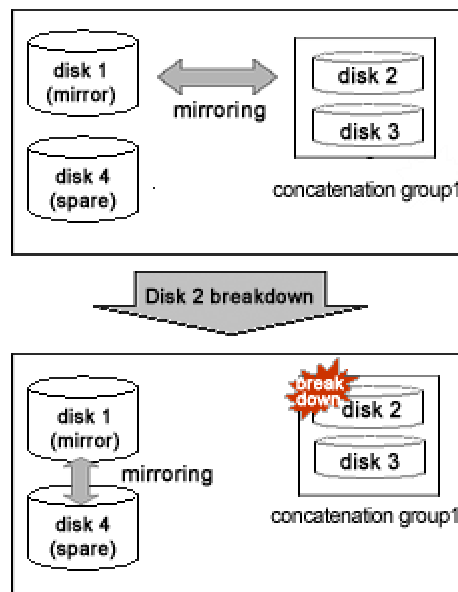
- **Hot Spare for Hierarchized Mirror Groups**  
If an I/O error occurs in a disk that is connected to a lower level group, a spare disk is automatically connected to the highest level mirror group, but not to the lower level group. For example, if a disk and a concatenation group are mirroring each other, and an I/O error occurs in one of the disks connected to the concatenation group, a spare disk is automatically connected in place of the concatenation group for mirroring.

A spare disk is selected independently of the disk case or controller number of a disk where an I/O error occurred.

In external hot spare mode (default), spare disks are selected randomly.

In internal hot spare mode, spare disks whose controller number is 0 and that do not belong to disk array units are selected.

Figure: Hot Spare for Hierarchized Mirror Groups



- **Number of Spare Disks**  
There is no limit to the number of spare disks that can be registered with one class. Although there is no general rule in deciding the number of spare disks, it is recommended to assign 10% of disks and lower level groups for spare disks. In other words, one spare disk for every 10 disks or lower level groups combined is a good rule of thumb.
- **Spare Disk Size**  
Spare disk automatic connection is restrained if there is not sufficient space on a spare disk to copy volumes in a mirror group. It is recommended to assign the largest size disk within the class for the spare disk.
- **Hot Spare for Proxy Volumes**  
Spare disks are not connected to groups that include proxy volumes. It is recommended to create proxy volumes in groups other than those that include volumes used by primary services, or on single disks.
- **Shadow Classes**  
Spare disks cannot be registered with shadow classes.
- **Disk Array Unit's Hot Spare Function**  
If disk array units with hot spare functions are mirrored, it is recommended to use their own hot spare functions.
- **Spare Disk Failure**  
If an I/O error occurs in a spare disk that was automatically connected to a mirror group, another spare disk will not automatically be connected in place of the failed spare disk.
- **Synchronization Copying Invoked by Hot Spare**  
The synchronization copying with hot spare is run at lower speed as compared to similar copying with other events (such as volume creation and disk creation) in order to suppress the load imposed on the system. By default, delay time by 50 milliseconds is set. To change this delay time, use the `sdxparam` command.



See

For details, see "D.11 `sdxparam` - Configuration parameter operations."

- **Spare Disk Manual Connection**  
In the within-case hot spare mode, if a disk case totally becomes inaccessible due to an I/O cable coming out or disk case going down, a spare disk that belongs another disk case is not automatically connected. For example, if disk case 1 shown in Figure: Hot Spare in Internal Mode] of "1.2.2 Hot Spare" is down, a spare disk (disk 4) is not automatically connected in place of disk 1.  
In such an event, follow the procedures below to manually recover the mirroring status by using a spare disk.

- 1) Change the spare disk to an undefined disk.



See

For disk type changing methods, see "Changing Disk Class Attributes" in "5.4.1 Class Configuration" when using the GDS Management View, or "D.7 `sdxattr` - Set objects attributes" when using the command.

- 2) Connect the disk in step 1) to the mirror group where the I/O error occurred.



See

For the disk connection methods, see "5.4.2 Group Configuration" when using the GDS Management View, or the description about the `-C` option in "D.2 `sdxdisk` - Disk operations" when using the command.

## A.2.8 Creating a Snapshot by Slice Detachment

---

A slice can be detached only from a mirror volume with a physical slice. . Therefore, if a disk is not connected directly to a mirror group, snapshot creation by slice detachment is impossible. In addition, a slice cannot be detached from a shadow volume.

## A.2.9 The Difference between a Mirror Slice and a Proxy Volume

---

Although data matches on mirrored slices or synchronized master volumes and proxy volumes, the purposes of use are different. Mirrored slices are equal one another, and their purpose is to maintain data redundancy in order to provide continuous access as long as any normal slice remains even if an error occurs in one of the slices.

However, even if the master volume and the proxy volume are synchronized, they are separate volumes and not equals. You may consider the master the primary volume, and the proxy the secondary volume. This means that you cannot continue accessing a master volume where all slices comprising the master volume are abnormal even if proxy volumes are normal. The purpose of proxy volumes is to create snapshots (saved copies from the master volume at certain moment) for a different service running concurrently with the primary service but not to improve the data redundancy of the master volume used in the primary service.

While the function of GDS Snapshot creating snapshots by detaching slices is a by-product of mirroring, the function itself is the primary purpose of GDS Snapshot by proxy volumes. Therefore, the application of proxy volumes provides more flexible disk configurations and service styles for snapshot management.



See "Figure: Difference between a Mirrored Slice and Synchronized Proxy Volume" in "1.5.1 Snapshot by Synchronization."

## A.2.10 Just Resynchronization Mechanism (JRM)

---

There are three types of Just Resynchronization Mechanism (JRM): for volumes, for slices and for proxies.

### JRM for Volumes

#### Overview

JRM for volumes speeds up the resynchronization process when booting the system after a system panic or the like. GDS records the changed portion in the private slice. The resynchronization copy performed at rebooting after an unexpected system failure copies the only portion that was written during the system down to realize high-speed resynchronization and minimize the load of copy processing.

#### Setting Methods

- From GDS Management View  
When creating the volume in the Volume Configuration dialog box, set the JRM mode (to "on" or "off"). The default value is "on". For details, see "5.2.1.3 Volume Configuration."
- With a Command  
When creating the volume with the `sdxvolume -M` command, specify the `-a jrm={on off}` option to set the JRM mode. By default, it is enabled. For details, see "D.4 sdxvolume - Volume operations."

#### Changing Methods

- From GDS Management View  
For the methods for changing the mode ("on" or "off") of JRM for volumes, see "5.4.3 Volume Configuration."

- With a Command  
To change the mode of JRM for volumes, specify the `-a jrm={on off}` option to the `sdxattr -V` command. For details, see "D.7 sdxattr - Set objects attributes."

### Checking Methods

- From GDS Management View  
For the mode ("on" or "off") of JRM for volumes, check the Volume Information Field in the Main Screen. For details, see "5.3.1.1 Confirming SDX Object Configuration."
- With a Command  
For the mode of JRM for volumes, use the `sdxinfo` command and check the JRM field of the displayed volume information. For details, see "D.6 sdxinfo - Display object configuration and status information."

## JRM for Slices

### Overview

JRM for slices speeds up the resynchronization process when reattaching a detached slice to the volume. GDS records the changes made on the volume and slice in the memory while the slice is being detached. The resynchronization copy performed when the detached slice is reattached copies the updated portions only to realize high-speed resynchronization.

JRM for slices becomes effective when a slice is detached while the `jrm` attribute of the slices is on. However, if a system is stopped or if the slice is taken over by the `sdxslice -T` command while the slice is detached, just resynchronization is not conducted when the temporarily detached slice is attached again. Resynchronization is performed by copying the entire data, not only the updated portions.

Therefore, if you plan to shut down the system, or have a slice taken over, attaching the slice to the volume in advance is highly recommended.

### Setting Methods

- From GDS Management View  
When detaching the slice with the [Operation]:[Detach Slice] menu in the Main Screen, the JRM mode is set to "on". Setting the mode to "off" is impossible.
- With a Command  
When detaching the slice with the `sdxslice -M` command, specify the `-a jrm={on off}` option to set the JRM mode. By default, it is enabled. For details, see "D.5 sdxslice - Slice operations."

### Changing Methods

- From GDS Management View  
Changing the mode of JRM for slices is impossible.
- With a Command  
To disable the mode of JRM for slices, specify the `-a jrm=off` option to the `sdxattr -S` command. Changing the mode to "on" is impossible. For details, see "D.7 sdxattr - Set objects attributes."

### Checking Methods

- From GDS Management View  
Checking the mode of JRM for slices is impossible.
- With a Command  
For the mode of JRM for slices, use the `sdxinfo` command with the `-e` long option and check the JRM field of the displayed slice information. For details, see "D.6 sdxinfo - Display object configuration and status information."

## JRM for Proxies

### Overview

JRM for proxies speeds up the just resynchronization process when joining a parted proxy again to the master and when the master data is restored from the proxy. GDS records the changes made on the master and the proxy on the memory while the proxy is parted. The just resynchronization conducted

when re-joining or restoring copies only the updated portions to realize high-speed synchronization.

JRM for proxies is enabled when the pjrm attribute of a proxy volume is set to "on" and the proxy volume is parted. However, if any node that is included in the scope of the class is stopped while the proxy is parted, just resynchronization is not put in operation. In other words, the entire data, not only the updated portions, is copied.

Therefore, if you plan to shut down the system, joining the proxy to the master in advance is highly recommended.

Reference to these matters is not necessary when you are using the copy function of a disk unit.

### Setting Methods

- From GDS Management View  
When parting the proxy with the [Operation]:[Proxy Operation]:[Part] menu in the Main Screen, set the JRM mode ("on" or "off") in the Part Proxy dialog box. The default value is "on". For details, see "Part" in "5.3.2.2 Backup(by Synchronization)."  
When relating the proxy with the [Operation]:[Proxy Operation]:[Relate] menu, the mode of JRM for proxies is disabled for the related proxy volume.
- With a Command  
When parting the proxy with the `sdxproxy Part` command, specify the `-a pjrm={on off}` option to set the mode of JRM for proxies. By default, it is enabled. For details, see " D.12 `sdxproxy` - Proxy object operations."

### Changing Methods

- From GDS Management View  
Changing the mode ("on" and "off") of JRM for proxies is impossible.
- With a Command  
To disable the mode of JRM for proxies, specify the `-a pjrm=off` option to the `sdxattr -V` command. Changing the mode to "on" is impossible. For details, see " D.7 `sdxattr` - Set objects attributes."

### Checking Methods

- From GDS Management View  
For the mode ("on" or "off") of JRM for proxies, check the Proxy Volume Information Field in the Main Screen. For details, see "5.3.1.2 Viewing Proxy Object Configurations."
- With a Command  
For the mode of JRM for proxies, use the `sdxinfo` command with the `-e` long option and check the PJRM field of the displayed volume information. For details, see " D.6 `sdxinfo` - Display object configuration and status information."

## A.2.11 Online Volume Expansion

---

- **Volume Configuration Limitations**  
Online volume expansion is available for volumes in the following configurations.
  - Single volume
  - Mirror volume
    - Any mirroring multiplicity is supported.
    - Hierarchized groups are supported.
- **Online Mirror Volume Expansion**  
For mirror volumes with mirroring multiplicity of two and higher, change the mirroring multiplicity to one, expand the volumes, and then execute synchronization copying for re-mirroring. See the outline of the operating procedures below. These operations are executable without stopping applications using the volumes.
  - 1) Disconnect disks and lower level groups from the mirror group to change the mirroring multiplicity to one.
  - 2) Expand the volume size with the `sdxvolume -S` command.
  - 3) Reconnect the disks and lower level groups disconnected in step 1) with the mirror group.If the mirror volumes are active, resynchronization copying is automatically performed after step

3) is done. If the mirror volumes are inactive, similar copying is automatically performed when they are started.

- **Stripe Type Volume and Concatenation Type Volume Expansion**

The capacity of stripe type volumes and concatenation type volumes cannot be expanded. To expand one of these types, back up data, recreate the volume, and then restore the data back to the volume. In configurations where a striping group or a concatenation group is connected to a mirror group (with any multiplicity), volume expansion applied using the striping or concatenation feature is possible.



See

For the methods of backing up and restoring, see "Chapter 6 Backing Up and Restoring."

- **Concatenation and Online Volume Expansion**

Even if there is no sufficient continuous free space after the last block of a volume, by concatenating unused disks, online volume expansion will be available. Online volume expansion is available for volumes that meet all of the following conditions:

- Volumes belong to a mirror group;
- To the mirror group, one or more concatenation groups are connected;
- To each of the concatenation groups, one or more disks are connected.

To use this function, create volumes in configuration conforming to these conditions in advance. For example, if there is only one available disk, connect only the disk to a concatenation group, connect the concatenation group to a mirror group, and create volumes in the mirror group.

- **Expansion of Areas Used by Applications**

After volumes are expanded, applications such as file systems and databases need to recognize the expanded areas with methods specific to the applications.

If an application using a volume cannot recognize an expanded volume area, do not expand the volume. If such a volume is expanded, the application may no longer operate normally or volume data may be unavailable. A volume that contains the GFS Shared File System cannot be expanded.

## **A.2.12 Object Operation When Using Proxy**

---

If classes, groups, and slices are related to master volumes or proxy volumes, operations that change the class attributes, change the group configurations or attributes, or handle the slices cannot be performed. To perform such operations, cancel the relationship between the master and the proxy once. For objects that are related to the master and the proxy:

- Master volumes can be started or stopped.
- Proxy volumes that are parted from the master can be started or stopped.
- New volumes can be created within a group other than the master group or the proxy group and existing volumes other than the master or the proxy can be removed.
- The hot spare function is available for groups with master volumes. Spare disks are not connected to groups with proxy volumes.

The following operations can be performed, but the operation will result in an error if copying is in process, or there is an EC session, between the master and the proxy.

- Attributes of master and proxy volumes can be changed with the `sdxattr -V` command.
- The disk connected to a group with a master volume and a proxy volume can be prepared for swapping with the `sdxswap -O` command. After the swapping is complete, the disk will be ready for use with the `sdxswap -I` command.
- Synchronization copying of the master volume or the proxy volume separated from the master volume can be started, cancelled, interrupted and resumed, and the parameters of the synchronization copying changed. However, synchronization copying between volumes or its parameters cannot be changed.
- Master volume can be restored with the `sdxfix -V` command.

For EC sessions between the master and the proxy, check the FUNC field displayed with the `sdxinfo` command. If the master and the proxy are in parted status, a copy session can be canceled with the `sdxproxy Cancel` command.

## A.2.13 Using the Advanced Copy Function in a Proxy Configuration

---

GDS Snapshot performs copying between the master and the proxy without imposing loads on primary servers or a SAN, by working in cooperation with Advanced Copy functions (EC, OPC) of ETERNUS6000 series, ETERNUS3000 series, and GR700 series disk arrays.

When the Advanced Copy function is available, it is used for copying between the master and the proxy. However, in the following situations, a soft copy function (a copy function of a GDS driver operating on a server) is used.

- The Advanced Copy function is not used when:
  - Use of soft copy functions was specified explicitly using the `sdxproxy` command with the `-e softcopy` option
  - The copy destination volumes are in a mirroring configuration.  
Note, however, that even if the master as the copy destination is in a mirroring configuration, OPC is available for copying from the proxy when:
    - Executing `[Operation]:[Proxy Operation]:[Restore]` in GDS Management View and selecting "No" to "Rejoin" in the `[Restore Master]` dialog box. For details see "5.3.3 Restore."
    - Executing the `sdxproxy Restore` command. For details see "D.12 `sdxproxy` - Proxy object operations."
  - A lower level group is connected to a group to which master volumes or proxy volumes belong.
  - The number of concurrent EC or OPC sessions has reached the upper limit defined by the disk array unit.



See

The number of allowed concurrent sessions is either the upper limit within one physical disk (LU) or the upper limit within one disk array unit. For details see the handbook of the relevant disk array.

For details about the upper limit of supported concurrent sessions, see the disk array device handbook.

- Multiple proxy volumes are related to a master volume and the number of proxy volumes with EC sessions has reached the upper limit (16 volumes).
- Disks that constitute the master or the proxy were registered with a class before installing the Advanced Copy function on the disk array unit.
- Disks that constitute the master or the proxy were registered with a class before installing GDS Snapshot.

In these situations, operations based on OPC functions are impossible. For such operations see "A.2.14 Instant Snapshot by OPC."

EC is used for synchronization copying from a master to a proxy after joining or rejoining them, copying for maintaining synchronization, and recording the portion updated while a master and a proxy are parted.

OPC is used for synchronization copying, instant snapshot processing, and restoring a master using the proxy data.

If any EC sessions exist between a master and a proxy, OPC cannot be initiated between the master and another proxy.

When both EC and OPC is available as with ETERNUS6000 series, EC has precedence over OPC. Once an EC session is stopped, you cannot use the EC function afterwards. EC sessions are stopped in the following situations.

- EC sessions are stopped when:
  - Executing `[Operation]:[Proxy Operation]:[Part]` in GDS Management View and selecting "Yes" to "Instant Snapshot" in the `[Part Proxy]` dialog box
  - Executing `[Operation]:[Proxy Operation]:[Update]` in GDS Management View
  - Executing `[Operation]:[Proxy Operation]:[Restore]` in GDS Management View
  - Canceling the EC sessions with the `sdxproxy Cancel` command
  - Joining a master and a proxy and using a soft copy function with the `sdxproxy Join -e softcopy` command
  - Creating instant snapshots with the `sdxproxy Part -e instant` command
  - Rejoining a master and a proxy and using a soft copy function with the `sdxproxy Rejoin -e softcopy` command

- Updating a proxy with the `sdxproxy Update` command
- Restoring master data back from a proxy with the `sdxproxy Restore` command

To make the EC function available after conducting these operations, break the relationship between the master and the proxy once and rejoin them.

To check the modes of the copying in execution, use either:

- The [Copy Type] field of the slice information field in GDS Management View
- The CPTYPE field displayed with the `sdxinfo` command

Additionally, the types and the statuses of sessions between the master and the proxy can be viewed in the FUNC field and the CPSTAT field displayed with the `sdxinfo` command.



**Note** **Advanced Copy Control**

When the Advanced Copy function is available, executing the `sdxproxy` command directs GDS to control the Advanced Copy on the master and the proxy. Do not use any other methods other than the `sdxproxy` command to apply such control on a master and a proxy.

## **A.2.14 Instant Snapshot by OPC**

---

The following functions are based on OPC functions of Fujitsu's ETERNUS storage systems.

- Instant snapshot by OPC
  - [Operation]:[Proxy Operation]:[Update] in GDS Management View
  - "Yes" to "Instant Snapshot" in the [Part Proxy] dialog box invoked through [Operation]:[Proxy Operation]:[Part] in GDS Management View
  - `sdxproxy Update` command
  - `sdxproxy Part -e instant` command
- Master restoration by OPC
  - "No" to "Rejoin" in the [Restore Master] dialog box invoked through [Operation]:[Proxy Operation]:[Restore] in GDS Management View
  - `sdxproxy Restore` command

These functions are unavailable with disk units other than the following types.

ETERNUS6000 series, ETERNUS3000 series, GR720, GR730, GR740

These functions are also unavailable under such conditions that prevent the use of Advanced Copy functions as described in "A.2.13 Using the Advanced Copy Function in a Proxy Configuration."

## **A.2.15 Ensuring Integrity of Snapshot Data**

---

If snapshots are created while an application is accessing the volume, the snapshots may result from incomplete volume data and the data integrity may not be ensured.

To ensure the integrity of your snapshot data, you must stop the application that is accessing the volume in advance. After creating the snapshot, start the application again.

For example, if the volume (master) has been used as a file system such as GFS and ext3, before and after creating snapshots, unmount the system and remount it in order to ensure the snapshot data integrity.

To create a snapshot while running the application, the file system or database system you are using to manage the data must be able to ensure data integrity.

For an example, see "6.2 Online Backup and Instant Restore through Proxy Volume."



## A.2.16 Data Consistency at the time of Simultaneous Access

---

When the same block within a volume is accessed simultaneously from multiple nodes, data consistency is maintained by access exclusion control performed by the application that accesses the shared disk simultaneously.

## A.2.17 Volume Access Mode

---

There are two types of volume access modes: "Default Access Mode" which is set by default as an access mode attribute and "Current Access Mode" which is set for a volume that is activated. "Current Access Mode" is valid only while the volume is activated and will become invalid when the volume is stopped. When the volume is restarted, it will start in "Default Access Mode," except for when the access mode is specified at the time of restart.

For example, if you wish to normally use the volume in the read and write mode, and temporarily switch to the read only mode, set the access mode attribute to "rw", and use the `sdxvolume -N` command specifying the `-e mode=ro` option to activate the volume in the read only mode temporarily.

The "default access mode" for a shadow volume is ro (read only) and it cannot be changed, but the "current access mode" can be set to either ro (read only) or rw (read and write).

## A.2.18 Operation in Cluster System

---

Understand and pay attention to the following points when changing the configuration in a cluster system.

- Before registering disks with a class, create disk resources through the cluster automatic resource registration.



See

For details about the automatic resource registration, see "Appendix H Register shared disk units"

- Disks for which disk resources has not been created yet cannot be registered with a shared class.
- For disks to be registered with a shadow class, disk resource creation is not required.
- Do not create disk resources of certain disks in multiple cluster domains.
- To register MPHD disks or MPLB disks from GDS Management View, after conducting automatic resource registration, execute [Update Physical Disk Information] in the [Operation] menu prior to the registration.
- To conduct object operation in a cluster system, enable cluster control.

## A.2.19 Disk Switch

---

The disk switch function is available only when using an application that controls the disk switch function. Unless the application's manual instructs to creation of switch groups and switch volumes, do not create switch groups.

## A.2.20 Shadow Volume

---

### Rebooting a Node

The configuration information of a shadow volume is not saved on the private slice, but managed in the memory. For this reason, the shadow volume configuration definitions are cleared when the node on which the shadow volume is defined is rebooted. However, the device special file remains. If such a device special file is left not deleted, issues as described below may occur.

Before intentional shutdowns, it is recommended to remove shadow volumes. If a shadow volume is removed in the proper procedure, the device special file is also deleted.

When a node is shut down leaving the relevant shadow volume not removed, or if a node on which a shadow volume is defined is rebooted unexpectedly because of an event such as a panic and a power cutoff, the device special file for the shadow volume must be deleted in the following procedure.

[Procedure]

1. Check the system for existing classes.  
In the following example, there are Class1, and Class2.

```
# sdxinfo -C
OBJ  NAME      TYPE      SCOPE          SPARE
-----
class Class1  shared   node1:node2    0
class Class2  shared   node1:node2    0
```

2. Find the directories containing the device special files of classes.  
In the following example, RootClass, Class1, and Class2 are the directories for the device special files of those existing classes, and \_adm and \_diag are the special files used by GDS. Class3, other than those directories, is the directory for the device special file of the extinct shadow class.

```
# cd /dev/sfdsk
# ls -F
Class1/   Class2/   Class3/   _adm@    _diag@
```

3. Delete the directory for the device special file of the extinct shadow class.

```
# rm -r Class3
```

Even if the device special file of an extinct shadow volume remains, no problem will arise if a shadow volume in the same configuration, of the same class name, and with the same volume name is re-created.

Otherwise, the following issues will occur. If a logical volume or a shadow volume is created in the situation that the device special file, `/dev/sfdsk/Shadow Class Name/[r]dsk/Shadow Volume Name`, of an extinct shadow volume remains, the minor number of the created volume may become the same as the minor number of `/dev/sfdsk/Shadow Class Name/[r]dsk/Shadow Volume Name`. If this happens the newly created volume is unintentionally accessed without recognizing that the shadow volume no longer exists, and it can cause application errors and corruption of data on the newly created volume.

## Accessing a Shadow Volume

Shadow volumes and the corresponding logical volumes are managed independently. For example, the change of the slice status in one volume is not updated in the slice status in the other volume. For this reason, you must note the following operational particulars when using shadow volumes.

## Synchronization of Shadow Volumes

When a shadow volume is created in another domain (domain beta) for the disk area managed as a mirror volume in a certain domain (domain alpha), the mirror volume in domain alpha and the shadow volume in domain beta cannot be accessed simultaneously. If they are accessed simultaneously, the following issues arise.

- If an I/O error occurs in the slice configuring the mirror volume in domain alpha, that slice becomes INVALID and is detached from the mirror volume. However, GDS in domain beta does not detect this I/O error, and consequently the shadow slice is not made INVALID and is not detached from the shadow volume. Here, synchronization of the shadow volume is not ensured.
- Likewise, if an I/O error occurs in the shadow slice in domain beta, the slice in the corresponding mirror volume in domain alpha is not made INVALID and is not detached from the mirror volume. Here, synchronization of the mirror volume is not ensured. If an I/O error occurs on the shadow slice, working around, such as swapping the disks and resynchronization copying of the mirror volume, is required in domain alpha.

These particulars apply when the disk area for a mirror volume and a shadow volume are identical. A mirror volume and a shadow volume corresponding to a replica of the mirror volume (a temporarily detached slice, a proxy volume or a copy destination disk area for a disk unit's copy function) can be accessed simultaneously.

## Just Resynchronization Mechanism (JRM) for Volumes

When a shadow volume is created in another domain (domain beta) for the disk area managed as a mirror volume in a certain domain (domain alpha) and accessed, the following must be set up for the mirror volume in domain alpha.

- Mirror volumes must be inactivated to prevent access to the mirror volume corresponding to the shadow volume.
- JRM for volumes must be enabled ("on") for the mirror volume corresponding to the shadow volume.

These settings are necessary for the following reasons.

If a node in domain alpha panics and resynchronization copying is conducted on the mirror volume in domain alpha while the shadow volume is accessed in domain beta, synchronization between the shadow volume and the mirror volume is no longer ensured. Though the settings as above, resynchronization copying is no longer conducted on the mirror volume in domain alpha even if a node in domain alpha panics.

The settings as above are necessary only for a mirror volume created for the disk area identical to the shadow volume's disk area. When a shadow volume corresponding to a replica of a mirror volume (a temporarily detached slice, a proxy volume or a copy destination disk area for a disk unit's copy function) is created, these settings are not necessary for the copy source mirror volume.



### Information Resynchronization Copying after Panic

Resynchronization copying is not conducted after panic when JRM for volumes is enabled ("on") and that volume is not written in. Resynchronization copying occurs after panic when JRM for volumes is disabled ("off") and that volume is active.

## Just Resynchronization Mechanism (JRM) for Slices

When a slice is temporarily detached from a mirror volume in a certain domain (domain alpha) and data is written from a shadow volume in another domain (domain beta) to the area of this volume or slice, JRM for slices must be disabled ("off") prior to re-attaching the slice.

If JRM for slices is enabled ("on"), the following issue arises.

When JRM for slices is enabled ("on"), only the difference between the volume and the slice is copied by re-attaching the slice. The difference information for the volume and the slice is managed by JRM for slices in domain alpha. However, JRM for slices in domain alpha does not recognize write events from domain beta, and the difference resulting from data being written from domain beta are not updated in the difference information. The difference resulting from write events from domain beta, therefore, are not copied when the slice is re-attached while JRM for slices is "on" in domain alpha. As a result, synchronization of the volume is no longer ensured.

## Just Resynchronization Mechanism (JRM) for Proxies

If a proxy volume is parted from the master in a certain domain (domain alpha) and data is written from a shadow volume in another domain (domain beta) to the area of this master or proxy, JRM for proxies must be disabled ("off") prior to rejoining the proxy. In addition, JRM for proxies must be disabled ("off") prior to restoring the master using the proxy.

If JRM for proxies is enabled ("on"), the following issues arise.

When JRM for proxies is enabled ("on"), only the difference between the master and the proxy is copied by rejoining or restoring. The difference information for the master and the proxy is managed by JRM for proxies in domain alpha. However, JRM for proxies in domain alpha does not recognize write events from domain beta, and the difference resulting from data being written from domain beta are not updated in the difference information. The difference resulting from write events from domain beta, therefore, are not copied when the proxy is rejoined or the master is restored while JRM for proxies is "on" in domain alpha. As a result, synchronization between the master and the proxy is no longer ensured.

When one of a disk unit's copy function (EC, REC) with a resynchronization feature based on equivalent copy capability is used for master-to-proxy copy processes, data written from domain beta is also updated in the difference information managed by these disk unit's copy functions. Under these circumstances, JRM for proxies do not have to be disabled ("off") prior to rejoining. Note, however, that JRM for proxies must be disabled ("off") prior to restoring since necessity of synchronization copying is

determined based on the difference information managed by JRM for proxies. To ensure the data integrity, it is recommended to disable JRM for proxies prior to rejoining even when a disk unit's copy function is used.



### Information **A Copy Function of a Disk Unit with Resynchronization Feature Based on Equivalent Copy**

When just resynchronization copying from a master to a proxy is conducted with one of disk unit's copy functions (EC, REC) with a resynchronization feature based on equivalent copy capability, this feature becomes effective regardless of whether JRM for proxies is "on" or "off".

### **Writing into Shadow Volumes**

Data may be written to a shadow volume even if the operation for writing is not especially intended. For example, executing mount(1M) (excluding when using the -o ro option), fsck(8) or mkfs(8) results in the write operation.

When a proxy is rejoined, a master is restored, or a slice is reattached once a shadow volume is created, it is recommended to disable the just resynchronization mechanism mode (JRM) regardless of whether or not data is written into the shadow volume in order to ensure the data integrity.

---

## **A.2.21 GDS Management View**

---

When any operation that changes the physical disk configuration, such as addition or deletion of disk units, is conducted during system operation, update physical disk information with new information. Execute [Update Physical Disk Information] on the [Operation] menu when:

- The power of the disk unit was turned on after the system was booted.
- The disk unit became unavailable for some kind of problem during system operation, but was recovered without system reboot.
- The configuration of devices was changed.

---

## **A.2.22 Upgrading Instructions**

---

The following incompatibility exists between PRIMECLUSTER GDS Snapshot 4.1A20 and 4.1A30.

- PRIMECLUSTER GDS Snapshot 4.1A20  
Even if delay time that is out of range of 0 to 1000 in milliseconds is specified with the -e delay option of the sdxproxy command, the sdxproxy command does not result in an error and the specified delay time will be in effect.
- PRIMECLUSTER GDS Snapshot 4.1A30  
If delay time that is out of range of 0 to 1000 in milliseconds is specified with the -e delay option of the sdxproxy command, the sdxproxy command results in an error and the following error message is output. *msec* indicates the delay time specified with -e delay=*msec*.

ERROR: <i>msec</i> : invalid delay value
--

## A.3 General Points

### A.3.1 Guidelines for Mirroring

---

Pay attention to the following guidelines when constructing mirroring configurations.

- Connecting disks and lower level groups with the same available sizes to mirror group is recommended.  
The available size of a mirror group (the capacity available as volume) is the same as the available size of the smallest disk or lower level group that is connected.  
When connecting disks or lower level groups with different available sizes to a mirror group, you will only be able to use the capacity of the smallest disk or lower level group. For example, if a 4 GB disk and a 9 GB disk is connected to one mirror group, only 4 GB out of the 9 GB disk will be accessible.
- Mirroring disks with similar performance specifications, or groups with the same configuration (including the performance specifications of disks that is connected to the group) is recommended.  
When mirroring disks with different performance specifications such as revolution speed, the read performance becomes unbalanced and the write performance will depend on the slower disk performance.  
The same applies when mirroring disks and groups, or when mirroring groups with different configuration.

### A.3.2 Guidelines for Striping

---

Pay attention to the following guidelines when constructing striping configurations.

- In order to improve I/O performance with striping, it is necessary to adjust the stripe width and the number of stripe columns depending on the way an application accesses the disk.  
If the striping configuration is not appropriate, you cannot gain much performance improvement. And, depending on the way an application accesses the disk, the performance may not improve even after adjusting the stripe width or the number of stripe columns.
- Do not make the stripe widths too large.  
The sizes of stripe groups and stripe volumes are rounded to the common multiple of stripe width times the number of stripe columns and cylinder size. Therefore, if the stripe width is too large, use of the disk area may be inefficient or a volume with the intended size may be created.
- Where possible, connect disks and lower level groups with the same available sizes to the same stripe group.  
The available size of the stripe group (available capacity as volumes) equals the available size of the smallest disk or the lower level group connected to the stripe group multiplied by the number of stripe columns and rounded down to the common multiple of stripe width times the number of stripe columns and cylinder size.  
When connecting disks or lower level groups with different available sizes to a stripe group, the larger disk or lower level group will only be able to use the capacity of the smaller disk or lower level group. For example, if a 4 GB disk and a 9 GB disk are connected to one stripe group, the 9 GB disk will only be able to use approximately 4 GB. Which means, the available size of stripe group will be approximately 8 GB (4 GB x 2).
- Where possible, striping across disks with similar performance specifications is recommended.  
When striping disks with different performance specifications such as revolution speed, the performance becomes unbalanced and will depend on the slower disk performance.
- Using striping in combination with mirroring is recommended.  
In a striping configuration, the risk of losing data from a disk failure increases as more disks are involved compared to a usual disk configuration.  
By mirroring stripe groups, both the I/O load balancing and data redundancy can be achieved at the same time.

### **A.3.3 Guidelines for Concatenation**

---

Pay attentions to the following guidelines when constructing concatenation configurations.

- The available size of the concatenation group (available capacity as volumes) equals the total of the available size of disks connected to the concatenation group.
- Where possible, concatenating disks with similar performance specifications is recommended. When concatenating disks with different performance specifications such as the revolution speed, the performance becomes unbalanced.
- Concatenation in combination with mirroring is recommended. When concatenating disks, the risk of losing data from a disk failure increases as more disks are involved compared to a usual disk configuration. By mirroring concatenation groups, large capacity and data redundancy can be achieved at the same time.

### **A.3.4 Guidelines for Combining Striping with Mirroring**

---

Pay attention to the following guidelines when striping and mirroring simultaneously.

- Where possible, mirroring stripe groups with similar configuration is recommended. See "A.3.1 Guidelines for Mirroring" and "A.3.2 Guidelines for Striping" also.

## **Appendix B Log Viewing with Web-Based Admin View**

For details, see the supplementary "Web-Based Admin View Operation Guide."





## **Appendix C Web-Based Admin View Operating Environment Setting**

For details, see the supplementary "Web-Based Admin View Operation Guide."



## Appendix D Command Reference

This appendix discusses the commands provided by GDS and GDS Snapshot. This appendix explains the format and facility of commands, specifiable options, and return values. GDS provides the following commands.

Command	Function
sdxcass	Class operations
sdxdisk	Disk operations
sdxgroup	Group operations
sdxvolume	Volume operations
sdxslice	Slice operations
sdxinfo	Display object configuration and status information
sdxattr	Change attribute values of an object
sdxswap	Swap disk
sdxfix	Restore a crashed object
sdxcopy	Synchronization copy operations
sdxparam	Configuration parameter operations

GDS Snapshot provides the following commands.

Command	Function
sdxproxy	Proxy object operations
sdxshadowdisk	Shadow disk operations
sdxshadowgroup	Shadow group operations
sdxshadowvolume	Shadow volume operations

## D.1 sdxclass - Class operations

### SYNOPSIS

`sdxclass -R -c class`

### DESCRIPTION

Use `sdxclass` to perform operations on class objects (excluding shadow class) specified by *class*. You must be superuser to use this command.

### PRIMARY OPTIONS

You can use the following option.

#### **-R**

##### **Remove**

Removes the class definition specified by *class*. The definition is removed from all nodes.

A disk registered with *class* will be removed with the *class*. However, if there is a group or a volume, the class will not be removed.

To place a removed disk under GDS management again, you need to re-register the physical disk in the class.

For further details, see "D.2 sdxdisk - Disk operations."

### SUB OPTIONS

SUB OPTIONS are as follows:

#### **-c *class***

The *class* indicates the class name that is the target of the operation.

### RETURNED VALUE

Upon successful completion, a value of 0 is returned. Otherwise, a non-zero value is returned.

## D.2 sdxdisk - Disk operations

### SYNOPSIS

```
sdxdisk -C -c class -g group -d disk,...  
        [-a attribute=value [,attribute=value]] [-e delay=msec ]
```

```
sdxdisk -D -c class -g group -d disk
```

```
sdxdisk -M -c class -a attribute=value [,attribute=value,...]  
        -d device=disk [:type ] [,device=disk [:type ],...] [-e chkps]
```

```
sdxdisk -R -c class -d disk
```

### DESCRIPTION

Use sdxdisk to perform operations on disk objects (excluding shadow disk) specified by *disk*. You must be superuser to use this command.

### PRIMARY OPTIONS

You can use either of the following options.

#### -C

##### Connect

Connects one or more disks (single disks or undefined disks) specified by *disk*,... to a group specified by *group*. The *class* indicates the class name with which *disk* is registered. To connect *disk* to a group other than a switch group, specify a disk connected to all the nodes that belong to the scope of *class*.

If no group with the same name as *group* exists, a new group is created. One class cannot include switch type groups and groups of other types together.

You cannot connect a spare disk to a group. You cannot connect a single disk to an existing group. Also, multiple single disks cannot be connected to the same group simultaneously.

The disk attribute will be changed to match the attribute of the group (mirror, stripe, concatenation, or switch) it is connected to. Disks and lower level groups connected to a group are mirrored, striped, concatenated or made switchable according to the type attribute.

Details on connecting a disk to a mirror group, a stripe group, a concatenating group and a switch group are explained below.

#### - When connecting to a mirror group

Disks and lower level groups connected to the same mirror group will mirror each other. When only one disk or one lower level group is connected, the volume created within that mirror group will not be mirrored. When configuring a mirroring environment with "n"-way multiplexing, "n" numbers of disks or lower level groups must be connected. A maximum of eight-way multiplex mirroring is supported.

When one or more volumes already exist within the mirror group specified by *group*, the slice configuration of disk or lower level group that is already connected to *group* will be automatically copied to the newly connected *disk*. Also, when there is an activated volume within *group*, volume data in addition to the slice configuration will be automatically copied upon returning from the sdxdisk command, therefore increasing the mirroring multiplicity.

By connecting a single disk with single volumes to a mirror group, single volumes will be changed to mirror volumes.

The available size of the mirror group (available capacity as volumes) will be the same as the available size of the smallest disk or the lower level group connected to the mirror group.

When a keep disk is connected, the available size of the group will be the same as the available size of the keep disk. If connecting *disk* results in decreasing the available size of *group*, a warning message will be sent to standard error output.

### - When connecting to a stripe group

Disks specified by *disk*,... will be connected to *group* in the order they are listed. Disks and lower level groups connected to the same stripe group will each configure a stripe column, and will be striped in the order they were connected. When only one disk or one lower level group is connected, a volume cannot be created within that stripe group. When striping "n" number of columns, "n" number of disks or lower level groups must be connected. A minimum of two columns and a maximum of 64 columns are supported.

When a stripe group specified by *group* already exists, stripe columns will be added after the existing stripe columns in *group*, in the order they are specified by *disk*,... However, you cannot increase stripe columns by connecting disks to stripe groups with volumes, or to stripe groups connected to a higher level group.

The available size of the stripe group (available capacity as volumes) equals the available size of the smallest disk or the lower level group connected to the stripe group multiplied by the number of stripe columns and rounded down to the common multiple of stripe width times stripe columns and cylinder size. If connecting *disk* decreases the available size of *group*, a warning message will be sent to standard error output.

You cannot connect a single disk to a stripe group.

### - When connecting to a concatenation group

Disks connected to the same concatenation group specified by *disk*,... will be concatenated in the order they are listed. A maximum of 64 disks can be concatenated.

The available size of the concatenation group (available capacity as volumes) equals the total of the available size of disks connected to the concatenation group.

The available size of the group can be increased by connecting disks to the existing concatenation group. When a concatenation group specified by *group* already exists, disks will be added in the order they are specified by *disk*,... after the disk that was last concatenated in *group*. However, you cannot add a disk to a concatenation group that is connected to the highest level stripe group which already has a volume. Also if the concatenation group specified by *group* is connected to a stripe group connected to a mirror group, a disk cannot be added.

You cannot connect a single disk to a concatenation group.

### - When connecting to a switch group

One of two disks connected to a switch group becomes the active disk and the other one becomes the inactive disk. Use the `sdxattr -G` command to switch the disks from active to inactive and vice versa. A switch group can connect a maximum of two disks. If only one disk is connected, the disk acts as the active disk and an active disk switch cannot be performed.

To create a switch group specified by *group*, specify the active disk in the `-a actdisk` option. The other disk not specified by the `-a actdisk` option becomes the inactive disk. When connecting *disk* to an existing switch group specified by *group*, the *disk* becomes the inactive disk.

Unlike mirror groups, even if a switch group specified by *group* already includes volumes, synchronization copying to the newly connected *disk* is not performed. To perform a disk switch for continuous services in the event of an error in the active disk, previously create copies of data from the active disk to the inactive disk with the disk unit's copy function and so on.

The available size of a switch group (capacity available for volumes) conforms to the available size of the smallest disk connected to the switch group. If the available size of *group* decreases as a result of *disk* connection, a warning message is sent to standard error output.

In classes that include disks other than undefined disks or groups other than switch groups, switch groups cannot be created.

*class* must be a shared class of which scope includes 2 nodes. The physical scope of the active disk and the inactive disk must also meet either the following conditions.

- The active disk and the inactive disk are connected to both the nodes included in the scope of *class* and are not connected to nodes not included in that scope.
- The active disk is connected to only one of the nodes included in the scope of *class* and the inactive disk is connected to the other node included in that scope.

## -D

### **Disconnect**

Disconnects a disk (including a spare disk) specified by *disk* from a group specified by *group*. The *class* indicates the class name with which the *disk* is registered, and the *group* indicates the group name to which *disk* is connected.

The disconnected disk will return to its original type attributes (single disk or undefined disk).

If only *disk* is connected to *group*, *group* will automatically be removed upon disconnecting *disk*. However, when *disk* is the only object connected to *group* and *group* is connected to a higher level group, disconnection will result in an error. In such case, disconnect *group* from the higher level group using the `sdxgroup -D` command, and then disconnect *disk*.

You cannot disconnect *disk* if the disconnection will result in a change in the status of any of the existing volumes within *group*.

Conditions on when you cannot disconnect a disk from a mirror group, a stripe group, a concatenation group or a switch group are explained below.

#### **- When disconnecting from a mirror group**

For example, you cannot disconnect *disk* from a mirror group if one or more volumes exist within the mirror group specified by *group*, and *disk* is the only object connected to *group*.

#### **- When disconnecting from a stripe group**

You cannot disconnect a disk from a stripe group with one or more existing volumes, or from a stripe group connected to a higher level group.

#### **- When disconnecting from a concatenation group**

The only disk you can disconnect from a concatenation group is the disk that was concatenated last.

Disks containing volume data cannot be disconnected from a concatenation group.

You cannot disconnect a disk from a concatenation group that is connected to the highest level stripe group with one or more existing volumes. Also, if the concatenation group specified by *group* is connected to a stripe group connected to a mirror group, the disk cannot be disconnected.

#### **- When disconnecting from a switch group**

Inactive disks can be disconnected regardless whether or not volumes exist.

The active disk can be disconnected only if the switch group includes no volume or inactive disk.

If the switch group includes volumes, before disconnecting the active disk, remove those volumes. If the switch group includes the inactive disk, switch the active disk to it with the `sdxattr -G` command and then disconnect the former active disk.

## -M

### **Make**

Registers one or more physical disks, specified by *device*, with *class*. The *class* gives the name of the destination class. Once physical disks have been registered, they can then be managed using GDS. A disk managed by GDS is called an SDX disk. Users will use the disk name specified by *disk* to perform operations on the disk.

If no class with the name specified by *class* already exists, then one is automatically created.



#### **Note**

Since the `sdxdisk` command initializes the registered physical disks, when registering a physical disk containing data, you must first create data backup.

## **-R**

### **Remove**

Removes a disk specified by *disk* from a class specified by *class*. The *class* indicates the class name with which the *disk* is registered.

Once the disk is removed, it can no longer be managed using GDS.

When the last disk is removed from a *class*, that class definition is automatically removed.

A disk cannot be removed when a volume exists within *disk*, or when *disk* is connected to a group.

If removal of *disk* will result in *class* closure, the *disk* cannot be removed. The *class* will be closed when it includes:

- less than 3 disks in ENABLE status and no disk normally accessible
- three to 5 disks in ENABLE status and less than 2 disks normally accessible
- six or more disks in ENABLE status and less than 3 disks normally accessible

## **SUB OPTIONS**

SUB OPTIONS are as follows:

### **-a *attribute=value[,attribute=value]* (when using **-C**)**

When using the **-C** option and defining a new group name with the **-g** option, a new group is automatically created. Using this option sets the *attribute* of the created *group* to *value*. The *attribute* indicates the attribute name, and *value* indicates the attribute value. Always separate *attribute* and *value* with an equal (=) sign. When indicating multiple attributes, specifiers should be combined using commas(,) as the delimiter.

If no group is created, indicating a different attribute value from the existing *group* will result in an error. To change the attribute value of an existing *group*, use the `sdxattr -G` command.

You can indicate the following combination to *attribute* and *value*.

If multiple attributes are indicated and an error results because of any part of them, the entire process is canceled.

### **type=mirror, type=stripe, type=concat or type=switch(default is mirror)**

Sets type attribute of *group*. When *class* is not a shared class of which scope includes 2 nodes, specifying "switch" will also result in an error.

#### **mirror**

Sets type attribute to "mirror".

#### **stripe**

Sets type attribute to "stripe".

#### **concat**

Sets type attribute to "concatenation".

#### **switch**

Sets type attribute to "switch".

### **width=*blks* (default is 32)**

Sets the stripe width of *group*. The *blks* indicates the stripe width in blocks (base 10). One block is 512 bytes. For *blks*, you can indicate an integer (one and larger) that is two raised to the power, which is equal to or smaller than the available size of the smallest disk specified by *disk*,.... If *group* is not a stripe group, this option will result in an error.

### **actdisk=*disk***

Sets the active disk of *group*. Specify a disk name of the active disk into *disk*. When *group* is an existing group, not specifying the **-a type=switch** option, or specifying a disk other than that specified by **-d** option into *disk*, will result in an error.



**-a *attribute=value*[,*attribute=value*,...]** (when using **-M**)

When using the **-M** option and defining a new class name with the **-c** option, a class is automatically created. Using this option sets the created *class attribute* to *value*.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign. Specifiers should be combined using commas(,) as the delimiter. If no class is created, this option is ignored.

To change the attributes of an existing class, use the `sdxattr -C` command.

You can use the following *attribute* and *value* pairs in any combination.

If multiple attributes are specified and an error results because of any part of them, the entire process is canceled.

### **type=shared**

Sets the attributes of *class* type.

#### **shared**

Sets the type attribute to "shared."

By combining this with the scope attribute, the objects in the *class* can be shared among multiple nodes including the current node.

A shared type class can include physical disks connected to all the nodes that belong to the scope. When the scope includes 2 nodes, disks connected to only one node in the scope can be registered as undefined disks.

### **scope=node [ :node:... ]**

Sets the node set which share the *class* whose type attribute is specified to be "shared".

In *node*, indicates a node identifier that is defined by PRIMECLUSTER.

### **hs=on** or **hs=off** (default is **on**)

It sets the operation of the hot spare.

#### **on**

Enables the hot spare.

#### **off**

Disables the hot spare. If the operation mode is set to off, spare disk automatic connection is restrained.

### **hsmode=exbox** or **hsmode=bybox** (default is **exbox**)

Sets the spare disk selection mode for automatic connection by hot spare.

#### **exbox**

Sets the spare disk selection mode to the external mode.

If an I/O error occurs in a disk of a disk array unit, this method selects a spare disk that belongs to a different disk case from that of the failed disk. If an I/O error occurs in a disk irrelevant to a disk array unit (such as an internal disk), it selects a spare disk that is connected to a different controller from that of the failed disk. When no applicable unconnected spare disk is found there, a spare disk that belongs to the same disk case or is connected to the same controller as that of the disk with the I/O error is selected.

#### **bybox**

Sets the spare disk selection mode to the internal mode.

If an I/O error occurs in a disk of a disk array unit, this method selects a spare disk that belongs to the same disk case as that of the failed disk. If an I/O error occurs in a disk irrelevant to a disk array unit (such as an internal disk), it selects a spare disk that is connected to the same controller as that of the failed disk. When no applicable unconnected spare disk is found there, spare disk automatic connection is restrained.

### **-c class**

The *class* indicates the class name to which the disk is registered or is to be registered, where the disk is the target of the operation.

**-d *device=disk* [:*type*] [,*device=disk* [:*type* ],...]** ( when using **-M** )

The *device* indicates the name of the physical disk, the *disk*, the name of the disk, and *type*, the type attribute of the disk.

The *device* must always be followed by an equal sign (=), and if a type is given, it is delimited from *disk* by a colon (:).

To register multiple *devices*, combine multiple definitions with a comma (,) as the delimiter.

The physical disk name can be specified with one of the following formats:

**sdX** ( for normal hard disks )

**emcpowerX** ( for emcpower disks )

X indicates the device identifier.

The *type* can be indicated as one of the following. If no type is indicated, the default value of undef (undefined disk) is used. If *device* is not connected to part of nodes included in the scope of *class*, or if a switch group exists in *class*, specifying a type other than "undef" will also result in an error.

**single**

Single disk. Single volumes can be created on it.

**spare**

Spare disk.

**undef**

Undefined disk.

When "spare" is specified for *type*, and the available size of *device* is smaller than the available size of the smallest mirror group in *class*, a warning message notifying that the hot spare function may not be available will be sent to standard error output.

When "single" is specified for *type*, *device* will be registered as a single disk.

For a "single" disk, you can create single volumes on it using the `sdxvolume` command without connecting the disk to any group.

**-d *disk*** (when using **-D**, **-R**)

The *disk* indicates the disk name that is the target of the operation.

**-d *disk*,...** (when using **-C**)

The *disk* indicates the disk name that is the target of the operation. To indicate multiple disks, separate each disk name with a comma (,) as the delimiter.

**-e *chkps*** (when using **-M**)

Registers *device* with *class* even if the private slice exists in the *device*, as far as disk identification information (class and disk names) stored in the private slice matches identification information of a disk already registered with the *class*. For example, if *device* contains a copy of the private slice of a disk that is already registered with *class*, to register the *device* to the *class*, turn on this option.

**-e *delay=msec*** (when using **-C**)

When a disk is connected to a mirror group, data contained in the volume will be copied as needed.

This option delays the issuing of the input/output request to the disk at the time of copying by milliseconds specified by *msec*, allowing adjustment for the influence on the application accessing the volume. The value is set to 0 by default. Values from 0 to 1000 may be specified for *msec*.

If *group* is not a mirror group, this option is ignored.

**-g *group*** (when using **-C**, **-D**)

The *group* indicates the group name to which the disk is connected, or is to be connected, where disk is the target of the operation.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.3 sdxcgroup - Group operations

### SYNOPSIS

```
sdxcgroup -C -c class -h hgroup -l lgroup,...  
          [-a attribute=value[,attribute=value]] [-e delay=msec ]
```

```
sdxcgroup -D -c class -h hgroup -l lgroup
```

```
sdxcgroup -R -c class -g group
```

### DESCRIPTION

Use sdxcgroup to perform operations on group objects (excluding shadow groups). You must be superuser to use this command.

### PRIMARY OPTIONS

You can use one of the following options.

#### -C

##### Connect

Connects one or more groups (stripe groups or concatenation groups) specified by *lgroup*,... to a group (a mirror group or stripe group) specified by *hgroup*. The *class* indicates the name of the class to which *lgroup* belongs.

When no group with the same name as *hgroup* exists, a group will be created automatically.

Groups specified by *hgroup* are referred to as higher level group, and groups specified by *lgroup* are referred to as lower level group.

Lower level groups and disks connected to the same higher level group are mirrored or striped according to the type attribute of the higher level group. Connecting a group to a higher level group does not change the type attribute of the lower level group.

You cannot connect groups when:

- *lgroup* is a mirror group or a switch group
- *hgroup* is a concatenation group or a switch group
- type attributes of *lgroup* and *hgroup* are the same

A group that already contains volumes cannot be connected to another group. Connecting a group to a mirror group and a stripe group is explained below.

#### - When connecting to a mirror group

You can connect one or more groups (stripe group or concatenation group) specified by *lgroup*,... to *hgroup* which is a mirror group.

Disks and lower level groups connected to the same mirror group will mirror each other. When only one disk or one lower level group is connected, volumes created within that mirror group will not be mirrored. When configuring a mirroring environment with "n"-way multiplexing, "n" numbers of disks or lower level groups must be connected. A maximum of eight-way multiplex mirroring is supported.

When one or more volumes already exist within the mirror group specified by *hgroup*, the slice configuration of disk or lower level group that is already connected to *hgroup* will be automatically copied to the newly connected *lgroup*. Also, when there is an activated volume within *hgroup*, volume data in addition to the slice configuration will be automatically copied upon returning from the sdxcgroup command, therefore increasing the mirroring multiplexity.

The available size of the mirror group (available capacity as volumes) will be the same as the available size of the smallest disk or the lower level group connected to the mirror group. If connecting *lgroup* decreases the available size of *hgroup*, a warning message will be sent to standard error output.

### - When connecting to a stripe group

You can connect one or more groups (concatenation group) specified by *lgroup,...* to *hgroup* which is a stripe group. Groups specified by *lgroup,...*, will be connected to *hgroup* in the order they are listed.

Disks and lower level groups connected to the same stripe group will each configure a stripe column, and will be striped in the order they are connected. When only one disk or one lower level group is connected, a volume cannot be created within that stripe group. When striping "n" number of columns, "n" number of disks or lower level groups must be connected. A minimum of two columns and a maximum of 64 columns are supported.

When a stripe group specified by *hgroup* already exists, stripe columns will be added after the stripe columns that already exist in *hgroup*, in the order they are specified by *lgroup,...*. However, you cannot increase the stripe columns by connecting groups to stripe groups with volumes, or to stripe groups connected to a higher level group.

The available size of a stripe group (available capacity as volumes) equals the available size of the smallest disk or the lower level group connected to the stripe group multiplied by the number of stripe columns and rounded down to the common multiple of stripe width times stripe columns and cylinder size. If connecting *lgroup* decreases the available size of *hgroup*, a warning message will be sent to standard error output.

## -D

### **Disconnect**

Disconnects group specified by *lgroup* from the higher level group *hgroup*. The *class* indicates class name to which *lgroup* belongs, and *hgroup* indicates the higher level group name to which *lgroup* is connected.

When *lgroup* is the only object connected to *hgroup*, *hgroup* will automatically be removed upon disconnecting *lgroup*. However, when *lgroup* is the only object connected to *hgroup*, and *hgroup* is connected to a higher level group, disconnection will result in an error. In such case, disconnect *hgroup* from its higher level group, and then disconnect *lgroup*.

You cannot disconnect *lgroup* if the disconnection may result in a change in the status of any existing volume within *hgroup*.

Conditions on when you cannot disconnect a group from a mirror group or a stripe group are explained below.

### - When disconnecting from a mirror group

For example, you cannot disconnect *lgroup* from a mirror group if one or more volumes exist within the mirror group specified by *hgroup*, and *lgroup* is the only object connected to *hgroup*.

### - When disconnecting from a stripe group

You cannot disconnect a lower level group from a stripe group with one or more existing volumes, or from a stripe group connected to a higher level group.

## -R

### **Remove**

Remove the group definition specified by *group*. The *class* indicates the class name to which *group* belongs.

Disks and lower level groups connected to *group* will be disconnected. The disconnected disk's attribute will return to its original setting (single disk or undefined disk).

*group* cannot be removed when one or more volumes exist within *group*, or when *group* is connected to a higher level group.

## SUB OPTIONS

SUB OPTIONS are as follows:

**-a *attribute=value*[,*attribute=value*] (when using -C)**

When using the -C option and defining a new group name with the -h option, a new group *hgroup* is automatically created. Using this option sets the *attribute* of the created *hgroup* to *value*.

The *attribute* indicates the attribute name, and *value* indicates the attribute value. Always separate *attribute* and *value* with an equal (=) sign. When indicating multiple attributes, specifiers should be combined using commas(,) as the delimiter.

If no group is created, indicating a different attribute value from the existing *hgroup* will result in an error. You cannot change the attribute value of an existing *hgroup*.

You can indicate the following combination to *attribute* and *value*.

If multiple attributes are indicated and an error results because of any part of them, the entire process is canceled.

**type=mirror or type=stripe (default is mirror)**

Sets the type attribute of *hgroup*.

**mirror**

Sets type attribute to "mirror."

**stripe**

Sets type attribute to "stripe."

**width=*blks* (default is 32)**

Sets the stripe width of *hgroup*. The *blks* indicates the stripe width in block number (base 10). One block is 512 bytes. For *blks*, you can indicate an integer that is two raised to the power (one and larger), which is equal to or smaller than the available size of the smallest group specified by *igroup*,... If *hgroup* is not a stripe group, this option will result in an error.

**-c *class***

The *class* indicates the class name to which the group belongs, where *group* is the target of the operation.

**-e delay=*msec* (when using -C)**

When a group is connected to a mirror group, data contained in the volume will be copied as needed. This option delays the issuing of the input/output request to the disk by milliseconds specified by *msec*, allowing adjustment for the effect on the application accessing the volume. Default is 0. Values from 0 to 1000 may be specified for *msec*.

If *hgroup* is not a mirror group, this option is ignored.

**-g *group* (when using -R)**

The *group* indicates the group name that is the target of the operation.

**-h *hgroup* (when using -C,-D)**

The *hgroup* indicates the higher level group name to which the lower level group is connected or is to be connected, where the lower level group is the target of the operation.

**-l *igroup* (when using -D)**

The *igroup* indicates the lower level group name that is the target of the operation.

**-l *igroup*,... (when using -C)**

The *igroup* indicates the lower level group name that is the target of the operation. To connect multiple groups, separate each group name with a comma (,) as the delimiter.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.4 sdxvolume - Volume operations

### SYNOPSIS

```
sdxvolume -F -c class [ -v volume,... ]
           [-e { allnodes|node=node [:node:... ] } ]

sdxvolume -M -c class { -g group | -d disk } -v volume -s size
           [ -a attribute=value [,attribute=value]] [ -e delay=msec ]

sdxvolume -N -c class [ -v volume,... ]
           [-e [allnodes|node=node [:node:...]],delay=msec, mode=val, nosync, unlock]

sdxvolume -R -c class -v volume

sdxvolume -S -c class -v volume -s size
```

### DESCRIPTION

Use sdxvolume to perform operations on volume objects (excluding shadow volumes) specified by *volume*.

You must be superuser to use this command.

### PRIMARY OPTIONS

You can use either of the following options.

#### -F

##### **o**ffline

Stops one or more volumes specified by *volume*. If the *-v* option is omitted, all volumes within the *class* are taken offline. Offline volumes cannot be accessed.

If synchronization copying of *volume* is in process, it cannot be stopped. You can stop the synchronization copying using the `sdxcopy -C` command.

The stopped volume will be activated when the node is rebooted. (Except for when the volume is locked.) Attempting this operation while *volume* is in use results in an error.

#### -M

##### **M**ake

Creates a volume specified by *volume*, within the highest level group specified by *group*, or within a single disk specified by *disk*. The *size* indicates the number of blocks on *volume*, *class* indicates the class name associated with the *group* or *disk*.

The same group or disk can contain a maximum of 4 volumes with their physical slice attribute set to "on". A maximum of 224 volumes can be created in total, including the volumes with physical slice attribute set to "off". When *-a pslice=value* option is omitted, volumes with physical slice attribute set to "on", will be created. However, note that you cannot create a volume with physical slice attribute set to "on", if *group* is either stripe group, concatenation group, or a mirror group where its only directly-connected object is a lower level group. In such case, you must indicate the *-a pslice=off* option, and set the physical slice attribute to "off."

After volume creation is complete, the volumes are started on a node where the command was executed and become accessible through the following special files.

```
/dev/sfdsk/class/rdsk/volume
/dev/sfdsk/class/dsk/volume
```

If *group* is a mirror group, the system will automatically execute a synchronization copying upon returning from the `sdxvolume` command.

The features of volumes created when *group* is mirror group, stripe group and switch group are explained below.

**- When group is a mirror group**

To ensure data availability, GDS restricts the mirroring on a single piece of disk unit. In the case of mirror groups, a mirror volume that consists of mirror-multiplexing equal to the number of connected disks or lower level groups is created (maximum of eight). When only one disk or one lower level group is connected, the volume created within that mirror group will not be mirrored.

If the last block number of the volume that is created within a mirror group is larger than the available size of any of the spare disks registered with *class*, a warning message is sent to standard error output informing you that the hot spare feature is disabled.

**- When group is a stripe group**

In a stripe group, stripe volumes with columns equal to the number of connected disk or lower level groups are created. When only one disk or lower level group is connected, volume cannot be created.

**- When group is a switch group**

In a switch group, switch volumes with redundancy equivalent to the number of connected disks (a maximum of 2) are created. If only one disk is connected, an active disk switch cannot be performed.

If the active disk is not connected to a node where the command was executed, the volumes are not started. To use the created switch volumes, perform an active disk switch with the `sdxattr -G` command, or move to a node to which the active disk is connected, and then start the volumes with the `sdxvolume -N` command.

**-N**

**oNline**

Activates one or more volumes specified by *volume*,... If the `-v` option is omitted, all volumes within *class* are activated.

Activated volumes can be accessed.

If there is a slice in TEMP status on the *volume*, a warning message is sent to standard error output.

When *volume* is a mirror volume, the system will determine whether synchronization has been lost upon returning from the `sdxvolume` command, and automatically execute a synchronization copying as needed (except for when `-e nosync` is specified).

When *volume* is a switch volume, it cannot be started on nodes to which the active disk is not connected.

**-R**

**Remove**

Removes the volume specified by *volume* and releases the disk area used in the group or the single disk.

If the specified volume is activated, this command results in an error.



Note

Be aware that any data stored on *volume* will be lost.

**-S**

**reSize**

Expands the size of a volume specified by *volume* to *size* blocks. *class* indicates the name of a class to which *volume* belongs.

*volume* must be a volume that belongs to any one of:

- A single disk
- A mirror group that consists of only one disk
- A mirror group that consists of only one lower level group

The size can be expanded even when the *volume* is active.

The first block of *volume* is not changed. If any area of a volume other than *volume* exists in the area of *size* blocks after the first block of *volume*, it results in an error.

## SUB OPTIONS

SUB OPTIONS are as follows:

### **-a *attribute=value*[,*attribute=value*] (when using **-M**)**

Use this to set an *attribute* for the *volume*.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value. Always separate *attribute* and *value* with an equal (=) sign.

When indicating multiple attributes, specifiers should be combined using commas(,) as the delimiter.

You can indicate the following combination to *attribute* and *value*.

If multiple attributes are specified and an error results because of any part of them, the entire process is canceled.

### **jrm=on or jrm=off (default is on)**

Sets the volume's JRM mode.

If a group other than a mirror group is specified in the **-g** option, this command results in an error.

**on**

JRM is "on".

**off**

JRM is "off".

### **pslice=on or pslice=off (default is on)**

Sets the physical slice attribute value of volume.

When using **-g** option to indicate a group in which a physical slice cannot be created (i.e. stripe group, concatenation group, or mirror group where the only object directly connected to mirror group is lower level group), you cannot set this option to "on."

**on**

Physical slice attribute value is set to "on".

Among slices composing the volume, the slices within the single disk, the disks that are connected to the switch group or the disks that are directly connected to the mirror group are registered to the VTOC, and physical slices are created.

**off**

Physical slice attribute value is set to "off".

None of the slices consisting the volume is registered to the partition table, and physical slice will not be created. When physical slice attribute is set to "off", the slice cannot be detached.

### **-c *class***

The *class* indicates the class name to which the volume that is the target of the operation belongs, or the class name in which the volume is to be created.

### **-d *disk* (when using **-M**)**

The *disk* indicates the single disk name in which the single volume will be created.

### **-e allnodes (when using **-F,-N**)**

Stops or activates the volume on all nodes included in the scope of *class*. Stopped nodes are ignored.

When neither this option nor **-e node=node** [:*node*:...] option is specified, volume is stopped or started only on the self-node.

### **-e delay=*msec* (when using **-M,-N**)**

If synchronization is not maintained when creating or activating a mirror volume, synchronization copying will take place automatically (except for when **-e nosync** is specified). This option delays the issuing of the input/output request to the disk at the time of copying by milliseconds specified by *msec*, allowing adjustment for the effect on the application accessing the volume. The value is set to 0 by default. Values from 0 to 1000 may be specified for *msec*.

If mirror group is not specified with *group*, this option is ignored.

### **-e mode=*val* (when using **-N**)**

Specifies the access mode for one or more volumes that will be activated.

*val* indicates either of the following options.

**rw**

Sets access mode for read and write.

**ro**

Sets access mode for read only. Opening a read-only volume in write mode will result in an error.

Although volume will be activated in the access mode specified by *val*, the access mode attribute for



volume will remain unchanged. Access mode specified by *val* ("Current Access Mode") is valid only while the volume is activated, and will become invalid once the volume is stopped. When the volume is restarted, it will start in the mode set by access mode attribute ("Default Access Mode"), except for when the access mode is specified at the time of restart.

In order to start a volume that is already activated on the current node in a different access mode, you must first stop the volume.

**-e node=***node[:node,...]*(when using **-F,-N**)

Stops or activates the volume on one or more specified nodes.

Stopped nodes are ignored. You must specify the node identifier of the node to stop or activate the volume to *node*. If a node not included in the scope of *class* is specified, the volume is not stopped or activated on any node.

If this option and the **-e allnodes** are both omitted, the volume is stopped or activated only on the current node.

**-e nosync** (when using **-N**)

Disables automatic synchronization copying after activating a mirror volume.

If mirror group is not specified with *group*, this option is ignored.



**Note**

Volumes that are activated using this option will not be mirrored.

In order to configure a mirroring environment, you must perform synchronization copying with the `sdxcopy -B` command.

**-e unlock** (when using **-N**)

The volume will be activated regardless to whether or not it is locked. Lock mode will not be changed unless you change it with the `stxattr -V` command.

**-g group** (when using **-M**)

The *group* indicates the group name in which the volume will be created.

**-s size** (when using **-M**)

Specifies the size of the *volume* being created, in blocks (base 10).

One block is 512 bytes.

When *group* indicates stripe group, the size of *volume* created will be *size* rounded up to a common multiple of stripe width multiplied by stripe columns and cylinder size. In other cases, the size of *volume* created will be *size* rounded up to the integer multiple of cylinder size.

**-s size** (when using **-S**)

Specifies the number of blocks (decimal number) to which the size of the specified *volume* is expanded. One block is 512 bytes.

The size of the expanded *volume* will be the size rounded up to the integer multiple of the cylinder *size*.

**-v volume** (when using **-M,-R**)

The *volume* indicates the volume name that is the target of operation.

**-v volume,...** (when using **-F,-N**)

The *volume,...* indicates one or more volume names that is the target of the operation. To indicate multiple volumes, separate each volume name with a comma (,) as the delimiter.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.5 sdxslice - Slice operations

### SYNOPSIS

```
sdxslice -F -c class -s slice,...  
sdxslice -M -c class -d disk -v volume [-a attribute=value[,attribute=value]]  
sdxslice -N -c class -s slice,...  
sdxslice -R -c class {-d disk -v volume|-s slice } [-e delay=msec,waitsync]  
sdxslice -T -c class -s slice,...
```

### DESCRIPTION

Use sdxslice to perform operations on slice objects (excluding shadow slices). You must be superuser to use this command.

### PRIMARY OPTIONS

You can use one of the following options.

#### -F

##### **oF**line

Stops the slice or slices specified by *slice*,... Offline slices cannot be accessed. The *slice* indicates the mirror slice name detached from the mirror volume using -M option.

The *class* indicates the class name to which *slice* belongs.  
Offline slices will be activated upon reboot.  
Attempting this command while *slice* is in use results in an error.

#### -M

##### **M**ake

Temporarily detaches one of the mirror slices used in a copy of a mirror volume specified by *volume* with mirroring multiplicity of two and higher, which is a part of the disk specified by *disk*. The *class* indicates the class name to which *volume* belongs.

Only when the physical slice attribute value of *volume* is "on", you can detach the slice. When the physical slice attribute is set to "off", you must turn it to "on" using the sdxattr -V command before executing this command.

Once detached, a special file is placed on the system. The path name is given below.

```
/dev/sfdsk/class/[r]dsk/disk.volume
```

User will be able to access the slice with this special file. You can use this slice to create a data backup of volume.

Only the node that detached the slice can access the slice. Other nodes sharing the *class* cannot access it. If you need to access from other nodes, you can take over the access right with the -T option.

A slice can be detached even though the *volume* is active.

You must ensure the integrity of backup data at the file-system layer or database layer.

If you are handling the volume as the file-system, for instance, there will be situations where you must regain integrity using the fsck command.



##### Note

Be aware that as long as a slice is not attached using the -R option, the degree of mirror multiplexing stays reduced.

#### -N

##### **oN**line

Activates the slice or slices specified by *slice*,... Activated slices can be accessed.

The *slice* indicates the mirror slice name detached from the mirror volume using -M option.

The *class* indicates the class name to which *slice* belongs.

## **-R**

### **Remove**

Reassembles the slice as part of the volume, where the slice is specified by *slice* or combination of *disk* and *volume*.

The *disk* and *volume* combination or *slice* indicates the mirror slice name disconnected from the mirror volume using -M option.

The *class* indicates the class name to which the slice belongs.

After (when using -e waitsync option, before) returning from the sdxslice command, the slice is automatically reassembled with the volume. If the volume is active at this time, a synchronization copy is executed.

Attempting this command while the slice is in use results in an error.

## **-T**

### **Takeover**

Takes over the slice or slices specified by *slice* from another node. When the takeover is complete, the slice will stop on the original node and be activated on the current node, allowing operation to the slice on the current node.

When a slice is attached to the volume after executing this command, the entire block will be copied regardless to the setting of JRM mode.

The *slice* indicates the mirror slice name disconnected from the mirror volume using -M option.

The *class* indicates the class name to which *slice* belongs.

## **SUB OPTIONS**

SUB OPTIONS are as follows:

### **-a *attribute=value*[,*attribute=value*] (when using -M)**

Sets the *attribute* attribute of the slice to be *value*.

Both attribute values become invalid at the point when the slice is assembled with the volume.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign.

Specifiers should be combined using a comma (,) as the delimiter.

You can use the following *attribute* and *value* pairs in any combination.

If multiple attributes are specified and an error results because of any part of them, the entire process is canceled.

### **jrm=on or jrm=off (default is on)**

Sets the slice's JRM mode.

#### **on**

JRM is on.

#### **off**

JRM is off.

### **mode=rw or mode=ro (rw when omitted)**

Sets the slice access mode.

#### **rw**

Sets access mode for read and write.

#### **ro**

Sets access mode for read only. Opening a read-only slice in write mode will result in an error.

### **-c *class***

The *class* indicates the class name to which the slice belongs.

### **-d *disk* (when using -M,-R)**

The *disk* indicates the disk name to which the slice belongs, where slice is the target of the operation.

**-e delay=msec** (when using **-R**)

This option delays the issuing of the input/output request to the disk at the time of synchronization copying of the slice detached from volume, by milliseconds specified by *msec*.

Always separate delay and *msec* with an equal (=) sign.

This option allows you to adjust the influence on the application accessing the volume.

The value is set to 0 by default. Values from 0 to 1000 may be specified for *msec*.

**-e waitsync** (when using **-R**)

When synchronization copying is executed, returns the command after the copying process is complete.

**-s slice** (when using **-R**)

The *slice* indicates slice that is the target of operation. Slice name should be specified in *disk.volume* format.

**-s slice,...** (when using **-F,-N,**and **-T**)

The *slice* indicates one or more slice names that is the target of the operation. To indicate multiple slices, separate each slice name with a comma (,) as the delimiter.

Slice name should be specified in *disk.volume* format.

**-v volume** (when using **-M,-R**)

Specifies the name of volume comprising the slice that is the target of the operation.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.  
Otherwise, a non-zero value is returned.

## D.6 **sdxinfo** - Display object configuration and status information

### SYNOPSIS

**sdxinfo** [ **-ACDGSV** ] [ **-c** *class* ] [ **-o** *object* ] [ **-e** *long* ]

### DESCRIPTION

Use **sdxinfo** to display configuration and status information of the objects which the current node shares. The **sdxinfo** command allows you to view configurations and status information of SDX objects, proxy objects and shadow objects.

### PRIMARY OPTIONS

Primary options indicate the type of objects to display. If nothing is specified, only information on the pertinent object will be displayed. Object types can be any combination of the following.

**-A**

**All**

Displays all specified objects, and information on all related objects.

This is equivalent to **-CDGSV**.

If any other options are combined with this, they are ignored.

**-C**

**Class**

Displays all specified objects, and class information of the related objects.

**-D**

**Disk**

Displays all specified objects, and disk information of the related objects.

**-G**

**Group**

Displays all specified objects, and group information of the related objects.

**-S**

**Slice**

Displays all specified objects, and slice information of the related objects.

**-V**

**Volume**

Displays all specified objects, and volume information of the related objects.

### SUB OPTIONS

SUB OPTIONS are used to specify the names of objects to display.

If nothing is specified, the command is interpreted as if all objects at the current node had been specified.

**-c** *class*

The *class* indicates the class name whose information will be displayed.

If this option is omitted, this command is interpreted as if all classes had been specified.

In case of combining this option with the **-o** option, the objects related to the specified *object* within the *class* is displayed.

**-e** *long*

Displays detailed object information.

**-o *object***

The *object* indicates the object name (class name, disk name, group name or volume name) whose information will be displayed.

If this option is omitted, this command is interpreted as if all object names had been specified.

In case of combining this option with the -c option, the objects related to the specified *object* within the *class* is displayed.

## DISPLAYED INFORMATION

Information displayed in response to the `sdxinfo` command can be interpreted as follows.

### Class information:

**OBJ**

Display class as an object classification.

**NAME**

Display class name.

**TYPE**

The following value is displayed.

**shared**

shared class

**SCOPE**

Node identifiers are separated by colons ( : ) as the scope attribute value.

**HS**

When -e long option is used, this displays hot spare operation, which can be either of the following.

**on**

Enables the hot spare.

**off**

Disables the hot spare.

Be aware that spare disk automatic connection is prevented.

For a shadow class it always displays "on", but the hot spare feature is practically invalid since a shadow class cannot include a spare disk.

**SPARE**

Displays the number of spare disks that are not connected to the group.

**SHADOW**

When the -e long option is specified, one of the following is displayed as the class description.

**0**

Class created with the `sdxdisk -M` command.

**1**

Shadow class created with the `sdxshadowdisk -M` command.

**HSMODE**

Displays one of the following values to indicate the spare disk selection method for automatic connection by hot spare when the -e long option is specified. For a shadow class, an asterisk (\*) is displayed.

**exbox**

External mode.

If an I/O error occurs in a disk of a disk array unit, this method selects a spare disk that belongs to a different disk case from that of the failed disk. If an I/O error occurs in a disk irrelevant to a disk array unit (such as an internal disk), it selects a spare disk that is connected to a different controller from that of the failed disk. When no applicable unconnected spare disk is found there, a spare disk that belongs to the same disk case or is connected to the same controller as that of

the disk with the I/O error is selected.

### **bybox**

Internal mode.

If an I/O error occurs in a disk of a disk array unit, this method selects a spare disk that belongs to the same disk case as that of the failed disk. If an I/O error occurs in a disk irrelevant to a disk array unit (such as an internal disk), it selects a spare disk that is connected to the same controller as that of the failed disk. When no applicable unconnected spare disk is found there, spare disk automatic connection is prevented.

## **Disk information:**

### **OBJ**

Displays disk as an object classification.

### **NAME**

Displays disk name.

### **TYPE**

The type attribute value can be any of the following:

#### **mirror**

Mirror. It is connected to a mirror group.

#### **stripe**

Stripe. It is connected to a stripe group.

#### **concat**

Concatenation. It is connected to a concatenation group.

#### **switch**

Switch. It is connected to a switch group.

#### **single**

Single. Single volumes can be created on it.

#### **spare**

Spare. Also the same when the spare disk is connected to the group.

#### **undef**

Undefined. Its usage is not determined yet.

### **CLASS**

Displays the class name to which the disk belongs.

### **GROUP**

Displays the group name to which the disk is connected. If the disk is not connected to any group, an asterisk (\*) is displayed.

### **DEVNAM**

Displays the physical disk name in one of the following formats. If the disk is not connected to the local node, an asterisk (\*) is displayed.

**sdX** (for normal hard disks)

**emcpowerX** (for emcpower disks)

X indicates the device identifier

### **DEVBLKS**

Displays the size of the physical disk. The size is given in blocks (sectors.) If the disk is not connected to the local node, 0 (zero) is displayed.

### **FREEBLKS**

When **-e** long option is used, displays the number of free allocatable blocks (sectors) for a new volume. If the disk is not a single disk, an asterisk (\*) is displayed.

### **DEVCONNECT**

Displays a list of node identifiers of the nodes to which the physical disk is connected, separated using colons (:) as delimiters. If disk resources are not created yet, an asterisk (\*) is displayed.

## STATUS

Disk status can be any of the following:

### ENABLE

Available for work.

### DISABLE

Not available for work.

### SWAP

Not available for work, but available for disk exchanging.

## E

When -e long option is used, displays error status of disk, which can be any of the following.

### 0

I/O error has not occurred on both the current node and the shared node.

### 1

I/O error has occurred on either the current node or the shared node.

## Group information:

### OBJ

Display group as an object classification.

### NAME

Display group name.

### CLASS

Displays the class name to which the group belongs.

### DISKS

Displays the name of disks or lower level groups that are connected to the group, separated using colons ( : ) as delimiters. In case of a stripe group, names are listed in the order they are striped. Likewise, if it is a concatenation group, names are listed in the order they are concatenated.

### BLKS

Displays size of the group, which is the total of available size of group (available capacity as volumes) plus one private slice size. Size is given in blocks (sectors).

### FREEBLKS

The number of free allocatable blocks for a new volume.  
If the group is a lower level group, an asterisk (\*) is displayed.

### SPARE

Displays the number of spare disks that can be connected to the group. Unless it is a mirror group, an asterisk (\*) is displayed.

### MASTER

When -e long option is used, it displays the group name of master group. When it is not a proxy group, an asterisk (\*) is displayed.

## TYPE

When -e long option is used, displays type attribute value, which can be any of the following.

### mirror

Mirror group.

### stripe

Stripe group.

### concat

Concatenation group.



**switch**

Switch group

**WIDTH**

When `-e` long option is used, displays stripe width in blocks (sectors). If the group is not a stripe group, an asterisk (\*) is displayed.

**ACTDISK**

Displays the disk name of the active disk when the `-e` long option is specified. Unless it is a switch group, an asterisk (\*) is displayed.

**Volume information:****OBJ**

Displays volume as an object classification.

**NAME**

Displays the name of the volume.

If it is an area that cannot be allocated (area for private slice) or that can be allocated but have not (unallocated area), an asterisk (\*) is displayed.

**TYPE**

When `-e` long option is used, displays type attribute, which can be any of the following.

**mirror**

Mirror. It belongs to a mirror group.

**stripe**

Stripe. It belongs to a stripe group.

**concat**

Concatenation. It belongs to a concatenation group.

**switch**

Switch. It belongs to a switch group.

**single**

Single. It belongs to a single disk.

**CLASS**

Displays the class name to which the volume belongs.

**GROUP**

Displays the highest level group name to which the volume belongs. When it belongs to a single disk, an asterisk (\*) is displayed.

**DISK**

When `-e` long option is used, displays the name of the single disk to which the volume belongs. When it belongs to a group, an asterisk (\*) is displayed.

**MASTER**

When `-e` long option is used, it displays the volume name of master volume. When it is not a proxy volume, an asterisk (\*) is displayed.

**PROXY**

When `-e` long option is used, it displays the proxy volume status in one of the two ways as given below. When it is not a proxy volume, an asterisk (\*) is displayed.

**Join**

The volume is being joined to a master volume.

**Part**

The volume is being parted from a master volume.

## SKIP

Displays the skip-resynchronization mode setting, which can be either of the following. If the volume is neither a mirror volume nor a single volume, an asterisk (\*) is displayed.

**on**

Skip resynchronization.

**off**

Execute resynchronization.



Note

Note that the interface for setting or changing this option is not available.

## JRM

Displays the just resynchronization mode setting, which can be either on or off. If the volume is neither a mirror volume nor a single volume, an asterisk (\*) is displayed.

**on**

JRM is on.

**off**

JRM is off.

## MODE

When -e long option is used, it displays the access mode attribute value (default access mode) for the current node, which can be either of the following. If it is either an area for private slice or an unallocated area, an asterisk (\*) is displayed.

**rw**

Read and write mode.

**ro**

Read only mode.

## CMODE

When -e long option is used, it displays the present access mode of the activated volume from the current node. If the volume is not activated, an asterisk (\*) is displayed.

**rw**

Read and write mode.

**ro**

Read only mode.

## LOCK

When -e long option is used, displays the lock mode of current node, which can be either of the following. If it is either a private area or an unallocated area, an asterisk (\*) is displayed.

**on**

The volume is locked from activating thereafter.

**off**

The volume is not locked from activating thereafter.

## 1STBLK

Displays the block (sector) number of the first block.

The block number is the logical block number, which is the offset in the group to which the volume belongs, and not the physical block number indicating the offset on the physical disk. However, when the volume belongs to a single disk, the block number will match the physical block number on the single disk. Also, when it belongs to a mirror group to which a disk is directly connected or a switch group, the block number will match the physical block number on the disk.

## LASTBLK

Displays the block (sector) number of the last block.

The block number is the logical block number, which is the offset in the group to which the volume belongs, and not the physical block number indicating the offset on the physical disk. However, when the volume belongs to a single disk, the block number will match the physical block number on the single disk. Also, when it belongs to a mirror group to which a disk is directly connected or a switch group, the block number will match the physical block number on the disk.

**BLOCKS**

Displays the size in blocks (sectors).

**STATUS**

Displays the volume status of the current node, which can be any of the following.

**ACTIVE**

Ready for work.

**STOP**

Stopped.

**INVALID**

Stopped, and cannot be activated due to problem with data.

**FREE**

Not yet allocated as a volume.

**PRIVATE**

An area reserved for GDS control, so cannot be allocated as a volume.

**PSLICE**

When -e long option is used, displays the physical slice attribute value, which can be either of the following. If it is either a private area or an unallocated area, an asterisk (\*) is displayed.

**on**

Physical slice attribute of the volume is set to "on." Among slices comprising volumes, slices on single disks, on disks connected to switch groups and on disks directly connected to mirror groups are registered with partition table and have physical slices. If a lower level group is the only object directly connected to mirror group, the volume will not have a physical slice, regardless to this attribute being set to "on." Also, when the volume belongs to either a stripe group or a concatenation group, this attribute value will never be set to "on."

**off**

Physical slice attribute of the volume is set to "off." The volume has no physical slices, and none of the slices in the volume is registered to the partition table.

For a shadow volume it always displays off no matter whether the shadow slice is registered with partition table.

**SNUM**

When the -e long option is specified, the slice number of the slice configuring the volume is displayed. If the physical slice attribute is off or no physical slice configures the volume, an asterisk (\*) is displayed.

**PJRM**

When the -e long option is specified, either of the following values is displayed to indicate the just resynchronization mechanism mode on the proxy volume.

If it is not a proxy volume parted from the master volume with the sdxproxy Part command, an asterisk (\*) is displayed.

**on**

JRM is "on".

**off**

JRM is "off".

**Slice information:****OBJ**

Display slice as an object classification.

**NAME**

When -e long option is used, this display the name of slice.

When the slice is not a mirror slice that is temporarily detached from the mirror volume using the sdxslice -M command, an asterisk (\*) is displayed.

**CLASS**

Displays the class name to which the slice belongs.

## **GROUP**

Displays the highest level group name to which the slice belongs.  
If it is a single slice, an asterisk (\*) is displayed.

## **DISK**

Displays the name of the disk or the lower level group (i.e. the group to which this slice belongs, among the groups that are directly connected to the relevant highest level group) to which the slice belongs. If the highest level group is a switch group, the disk name of the active disk is displayed. If the highest level group is a stripe group or a concatenation group, an asterisk (\*) is displayed.

## **VOLUME**

Displays the volume name to which the slice belongs.

## **JRM**

When -e long option is used, displays the just resynchronization mode setting, which can be either on or off.

When the slice is not a mirror slice that is temporarily detached from the mirror volume using the sdxslice -M command, an asterisk (\*) is displayed.

### **on**

JRM is on.

### **off**

JRM is off.

## **MODE**

When -e long option is used, displays the access mode, which can be either of the following.

When the slice is not a mirror slice that is temporarily detached from the mirror volume using the sdxslice -M command, an asterisk (\*) is displayed.

### **rw**

Read and write mode.

### **ro**

Read only mode.

## **STATUS**

Displays the slice status on the current node, which can be any of the following.

### **ACTIVE**

Ready for work.

### **STOP**

Stopped.

### **INVALID**

Due to a problem with data, temporarily detached from the volume.

### **COPY**

Specifies a copy is underway, to maintain data uniformity.

### **TEMP**

Temporarily detached from volume. Slice is operating in isolation.

### **TEMP-STOP**

Temporarily detached from volume. Slice is stopping in isolation.

### **NOUSE**

Stopped, with no operations possible.

## **COPY**

When -e long option is used, this displays either of the following copying process statuses.  
When not in COPY status, an asterisk (\*) is displayed.

### **run**

Copying is underway.

### **bg**

Copying is in process in the background, but you can access valid data.

### **intr**

Copying has been interrupted. Executing the sdxcopy -I command interrupts the copy.

### **wait**

Since many copying processes are in progress, it is waiting to be scheduled.

**CURBLKS**

When -e long option is used, this displays the number of blocks (sectors) that has been copied.  
When CURBLKS and the later described COPYBLKS match, all copying has been completed.  
When the slice is not in COPY status an asterisk (\*) is displayed.

**COPYBLKS**

When -e long option is used, this displays the number of blocks (sectors) that needs to be copied.  
Usually this is the same size as the volume size it is registered with, but when just resynchronization copying is in process, the number of blocks that actually needs to be copied will be displayed.  
When the slice is not in COPY status, an asterisk (\*) is displayed.

**DLY**

When -e long option is used, this displays the copy delay time in milliseconds.  
When not in COPY status, an asterisk (\*) is displayed.

**CPTYPE**

When -e long option is used, one of the following values is displayed as copy function which is used for copying process between the master and the proxy. When copying is not in process, or if the volume to which the slice belongs is not a target volume of copying process between a master volume and a proxy volume, an asterisk (\*) is displayed.

**soft**

Copying is in process using the soft copy function provided by the GDS sfdsk driver.

**EC**

Copying is in process using the Equivalent Copy function.

**OPC**

Copying is in process using the One Point Copy function. When the master and the proxy have been joined, and the copy source volume is active, the soft copy function may be involved in part of the copying process.

**REC**

Copying is in process using the Remote Equivalent Copy function.

**ROPC**

Copying is in process using the Remote One Point Copy function. When the master and the proxy have been joined, and the copy source volume is active, the soft copy function may be involved in part of the copying process.

**CPSOURCE**

When -e long option is used, the volume name of source proxy volume which is used to restore data is displayed. When the volume to which the slice belongs is not a target master volume of restoring process, an asterisk (\*) is displayed.

**FUNC**

Displays one of the following values to indicate the session type of the disk unit's copy function when the -e long option is specified. If there is no session, an asterisk (\*) is displayed.

**EC**

Source or target of the Equivalent Copy session.

**OPC**

Source or target of the One Point Copy session.

**REC**

Source or target of the Remote Equivalent Copy session.

**ROPC**

Source or target of the Remote One Point Copy session.

**CPSTAT**

Displays one of the following values to indicate the session status of the disk unit's copy function when the -e long option is specified. If there is no session, an asterisk (\*) is displayed.

**equiv**

Synchronized.

**copy**

In process of copying.

**suspend**

EC or REC session suspended.

**error**

Suspended due to an error.

**halt**

Hardware suspended.

**PARTNER**

When the `-e` long is specified, displays a destination slice name if the slice is the source in the session of the disk unit's copy function, or a source slice name if it is the destination. If there is no session, an asterisk (\*) is displayed.

**RETURNED VALUE**

Upon successful completion, a value of 0 is returned.  
Otherwise, a non-zero value is returned.

**USAGE EXAMPLES**

Displays all the objects within the current node.

```
sdxinfo
```

Displays information on all the disks registered with the class called "Class1."

```
sdxinfo -A -c Class1
```

Use this to check whether an object called "foo" is currently being used.

```
sdxinfo -o foo
```

**CAUTION**

Additional information may be displayed in accordance with new functions provided.

## D.7 sdxattr - Set objects attributes

### SYNOPSIS

```
sdxattr -C -c class -a attribute=value[,attribute=value,...]
sdxattr -D -c class -d disk -a attribute=value[,attribute=value]
sdxattr -G -c class -g group -a attribute=value[,attribute=value]
sdxattr -S -c class -s slice -a attribute=value[,attribute=value]
sdxattr -V -c class -v volume -a attribute=value[,attribute=value,...]
```

### DESCRIPTION

Use sdxattr to change attribute values of objects (excluding shadow objects) on the current node. You must be superuser to use this command.

### PRIMARY OPTIONS

Primary options are used to specify the category of the object whose attributes are to be set.

**-C**

#### **Class**

Set the attributes of the class specified by *class*.

**-D**

#### **Disk**

Set the attributes of the disk specified by *disk*.

The *class* indicates the class name with which *disk* is registered.

If *disk* is connected to a group, or if there is a volume within *disk*, it will result in an error, and the change in attribute will not be executed.

**-G**

#### **Group**

Set the attributes of the group specified by *group*.

The *class* indicates the class name to which *group* belongs.

When *group* is connected to another group, this option will result in an error and you cannot change the attribute. Also, this option will result in an error if there are one or more activated volumes within the group. Stop all volumes before executing this command.

**-S**

#### **Slice**

Set the attributes of the slice specified by *slice*.

The *class* indicates the class name to which the *slice* belongs.

**-V**

#### **Volume**

Set the attributes of the volume specified by *volume*.

The *class* indicates the class name to which the *volume* belongs.

### SUB OPTIONS

SUB OPTIONS are as follows:

**-a *attribute=value[,attribute=value,...]*** (when using **-C**)

Sets the *attribute* attribute of the *class* to be *value*.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign.

Specifiers should be combined using a comma (,) as the delimiter.

You can use the following *attribute* and *value* pairs in any combination.

When attempting to set multiple attributes, if any of them result in an error, no attributes are set.

**scope=node:node:...**

For a "shared" type class, changes the nodes which share the class.

When there is an activated volume within *class*, you can add new nodes, but you cannot remove a node that has already been included in the scope. In order to remove a node, you must execute the command after first stopping all volumes.

If the *node* settings are not all complete, this will result in an error.

The *node* indicates a node identifier that is defined by PRIMECLUSTER.

The *scope* can indicate a maximum of 4 nodes.

Changing the scope of a class fails if the class is a shared class that includes a disk of which physical scope is 1 node, or that includes a switch group.

**hs=on or hs=off**

It sets the operation of the hot spare.

You can make changes regardless to whether there is an activated volume within *class*.

**on**

Enables the hot spare.

**off**

Disables the hot spare. Spare disk automatic connection is restricted.

**hsmode=exbox or hsmode=bybox**

Changes the spare disk selection mode for automatic connection by hot spare.

This operation is available regardless whether or not there are active volumes within *class*.

**exbox**

Changes the spare disk selection method to the external mode.

If an I/O error occurs in a disk of a disk array unit, this method selects a spare disk that belongs to a different disk case from that of the failed disk. If an I/O error occurs in a disk irrelevant to a disk array unit (such as an internal disk), it selects a spare disk that is connected to a different controller from that of the failed disk. When no applicable unconnected spare disk is found there, a spare disk that belongs to the same disk case or is connected to the same controller as that of the disk with the I/O error is selected.

**bybox**

Changes the spare disk selection mode to the internal mode.

If an I/O error occurs in a disk of a disk array unit, this method selects a spare disk that belongs to the same disk case as that of the failed disk. If an I/O error occurs in a disk irrelevant to a disk array unit (such as an internal disk), it selects a spare disk that is connected to the same controller as that of the failed disk. When no applicable unconnected spare disk is found there, spare disk automatic connection is restrained.

**name=classname**

Sets the class name to *classname*.

When there is an activated volume within *class*, it results in an error. First, you must stop all volumes.

When changing a class name through this operation, the paths of special files for volumes are also changed, so you must update the files in which the paths are described, such as /etc/fstab.

***/dev/sfdsk/classname/[r]dsk/volume\_name***

**-a attribute=value[,attribute=value] (when using -D)**

Sets the *attribute* attribute of the *disk* to be *value*.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign.

Specifiers should be combined using a comma (,) as the delimiter.

You can use the following *attribute* and *value* pairs in any combination.

When attempting to set multiple attributes, if any of them result in an error, no attributes are set.

**type=single, type=spare or type=undef**

Sets the SDX disk type attribute. If *disk* is not connected to part of nodes included in the scope of *class*, or if a switch group exists in *class*, changing the type attribute of *disk* fails.

**single**

Sets the type attribute to "single."

Single volume may be created within *disk* thereafter.



**spare**

Sets the type attribute to "spare."

The disk will be used as a spare disk thereafter.

When the available size of *disk* is smaller than the available size of the smallest mirror group within *class*, a warning message notifying that the hot spare function may not be available will be sent to standard error output.

**undef**

Sets the type attribute to "undef."

Hereinafter, this disk will be regarded as an undefined disk, which use is not yet determined.

**name=diskname**

Sets the name of a disk to *diskname*.

**-a attribute=value[,attribute=value] (when using -G)**

Sets the *attribute* attribute of the *group* to be *value*.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign. To set multiple attributes, specify sets of these specifiers in comma-delimited format.

Specify any of the following sets into *attribute* and *value*.

When multiple attributes are specified, the entire process is canceled in the event of an error in part of the processes.

**name=groupname**

Sets the name of the group to *groupname*.

**actdisk=disk**

Changes the active disk of the switch group specified by *group* to *disk*.

**-a attribute=value (when using -S)**

Sets the *attribute* attribute of the detached *slice* to be *value*.

Both attribute values become invalid at the point when the slice is assembled with the volume using the -R option.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign.

Specifiers should be combined using a comma (,) as the delimiter.

You can use the following *attribute* and *value* pairs in any combination.

When attempting to set multiple attributes, if any of them result in an error, no attributes are set.

**jrm=off**

Turns the slice's just resynchronization mechanism mode to "off".

It could be turned "off" regardless to the slice status.

To set the jrm to "on," attach the slice to the volume and then detach it again.

**mode=rw or mode=ro**

Changes the access mode of current node for slice.

It will result in an error when *slice* is activated. Execute after stopping it.

**rw**

Sets access mode for read and write.

**ro**

Sets access mode for read only.

Opening a read-only volume in write mode will result in an error.

**-a attribute=value[,attribute=value,...] (when using -V)**

Sets the *attribute* attribute of the *volume* to be *value*.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value.

Always separate *attribute* and *value* with an equal (=) sign.

Specifiers should be combined using comma (,) as the delimiter.

You can use the following *attribute* and *value* pairs in any combination.  
When attempting to set multiple attributes, if any of them result in an error, no attributes are set.

### **jrm=on or jrm=off**

Turn the JRM mode "on" or "off".

You can make changes regardless to whether *volume* is activated or not.

If *volume* belongs to a group other than a mirror group, this command results in an error.

**on**

JRM is "on".

**off**

JRM is "off".

### **lock=on or lock=off**

Changes the lock mode of current node for *volume*.

You can make changes regardless to whether *volume* is activated or not.

**on**

The volume is locked from activating thereafter.

**off**

The volume is not locked from activating thereafter.

### **mode=rw or mode=ro**

Changes the access mode of current node for *volume*.

When *volume* is activated, it results in an error. First, you must stop the *volume*.

**rw**

Sets access mode for read and write.

**ro**

Sets access mode for read only.

Opening a read-only volume in write mode will result in an error.

### **name=volumename**

Sets the name of the volume to *volumename*.

It will result in an error when *volume* is activated. Execute after stopping *volume*.

### **pjrm=off**

The parted proxy volume's just resynchronization mechanism mode for rejoining or restoring is turned "off".

This can be changed regardless of whether volume is active or inactive.

This attribute value becomes invalid as volume is rejoined to the master volume with the Rejoin or RejoinRestore option of the *sdxproxy* command.

The value cannot be set to "on". To turn "on", the volume must be rejoined to the master volume once and then be parted again.

### **pslice=on or pslice=off**

Turns the physical slice attribute value to be "on" or "off", respectively.

If *volume* is activated, to avoid an error, stop *volume* before indicating this option. This option will also result in an error if there is a detached slice within *volume*. In such case, attach the slice before indicating this option.

**on**

The physical slice attribute value of volume is set to be "on".

Among the slices consisting *volume*, any slice on a single disk, and any slices on disks that are directly connected to a mirror group will be registered to the partition table.

You cannot change this option to "on" when *volume* belongs to a group that cannot create a physical slice (stripe group, concatenation group, or a mirror group whose only directly-connecting group is a lower level group), or when there is a maximum number (four) of volumes with its physical slice attribute set to "on" within the same group or single disk.

**off**

The physical slice attribute value of volume is set to be "off".

**-c *class***

The *class* indicates the class name to which the object belongs, or is to be changed, where object is the target of the change.

**-d *disk***

The *disk* indicates the disk name that is the target of the change.

**-g *group***

The *group* indicates the group name that is the target of the change.

**-s *slice***

The *slice* indicates the slice name that is the target of the change.  
Slice name should be specified in *disk.volume* format.

**-v *volume***

The *volume* indicates the volume name that is the target of the change.

## **RETURNED VALUE**

Upon successful completion, a value of 0 is returned.  
Otherwise, a non-zero value is returned.

## D.8 sdxswap - Swap disk

### SYNOPSIS

```
sdxswap -I -c class -d disk [ -e delay=msec,nowaitsync ]  
sdxswap -O -c class -d disk
```

### DESCRIPTION

Use sdxswap to make a disk (excluding a shadow disk) registered with GDS exchangeable and to restore the disk after swapping.

You must be superuser to use this command.

This command is mostly used when swapping crashed disks.

### PRIMARY OPTIONS

You can use either of the following options.

**-I**

#### **swapIn**

Returns the disk (specified by *disk*) to a usable state, and restores its original status and configuration. You must execute the command after a crashed disk has been swapped.

The *disk* indicates the disk name that is now available for swapping using the -O option. The *class* indicates the class name with which the *disk* is registered.

The physical disk size of *disk* must be equal to or larger than the original physical disk size.

When the highest level group of *disk* is a mirror group, slice configuration or volume contents is copied as needed, and the sdxswap command returns the control once the copying is complete (when using -e nowaitsync option, before the copying process).

If a spare disk is substituted in place of *disk*, then once the redundancy of all related volumes has been restored, the spare disk is disconnected.

**-O**

#### **swapOut**

Detaches all slices within the disk (specified by *disk*), and sets the disk to exchangeable status. This command must be executed before swapping out a crashed disk.

The *class* indicates the class name with which the *disk* is registered.

If *disk* is already nonusable, the status is rechecked and a warning message is sent to standard error output.

The following details explain unexchangeable and exchangeable conditions when *disk* is not connected to a group and when the highest level group of *disk* is a mirror group, a stripe group, a concatenation group or a switch group.

#### **- When *disk* is not connected to a group**

When volumes exist in *disk*, the *disk* cannot be made exchangeable.

#### **- When the highest level group of *disk* is a mirror group**

When volumes exist in the highest level mirror group of *disk* and detaching slices within the *disk* can change the volume configurations and statuses, the *disk* cannot be made exchangeable.

For example, if there are volumes in the highest level mirror group of *disk*, and if only *disk* is connected to that group, detaching slices within the *disk* will change the configurations and statuses of the volumes. Therefore, the *disk* cannot be made exchangeable.

#### **- When the highest level group of *disk* is a stripe group**

When the highest level group of *disk* is a stripe group, the *disk* cannot be made exchangeable by

detaching slices within the *disk*.

**- When the highest level group of *disk* is a concatenation group**

When the highest level group of *disk* is a concatenation group, the *disk* cannot be made exchangeable by detaching slices within the *disk*.

**- When the highest level group of *disk* is a switch group**

When *disk* is the inactive disk, the *disk* can be made exchangeable regardless whether or not there are volumes.

When *disk* is the active disk, the *disk* can be made exchangeable by detaching slices within the *disk* only if the switch group includes no switch volume or connected inactive disk.

When the switch group includes volumes, remove those volumes in order to make the active disk exchangeable. When it includes the connected inactive disk, perform an active disk switch with the `sdxattr -G` command and then make the former active disk exchangeable.

## SUB OPTIONS

SUB OPTIONS are as follows:

**-c *class***

The *class* indicates the class name to which *disk* belongs, where *disk* is the target of the operation.

**-d *disk***

The *disk* indicates the disk name that is the target of the operation.

**-e *delay=msec* (when using -l)**

When restoring the disk, data contained in the volume is copied as needed.

This option delays the issuing of the input/output request to the disk at the time of copying, by milliseconds specified by *msec*.

This option allows you to adjust the influence on the application accessing the volume.

The value is set to 0 by default.

Values from 0 to 1000 may be specified for *msec*.

**-e *nowaitsync* (when using -l)**

Returns the command before the copying is complete.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.9 sdxfix - Restore a failed object

### SYNOPSIS

```
sdxfix -C -c class
sdxfix -D -c class -d disk [-e online]
sdxfix -V -c class { -g group | -d disk } -v volume
```

### DESCRIPTION

Use sdxfix to restore failed objects (excluding shadow objects). In such a case, you need to restore integrity from backup data or check by executing the "fsck" command. The "sdxfix" command can be executed with superuser access privileges only.

### PRIMARY OPTIONS

You can use one of the following options.

#### -C

##### Class

Restores closed *class* to a normal status on the local node.

This command can restore *class* when the *class* includes a configuration database normally accessible and:

- less than 3 disks in ENABLE status and 1 or more disks normally accessible
- three to 5 disks in ENABLE status and 2 or more disks normally accessible
- six or more disks in ENABLE status and 3 or more disks normally accessible

After restoration is complete, objects within the *class* are restored to the previous status. However, the status of volumes that were in ACTIVE status before the class closure will be STOP.

#### -D

##### Disk

Restores the state of disk *disk* that detected an I/O error.

Reads all *disk* area, and if there's no problem, then clears the error status.

If there's a volume in the highest-level group to which *disk* belongs, or *disk* has a single volume, the volume should be stopped or inactive (STOP or INVALID) on all nodes (except when -e online is specified).

When *disk* is connected to a switch group, restoration fails. To clear an I/O error in a disk connected to a switch group, use the sdxswap -O command to make the disk exchangeable and then use the sdxswap -I command to make the disk useable.

#### -V

##### Volume

Restores a slice with invalid data (INVALID) or a not-ready slice (NOUSE) specified by a set of *disk* and *volume* or by a set of *group* and *volume* to the STOP status to restore the *volume* with invalid data (INVALID) to the STOP status.

The *volume* should be stopped or inactive (STOP or INVALID) on all nodes. The slice state specified in the combination of *disk* and *volume*, or *group* and *volume* should be INVALID or NOUSE.

Reads the entire slice specified in the combination of *disk* and *volume*, or *group* and *volume*, and if there's no problem, changes the state to STOP, and then changes the state of a stopped slice to INVALID.

### SUB OPTIONS

You can use the following sub-options.

#### -c *class*

Specify a name of the class to which the object belongs.

**-d *disk*** (when using **-D**)  
Specify a name of the disk.

**-d *disk*** (when using **-V**)  
When *volume* is a mirror volume, specify a name of the disk that is connected to the mirror group to which the *volume* belongs. This disk should have the INVALID mirror slice that needs to be restored to STOP.

Specify a single disk name when *volume* is a single volume.

When *volume* is a switch volume, specify a disk name of the active disk connected to a switch group that includes the *volume* into *disk*. Do not set an inactive disk name to *disk*.

**-e *online*** (when using **-D**)  
Restores the object even when the highest level group to which *disk* belongs or the single disk specified by *disk* includes an active volume.

**-g *group*** (when using **-V**)  
When *volume* is a mirror volume, specify a name of lower-level group that is connected to the mirror group to which the *volume* belongs. This group should have the INVALID mirror slice that needs to be restored to STOP.

Specify a name of the highest-level stripe group when *volume* is a stripe volume.

Specify a name of the highest-level concatenation group when *volume* belongs to the highest-level concatenation group.

**-v *volume*** (when using **-V**)  
Specify a name of the volume.

## **RETURNED VALUE**

When it is normally terminated, "0" is returned. Otherwise, a non-zero value is returned.

## D.10 sdxcopy - Synchronization copying operation

### SYNOPSIS

```
sdxcopy -B -c class -v volume,... [ -e delay=msec,nowaitsync ]  
sdxcopy -C -c class -v volume,...  
sdxcopy -I -c class -v volume,...  
sdxcopy -P -c class -v volume,... -e delay=msec
```

### DESCRIPTION

Use sdxcopy to access to synchronization copying for volume objects (excluding shadow volumes) specified by *volume*.

You must be superuser to use this command.

### PRIMARY OPTIONS

You can use one of the following options.

#### -B

##### **Begin**

Attaches slices that have been detached from mirror volumes specified by *volume*,... and executes synchronization copying. The command returns control after the synchronization copying is complete (before the copying process starts when using the -e nowaitsync option). *class* indicates the class name to which the *volume* belongs.

Slices on the *volume* that have a status of INVALID are attached, and then a synchronization copying is executed.

Slices with a status of TEMP\* or NOUSE are not attached.

If there are any slices currently involved in a synchronization copy on the *volume*, this command will terminate with an error.

This command resumes copying from the point where it was interrupted with the -I option.

Synchronization copying will be executed while the volume is activated or stopped.

#### -C

##### **Cancel**

Cancels synchronization copying in process or interrupted on the mirror volume or mirror volumes specified by *volume*,... After the cancel is completed, the system returns from the command.

The *class* indicates the class name to which *volume* belongs.

-B option executes the copying process again.

#### -I

##### **Interrupt**

Interrupts the synchronization copying currently underway on a mirror volume or mirror volumes specified by *volume*,...

The command returns after the interrupt is completed.

The *class* indicates the class name to which *volume* belongs.

-B option executes the copying process from the point where it was interrupted.

#### -P

##### **Parameter**

Changes parameter related to the synchronization copying in process or interrupted on the mirror volume or mirror volumes specified by *volume*,...

The *class* indicates the class name to which *volume* belongs.

Copying in process will resume after the parameter is changed.

The current status of synchronization copying that is either in process or interrupted can be checked by executing the sdxinfo -S command.



## SUB OPTIONS

SUB OPTIONS are as follows:

**-c *class***

The *class* indicates the class name to which *volume* belongs.

**-e *delay=msec*** (when using **-B,-P**)

Delays the issuing of the input/output request to the disk at the time of copying by milliseconds specified by *msec*. This option allows you to adjust the influence on the application accessing the volume.

The value is set to 0 by default.

If the copying process is either completed or canceled, the delay time will return to default (0).

Values from 0 to 1000 may be specified for *msec*.

**-e *nowaitsync*** (when using **-B**)

Returns the command before the copying is complete.

**-v *volume*,...**

The *volume* indicates the volume name that is the target of the operation.

To indicate multiple *volumes*, separate each volume name with a comma (,) as the delimiter.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.11 sdtparam - Configuration parameter operations

### SYNOPSIS

```
sdtparam -G [ -p param,... ]  
sdtparam -S -p param=val [ ,param=val,... ] [ -e default ]
```

### DESCRIPTION

Use `sdtparam` to perform operations on GDS configuration parameter.  
You must be superuser to use this command.

### PRIMARY OPTIONS

You can use one of the following options.

#### -G

##### Get

Displays the current value of configuration parameter or parameters specified by *param*.  
When using cluster system, parameter value of the current node will be displayed. If the `-p` option is omitted, all configuration parameters are displayed.

#### -S

##### Set

Sets the value specified by *val* to the configuration parameter or parameters specified by *param*.  
When using cluster system, this option sets the parameter value of the current node. The new value becomes valid upon returning from the command, and rebooting the system will not change the value.

### SUB OPTIONS

SUB OPTIONS are as follows:

#### -e *default* (when using -S)

Resets all configuration parameter values to default.  
When indicated at the same time as `-p` option, this option is ignored.

#### -p *param*,...(when using -G)

Displays the configuration parameter *param* value.

#### -p *param=val*[,*param=val*,...] (when using -S)

Sets *val* to configuration parameter *param*.

You can indicate the following combination to *param* and *val*.

#### **copy\_concurrency=***num*

Sets the maximum number of synchronization copying you can execute simultaneously to *num*.  
The value is set to 8 by default.  
Values from 1 to 1024 may be specified for *num*.

#### **copy\_delay=***msec*

This option delays the synchronization copying by milliseconds specified by *msec*, when the copying is initiated by an event other than hot spare.  
The value is set to 0 by default.  
Values from 0 to 1000 may be specified for *msec*.

#### **spare\_copy\_delay=***msec*

This option delays the synchronization copying by milliseconds specified by *msec*, when the copying is initiated by hot spare.  
The value is set to 50 by default. Values from 0 to 1000 may be specified for *msec*.

**RETURNED VALUE**

Upon successful completion, a value of 0 is returned.  
Otherwise, a non-zero value is returned.

**CAUTION**

The default values and the range of values you can use for configuration parameters may change in the future.

## D.12 sdxproxy - Proxy object operations

### SYNOPSIS

```
sdxproxy Break -c class -p proxy [-e force,restore]

sdxproxy Cancel -c class -p proxy

sdxproxy Join -c class -m master -p proxy
                [-a mvol=pvol:jrm[:pslice][,mvol=pvol:jrm[:pslice],...]]
                [-e delay=msec,softcopy,syncmode,waitsync]

sdxproxy Part -c class -p proxy,... [-a attribute=value] [-e instant,mode=val,unlock]

sdxproxy Rejoin -c class -p proxy,...[-e delay=msec,softcopy,waitsync]

sdxproxy RejoinRestore -c class -p proxy,...
                       [-e delay=msec,instant,nowaitsync,softcopy]

sdxproxy Relate -c class -m master -p proxy

sdxproxy Restore -c class -p proxy,... [-e instant,nowaitsync]

sdxproxy Swap -c class -p proxy

sdxproxy Update -c class -p proxy,... [-e instant,nowaitsync]
```

### DESCRIPTION

Use sdxproxy to perform operations on proxy objects.  
You must have superuser authority to execute the sdxproxy command.



[See](#)

- A.1.7 Exception to Proxy Configuration
- A.1.8 Number of Proxy Volumes
- A.1.9 Proxy Volume Size
- A.1.10 Proxy Group Size
- A.2.13 Using the Advanced Copy Function in a Proxy Configuration
- A.2.14 Instant Snapshot by OPC

### PRIMARY OPTIONS

You can use one of the following options.

#### Break

Cancels master-proxy relationship between a pair of volumes or groups. You can cancel the relationship when the objects are either joined or parted.

*proxy* indicates a proxy volume or a proxy group to be cancelled. You cannot indicate a proxy volume within a proxy group for the cancellation.

Even after the relationship has been cancelled, both the proxy volume and the proxy volume within a proxy group can be used as ordinary volumes retaining their volume data and attributes.

You can also cancel the relationship when master volume and proxy volume are in use. However, if the joined master volume is in use, the data integrity of the proxy volume should be ensured by either the file system layer or database layer whichever is managing the data, after the relationship is cancelled. For example, if a master volume is being used as a file system, you must use the `umount(8)` command to unmount the file system, before canceling the relationship.

The command will result in an error when:

- Copying is in process from the master volume to the proxy volume (except for when using

- e force)
- Copying is in process from the proxy volume to the master volume (except for when using -e restore)

## Cancel

Cancels (releases) sessions of disk unit copy functions existing between parted proxies and masters.

Specify a parted proxy group or a parted proxy volume into *proxy*. When *proxy* is a proxy group, all parted proxies within the proxy group become targets. A parted proxy volume within a proxy group can also be specified.

Sessions can be canceled even if master volumes and proxy volumes are in use. Even after canceling the sessions, the masters and the proxies are left parted. The volume statuses of the masters and the proxies also remain unchanged. However, if sessions are canceled when copying from masters to proxies and vice versa are in process, data becomes INVALID. If that happens, perform copying again, and the volume statuses will be restored when the copying is complete.

## Join

A pair of volumes or a pair of groups are related and joined as master and proxy.

When joining a pair of volumes, synchronization copying of master to proxy volume is performed after returning from the command (when using -e waitsync, before returning from the command). The size of the master and proxy volume must be the same. Also, they must be volumes belonging to different groups.

When joining a pair of groups, proxy volumes with the same offset and size as the master volumes within master group will be created in the proxy group. Synchronization copying from master to proxy volume is also performed after returning from the command (when using -e waitsync, before returning from the command). The access mode of the proxy volume that will be created in the proxy group is set to ro (read-only). The size of a proxy group must be larger than the last block number of volumes within the master group. When there is no volume in a master group or there is already a volume in a proxy group, joining fails resulting in an error. In addition, a master group and a proxy group must be mirror groups.

When copy functions of disk units are available, synchronization copying from masters to proxies is performed with those copy functions (except for when using -e softcopy).

A proxy volume that is joined cannot be accessed or activated. In order to access the proxy volume, part the volume from master using the Part option, or break the relationship with master using the Break option.

You can create multiple snapshots by joining another proxy to a master, which is already joined with a proxy. However, the total number of slices which belong to the master volume and slices which belong to the proxy volumes that are related to the master volume must not exceed 32.

Master volumes that are already related to a proxy cannot be joined to another master as a proxy. Also, a proxy that is already related to a master cannot be joined with another proxy.

The command will result in an error when:

- The proxy volume is activated
- There is a slice that is temporarily detached or in copying process among any of the slices comprising the proxy volume or the master volume
- Copying is in process between the master volume and the other proxy volume
- The master volume is in INVALID status

## Part

Separates a proxy or proxies in the joined status from the master. The master-proxy relationship will be maintained after parted. The parted proxy volume will be the snapshot containing the copy of the master volume data at the time of parting. By using the parted proxy volume, you can for instance, create a backup of the master volume at the time it was parted, or use it for other purposes.

*proxy* indicates a proxy volume or a proxy group in joined status. When proxy group is specified, all proxy volumes in the group will be parted. You can also indicate a proxy volume in joined status in a proxy group.

After the parting is complete, the proxy volume will be activated as an independent volume and become accessible using the following special file.

`/dev/sfdsk/class/[r]dsk/volume_name`

When the proxy volume belongs to a "shared" type class, it will be activated on all nodes defined in the class scope.

You can part master and proxy volumes even if the master volume is activated, but the data integrity of the parted proxy volume must be ensured by the file system layer or the database layer that is managing the data. For instance, if you are using the master volume as a file system, you must use the `umount(8)` command to unmount the file system, before parting.

The command will result in an error when:

- Copying is in process from the master volume to the proxy volume (except for when using `-e instant`)
- Copying is in process from the proxy volume to the master volume

## Rejoin

Rejoins one or more parted proxy with the master.

*proxy* indicates parted proxy volume, or proxy group. When proxy group is indicated, all proxy volumes in the group will be rejoined. You can also indicate the parted proxy volume in a proxy group.

Synchronization copying of master to proxy volume is performed after returning from the command (when using `-e waitsync`, before returning from the command). When copy functions of disk units are available, synchronization copying is performed with those copy functions (except for when using `-e softcopy`).

When more than one proxy volume related to the same master volume is specified simultaneously, this command will result in an error.

The command will result in an error when:

- The proxy volume is activated
- There is a slice that is in copying process among any of the slices comprising the proxy volume or the master volume
- Copying is in process between the master volume and the other proxy volume
- The master volume is in INVALID status

## RejoinRestore

Rejoins a proxy or proxies in the parted status with the master and restores the master volume data using the proxy volume data. Master volume data are restored by synchronization copying from the proxy volume to the master volume. When executing the command using this option, the master volume data are overwritten with the proxy volume data.

*proxy* indicates a proxy volume or a proxy group in parted status. When a proxy group is specified, all proxy volumes in the group will be rejoined and data of the related master volumes will be restored. You can also indicate a proxy volume in parted status in a proxy group.

Synchronization copying of master to proxy volume is performed before returning from the command (when using `-e waitsync`, after returning from the command). When copy functions of disk units are available, synchronization copying from masters to proxies is performed with those copy functions (except for when using `-e softcopy`).

When more than one proxy volume related to the same master volume is specified simultaneously, this command will result in an error.

The command will result in an error when:

- The master volume or the proxy volume is activated
- There is a slice that is in copying process among any of the slices comprising the proxy volume or the master volume
- Copying is in process between the master volume and the other proxy volume
- The master volume is in INVALID status

## Relate

Relates and parts a pair of volumes or a pair of groups as a master and a proxy. This operation does not change data, statuses and attributes of the master and the proxy. To the related master and proxy, sessions by copy functions of disk units are not set.

To relate a pair of volumes, the master volume and the proxy volume must be equivalent in size.

Those volumes also need to be volumes that belong to different mirror groups or different single disks.

To relate a pair of groups, the volume layout (offsets and sizes) of the master group and the proxy group must be same. Those groups also need to be mirror groups.

For masters to whom proxies are already related, other additional proxies can be related. However, the number of slices comprising a master volume and all the proxy volumes related to the master is limited to 32 in total.

Masters to whom proxies are already related cannot be related as proxies to other masters, or for proxies already related to masters, other proxies cannot be related.

This command will result in an error when:

- A slice being copied or temporarily detached exists in the master volume or the proxy volume
- Copying is in process between the master volume and another proxy volume

## Restore

Copies data from a parted proxy to a master and restores contents of the master. With the OPC function, data of the proxy at the moment is copied (overwritten) to the master. The command returns control after the copying is complete (right after the copying starts when using the `-e instant` option and the `-e nowaitsync` option). If the OPC function is unavailable, the command fails.

Specify one or more proxy groups or parted proxy volumes into *proxy*. When *proxy* is a proxy group, all the parted volumes within the proxy group become targets. A parted proxy volume within a proxy group can also be specified. Do not specify multiple proxy volumes related to the same master volume simultaneously.

The Restore operations can be performed even if proxy volumes are active, but it is necessary to secure consistency of data copied to master volumes in the file system layer or database layer managing the data. For example, if the proxy volume is used as a file system, unmount the file system with the `umount(8)` command and then perform restoration.

This command will result in an error when:

- The master volume is active
- A slice being copied exists in the master volume or the proxy volume
- Copying is in process between the master volume and another proxy volume
- A proxy volume joined to the master volume exists
- The proxy volume is in INVALID status

## Swap

Swaps the master's slices with the proxy's slices.

*proxy* indicates a proxy volume or a proxy group in the joined status. You cannot indicate a proxy volume within a proxy group for swapping.

You can swap the slices when master is in use.

The command will result in an error when:

- There is a slice that is in copying process among any of the slices comprising the proxy volume or the master volume
- Copying is in process between the master volume and the other proxy volume
- The proxy volume is in INVALID status
- Between a master and a proxy, an EC session exists

## Update

Copies data from a master to a parted proxy and updates contents of the proxy. With the OPC function, data of the master at the moment is copied (overwritten) to the proxy. The command returns control after the copying is complete (right after the copying starts when using the `-e instant` option and the `-e nowaitsync` option).

Updated proxy volumes become snapshots that have copies (replicas) of data of master volumes at the moment. By use of the updated proxy volumes, creating backups of master volumes at the moment and running other services become possible.

Specify one or more proxy groups or parted proxy volumes into *proxy*. When a proxy group is specified, all the parted proxy volumes within the proxy group become targets. A parted proxy volume

within a proxy group can also be specified. Do not specify multiple proxy volumes related to the same master volume simultaneously.

The Update operations can be performed even if master volumes are active, but it is necessary to secure consistency of data copied to proxy volumes in the file system layer or database layer managing the data. For example, if the master volume is used as a file system, unmount the file system with the `umount(8)` command and then perform update.

This command will result in an error when:

- The proxy volume is active
- A slice being copied exists in the master volume or the proxy volume
- Copying is in process between the master volume and another proxy volume
- The master volume is in an INVALID status

## SUB OPTIONS

You can indicate the following SUB OPTIONS.

### **-a *attribute=value*** (when using **Part**)

Sets *attribute* that is the attribute of the parted proxy volume to *value*. This attribute value becomes invalid as the proxy volume is rejoined to the master volume with the **Rejoin** or **RejoinRestore** option.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value. The equal sign (=) is always necessary between *attribute* and *value*.

You can specify one of the following combinations to attribute and value.

### **pjrm=on** or **pjrm=off** (default is **on**)

Sets the just resynchronization mechanism mode for proxies.

**on**

Turns "on" the just resynchronization mechanism mode for proxies.

**off**

Turns "off" the just resynchronization mechanism mode for proxies.

### **-a *mvol=pvol:jrm [:pslice] [,mvol=pvol:jrm [:pslice] ,... ]*** (when using **Join**)

Indicates the proxy volume attributes.

*mvol* must always be followed by an equal (=) sign, and *pvol*, *jrm* and *pslice* are delimited by a colon (:). When indicating attributes for more than one proxy volume, combine specifiers with a comma (,) as the delimiter.

When joining a pair of groups, *mvol* indicates the volume name of the master volume within the master group. *pvol* indicates the volume name of the proxy volume that will be created in the proxy group corresponding to the master volume specified by *mvol*, *jrm* indicates the just resynchronization mechanism mode for the volume (on or off), and *pslice* indicates the physical slice attribute (on or off). You must specify attributes for all proxy volumes created for respective master volumes within the master group. If *:pslice* is omitted, the physical slice attribute of the proxy volume will be equivalent to that of the corresponding master volume.

When joining a pair of volumes, *mvol* indicates the volume name of the master volume, *pvol* indicates the volume name of the proxy volume, *jrm* indicates the just resynchronization mechanism mode for the volume (on or off), and *pslice* indicates the physical slice attribute (on or off). *mvol* and *pvol* should match with *master* and *proxy* respectively. When not using this option, the proxy volume attributes will be the same as before it was joined.

### **-c *class***

*class* indicates the class name to which the master object or proxy object that will be the target of operation belongs

### **-e *delay=msec*** (when using **Join**, **Rejoin**, **RejoinRestore**)

Delays the issuing of the input/output request to the disk at the time of synchronization copying between master volume and proxy volume by milliseconds, specified by *msec*.

This option allows users to adjust the influence on the application accessing the master volume.



When copying is performed with a copy function of a disk unit, this option setting is ignored.

When this option is omitted, the delay will be set to 0.

Values from 0 to 1000 may be specified for *msec*.

**-e force (when using Break)**

Forcibly breaks master-proxy relationship even when copying is in process between master and proxy.

When using this option, copying process will be cancelled and the status of the proxy volume will become INVALID.

**-e instant (when using Part)**

Parts proxy volumes and creates virtual snapshots of master volumes with the OPC function even if copying from the master to the proxy is in process. After returning from the command, the parted proxy volume will become accessible before the copying is complete and will serve as a snapshot containing data of the master volume at the time of parted. When copying is in process from proxy to master, or if hard copy function is not available, the command will result in an error.

**-e instant (when using RejoinRestore)**

Restoration will be completed instantly, and the command will be returned. After returning from the command, synchronization copying from proxy volume to master volume will automatically begin. Although the copying is still in process, restoration will appear to be complete. After returning from the command, the master volume can be activated and accessed, before the copying is complete. Master volume data will appear to have been overwritten by the proxy volume data at the time of executing the command.

**-e instant (when using Restore, Update)**

Instantly completes restore or update and returns from the command. After returning from the command, background copying with the OPC function is performed. Before the background copying is complete, you may start the copy destination volumes for access to valid data.

**-e mode=*val* (when using Part)**

Indicates the access mode of the proxy volume which will be activated. *val* indicates either of the following options.

**rw**

Sets access mode for read and write.

**ro**

Sets access mode for read only.

Opening a read-only volume in write mode will result in an error.

Although proxy volume will be activated in the access mode specified by *val*, the proxy volume's access mode attribute will remain unchanged. Access mode specified by *val* is valid only while the proxy volume is activated ("Current Access Mode") and will become invalid once the proxy volume is stopped. When the proxy volume is restarted, it will start in the mode set by access mode attribute ("Default Access Mode"), except for when the access mode is specified at the time of restart.

When this option is omitted, proxy volume will be activated in the access mode according to the access mode attribute set on each node.

**-e nowaitsync (when using RejoinRestore, Restore, Update)**

Returns control from the command right after copying starts. After returning from the command, wait until the copying is complete to start the copy destination volumes. To start copy destination volumes without waiting until copying is complete, use the **-e instant** option. When the **-e instant** option is specified simultaneously, this option is ignored.

**-e restore (when using Break)**

Cancels copying from a proxy to a master when such copying is in process and forces the command to break the relationship between the master and the proxy.

If copying is canceled and relationships are broken with this option, the master volumes after this operation becomes INVALID.

**-e softcopy (when using Join, Rejoin or RejoinRestore)**

Copy function of the disk unit will not be used for synchronization copying between master volume and proxy volume.

**-e syncmode** (when using **Join**)

When the REC function is used for synchronization copying from the master volume to the proxy volume, the transmission mode of REC is set to the synchronous transfer mode. The default is the asynchronous transfer mode. When the REC function is not used, this option setting is ignored.

**-e unlock** (when using **Part**)

The proxy volume will be activated regardless to whether it is locked. Lock mode will not be changed unless you change it with the `sdxattr -V` command.

**-e waitsync** (when using **Join** or **Rejoin**)

When performing synchronization copying, returns from the command after the copying is complete.

**-m master** (when using **Join**, **Relate**)

Specifies the master volume or the master group that is joined or related.

*master* indicates the volume name of the master volume, or the group name of the master group.

**-p proxy** (when using **Break**, **Cancel**, **Join**, **Swap** or **Relate**)

*proxy* indicates a proxy volume or a proxy group that is the target of the operation.

*proxy* indicates the volume name of the proxy volume, or the group name of the proxy group.

**-p proxy,...** (when using **Part**, **Rejoin**, **RejoinRestore**, **Restore** or **Update**)

*proxy* indicates one or more proxy volumes or proxy groups that will be the target of the operation. When indicating more than one volume or group, they must belong in the same class.

*proxy* indicates the volume name of the proxy volume or the group name of the proxy group that will be the target of the operation

When indicating more than one volume name or group name, combine them with a comma (,) as the delimiter.

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.13 sdxshadowdisk - Shadow disk operations

### SYNOPSIS

```
sdxshadowdisk -C -c class -g group -d disk,...
                [-a attribute=value[,attribute=value]]
sdxshadowdisk -D -c class -g group -d disk
sdxshadowdisk -M -c class
                -d device=disk[:type][,device=disk[:type],...]
sdxshadowdisk -R -c class -d disk
```

### DESCRIPTION

Use sdxshadowdisk to perform operations on shadow disks specified by *disk*. You must be superuser to use this command.

### PRIMARY OPTIONS

You have a choice of the following options.

#### -C

##### **Connect**

Connects one or more shadow disks (single type or undefined type) specified by *disk*,... to a group specified by *group*. *class* indicates the name of the shadow class with which *disk* is registered.

If there is no shadow group with the name specified by *group*, it is created automatically.

You cannot connect a shadow disk of the single type to an existing shadow group. Also, multiple shadow disks of the single type cannot be connected to the same shadow group simultaneously.

The type attribute of a shadow disk connected to a shadow group will be changed to match the type attribute of that group (mirror, stripe or concatenation). Shadow disks and lower level groups that are connected to the same shadow group will be mirrored, striped or concatenated, depending on their type attributes.

Details about connecting a shadow disk to a shadow group of the mirror type, the stripe type and the concatenation type are described below respectively.

#### - When connecting to a shadow group of the mirror type

Shadow disks and lower level shadow groups connected to the same shadow group of the mirror type will be mirrored one another. When only one shadow disk or one lower level shadow group is connected to a shadow group of the mirror type, the shadow volume created within that shadow group will not be mirrored. When configuring a mirroring environment with "n"-way multiplexing, "n" numbers of shadow disks or lower level shadow groups must be connected. A maximum of eight-way multiplex mirroring is supported.

If a shadow disk is connected to a shadow group of the mirror type including a shadow volume, synchronization copying of the shadow volume is not performed. To ensure synchronization for a shadow volume of the mirror type, the mirror volume must be properly synchronized with GDS that manages the mirror volumes corresponding to shadow volumes.

By connecting a shadow disk of the single type including a shadow volume to a group of the mirror type, the shadow volume can also be changed from the single type to the mirror type.

The available size of a shadow group of the mirror type (available capacity as shadow volumes) will be the same size as that of the smallest shadow disk or lower level shadow group connected. If connecting *disk* results in a decrease in the available size of *group*, a warning message will be sent to standard error output.

#### - When connecting to a shadow group of the stripe type

Shadow disks specified by *disk*,... will be connected to *group* in the order they are listed. Disks connected to a stripe group in another domain should be connected in the same order. For the disk connecting order, check the DISKS field displayed with the sdxinfo -G command. Respective

shadow disks and lower level shadow groups connected to the same shadow group of the stripe type will configure stripe columns, and will be striped in the order they were connected. When only one shadow disk or one lower level shadow group is connected to a shadow group of the stripe type, a shadow volume cannot be created within that shadow group. When striping over "n" number of columns, "n" number of shadow disks or lower level shadow groups must be connected. Striping is supported over a minimum of two columns and a maximum of 64 columns.

When a shadow group of the stripe type specified by *group* already exists, stripe columns will be added after the existing stripe columns in group, in the order they are specified by *disk*,... However, a shadow disk with the available size smaller than the stripe width cannot be connected to the existing shadow group of the stripe type. In addition, you cannot increase stripe columns by connecting shadow disks to a stripe group with a shadow volume, or to a stripe group connected to a higher level shadow group.

The available size of a shadow group of the stripe type (available capacity as shadow volumes) equals the available size of the smallest shadow disk (or lower level shadow group) multiplied by the number of stripe columns, and rounded down to the common multiple of the stripe width times stripe columns and the cylinder size. If connecting *disk* decreases the available size of *group*, a warning message will be sent to standard error output.

You cannot connect a shadow disk of the single type to a shadow group of the stripe type.

#### - When connecting to a shadow group of the concatenation type

Shadow disks connected to the same shadow group of the concatenation type will be concatenated in the order they are specified by *disk*,... Disks connected to a concatenation group in another domain should be connected in the same order. The connection order of disks can be viewed in the DISKS field provided by the `sdxinfo -G` command. A maximum of 64 disks can be concatenated.

The available size of a shadow group of the concatenation type (available capacity as shadow volumes) equals the total of the available size of connected shadow disks.

The available size of an existing shadow group of the concatenation type can be increased by connecting shadow disks. When a shadow group of the concatenation type specified by *group* already exists, shadow disks will be concatenated in the order they were specified by *disk*,... after the disk that was last concatenated in *group*. However, you cannot add a shadow disk to a lower level shadow group of the concatenation type if the highest level shadow group of the stripe type already has a shadow volume. Also if the order of connecting shadow groups from the higher level is the mirror type, the stripe type and the concatenation type, a shadow disk cannot be connected to the lowest level shadow group of the concatenation type. You cannot connect a shadow disk of the single type to a shadow group of the concatenation type.

### -D

#### **Disconnect**

Disconnects a shadow disk specified by *disk* from a shadow group specified by *group*. *class* indicates the name of the shadow class with which the *disk* is registered, and *group* indicates the name of the shadow group to which *disk* is connected.

The disconnected shadow disk will have the original type attribute again (single or undefined).

If only *disk* is connected to *group*, *group* will automatically be removed upon disconnecting *disk*. However, when *disk* is the only object connected to *group* and *group* is connected to a higher level shadow group, disconnection will result in an error. In such a case, disconnect *group* from the higher level shadow group using the `sdxshadowgroup -D` command, and then disconnect *disk*.

You cannot disconnect *disk* if the disconnection will result in a change on the status of any of the existing shadow volumes within *group*.

Conditions that do not allow you to disconnect a shadow disk from a shadow group of the mirror type, stripe type or concatenation type are as below.

#### - When disconnecting from a shadow group of the mirror type

For example, you cannot disconnect disk from a shadow group of the mirror type specified by *group* if a shadow volume exists within the group, and *disk* is the only object connected to *group*.

**- When disconnecting from a shadow group of the stripe type**

You cannot disconnect a shadow disk from a shadow group of the stripe type including an existing shadow volume, or from a shadow group of the stripe type connected to a higher level shadow group..

**- When disconnecting from a shadow group of the concatenation type**

The only disk you can disconnect from a shadow group of the concatenation type is the shadow disk that was concatenated last.

A shadow disk containing shadow volume data cannot be disconnected from a shadow group of the concatenation type.

You cannot disconnect a shadow disk from a lower level shadow group of the concatenation type if the highest level shadow group has an existing shadow volume. Also, if the order of connecting shadow groups from the higher level is the mirror type, stripe type and the concatenation type, a shadow disk cannot be disconnected from the lowest level shadow group of the concatenation type.

**-M**

**Make**

Registers one or more physical disks specified by *device* with a shadow class. *class* indicates the name of the destination shadow class. Once physical disks are registered, they can then be managed by GDS. Accordingly, the user can perform operations on the disk by use of the disk name specified by *disk*. However, device will be no longer managed by GDS if the current node is rebooted or if the GDS daemon on the current node is re-launched because the configuration information of a shadow class is only retained on the memory of the current node but not stored on the private slice.

If no shadow class with the same name as *class* exists, then it is automatically created. The type attribute of the shadow class is "local," and objects in the shadow class are only available on the current node.

A shadow class can include physical disks that are not registered with other classes in the current domain and on which the private slices of GDS exist. In other words, a shadow class can include physical disks that are registered with classes in other domains. Physical disks can be registered with the same shadow class when they are registered with classes that have the same names in other domains. In addition, disks with the private slices of different sizes cannot be registered with the same shadow class.

While contents on physical disks registered by the *sdxdisk* command are initialized, contents on physical disks registered by the *sdxshadowdisk* command are not changed.

**-R**

**Remove**

Removes a shadow disk specified by *disk* from a shadow class specified by *class*. *class* indicates the name of the shadow class with which *disk* is registered.

The removed shadow disk is no longer managed by GDS.

When the last shadow disk is removed from *class*, the shadow class definition is automatically removed as well.

A shadow disk cannot be removed when a shadow volume exists within *disk*, or when *disk* is connected to a shadow group.

**SUB OPTIONS**

SUB OPTIONS are as follows:

**-a *attribute=value*[,*attribute=value*] (when using -C)**

When using the -C option and defining a new group name with the -g option, a new shadow group is automatically created. This option sets *attribute* to *value* for the created *group* attribute.

The *attribute* indicates the attribute name, and the *value* indicates the attribute value. The equal sign (=) is always necessary between attribute and value. When specifying multiple attributes, each specifier set must be separated by a comma (,).

If no shadow group is created, specifying *value* different from the existing *group* attribute value will result in an error. You cannot change the attribute value of an existing *group*.

You can specify one of the following combination to *attribute* and *value*.

If multiple attributes are specified and any error occurs, the entire process is canceled.

**type=mirror, type=stripe or type=concat** (default is **mirror**)

Sets the type attribute of *group*.

**mirror**

Sets type attribute to "mirror".

**stripe**

Sets type attribute to "stripe".

**concat**

Sets type attribute to "concatenation".

**width=blks** (default is 32)

Sets the stripe width of *group*. *blks* indicates the stripe width in blocks (decimal numbers). One block is 512 bytes. For *blks*, you can specify an integer that is the power of two, is equal to or larger than one and is equal to or smaller than the available size of the smallest shadow disk among disks specified by *disk*,... If *group* is not the stripe type, this option will cause an error.

**-c class**

*class* indicates the name of the shadow class to which the shadow disk as the object of the operation is registered or is to be registered.

**-d device=disk [ :type ] [ ,device=disk [ :type ] ,... ]** ( when using **-M** )

*device* indicates the name of the physical disk, *disk*, the name of the disk, and *type*, the type attribute of the shadow disk. The equal sign (=) always follows *device*, and if *type* is specified it must be separated from *disk* by a colon (:). To register multiple *devices*, separate each specifier set as above with a comma (,).

The physical disk name can be specified with one of the following formats:

**sdX** ( for normal hard disks )

**emcpowerX** ( for emcpower disks )

X indicates the device identifier.

When *device* is registered with a class in another domain, the same disk name as that in the domain must be specified to *disk*.

One of the following types can be specified to *type*. The default value for the registered shadow disk is the undefined type.

**single**

Single type.

**undef**

Undefined type.

If *single* is specified to *type*, *device* is registered as a shadow disk of the *single* type. For the shadow disk of the *single* type, a shadow volume of the *single* type can be created with the *sdxshadowvolume* command even if it is not connected to a shadow group.

**-d disk** (when using **-D**, **-R**)

*disk* indicates the name of the shadow disk that is the object of the operation..

**-d** *disk*,... (when using **-C**)

*disk* indicates the name of the shadow disk that is the object of the operation. To connect multiple shadow disks, separate each disk name with a comma (,).

**-g** *group* (when using **-C,-D**)

*group* indicates the name of the shadow group to which the shadow disk as the object of the operation is connected or is to be connected.

## **RETURNED VALUE**

Upon successful completion, a value of 0 is returned.  
Otherwise, a non-zero value is returned.

## D.14 sdxshadowgroup - Shadow group operations

### SYNOPSIS

```
sdxshadowgroup -C -c class -h hgroup -l lgroup,...  
                [-a attribute=value[,attribute=value]]  
sdxshadowgroup -D -c class -h hgroup -l lgroup  
sdxshadowgroup -R -c class -g group
```

### DESCRIPTION

Use sdxshadowgroup to perform operations on shadow groups.  
You must be superuser to use this command.

### PRIMARY OPTIONS

You have a choice of the following options.

#### -C

##### **C**onnect

Connects one or more shadow groups (stripe type or concatenation type) specified by *lgroup,...* to a shadow group (mirror type or stripe type) specified by *hgroup*. *class* indicates the name of the shadow class to which *lgroup* belongs.

When no shadow group with the same name as *hgroup* exists, it is created automatically.

A shadow group specified by *hgroup* is referred to as a higher level shadow group, and a shadow group specified by *lgroup* is referred to as a lower level shadow group.

Lower level shadow groups and shadow disks connected to the same higher level shadow group are mirrored or striped according to the type attribute of the higher level shadow group. Connecting a shadow group to a higher level shadow group does not change the type attribute of the lower level shadow group.

You cannot connect shadow groups when:

- *lgroup* is the mirror group
- *hgroup* is the concatenation group
- type attributes of *lgroup* and *hgroup* are the same

In addition, a shadow group already including a shadow volume cannot be connected to another shadow group.

Details about connecting a shadow group to a shadow group of the mirror type or stripe type are described below respectively.

#### - When connecting to a shadow group of the mirror type

One or more shadow groups (stripe type or concatenation type) specified by *lgroup,...* can be connected to a shadow group of the mirror type specified by *hgroup*.

Shadow disks and lower level shadow groups connected to the same shadow group of the mirror type will be mirrored one another. When only one shadow disk or one lower level shadow group is connected, the shadow volume created within that shadow group of the mirror type will not be mirrored. When configuring a mirroring environment with "n"-way multiplexing, "n" numbers of shadow disks or lower level shadow groups must be connected. A maximum of eight-way multiplex mirroring is supported.

If a lower level shadow group is connected to a shadow group of the mirror type with a shadow volume, synchronization copying for the shadow volume is not performed. To ensure synchronization for a shadow volume of the mirror type, the mirror volume must be properly synchronized with GDS that manages the mirror volume corresponding to the shadow volume.

The available size of a shadow group of the mirror type (available capacity as shadow volumes) will be the same as that of the smallest shadow disk or lower level shadow group connected. If connecting *lgroup* decreases the available size of *hgroup*, a warning message will be sent to standard error output.



**- When connecting to a shadow group of the stripe type**

One or more shadow groups (concatenation type) specified by *lgroup*,... can be connected to a shadow group of the stripe type specified by *hgroup*. Shadow groups specified by *lgroup*,... will be connected to *hgroup* in the order they are listed. Lower level groups connected to a stripe group in another domain should be connected in the same order. The order of connecting lower level groups can be viewed in the DISKS field provided by the `sdxinfo -G` command.

Respective shadow disks and lower level shadow groups connected to the same shadow group of the stripe type will configure stripe columns and will be striped in the order they were connected. When only one shadow disk or one lower level shadow group is connected, a shadow volume cannot be created within that shadow group. When striping over "n" number of columns, "n" number of shadow disks or lower level shadow groups must be connected. Striping is supported over a minimum of two columns and a maximum of 64 columns.

When a shadow group of the stripe type specified by *hgroup* already exists, stripe columns will be added after the existing stripe columns in *hgroup*, in the order they are specified by *lgroup*,... However, a shadow group with the available size smaller than the stripe width cannot be connected to an existing shadow group of the stripe type. In addition, you cannot increase stripe columns by connecting shadow groups to a stripe group with a shadow volume, or to a stripe group connected to a higher level shadow group.

The available size of a shadow group of the stripe type (available capacity as shadow volumes) equals the available size of the smallest shadow disk (or the lower level shadow group) connected multiplied by the number of stripe columns, and rounded down to the common multiple of the stripe width times stripe columns and the cylinder size. If connecting *lgroup* decreases the available size of *hgroup*, a warning message will be sent to standard error output.

**-D**

**Disconnect**

Disconnects a shadow group specified by *lgroup* from a shadow group specified by *hgroup*. *class* indicates the name of the shadow class to which *lgroup* belongs, and *hgroup* indicates the name of the higher level shadow group to which *lgroup* is connected.

When *lgroup* is the only object connected to *hgroup*, *hgroup* will automatically be removed upon disconnecting *lgroup*. However, when *lgroup* is the only object connected to *hgroup*, and *hgroup* is connected to a higher level shadow group, disconnection will result in an error. In such a case, disconnect *hgroup* from its higher level shadow group, and then disconnect *lgroup*.

You cannot disconnect *lgroup* if the disconnection may result in a change in the status of any existing shadow volume within *hgroup*.

Conditions that do not allow you to disconnect a shadow group from a shadow group of the mirror type or a shadow group of the stripe type are as below.

**- When disconnecting from a higher level shadow group of the mirror type**

For example, you cannot disconnect *lgroup* from a shadow group of the mirror type specified by *hgroup* if a shadow volume exists within that group, and *lgroup* is the only object connected to *hgroup*.

**- When disconnecting from a shadow group of the stripe type**

You cannot disconnect a lower level shadow group from a shadow group of the stripe type with an existing shadow volume, or from a shadow group of the stripe type connected to a higher level shadow group.

**-R**

**Remove**

Removes the shadow group definition specified by *group*. *class* indicates the name of the shadow class to which *group* belongs.

Shadow disks and lower level shadow groups connected to *group* will be disconnected. The disconnected shadow disk will have the original type attribute (single or undefined).

The definition cannot be removed when a shadow volume exists within *group*, or when *group* is connected to a higher level shadow group.

## SUB OPTIONS

SUB OPTIONS are as follows:

**-a *attribute=value*[,*attribute=value*] (when using -C)**

When using the -C option and defining a new group name with the -h option, a new shadow group, *hgroup*, is automatically created. This option sets *attribute* to *value* for the created *hgroup* attribute.

*attribute* indicates the attribute name, and *value* indicates the attribute value. The equal sign (=) is always necessary between *attribute* and *value*. When specifying multiple attributes, each specifier must be separated by a comma (,).

If no shadow group is created, specifying an attribute value different from the attribute value of the existing *hgroup* will result in an error. You cannot change the attribute value of an existing *hgroup*.

You can specify one of the following combinations to *attribute* and *value*.

If multiple attributes are specified and any error occurs, the entire process is canceled.

**type=mirror or type=stripe (default is mirror)**

Sets the type attribute of *hgroup*.

**mirror**

Sets the type attribute to "mirror".

**stripe**

Sets the type attribute to "stripe".

**width=*blks* (default is 32)**

Sets the stripe width of *hgroup*. *blks* indicates the stripe width in blocks (decimal numbers). One block is 512 bytes. For *blks*, you can specify an integer that is the power of two, equal to or larger than one and equal to or smaller than the available size of the smallest shadow group among groups specified by *lgroup*,... If *hgroup* is not the stripe type, this option will cause in an error.

**-c *class***

*class* indicates the name of the shadow class to which the shadow group as the object of the operation belongs.

**-g *group* (when using -R)**

*group* indicates the name of the shadow group that is the object of the operation.

**-h *hgroup* (when using -C,-D)**

*hgroup* indicates the name of the higher level shadow group to which the lower level shadow group as the object of the operation is connected or is to be connected.

**-l *lgroup* (when using -D)**

*lgroup* indicates the name of the lower level shadow group as the object of the operation.

**-l *lgroup*,... (when using -C)**

*lgroup* indicates the name of the lower level shadow group as the object of the operation. To connect multiple shadow groups, separate each group name with a comma (,).

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.15 sdxshadowvolume - Shadow volume operations

### SYNOPSIS

```
sdxshadowvolume -F -c class [-v volume,...]
sdxshadowvolume -M -c class {-g group | -d disk} -v volume -s size
sdxshadowvolume -N -c class [-v volume,...] [-e mode=val]
sdxshadowvolume -R -c class -v volume
```

### DESCRIPTION

Use sdxshadowvolume to perform operations on shadow volumes specified by *volume*. You must be superuser to use this command.

### PRIMARY OPTIONS

You have a choice of the following options.

#### -F

##### **oF**ffline

Stops one or more shadow volumes specified by *volume,...* By default all of the shadow volumes within *class* are turned off. Volumes turned off cannot be accessed.

This results in an error if the volume is in use.

#### -M

##### **Ma**ke

Creates a shadow volume specified by *volume* to the highest level shadow group specified by *group* or to a shadow disk of the single type specified by *disk*. *size* indicates the number of blocks on *volume*, *class* indicates the name of the shadow class associated with *group* or *disk*.

The access mode of the created shadow volume is ro (read-only). Synchronization copying is not performed on shadow volumes of the mirror type. The attribute of just resynchronization mode is off. In addition, the physical slice attribute is always off even though the slice configuring the shadow volume is registered with partition table. A maximum of 224 shadow volumes can be created to the same *group* or *disk*.

Once created, the shadow volume is activated, and can be accessed using the following special files:

```
/dev/sfdsk/class/rdisk/volume
/dev/sfdsk/class/dsk/volume
```

Create shadow volumes conforming to the following rules in order to use the shadow volumes to access data in the corresponding logical volumes.

- Equal to the corresponding logical volumes in size. For volume sizes, check the BLOCKS field displayed with the sdxinfo -V command.
- Whose first block numbers must be consistent with the first block numbers of the corresponding logical volumes. Therefore, create shadow volumes within the same shadow group or shadow disk in ascending order in conformity to the order of the first block numbers of the corresponding logical volumes. For the first block numbers of volumes, check the 1STBLK field displayed with the sdxinfo -V command.

The features of created shadow volumes when *group* is the mirror type and the stripe type are as below.

#### - When *group* is a mirror type

In a shadow group of the mirror type, shadow volumes of the mirror type with mirror-multiplexing equal to the number of shadow disks or lower level shadow groups connected (maximum of eight) are created. When only one shadow disk or one lower level shadow group is connected, the created shadow volume will not be mirrored.

Synchronization copying is not performed on shadow volumes even if the shadow volumes of the

mirror type are created. To ensure synchronization for a shadow volume of the mirror type, the mirror volume must be properly synchronized with GDS that manages the mirror volume corresponding to the shadow volume.

**- When group is a stripe group**

In a shadow group of the stripe type, shadow volumes of the stripe type with columns equal to the number of shadow disks or lower level shadow groups connected (maximum of 64) are created. When only one shadow disk or one lower level shadow group is connected, a shadow volume cannot be created.

**-N**

**oNline**

Activates one or more shadow volumes specified by *volume*,... By default all of the shadow volumes within *class* are activated. The activated shadow volumes become accessible.

Synchronization copying is not performed on shadow volumes if the shadow volumes of the mirror type are activated. To ensure synchronization for a shadow volume of the mirror type, the mirror volume must be properly synchronized with GDS that manages the mirror volume corresponding to the shadow volume.

**-R**

**Remove**

Removes a shadow volume specified by *volume* and releases the disk area that the shadow group or the shadow disk of the single type was occupying.

This results in an error if the shadow volume is active.

No data stored on *volume* will be lost due to this removal of *volume*.

## SUB OPTIONS

SUB OPTIONS are as follows:

**-c class**

*class* indicates the name of the shadow class to which the shadow volume as the object of the operation belongs, or in which the shadow volume is to be created.

**-d disk (when using -M)**

*disk* indicates the name of a shadow disk of the single type in which the shadow volume of the single type will be created.

**-e mode=*val* (when using -N)**

Specifies the access mode for one or more shadow volumes that will be activated.

You can specify either of the following options to *val*.

**rw**

Sets the access mode to read and write.

**ro**

Sets the access mode to read only. Opening a read-only volume in the write mode will result in an error.

Although shadow volumes are activated in the access mode specified by *val*, the access mode attributes for the shadow volumes will remain unchanged. The access mode specified by *val* ("current access mode") is valid only while the shadow volume is active and will become invalid once the shadow volume is stopped. When the shadow volume is restarted, it will start in the mode according to the access mode attribute ("default access mode") unless the access mode is specified by this option.

In order to start a shadow volume that is already activated on the current node in a different access mode, you must first stop the shadow volume.

**-g** *group* (when using **-M**)

*group* indicates the name of a shadow group in which the shadow volume will be created.

**-s** *size* (when using **-M**)

Specifies the size of the *volume* being created in blocks (decimal numbers). One block is 512 bytes.

When *group* is the stripe type, the size of *volume* created will be the *size* rounded up to a common multiple of the stripe width multiplied by stripe columns and the cylinder size. In other cases, the size of *volume* created will be the *size* rounded up to the integer multiple of the cylinder size.

**-v** *volume* (when using **-M,-R**)

*volume* indicates the name of the shadow volume as the object of the operation.

**-v** *volume,...* (when using **-F,-N**)

*volume,...* indicates the names of one or more shadow volumes as the objects of the operation. To specify multiple shadow volumes, separate each volume name with a comma (,).

## RETURNED VALUE

Upon successful completion, a value of 0 is returned.

Otherwise, a non-zero value is returned.

## D.16 Volume Creation Using Command

In this section, operation outline of volume creation is explained. Please use as a reference when configuring the environment. For details, see the Command Reference.



### Note Resource Registration

In order to define the configuration of GDS objects such as classes and volumes, PRIMECLUSTER resources must be registered in advance. For details about registering resources, see "Appendix H Register shared disk units."

### (1) Creating a mirror volume

The following example shows the procedures for creating a volume by mirroring physical disks named sda and sdb.

#### 1) Registering disks to class

Register the physical disks with a class. When the specified class does not exist, it will be created automatically.

Example: Registering physical disks sda and sdb with shared class "Class1", which is shared on node1 and node2, and naming those disks Disk1 and Disk2.

```
# sdxdisk -M -c Class1 -a type=shared,scope=node1:node2 -d sda=Disk1,sdb=Disk2
```

#### 2) Connecting the disks to a mirror group

Connect the disks to a mirror group. When the specified mirror group does not exist, it will be created automatically.

Example: Connecting "Disk1" and "Disk2" to mirror group "Group1".

```
# sdxdisk -C -c Class1 -g Group1 -d Disk1,Disk2
```

#### 3) Creating a mirror volume

Create a volume within the mirror group.

Example: Creating a volume of 1,000 blocks within mirror group "Group1", and assigning a volume name "Volume1".

```
# sdxvolume -M -c Class1 -g Group1 -v Volume1 -s 1000
```

After returning from the command, synchronization copying will automatically begin.

#### 4) Confirming the completion of procedure

Confirm that the synchronization copying is complete.

**Example: Confirming synchronization copying of volume "Volume1" is complete.**

```
# sdxinfo -S -o Volume1
OBJ   CLASS  GROUP  DISK   VOLUME  STATUS
-----
slice Class1  Group1 Disk1   Volume1 ACTIVE
slice Class1  Group1 Disk2   Volume1 COPY
```

If the all the displayed slices' STATUS fields are "ACTIVE", synchronization copying is complete.

If the synchronization copying is still in progress, "COPY" will be displayed in the STATUS field. Using the -e long option, you can check the progress of the synchronization copying. For details, see "D.6 sdxinfo - Display object configuration and status information."

## (2) Creating a single volume

The following example shows the procedures for creating a single volume using a physical disk named sda.

### 1) Registering a disk to class

Register the physical disk with a class. When the specified class does not exist, it will be created automatically.

Example: Registering physical disk sda to shared class "Class1", which is shared on node1 and node2, and assigning the name "Disk1".

```
# sdxdisk -M -c Class1 -a type=shared,scope=node1:node2 -d sda=Disk1:single
```

### 2) Creating a single volume

Create a volume within the single disk.

Example: Creating a volume of 1,000 blocks within single disk "Disk 1", and assigning a volume name "Volume1".

```
# sdxvolume -M -c Class1 -d Disk1 -v Volume1 -s 1000
```

## (3) Creating a large-capacity volume (using concatenation)

The following example shows the procedures for creating a volume by concatenating physical disks named sda and sdb.

### 1) Registering disks to class

Register the physical disks with a class. When the specified class does not exist, it will be created automatically.

Example: Registering physical disks sda and sdb with shared class "Class1", which is shared on node1 and node2, and naming those disks Disk1 and Disk2.

```
# sdxdisk -M -c Class1 -a type=shared,scope=node1:node2 -d sda=Disk1,sdb=Disk2
```

### 2) Connecting the disks to a concatenation group

Connect the disks to a concatenation group. When the specified concatenation group does not exist, it will be created automatically.

Example: Connecting "Disk1" and "Disk2" to concatenation group "Group1".

```
# sdxdisk -C -c Class1 -g Group1 -d Disk1,Disk2 -a type=concat
```

### 3) Creating a large-capacity volume

Create a volume within the concatenation group.

Example: Creating a volume of 1,000,000,000 blocks within concatenation group "Group1", and assigning a volume name "Volume1".

```
# sdxvolume -M -c Class1 -g Group1 -v Volume1 -s 1000000000 -a pslice=off
```

## (4) Creating a stripe volume

The following example shows the procedures for creating a volume by striping physical disks named sda and sdb.

### 1) Registering disks to class

Register the physical disks with a class. When the specified class does not exist, it will be created automatically.

Example: Registering physical disks sda and sdb with shared class "Class1", which is shared on node1 and node2, and naming those disks Disk1 and Disk2.

```
# sdxdisk -M -c Class1 -a type=shared,scope=node1:node2 -d sda=Disk1,sdb=Disk2
```

## 2) Connecting the disks to a stripe group

Connect the disks to a stripe group. When the stripe group does not exist, it will be created automatically.

Example: Connecting "Disk1" and "Disk2" to stripe group "Group1".

```
# sdxdisk -C -c Class1 -g Group1 -d Disk1,Disk2 -a type=stripe,width=32
```

## 3) Creating a stripe volume

Create a volume within the stripe disk.

Example: Creating a volume of 1,000 blocks within stripe group "Group1", and assigning a volume name "Volume1".

```
# sdxvolume -M -c Class1 -g Group1 -v Volume1 -s 1000 -a pslice=off
```

## (5) Creating a mirror volume (Combining striping and mirroring)

The following example shows the procedures for creating a volume by constructing stripe groups with physical disks sda and sdb and other physical disks sdc and sdd respectively and then mirroring the two stripe groups.

### 1) Registering disks to class

Register the physical disks with a class. When the specified class does not exist, it will be created automatically.

Example: Registering physical disks sda, sdb, sdc and sdd to shared class "Class1", which is shared on node1 and node2, and assigning the names "Disk1", "Disk2", "Disk3" and "Disk4" respectively.

```
# sdxdisk -M -c Class1 -a type=shared,scope=node1:node2 -d sda=Disk1,sdb=Disk2,sdc=Disk3,sdd=Disk4
```

### 2) Connecting the disks to a stripe group

Connect the disks to a stripe group. When the stripe group does not exist, it will be created automatically.

Example: Connecting "Disk1" and "Disk2" to stripe group "Group1".

```
# sdxdisk -C -c Class1 -g Group1 -d Disk1,Disk2 -a type=stripe,width=32
```

Connecting Disk3 and Disk4 to stripe group Group2.

```
# sdxdisk -C -c Class1 -g Group2 -d Disk3,Disk4 -a type=stripe,width=32
```

### 3) Connecting the stripe groups to a mirror group

Connect the stripe groups to a mirror group. When the specified mirror group does not exist, it will be created automatically.

Example: Connecting the stripe group "Group1" and "Group2" to mirror group "Group3".

```
# sdxgroup -C -c Class1 -h Group3 -l Group1,Group2 -a type=mirror
```

The "-a type=mirror" option is omissible.

### 4) Creating a mirror volume

Create a volume within the highest level mirror group.

Example: Creating a volume of 1,000 blocks within mirror group "Group3", and assigning a volume name "Volume1".



```
# sdxvolume -M -c Class1 -g Group3 -v Volume1 -s 1000 -a pslice=off
```

After returning from the command, synchronization copying will automatically begin.

### 5) Confirming the completion of procedure

Confirm that the synchronization copying is complete.

Example: Confirming synchronization copying of volume "Volume1" is complete.

```
# sdxinfo -S -o Volume1
OBJ   CLASS  GROUP  DISK   VOLUME STATUS
-----
slice Class1  Group3 Group1 Volume1 ACTIVE
slice Class1  Group3 Group2 Volume1 COPY
```

If the all the displayed slices' STATUS fields are "ACTIVE", synchronization copying is complete. If the synchronization copying is still in progress, "COPY" will be displayed in the STATUS field. Using the -e long option, you can check the progress of the synchronization copying.

For details, see "D.6 sdxinfo - Display object configuration and status information."

## D.17 Snapshot Creation Using Command

In this section, operation outline of snapshot creation using command is explained. Please use it as a reference when configuring the environment. For details, see the Command Reference.

### 1) Joining the proxy volume with master volume

Join the proxy volume with the master volume, and copy the master volume data to the proxy volume. Before joining them, stop the proxy volume.

**Example: Master volume Volume1 and proxy volume Volume2 will be joined.**

```
# sdxvolume -F -c Class1 -v Volume2 -e allnodes
# sdxproxy Join -c Class1 -m Volume1 -p Volume2
```

After returning from the command, synchronization copying will automatically be performed.



#### Information

-m option and -p option can indicate groups as well as volumes. When indicating a group, all volumes within the group will be copied.

### 2) Confirming the completion of copying

Confirm that the synchronization copying is complete.

**Example: Confirming synchronization copying of proxy volume (Volume2) is complete.**

```
# sdxinfo -S
OBJ   CLASS  GROUP  DISK  VOLUME  STATUS
-----
slice Class1  Group1 Disk1  Volume1 ACTIVE
slice Class1  Group1 Disk2  Volume1 ACTIVE
slice Class1  Group2 Disk3  Volume2 STOP
slice Class1  Group2 Disk4  Volume2 STOP
```

If all the displayed slices' STATUS fields are "STOP", synchronization copying is complete. If the synchronization copying is still in progress, "COPY" will be displayed in the STATUS field.

### 3) Creating a snapshot

In order to create a snapshot, part the proxy volume from the master volume after confirming that synchronization copying is complete.

```
# sdxproxy Part -c Class1 -p Volume2
```



#### Note

In order to create a snapshot properly, you should either stop the application, or secure integrity at the file system layer or database layer that is managing the data. For instance, if you are using the file system, use the umount(8) command to unmount the file system before separating the proxy volume from the master volume.

### 4) Creating a backup

In order to make a data backup from the snapshot you created, follow the procedures of whatever backup tool you are using.

If you are not using the snapshot function after creating the backup, cancel the master-proxy relationship.

### 5) Rejoining the proxy volume with master volume

In order to synchronize the proxy volume with the master volume again, remove the created snapshot, and rejoin the proxy volume with the master volume. Before rejoining them, stop the proxy volume.

```
# sdxvolume -F -c Class1 -v Volume2 -e allnodes  
# sdxproxy Rejoin -c Class1 -p Volume2
```

Synchronization copying will be performed as in procedure 1.

When creating more than one backup using the same proxy volume, repeat procedures 2 to 4.

### 6) Canceling the master-proxy relationship

Cancel the master-proxy relationship, and finish using the snapshot function.

```
# sdxproxy Break -c Class1 -p Volume2
```



# Appendix E GDS Messages

This section explains the messages for GDS drivers, daemons and commands.

## E.1 Web-Based Admin View Messages

For message numbers 0001 to 0099, see "Web-Based Admin View Operation Guide" for details.

## E.2 Driver Messages

Driver messages are output to a log file or console via the syslog interface.

### Variable Names

Italicized words in the messages are variable names, and the actual output will vary depending on the situation. The meaning and the format of the variable names used in the message explanations are described below.

Variable names	Descriptions
<i>v_devno</i>	Volume device number (hexadecimal number)
<i>v_maj</i>	Volume major number (decimal number)
<i>v_min</i>	Volume minor number (decimal number)
<i>p_devno</i>	Physical slice device number (hexadecimal number )
<i>p_maj</i>	Physical slice major number (decimal number)
<i>p_min</i>	Physical slice minor number (decimal number)
<i>pdevtree</i>	Physical disk device tree path name. ex) /dev/sda
<i>pdevinst</i>	Physical disk instance name. ex) sd77
<i>blkno dk</i>	I/O requested disk block number (decimal number)
<i>blkno sl</i>	I/O requested slice block number (decimal number)
<i>length</i>	I/O transfer requested block size (in bytes, decimal number)
<i>resid</i>	Residing block size (in bytes, decimal number)
<i>errno</i>	System call error number (decimal number)
<i>offlag</i>	Open flag (hexadecimal number)
<i>second</i>	Elapsed time (in seconds, decimal number)
<i>lbolt</i>	Elapsed time after system boot (in ticks, hexadecimal number)
<i>details</i>	Details

### Explanation

Messages output by the driver are shown below in the order of severity. There are three levels of severity.

Level of importance	Descriptions	facility.level
PANIC	This message is displayed when an event that will stop the system is detected.	kern.notice
WARNING	This message is output when an abnormal event is detected. It will not always result in an error.	kern.warning
NOTICE	This message is output to record driver operation and the like.	kern.notice

facility.level is the priority of a message passed from the GDS driver to syslogd(8). The output destinations of messages are defined in the /etc/syslog.conf configuration file of syslogd(8) and can be changed by modifying the definitions in /etc/syslog.conf. For details, see the description about syslog.conf(5) in this manual.

## E.2.1 Warning Messages

---

### **WARNING: sfdsk: read error on slice:**

**volume info :** devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** blk in disk=*blknodk*, blk in slice=*blknosl*  
                  length=*length*, resid=*resid*, errno=*errno*

#### **Explanation**

Read request sent to slice terminated abnormally.

#### **Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

### **WARNING: sfdsk: write error on slice:**

**volume info :** devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** blk in disk=*blknodk*, blk in slice=*blknosl*  
                  length=*length*, resid=*resid*, errno=*errno*

#### **Explanation**

Write request sent to slice terminated abnormally.

#### **Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk driver.

### **WARNING: sfdsk: read error on disk:**

**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** blk in disk=*blknodk*, blk in slice=*blknosl*  
                  length=*length*, resid=*resid*, errno=*errno*

#### **Explanation**

Read request sent to disk terminated abnormally.

#### **Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

### **WARNING: sfdsk: write error on disk:**

**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** blk in disk=*blknodk*, blk in slice=*blknosl*  
                  length=*length*, resid=*resid*, errno=*errno*

#### **Explanation**

Write request sent to disk terminated abnormally.

#### **Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: read and writeback error on slice:**

**volume info :** devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** blk in disk=*blknodk*, blk in slice=*blknosl*  
                  length=*length*, resid=*resid*, errno=*errno*

**Explanation**

Read request and writeback request sent to slice terminated abnormally. Writeback is a process to read data from other slices in the event of a read error.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: open error on slice:**

**volume info :** devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** blk in disk=*blknodk*, blk in slice=*blknosl*  
                  oflag=*oflag*, errno=*errno*

**Explanation**

Open request sent to slice returned abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: open error on disk:**

**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** oflag=*oflag*, errno=*errno*

**Explanation**

Open request sent to disk returned abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: close error on disk:**

**device info :** devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*pdevtree (pdevinst)*  
**error info :** oflag=*oflag*, errno=*errno*

**Explanation**

Close request sent to disk returned abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: NVURM read error on disk:**

**volume info : devno(maj,min)=v\_devno(v\_maj,v\_min)**

**device info : devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**error info : blk in disk=blknodk, blk in slice=blknos/  
length=length, resid=resid, errno=errno**

**Explanation**

NVURM read request sent to disk terminated abnormally.

NVURM is volume update area map information which is stored on the disk for just resynchronization.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: NVURM write error on disk:**

**volume info : devno(maj,min)=v\_devno(v\_maj,v\_min)**

**device info : devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**error info : blk in disk=blknodk, blk in slice=blknos/  
length=length, resid=resid, errno=errno**

**Explanation**

NVURM write request sent to disk terminated abnormally.

NVURM is volume update area map information which is stored on the disk for just resynchronization.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: volume status log write error on disk:**

**device info : devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**error info : blk in disk=blknodk, blk in slice=blknos/  
length=length, resid=resid, errno=errno**

**Explanation**

Write request of volume status log sent to disk terminated abnormally.

Volume status log records if the volume closed normally in the event of a system crash.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: sfdsk: failed to abort I/O requests on disk:**

**device info : devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**error info : errno=errno**

**Explanation**

The request to cancel I/O requests on the physical device indicated in device info (an ioctl request for the mphd or mplb driver) ended abnormally.

**Resolution**

Collect investigation material and contact your local customer support.



**WARNING: sfdsk: hook for device is not GDS entry.**

**Explanation**

Inconsistency was found in management information of disks registered with GDS, but it does not affect the operation.

**Resolution**

None

## **E.2.2 Information Messages**

---

**sfdsk: driver started up**

**Explanation**

The driver has been installed into the system.

**sfdsk: received shutdown request**

**Explanation**

A shutdown request from the sdxservd daemon has been received.

**sfdsk: volume status log updated successfully,details**

**Explanation**

Write request sent to the volume status log terminated normally. Volume status log records if the volume closed normally when a system failure occurred.

**NOTICE: sfdsk: I/O error on slice:**

**volume info: devno(maj,min)=v\_devno(v\_maj,v\_min)**

**device info: devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=device**

**error info: blk in disk=blknodk, blk in slice=blknosl  
length=length, resid=resid, errno=errno  
flags=b\_flags**

**Explanation**

I/O request sent to slice terminated abnormally.

**NOTICE: sfdsk: read error and writeback success on slice:**

**volume info : devno(maj,min)=v\_devno(v\_maj,v\_min)**

**device info : devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**error info : blk in disk=blknodk, blk in slice=blknosl  
length=length**

**Explanation**

Read request sent to slice terminated abnormally, but has been recovered by writeback process. Writeback is a process to read data from other slices in the event of a read error.

**NOTICE: sfdsk: copy timeout. no response from sdxservd daemon:  
volume info: devno(maj,min)=v\_devno(v\_maj,v\_min)**

**Explanation**

Synchronization copying was not performed since there was no response from the sdxservd daemon.

**NOTICE: sfdsk: processing has taken long time on disk:  
device info: devno(maj,min)=p\_devno(p\_maj, p\_min)  
devname=pdevtree (pdevinst)  
request info: lapsed seconds=second, start lbolt=lbolt  
details**

**Explanation**

While *second* seconds have passed since the I/O request indicated in *details* was issued for the physical device indicated in device info, the I/O request is not complete yet.

**NOTICE: sfdsk: processing has taken long time on volume:  
volume info: devno(maj,min)=v\_devno(v\_maj,v\_min)  
request info: lapsed seconds=second, start lbolt=lbolt  
details**

**Explanation**

While *second* seconds have passed since the I/O request indicated in *details* was issued for the volume indicated in volume info, the I/O request is not complete yet.

**NOTICE: sfdsk: abort I/O requests on disk:  
device info: devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**Explanation**

The process to cancel I/O requests on the physical disk indicated in device info has started.

**NOTICE: sfdsk: succeeded to abort I/O requests on disk:  
device info: devno(maj,min)=p\_devno(p\_maj,p\_min)  
devname=pdevtree (pdevinst)**

**Explanation**

The request to cancel I/O requests on the physical device indicated in device info (an ioctl request for the mphp or mplb driver) ended normally.

**NOTICE: sfdsk: HBA='adapter' varyio=enabled**

**Explanation**

Variable length IO is available for *adapter*, a SCSI host bus adapter, and SCSI disks connected to *adapter*.

**NOTICE: sfdsk: HBA='adapter' varyio=disabled**

**Explanation**

Variable length IO is unavailable for *adapter*, a SCSI host bus adapter, and SCSI disks connected to *adapter*.

**NOTICE: sfdsk: variable length IO is enabled**

**Explanation**

The sfdsk driver can make use of variable length IO.

**NOTICE: sfdsk: variable length IO is disabled**

**Explanation**

The sfdsk driver cannot make use of variable length IO.

## **E.2.3 Internal Error Messages**

---

**severity: sfdsk: internal error, details**

**Explanation**

Internal error occurred. *details* indicates the cause of the error. *severity* indicates the severity of the message.

**Resolution**

Collect investigation material and contact your local customer support.

## E.3 Daemon Messages

Daemon messages are output to the GDS log file.

`/var/opt/FJSVsdx/msglog/sdxserved.log`

Output format to log file is as follows.

*Mon Day HH:MM:SS SDX:daemon: severity: message*

*Mon* gives the month the message was output, *Day* the date, *HH* the hour, *MM* the minute, *SS* the second, *daemon* the daemon program name, *severity* the severity of the message, and *message* the message text.

Depending on the settings of the syslog, daemon message will also be output to syslog log file and console.

### Variable Names

Italicized words in the messages are variable names, and the actual output will vary depending on the situation. The meaning and the format of the variable names used in the message explanations are described below.

Variable names	Descriptions
<i>class</i>	Class name
<i>disk</i>	Disk name
<i>group</i>	Group name
<i>lgroup</i>	Lower level group name
<i>hgroup</i>	Higher level group name
<i>volume</i>	Volume name
<i>disk.volume</i>	Slice name
<i>object.volume</i>	Slice name
<i>proxy</i>	Proxy object name
<i>object</i>	Object name
<i>type</i>	Type attribute value
<i>status</i>	Object status
<i>device</i>	Physical disk name (sdX) X indicates the device identifier.
<i>pslice</i>	Physical slice name (sdXN) X indicates the device identifier and N indicates the slice number by a natural number. The device tree file name may also be used. ex) "sd@2,0:a,raw"
<i>psdevtree</i>	Device tree path name to a physical slice ex) /dev/sd@2,0:a
<i>v_devno</i>	Volume device number (hexadecimal number)
<i>v_maj</i>	Volume major number (decimal number)
<i>v_min</i>	Volume minor number (decimal number)
<i>p_devno</i>	Physical slice device number (hexadecimal number)
<i>p_maj</i>	Physical slice major number (decimal number)
<i>p_min</i>	Physical slice minor number (decimal number)
<i>blknodk</i>	I/O requested disk block number (decimal number)
<i>blknosl</i>	I/O requested slice block number (decimal number)
<i>length</i>	I/O transfer requested block size (in bytes, decimal number)
<i>resid</i>	Residing block size (in bytes, decimal number)
<i>errno</i>	System call error number (decimal number)
<i>sdxerrno</i>	GDS defined internal error number (decimal number)
<i>node</i>	Node identifier, or node name
<i>oflag</i>	Open flag (hexadecimal number)
<i>ioctlcmd</i>	The ioctl command name
<i>timeofday</i>	Character string indicating current time and date
<i>details</i>	Details
<i>sdxfunc</i>	GDS function name
<i>exitstat</i>	Value indicating command end status (decimal number)
<i>cmdline</i>	Character indicating command line

## Explanation

Messages output by the daemon are shown below in the order of its severity. There are four levels of severity.

Level of importance	Descriptions	facility.level
HALT	This message is output when an event that will halt all services provided by GDS is detected.	user.crit
ERROR	This message is output when an event that will halt some services is detected.	user.err
WARNING	This message is output when an abnormal event is detected. It will not always result in a service halt.	user.warning
INFO	This message is output to record the daemon operation and the like. Usually, it can be ignored.	user.info

facility.level is the priority of a message passed from the GDS driver to syslogd(8). The output destinations of messages are defined in the /etc/syslog.conf configuration file of syslogd(8) and can be changed by modifying the definitions in /etc/syslog.conf. For details, see the description about syslog.conf(5) in this manual.

### E.3.1 Halt Messages

---

#### **HALT: failed to create a new thread, errno=*errno***

##### Explanation

Function pthread\_create() terminated abnormally. Process cannot be continued. The daemon process will be exited.

##### Resolution

When error number information *errno* is insufficient for identifying the cause and recovery, collect investigation material and contact your local customer support.

#### **HALT: cannot open driver administrative file, errno=*errno***

##### Explanation

GDS driver(sfdsk) administrative file cannot be opened. Process cannot be continued. The daemon process will be exited.  
This message is output when files under /dev/sfdsk directory cannot be accessed.

##### Resolution

Collect investigation material, and contact your local customer support.

#### **HALT: startup failed, no enough address space**

##### Explanation

Startup failure due to unsuccessful memory allocation. Process cannot be continued. The daemon process will be exited.

##### Resolution

Confirm you have sufficient memory or swap area.

**HALT: failed to respawn *daemon* daemon, osfunc=*osfunc*, errno=*errno***

**Explanation**

The *daemon* terminated abnormally and failed to restart. The failure was caused by abnormal termination of OS *osfunc* function.  
The error number is *errno*. This message is output via syslog.

**Resolution**

When error number information is insufficient for identifying the cause, collect investigation material, and contact your local customer support.

**HALT: cannot start node-down recovery for remote node *node*, no enough space, osfunc=*osfunc*, errno=*errno***

**Explanation**

Could not recover the crashed remote node *node* due to unsuccessful memory allocation. Process cannot be continued. The daemon process will be exited.

**Resolution**

The OS *osfunc* function terminated abnormally, and error number is *errno*. Confirm you have sufficient memory or swap area, and recover.

---

### E.3.2 Error Messages

---

**ERROR: read error on *status* slice *object.volume*, class=*class*:  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
devname=*device*  
error info: blk in disk=*blknodk*, blk in slice=*blknosl*,  
length=*length*, resid=*resid*, errno=*errno***

**Explanation**

Read request sent to slice *object.volume* in *status* status terminated abnormally. Read request sent to volume configured by this slice, or to slice accessible in isolation was returned as error. You must attempt recovery as soon as possible since the application may cease normal operation.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and promptly recover the disk.

**ERROR: write error on *status* slice *object.volume*, class=*class*:  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
devname=*device*  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
length=*length*, resid=*resid*, errno=*errno***

**Explanation**

Write request sent to slice *object.volume* in *status* status terminated abnormally. Write request sent to volume configured by this slice, or to slice accessible in isolation was returned as error. You must attempt recovery as soon as possible since the application may cease normal operation.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and promptly recover the disk.

**ERROR: open error on *status slice object.volume*, class=*class*:**  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
devname=*device*  
error info:oflag=*oflag*, errno=*errno*

**Explanation**

Open request sent to slice *object.volume* in *status* status terminated abnormally. Open request sent to volume configured by this slice, or to slice accessible in isolation was returned as error.  
You must attempt recovery as soon as possible since the application may cease to operate normally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and promptly recover the disk.

**ERROR: read error and writeback error on *status slice object.volume*, class=*class*:**

volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
devname=*device*  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
length=*length*, resid=*resid*, errno=*errno*

**Explanation**

Read request and writeback request sent to slice *object.volume* in *status* status terminated abnormally. Read request sent to volume configured by this slice, or to slice accessible in isolation was returned as error.  
You must attempt recovery as soon as possible since the application may not operate normally. Writeback is a process to read data from other slices in the event of a read error.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and promptly recover the disk.

**ERROR: *volume*: closed down volume, class=*class***

**Explanation**

*volume* has closed down.

**Resolution**

Promptly attempt recovery by identifying the cause of failure by referring to GDS log message that was output immediately before the error.  
For information on recovery, see "F.1.3 Volume Status Abnormality."

**ERROR: *class*: cannot startup class, no valid configuration database, sdxerrno=*errno***

**Explanation**

Could not startup *class*, since valid *class* configuration database could not be found.  
This message is output when all disks (or the majority of disks) registered with *class* are unavailable.

**Resolution**

See "F.1.4 Class Status Abnormality."

**ERROR: *class*: cannot startup class, too few valid configuration database replicas, sdxerno=*errno***

**Explanation**

Could not startup *class*, due to insufficient number of valid databases. This message is output when many disks registered with *class* are unavailable.

**Resolution**

See "F.1.4 Class Status Abnormality."

**ERROR: *class*: closing down class, no valid configuration database**

**Explanation**

*class* was closed since no valid *class* configuration database was found. This message is output when all disks (or the majority of disks) registered with *class* are unavailable.

**Resolution**

See "F.1.4 Class Status Abnormality."

**ERROR: *class*: closing down class, too few valid configuration database replicas**

**Explanation**

*class* was closed due to insufficient number of valid database. This message is output when many disks registered with *class* are unavailable.

**Resolution**

See "F.1.4 Class Status Abnormality."

**ERROR: failed to send request message on node *node*, *details***

**Explanation**

Sending request message from *node* was unsuccessful.

**Resolution**

Collect investigation material and contact your local customer support.

**ERROR: timeout on receiving reply message from node *node*, *details***

**Explanation**

Timeout occurred while receiving a reply message from a remote node *node*.

**Resolution**

Collect investigation material and contact your local customer support.

**ERROR: rejected request message on remote node *node*, *details***

**Explanation**

Processing a request message on a remote node *node* was unsuccessful.

**Resolution**

Investigate the *node* message log and take necessary actions. If recovery is impossible, collect investigation material and contact your local customer support.



**ERROR: *class*: failed to start *type* volumes, *status* volume *volume* exists, *node*=*node***

**Explanation**

Starting volumes within the class *class* failed on the node *node* since the *status* volume *volume* exists. *type* is the *class* type attribute.

**Resolution**

*volume* is in abnormal status. First, you must recover normal status. For information on recovery, see "F.1.3 Volume Status Abnormality."

**ERROR: *class*: failed to start and standby *type* volumes, *status* volume *volume* exists, *node*=*node***

**Explanation**

Starting and putting on standby volumes within the class *class* failed on the node *node* since the *status* volume *volume* exists. *type* is the *class* type attribute.

**Resolution**

*volume* is in abnormal status. First, you must recover normal status. For information on recovery, see "F.1.3 Volume Status Abnormality."

**ERROR: *class*: failed to stop and standby *type* volumes, *status* volume *volume* exists, *node*=*node***

**Explanation**

Stopping and putting on standby volumes within the class *class* failed on the node *node* since the *status* volume *volume* exists. *type* is the *class* type attribute.

**Resolution**

*volume* is in abnormal status. First, you must recover normal status. For information on recovery, see "F.1.3 Volume Status Abnormality."

**ERROR: *class*: failed to stop *type* volumes, *status* volume *volume* exists, *node*=*node***

**Explanation**

Stopping volumes within the class *class* failed on the node *node* since the *status* volume *volume* exists. *type* is the *class* type attribute.

**Resolution**

Recover closed *class*. There may be multiple disk failures. Identify the cause based on object status, GDS log message, and syslog message.  
For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *class*: failed to start *type* volumes, class closed down, *node*=*node***

**Explanation**

Starting volumes failed since the class *class* has been closed down. *type* is the *class* type attribute.

**Resolution**

Recover closed *class*. There may be multiple disk failures. Identify the cause based on object status, GDS log message, and syslog message.  
For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *class*: failed to start and standby *type* volumes, class closed down, node=*node***

**Explanation**

Starting and putting on standby volumes failed. since the class *class* has been closed down. *type* is the *class* type attribute.

**Resolution**

Recover closed *class*. There may be multiple disk failures. Identify the cause based on object status, GDS log message, and syslog message.  
For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *class*: failed to stop and standby *type* volumes, class closed down, node=*node***

**Explanation**

Stopping and putting on standby volumes failed. since the class *class* has been closed down. *type* is the *class* type attribute.

**Resolution**

Recover closed *class*. There may be multiple disk failures. Identify the cause based on object status, GDS log message, and syslog message.  
For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *class*: failed to stop *type* volumes, class closed down, node=*node***

**Explanation**

Stopping volumes failed since the class *class* has been closed down. *type* is the *class* type attribute.

**Resolution**

Recover closed *class*. There may be multiple disk failures. Identify the cause based on object status, GDS log message, and syslog message.  
For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *class*: closing down class, cluster-wide lock failure, sdxerno=*sdxerno***

**Explanation**

Abnormal exclusive control between cluster system nodes occurred.  
Since the process cannot be continued, *class* will be closed.

**Resolution**

Collect investigation material and contact your local customer support.

**ERROR: *class*: cannot startup class, cluster-wide lock failure, sdxerno=*erno***

**Explanation**

Abnormal exclusive control between cluster system nodes occurred.  
Since the process cannot be continued, *class* could not be started.

**Resolution**

Collect investigation material and contact your local customer support.

**ERROR: *class*: closing down class, cluster communication failure, sdxerno=*sdxerno***

**Explanation**

Transmission failure between cluster system nodes occurred. Since the process cannot be continued, *class* will be closed.

**Resolution**

Collect investigation material and contact your local customer support.

**ERROR: *proxy*: failed to copy with OPC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

While performing copying between proxy volume *proxy* and master volume with the OPC function, an I/O error occurred, and the copying process failed. "source" specifies the original slice name, "target" the mirror slice name, and "class" to which the original slice and mirror slice belong.

**Resolution**

Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

**ERROR: *proxy*: failed to copy with EC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

While performing copying between proxy volume *proxy* and master volume with the EC function, an I/O error occurred, and the copying process failed. "source" specifies the original slice name, "target" the mirror slice name, and "class" to which the original slice and mirror slice belong.

**Resolution**

Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

**ERROR: *proxy*: failed to start OPC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

An error occurred between proxy volume *proxy* and the master volume when OPC was initiated. "source" indicates the copy source slice name, "target" indicates the copy destination slice name, and "class" indicates a class to which the copy source and destination slices belong.

**Resolution**

Identify the cause based on log messages of GDS, disk drivers, Fujitsu's ETERNUS storage system and such, and promptly recover the disk.

**ERROR: *proxy*: failed to stop OPC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

When stopping OPC between proxy volume *proxy* and master volume, an error occurred. "source" specifies the original slice name, "target" the mirror slice name, and "class" to which the original slice and mirror slice belong.

**Resolution**

Identify the cause by referring to the log messages for GDS, disk driver, Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

**ERROR: *proxy*: failed to start EC session, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

An error occurred between proxy volume *proxy* and the master volume when EC was initiated. "source" indicates the copy source slice name, "target" indicates the copy destination slice name, and "class" indicates a class to which the copy source and destination slices belong.

**Resolution**

Identify the cause based on log messages of GDS, disk drivers, Fujitsu's ETERNUS storage system and such, and promptly recover the disk.

**ERROR: *proxy*: failed to stop EC session, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

When stopping EC session between proxy volume *proxy* and master volume, an error occurred. "source" specifies the original slice name, "target" the mirror slice name, and "class" to which the original slice and mirror slice belong.

**Resolution**

Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that was output right before the occurrence of the error, and restore.

**ERROR: *proxy*: failed to suspend EC session, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

When temporarily suspending EC session between proxy volume *proxy* and master volume, an error occurred. "source" specifies the original slice name, "target" the mirror slice name, and "class" to which the original slice and mirror slice belong.

**Resolution**

Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

**ERROR: *proxy*: failed to resume EC session, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

When resuming EC session between proxy volume *proxy* and master volume, an error occurred. "source" specifies the original slice name, "target" the mirror slice name, and "class" to which the original slice and mirror slice belong.

**Resolution**

Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

**ERROR: *disk* is is bound to RAW device. disabled disk, class=*class***

**Explanation**

A disk registered with GDS was found that it was bound to a RAW device, and the disk was disabled.

**Resolution**

*disk* has been bound to a RAW device.  
Cancel the bind to a RAW device using the raw(8) command for the disk registered with GDS. You may not bind disks registered with GDS to RAW devices.

### E.3.3 Warning Messages

---

**WARNING: read error on *status slice object.volume*, class=*class*:**  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*device*  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
                  length=*length*, resid=*resid*, errno=*errno*

**Explanation**

Read request sent to slice *object.volume* in *status* status terminated abnormally. Slice with abnormality will be detached.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: write error on *status slice object.volume*, class=*class*:**  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*device*  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
                  length=*length*, resid=*resid*, errno=*errno*

**Explanation**

Write request sent to slice *object.volume* in *status* status terminated abnormally. Slice with abnormality will be detached.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: open error on *status slice object.volume*, class=*class*:**  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*device*  
error info:oflag=*oflag*, errno=*errno*

**Explanation**

Open request sent to slice *object.volume* in *status* status terminated abnormally. Slice with abnormality will be detached.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: read error and writeback error on *status* slice *object.volume*, class=*class*:**  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*device*  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
                  length=*length*, resid=*resid*, errno=*errno*

**Explanation**

Read request and writeback request sent to slice *object.volume* in *status* status terminated abnormally. Slice with abnormality will be detached.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: NVURM write error on disk *disk*, class=*class*:**  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
                  volume=*volume*, class=*class*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
                  devname=*device*  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
                  length=*length*, resid=*resid*, errno=*errno*

**Explanation**

NVURM write request sent to disk *disk* terminated abnormally. Although just resynchronization process on *volume* will be temporarily disabled, it will automatically attempt recovery.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *object.volume*: detached *status* slice by an I/O error, class=*class***

**Explanation**

Since an I/O error occurred on slice *object.volume* in *status* status, the slice was detached from the volume.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: open error on private slice *pslice*, oflag=*oflag*, errno=*errno***

**Explanation**

Open request sent to disk private slice *pslice* terminated abnormally. It will automatically search for a normal alternate disk and attempt recovery.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: read error on private slice *p\_devno(p\_maj,p\_min)*, offset=*blkno*, length=*length*, resid=*resid*, erro=*erro***

**Explanation**

Read request sent to disk private slice *p\_devno(p\_maj,p\_min)* terminated abnormally. It will automatically search for a normal alternate disk and attempt recovery.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: write error on private slice *p\_devno(p\_maj,p\_min)*, offset=*blkno*, length=*length*, resid=*resid*, erro=*erro***

**Explanation**

Write request sent to disk private slice *p\_devno(p\_maj,p\_min)* terminated abnormally. It will automatically search for a normal alternate disk and attempt recovery.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: close error on private slice *p\_devno(p\_maj,p\_min)*, erro=*erro***

**Explanation**

Close request sent to disk private slice *p\_devno(p\_maj,p\_min)* terminated abnormally. It will automatically search for a normal alternate disk and attempt recovery.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *sdxfunc: pslice*: open error, erro=*erro***

**Explanation**

Open request sent to physical slice *pslice* terminated abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *sdxfunc: pslice*: read error, erro=*erro***

**Explanation**

Read request sent to physical slice *pslice* terminated abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *sdxfunc: pslice*: write error, erro=*erro***

**Explanation**

Write request sent to physical slice *pslice* terminated abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *sdxfunc: pslice: ioctl error, request=ioctlcmd, errno=errno***

**Explanation**

The ioctl request sent to physical slice *pslice* terminated abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *sdxfunc: pslice: close error, errno=errno***

**Explanation**

Close request sent to physical slice *pslice* terminated abnormally.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *volume: failed to enable JRM, no available NVURM space, class=class***

**Explanation**

Resuming just resynchronization process on *volume* was unsuccessful due to insufficient number of normal disks.

**Resolution**

Check the disk status within *class*. Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *volume: failed to retrieve NVURM from disk disk, class=class***

**Explanation**

NVURM read request of *volume* from *disk* was unsuccessful. Just resynchronization will switch to copying of the entire disk. NVURM is a volume update area map information stored in the disk for just resynchronization.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and recover the disk.

**WARNING: *disk: device: disabled disk, class=class***

**Explanation**

*disk* is disabled since its data is invalid. *device* is the physical disk name corresponding to *disk*.

**Resolution**

Disk configuration may have been wrongly changed or disk data could be damaged. Check for improper connection change of I/O cables and disk swap. Also check for disk driver log message regarding the disk in question, and attempt recovery.

**WARNING: *volume: volume synchronization failed, class=class***

**Explanation**

Synchronization copying of *volume* was unsuccessful.

**Resolution**

Attempt recovery by identifying the cause of failure by referring to GDS log message and disk driver message that were output immediately before the error.



**WARNING: *volume*: volume just resynchronization failed, class=*class***

**Explanation**

Just resynchronization copying of *volume* was unsuccessful.

**Resolution**

Attempt recovery by identifying the cause of failure by referring to GDS log message and disk driver message that were output immediately before the error.

**WARNING: *class*: unknown class file found**

**Explanation**

Class file *class* which does not exist in class database file was found while booting the system.

**WARNING: invalid configuration database ID information, sdxerno=*sdxerno*, class=*class*:  
*psdevtree***

**Explanation**

Since the configuration database ID information was invalid, private slice *psdevtree* was not used as the configuration database for class *class*.

**Resolution**

Disk configuration may have been wrongly changed or disk data could be damaged. Check for improper connection change of I/O cables and disk swap. Also check for disk driver log message regarding the disk in question, and attempt recovery.

**WARNING: *class*: too few valid configuration database replicas**

**Explanation**

There are not sufficient valid configuration databases for *class*.  
This message is displayed when the majority of disks registered with *class* are unavailable.  
Leaving it as is may cause serious problems.

**Resolution**

For details, see "F.1.4 Class Status Abnormality."

**WARNING: cannot open message logging file, ermo=*ermo*  
*/var/opt/FJSVsdx/msglog/daemon.log***

**Explanation**

Opening GDS log file was unsuccessful. This message will be output via syslog. Although the message will not be output on GDS logfile, it does not affect other processes.

**Resolution**

Collect investigation material, and contact your local customer support regarding recovery.

**WARNING: cannot write message logging file, ermo=*ermo*  
*/var/opt/FJSVsdx/msglog/sdxservd.log***

**Explanation**

Writing to GDS log file was unsuccessful. This message will be output via syslog. Although the message will not be output on GDS logfile, it does not affect other processes.

**Resolution**

Collect investigation material and contact your local customer support.

**WARNING: failed to reply message to node *node*, details**

**Explanation**

Replying to remote node *node* was unsuccessful.

**Resolution**

Collect investigation material and contact your local customer support.

**WARNING: class: failed to change class resource status on remote node *node*, status=*new-status*, sdxerno=*sdxerno***

**Explanation**

Changing class resource status to *new-status* on a remote node *node* was unsuccessful.

**Resolution**

Collect investigation material and contact your local customer support.

**WARNING: sdxfunc: no enough address space, osfunc=*osfunc*, ermo=*ermo***

**Explanation**

The OS *osfunc* function returned an error. The error number is *ermo*.

**Resolution**

Confirm you have sufficient memory or swap area.

**WARNING: respawned *daemon* daemon successfully**

**Explanation**

Although *daemon* terminated abnormally, it completed normally by restarting. This message will be output via syslog.

**Resolution**

To identify the cause of *daemon* terminating abnormally, collect investigation material and contact your local customer support.

**WARNING: device: failed to restore VTOC on disk, sdxerno=*sdxerno***

**Explanation**

Recovering physical disk *device* format information was unsuccessful. Disk failure may have occurred.

**Resolution**

Use fdisk(8) command to recover format information.

**WARNING: volume: cannot copy to one or more slices in volume**

**Explanation**

Copying process could not be performed on some slices within *volume*.

**Resolution**

Execute synchronization copying as needed.

**WARNING: *device*: write error, *erno*=*erno***

**Explanation**

A write error occurred in physical disk *device* and recovering the partition table of *device* failed. A disk failure may have occurred.

**Resolution**

To recover the partition table of *device*, use the fdisk(8) command. If *device* is write-locked by the disk unit's copy function, no action is required.

**WARNING: *device*: read error, *erno*=*erno***

**Explanation**

A read error occurred in physical disk *device* and recovering the partition table of *device* failed. A disk failure may have occurred.

**Resolution**

To recover the partition table of *device*, use the fdisk(8) command.

**WARNING: *proxy*: too many EC/OPC sessions**

**Explanation**

The number of EC or OPC sessions within the physical disk (LU) or the disk array body has reached the upper limit of supported concurrent sessions. For this reason, a new EC or OPC session cannot be started. Copying is started using the soft copy function.

**Resolution**

To make copying by EC or OPC available, cancel the relationship between the proxy volume *proxy* and the master volume, wait until the running session ends and try this command again. Alternatively, according to need, cancel the running session using the sdxproxy Cancel command, the sdxproxy Break command, or [Operation]:[Proxy Operation]:[Break] and try this command again.

**WARNING: *proxy*: failed to start OPC, *source*=*disk.volume*, *target*=*disk.volume*, *class*=*class***

**Explanation**

An error occurred when OPC was started between the proxy volume *proxy* and the master volume. "source" is the copy source slice name, "target" is the copy destination slice name, and "class" is the name of the class to which the copy source slice and the copy destination slice belong. Copying is started between *proxy* and the master using the soft copy function.

**Resolution**

The cause of this error may derive from a faulty disk array unit. Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

**WARNING: *proxy*: failed to start EC session, *source*=*disk.volume*, *target*=*disk.volume*, *class*=*class***

**Explanation**

An error occurred when an EC session was started between the proxy volume *proxy* and the master volume. "source" is the copy source slice name, "target" is the copy destination slice name, and "class" is the name of the class to which the copy source slice and the copy destination slice belong. Copying is started between *proxy* and the master using the soft copy function.

**Resolution**

The cause of this error may derive from a faulty disk array unit. Identify the cause by referring to the log messages for GDS, disk driver, and Fujitsu Storage Systems ETERNUS that were output right before the occurrence of the error, and restore.

### E.3.4 Information Messages

---

**INFO: read error and writeback success on *status* slice *object.volume*, class=*class*:  
volume info:devno(maj,min)=*v\_devno(v\_maj,v\_min)*  
device info:devno(maj,min)=*p\_devno(p\_maj,p\_min)*  
devname=*device*,  
error info:blk in disk=*blknodk*, blk in slice=*blknosl*,  
length=*length***

#### Explanation

Read request sent to slice *object.volume* in status *status* was unsuccessful, but was restored by writeback process.  
Writeback is a process to read data from other slices in the event of a read error.

**INFO: *volume*: temporarily disabled JRM, class=*class***

#### Explanation

Although just resynchronization process on *volume* will be temporarily disabled, it will automatically attempt recovery.

**INFO: *disk*: failed to connect spare disk for disk *disk*, group=*group*, class=*class*,  
sdxerrno=*sdxerrno***

#### Explanation

Connecting spare disk *disk* to *group* instead of *disk* was unsuccessful.

**INFO: *disk*: failed to connect spare disk for group *lgroup*, the highest level  
group=*hgroup*, class=*class*, sdxerrno=*sdxerrno***

#### Explanation

The attempt to connect spare disk *disk* to the highest level group *hgroup* in place of group *lgroup* failed.

**INFO: *disk*: connected spare disk for disk *disk*, group=*group*, class=*class***

#### Explanation

Connected spare disk *disk* to *group* instead of *disk*.

**INFO: *disk*: connected spare disk for group *lgroup*, the highest level group=*hgroup*,  
class=*class***

#### Explanation

Spare disk *disk* was connected to the highest level group *hgroup* in place of group *lgroup*.

**INFO: *group*: free blocks are reduced in group, class=*class***

#### Explanation

Free blocks of *group* were reduced.

**INFO: *volume*: reallocated NVURM space and enabled JRM successfully, class=*class***

**Explanation**

Just resynchronization process of *volume* was resumed.

**INFO: *volume*: retrieved NVURM from disk *disk* successfully, class=*class***

**Explanation**

NVURM read request of *volume* sent from *disk* was successful. Just resynchronization process will be resumed.

NVURM is volume update area map information stored in the disk for just resynchronization.

**INFO: *volume*: no need to retrieve NVURM, *sdxerno*=*sdxerno*, class=*class***

**Explanation**

NVURM was not retrieved. Entire copying of volume will start.

**INFO: *disk: pslice*: failed to open physical special files exclusively, *erno*=*erno***

**Explanation**

Opening *disk* physical slice *pslice* exclusively was unsuccessful.

**INFO: *disk: device*: disk ID information is invalid, *sdxerno*=*sdxerno***

**Explanation**

Disk ID information of *disk* is invalid. *disk* will be automatically disabled.

**INFO: *disk*: enabled disk, class=*class***

**Explanation**

Disabled *disk* was enabled.

**INFO: *volume*: volume synchronization started, class=*class***

**Explanation**

Started synchronization copying process of *volume*.

**INFO: *volume*: volume just resynchronization started, class=*class***

**Explanation**

Started just resynchronization process of *volume*.

**INFO: *volume*: volume synchronization canceled, class=*class***

**Explanation**

Canceled synchronization copying of *volume*.

**INFO: *volume*: volume just resynchronization canceled, class=*class***

**Explanation**

Canceled just resynchronization process of *volume*.

**INFO: *volume*: volume synchronization completed successfully, class=*class***

**Explanation**

Synchronization copying process of *volume* completed successfully.

**INFO: *volume*: volume just resynchronization completed successfully, class=*class***

**Explanation**

Just resynchronization process of *volume* completed successfully.

**INFO: *object*: failed to update configuration database, class=*class***

**Explanation**

Updating configuration database was unsuccessful since all (or majority of) disks registered with *class* was unavailable. Usually, class will be closed following this message.

**INFO: *sdxservd* daemon started up**

**Explanation**

The *sdxservd* daemon started up. GDS will be started.

**INFO: stopped all services by shutdown request, *timeofday***

**Explanation**

All GDS services were stopped in response to a shutdown request.

**INFO: cannot open class database file, *errno=errno***

**Explanation**

Could not open class database file while booting the system. Will automatically attempt recovery.

**INFO: class database file corrupted**

**Explanation**

Corrupted class database file was detected while booting the system.  
Will automatically attempt recovery.

**INFO: *class*: cannot open class file, *errno=errno***

**Explanation**

Could not open class file *class* while booting the system. Will automatically attempt recovery.

**INFO: *class*: class file corrupted**

**Explanation**

Corrupted class file *class* was detected while booting the system. Will automatically attempt recovery.

**INFO: class database file updated successfully**

**Explanation**

Class database file was updated.

**INFO: *class*: class file updated successfully**

**Explanation**

Class file *class* was updated.

**INFO: cannot write class database file, *ermo=ermo***

**Explanation**

Could not write to class database file. Will automatically attempt recovery.

**INFO: *class*: cannot write class file, *ermo=ermo***

**Explanation**

Could not write to class file *class*. Will automatically attempt recovery.

**INFO: cannot check configuration database ID information, *sdxermo=sdxermo*, *class=class*:  
*psdextree***

**Explanation**

Opening or reading private slice *psdextree* was unsuccessful, and could not check configuration database ID information of *class*.

**INFO: cannot check configuration database, *sdxermo=sdxermo*, *class=class*:  
*psdextree***

**Explanation**

Opening or reading private slice *psdextree* was unsuccessful, and could not check configuration database of *class*.

**INFO: configuration database corrupted, *sdxermo=sdxermo*, *class=class*:  
*psdextree***

...

**Explanation**

Since there was a check-sum error in the configuration database, private slice *psdevtree...* was not used as the configuration database for class *class*.

**INFO: configuration database defeated, sdxerno=*sdxerno*, class=*class*:  
*psdextree***

...

**Explanation**

The configuration database for class *class* stored on private slice *psdevtree...* was determined to be invalid in a validity check.

**INFO: *class*: valid configuration database replicas exist on:  
*psdextree***

...

**Explanation**

The valid configuration database for class *class* was determined. *psdevtree...* is the private slice storing the valid configuration database.

**INFO: *class*: starting up class**

**Explanation**

*class* will be started.

**INFO: cannot update configuration database replica, sdxerno=*sdxerno*, class=*class*:  
*psdextree***

**Explanation**

Updating a replica of the configuration database for class *class* stored on private slice *psdevtree* failed.

**INFO: *class*: relocated configuration database replicas on:  
*psdextree***

...

**Explanation**

A replica of the configuration database for class *class* was relocated onto private slice *psdevtree...*

**INFO: *disk*: disconnected spare disk from group *group*, class=*class***

**Explanation**

Disconnected spare disk *disk* from *group*.

**INFO: *group*: free blocks are increased in group, class=*class***

**Explanation**

Number of free blocks in *group* increased.

**INFO: failed to create a new thread, ermo=*ermo***

**Explanation**

Function `pthread_create()` terminated abnormally.



**INFO: cannot open configuration parameter file *filename*, erro=*ermo***

**Explanation**

Opening configuration parameter *filename* was unsuccessful.

**INFO: cannot read configuration parameter file, erro=*ermo***

**Explanation**

Reading configuration parameter files was unsuccessful.

**INFO: received unexpected data from sfdsk driver and ignored**

**Explanation**

Unexpected data was received from sfdsk driver and was ignored.

**INFO: received unexpected event from sfdsk driver and ignored, *details***

**Explanation**

Unexpected event was received from sfdsk driver and was ignored.  
*details* displays the details about the event.

**INFO: *class*: class closed down, node=*node***

**Explanation**

*class* on *node* was closed.

**INFO: command executed:  
*cmdline***

**Explanation**

Command *cmdline* was executed.

**INFO: command exited, exit-status=*exitstat*:  
*cmdline***

**Explanation**

Processing *cmdline* is complete.

**INFO: trying to execute command:  
*cmdline***

**Explanation**

The *cmdline* command is about to be executed.

**INFO: failed to execute command:  
*cmdline***

**Explanation**

The *cmdline* command failed.

**INFO: *class*: changed class resource status on remote node *node*, *old-status* => *new-status***

**Explanation**

Class resource status on remote node *node* was changed from *old-status* to *new-status*.

**INFO: *class*: changed class resource status on current node *node*, *old-status* => *new-status***

**Explanation**

Class resource status on current node *node* was changed from *old-status* to *new-status*.

**INFO: *class*: started *type* volumes, *node=node***

**Explanation**

Starting volumes that belong to the class *class* was completed on the node *node*. *type* is the *class* type attribute.

**INFO: *class*: started and stood by *type* volumes, *node=node***

**Explanation**

Starting and putting on standby volumes that belong to the class *class* was completed on the node *node*. *type* is the *class* type attribute.

**INFO: *class*: stopped and stood by *type* volumes, *node=node***

**Explanation**

Stopping and putting on standby volumes that belong to the class *class* was completed on the node *node*. *type* is the *class* type attribute.

**INFO: *class*: stopped *type* volumes, *node=node***

**Explanation**

Stopping volumes that belong to the class *class* was completed on the node *node*. *type* is the *class* type attribute.

**INFO: cannot connect spare disk, cluster-wide lock failure, *class=class*, *sdxerrno=sdxerrno***

**Explanation**

Due to the occurrence of an abnormal exclusive control between cluster system nodes, the spare disk could not be connected.

**INFO: cannot connect spare disk, too few valid configuration database replicas, class=*class*, disk=*disk***

**Explanation**

Could not connect spare disk due to insufficient number of valid configuration databases.

**INFO: cannot connect spare disk, hot spare disabled, class=*class*, disk=*disk***

**Explanation**

Spare disk *disk* could not be connected since the hot spare is disabled.

**INFO: *class*: started class-down recovery for remote node *node***

**Explanation**

Closed class *class* on remote node *node* will be recovered.

**INFO: *class*: class-down recovery failed, already class-down on current node *node***

**Explanation**

Attempted recovery of closed class *class* on remote node *node*. Recovery was unsuccessful since the class was also closed on current node *node*.

**INFO: *class*: class-down recovery failed, sdxerno=*sdxerno***

**Explanation**

Recovering closed class *class* was unsuccessful.

**INFO: *class*: class-down recovery completed successfully**

**Explanation**

Recovering closed class *class* was successful.

**INFO: *class*: started node-down recovery for remote node *node***

**Explanation**

Started node-down recovery on remote node *node*.

**INFO: *class*: started shutdown recovery for remote node *node***

**Explanation**

Started shutdown recovery on remote node *node*.

**INFO: *class*: node-down recovery failed, already class-down on current node *node***

**Explanation**

Recovering node-down was unsuccessful since *class* was in closed status on current node.

**INFO: *class*: shutdown recovery failed, already class-down on current node *node***

**Explanation**

Recovering shutdown was unsuccessful since *class* was in closed status on current node.

**INFO: *class*: node-down recovery failed, *sdxerno*=*sdxerno***

**Explanation**

Recovering *class* from node-down was unsuccessful.

**INFO: *class*: shutdown recovery failed, *sdxerno*=*sdxerno***

**Explanation**

Recovering *class* from shutdown was unsuccessful.

**INFO: *class*: node-down recovery completed successfully**

**Explanation**

Recovering *class* from node-down completed successfully.

**INFO: *class*: shutdown recovery completed successfully**

**Explanation**

Recovering *class* from shutdown completed successfully.

**INFO: *object.volume*: failed to update slice error information, class closed down, class=*class***

**Explanation**

Updating error information of slice *object.volume* was unsuccessful due to *class* in closed status.

**INFO: *volume*: failed to disable JRM, class closed down, class=*class***

**Explanation**

Disabling just resynchronization process on *volume* was unsuccessful due to *class* in closed status.

**INFO: *object.volume*: failed to detach slice, class closed down, class=*class***

**Explanation**

Detaching slice *object.volume* was unsuccessful due to *class* in closed status.

**INFO: *volume*: failed to restart volume, class closed down, class=*class***

**Explanation**

Restarting *volume* was unsuccessful due to *class* in closed status.

**INFO: open error on *status* slice *object.volume*, class=*class***

**Explanation**

Open request sent to slice *object.volume* in *status* status terminated abnormally.

**INFO: *class*: trying to identify class master, *details***

**Explanation**

Trying to identify the class master for shared class *class*.

**INFO: *class*: identified class master, node=*node***

**Explanation**

The master for shared class *class* has been identified as node *node*.

**INFO: *class*: searching class master**

**Explanation**

Searching the master for shared class *class*.

**INFO: *class*: class master found, node=*node***

**Explanation**

The master for shared class *class* was found to be node *node*.

**INFO: *class*: class master not found**

**Explanation**

The master for shared class *class* cannot be found.

**INFO: *class*: got class master privilege**

**Explanation**

Trying to obtain master privilege for shared class *class*.

**INFO: *class*: broadcasted class master information to remote nodes**

**Explanation**

The master information for shared class *class* has been broadcast to the remote nodes.

**INFO: *class*: received confirmations of class master information from remote node *node***

**Explanation**

Received master confirmation for shared class *class* from remote node *node*.

**INFO: waiting for outstanding event operations, *details***

**Explanation**

Waiting for event operations in process.

**INFO: completed outstanding event operations**

**Explanation**

Event operations in process have been completed.

**INFO: *class*: trying to release class master privilege, *details***

**Explanation**

Trying to release master privilege for shared class *class*.

**INFO: *class*: released class master privilege**

**Explanation**

Master privilege for shared class *class* has been released.

**NFO: *proxy*: started to copy with OPC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

Copying between proxy volume *proxy* and master volume with the OPC function started. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: completed copying with OPC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

Copying between proxy volume *proxy* and master volume with the OPC function is completed. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: canceled copying with OPC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

Copying between proxy volume *proxy* and master volume with the OPC function was cancelled. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: EC session started, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

EC session between proxy volume *proxy* and master volume started. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: completed copying with EC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

Copying between proxy volume *proxy* and master volume with the EC function is completed. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: canceled copying with EC, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

Copying between proxy volume *proxy* and master volume with the EC function was cancelled. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: EC session stopped, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

EC session between proxy volume *proxy* and master volume with the EC function was stopped. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: EC session suspended, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

EC session between proxy volume *proxy* and master volume with the EC function has been temporarily suspended. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

**INFO: *proxy*: EC session resumed, source=*disk.volume*, target=*disk.volume*, class=*class***

**Explanation**

EC session between proxy volume *proxy* and master volume with the EC function was resumed. "source" specifies the original slice name, "target" the mirror slice name, and "class" the name of the class to which the original slice and mirror slice belong.

---

### **E.3.5 Internal Error Messages**

---

***daemon: severity: module: internal error, details***

**Explanation**

Internal error occurred. *details* gives the cause of the error, *daemon* gives the name of the daemon program, *severity* gives the severity of the message, and *module* the module name (usually the internal function name) where the event was detected.

**Resolution**

Collect investigation material and contact your local customer support.

## E.4 Command Messages

Command messages will be sent to standard output or standard error output. Output format is as follows.

*SDX:command: severity: message*

The *command* gives the command name, *severity* the severity of the message, and *message* the message text body.

Command message will also be output to the following log file.

*/var/opt/FJSVsdx/msglog/sdxservd.log*

Output format is as follows.

*Mon Day HH:MM:SS SDX:daemon: severity: message*

*Mon* gives the month the message was output, *Day* the date, *HH* the hour, *MM* the minute, *SS* the second, *daemon* the daemon program name, *severity* the severity of the message, and *message* the message text body.

### Variable Names

Italicized words in the messages are variable names, and the actual output will vary depending on the situation. The meaning and the format of the variable names used in the message explanation are described below.

Variable names	Descriptions
<i>class</i>	Class name
<i>disk</i>	Disk name
<i>group</i>	Group name
<i>lgroup</i>	Lower level group name
<i>hgroup</i>	Higher level group name
<i>volume</i>	Volume name
<i>disk.volume</i>	Slice name
<i>object.volume</i>	Slice name
<i>object</i>	Object name (or physical disk name)
<i>master</i>	Master object name
<i>proxy</i>	Proxy object name
<i>status</i>	Object status
<i>device</i>	Physical disk name (sdX) X indicates the device identifier.
<i>pslice</i>	Physical slice name (sdXN) X indicates the device identifier and N indicates the slice number by a natural number. The device tree file name may also be used. ex) "sd@2,0:a,raw"
<i>node</i>	Node identifier or node name.
<i>attribute</i>	Attribute name
<i>value</i>	Attribute value
<i>param</i>	Parameter name
<i>val</i>	Parameter value
<i>size</i>	Number of blocks (in sectors), (decimal number)
<i>option</i>	Command options.
<i>usage</i>	Syntax when using command.
<i>letter</i>	Characters
<i>details</i>	Details
<i>errno</i>	System call error number (decimal)
<i>sdxerrno</i>	Internal error number defined by GDS (decimal)
<i>string</i>	Other character strings



## Explanation

Messages output by command are shown below in the order of severity.  
There are four levels of severity.

Level of importance	Descriptions
ERROR	This message is output to indicate the cause when a command terminates abnormally.
WARNING	This message is output as a warning when an abnormal event is detected. It will not always result in an abnormal termination of the command.
INFO	This message is output to record the command operation.
TO FIX	This message is output to show you how to fix the command. Usually it outputs command syntax.

### E.4.1 Error Messages

---

#### **ERROR: connection timeout**

##### **Explanation**

Connection failed due to no response from sdxservd daemon.

##### **Resolution**

Confirm that sdxservd daemon process was started normally.

#### **ERROR: not privileged**

##### **Explanation**

Executed user is not a superuser.

##### **Resolution**

You need to have superuser authority for execution.

#### **ERROR: *option*: illegal option**

##### **Explanation**

*option* is illegal.

##### **Resolution**

See the Fix Messages described later, or the "Appendix D Command Reference " in the manual.

#### **ERROR: syntax error**

##### **Explanation**

Executed command has a syntax error.

##### **Resolution**

See the Fix Messages described later, or the "Appendix D Command Reference " in the manual.

#### **ERROR: *string*: name too long**

##### **Explanation**

Object name, file name or node identifier specified with *string* is too long.

##### **Resolution**

Specify the proper object name or node identifier.

**ERROR: *object*: name contains invalid character "*letter*"**

**Explanation**

Object name *object* contains an invalid character *letter*.

**Resolution**

You can only use alphanumeric characters, -(hyphen), and \_(underscore) when specifying an object name.

**ERROR: *object*: name starting with "-" or "\_" is invalid**

**Explanation**

Object name starting with "-" (hyphen) or "\_" (underscore) is invalid.

**Resolution**

Specify an object name that starts with an alphanumeric character.

**ERROR: *device*: illegal physical disk name**

**Explanation**

Physical disk name *device* is illegal.

**Resolution**

See the "Appendix D Command Reference" in the manual

**ERROR: *object*: object names must be unique within a class**

**Explanation**

Object names within a class must be unique.

**Resolution**

Specify a unique object name.

**ERROR: *module*: environment error, *details***

**Explanation**

The command cannot be executed due to an error in the environment. The error causes only a failure in command execution but has no effect on the system. Volumes can be accessed continuously. *module* is the name of the module on which this event was detected (usually an internal variable name). *details* indicates the error in detail.

**Resolution**

Collect investigation material and contact your local customer support.

**ERROR: *class* is a shadow class**

**Explanation**

The class *class* is a shadow class. The attempted operation is not supported for shadow classes.

**Resolution**

See "Appendix D Command Reference," and specify the proper command name and class name.

**ERROR: *attribute*: invalid attribute name**

**Explanation**

The same *attribute* has already been specified.

**Resolution**

See the "Appendix D Command Reference" in the manual

**ERROR: *attribute*: attribute name duplicated**

**Explanation**

The same attribute *attribute* has already been specified.

**Resolution**

You can only specify an attribute once.

**ERROR: *attribute=value*: invalid attribute value**

**Explanation**

Attribute value *value* is invalid.

**Resolution**

See the "Appendix D Command Reference" in the manual.

**ERROR: *node*: node name duplicated**

**Explanation**

The same node identifier has already been specified.

**Resolution**

You can only specify a node identifier once.

**ERROR: too many nodes in scope**

**Explanation**

There are too many nodes specified in scope.

**Resolution**

Up to 4 nodes can be specified in a scope.

**ERROR: *class*: cannot operate shared objects, cluster control facility not ready**

**Explanation**

Cannot operate shared object since cluster control is not operating.

**Resolution**

Start cluster control and try again.

**ERROR: current node must be specified as scope**

**Explanation**

Current node is not specified for shared class scope.

**Resolution**

Always specify a node set which includes current node as scope.

## **ERROR: multi-nodes must be specified for shared class**

### **Explanation**

Only one node is specified for shared class scope.

### **Resolution**

Multiple node identifiers must be specified in a scope.

## **ERROR: *node*: unknown node**

### **Explanation**

Node *node* does not exist.

### **Resolution**

First check the cluster system environment. Then, try again after changing the node identifier.

## **ERROR: *class*: not shared by the current node**

### **Explanation**

Scope of *class* does not include the current node.  
In the cluster system, *class* created in another node is specified.

### **Resolution**

Execute command on a node sharing *class*. Or, change the specified class name.

## **ERROR: *node*: current node not active in *class***

### **Explanation**

Current node *node* is a standby node in shared class *class*.

### **Resolution**

Execute the command from another active node.

## **ERROR: too many disks in *class***

### **Explanation**

There are too many disks registered with *class*.

### **Resolution**

Create a new class.

## **ERROR: *device*: not connected to *node***

### **Explanation**

The physical disk *device* has not been connected to the node *node*, or registration of the cluster resources is not completed properly.

### **Resolution**

Check the system configuration and the resource configuration. If *device* has not been connected to *node*, specify the proper physical device name and re-execute the command. If no disk resource exists, execute automatic resource registration of the cluster and re-execute the command.

**ERROR: *device*: not a shared device**

**Explanation**

Physical disk *device* is not a shared disk unit, or cluster shared disk definition may not be correct.

**Resolution**

Check the system configuration and execute the command.

**ERROR: *device*: already exists in *class***

**Explanation**

Physical disk *device* is already registered with *class*.

**Resolution**

Same physical disk cannot be registered with more than one class. Specify the correct physical disk.

**ERROR: *device*: already exists in another class**

**Explanation**

Physical disk *device* is already registered with another class. This class is not shared by current node.

**Resolution**

Same physical disk cannot be registered with more than one class. Specify the correct physical disk.

**ERROR: *device*: no such device**

**Explanation**

There is no physical disk *device*.

**Resolution**

Specify the correct physical disk name.

**ERROR: *device*: cannot open, *errno*=*errno***

**Explanation**

Cannot open physical disk *device*.

**Resolution**

Confirm that physical disk *device* is operating normally.

**ERROR: *device*: not a hard disk**

**Explanation**

Physical disk *device* is not a hard disk.

**Resolution**

GDS cannot manage devices other than hard disks.

**ERROR: *device*: disk driver *driver* not supported**

**Explanation**

Physical disk unit with the driver name *driver* is not supported.

**Resolution**

There is no resolution.

## **ERROR: *device*: illegal format**

### **Explanation**

Physical disk *device* format is illegal.

### **Resolution**

Check the format status.

## **ERROR: *object*: device busy**

### **Explanation**

Object *object* is in use.

### **Resolution**

Change to unused status and execute the command again.

## **ERROR: *object*: linked to a cluster service**

### **Explanation**

Object *object* is used in a cluster application.

### **Resolution**

Check the cluster environment settings.

## **ERROR: *device*: configuration information exists in private slice**

### **Explanation**

Configuration information exists in the private slice of physical disk *device*, and registering *device* failed. The following events a) to d) are the possible causes.

- a) *device* is already registered with another class.
- b) *device* is registered with a class in another domain.
- c) The entire SDX disk was copied or is being copied to *device* with the disk unit's copy function.
- d) After registering *device* with the class, it was not removed normally, and the private slice and configuration information remain existent in *device* illegally.

### **Resolution**

Check the system configuration and so on, and identify the cause among a) to d). If the cause is a) or b), there is no resolution. If it is c) or d), use the fdisk(8) command and remove the slice whose tag is 33 as needed.

## **ERROR: *device*: type cannot be specified except undef**

### **Explanation**

The class scope includes a node to which physical disk *device* is unconnected and the type can only be set to "undef" for registering *device* to the class.

### **Resolution**

See the "Appendix D Command Reference."

## **ERROR: *class*: *device*: type cannot be specified except undef**

### **Explanation**

Class *class* includes a switch group and the type can only be set to "undef" for registering physical disk *device* to *class*.

### **Resolution**

See the "Appendix D Command Reference."

**ERROR: *device*: write error, erro=*ermo***

**Explanation**

A write error occurred in physical disk *device*.

**Resolution**

A disk failure may have occurred. Identify the cause by referring to disk driver log messages and so on, and recover the disk. If *device* is a write-locked disk that is connected to a switch group, register writable disks first.

**ERROR: *device*: disk ID information differs from all disks in specified class**

**Explanation**

The specified class includes no disk with the same disk ID information (class name and disk name) as that stored in the private slice of physical disk *device*. For this reason, *device* cannot be registered with the specified class.

**Resolution**

Check the system configuration and so on, and specify a physical disk and class name properly.

**ERROR: *object*: name already assigned to another object**

**Explanation**

Object name *object* already exists within class. You cannot create multiple objects with the same name.

**Resolution**

Specify another name and execute the command again.

**ERROR: cannot connect to sdxcltd**

**Explanation**

Connecting to sdxcltd, a GDS cluster linked daemon was unsuccessful.

**Resolution**

When unable to identify the cause, collect investigation material and contact your local customer support.

**ERROR: physical device driver returned an error, erro=*ermo***

**Explanation**

Physical disk driver returned an error.

**Resolution**

Identify the cause by referring to error number and message log.

**ERROR: special file operation failed, erro=*ermo***

**Explanation**

Operating special file was unsuccessful.

**Resolution**

Identify the cause by referring to error number and GDS log message.

**ERROR: sfdsk driver returned an error, errno=*errno***

**Explanation**

GDS driver returned an error.

**Resolution**

Identify the cause by referring to error number, GDS log message, and syslog message.

**ERROR: sfdsk driver returned a temporary error, try again for a while**

**Explanation**

GDS driver returned a temporary error.

**Resolution**

Execute the command again after a while.

**ERROR: *class*: class closed down**

**Explanation**

*class* is closed. You cannot operate objects within a closed class.

**Resolution**

A number of disk failures may have occurred. Identify the cause by referring to object status, GDS log message, and syslog message. For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *class*: class closed down on another node**

**Explanation**

*class* is closed on another node. You cannot operate objects within a closed class.

**Resolution**

Recover closed *class*. A number of disk failures may have occurred. Identify the cause by referring to object status, GDS log message, and syslog message. For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *device*: invalid physical device, managed by *driver***

**Explanation**

Specified physical disk is managed by *driver*, and is therefore invalid.

**Resolution**

Confirm the I/O and cluster system configuration, and specify the correct physical disk name.

**ERROR: *device*: IDE disk cannot be specified as spare disk**

**Explanation**

The *device* is an IDE disk. It cannot be used as a spare disk.

**Resolution**

For spare disks, use disks other than IDE disks.



**ERROR: *class*: no such class**

**Explanation**

Cannot find *class*.

**Resolution**

Check GDS configuration.

**ERROR: *group*: not a group**

**Explanation**

*group* is not a group name.

**Resolution**

There is another object with the name *group* within class. Check configuration.

**ERROR: *group* is a lower level stripe group**

**Explanation**

Group *group* is a stripe group connected to another group. Disks and groups cannot be connected to or disconnected from *group*.

**Resolution**

Disconnect *group* from the higher level group as necessary.

**ERROR: *group*: connected to a lower level stripe group**

**Explanation**

Group *group* is connected to a lower level stripe group. Disks and groups cannot be connected to or disconnected from *group*.

**Resolution**

Disconnect the higher level stripe group of *group* from its higher level group as necessary.

**ERROR: too many groups in *class***

**Explanation**

Class *class* already has the maximum number of groups possible.  
A maximum of 100 groups can be created within, and a maximum of 1024 groups can be created within a local class or a shared class.

**Resolution**

Create a new class.

**ERROR: too many disks and/or groups are connected to *group***

**Explanation**

The maximum number of disks or lower level groups are already connected to group *group*.

**Resolution**

There is no resolution. With mirror groups, the maximum number of objects you can connect is eight. With stripe groups and concatenation groups, it is sixty-four.

**ERROR: *object*: smaller than stripe width of group *group***

**Explanation**

Since the available size of disk or lower level group indicated by *object* is smaller than the stripe width of group *group*, *object* cannot be connected to *group*.

**Resolution**

Either execute the command after indicating a disk or lower level group with sufficient size, or remove group *group* and adjust the stripe width to a smaller size.

**ERROR: *class*: three or more nodes exist in class scope**

**Explanation**

The scope of class *class* includes 3 or more nodes. Classes whose scope includes 3 or more nodes cannot include switch groups.

**Resolution**

There is no resolution.

**ERROR: *disk*: *type* disk exists in *class***

**Explanation**

Class *class* includes disk *disk* of the *type* type. The attempted operation is not supported for classes that include *type* type disks.

**Resolution**

There is no resolution.

**ERROR: *group*: group other than switch group exists in *class***

**Explanation**

Class *class* includes group *group* that is not a switch group, such as a mirror group, a concatenation group, or a stripe group. The attempted operation is not supported for classes that include such groups.

**Resolution**

There is no resolution.

**ERROR: active disk must be specified**

**Explanation**

When creating a switch group, specify the active disk in the `-a actdisk=disk` option.

**Resolution**

Specify the option properly, referring to "Appendix D Command Reference."

**ERROR: *disk*: active disk not specified by `-d` option**

**Explanation**

Active disk *disk* was not specified in the `-d` option.

**Resolution**

When creating a switch group, specify one of the disks specified in the `-d` option of the `sdxdisk -C` command as the active disk.

## **ERROR: too many disks specified**

### **Explanation**

Too many disks specified.

### **Resolution**

Specify a proper number of disks. For the number of disks that can be connected to a group, see " A.1.3 Number of Disks."

## **ERROR: *disk*: physical scope is not included in class scope**

### **Explanation**

The physical scope of disk *disk* is not included in the class scope and *disk* cannot be connected to a switch group.

### **Resolution**

Specify a proper disk name. For the physical scope of a disk, use the `sdxinfo -D` command and check the displayed DEVCONNECT field. For the class scope, use the `sdxinfo -C` command and check the displayed SCOPE field.

## **ERROR: *disk*: physical scope must be same as class scope**

### **Explanation**

The physical scope of the active disk matches the class scope and therefore the physical scope of an inactive disk must match the class scope. The physical scope of disk *disk* does not match the class scope and *disk* cannot be connected to the switch group as the inactive disk.

### **Resolution**

Specify a proper disk name. For the physical scope of a disk, use the `sdxinfo -D` command and check the displayed DEVCONNECT field. For the class scope, use the `sdxinfo -C` command and check the displayed SCOPE field.

## **ERROR: *disk*: physical scope must include only *node***

### **Explanation**

The physical scope of inactive disk *disk* can include only node *node*.

### **Resolution**

Specify a proper disk name, referring to "Appendix D Command Reference."

## **ERROR: *disk*: class scope is not included in physical scope**

### **Explanation**

The class scope is not included in the physical scope of disk *disk* and *disk* cannot be connected to a group other than a switch group.

### **Resolution**

Specify a proper disk name. For the physical scope of a disk, use the `sdxinfo -D` command and check the displayed DEVCONNECT field. For the class scope, use the `sdxinfo -C` command and check the displayed SCOPE field.

## **ERROR: *disk*: no such disk**

### **Explanation**

There is no *disk*.

### **Resolution**

Check GDS configuration.

**ERROR: *object*: already connected to *group***

**Explanation**

Disk or group indicated by *object*, is already connected to group *group*.

**Resolution**

Indicate the correct disk name or group name.

**ERROR: *disk* is a spare disk**

**Explanation**

*disk* is a spare disk. You cannot connect a spare disk to a group.

**Resolution**

Change the disk attributes to undefined, and execute the command again.

**ERROR: *object* not in *status* status**

**Explanation**

Object status is not *status* status.

**Resolution**

Confirm that object status is *status*, and execute the command again.

**ERROR: *object* too small**

**Explanation**

The size of *object* is too small.

**Resolution**

Check the necessary object size and specify a larger object.

**ERROR: another disk must be connected to *group***

**Explanation**

You must connect another disk to *group*.

**Resolution**

Connect another disk and execute the command again.

**ERROR: invalid physical slice number, *pslice\_num***

**Explanation**

Invalid physical slice number *pslice\_num* is specified.

**Resolution**

For physical slice number *pslice\_num*, specify an integer from 0 to 7, excluding 2.

**ERROR: *pslice* is private slice**

**Explanation**

Physical slice *pslice* is a private slice.

**Resolution**

Do not specify a private slice for physical slice number.

**ERROR: *pslice*: corresponding volume attributes must be specified**

**Explanation**

Volume attributes corresponding to physical slice *pslice* is not specified.

**Resolution**

See the "Appendix D Command Reference" in the manual

**ERROR: *object*: invalid size**

**Explanation**

Object size of *object* is invalid.

**Resolution**

See the "Appendix D Command Reference" in the manual

**ERROR: two or more single disks cannot be connected to a group**

**Explanation**

More than one single disk cannot be connected to a group.

**Resolution**

See the "Appendix D Command Reference" in the manual.

**ERROR: *disk*: single disk cannot be connected to existing group**

**Explanation**

Single disk *disk* cannot be connected to the existing group.

**Resolution**

See the "Appendix D Command Reference" in the manual.

**ERROR: two or more IDE disks cannot be connected to a group**

**Explanation**

More than one IDE disk cannot be connected to a group.

**Resolution**

When mirroring an IDE disk, use it with disks other than IDE disks.

**ERROR: too few valid configuration database replicas**

**Explanation**

There is not sufficient valid configuration database for class.

This message is displayed when the majority of disks registered with class are unavailable.

Leaving it as is may cause serious problems.

**Resolution**

For details, see "F.1.4 Class Status Abnormality."

**ERROR: *group*: no ENABLE disk in group**

**Explanation**

Group *group* does not have a disk in ENABLE status connected.

**Resolution**

See "F.1.2 Disk Status Abnormality" and recover disks connected to the group.

**ERROR: *msec*: invalid delay value**

**Explanation**

Delay time *msec* is invalid.

**Resolution**

See the "Appendix D Command Reference" in the manual.

**ERROR: *lgroup*: mirror group cannot be connected to another group**

**Explanation**

Since group *lgroup* is a mirror group, it cannot be connected to another group.

**Resolution**

There is no resolution.

**ERROR: *hgroup*: any group cannot be connected to a concatenation group**

**Explanation**

Since group *hgroup* is a concatenation group, any group cannot be connected to *hgroup*.

**Resolution**

There is no resolution.

**ERROR: *lgroup*: same type as higher level group *hgroup***

**Explanation**

Since the type attributes for group *lgroup* and higher level group *hgroup* are the same, *lgroup* cannot be connected to *hgroup*.

**Resolution**

There is no resolution.

**ERROR: *hgroup*: same name as a lower level group**

**Explanation**

For the higher level group name, name *hgoup*, which is the same as the lower level group was indicated.

**Resolution**

Different names must be indicated for higher level group and lower level group.

**ERROR: *hgroup*: any group cannot be connected to a switch group**

**Explanation**

Group *hgroup* is a switch group and any group cannot be connected to.

**Resolution**

There is no resolution.

**ERROR: *lgroup*: switch group cannot be connected to another group**

**Explanation**

Group *lgroup* is a switch group and cannot be connected to any group.

**Resolution**

There is no resolution.

**ERROR: *size*: must be more than zero**

**Explanation**

*size* must be a positive integer.

**Resolution**

Specify the correct size.

**ERROR: *size*: invalid size**

**Explanation**

The indicated size *size* is invalid.

**Resolution**

See the "Appendix D Command Reference" in the manual.

**ERROR: *group*: no such group**

**Explanation**

There is no *group*.

**Resolution**

Specify the correct group name.

**ERROR: *group*: not the highest level group**

**Explanation**

Group *group* is not the highest level group.

**Resolution**

Disconnect *group* from the higher level group as necessary.

**ERROR: too many volumes exist in *object***

**Explanation**

The maximum number of volumes, already exists in the object *object*, that is a class, a group, or a single disk.

**Resolution**

No resolution.

## **ERROR: *group*: volume with physical slice(s) cannot be created**

### **Explanation**

Although a physical slice cannot be created in group *group*, an attempt was made to create a volume with one or more physical slices.

### **Resolution**

Create a volume without any physical slice for *group* either by:

- Turning "off" the "Physical Slice" option in the Volume Configuration dialog box when creating the volume for *group* in GDS Management View
- Using the `-a pslice=off` option when creating the volume for *group* with the `sdxvolume -M` command

If *group* is a mirror group, connect one or more disks to the *group* in order to create a volume with a physical slice.

## **ERROR: *object*: no enough space**

### **Explanation**

There is not enough space in the group or single disk indicated by *object*.

### **Resolution**

Change the size you specify as needed.

## **ERROR: *disk*: not a single disk**

### **Explanation**

The disk *disk* is not a single disk.

### **Resolution**

Specify an appropriate disk.

## **ERROR: *disk*: *status* disk connected to *group***

### **Explanation**

*disk* in *status* status is connected to *group*.

### **Resolution**

Restore the status of *disk* and execute the command again.

## **ERROR: *status* disk exists in *group***

### **Explanation**

The *disk* in *status* status is connected to *group*, or it connected to the lower level group of *group*.

### **Resolution**

Confirm the disk status, and please execute a command again after canceling the *status* status if necessary.

## **ERROR: *disk*: no such disk in *group***

### **Explanation**

*disk* is not connected to *group*.

### **Resolution**

Specify the correct disk name or group name.



**ERROR: *disk: device: device busy***

**Explanation**

Disk *disk* (physical disk name is *device* ) is in use.

**Resolution**

Change to unused status, and execute the command again.

**ERROR: too many nodes specified**

**Explanation**

Too many nodes were specified.

**Resolution**

Specify the nodes included in the scope of the class.

**ERROR: *node: not in scope, class=class***

**Explanation**

Node *node* is not included in the scope of the class *class*.

**Resolution**

Confirm your GDS configuration, specify a node included in the scope of the class, and try the command again.

**ERROR: *volume: cannot start, class closed down, node=node***

**Explanation**

Starting volume *volume* was unsuccessful due to the class to which *volume* belongs in closed status.

**Resolution**

Recover the closed class. A number of disk failures may have occurred. Identify the cause by referring to object status, GDS log message, and syslog message. For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *volume: no such volume***

**Explanation**

There is no *volume*.

**Resolution**

Specify the correct volume name.

**ERROR: *object in status status***

**Explanation**

*object* is in *status* status.

**Resolution**

Confirm the object status, and cancel the status if necessary.

## **ERROR: some ACTIVE volumes exist in *class*, *node=node***

### **Explanation**

Volumes of class *class* are already active on node *node*, and volumes of *class* cannot be started or created on this node.

### **Resolution**

Create volumes of class *class* on node *node*. Alternatively, as needed, stop all the volumes within *class* on *node* and then start or create volumes on this node. If a cluster application to which *class* resources are registered is operating or is standby on *node*, before stopping volumes on *node*, stop the cluster application.

## **ERROR: *volume*: some ACTIVE volumes exist in *class*, *node=node***

### **Explanation**

Volumes of class *class* are already active on node *node*, and volume *volume* cannot be started on this node.

### **Resolution**

Stop all the volumes within class *class* on node *node* as needed, start volume *volume* on this node. If a cluster application to which *class* resources are registered is operating or is standby on *node*, before stopping volumes on *node*, stop the cluster application.

## **ERROR: *volume*: active disk not connected to *node***

### **Explanation**

The active disk of a switch group including switch volume *volume* is not connected to node *node* and the *volume* cannot be started on the *node*.

### **Resolution**

If an inactive disk is connected to a switch group including *volume*, use the `sdxattr -G` command and switch the inactive disk to the active disk in order to start *volume* on *node*.

## **ERROR: lock is set on volume *volume*, *node=node***

### **Explanation**

The "Lock volume" mode for volume *volume* on node *node* is turned on.

### **Resolution**

Turn off the "Lock volume" mode, or use the `-e unlock` option as necessary.

## **ERROR: *volume*: cannot stop, class closed down, *node=node***

### **Explanation**

Stopping volume *volume* was unsuccessful due to the class to which *volume* belongs being in closed status.

### **Resolution**

Recover the closed class. A number of disk failures may have occurred. Identify the cause by referring to object status, GDS log message, and syslog message. For information on recovery, see "F.1.4 Class Status Abnormality."

**ERROR: *object.volume*: *status* slice exists in *object***

**Explanation**

Slice *object.volume* in *status* status exists in *object*.

**Resolution**

Confirm the object status, and cancel the status if necessary.

**ERROR: *object* in *status* status, *node*=*node***

**Explanation**

*object* on *node* is in *status* status.

**Resolution**

Confirm the object status, and cancel the status if necessary.

**ERROR: *volume*: stripe type volume cannot be resized**

**Explanation**

*volume* cannot be resized since it is a volume created in a stripe group.

**Resolution**

There is no resolution.

**ERROR: *volume*: concat type volume cannot be resized**

**Explanation**

*volume* cannot be resized since it is a volume created in a concatenation group.

**Resolution**

There is no resolution.

**ERROR: *volume*: consists of multiple mirror slices**

**Explanation**

The volume *volume* cannot be resized since it is a mirror volume that consists of multiple mirror slices.

**Resolution**

Disconnect disks and lower level groups from the mirror group so that *volume* is composed of only one mirror slice, and try this command again.

**ERROR: spare disk connected for *disk***

**Explanation**

Spare disk is connected instead of *disk*.

**Resolution**

First recover the *disk* status.

**ERROR: *object*: the last disk or group in lower level group *group***

**Explanation**

Object *object* is the only disk or group connected to the lower level group *group*. You cannot disconnect *object* from *group*.

**Resolution**

Disconnect *group* from the higher level group as necessary.

**ERROR: *disk*: not connected to the end of concatenation group *group***

**Explanation**

Disk *disk* is not the disk that was concatenated last to the concatenation group *group*. You cannot disconnect *disk* from *group*.

**Resolution**

There is no resolution.

**ERROR: *disk*: disk space is assigned to volume *volume***

**Explanation**

Area on disk *disk* is allocated to volume *volume*. You cannot disconnect *disk* from the concatenation group.

**Resolution**

First remove *volume* if necessary.

**ERROR: *group*: inactive disk *disk* is connected**

**Explanation**

Group *group* has a connected inactive disk and physical disk swapping or disconnection cannot be performed for the active disk of the *group*.

**Resolution**

According to need, switch the active disk with the `sdxattr -G` command and perform physical disk swapping or disconnection for the previous active disk.

**ERROR: *lgroup*: no such group in *hgroup***

**Explanation**

Group *lgroup* is not connected to group *hgroup*.

**Resolution**

Indicate the correct group name.

**ERROR: one or more volumes exist in *group***

**Explanation**

One or more volumes exist in group *group*.

**Resolution**

First, remove the volume if necessary.

**ERROR: *disk* connected to *group***

**Explanation**

*disk* is connected to *group*.

**Resolution**

First, disconnect the disk from group if necessary.

**ERROR: *disk*: The last ENABLE disk in class cannot be removed**

**Explanation**

When there is a disk in SWAP or DISABLE status in class, you cannot remove the last disk in ENABLE status.

**Resolution**

First recover the disk in SWAP or DISABLE status. Or, you can register a new disk with class.

**ERROR: *disk*: cannot be removed to avoid class closing down**

**Explanation**

Disk *disk* stores the configuration database and removal of *disk* will result in class closure. For this reason, the disk cannot be removed.

**Resolution**

Recover or remove a failed disk in the class that includes *disk* and then remove *disk*.

**ERROR: one or more volumes exist in *disk***

**Explanation**

Volume exists in *disk*.

**Resolution**

First, remove the volume as needed.

**ERROR: one or more groups exist in *class***

**Explanation**

Group exists in *class*.

**Resolution**

First, remove the group as needed.

**ERROR: *disk*: *status* disk exists in *class***

**Explanation**

*disk* in *status* status exists in *class*.

**Resolution**

First, restore the *disk* in *status* status.

**ERROR: *volume*: *status* volume exists in *class*, *node*=*node***

**Explanation**

*volume* in *status* status on *node* exists in *class*.

**Resolution**

First, change the volume status as needed.

**ERROR: *disk*: no such disk**

**Explanation**

There is no disk *disk*.

**Resolution**

Specify the correct disk name.

**ERROR: *volume*: not associated with *object***

**Explanation**

You cannot specify slice by combining disk or group indicated by *object*, and volume indicated by *volume*.

**Resolution**

See the "Appendix D Command Reference" and indicate the correct disk name, group name or volume name.

**ERROR: *volume* is single volume**

**Explanation**

The *volume* is a single volume. Slices of single volumes cannot be detached.

**Resolution**

There is no resolution.

**ERROR: *volume*: not a mirror volume**

**Explanation**

Volume *volume* is not a mirror volume.

**Resolution**

There is no resolution.

**ERROR: *object*: not connected to the highest level group**

**Explanation**

Disk or group indicated by *object* is not connected to the highest-level group.

**Resolution**

See the "Appendix D Command Reference" and indicate the correct disk name or group name.

**ERROR: *disk*: The last ENABLE disk in class cannot be swapped**

**Explanation**

You cannot swap the last ENABLE status disk in class.

**Resolution**

Depending on the configuration, use a different method to avoid such situations. For example, you can register a new disk with class.

**ERROR: *disk: volume in status status***

**Explanation**

There is a volume related to disk in *status* status.

**Resolution**

First, recover the volume in *status* status.

**ERROR: *disk: the highest level group is not a mirror group***

**Explanation**

The highest level group of disk *disk* is not a mirror group. You cannot swap *disk*.

**Resolution**

There is no resolution.

**ERROR: *disk: cannot be swapped to avoid class closing down***

**Explanation**

Disk *disk* stores the configuration database and swapping *disk* will close down the class. Therefore, swapping the disk is prevented.

**Resolution**

Replace a failed disk included in the class to which *disk* is registered and swap *disk*.

**ERROR: *disk: device: device busy on node node***

**Explanation**

Disk *disk* (the physical disk name is *device*) is in use on *node*.

**Resolution**

Change the status to unused and retry the command.

**ERROR: *disk: device: cannot open, erro=ermo***

**Explanation**

Cannot open physical disk *device*.

**Resolution**

Confirm that physical disk *device* is operating normally.

**ERROR: *disk: device: not a hard disk***

**Explanation**

Physical disk *device* is not a hard disk.

**Resolution**

Specify a hard disk.

**ERROR: *disk: device: illegal format***

**Explanation**

Physical disk *device* format is illegal.

**Resolution**

Check the format of the physical disk.

**ERROR: *disk busy - /dev/rdisk/pslice***

**Explanation**

Physical disk *pslice* is in use.

**Resolution**

Change to unused status and execute the command again.

**ERROR: *disk: device: linked to a cluster service***

**Explanation**

Physical disk *device* is used in a cluster application.

**Resolution**

Check the settings for cluster environment.

**ERROR: *disk: device: not enough size***

**Explanation**

Physical disk size is too small.

**Resolution**

Specify a physical disk with sufficient size.

**ERROR: *disk: device: not connected to node***

**Explanation**

Physical disk *device* is not connected to *node*. Or, cluster shared disk definition may not be correct.

**Resolution**

Check the system configuration and execute the command.

**ERROR: *disk: device: invalid disk, managed by driver***

**Explanation**

Specified disk (corresponding physical disk name is *device*) is managed by *driver*, and is therefore invalid.

**Resolution**

Confirm the I/O and cluster system configuration, and specify the correct physical disk name.

**ERROR: *object.volume: status slice exists in group***

**Explanation**

Slice *object.volume* in *status* status already exists in group *group*.

**Resolution**

Check the object status and cancel *status* status if necessary.



**ERROR: *disk* : *device* : invalid disk on node *node*, managed by *driver***

**Explanation**

On node *node*, physical disk *device* of disk *disk* has been managed by driver *driver* and cannot be handled.

**Resolution**

Check I/O configurations and cluster system configurations, and specify the correct physical disk name.

**ERROR: *disk* : *device* : cannot open on node *node*, erro=*ermo***

**Explanation**

Physical disk *device* cannot be opened on node *node*.

**Resolution**

Check whether or not the physical disk *device* is operating normally.

**ERROR: *disk* : *device* : *node* : not a hard disk**

**Explanation**

Physical disk *device* on node *node* is not a hard disk.

**Resolution**

Specify a hard disk.

**ERROR: *disk* : *device* : *node* : illegal format**

**Explanation**

The format of physical disk *device* on node *node* is incorrect.

**Resolution**

Check the format.

**ERROR: *disk* busy on node *node* - /dev/rdisk/ *pslice***

**Explanation**

Physical disk *pslice* on node *node* is in use.

**Resolution**

Change the status to unused and retry the command.

**ERROR: *disk* : *device* : *node* : linked to a cluster service**

**Explanation**

Physical disk *device* on node *node* is being used by cluster applications.

**Resolution**

Check the cluster environment configurations.

**ERROR: *disk* : *device* : *node*: not enough size**

**Explanation**

The size of physical disk *device* on node *node* is too small.

**Resolution**

Replace it with a physical disk of sufficient size.

**ERROR: *disk* : *device* : read error, *ermo*=*ermo***

**Explanation**

A read error occurred on physical disk *device*.

**Resolution**

The possible cause is a disk failure. Identify the cause based on disk driver log messages and so on and restore the disk.

**ERROR: *disk* : *device* : *node*: read error, *ermo*=*ermo***

**Explanation**

A read error occurred on physical disk *device* on node *node*.

**Resolution**

The possible cause is a disk failure. Identify the cause based on disk driver log messages and so on and restore the disk.

**ERROR: *object*: read error, *ermo*=*ermo***

**Explanation**

Read error occurred on the disk indicated by *object*, or on a disk connected to group indicated by *object*, or on a disk connected to a lower level of the group indicated by *object*.

**Resolution**

Disk failure may have occurred. Identify the cause by referring to disk driver log message, and promptly recover the disk. Otherwise, indicate another disk or group that is normal.

**ERROR: *object*.*volume* : no such slice**

**Explanation**

You cannot specify slice by combining disk or group indicated by *object*, and volume indicated by *volume*.

**Resolution**

See the "Appendix D Command Reference" and indicate the correct disk name, group name or volume name.

**ERROR: *disk*: not active disk**

**Explanation**

Disk *disk* is not the active disk.

**Resolution**

Specify the active disk.

## **ERROR: *disk*: I/O error not occur**

### **Explanation**

An I/O error has not been occurred on the disk *disk*. There is no need to restore *disk* using the `sdxfix -D` command.

### **Resolution**

There is no resolution.

## **ERROR: *disk*: connected to switch group**

### **Explanation**

Disk *disk* is connected to a switch group and an I/O error status of the *disk* cannot be removed.

### **Resolution**

Make *disk* switchable with the `sdxswap -O` command or [Operation]:[Swap Physical Disk] in GDS Management View to remove an I/O error status of *disk*. After this procedure, swap the physical disks of *disk* according to need and then make the *disk* usable again with the `sdxswap -I` command or [Operation]:[Restore Physical Disk] in GDS Management View.

## **ERROR: *volume*: cannot restart to copy, cancel current interrupted copy operation by `sdxcopy` command with `-C` option**

### **Explanation**

Copying process within volume could not be resumed.

### **Resolution**

Cancel the interrupted copying process according to need and resume copying.

## **ERROR: *class*: class names must be unique within a domain**

### **Explanation**

Class with the same name as *class* already exists within the cluster domain.

### **Resolution**

Specify another class name.

## **ERROR: one or more disks not connected to *node***

### **Explanation**

Disks that are not connected to node *node* or for which disk resources are not created yet exist in the class.

### **Resolution**

Check the hardware configuration and the disk configuration within the class. If disk resources are not created yet, create those through the automatic resource registration. For the automatic resource registration, see "Appendix H Register shared disk units."

## **ERROR: *disk*: IDE disk cannot be specified as spare disk**

### **Explanation**

The *disk* is an IDE disk. It can not be used as a spare disk.

### **Resolution**

For spare disks, use disks other than IDE disks.

**ERROR: *object*: no such object**

**Explanation**

There is no *object*.

**Resolution**

Specify the correct object name.

**ERROR: *class*: shared objects information not yet available, try again for a while**

**Explanation**

Share object information is not available yet.

**Resolution**

Wait until cluster control is activated, and execute the command again.

**ERROR: *node*: node in stopped status**

**Explanation**

You cannot proceed with operation since *node* is in stopped status.

**Resolution**

Start the *node* and try again.

**ERROR: *node*: node in abnormal status**

**Explanation**

*node* is in an abnormal status and cannot proceed with operation.

**Resolution**

Confirm the normal startup of *node* and try again.

**ERROR: cluster communication failure**

**Explanation**

Cannot proceed with operation since communication failed with cluster.

**Resolution**

Check that the cluster system and GDS are operating normally.  
After recovering, try again.

**ERROR: cluster communication failure, *sdxerno*=*sdxerno***

**Explanation**

Cannot proceed with operation since communication failed with cluster.

**Resolution**

Check that the cluster system and GDS are operating normally.  
After recovering, try again.

**ERROR: cluster communication failure, remote-node=*node*, sdxerno=*sdxerno***

**Explanation**

Cluster communication with remote node *node* failed. Operation cannot be performed.

**Resolution**

Check that the cluster system and GDS are operating normally.  
After recovering, try again.

**ERROR: *disk*: not connected to any group**

**Explanation**

Disk *disk* is not connected to a group.

**Resolution**

See the "Appendix D Command Reference" in the manual and specify the correct disk name.

**ERROR: *volume*: *status* volume exists in *group***

**Explanation**

There is *volume* in *status* status in *group*.

**Resolution**

Recover volume status, and execute the command again.

**ERROR: *disk*: two or more disks connected to *group***

**Explanation**

*group* to which *disk* is connected has two or more disks connected.

**Resolution**

See the "Appendix D Command Reference" and complete preparation correctly. Then execute the command again.

**ERROR: illegal slice name**

**Explanation**

Slice name includes a "." (period).

**Resolution**

Specify the correct slice name.

**ERROR: *disk.volume* cannot be operated on the current node, take over by *sdxslice* command with *-T* option**

**Explanation**

Slice *disk.volume* cannot be operated on the current node.

**Resolution**

Take over the slice by executing *sdxslice -T* command.

### **ERROR: *volume*: physical slice attribute value is off**

#### **Explanation**

The physical slice attribute value of volume *volume* is "off." A slice in a volume without physical slices cannot be detached.

#### **Resolution**

Retry the command after turning the physical slice attribute of *volume* to be "on" according to need.

### **ERROR: *object*: device busy on node *node***

#### **Explanation**

*object* is in use on *node*.

#### **Resolution**

Change to unused status and execute the command again.

### **ERROR: *param*: invalid parameter name**

#### **Explanation**

Parameter name *param* is invalid.

#### **Resolution**

See the "Appendix D Command Reference" in the manual.

### **ERROR: *param =val*: invalid parameter value**

#### **Explanation**

Parameter value *val* is invalid

#### **Resolution**

See the "Appendix D Command Reference" in the manual.

### **ERROR: *param*: parameter name duplicated**

#### **Explanation**

The same parameter name *param* has already been specified.

#### **Resolution**

You can only use a parameter name once.

### **ERROR: *copy\_concurrency=val*: value more than or equal to the number of actually running copy operations must be specified**

#### **Explanation**

Value smaller than the number of copying currently in process has been specified for *copy\_concurrency* parameter.

#### **Resolution**

Set a value more than or equal to the number of copying currently in process for *copy\_concurrency* parameter.

**ERROR: mode=*string*: access mode duplicated**

**Explanation**

Multiple access modes have been specified.

**Resolution**

Specify only one access mode.

**ERROR: mode=*string*: invalid access mode**

**Explanation**

Access mode value *string* is invalid.

**Resolution**

See the "Appendix D Command Reference" in the manual.

**ERROR: volume: already started with different access mode, node=*node***

**Explanation**

`sdxvolume -M` command was executed with `-e mode=val` option on node *node*, but volume *volume* had already been started with a different access mode.

**Resolution**

Stop volume *volume* according to need and execute again.

**ERROR: volume: related to proxy volume *proxy***

**Explanation**

Volume *volume* is a master volume related to proxy volume *proxy*.

**Resolution**

Cancel the relationship of master volume *volume* and proxy volume *proxy* according to need and execute again.

**ERROR: volume: related to master volume *master***

**Explanation**

Volume *volume* is a proxy volume related to master volume *master*.

**Resolution**

Cancel the relationship of master volume *master* and proxy volume *volume* according to need and execute again.

**ERROR: volume: related to proxy volume *proxy* with EC**

**Explanation**

EC session exists between volume *volume* and proxy volume *proxy*.

**Resolution**

Cancel the relationship of master volume *volume* and proxy volume *proxy* according to need, and execute again.

**ERROR: *volume*: related to master volume *master* with EC**

**Explanation**

EC session exists between volume *volume* and master volume *master*.

**Resolution**

Use the `sdxproxyCancel` command and cancel the EC session between master volume *volume* and proxy volume *volume* according to need.

Alternatively, as needed, cancel the relationship of master volume *volume* and proxy volume *volume*, and execute again.

**ERROR: *proxy* : no parted proxy volume in proxy group**

**Explanation**

No parted proxy volume exists in proxy group *proxy*.

**Resolution**

According to need, part the proxy and retry the command.

**ERROR: *volume*: parted proxy volume**

**Explanation**

Volume *volume* is a proxy volume parted from master volume.

**Resolution**

Rejoin the proxy volume to the master volume or break the relationship between them according to need and try this command again.

**ERROR: *group*: related to proxy group *proxy***

**Explanation**

Group *group* is a master group related to proxy group *proxy*.

**Resolution**

Cancel the relationship between master group *group* and proxy group *proxy* according to need, and execute again.

**ERROR: *group*: related to master group *master***

**Explanation**

Group *group* is a proxy group related to master group *master*.

**Resolution**

Cancel the relationship between master group *master* and proxy group *group* according to need, and execute again.

**ERROR: *volume*: related to master or proxy volume**

**Explanation**

Volume *volume* is related to either master volume or proxy volume.

**Resolution**

Cancel the relationship between master and proxy according to need, and execute again.



**ERROR: *volume*: joined to master volume *master***

**Explanation**

Volume *volume* is a proxy volume joined to master volume *master*.

**Resolution**

Part volume *volume* from master volume *master*, or cancel the relationship with master volume according to need, and execute again.

**ERROR: *volume*: copying from master volume *master***

**Explanation**

Data is being copied from master volume *master* to volume *volume*.

**Resolution**

After the copying process is complete, execute again.

**ERROR: *volume*: copying from proxy volume *proxy***

**Explanation**

Data is being copied from proxy volume *proxy* to volume *volume*.

**Resolution**

After the copying process is complete, execute again.

**ERROR: *object*: not volume nor group**

**Explanation**

Object *object* is neither a volume or a group.

**Resolution**

Check the GDS configuration and specify the correct volume name or group name. Execute again.

**ERROR: different types of objects, *master=master*, *proxy=proxy***

**Explanation**

Different types of objects were specified as *master* and *proxy*.

**Resolution**

Specify a pair of volumes or groups as *master* and *proxy*.

**ERROR: *object*: same name as master**

**Explanation**

The object name specified for proxy is the same as the master's name, *object*.

**Resolution**

Different object names must be specified for *master* and *proxy*.

**ERROR: *group*: not a mirror group**

**Explanation**

Group *group* is not a mirror group.

**Resolution**

There is no resolution.

**ERROR: no volume exists in *group***

**Explanation**

There are no volumes in group *group*.

**Resolution**

Use `sdxvolume -M` command to create a volume within group *group*, and execute again.

**ERROR: too many proxy volumes are related to *master***

**Explanation**

There are too many proxy volumes related to master object *master*.

**Resolution**

Check the GDS configuration. For details about the number of proxy volumes, see "A.1.8 Number of Proxy Volumes."

**ERROR: *master*: corresponding proxy volume name must be specified**

**Explanation**

There is no corresponding proxy volume name specified for master volume *master*.

**Resolution**

See the "D.12 `sdxproxy` – Proxy object operations."

**ERROR: *proxy*: proxy volume name duplicated**

**Explanation**

The same proxy volume name *proxy* is specified for more than one master volume.

**Resolution**

Specify a different name, and execute again.

**ERROR: *volume*: no such volume in *group***

**Explanation**

Volume *volume* does not exist.

**Resolution**

Specify the correct volume name or group name, and execute again.

## **ERROR: *object*: object name duplicated**

### **Explanation**

The duplicate object name *object* was specified, or the group to which the volume *object* belongs and the volume *object* itself were specified at the same time.

### **Resolution**

You can specify an object only once.

## **ERROR: *proxy*: already parted**

### **Explanation**

Proxy volume *proxy* is already parted.

### **Resolution**

Nothing needs to be done.

## **ERROR: one point copy not available**

### **Explanation**

Instant snapshot cannot be created, since OPC (One Point Copy) function is not available. Possible causes are as follows.

- a. OPC/ROPC settings of the disk array unit are incorrect.
- b. When disks that constitute the master or the proxy were registered with a class, OPC/ROPC of the disk array unit was not properly set.
- c. When disks that constitute the master or the proxy were registered with a class, there was no GDS Snapshot installation.
- d. After installing GDS Snapshot, the node was not rebooted.
- e. Some EC/REC sessions exist between the master and another proxy.
- f. The copy destination volumes are in mirroring configuration.
- g. A lower level group is connected to a group to which master volumes or proxy volumes belong.
- h. The disk array unit to which the master or the proxy belongs does not support the OPC function.
- i. There is an error in the disk array unit or Fibre Channel paths.

### **Resolution**

Check disk array unit settings and GDS object configurations and identify the cause. According to the cause, take one of the following actions.

- When "a" is relevant, set OPC/ROPC of the disk array unit properly, remove disks that constitute the master and the proxy from the class once and re-register them.
- When "b" or "c" is relevant, among disks that constitute the master and the proxy, remove those that were registered with a class before setting OPC/ROPC or installing GDS Snapshot from the class once and re-register them.
- When "d" is relevant, reboot the node, remove disks that constitute the master and the proxy from the class once and re-register them.
- When "e" is relevant:
  - Part the proxy with EC/REC sessions from the relevant master and then cancel the EC/REC sessions with the `sdxproxy Cancel` command.
  - Break the relationship between the relevant master and the proxy with EC/REC sessions.
- When "f", "g" or "h" is relevant, select other objects, or change the object configuration, as needed.
- When "i" is relevant, recover the disk array unit or the Fibre Channel path from the error.

## **ERROR: *proxy*: already joined**

### **Explanation**

Proxy object *proxy* is already joined.

### **Resolution**

Nothing needs to be done.

**ERROR: *proxy*: not joined to master**

**Explanation**

Proxy volume *proxy* is not joined to master volume.

**Resolution**

Join the proxy volume *proxy* to master volume according to need, and execute again.

**ERROR: *proxy*: no such proxy object**

**Explanation**

Proxy object *proxy* cannot be found.

**Resolution**

Specify the correct proxy object name.

**ERROR: *master*: no such master object**

**Explanation**

Master object *master* cannot be found.

**Resolution**

Specify the correct master object name.

**ERROR: *volume*: exists in proxy group**

**Explanation**

Volume *volume* is a proxy volume in proxy group.

**Resolution**

Specify a proxy group and execute again.

**ERROR: *group*: not a proxy group**

**Explanation**

*group* is not a proxy group name.

**Resolution**

Check the GDS configuration, and specify a proxy group name.

**ERROR: *volume*: copying with EC**

**Explanation**

Volume *volume* is in the process of EC copy.

**Resolution**

After the copying process is complete, execute again.

**ERROR: *volume*: copying with OPC**

**Explanation**

Volume *volume* is in the process of OPC copy.

**Resolution**

After the copying process is complete, execute again.

## **ERROR: *volume*: related to same master volume as proxy *proxy*, master=*master***

### **Explanation**

Volume *volume* is related to the same master volume *master* as proxy volume *proxy*.

### **Resolution**

Specify the correct volume name. For details, see "Appendix D Command Reference."

## **ERROR: master and proxy exist in same group *group***

### **Explanation**

The specified master volume and proxy volume exist in the same group *group*.

### **Resolution**

Specify volumes that exist in different groups.

## **ERROR: *proxy*: joined to master with EC, rejoin them by soft copy and try again**

### **Explanation**

Slices cannot be swapped since EC session in process between proxy *proxy* and master.

### **Resolution**

Cancel the EC session by either of the following operations according to need and try this command again.

- Part the master and the proxy once and rejoin them with the `sdxproxy Rejoin -e softcopy` command.
- Break the relationship between the master and the proxy once and rejoin them with the `sdxproxy Join -e softcopy` command.

## **ERROR: *proxy*: failed to start soft copy**

### **Explanation**

An error occurred when synchronization copying with the soft copy function was started between the proxy volume *proxy* and the master volume.

### **Resolution**

Collect investigation material and contact your local customer support.

## **ERROR: OPC not available**

### **Explanation**

The OPC function is unavailable. Possible causes are as follows.

- a. OPC/ROPC settings of the disk array unit are incorrect.
- b. When disks that constitute the master or the proxy were registered with a class, OPC/ROPC of the disk array unit was not properly set.
- c. When disks that constitute the master or the proxy were registered with a class, there was no GDS Snapshot installation.
- d. After installing GDS Snapshot, the node was not rebooted.
- e. Some EC/REC sessions exist between the master and another proxy.
- f. The copy destination volumes are in mirroring configuration.
- g. A lower level group is connected to a group to which master volumes or proxy volumes belong.
- h. The disk array unit to which the master or the proxy belongs does not support the OPC function.
- i. There is an error in the disk array unit or Fibre Channel paths.

### **Resolution**

Check disk array unit settings and GDS object configurations and identify the cause. According to the cause, take one of the following actions.

- When "a" is relevant, set OPC/ROPC of the disk array unit properly, remove disks that constitute the master and the proxy from the class once and re-register them.
- When "b" or "c" is relevant, among disks that constitute the master and the proxy, remove those that were registered with a class before setting OPC/ROPC or installing GDS Snapshot from the

- class once and re-register them.
- When "d" is relevant, reboot the node, remove disks that constitute the master and the proxy from the class once and re-register them.
- When "e" is relevant:
  - Part the proxy with EC/REC sessions from the relevant master and then cancel the EC/REC sessions with the `sdxproxy Cancel` command, or
  - Break the relationship between the relevant master and the proxy with EC/REC sessions.
- When "f", "g" or "h" is relevant, select other objects, or change the object configuration, as needed.
- When "i" is relevant, recover the disk array unit or the Fibre Channel path from the error.

## **ERROR: EC not available**

### **Explanation**

The EC function is unavailable. Possible causes are as follows.

- a. EC/REC settings of the disk array unit are incorrect.
- b. When disks that constitute the master or the proxy were registered with a class, EC/REC of the disk array unit was not properly set.
- c. When disks that constitute the master or the proxy were registered with a class, there was no GDS Snapshot installation.
- d. After installing GDS Snapshot, the node was not rebooted.
- e. Multiple proxy volumes are related to a master volume and the number of proxy volumes with EC sessions has reached the upper limit (16 volumes).
- f. The copy destination volumes are in mirroring configuration.
- g. A lower level group is connected to a group to which master volumes or proxy volumes belong.

### **Resolution**

Check disk array unit settings and GDS object configurations and identify the cause. According to the cause, take one of the following actions.

- When "a" is relevant, set EC/REC of the disk array unit properly, remove disks that constitute the master and the proxy from the class once and re-register them.
- When "b" or "c" is relevant, among disks that constitute the master and the proxy, remove those that were registered with a class before setting EC/REC or installing GDS Snapshot from the class once and re-register them.
- When "d" is relevant, reboot the node, remove disks that constitute the master and the proxy from the class once and re-register them.
- When "e" is relevant:
  - Part the proxy with EC/REC sessions from the relevant master and then cancel the EC/REC sessions with the `sdxproxy Cancel` command
  - Break the relationship between the relevant master and the proxy with EC/REC sessions.
- When "f" or "g" is relevant, select other objects, or change the object configuration, as needed.

## **ERROR: proxy: too many EC/OPC sessions**

### **Explanation**

The number of EC or OPC sessions within the physical disk (LU) or the disk array body has reached the upper limit of supported concurrent sessions. For this reason, a new EC or OPC session cannot be started.

### **Resolution**

To make copying by EC or OPC available, wait until the running session ends, and try this command again. Alternatively, according to need, cancel the running session using the `sdxproxy Cancel` command, the `sdxproxy Break` command, or `[Operation]:[Proxy Operation]:[Break]` and try this command again.

## **ERROR: proxy: offset is different from master volume *master***

### **Explanation**

Master volume *master* and proxy volume *proxy* have different top block (sector) numbers. The top block number is not a physical block number that indicates the offset on a physical disk, but is a logical block number that indicates the offset within a group (or a single disk) to which the volume belongs. The layout (offsets and sizes) of volumes within a master group and a proxy group must be consistent.

### **Resolution**

For the proxy group, choose a group in which the layout of volumes are consistent with those of the

master group. For the top block (sector) numbers and the sizes of volumes, use the `sdxinfo` command and check the `1STBLK` field and the `BLOCKS` field of the displayed volume information.

**ERROR: *proxy*: number of volumes is different from master group *master***

**Explanation**

Master group *master* and proxy group *proxy* include different numbers of volumes. The layout (offsets and sizes) of volumes within a master group and a proxy group must be consistent.

**Resolution**

For the proxy group, choose a group in which the layout of volumes are consistent with those of the master group.

**ERROR: *class*: not a shadow class**

**Explanation**

The class *class* is not a shadow class.

**Resolution**

See "Appendix D Command Reference," and specify the proper command name and class name.

**ERROR: *device*: no configuration information**

**Explanation**

Since no configuration information resides in the private slice on the physical disk *device* or no private slice exists on *device*, *device* cannot be registered with a shadow class. The cause of this error may be one of the following.

- a) *device* is not registered with a class in another domain yet.
- b) *device* is already registered with a class in another domain, but has not been enabled.

**Resolution**

Check on the system configuration and so on, and identify which cause among a) and b) applies. If a) applies, see "Appendix D Command Reference" and specify the proper command name and physical disk name. If b) applies, restore *device* in another domain.

**ERROR: *device*: registered with illegal class in another domain**

**Explanation**

While the physical disk *device* is already registered with a different class in another domain, registering it with the same shadow class was attempted.

**Resolution**

You should register a disk registered with the same class in another domain with one shadow class.

**ERROR: *device*: *disk*: not same as disk name *diskname* in another domain**

**Explanation**

The physical disk *device* is already registered as *diskname* with a class in another domain. It cannot be registered with a shadow class as *disk* that is another disk name.

**Resolution**

Specify the same disk name *diskname* as in another domain.

### **ERROR: *device*: private slice size not same as another disk in *class***

#### **Explanation**

Since the physical disk *device* has the private slice that is unequal in size compared to the private slices on other disks registered with the shadow class *class*, it cannot be registered with *class*.

#### **Resolution**

Check on the system configuration and so on, and specify the proper disk name and shadow class name.

### **ERROR: *device* is bound to RAW device.**

#### **Explanation**

The device to be registered has been bound to a RAW device.  
With GDS, you cannot register disks bound to a RAW device.

#### **Resolution**

Register a disk not bound to a RAW device. Alternatively, cancel the bind to a RAW device using the `raw(8)` command and then register.

### **ERROR: no license**

#### **Explanation**

The environment has an error and the command cannot be executed. The command is inexecutable but there is no other influence on the system. Volumes can be continuously accessed without problems.

#### **Resolution**

Collect investigation material and contact your local customer support.

### **ERROR: *class*: some node trying to get class master privilege**

#### **Explanation**

The local node or some other node is trying to obtain master privileges of shared class *class*.

#### **Resolution**

If necessary, try the command again after a while.

### **ERROR: *class*: class master not found**

#### **Explanation**

The class master of shared class *class* is not found and no operation is possible for *class*.

#### **Resolution**

If necessary, try the command again after a while.



## E.4.2 Warning Messages

---

### **WARNING: spare disk *disk* too small**

#### **Explanation**

The size of *disk* is too small and may not function as a spare disk.

#### **Resolution**

Specify a larger disk, and execute the command again.

### **WARNING: device: write error, *erno*=*erno***

#### **Explanation**

A write error occurred in physical disk *device*.

#### **Resolution**

If *device* is a write-locked disk that is connected to a switch group, no action is required. Otherwise, remove *device* from the class and check the status of *device*.

### **WARNING: group: free blocks are reduced**

#### **Explanation**

Free blocks on *group* was reduced.

#### **Resolution**

You may not be able to create a volume with sufficient capacity. Execute commands as needed and attempt recovery.

### **WARNING: another disk must be connected to *group***

#### **Explanation**

You must connect another disk to *group*.

#### **Resolution**

Connect another disk.

### **WARNING: object: copying not completed successfully**

#### **Explanation**

Synchronization copying did not complete successfully.

#### **Resolution**

Disk failure may have occurred. Identify the cause by referring to GDS log message and syslog message.

### **WARNING: object: gave up wait for the completion of copying by a cancel request**

#### **Explanation**

Synchronization copying was canceled before completion.

#### **Resolution**

Check the status of the object *object*. If the copying is in progress, there is no need to work around. If it is not in progress, re-execute the copying where it is necessary. For details, see "Appendix D Command Reference."

**WARNING: *group*: no spare disk available**

**Explanation**

There is no valid spare disk in *group*.

**Resolution**

Define a spare disk as needed.

**WARNING: *object.volume*: cannot attached due to in *status* status**

**Explanation**

Slice *object.volume* could not be attached since it is in *status* status.

**Resolution**

Check the slice status and cancel *status* status as needed.

**WARNING: *volume*: no need to resize volume**

**Explanation**

There was no need to resize the volume *volume*.

**Resolution**

There is no need to work around in particular.

**WARNING: *disk.volume*: special file(s) not found**

**Explanation**

Special file for slice *disk.volume* could not be found.

**Resolution**

No particular resolution required.

**WARNING: *node*: node in stopped *status***

**Explanation**

*node* is in STOP status.

**Resolution**

No particular resolution required. However, promptly activating the node is recommended.

**WARNING: *node*: node in abnormal status**

**Explanation**

*node* is in abnormal status.

**Resolution**

No particular resolution required. However, promptly recovering the node and activating it normally is recommended.

**WARNING: *object*: already in *status* status**

**Explanation**

*object* is already in *status* status.

**Resolution**

No particular resolution required.

**WARNING: *disk : device : write error, erro=ermo***

**Explanation**

A write error occurred on physical disk *device*.

**Resolution**

If *device* is a write-locked disk that is connected to a switch group, no action is required. In other situations, have *device* switchable and check its status.

**WARNING: *disk : device : node: write error, erro=ermo***

**Explanation**

A write error occurred on physical disk *device* on node *node*.

**Resolution**

If *device* is a write-locked disk that is connected to a switch group, no action is required. In other situations, have *device* switchable and check its status.

**WARNING: *object: no need to update attribute value***

**Explanation**

There was no need to change *object* attributes.

**Resolution**

No particular resolution required.

**WARNING: *ignored parameter param for EC***

**Explanation**

Since EC function will be used for copying, parameter *param* was ignored.

**WARNING: *ignored parameter param for OPC***

**Explanation**

Since OPC function will be used for copying, parameter *param* was ignored.

**WARNING: *proxy: no session***

**Explanation**

No copy session exists between the proxy *proxy* and the master

**Resolution**

There is no resolution.

### E.4.3 Information Messages

---

**INFO: waiting for a response from sdxservd daemon...**

**Explanation**

Awaiting response from sdxservd daemon.

**INFO: *class*: created class**

**Explanation**

*class* was created.

**INFO: *disk*: created disk**

**Explanation**

*disk* was registered.

**INFO: *device*: disabled access to physical special files  
*/dev/devices\****

**Explanation**

Physical special files can no longer be accessed.

**INFO: *class*: removed class**

**Explanation**

*class* was removed.

**INFO: *group*: created group**

**Explanation**

*group* was created.

**INFO: *disk*: connected disk to group *group***

**Explanation**

*disk* was connected to *group*.

**INFO: *igroup*: connected group to another group *hgroup***

**Explanation**

Group *igroup* was connected to another group *hgroup*.

**INFO: *group*: removed group**

**Explanation**

*group* was removed.

**INFO: *object*: waiting for the completion of copying...**

**Explanation**

Waiting for synchronization copying to complete.

**INFO: *object*: copying completed successfully**

**Explanation**

Synchronization copying completed successfully.

**INFO: *volume*: created volume**

**Explanation**

*volume* was created.

**INFO: *volume*: started volume on node *node*  
/dev/sfdsk/class/rdisk/*volume*  
/dev/sfdsk/class/dsk/*volume***

**Explanation**

*volume* was started on *node*. You can now access via special file.

**INFO: *volume*: stopped volume on node *node***

**Explanation**

*volume* was stopped on *node*.

**INFO: *volume*: removed volume**

**Explanation**

*volume* was removed.

**INFO: *volume*: resized volume**

**Explanation**

The volume *volume* was resized.

**INFO: *disk*: disconnected disk from group *group***

**Explanation**

*disk* was disconnected from *group*.

**INFO: *lgroup*: disconnected group from group *hgroup***

**Explanation**

Group *lgroup* was disconnected from *hgroup*.

**INFO: *group*: free blocks are increased**

**Explanation**

Free blocks on *group* increased.

**INFO: *disk*: removed disk**

**Explanation**

*disk* was removed from class.

**INFO: *device*: enabled access to physical special files  
*/dev/devices\****

**Explanation**

You can now access to physical special files.

**INFO: *disk.volume*: temporarily detached slice from volume *volume***

**Explanation**

Slice *disk.volume* was temporarily disconnected from *volume*.

**INFO: *disk.volume*: enabled slice  
*/dev/sfdsk/class/rdisk/disk.volume*  
*/dev/sfdsk/class/dsk/disk.volume***

**Explanation**

Slice *disk.volume* can now be accessed partly from volume.  
Use the special file displayed.

**INFO: *disk.volume*: disabled slice**

**Explanation**

Slice *disk.volume* cannot be accessed any more.

**INFO: *disk.volume*: reattached slice to volume *volume***

**Explanation**

Slice *disk.volume* is attached to *volume* again.

**INFO: *disk: device: ready for swapping***

**Explanation**

*disk* is ready for swapping. When using a file unit device supporting active swap, you may remove the physical disk *device* at this point for swapping. If active swap is not supported, shut down the system and turn off the power before removing the physical disk.

**INFO: *disk: device: ready for swapping on node node***

**Explanation**

On node *node*, physical disk *device* of disk *disk* is now switchable. If the disk unit supports hot swap, you may simply demount *device* connected to *node* and perform swapping. If it does not support hot swap, shut down *node* and turn off the power, then demount *device* and perform swapping.

**INFO: *disk: restored disk***

**Explanation**

*disk* restored its original status.

**INFO: *volume: recovered volume with disk disk***

**Explanation**

*volume* was recovered based on *disk*.

**INFO: *volume: recovered volume with lower level group lgroup***

**Explanation**

Volume *volume* was recovered based on lower level group *lgroup*.

**INFO: *volume : recovered volume in group group***

**Explanation**

Volume *volume* in group *group* has been recovered.

**INFO: *disk: recovered disk***

**Explanation**

disk *disk* was recovered.

**INFO: *no unsynchronized slice in volume***

**Explanation**

There is no slice that needs synchronization copying to be started, or resumed in *volume*.

**INFO: *volume: copying canceled***

**Explanation**

Synchronization copying of *volume* has been cancelled.

**INFO: no copying slice exists in *volume***

**Explanation**

There is no slice in the process of copying in *volume*.

**INFO: *volume*: copying interrupted**

**Explanation**

Synchronization copying on *volume* was canceled.

**INFO: *class*: updated attribute value of class**

**Explanation**

Attribute value of *class* was updated.

**INFO: *disk*: updated attribute value of disk**

**Explanation**

Attribute value of *disk* was updated.

**INFO: *group*: updated attribute value of group**

**Explanation**

Attribute value of *group* was updated.

**INFO: *volume*: updated attribute value of volume**

**Explanation**

Attribute value of *volume* was updated.

**INFO: *slice*: updated attribute value of slice**

**Explanation**

Attribute value of *slice* was updated.

**INFO: *disk.volume*: started slice on node *node*  
/dev/sfdsk/class/rdisk/*disk.volume*  
/dev/sfdsk/class/dsk/*disk.volume***

**Explanation**

Slice *disk.volume* was activated on *node*.  
You may access slice by using the displayed special file.



**INFO: *disk.volume*: stopped slice on node *node***

**Explanation**

Slice *disk.volume* was stopped on *node*.  
You cannot access slice until it is activated again.

**INFO: *disk.volume*: disabled slice on remote node *node***

**Explanation**

Slice *disk.volume* has become unavailable on remote node *node*.

**INFO: *disk.volume*: current node *node* took over and started slice  
*/dev/sfdsk/class/rdisk/disk.volume*  
*/dev/sfdsk/class/dsk/disk.volume***

**Explanation**

Current node took over slice *disk.volume* and activated it.  
You may access to slice by using the displayed special file.

**INFO: *volume*: copying parameter modification completed**

**Explanation**

Modification of parameter related to the copying process of *volume* is complete.

**INFO: configuration parameters modification completed**

**Explanation**

Modification of configuration parameter is complete.

**INFO: *proxy*: related and joined to master *master***

**Explanation**

Proxy *proxy* has been related and joined to master *master*.

**INFO: *proxy*: created proxy volume**

**Explanation**

Proxy volume *proxy* has been created.

**INFO: *proxy*: broken relation with master**

**Explanation**

Relationship between master and proxy *proxy* has been cancelled.

**INFO: *proxy*: forced to break relation with master**

**Explanation**

Relationship between master and proxy *proxy* was forcibly cancelled.

**INFO: *proxy*: must be recovered by `sdxfix(8)` command**

**Explanation**

In order to restore data of proxy *proxy*, volume must be restored with `sdxfix(8)` command.

**INFO: *volume*: started volume on node *node*, ...**  
**`/dev/sfdsk/class/rdsk/volume`**  
**`/dev/sfdsk/class/dsk/volume`**

**Explanation**

Volume was started on node *node*, .... It can be accessed with the special file.

**INFO: *proxy*: parted from master**

**Explanation**

Proxy *proxy* was parted from master.

**INFO: *proxy*: parted from master instantly**

**Explanation**

Proxy *proxy* was immediately parted from master.

**INFO: *proxy*: rejoined to master *master***

**Explanation**

Proxy *proxy* has been rejoined to master *master*.

**INFO: *master*: restored data from proxy *proxy***

**Explanation**

Copying from proxy *proxy* to master *master* started.

**INFO: *master*: restored data from proxy *proxy* instantly**

**Explanation**

Copying from proxy *proxy* to master *master* started. Even if the copying is in process, you may start master volumes and access valid data.

**INFO: swapped slice components between master and proxy**

**Explanation**

Slice comprising proxy and slice comprising master have been swapped.

**INFO: *proxy*: session canceled**

**Explanation**

A session of the disk unit's copy function that resided between proxy *proxy* and the master was canceled.

**INFO: *proxy*: related to master *master***

**Explanation**

Master *master* and proxy *proxy* were related and parted.

**INFO: *proxy*: updated data with master *master***

**Explanation**

Data of proxy *proxy* was updated with data of master *master*.

**INFO: *proxy*: updated data with master *master* instantly**

**Explanation**

Data of proxy *proxy* was instantly updated with data of master *master*.

**INFO: *class*: current node *node* got class master privilege**

**Explanation**

Local node *node* obtained master privileges of shared class *class*.

---

#### **E.4.4 Fix Messages**

---

**TO FIX: Usage: *usage***

**Resolution**

Follow the *usage* syntax and execute the command.

---

#### **E.4.5 Internal Error Messages**

---

**command: *severity*: module: internal error, details**

**Explanation**

Internal error occurred. *details* gives you the cause of error, *command* the command name, *severity* the severity of message, and *module* the module name (usually the internal function name) where the event was detected.

**Resolution**

Collect investigation material and contact your local customer support.

## E.5 Operation Management View Messages

This section describes the GDS Management View messages.

### E.5.1 Error Messages

---

#### 5000 Command line output: *details*

##### Explanation

An error given in *details* occurred when executing the command.

##### Resolution

Identify the cause referring to the command line output. When you cannot identify the cause, contact your local customer support.

#### 5001 Executing the command was unsuccessful. Continue?

##### Explanation

Executing the command was unsuccessful.

##### Resolution

Click <Yes> to continue, or <No> to cancel.

#### 5002 The following exceptions were received: *details*

##### Explanation

Exceptions given in *details* occurred.

##### Resolution

Contact your local customer support.

#### 5003 The input object name is already defined. Change the object name.

##### Explanation

The input object name is already defined.

##### Resolution

Change the object name.

#### 5004 An error occurred in the log monitoring mechanism on node *node*. Monitoring tasks will stop. Check the status of the node and update to the latest information.

##### Explanation

An error such as a node crash was detected in the log monitoring mechanism.

##### Resolution

Verify that the node is in operation, and update to the latest information.

**5005 Reading GDS configuration information was unsuccessful on node *node*. Contact the system administrator.**

**Explanation**

Reading GDS configuration information was unsuccessful on node *node*.

**Resolution**

Contact your local customer support.

**5006 An invalid class name was specified. Input another name.**

**Explanation**

An invalid character was specified for the class name.

**Resolution**

Click <OK> and type a class name again. For information on characters available for class names, see "A.1.1 Object Name."

**5007 An invalid group name was specified. Input another name.**

**Explanation**

An invalid character was specified for the group name.

**Resolution**

Click <OK> and type a group name again. For information on characters available for group names, see "A.1.1 Object Name."

**5008 An invalid volume name was specified. Input another name.**

**Explanation**

An invalid character was specified for the volume name.

**Resolution**

Click <OK> and type a volume name again. For information on characters available for volume names, see "A.1.1 Object Name."

**5009 An invalid disk name was specified. Input another name.**

**Explanation**

An invalid character was specified for the disk name.

**Resolution**

Click <OK> and type a disk name again. For information on characters available for disk names, see "A.1.1 Object Name."

**5010 The file system type or mount point has not been set. Select a file system type and/or input a mount point.**

**Explanation**

<Create> was selected before specifying the file system type or mount point.

**Resolution**

Select a file system type and/or input a mount point, and click <Create>.

**5011 The input mount point is already in use. Change the mount point.**

**Explanation**

The input mount point is already in use with a different file system.

**Resolution**

Change the mount point, and click <Create>.

**5015 There are unexpected device ID on node *node*. Exit.**

**Explanation**

Starting GDS Management View failed due to unsupported GDS configuration information on node *node*.

**Resolution**

Contact your local customer support.

**5016 The order of removing disks and groups composing the group is wrong. Remove them in the proper order.**

**Explanation**

A disk or group not removable was selected.

**Resolution**

When you remove disks and groups composing a group, you must select them in the inverse order against the added order.

**5017 The stripe width setting is wrong. Specify an integer that is the power of 2.**

**Explanation**

An invalid character or value was specified for the stripe width.

**Resolution**

Click <OK>, and input a valid stripe width again.

**5018 The specified name is already in use. Input another name.**

**Explanation**

The specified name is in use.

**Resolution**

Click <OK> and type an unused class name, and then click <Next>.

**5019 An invalid volume size was specified. Input a valid size.**

**Explanation**

The specified volume size is invalid.

**Resolution**

Click <OK> and type a valid volume size. When the maximum size is unknown, you can apply the maximum size to [Volume Size] by clicking <Maximum Size>.

## E.5.2 Warning Messages

---

### 5000 Command line output: *details*

#### Explanation

An error given in *details* occurred when executing the command.

#### Resolution

Identify the cause referring to the command line output. When you cannot identify the cause, contact your local customer support.

### 5100 The input class name is already defined. Register the disk with this class?

#### Explanation

The input class name is already defined.

#### Resolution

Click <Yes> to add the disk to the specified class, or <No> to register the disk with another class.

### 5101 The input group name is already defined. Connect the disk with this group?

#### Explanation

The input group name is already defined.

#### Resolution

Click <Yes> to additionally connect the disk with the specified group, or <No> to connect the disk with another group.

### 5102 If you register a physical disk with this class, all the data contained in the disk will be lost. Continue?

#### Explanation

All of the disk data will be lost if a physical disk is registered with the class.

#### Resolution

Click <Yes> to register the physical disk, or <No> to cancel.

### 5103 If you remove the volume, all the volume data will be lost. Continue?

#### Explanation

All of the volume data will be lost if you remove the volume.

#### Resolution

Click <Yes> to remove the volume, or <No> to cancel.

### 5104 The volume will be recovered. Continue?

#### Explanation

A volume that cannot be accessed will be recovered.

#### Resolution

Click <Yes> to recover the volume, or <No> to cancel.

**5105 The selected disk is normal. Swap disks?**

**Explanation**

A normal disk was selected for swapping.

**Resolution**

Click <Yes> to swap the disks, or <No> to cancel.

**5108 The physical disk will be excluded from GDS.  
To swap the disks, the relative slices will stop. Detaching the disk will affect data reliability. Continue?**

**Explanation**

When physical disks are swapped, the relative slices will stop operating until the restore is complete, and the reliability of the volume will fall.

**Resolution**

Click <Yes> to swap physical disks, or <No> to cancel.

**5109 The physical disk will be included into GDS.  
All the data in the physical disk will be lost. Continue?**

**Explanation**

When a physical disk is restored, data in the physical disk will be lost.

**Resolution**

Click <Yes> to restore the physical disk, or <No> to cancel.

**5110 The slice will be detached from the volume.  
Detaching the slice will affect data reliability. Continue?**

**Explanation**

Detaching a slice will lower reliability.

**Resolution**

Click <Yes> to detach the slice, or <No> to cancel.

**5111 The detached slice will be attached to the volume.  
All the data in the slice will be lost. Continue?**

**Explanation**

When a slice is attached, data in the slice will be overwritten.

**Resolution**

Click <Yes> to attach the slice, or <No> to cancel.

**5112 The volume will be stopped. Continue?**

**Explanation**

The volume will be stopped.

**Resolution**

Click <Yes> to stop volume, or <No> to cancel.



**5113 Removing all disks configuring a class will result in removing the class as well. Continue?**

**Explanation**

Removing all the disks in a class will result in removing the class as well.

**Resolution**

Click <Yes> to remove class, or <No> to cancel.

**5114 Removing all disks configuring a group will result in removing the group as well. Continue?**

**Explanation**

Removing all the disks in a group will result in removing the group as well.

**Resolution**

Click <Yes> to remove group, or <No> to cancel.

**5115 A file system will be created on the selected volume. Continue?**

**Explanation**

A file system will be created on the selected volume. If you continue, all the data in the volume will be lost.

**Resolution**

Click <Yes> to create a file system, or <No> to cancel.

**5116 The selected volume will be unmounted and mount information for the volume will be deleted. It will be no longer available as a file system. Continue?**

**Explanation**

The selected volume will be unmounted and mount information for the volume will be deleted. If it is deleted, the volume can no longer be used as a file system.

**Resolution**

Click <Yes> to remove the file system, or <No> to cancel.

**5117 Changes on GDS configuration information were detected. Update to the latest configuration information?**

**Explanation**

Commands other than those from GDS Management View changed the GDS configuration information.

**Resolution**

Click <Yes> to update the GDS Management View display, or <No> to cancel.

**5118 The physical disk will be excluded from GDS. Continue?**

**Explanation**

In order to swap physical disks, an unused disk or a spare disk will be detached. This will not affect the volume.

**Resolution**

Click <Yes> to swap the physical disks, or <No> to cancel.

**5119 A file system has been configured on the selected volume. Remove the file system from the File System Configuration.**

**Explanation**

The volume cannot be removed because of a file system configured on the volume.

**Resolution**

Remove the file system from the [File System Configuration] screen and then remove the volume.

**5120 Physical disks have not been swapped. Verify that swapping took place. Are you sure you want to attach the disk?**

**Explanation**

Physical disks may not have been swapped.

**Resolution**

Click <Yes> to attach the physical disk, or <No> to cancel.

**5121 The status of physical disks connected to the system will be updated. Updating to the latest information may consume time. Continue?**

**Explanation**

The status of physical disks connected to the system will be updated. During the update, operations from GDS Management View are unavailable.

**Resolution**

Click <Yes> to update the physical disk information, or <No> to cancel.

**5122 A file system is created in a volume within the class. You cannot change attributes because it will change the path. Change the attributes after removing the file system.**

**Explanation**

Renaming the class containing a volume with a file system was attempted.

**Resolution**

Renaming the class will result in changing the access path. First delete the file system.

**5123 A file system is created in the volume. You cannot change attributes because it will change the path. Change the attributes after removing the file system.**

**Explanation**

Renaming the volume with a file system was attempted.

**Resolution**

Renaming the volume will result in changing the access path. First delete the file system.

**5128 The slice will become inaccessible. Continue?**

**Explanation**

The detached slice will be made inaccessible.

**Resolution**

Click <Yes> to restrict access to the slice, or <No> to allow access.

**5129 The slice will become accessible. Continue?**

**Explanation**

The slice being inaccessible as a result of stopping the slice will be made accessible.

**Resolution**

Click <Yes> to allow access to the slice, or <No> to restrict access.

**5130 Copying of the slice will be cancelled. It cannot be resumed. Continue?**

**Explanation**

The resynchronization copying of the slice will be cancelled. The copying cannot be resumed from the point where it stops.

**Resolution**

Click <Yes> to cancel resynchronization copying, or <No> not to cancel.

**5131 Copying of the slice will start (resume). Continue?**

**Explanation**

Copying of the slice will be performed to synchronize the data of the slices of the volume. Copying of an interrupted slice will resume as well.

**Resolution**

Click <Yes> to start (resume) resynchronization copying, or <No> not to start (resume).

**5133 An error was detected on *node*. Display *node* information?**

**Explanation**

An object error was detected on another node *node* configuring the cluster system.

**Resolution**

Click <Yes> to display the information about the node with the error, or <No> not to display.

**5134 An error was detected on *node*. Check the node for the status after the current operation ends.**

**Explanation**

An object error was detected on another node *node* configuring the cluster system.

**Resolution**

Check the node on which the error was detected for the status after the current operation ends.

**5139 The proxy will be joined. The joined proxy will lose existing data. Do you want to continue?**

**Explanation**

If the proxy is joined, its contents are overwritten by data of the master volume.

**Resolution**

Click <Yes> to join the proxy, or <No> to cancel the process.

**5140 The proxy will be joined. Do you want to continue?**

**Explanation**

The master group and the proxy group will be joined.

**Resolution**

Click <Yes> to join the proxy, or <No> to cancel the process.

**5141 The proxy will be related. Do you want to continue?**

**Explanation**

The master and the proxy will be related.

**Resolution**

Click <Yes> to relate the proxy, or <No> to cancel the process.

**5142 The relationship between the master and the proxy will be canceled. Do you want to continue?**

**Explanation**

The relationship between the master and the proxy will be canceled.

**Resolution**

Click <Yes> to break the relationship, or <No> to cancel the process.

**5143 The master and the proxy will be rejoined. The rejoined proxy will lose existing data. Do you want to continue?**

**Explanation**

If the proxy is rejoined, its contents are overwritten by data of the master.

**Resolution**

Click <Yes> to rejoin the proxy, or <No> to cancel the process.

**5144 Slices of the master and the proxy will be swapped. Do you want to continue?**

**Explanation**

Slices comprising the master and the proxy are switched.

**Resolution**

Click <Yes> to switch the slices, or <No> to cancel the process.

---

---

**E.5.3 Information Messages**

**5200 The slice was attached and the volume will be restored.**

**Explanation**

The slice was attached. In order to restore the volume, resynchronization copying will start.

**Resolution**

Click <OK> and close the message. Check the completion of resynchronization copying in volume information field on the main screen.

For the resynchronization copying progress, check the slice information field in the main screen.

**5201 The volume restore is complete.**

**Explanation**

The volume was restored.

**Resolution**

Click <OK> and close the message.

**5202 The slice was detached. The following access path is available.  
Access path: /dev/sfdsk/classname/rdisk/slicename**

**Explanation**

The slice was detached to make it accessible.

**Resolution**

Pay attention to the displayed access path and click <OK>.

**5203 The disk was excluded from GDS. Swap physical disks. Ask your local customer support for physical disk swapping.**

**Explanation**

For swapping, a physical disk was excluded from GDS.

**Resolution**

Click <OK>, and contact your local customer support to have physical disks swapped.

**5204 The physical disk was included into GDS. The volume will be restored.**

**Explanation**

The swapped physical disk was placed under GDS.

**Resolution**

Click <OK> and close the message.

**5206 A file system was created.**

**Explanation**

A file system was created and mount information was added.

**Resolution**

Click <OK> and close the message.

**5207 The file system was removed.**

**Explanation**

The file system and mount information was removed.

**Resolution**

Click <OK> and close the message.

**5209 Status monitoring will be updated every *interval* minutes.**

**Explanation**

The object monitoring interval was changed.

**Resolution**

Click <OK> and close the message.

**5210 Exit Global Disk Services?**

**Explanation**

GDS Management View will be exited.

**Resolution**

Click <Yes> to exit, or <No> to continue operation.

**5213 The slice was made inaccessible. Activate the slice in order to gain access.**

**Explanation**

The detached slice was made inaccessible.

**Resolution**

Click <OK> and close the message.

**5214 The slice is now accessible.**

**Explanation**

The detached slice was made accessible.

**Resolution**

Click <OK> and close the message.

**5215 The selected slice is neither temp nor temp-stop status. You cannot perform the operation on the selected slice.**

**Explanation**

The [Stop/Activate Slice] operation was attempted on a slice not in the temp or temp-stop status.

**Resolution**

Click <OK> and close the message.

**5216 Copying of the slice was cancelled.**

**Explanation**

The resynchronization copying of the slice was cancelled.

**Resolution**

Click <OK> and close the message.

**5217 Copying of the slice was started (resumed).**

**Explanation**

The resynchronization copying was started (resumed).

**Resolution**

Click <OK> and close the message.

**5218 The selected volume is not in the copy status. You cannot perform the operation on the selected volume.**

**Explanation**

The copying operation was attempted while the slices in the volume are not in the copy status.

**Resolution**

Click <OK> and close the message.

**5219 The selected slice is neither active nor stop status. You cannot perform the operation on the selected slice.**

**Explanation**

The [Detach Slice] operation was attempted on a slice not in the active or stop status.

**Resolution**

Click <OK> and close the message.

**5220 There is a slice in the copy status in the selected volume. You cannot perform the operation on the selected volume.**

**Explanation**

The [Start Copying] operation was attempted on a volume with a slice in process of synchronization copying.

**Resolution**

Click <OK> and close the message.

**5221 There are no slices in the invalid status in the selected volume. You cannot perform the operation on the selected volume.**

**Explanation**

The [Start Copying] operation was attempted on a volume without a slice in the invalid status.

**Resolution**

Click <OK> and close the message.

**5222 There is no available proxy for the selected master. Create a volume that can be joined first.**

**Explanation**

There is no volume that can be joined as a proxy volume to the selected master volume.

**Resolution**

Create a volume to be joined to the master volume first conforming to the following rules.

- Belonging to the class to which the master volume belongs
- Belonging to a mirror group or to a single disk
- Not belonging to the group to which the master volume belongs
- Not belonging to the single disk to which the master volume belongs
- Not related to another volume as a master or a proxy
- Equivalent to the master volume in size

For details, see “A.1.7 Exception to Proxy Configuration” and the description of the Join option in “D.12 sdxcproxy - Proxy object operations.”

**5223 There is no available proxy for the selected master. Create a group that can be joined first.**

**Explanation**

There is no group that can be joined as a proxy group to the master group.

**Resolution**

Create a group to be joined to the master group first conforming to the following rules.

- Belonging to the class to which the master group belongs
- Mirror type
- Not related to another group as a master or a proxy
- Including no volume

For details, see “A.1.7 Exception to Proxy Configuration” and the description of the Join option in “D.12 sdxproxy - Proxy object operations.”

**5224 Joining the proxy succeeded. Synchronization copying will start.**

**Explanation**

The master and the proxy were joined. Synchronization copying from the master to the proxy will start to synchronize them.

**Resolution**

Click <OK> to close the message. Check the synchronization copying status in the slice information field in the main screen.

**5225 There is no available proxy for the selected master. Create a volume that can be related first.**

**Explanation**

There is no volume that can be related to the selected master volume.

**Resolution**

Create a volume to be related to the master volume conforming to the following rules.

- Belonging to the class to which the master volume belongs
- Belonging to a mirror group or to a single disk
- Not belonging to the group to which the master volume belongs
- Not belonging to the single disk to which the master volume belongs
- Not related to another volume as a master or a proxy
- Equivalent to the master volume in size

For details, see “A.1.7 Exception to Proxy Configuration” and the description of the Relate option in “D.12 sdxproxy - Proxy object operations.”

**5226 There is no available proxy for the selected master. Create a group that can be related first.**

**Explanation**

There is no group that can be related as a proxy group to the selected master group.

**Resolution**

Create a group to be related to the master group conforming to the following rules.

- Belonging to the class to which the master group belongs
- Mirror type
- Not related to another group as a master or a proxy
- Whose volume configuration (offset and size) is same as that of the master group

For details, see “A.1.7 Exception to Proxy Configuration” and the description of the Relate option in “D.12 sdxproxy - Proxy object operations.”



### **5227 Relating the proxy succeeded.**

#### **Explanation**

The master and the proxy were related.

#### **Resolution**

Click the <OK> to close the message.

### **5228 Breaking the proxy succeeded.**

#### **Explanation**

The relationship between the master and the proxy was broken.

#### **Resolution**

Click the <OK> to close the message.

### **5229 Parting the proxy succeeded.**

#### **Explanation**

The proxy was parted from the master.

#### **Resolution**

Click <OK> to close the message.

### **5230 Rejoining the proxy succeeded. The master and the proxy will be resynchronized**

#### **Explanation**

The master and the proxy were rejoined. Resynchronization copying from the master to the proxy will start to resynchronize them.

#### **Resolution**

Click <OK> to close the message. Check the resynchronization copying status in the volume Information field in the main screen. For the resynchronization copying progress, check the slice information field in the main screen.

### **5231 The update of the proxy started.**

#### **Explanation**

Copying from the master to the proxy started.

#### **Resolution**

Click <OK> to close the message. Check the completion of the resynchronization copying in the volume information field in the main screen. For the copying status, check the slice information field in the main screen.

### **5232 Updating the proxy succeeded. Start up the proxy to access.**

#### **Explanation**

Copying from the master to the proxy started. Before the copying ends, you may start the proxy for access. Contents of the proxy are overwritten by the data of the master at the moment when the copying started.

#### **Resolution**

Click <OK> to close the message. Check the copying status. in the slice information field on the main screen.

### **5233 The restoration of the master started.**

#### **Explanation**

Copying from the proxy to the master started.

#### **Resolution**

Click <OK> to close the message. Check the completion of the copying in the volume information field in the main screen. For the copying progress, check the slice information field on the main screen.

### **5234 Restoring the master succeeded. Start up the master to access.**

#### **Explanation**

Copying from the proxy to the master started. Before the copying ends, you may start the master for access. Contents of the master are overwritten by the data of the proxy at the moment when the copying started.

#### **Resolution**

Click <OK> to close the message. Check the copying status in the volume information field on the main screen.

### **5235 Rejoining the proxy succeeded, and the restoration of the master started.**

#### **Explanation**

The master and the proxy were rejoined and synchronization copying from the proxy to the master started.

#### **Resolution**

Click <OK> to close the message. Check the completion of the resynchronization copying in the volume information field on the main screen. For the resynchronization copying progress, check the slice information field in the main screen.

### **5236 Rejoining the proxy and restoring the master succeeded. Start up the master to access.**

#### **Explanation**

The master and the proxy were rejoined and resynchronization copying from the proxy to the master started. Before the resynchronization copying ends, you may start the master for access. Contents of the master are overwritten by the data of the proxy at the moment when the copying started.

#### **Resolution**

Click <OK> to close the message. For the resynchronization copying status, check the slice information field on the main screen.

### **5237 Swapping the slices of the master and the proxy succeeded.**

#### **Explanation**

Slices comprising the master and the proxy were switched.

#### **Resolution**

Click <OK> to close the message.

# Appendix F Troubleshooting

Since GDS automatically attempts recovery when various abnormalities are detected, users do not have to be aware of most abnormalities. However, depending on the type of abnormality, or when multiple abnormalities occur at the same time, automatic recovery may not be possible. This section describes resolutions to failures on objects and physical disks managed by GDS for systems administrators.



Note

As a general rule, all operations explained here must be executed with superuser authority.

## F.1 Resolving Problems

### F.1.1 Slice Status Abnormality

#### (1) Mirror slice configuring the mirror volume is in INVALID status.

##### *Explanation*

You can check the status of the slice configuring the volume as follows.

# sdxinfo -S -o Volume1					
OBJ	CLASS	GROUP	DISK	VOLUME	STATUS
-----	-----	-----	-----	-----	-----
slice	Class1	Group1	Object1	Volume1	ACTIVE
slice	Class1	Group1	Object2	Volume1	INVALID

In this example, among the slices that exist in volume Volume1, the slice within Object2 is in INVALID status, as shown in the STATUS field. Object2 is a disk or lower level group connected to the highest level mirror group Group1.

The following five events could possibly cause the INVALID status of the mirror slice Volume1.Object2.

- An I/O error occurred on the mirror slice Volume1.Object2.
  - (Cause a)  
A disk component relevant to Object2 failed to operate properly, and an I/O error occurred on the mirror slice Volume1.Object2.
  - (Cause b)  
A component other than disks relevant to Object2 (such as an I/O adapter, an I/O cable, an I/O controller, a power supply, and a fan) failed to operate properly, and an I/O error occurred on the mirror slice Volume1.Object2.
- An I/O error occurred on the mirror slice Volume1.Object1 during synchronization copying to the mirror slice Volume1.Object2.
  - (Cause a')  
A disk component relevant to Object1 failed to operate properly during synchronization copying to the mirror slice Volume1.Object2, and an I/O error occurred on the copy source slice Volume1.Object1.
  - (Cause b')  
A component other than disks relevant to Object1 (such as an I/O adapter, an I/O cable, an I/O controller, a power supply and a fan) failed to operate properly during synchronization copying to the mirror slice Volume1.Object2, and an I/O error occurred on the mirror slice Volume1.Object1 that is the copy source.
- Others
  - (Cause c)  
Synchronization copying to the mirror slice Volume1.Object2 was cancelled as the result of a cause such as [Cancel Copying] selection from the GDS Management View, sdxcopy command execution, or a power outage.

## Resolution

1) Identify the physical disk name of a faulty disk using the `sdxinfo` command.

### (Example A1)

# <code>sdxinfo -G -o Volume1</code>										
OBJ	NAME	CLASS	DISKS	BLKS	FREEBLKS	SPARE				
group	Group1	Class1	Object1:Object2	17596416	17498112	0				

# <code>sdxinfo -D -o Volume1 -e long</code>										
OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	FREEBLKS	DEVCONNECT	STATUS	E
disk	Object1	mirror	Class1	Group1	sda	17596416	*	node1:node2	ENABLE	*
disk	Object2	mirror	Class1	Group1	sdb	17596416	*	node1:node2	ENABLE	*

In this example, Object2 is a disk connected with the highest level group Group1. As indicated in the E field, an I/O error occurred on the disk Object2, and the possible cause is (Cause a) or (Cause b). The physical disk name of the disk Object2 is sdb as shown in the DEVNAM field.

In example A1, if the value 0 is in the E field of the disk Object2 including a slice in the INVALID status and if the value 1 is in the E field of the disk Object1 that is mirrored with the disk Object2, it indicates an I/O error occurred on the disk Object1 and the possible cause is (Cause a') or (Cause b'). In such a case, see "(2) The copy destination slice was made INVALID due to an I/O error generated on the copy source slice during synchronization copying" and perform restoration.

If the E field of any disk does not contain the value 1 in example A1, the possible cause is (Cause c).

### (Example B1)

# <code>sdxinfo -G -o Volume1</code>										
OBJ	NAME	CLASS	DISKS	BLKS	FREEBLKS	SPARE				
group	Group1	Class1	Object1:Object2	35127296	35028992	0				
group	Object1	Class1	Disk1:Disk2	35127296	*	*				
group	Object2	Class1	Disk3:Disk4	35127296	*	*				

# <code>sdxinfo -D -o Volume1 -e long</code>										
OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	FREEBLKS	DEVCONNECT	STATUS	E
disk	Disk1	stripe	Class1	Object1	sda	17596416	*	node1:node2	ENABLE	0
disk	Disk2	stripe	Class1	Object1	sdb	17596416	*	node1:node2	ENABLE	0
disk	Disk3	stripe	Class1	Object2	sdc	17682084	*	node1:node2	ENABLE	1
disk	Disk4	stripe	Class1	Object2	sdd	17682084	*	node1:node2	ENABLE	0

In this example, Object2 is a lower level group connected with the highest level group Group1. As indicated in the E field, an I/O error occurred on the disk Disk3 connected with Object2 and the possible cause is (Cause a) or (Cause b). The physical disk name corresponding to Disk3 is sdc as shown in the DEVNAM field.

In example B1, if the 0 value is in the E field of the disks (Disk3 and Disk4) connected with the lower level group Object2 including a slice in the INVALID status and if the value 1 is in the E field of the disk (Disk1 or Disk2) connected with the lower level group Object1 that is mirrored with the lower level group Object2, it indicates an I/O error occurred on the disk connected with Object1 and the possible cause is (Cause a') or (Cause b'). In such a case, see "(2) The copy destination slice was made INVALID due to an I/O error generated on the copy source slice during synchronization copying" and perform restoration.

In example B1, if there are no disks with "1" in the E field, the possible cause of the INVALID status is (Cause c).

2) Refer to disk driver log messages and check the physical disk abnormalities.

The causes of disk hardware failures can be failures or defects in components such as I/O adapters, I/O cables, I/O controllers, power supplies, and fans other than the disks.

Contact your local customer support and specify which component failed, or might be defective.

If there are no failures or defective components, the possible cause of the INVALID is (Cause c).

If the possible cause is (Cause a), follow procedures 3a to 5a, and 6. If the possible cause is (Cause b), follow procedures 3b and 6. If the possible cause is (Cause c), follow procedures 3c and 6.

3a) If the possible cause is (Cause a), you must perform the following operations, before and after the disk swap. For procedures on swapping disks from Operation Management View, see "5.3.4 Disk Swap."

Before swapping the disks, execute the following command.

**(Example A3)**

```
# sdxswap -O -c Class1 -d Object2
```

In the example, disk Object2 connected to the highest level group Group1 will be swapped.

**(Example B3)**

```
# sdxswap -O -c Class1 -d Disk3
```

In the example, disk Disk3 will be swapped. Disk3 is a disk connected to lower level group Object2, which is a lower level group of the highest level group Group1.

4a) Swap disks.

5a) After swapping disks, execute the following command.

**(Example A5)**

```
# sdxswap -I -c Class1 -d Object2
```

or

**(Example B5)**

```
# sdxswap -I -c Class1 -d Disk3
```

3b) If the possible cause is (Cause b), you must shut down the system. Recover the failed component and boot your system. Mirroring status will be recovered by automatic synchronization copying.

3c) If the possible cause is (Cause c), perform synchronization copying of mirror volume.

```
# sdxcopy -B -c Class1 -v Volume1
```

6) You can confirm the recovery of the slice configuring the volume, as shown below.

```
# sdxinfo -S -o Volume1
OBJ CLASS GROUP DISK VOLUME STATUS
---- -
slice Class1 Group1 Object1 Volume1 ACTIVE
slice Class1 Group1 Object2 Volume1 ACTIVE
```

In this example, the slices within Object1 and Object2 are both in ACTIVE status. This indicates that the recovery process was completed successfully.

## (2) The copy destination slice was made INVALID due to an I/O error generated on the copy source slice during synchronization copying.

### Explanation

When an I/O error occurs on the copy source slice during synchronization copying, the copy destination slice becomes INVALID while the source slice is still ACTIVE.

The following two events are possible causes.

#### (Cause a')

A disk component of the copy source failed to operate properly, and an I/O error occurred on the copy source slice.

#### (Cause b')

A component other than the copy source disk (such as an I/O adapter, an I/O cable, an I/O controller, a power supply, and a fan) failed to operate properly during synchronization copying, and an I/O error occurred on the copy source slice.

For details about determining whether the status is relevant to one of these events and identifying the physical disk name of a faulty disk, see [Explanation] and procedure 1) of [Resolution] described in "(1) Mirror slice configuring the mirror volume is in INVALID status."

### Resolution

First examine the physical disk abnormalities referring to disk driver log messages and so on. Then contact your local customer support and locate the disabled or faulty part.

If the possible cause is (Cause b'), shut down the system once, repair the disabled part, and reboot the system. Consequently, synchronization copying is performed and the mirroring status is restored.

If the possible cause is (Cause a'), take the following procedures for restoration.

The following illustrates restoration procedures when the class name is Class1, the volume name is Volume1, the name of a faulty disk of the copy source is Disk1, and the name of a disk of the copy destination is Disk2 as examples.

```
# sdxinfo -S -o Volume1
OBJ CLASS GROUP DISK VOLUME STATUS
-----
slice Class1 Group1 Disk1 Volume1 ACTIVE
slice Class1 Group1 Disk2 Volume1 INVALID

# sdxinfo -G -o Volume1
OBJ NAME CLASS DISKS BLKS FREEBLKS SPARE
-----
group Group1 Class1 Disk1:Disk2 17596416 17498112 0

# sdxinfo -D -o Volume1 -e long
OBJ NAME TYPE CLASS GROUP DEVNAM DEVBLKS FREEBLKS DEVCONNECT STATUS E
-----
disk Disk1 mirror Class1 Group1 sda 17596416 * node1:node2 ENABLE 1
disk Disk2 mirror Class1 Group1 sdb 17596416 * node1:node2 ENABLE 0
```

1) Exit applications using the volume.

2) Unmount the file system on the volume when it has been mounted.  
In this example, the volume has been used as an ext3 file system.

```
# umount /dev/sfdsk/Class1/dsk/Volume1
```

Depending on the part or the severity of failure in disks that compose the volume, the umount command may fail due to an I/O error. In this event, unmount the file system performing steps 2-1) through 2-3).

2-1) If class Class1 is registered with a cluster application, remove the cluster application.

2-2) Comment out the /dev/sfdsk/Class1/dsk/Volume1 line in the /etc/fstab file to prevent mounting of the volume after the system is rebooted.

2-3) Reboot the system.

3) Stop the volume.

```
# sdxvolume -F -c Class1 -v Volume1 -e allnodes
```

4) Restore the status of the copy destination slice in the INVALID status.

```
# sdxfix -V -c Class1 -d Disk2 -v Volume1
```

5) Verify that the restored copy destination slice is in the STOP status and the copy source slice is the INVALID status now.

```
# sdxinfo -S -o Volume1
OBJ CLASS GROUP DISK VOLUME STATUS
----
slice Class1 Group1 Disk1 Volume1 INVALID
slice Class1 Group1 Disk2 Volume1 STOP
```

6) Start the volume.

```
# sdxvolume -N -c Class1 -v Volume1
```

7) The consistency of volume data may have lost. Restore the backup data or perform repair using the fsck(8) command if necessary.



**Note**

If the I/O error occurred on the copy source slice during resynchronization copying after the system down, restoration may possibly be performed with the fsck(8) command.

When step 2-2) was performed, undo the edit that was made in the /etc/fstab file.

When step 2-1) was performed, re-create the cluster application removed in step 2-1).

8) Remove the faulty copy source disk from GDS management to make it a replaceable status.

```
# sdxswap -O -c Class1 -d Disk1
```

9) Swap the faulty copy source.

10) Put the swapped disk back in control of GDS management to make it available.

```
# sdxswap -I -c Class1 -d Disk1
```

### (3) Slice configuring the volume is in TEMP status.

#### **Explanation**

The slice was not attached after it has been detached with the sdxslice command. Or else, you have not performed [Attach Slice] after performing [Detach Slice] from Operation Management View.

#### **Resolution**

Attach the slice again with the sdxslice command, or perform [Attach Slice] from Operation Management View as necessary.

#### **(4) Slice configuring volume is in TEMP-STOP status.**

##### ***Explanation***

The slice was not activated after it has been stopped with the sdxslice command, or the detached node is not current node.  
Or else, you have not performed [Stop Slice] after performing [Activate Slice] from Operation Management View.

##### ***Resolution***

Activate slice or take over slice with the sdxslice command as needed. Or, perform [Activate Slice] from Operation Management View.

#### **(5) Slice configuring the volume is in COPY status.**

##### ***Explanation***

In order to attach a slice, synchronization copying is currently in process.  
Or, synchronization copying is in process between master and proxy.

##### ***Resolution***

Wait until synchronization copying is complete. Note that a slice in the process of synchronization copying will not restrict you from accessing an active volume.

#### **(6) Slice configuring the volume is in NOUSE status.**

##### ***Explanation***

When the status of disk related to slice is either in DISABLE or SWAP status, the slice becomes NOUSE to inhibit slice operation.

##### ***Resolution***

Recover disk in DISABLE or SWAP status. For details, see "F.1.2 Disk Status Abnormality."



## F.1.2 Disk Status Abnormality

---

### (1) Disk is in DISABLE status.

#### **Explanation**

Disk ID information is checked when booting the system. If the ID information is regarded invalid at the time, the disk will become DISABLE.

There are three reasons that may cause this DISABLE status.

(Cause a)

You changed the I/O cable connection and booted the system.

(Cause b)

You swapped the disk and booted the system.

(Cause c)

Disk failure occurred.

Resolutions are described for each of the above causes.

#### **Resolution**

a) If the possible cause is (Cause a), shut down the system, connect the I/O cables correctly and boot again.

b) If the possible cause is (Cause b), shut down the system, return the original disk and boot again. If swapping the disks is necessary, follow the procedures and swap the disks. For procedures on disk swapping, see "D.8 sdxswap - Swap disk" and "5.3.4 Disk Swap."

c) If the possible cause is (Cause c), follow the procedures and swap the disks.

### (2) Disk is in SWAP status.

#### **Explanation**

If the sdxswap -O command is executed, or [Swap Physical Disk] is performed from Operation Management View, the disk will be in SWAP status.

#### **Resolution**

Complete the disk swap and restore the disk by either executing the sdxswap -I command, or by performing [Restore Physical Disk] from Operation Management View.

### (3) Disk is in I/O error status.

#### **Explanation**

If an I/O error occurs, 1 (one) is given as a value in the E field displayed with the sdxinfo -e long command and the disk icon in GDS Management View turns red. Here, the disk is in ENABLE (enabled) status.

#### **Resolution**

Check the statuses of volumes and slices relative to the disk with an I/O error, and perform restoration, referring to " F.1.3 Volume Status Abnormality " or " F.1.1 Slice Status Abnormality." The I/O error status is removed, during the restoration by:

- Executing [Swap Physical Disk] in GDS Management View for the relevant disk
- Executing the sdxswap -O command for the relevant disk
- Normally ending synchronization copying of volumes relative to the relevant disk

If none of the operations above are performed during the recovery and the I/O error remains unremoved, after restoring hardware and such, use the sdxfix -D command to remove the I/O error status.

```
# sdxfix -D -c class name -d disk name -e online
```

## F.1.3 Volume Status Abnormality

### (1) Mirror volume is in INVALID status.

#### Explanation

You can confirm the status of the volume as shown below.

# sdxinfo -V -o Volume1									
OBJ	NAME	CLASS	GROUP	SKIP	JRM	1STBLK	LASTBLK	BLOCKS	STATUS
volume	*	Class1	Group1	*	*	0	65535	65536	PRIVATE
volume	Volume1	Class1	Group1	off	on	65536	17596415	17530880	INVALID

In this example, volume Volume1 that exists in the highest level group Group1 is in INVALID status, as shown in the STATUS field.

If none of the mirror slices consisting the mirror volume contains valid data (ACTIVE or STOP), the mirror volume becomes INVALID. You cannot start a volume in INVALID status.

There are two reasons that may cause this INVALID status.

**(Cause a)** Disk is in DISABLE status.

**(Cause b)** Master-proxy relationship was cancelled forcibly while master data was being copied to proxy.

#### Resolution

1) Confirm that there is a disk in DISABLE status within the group with which the volume is associated as follows.

#### (Example A1)

# sdxinfo -G -o Volume1						
OBJ	NAME	CLASS	DISKS	BLKS	FREEBLKS	SPARE
group	Group1	Class1	Disk1:Disk2	17596416	0	0

# sdxinfo -D -o Volume1								
OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk1	mirror	Class1	Group1	sda	17596416	node1:node2	ENABLE
disk	Disk2	mirror	Class1	Group1	sdb	17596416	node1:node2	DISABLE

In this example, disks Disk1 and Disk2 are connected to the highest level mirror group Group1. The disk Disk2 is in DISABLE status as shown in the STATUS field.

#### (Example B1)

# sdxinfo -G -o Volume1						
OBJ	NAME	CLASS	DISKS	BLKS	FREEBLKS	SPARE
group	Group1	Class1	Group2:Group3	35127296	17530880	0
group	Group2	Class1	Disk1:Disk2	35127296	*	0
group	Group3	Class1	Disk3:Disk4	35127296	*	0

# sdxinfo -D -o Volume1								
OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk1	stripe	Class1	Group2	sda	17596416	node1:node2	ENABLE
disk	Disk2	stripe	Class1	Group2	sdb	17596416	node1:node2	DISABLE
disk	Disk3	stripe	Class1	Group3	sdc	17682084	node1:node2	ENABLE
disk	Disk4	stripe	Class1	Group3	sdd	17682084	node1:node2	ENABLE

In this example, lower level stripe groups, Group2 and Group3 are connected to the highest level mirror group Group1. Disk Disk2 which is connected to Group2 is in DISABLE status as shown in the STATUS field.

2) If the possible cause is (Cause a), restore the disk by following the procedures in " F.1.2 Disk Status Abnormality."

3) From the disks and lower level groups connected to the highest level mirror group, determine the disk or lower level group to which the slice you will use to recover data belongs. Then, execute the sdxfix command to recover data.

**(Example A3)**

```
# sdxfix -V -c Class1 -d Disk1 -v Volume1
```

In this example, Volume1 is recovered after a slice in disk Disk1.

**(Example B3)**

```
# sdxfix -V -c Class1 -g Group3 -v Volume1
```

In this example, Volume1 is recovered after a slice in lower level stripe group Group3.

4) Start the volume.

```
# sdxvolume -N -c Class1 -v Volume1 -e nosync
```

5) Access Volume1 and check its contents. Restore backup data or run fsck to regain data integrity as necessary.

6) Perform synchronization copying on volume.

```
# sdxcopy -B -c Class1 -v Volume1
```

**(2) Single volume is in INVALID status.**

**Explanation**

You can confirm the status of the volume as shown below.

```
# sdxinfo -V -o Volume1
```

OBJ	NAME	CLASS	GROUP	SKIP	JRM	1STBLK	LASTBLK	BLOCKS	STATUS
volume	*	Class1	Disk1	*	*	0	32767	32768	PRIVATE
volume	Volume1	Class1	Disk1	off	on	32768	65535	32768	INVALID
volume	*	Class1	Disk1	*	*	65536	8421375	8355840	FREE

In the example, the single volume Volume1 that exists on single disk Disk1 is in INVALID status, as shown in the STATUS field.

You cannot start a volume in INVALID status.

There are two reasons that may cause this INVALID status.

**(Cause a)** Single disk is in DISABLE status. In this case, the single slice becomes NOUSE status.

**(Cause b)** Master-proxy relationship was cancelled forcibly while master data was being copied to proxy.

**Resolution**

1) Confirm that the single disk is in DISABLE status as shown below.

```
# sdxinfo -D -o Volume1
```

OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk1	single	Class1	*	sda	8355840	node1:node2	DISABLE

In this example, the single disk Disk1 is in DISABLE status, as shown in the STATUS field.

2) If the possible cause is (Cause a), restore the disk by following the procedures given in section " F.1.2 Disk Status Abnormality."

3) Execute the sdxfix command to recover the single volume's data.

```
# sdxfix -V -c Class1 -d Disk1 -v Volume1
```

4) Start the volume.

```
# sdxvolume -N -c Class1 -v Volume1
```

5) Access Volume1 and check its content. Restore backup data or run the fsck command to regain data integrity as necessary.

### (3) Stripe volume or volume in concatenation group is in INVALID status.

#### Explanation

You can confirm the status of the volume as shown below.

```
# sdxinfo -V -o Volume1
```

OBJ	NAME	CLASS	GROUP	SKIP	JRM	1STBLK	LASTBLK	BLOCKS	STATUS
volume	*	Class1	Group1	*	*	0	65535	65536	PRIVATE
volume	Volume1	Class1	Group1	off	on	65536	17596415	17530880	INVALID

In this example, volume Volume1 that exists in the highest level group Group1 is in INVALID status, as shown in the STATUS field.

If any of the disks related to volume is in DISABLE status, the slices consisting that volume become NOUSE status, and the volume becomes INVALID. You cannot start a volume in INVALID status.

#### Resolution

1) You can confirm the status of the disk related to the volume as shown below.

```
# sdxinfo -G -o Volume1 -e long
```

OBJ	NAME	CLASS	DISKS	BLKS	FREEBLKS	SPARE	MASTER	TYPE	WIDTH
group	Group1	Class1	Group2:Group3	70189056	65961984	*	*	stripe	32
group	Group2	Class1	Disk1:Disk2	35127296	*	*	*	concat	*
group	Group3	Class1	Disk3:Disk4	35127296	*	*	*	concat	*

```
# sdxinfo -D -o Volume1
```

OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk1	concat	Class1	Group2	sda	17596416	node1:node2	ENABLE
disk	Disk2	concat	Class1	Group2	sdb	17596416	node1:node2	DISABLE
disk	Disk3	concat	Class1	Group3	sdc	17682084	node1:node2	ENABLE
disk	Disk4	concat	Class1	Group3	sdd	17682084	node1:node2	ENABLE

In this example, the lower level concatenation groups Group2 and Group3 are connected to the highest level stripe group Group1, and Disk2 connected to Group2 is in DISABLE status as shown in the STATUS field.

2) Follow the procedures in "F.1.2 Disk Status Abnormality" and restore the disk status.

3) Execute the sdxfix command to recover the volume's data. With -g option, indicate the highest level group name (in this example, Group1).

```
# sdxfix -V -c Class1 -g Group1 -v Volume1
```

4) Start the volume.

```
# sdxvolume -N -c Class1 -v Volume1
```

5) Access Volume1 and check its content. Restore backup data or run the fsck command to regain data integrity as necessary.

#### (4) Master volume is in INVALID status.

##### Explanation

If the copying process fails while copying data from the proxy volume to the master volume because of an I/O error or such, the status of the master volume to which the data is being copied becomes INVALID.

##### Resolution

1) Check if there is a DISABLE status disk in the group to which the volume belongs with the following command.

```
# sdxinfo -D -o Volume1
```

OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk1	mirror	Class1	Group1	sda	8421376	*	ENABLE
disk	Disk2	mirror	Class1	Group1	sdb	8421376	*	DISABLE

In this example, Disk2 is in DISABLE status.

If there is a disk in DISABLE status, see section "(1) Disk is in DISABLE status" in " F.1.2 Disk Status Abnormality," and check which of the causes listed in that section applies. If the possible cause is (Cause a) or (Cause b), follow the procedures and restore the disk.

2) Follow the procedures given in section "(1) Mirror slice configuring the mirror volume is in INVALID status" in " F.1.1 Slice Status Abnormality," and check if there is a disk hardware abnormality. If there is, identify the faulty part. When the abnormality is caused by a failed or defective non-disk component, repair the faulty part.

3) Procedures to restore the data for different scenarios are given below.

##### Non-disk component failure

- When recovering data using the proxy volume:  
-> Follow procedures (a) to restore.
- When recovering data using backup data on media such as tapes:  
-> Follow procedures (b) to restore.

##### When caused by a disk component failure:

- When the disk does not belong to master group:
  - some disks connected to the group have failed  
-> Follow procedures (c) to restore.
  - all disks connected to the group have failed  
-> Follow procedures (d) to restore.
- When the disk belongs to master group:  
-> Follow procedures (e) to restore.

##### a) Procedures to recover master volume data using proxy volume.

a1) In order to check if the proxy volume that will be used to recover data is separated from the master volume, execute the sdxinfo -V -e long command, and check the PROXY field.

a2) If the proxy volume is not separated, execute the following command.

```
# sdxproxy Part -c Class1 -p Volume2
```

a3) Exit all applications accessing the proxy volume. When using the proxy volume as a file system, execute umount. When using the proxy volume as a file system, execute umount.

a4) If the proxy volume is started, execute the following command.

```
# sdxvolume -F -c Class1 -v Volume2
```

a5) Recover master volume data using the proxy volume's data.

```
# sdxproxy RejoinRestore -c Class1 -p Volume2
```

**b) Procedures to recover data using backup data.**

b1) When the volume is in INVALID status, you must first change it to STOP status. Decide on the disk (slice) you wish to use to recover data, and execute the sdxfix command.

```
# sdxfix -V -c Class1 -d Disk1 -v Volume1
```

In this example, Volume 1 is restored after a slice in Disk 1.

b2) When the volume to be restored is stopped, start it with the following command.

```
# sdxvolume -N -c Class1 -v Volume1 -e nosync
```

b3) Access the volume to be restored and check its contents. Restore backup data or run fsck to regain data integrity as necessary.

b4) When mirroring is configured with the volume, perform synchronization copying.

```
# sdxcopy -B -c Class1 -v Volume1
```

**c) Procedures to swap some disks connected to the group.**

c1) If you restore the INVALID master volume later using data of a proxy volume related to the master volume, or use data of proxy volumes related to the master volume after restoring it, part the proxy volumes using the sdxproxy Part command.

```
# sdxproxy Part -c Class1 -p Volume2
```

c2) When there is a volume in INVALID status in the group, change it to STOP status with the sdxfix -V command. -d option indicates the disk without abnormality.

```
# sdxfix -V -c Class1 -d Disk1 -v Volume1
```

c3) Follow the procedures and swap the disks. For details, see the "D.8 sdxswap - Swap disk" and "5.3.4 Disk Swap."

c4) Recover the master volume data. If data will be recovered using the proxy volume, follow procedures described in (a). If data will be recovered using backup data on media such as tapes, follow procedures described in (b).

**d) Procedures to swap all disks connected to the group.**

d1) Exit all applications accessing the master volume and the proxy volume that will be used to recover data. When using the proxy volume or the master volume as a file system, execute unmount.

d2) Stop the master volume and proxy volume in (d1).

```
# sdxvolume -F -c Class1 -v Volume1  
# sdxvolume -F -c Class1 -v Volume2
```

d3) Execute the sdxproxy RejoinRestore command and restore the master volume data using proxy volume in (d1). If the command terminates normally and the master volume is not in INVALID status, restoration process is complete, and you do not need to perform procedures (d4) and after.

```
# sdxproxy RejoinRestore -c Class1 -p Volume2
```

d4) Execute the sdxproxy Swap command and swap the slices of the master volume with the proxy volume in (d1).

```
# sdxproxy Swap -c Class1 -p Volume2
```

d5) By performing procedure (d4), the status of master volume will not be in INVALID status, and the status of the proxy volume becomes INVALID. Follow the procedures given in section "(5) Proxy volume is in INVALID status" in "F.1.3 Volume Status Abnormality," and restore the proxy volume in INVALID status.

d6) Execute the sdxproxy Swap command and swap the slices of the master volume and the proxy volume you swapped in procedure (d4).

```
# sdxproxy Swap -c Class1 -p Volume2
```

**e) Procedures to swap disks connected to the master group.**

e1) Exit all applications accessing the master group, and volumes in the proxy group that will be used to recover data. When using the volume as a file system, execute unmount.

e2) Stop all volumes in the master group and the proxy group in (e1).

```
# sdxvolume -F -c Class1 -v Volume1
# sdxvolume -F -c Class1 -v Volume2
```

e3) Execute the sdxproxy RejoinRestore command and restore the master group data using the proxy group in (e1). If the command terminates normally and all master volumes are not in INVALID status, restoration process is complete, and you do not need to perform procedures (e4) and after.

```
# sdxproxy RejoinRestore -c Class1 -p Volume2
```

e4) Execute the sdxproxy Swap command and swap the slices of the master group and the proxy group in (e1).

```
# sdxproxy Swap -c Class1 -p Group2
```

e5) By performing procedure (e4), the master volume will not be in INVALID status and the status of the proxy volume becomes INVALID. Follow the procedures given in section "(5) Proxy volume is in INVALID status" in "F.1.3 Volume Status Abnormality," and restore the proxy volume in INVALID status.

e6) Execute the sdxproxy Swap command and swap the slices of the master group and the proxy group you swapped in procedure (e4).

```
# sdxproxy Swap -c Class1 -p Group2
```

**(5) Proxy volume is in INVALID status.**

**Explanation**

If the copying process fails while copying data from the master volume to the proxy volume because of an I/O error or such, the status of the proxy volume to which the data is being copied becomes INVALID.

**Resolution**

1) Check if there is a DISABLE status disk in the group to which the volume belongs with the following command.

```
# sdxinfo -D -o Volume1
```

OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk1	mirror	Class1	Group1	sda	8421376	*	ENABLE
disk	Disk2	mirror	Class1	Group1	sdb	8421376	*	DISABLE

In this example, Disk2 is in DISABLE status.

If there is a disk in DISABLE status, see section "(1) Disk is in DISABLE status" in "F.1.2 Disk Status Abnormality," and check which of the causes (There are three causes a, b and c listed.) listed in that section applies. If it is due to (Cause a) or (Cause b), follow the procedures and restore the disk.

2) Follow the procedures given in "(1) Mirror slice configuring the mirror volume is in INVALID status" in "F.1.1 Slice Status Abnormality," and check if there is a disk hardware abnormality. If there is, identify the faulty part. When the abnormality was caused by a failed or defective non-disk component, repair the faulty part.

3) Procedures to restore the data for different scenarios are given below.

- When caused by a non-disk component failure:
  - > Follow procedures (a) to restore.
- When caused by a disk component failure:
  - disk does not belong to proxy group
    - some disks connected to the group have failed
      - > Follow procedures (b) to restore.
    - When all disks connected to the group have failed:
      - > Follow procedures (c) to restore.
  - When the disk belongs to proxy group:
    - > Follow procedures (d) to restore.

**a) Procedures to recover proxy volume data using the master volume.**

a1) In order to check if the proxy volume is separated from the master volume, execute the `sdxinfo -V -e` long command, and check the PROXY field.

a2) If the proxy volume is not separated, execute the following command.

```
# sdxproxy Part -c Class1 -p Volume2
```

a3) Rejoin the proxy volume with the master volume.

```
# sdxproxy Rejoin -c Class1 -p Volume2
```

**b) Procedures to swap some disks connected to the group.**

b1) Cancel the relationship with master volume using the `sdxproxy Break` command.

```
# sdxproxy Break -c Class1 -p Volume2
```

b2) Separate the volumes that are in INVALID status in the group with the `sdxfix -V` command, and change them to STOP status. `-d` option indicates the disk without abnormality.

```
# sdxfix -V -c Class1 -d Disk1 -v Volume2
```

b3) Follow the procedures and swap the disks. For details, see the "D.8 sdxswap - Swap disk," or section "5.3.4 Disk Swap."

b4) Join the master and the proxy again with the `sdxproxy Join` command.

```
# sdxproxy Join -c Class1 -m Volume1 -p Volume2
```

**c) Procedures to swap all disks connected to the group.**

c1) Cancel the relationship with the master using the `sdxproxy Break` command.

```
# sdxproxy Break -c Class1 -p Volume2
```

c2) Exit all applications accessing the volume in the group. When using the volume as a file system, execute `umount`.

c3) Stop all volumes in the group.

```
# sdxvolume -F -c Class1 -v Volume2
```

c4) Check the volume configuration of the group (such as volume names and sizes) with the `sdxinfo` command, and keep a note of it.

c5) Remove all volumes in the group.

```
# sdxvolume -R -c Class1 -v Volume2
```

c6) Follow the procedures and swap the disks. For details, see the "D.8 sdxswap - Swap disk," or section "5.3.4 Disk Swap."

c7) Create the volume that you removed in procedure (c5) again.

```
# sdxvolume -M -c Class1 -g Group2 -v Volume2 -s size
```

c8) Stop the volume you created in procedure (c7).

```
# sdxvolume -F -c Class1 -v Volume2
```

c9) Join the master volume and the proxy volume again, with the `sdxproxy Join` command.

```
# sdxproxy Join -c Class1 -m Volume1 -p Volume2
```

**d) Procedures to swap disks connected to the proxy group.**

d1) Cancel the relationship with the master using the `sdxproxy Break` command.

```
# sdxproxy Break -c Class1 -p Group2
```

d2) Exit all applications accessing the volume in the group. When using the volume as a file system, execute `umount`.

d3) Stop all volumes in the group.

```
# sdxvolume -F -c Class1 -v Volume2
```

d4) Remove all volumes in the group.

```
# sdxvolume -R -c Class1 -v Volume2
```

d5) Follow the procedures and swap the disks. For details, see the "D.8 sdxswap - Swap disk," or



section "5.3.4 Disk Swap."

d6) Join the master group and the proxy group again with the sdxproxy Join command.

```
# sdxproxy Join -c Class1 -m Group1 -p Group2 -a Volume1=Volume2:on
```

## **(6) Volume is in STOP status.**

### ***Explanation***

Usually, a volume is automatically started when the system is booted and becomes ACTIVE. The status will change to STOP, when you stop the volume with the sdxvolume -F command, or perform [Stop Volume] from Operation Management View. Accessing a volume in STOP status will result in an I/O error.

### ***Resolution***

Start the volume with the sdxvolume -N command, or perform [Start Volume] from Operation Management View as necessary.

## **(7) I/O error occurs although mirror volume is in ACTIVE status.**

### ***Explanation***

A mirror volume consists of multiple slices, and in an event of an I/O error, the crashed slice will be detached. Therefore, accessing the volume will complete normally.

However, when an I/O error occurs when only one slice is ACTIVE amongst those configuring the volume, accessing the volume will result in an error. At such time, the status of the slice and the volume remains ACTIVE.

Probable situations resulting in such a problem will be described using a two-way multiplex mirroring configuration, where two disks or two lower level groups are connected to a group. As an example, means to circumvent such problems will also be described.

#### **(Situation 1)**

One of the slices was detached with the sdxslice -M in order to backup volume data. While accessing the volume, an I/O error occurred with the other slice.

#### **(Prevention 1)**

Before executing the sdxslice -M command, connect a reserved disk and temporarily configure a three-way multiplex mirroring, or make the mirrored volume available for backup.

#### **(Situation 2)**

While restoring a slice with an I/O error, an I/O error also occurred on another slice.

#### **(Prevention 2)**

By securing a spare disk within the class, effects due to delay in restoring the slice will be avoided to a certain degree

### ***Resolution***

1) Identify the cause of I/O error occurrence in the last ACTIVE slice, by referring to the disk driver log message.

Resolutions are described below assuming the following three circumstances:

- a) Error occurred due to a disk component failure. Will attempt recovery using backup data.
- b) Error occurred due to a disk component failure. Will attempt data recovery from a slice in INVALID status.
- c) Error occurred due to a failed or a defective non-disk component failure.

2a) When the error was caused by a disk component failure, no slice with valid data exists. Restore data from the backup data following the procedures given below.

3a) Exit the application accessing the volume. When the volume is used as a file system, execute the unmount command.

When I/O error occurs on the unmount command, execute -f option of the unmount command.

4a) Stop the volume with the sdxvolume command.

```
# sdxvolume -F -c Class1 -v Volume1
```

5a) If there is a TEMP status slice within the volume, attempt recovery following the procedures given in "F.1.1 Slice Status Abnormality."

6a) If there is a NOUSE status slice within the volume, attempt recovery following the procedures given in "F.1.1 Slice Status Abnormality."

7a) Record the volume size which can be checked as follows.

# sdxinfo -V -o Volume1									
OBJ	NAME	CLASS	GROUP	SKIP	JRM	1STBLK	LASTBLK	BLOCKS	STATUS
volume	*	Class1	Group1	*	*	0	32767	32768	PRIVATE
volume	Volume1	Class1	Group1	off	on	32768	4161535	4128768	STOP

In this example, the volume size would be 4128768 blocks given in Volume1 BLOCKS field.

8a) Remove the volume with the sdxvolume command.

```
# sdxvolume -R -c Class1 -v Volume1
```

9a) Swap disks following the procedures given in "5.3.4 Disk Swap" and "D.8 sdxswap - Swap disk."

10a) Create a volume with the sdxvolume command again. For the *number\_of\_sectors*, use the size recorded in (7a), in this example, 4128768.

```
# sdxvolume -M -c Class1 -g Group1 -v Volume1 -s number_of_sectors
```

11a) Finally, restore the backup data to Volume1.

2b) When the error was caused by a disk failure, and no backup data exists, or even if it did, if the data is too old, restore data from the detached INVALID status slice, following the procedures given below.

3b) Exit the application accessing the volume. When the volume is used as a file system, execute the unmount command.

When I/O error occurs on the unmount command, execute -f option of the unmount command.

4b) Stop the volume with the sdxvolume command.

```
# sdxvolume -F -c Class1 -v Volume1
```

5b) If there is a TEMP status slice within the volume, attempt recovery following the procedures given in "F.1.1 Slice Status Abnormality."

6b) If there is a NOUSE status slice within the volume, attempt recovery following the procedures given in "F.1.1 Slice Status Abnormality."

7b) Determine the original mirror slice after the volume is recovered. Then, execute the sdxfix command.

**(Example 1)**

```
# sdxfix -V -c Class1 -d Disk2 -v Volume1
```

In this example, data is recovered from a mirror slice in the disk Disk2 which is connected to the highest level mirror group.

### (Example 2)

```
# sdxfix -V -c Class1 -g Group2 -v Volume1
```

In this example, data is recovered from a mirror slice in the lower level group Group2 which is connected to the highest level mirror group.

8b) Start the volume.

```
# sdxvolume -N -c Class1 -v Volume1 -e nosync
```

9b) Create backup of Volume1 and regain data integrity by running fsck as necessary.

10b) Lastly, swap disks following the procedures given in "5.3.4 Disk Swap" and "D.8 sdxswap - Swap disk."

2c) When the error was caused by a failed or defective non-disk component, the slice with valid data exists within the disk. Shut down the system and recover the failed component and reboot your system. Mirroring status will be recovered by automatic synchronization copying.

## (8) An I/O error occurs on a single volume.

### Explanation

Since a single volume consists of only one slice, accessing the volume at the time of an I/O error will result in an error. However, the status of slice and volume will remain ACTIVE.

### Resolution

1) Identify the cause of I/O error occurrence by referring to the disk driver log message. How to resolve the problem is described in two cases: (a) a disk component failure occurs and backup data is used for restoration and (b) a non-disk component failure occurs.

2a) In the event of a disk component failure, there will be no slice with valid data. Follow the procedures below and restore the data using the backup data. In this example, Disk1 (sda) has a failure.

```
# sdxinfo -D -o Disk1
OBJ  NAME  TYPE  CLASS  GROUP  DEVMAM  DEVBLKS  DEVCONNECT  STATUS
-----
disk  Disk1  single  Class1  *      sda      8493876  node1:node2  ENABLE
```

3a) Search the volumes within the faulty disk using the sdxinfo command. And record the volume size,

```
# sdxinfo -V -o Disk1
OBJ  NAME  CLASS  GROUP  SKIP  JRM  1STBLK  LASTBLK  BLOCKS  STATUS
-----
volume  *      Class1  Disk1  *      *      0      32767  32768  PRIVATE
volume  Volume1  Class1  Disk1  off   on    32768  65535  32768  ACTIVE
volume  Volume2  Class1  Disk1  off   on    65536  4194303  4128768  ACTIVE
volume  *      Class1  Disk1  *      *      4194304  8421375  4227072  FREE
```

In this example, Volume1 and Volume2 are within the faulty Disk1. The size of Volume1 would be 32,768 blocks as shown in the BLOCKS field. The size of Volume2 would be 4,128,768 blocks as shown in the BLOCKS field.

4a) Exit the application accessing the volume. When the volume is used as a file system, execute the unmount command. When I/O error occurs on the unmount command, execute -f option of the unmount command.

5a) Stop the volume with the sdxvolume command.

```
# sdxvolume -F -c Class1 -v Volume1,Volume2
```

6a) Remove the volumes with the sdxvolume command.

```
# sdxvolume -R -c Class1 -v Volume1  
# sdxvolume -R -c Class1 -v Volume2
```

7a-1) Before swapping the disks, execute the following command.

```
# sdxswap -O -c Class1 -d Disk1
```



**Note**

If the disk is the only remaining disk in the disk class, it will result in an error as shown below. In such case, follow the procedures 7a-2, 8a-2 and 9a-2.

SDX:sdxswap: ERROR: Disk1: The last ENABLE disk in class cannot be swapped

8a-1) Swap the disks.

9a-1) After swapping the disks, execute the following command.

```
# sdxswap -I -c Class1 -d Disk1
```

7a-2) Before swapping the disks, execute the following command.

Caution: If no error occurs after performing 7a-1), you do not have to perform procedures in 7a-2), 8a-2) and 9a-2).

```
# sdxdisk -R -c Class1 -d Disk1
```

8a-2) Swap the disks.

9a-2) After swapping the disks, execute the following command.

```
# sdxdisk -M -c Class1 -d sda=Disk1:single
```

10a) Create volumes with the sdxvolume command again. For the `-s` option, use the size recorded in (3a), in this example.

```
# sdxvolume -M -c Class1 -d Disk1 -v Volume1 -s 32768  
# sdxvolume -M -c Class1 -d Disk1 -v Volume2 -s 4128768
```

11a) Finally, restore the backup data to Volume1 and Volume2.

2b) In the event of a non-disk component failure, reboot your system after shutting down the system and recovering the failed component. There is no need to restore data, since slice data is valid.

## (9) An I/O error occurs on a stripe volume or a volume in a concatenation group.

### Explanation

Since a stripe volume or a volume within a concatenation group consists of only one slice, accessing the volume at the time of an I/O error will also result in an error. However, the status of slice and volume will remain ACTIVE.

### Resolution

1) Identify the cause of I/O error occurrence by referring to the disk driver log message. You can confirm the error status of the disk related to volume and the physical disk name as shown below.

```
# sdxinfo -D -o Volume1 -e long
```

OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	FREEBLKS	DEVCONNECT	STATUS	E
disk	Disk1	concat	Class1	Group2	sda	17596416	*	node1:node2	ENABLE	0
disk	Disk2	concat	Class1	Group2	sdb	17596416	*	node1:node2	ENABLE	1
disk	Disk3	concat	Class1	Group3	sdc	17682084	*	node1:node2	ENABLE	0
disk	Disk4	concat	Class1	Group3	sdd	17682084	*	node1:node2	ENABLE	0

In this example, an I/O error occurs on Disk2, as shown in the E field. The physical disk name corresponding to Disk2 is sdb, as shown in the DEVNAM field.

How to resolve the problem is described in two cases:

- (a) a disk component failure occurs, and you wish to restore the disk using the backup data
- (b) a non-disk component failure occurs.

2a) In case of a disk component failure, there will be no slices with valid data. Follow the procedures below and restore the data using the backup data.

3a) Record the configuration information of the group that was related to the failed disk using the sdxinfo command.

```
# sdxinfo -G -o Disk2 -e long
```

OBJ	NAME	CLASS	DISKS	BLKS	FREEBLKS	SPARE	MASTER	TYPE	WIDTH
group	Group1	Class1	Group2:Group3	70189056	65961984	*	*	stripe	32
group	Group2	Class1	Disk1:Disk2	35127296	*	*	*	concat	*
group	Group3	Class1	Disk3:Disk4	35127296	*	*	*	concat	*

In this example, the lower level concatenation groups Group2 and Group3 are connected to the highest level stripe group Group1. The disks Disk1 and Disk2 are connected to Group2, and the disks Disk3 and Disk4 are connected to Group3. The stripe width for Group1 is 32 blocks.

4a) Search the volumes that exist in the highest level group that are related to the faulty disk using the sdxinfo command.

```
# sdxinfo -V -o Disk2
```

OBJ	NAME	CLASS	GROUP	SKIP	JRM	1STBLK	LASTBLK	BLOCKS	STATUS
volume	*	Class1	Group1	*	*	0	65535	65536	PRIVATE
volume	Volume1	Class1	Group1	*	*	65536	98303	32768	ACTIVE
volume	Volume2	Class1	Group1	*	*	98304	4227071	4128768	ACTIVE
volume	*	Class1	Group1	*	*	4227072	70189055	65961984	FREE

In this example, Volume1 and Volume2 exist in the highest level group Group1, that is related to the faulty disk Disk2. The size of Volume1 is 32768 blocks, and the size of Volume2 is 4128768 blocks as shown in the BLOCKS field.

5a) Exit the application accessing the volume. When the volume is used as a file system, execute unmount command. When I/O error occurs on unmount command, execute -f option of unmount command.

6a) Stop the volume with the sdxvolume command.

```
# sdxvolume -F -c Class1 -v Volume1,Volume2
```

7a) Remove the volumes with the sdxvolume command.

```
# sdxvolume -R -c Class1 -v Volume1
# sdxvolume -R -c Class1 -v Volume2
```

8a) Disconnect the faulty disk from the group. If the group is in a hierarchical structure, disconnect from the higher group in descending order.

```
# sdxgroup -D -c Class1 -h Group1 -l Group2
# sdxdisk -D -c Class1 -g Group2 -d Disk2
```

In this example, the faulty disk Disk2 is connected to Group2, and Group2 is connected to Group1. Therefore, you should disconnect Group2 first, and then Disk2.

9a-1) Before swapping the disks, execute the following command.

```
# sdxswap -O -c Class1 -d Disk2
```

Caution: When the disk is the last disk in disk class, it will result in the following error. In such case, follow the procedures 9a-2), 10a-2) and 11a-2).

SDX:sdxswap: ERROR: Disk2: The last ENABLE disk in class cannot be swapped

10a-1) Swap the disks.

11a-1) After swapping the disks, execute the following command.

```
# sdxswap -I -c Class1 -d Disk2
```

9a-2) Before swapping the disks, execute the following command.

Caution: If no error occurs after performing 9a-1), you do not have to perform procedures in 9a-2), 10a-2) and 11a-2).

```
# sdxdisk -R -c Class1 -d Disk2
```

10a-2) Swap the disks.

11a-2) After swapping the disks, execute the following command.

```
# sdxdisk -M -c Class1 -d c1t2d0=Disk2
```

12a) Connect the swapped disk to the group, referring to the group information recorded in (3a). If the groups were in a hierarchical structure, connect the groups in an ascending order.

```
# sdxdisk -C -c Class1 -g Group2 -d Disk2
# sdxgroup -C -c Class1 -h Group1 -l Group2 -a type=stripe,width=32
```

13a) Create volumes with the sdxvolume command again. For the -s option, use the size recorded in (4a), in this example, 32768 and 4128768.

```
# sdxvolume -M -c Class1 -g Group1 -v Volume1 -s 32768 -a pslice=off
# sdxvolume -M -c Class1 -g Group1 -v Volume2 -s 4128768 -a pslice=off
```

14a) Finally, restore the backup data to Volume1 and Volume2.

2b) In case of a non-disk component failure, reboot your system after shutting down the system and recovering the failed component. There is no need to restore data, since the slice data is valid.

## F.1.4 Class Status Abnormality

### (1) Class becomes closed status during operation.

#### Explanation

The class becomes closed when the number of configuration databases which store information on object configuration and object status within a class is insufficient, or when the communication error between nodes occurs in a cluster environment.

All objects within a closed class are inaccessible.

An Insufficient number of configuration databases will occur under the following conditions:

1. When there are no disks that can be accessed normally, if there are two or less disks in ENABLE status.
2. When there are one or less disks that can be accessed normally, if there are three to five disks in ENABLE status.
3. When there are two or less disks that can be accessed normally, if there are more than six disks in ENABLE status.

#### Resolution

1) You can check whether or not a class was closed during operation as follows. Do not reboot the system or restart sdxservd daemon, as it will make the checking impossible.

# /etc/opt/FJSVsdx/bin/sdxdcdown							
CLASS	DOWN	REASON	NDK	NEN	NDB	NLDB	DEVNAM
Class1	no	-	10	10	8	0	sda:sdb:sdc:sdd:sde:sdf:sdg:sdh
Class2	yes	Comm	10	10	8	0	sdi:sdj:sdk:sdl:sdm:sdn:sdo:sdp
Class3	yes	FewDB	10	10	1	7	sdq
Class4	yes	NoDB	10	10	0	8	-

In this example, Class2, Class3, and Class4 with "yes" in the DOWN field are closed. The cause shown in the REASON field are as follows.

(Cause 1)

Comm Communication failure between nodes.

(Cause 2)

FewDB Insufficient number of valid configuration databases.

(Cause 3)

NoDB No valid configuration database.

2) Depending on specific causes, recovery may be difficult.

First, collect the investigation material.

For information on how to collect the investigation material, see "F.2 Collecting Investigation Material."

Resolutions are described for the following two cases:

- a) closed due to a communication error
- b) closed due to an insufficient number of configuration databases

3a) In case of Cause 1, contact your local customer support.

3b) In case of Cause 2 or 3, all (or the majority) of the disks registered with class have abnormalities. You can check the disks registered with class follows.

# sdxinfo -D -c Class3								
OBJ	NAME	TYPE	CLASS	GROUP	DEVNAM	DEVBLKS	DEVCONNECT	STATUS
disk	Disk31	mirror	Class3	Group1	sda	8847360	*	ENABLE
disk	Disk32	mirror	Class3	Group1	sdb	8847360	*	ENABLE
disk	Disk33	mirror	Class3	Group2	sde	8847360	*	ENABLE
disk	Disk34	mirror	Class3	Group2	sdf	8847360	*	ENABLE
disk	Disk35	mirror	Class3	Group3	sdc	17793024	*	ENABLE
disk	Disk36	mirror	Class3	Group3	sdg	17793024	*	ENABLE
disk	Disk37	mirror	Class3	Group4	sdd	17793024	*	ENABLE
disk	Disk38	mirror	Class3	Group4	sdh	17793024	*	ENABLE
disk	Disk39	spare	Class3	Group1	sdr	17793024	*	ENABLE
disk	Disk40	spare	Class3	*	sds	17727488	*	ENABLE

In this example, ten disks from Disk31 to Disk40 are registered with Class3. Physical disk names are shown in the DEVNAM field. Identify the cause of abnormality with these physical disks by referring to disk driver log messages. The cause of abnormality could be either of the following:

- (Failure 1)  
Failed or defective non-disk component.
- (Failure 2)  
Failed disk component.

4b) Open the GDS configuration parameter file with an editor.

```
# vi /etc/opt/FJSVsdx/sdx.cf
```

Add the following line at the end of the file.

```
SDX_DB_FAIL_NUM=0
```

5b) Shut down the system.

6b) In the event of Failure 1, recover the failed non-disk component (such as an I/O adapter, an I/O cable, a controller, a power supply, a fan).

7b) Reboot the system.

8b) Confirm that objects within the class are accessible.

```
# sdxinfo -c Class3
```

If nothing is displayed, recovery was unsuccessful. You will have to contact your local customer support. If information is displayed normally, proceed with the following procedures.

9b) In case of Failure 2 when a disk component has failed, follow the procedures in "5.3.4 Disk Swap," or "D.8 sdxswap - Swap disk," and swap the disks.

10b) After completing the recovery for both Failure 1 or 2, confirm the number of valid configuration databases as described below.

```
# /etc/opt/FJSVsdx/bin/sdxdcdown
```

CLASS	DOWN	REASON	NDK	NEN	NDB	NLDB	DEVNAM
Class1	no	-	10	10	8	0	sda:sdb:sd:c:sdd:sde:sdf:sdg:sdh
Class2	no	-	10	10	8	0	sdi:sdj:sdk:sd:l:sdm:sdn:sdo:sdp
Class3	no	-	10	10	8	0	sdq:sdr:sds:sdt:sdu:sdv:sdw:sdx
Class4	no	-	10	10	8	0	sdaa:sdab:sdac:sdad:sdae:sdaf:sdag:sdah

NLDB field gives the insufficient number of configuration databases. If this value is "0," the problem is resolved. If this value is "1" or more, there are still disks that have not been recovered. In the above example, all NLDB fields display "0," indicating the successful recovery.

11b) Open the GDS configuration parameter file with an editor.

```
# vi /etc/opt/FJSVsdx/sdx.cf
```

Remove the following line added at the end in procedure (5b).

```
SDX_DB_FAIL_NUM=0
```

12b) Reboot the system.

If you cannot perform the recovery with the described procedures, contact your local customer support.



## **F.1.5 GDS Management View Abnormality**

---

In this section, how to resolve problems related to Web-Based Admin View and GDS Operation Management View is explained. For problems described in (1) to (18), see "Web-Based Admin View Operation Guide" and follow the troubleshooting procedures.

**(1) Web-Based Admin View top menu is not displayed. Or, even if it is, the screen is not displayed properly, or messages such as 0001 to 0050 are displayed.**

**(2) The screen does not change even after confirming 0002, 0003, or 0007 message. Or, the browser freezes when trying to reconnect to a previously-connected management server.**

**(3) The following message may be displayed on the management server or the Java console screen of the monitor node.**

`FJSVwvbs:OutOfMemoryError`

**(4) When trying to access the management server from browser, message 0005 appears and the top menu is not displayed, or the message itself may not appear.**

**(5) Web-Based Admin View stopped operating after changing the IP address of the public LAN.**

**(6) When using the PRIMECLUSTER Operation Management View, existing cluster nodes are not displayed on the each product's node list screen.**

**(7) The screen does not change even after confirming the dialog message.**

**(8) The following message is displayed on the browser status bar, and top menu is not displayed.**

`Applet com.fujitsu.webview.base.WebViewMain`

`error:java.lang.NoClassDefFoundError: com/sun/java/swing/JApplet`

**Or, the Operation Management View for each product can not be displayed.**

**(9) It takes more than a few minutes before the Web-Based Admin View top menu is displayed. Or, even if it is, it freezes or message "0001 Cannot call Management server" may be displayed.**

**(10) When using a Java Plug-in on a client browser, and using a mouse on the Web-Based Admin View top menu or Operation Management View for each product, the screen has problems such as displaying small flecks in the area where the mouse has been used.**

**(11) When starting or restarting the Web-Based Admin View, or rebooting the node, the message "FJSVwvbs:webview.cnf abnormal" or "Node process abnormal end." are displayed, and Web-Based Admin View does not start.**

**(12) When starting or restarting the Web-Based Admin View, message "WARNING:unrelated 'httpip' property." is displayed.**

**(13) After installing the manual package, the title menu for the manual is not displayed. Or, even after removing the manual package, the title menu for the manual is still displayed.**

**(14) When trying to start the menu for products that are the target of management from the Web-Based Admin View top menu, the message "0023 This application must be started at the following URL" is displayed.**

**(15) Start button for products that are the target of management have disappeared from the Web-Based Admin View top menu.**

**(16) Even after disconnecting from the management server, messages 0002, 0003 or 0007 are not displayed on the client.**

**(17) When trying to access the management server from the browser, message 0006 appears, and the top menu is not displayed.**

**(18) When entering the user ID and the password for authentication, error message 0016 appears, and authentication fails.**

**(19) GDS Management View cannot be started.**

#### ***Resolution***

**a. When GDS is not installed.**

Install GDS.

**b. When GDS is installed.**

Check for the following possible causes:

- Wrong URL specified.
- Wrong method caused due to specifying host name.
- Environment cannot use Swing.
- Different browser type and level than recommended.

**(20) Disk information such as mphd and mplb will not be displayed on GDS Operation Management View. Or, the disk sizes for disk units are not displayed correctly.**

#### ***Explanation***

GDS does not recognize disks properly. Possible causes are as follows.

- The power of the disk unit was turned on after the system was booted.
- The disk unit became unavailable for some kind of problem during system operation.
- The configuration of devices such as mphd and mplb was changed during system operation.

#### ***Resolution***

Perform the following steps 1) through 2).

1. Use the fdisk(8) command and check whether the OS recognizes the disk unit. If it does not recognize, identify the cause based on log messages of disk drivers stored in the /var/log/messages file and such.
2. Execute [Update Physical Disk Information] in the [Operation] menu.

## F.1.6 Cluster System Related Error

---

### (1) The PRIMECLUSTER CF `clinitreset(1M)` command ends abnormally outputting an error message # 6675.

#### Explanation

When a class exists in a cluster system, initializing the PRIMECLUSTER resource database with the PRIMECLUSTER CF `clinitreset` command results in that the `clinitreset` command fails outputting the following error message.

```
FJSVcluster: ERROR: clinitreset: 6675: Cannot run this command
because Global Disk Services has already been set up.
```

When a node containing a shadow class is rebooted because of an event such as shutdown and panic, the shadow class is deleted, but the `/dev/sfdsk/Class Name` directory is not deleted. If the `clinitreset` command is executed here, the command also fails outputting the error message as above.

#### Resolution

1. On all nodes in the cluster system, view the configuration of objects and delete a class if any exists. If a class is deleted, volume data will be lost. If necessary, back up volume data in advance.



For using GDS Management View, see "5.5 Removals."



For using commands, see "Appendix D Command Reference."

2. On all nodes in the cluster system, check whether a class directory exists in the `/dev/sfdsk` directory, and delete a class directory if any exists. The following shows an example when a directory of class `Class1` exists.  
`_adm` and `_diag` are special files used by GDS and cannot be deleted.

```
# cd /dev/sfdsk
# ls
_adm _diag Class1
# rm -rf Class1
```

### (2) Cluster applications become "Inconsistent".

#### Explanation

If a shared class is not to be used as an RMS resource, volumes included in the class are started on node startup. If a cluster application that uses those volumes are started there, the cluster application becomes "Inconsistent" because the volumes are already active. By default, classes are not to be used as RMS resources. Classes can be made available as RMS resources either by:

- Registering them in resources used by cluster applications through the Web-Based Admin View's userApplication Configuration Wizard
- Specifying them and using the `hvgdsetup -a` command

#### Resolution

Make the shared class available as an RMS resource with one of the following methods. After performing the procedures, restart the cluster application.

- If the class is not registered in resources used by the cluster application, register it through the userApplication Configuration Wizard.

- If the class is already registered in resources used by the cluster application, execute the following command.

```
# /usr/opt/reliant/bin/hvgdsetup -a Class Name
~
Do you want to continue with these processes ? y
```

### (3) The GFS Shared File System is not mounted on node startup.

#### **Explanation**

If a shared class is to be used as an RMS resource, volumes included in the class are not started on node startup. Therefore, the GFS Shared File System on those volumes is not mounted on node startup. By default, classes are not to be used as RMS resources, but they are made available as RMS resources either by:

- Registering them in resources used by cluster applications through the Web-Based Admin View's userApplication Configuration Wizard
- Specifying them and using the hvgdsetup -a command

#### **Resolution**

Take one of the following actions.

a) When using the shared class as an RMS resource, do not create the GFS Shared File System on volumes in the class, but create it on volumes in a difference class.

b) When not using the shared class as an RMS resource, make the class unavailable as an RMS resource again with one of the following methods. After performing the procedures, reboot the system.

- If the class is registered in resources used by the cluster application, remove it through the userApplication Configuration Wizard.
- If the class is not registered in resources used by the cluster application, execute the following command.

```
# /usr/opt/reliant/bin/hvgdsetup -d Class Name
~
Do you want to continue with these processes ? y
~
Do you need to start volumes in the specified disk class ? n
```

## **F.1.7 Shadow Object Errors**

---

### (1) A shadow volume became inaccessible.

#### **Explanation**

The configuration information of a shadow volume is not saved on the private slice, but managed in the memory. For this reason, the shadow volume configuration is cleared when the node on which the shadow volume is defined is rebooted, and thus the shadow volume becomes inaccessible.

#### **Resolution**

Delete the device special file according to the procedure described in "Rebooting a Node" in " A.2.20 Shadow Volume," and re-create a shadow volume.

## F.1.8 Disk Unit Error

---



### Note Identifying a Failed Disk Unit

GDS Management View shows disks with I/O errors by indicating them with red disk icons. In addition, if an I/O error occurs, 1 (one) is given as a value in the E field displayed with the `sdxinfo -e` long command.

However, for hardware-related errors, identify the failed parts based on log messages of disk drivers stored in the `/var/log/messages` file and such.

If GDS Management View and the `sdxinfo` command indicate the invalid status of slice data (invalid or INVALID), the slice data is invalid. It does not necessarily indicate that that slice has a hardware error.

For example, if an I/O error occurs on the copy source slice due to a hardware error during the synchronization copying of a mirror volume, data on the copy destination slice becomes invalid, but the copy source slice is still indicated as normal (active, stop, ACTIVE, or STOP). For details, see "F.1.1 Slice Status Abnormality."



See

For the methods of swapping disks, see "1.2.3 Hot Swap," "5.3.4 Disk Swap," and "D.8 sdxswap - Swap disk."

## F.2 Collecting Investigation Material

When a problem occurs with GDS that can not be resolved by referring to this Troubleshooting appendix, collect the investigation material and contact your local customer support.



Note

- Investigation material must be collected promptly. With the passage of time, necessary information may be lost.
- When using GDS with a cluster system, collect investigation material from all nodes.

### F.2.1 Collecting with pclsnap Command

The pclsnap command, a Linux system information collection tool, is provided in the FJSVpclsnap package. When a problem occurs with GDS, this command allows you to collect investigation material to identify the cause.

How to execute the pclsnap command is as follows.

1. Log in with superuser authority.
2. Execute the pclsnap command as follows.

```
# /opt/FJSVpclsnap/bin/pclsnap -h output
```

or

```
# /opt/FJSVpclsnap/bin/pclsnap -a output
```

*output* indicates either the special file name (example: /dev/st0) of the output media which stores the investigation material collected with the pclsnap command, or the output file name.

### F.2.2 Collecting Initial Investigation Material with sdxsnap.sh Script

Since the amount of information collected with the pclsnap command is rather large, it is often required to be sent to us by postal mail, and not by e-mail, which could take a few days before our engineer can start investigating.

In order to promptly start the initial investigation, you can only collect the essential information with the sdxsnap.sh script. If you wish to reduce the time lapse before the actual investigation starts, collect information using both the sdxsnap.sh script and the pclsnap command, and e-mail this initial investigation material first.

Execute the sdxsnap.sh script by following the procedures below.

1. Login with the superuser authority.
2. Execute the sdxsnap.sh script

```
# /etc/opt/FJSVsdxbin/sdxsnap.sh -s [-d dir]  
# cd dir  
# tar cvf snap.tar SDXSNAP
```

Use the -d option to save the material in the *dir* directory. If the -d option is not used, the material will be saved in the /tmp directory.

Send us the snap.tar file, which will be used as initial investigation material.

# Appendix G Frequently Asked Questions (FAQ)

In this section, frequently asked questions (FAQ) have been compiled.

## G.1 Operation Design

Questions regarding system operation design are explained.

### **Q: Is operation possible without using GUI (GDS Operation Management View)?**

**A**

Yes. Two types of operation management interfaces are available: the GUI (Graphical User Interface) and the CLI (Command Line Interface).

### **Q: Is it possible to operate GDS Snapshot through the GUI (GDS Management View)?**

**A**

Proxy object operations are possible. To operate shadow objects, use the CLI (Command Line Interface).

### **Q: Can disks with different performance specifications, such as size and revolution speed, be mirrored?**

**A**

Yes. However, when mirroring disks of different sizes, you will only be able to use the capacity of the smaller disk. For example, if a 4 GB disk and a 9 GB disk are mirrored, only 4 GB out of the 9 GB disk will be accessible. Also, when mirroring disks with different specifications such as revolution speed, the read performance becomes unbalanced and the write performance will depend on the slower disk's performance. Therefore, it is recommended to mirror disks with the same specifications. See "A.3.1 Guidelines for Mirroring" also.

### **Q: Do disks have to have equivalent attributes, such as size and revolution speed when I relate them as master and proxy?**

**A**

You can relate disks as master and proxy even if they have different attributes, such as size and revolution speed. However, it will result in that the read performance becomes unequal and the write performance depends on the slower disk's performance. Therefore, it is recommended to relate disks with the same attributes wherever possible. For details see also "A.1.7 Exception to Proxy Configuration."

### **Q: Can I combine mirroring and striping?**

**A**

You can mirror stripe groups. However, you cannot stripe across multiple mirror groups, using them as stripe columns.

### **Q: Do I need to specify certain settings in order to perform a hot swap?**

**A**

No. As long as the disk you are using supports hot swap, no special settings are required. For procedures, see "5.3.4 Disk Swap" and "D.8 sdxswap – Swap disk."

**Q: What is the purpose of single volume?**

**A**

GDS is a software which improves availability and manageability of the disk data. Single volume is used when users need to improve manageability, rather than availability. For details, see "1.3.2 Centralized Disk Management."

For example, if availability is already realized by using disk arrays, the user will benefit by using the physical disk as a single volume excluded from mirroring, to improve its manageability.

Using GDS to manage the volume as a single volume not only ensures consistent management, but also allows you to include the volume in the mirroring configuration without stopping the service application, when the need arises.

**Q: Can the messages output by GDS driver, daemon and command be displayed in a language other than English?**

**A**

No. The messages are in English only.

**Q: Are there any restrictions on the LUN (Logical Unit) configuration within the disk array when managing the disk array using GDS?**

**A**

There are no restrictions on LUN RAID level or size.

**Q: Are the mirrored disks differentiated as "primary disk" and "secondary" disks?**

**A**

No. Disks mirrored with GDS are handled in the same way. Both the user and the application do not differentiate between the two.

**Q: When reading the mirror volume, which one of the slices in the volume will the IO request be issued to?**

**A**

When a read is issued to a volume, the last access block number of each disk in the volume will be checked. The slice with the value closest to the read block number will be read. In other words, disk with the shortest seek distance will be read first. When a write is issued to a volume, it will write to all slices in the volume (except from slice in ACTIVE status) and will return the write results after completing the write request on all slices.

**Q: When mirroring a disk array that supports the hot spare function, which is preferable; using the spare disk of the disk array, or that of GDS?**

**A**

Generally, using the spare disk of the disk array is recommended.

**Q: Which of the GDS messages should be monitored?**

**A**

Generally, the following messages given in the "Appendix E GDS Messages" should be monitored.

- PANIC or WARNING level messages and Internal error messages shown in "E.2 Driver Messages"
- HALT, ERROR, or WARNING level messages and Internal error messages shown in "E.3 Daemon Messages"

For details, see "Appendix E GDS Messages."



**Q: When configuring a 24-hour operation system, would files of incremental size such as message log file have any effect?**

**A**

Since GDS's message log file is managed so that it will not exceed a certain size, there are no effects that need to be considered.

## G.2 Environment Configuration

Questions regarding environment configuration are explained.

**Q: How do I make a backup of a disk that is managed with GDS?**

**A**

The backup of GDS's logical volume can be made in the same way as a physical slice. However, make sure you make a backup using the logical volume. For example, do not boot from the CD-ROM and make a backup using the physical slice since correct data may not be saved.

You can also efficiently create a backup with the mirroring function. For details, see "1.3.8 Snapshot by Slice Detachment," "D.5 sdxslice - Slice operations," "5.3.2 Backup" and "6.1 Backing Up and Restoring Shared Disks."

By making use of GDS Snapshot proxy volumes, you can also configure backup administration that minimizes operation stop time. For details, see "1.5 Proxy Volume," "D.12 sdxproxy - Proxy object operations," "D.17 Snapshot Creation Using Command," "5.3.2.2 Backup (by Synchronization)," "5.3.2.3 Backup (by OPC)" and "6.2 Online Backup and Instant Restore through Proxy Volume."

Furthermore, backing up data through a server outside the domain running the service using GDS snapshot shadow volumes can hold the effects on the service to a minimum. For details, see "1.6 Shadow Volume" and "6.3 Backing Up and Restoring through an External Server."

**Q: I cannot execute the command fdisk(8) on a physical disk managed by GDS (for example sda). How can I solve this problem?**

**A**

Since access to physical disks is restrained by GDS, you cannot perform this operation. For details, see "A.2.2 Restraining Access to Physical Special File."

If all you need to do with the commands fdisk(8) is to refer to the disk contents, you may do so by booting from the CD-ROM, and specifying the physical disk. However, you must exercise care since updating the data on the physical disk will damage the mirrored data. For details, see "A.2.3 Booting from a CD-ROM Device."

## G.3 Operation

Questions regarding system operation are explained.

**Q: Do I need to exit all applications to perform synchronization copying on the volume? If it is not necessary, what effects should I be aware of?**

**A**

You do not need to exit the applications, but there could be some affect on the performance. If the application response is seriously affected, you can either suspend the synchronization copying, or adjust the load caused by the copying.

**Interrupting the copying:**

```
# sdxcopy -l -c class name -v volume name
```

**Resuming the copying:**

```
# sdxcopy -B -c class name -v volume name
```

**Adjusting the copying load:**

```
# sdxcopy -P -c class name -v volume name
```

**Checking the copying process:**

```
# sdxinfo -S -c class name -o volume name -e long
```

For details, see the command reference.

**Q: How do I change the volume size?****A**

Volumes can be expanded with the `sdxvolume -S` command during the service. For details, see "1.3.7 Online Volume Expansion."

The procedure for changing the volume size while suspending access to the volume is as follows.

- 1) Exit the application that is accessing the volume
- 2) Backup the volume data as necessary.
- 3) Remove the volume.
- 4) Check if you can create a volume with the necessary size in the group in which the volume existed. You can check this as follows.

```
# sdxinfo -V -c class name -o group name
```

The volume which indicates FREE in the STATUS field will be available. The BLOCKS field indicates the volume size. When there is more than one available volume, the maximum size you can create would be the size of the largest available volume.

5) When you cannot secure sufficient size within the group by performing procedure (4), backup a different volume and remove it, or create another group.

6) Specify a new size and create the volume. When creating the volume, specifying the same name as the volume you removed in procedure (3) will reduce the amount of information you have to input when changing the device definition information in the following procedure.

7) Update the device definition information such as `/etc/fstab`, as necessary.

8) Restore the backup data you collected in procedure (2).

**Q: When parting a proxy volume in a shared class, I want to lock the proxy volume not to start up on certain nodes.****A**

Before parting a proxy volume, on nodes where you want to restrain the proxy volume from startup, turn on the proxy volume lock mode with the `sdxattr -V` command. For details on the lock mode, see " Lock Volume " and " D.7 sdxattr - Set objects attributes."

**Q: Is there a possibility that joined proxy volumes are parted automatically?.****A**

In the situation where the master volume has only one ACTIVE slice, if an I/O error occurs in that slice, the proxy volume in STOP or INVALID status that is joined with the master volume through soft copy is parted automatically.

# Appendix H Register shared disk units

## H.1 Preparation to Use GDS

In a cluster environment, GDS needs to recognize how disk units are connected to each node. GDS determines the physical configuration and allows users to set the logical scope then define disk classes, disk groups and volumes.

Therefore, in order to create disk classes, disk groups and volumes, you must first understand how disk units are connected to the cluster system.

This appendix describes how to create physical disk connection configuration that is necessary to define disk classes, disk groups and volumes.

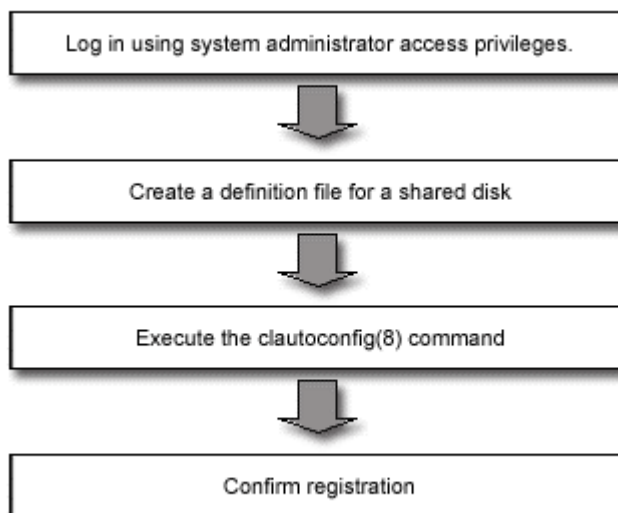
## H.2 Registration of shared disk units

It is a feature to register shared disk units connected to nodes in the resource database.

## H.3 Registration procedure

Take the following steps to register shared disk units in the resource database.

Chart: Flow of Shared Disk Unit Registration



**Note**

- This procedure should be done on an arbitrary node that consists of a cluster system.
- Do not execute the clautoconfig(8) command on the node where the command is running, or on any other nodes. If it occurs, registration is not properly done. You need to proceed with the following on all nodes consisting of a cluster system, and then take the above steps:
  - 1- Reset the resource database  
Execute the clinitreset(8) command to reset. See the manual for details.
  - 2- Restart a node

- First, you need to check the following:
  - 1- Setup of CF, CIP, and resource database are done on all nodes. See “PRIMECLUSTER Cluster Foundation (CF) (Linux) Configuration and Administration Guide” for details.
  - 2- All nodes are physically connected to the shared disk units.
  - 3- All nodes start in a multi-user mode.
  - 4- Setup of multi-path is done when using the shared disk units.
- Do not execute the setup command of GDS during the clautoconfig(8) command’s running. If it occurs, the setup command might detect an error.

## Register shared disk units

Take the following steps on an arbitrary node that consists of a cluster system.

### 1. Log in using system administrator access privileges

Log in to any node using system administrator access privileges.

### 2. Create a definition file for a shared disk

It is a file to define setup information of shared disk units connected to all nodes consisting of a cluster system.

The definition file is created in the following format:

```
“Resource key name” “Device name” “Node identification name”
“Resource key name” “Device name” “Node identification name”
:
```

- “Resource key name, Device name, and Node identification name” is defined in one string by shared disk unit connected to each node.
- “Resource key name, Device name, and Node identification name” are space-delimited.

Each name requires the following setup:

#### Resource key name

Specify a resource key name that has a sharing relation with each shared disk unit. The name should be the same by device shared between nodes. The name is also specified in the “shd number” format. This “shd” is a fixed string. Specify arbitrary 4 digits for number. If multiple devices are shared between multiple nodes, specify a different number.

Example

When sharing “/dev/sdb” and “/dev/sdc” between nodes

/dev/sdb resource key name: shd0001

/dev/sdc resource key name: shd0002

#### Device name

Specify a device name of a shared disk unit. The device name should be in the “/dev/device” format.

Example

When the device is “/dev/sdb”, the device name should be “/dev/sdb”.

#### Node identification name

Specify an identification name of the node connected to a device. Confirm the name by executing the clgettree(1) command. See the manual for details.

Example

node1 and node2 are identification names.

```
# /etc/opt/FJSVcluster/bin/clgettree
Cluster 1 cluster
Domain 2 PRIME
Shared 7 SHD_PRIME
Node 3 node1 ON
Node 5 node2 ON
```

When “/dev/sdb” and “/dev/sdc” are shared between node1 and node2, the definition file of a shared disk is as follows:

```
shd0001 /dev/sdb node1
shd0001 /dev/sdb node2
shd0002 /dev/sdc node1
shd0002 /dev/sdc node2
```

### 3. Execute the clautoconfig(8) command

Execute the clautoconfig(8) command to register the definition file that is created at (2) in the resource database. Specify the command in the following format:

#### Format

```
/etc/opt/FJSVcluster/bin/clautoconfig -f Full path of the definition file
```

#### Example

```
# /etc/opt/FJSVcluster/bin/clautoconfig -f /var/tmp/diskfile
```



#### Note

- When the clautoconfig(8) command is abnormally terminated, follow the displayed corrective action. See “PRIMECLUSTER Cluster Foundation (CF) (Linux) Configuration and Administration Guide” for the clautoconfig(8) command messages.
- The clautoconfig(8) command is not aware that the shared disk unit is physically connected. If you found a mistake of setting the definition file after executing the clautoconfig(8) command, you need to proceed with the following on all nodes consisting of a cluster system, and then take the above steps:
  - 1- Reset the resource database  
Execute the clinitreset(8) command to reset. See the manual for details.
  - 2- Restart a node

### 4. Confirm registration

Confirm information of the shared disk unit registered in the resource database using the clgettree(1) command.

#### Example

The following example shows that the shared disk unit “/dev/sdb” and “/dev/sdc” are shared between node1 and node2.

The resource of the “DISK” class is a device of the shared disk unit. The resource of the “SHD\_DISK” class indicates a sharing relation.

```
# /etc/opt/FJSVcluster/bin/clgettree
Cluster 1 cluster
  Domain 2 PRIME
    Shared 7 SHD_PRIME
      SHD_DISK 14 SHD_Disk14 UNKNOWN
        DISK 15 sdb UNKNOWN node1
        DISK 16 sdb UNKNOWN node2
      SHD_DISK 17 SHD_Disk17 UNKNOWN
        DISK 18 sdc UNKNOWN node1
        DISK 19 sdc UNKNOWN node2
    Node 3 node1 ON
      DISK 15 sdb UNKNOWN
      DISK 18 sdc UNKNOWN
    Node 5 node2 ON
      DISK 16 sdb UNKNOWN
      DISK 19 sdc UNKNOWN
```

## H.4 Command Reference

- `clautoconfig(8)`
- `cldelrsc(8)`

### H.4.1 `clautoconfig(8)`-Execute automatic resource registration

#### SYNOPSIS

```
/etc/opt/FJSVcluster/bin/clautoconfig -f diskfile  
/etc/opt/FJSVcluster/bin/clautoconfig -r -n
```

#### DESCRIPTION

This command enters information of the following hardware connected to a cluster system in the resource database.

- Shared disk devices  
Information of shared disk devices specified in the shared disk file is entered in the resource database.
- Network devices  
Network devices are automatically detected and entered in the resource database.

You can execute this command on any single node in the cluster system with the Administrator's authority.

While this command is running, you may not execute this command on the given node or another node.

#### OPTIONS

**-f *diskfile***

Registers the shared disk unit in the resource database. Specify the definition file for *diskfile* by full path.

**-n**

Registers the network device in the resource database. The detected network device is active. The inactive devices are not detected. Execute the `ifconfig(8)` command to confirm if the network device is running.

**-r**

Registers the resource automatically.

#### EXIT STATUS

**0:**

Normal termination.

**Other than 0:**

Abnormal termination.

#### NOTES

You need to check the following before executing the command:

- Setup of CF, CIP and resource database are done on all nodes
- All nodes are connected to the shared disk units.
- All nodes start in a multi-user mode.

## H.4.2 **cldelrsc(8)-Delete resource created by automatic resource registration**

### SYNOPSIS

`/etc/opt/FJSVcluster/bin/cldelrsc -r rid`

### DESCRIPTION

The `/etc/opt/FJSVcluster/bin/cldelrsc -r rid` command deletes the resource of the resource database created by automatic resource registration. With the automatic resource registration feature, the following information are registered:

Resource class name	Meaning
DISK	Resource of the shared disk unit
SHD_DISK	Shared resource of the shared disk unit
Ethernet	Resource of the network device

Check the resource to be deleted executing the `clgettree(1)` command.

If only the shared resource remains after deleting the resource of the shared disk unit, you need to delete the shared resource as well. You can execute this command on any single node in the cluster system with the Administrator's authority.

### OPTIONS

**-r rid**

Specify *rid* for the resource ID of the resource to be deleted. Check the resource ID executing the `clgettree(1)` command.

### EXIT STATUS

**0:**

Normal termination.

**Other than 0:**

Abnormal termination.

### NOTES

Delete the resource of the shared disk unit after removing GDS setups when the resource is used for GDS.

### EXAMPLES

When deleting the `/dev/sdc` device resource of the disk unit shared between node1 and node2

1. Check the resource ID of the `/dev/sdc` resource executing the `clgettree(1)` command. In the following example, the resource ID is 18 for node1, and 19 for node2.

```
# /etc/opt/FJSVcluster/bin/clgettree
Cluster 1 cluster
  Domain 2 PRIME
    Shared 7 SHD_PRIME
      SHD_DISK 14 SHD_Disk14 UNKNOWN
        DISK 15 sdb UNKNOWN node1
        DISK 16 sdb UNKNOWN node2
      SHD_DISK 17 SHD_Disk17 UNKNOWN
        DISK 18 sdc UNKNOWN node1
        DISK 19 sdc UNKNOWN node2
    Node 3 node1 ON
      DISK 15 sdb UNKNOWN
      DISK 18 sdc UNKNOWN
    Node 5 node2 ON
      DISK 16 sdb UNKNOWN
      DISK 19 sdc UNKNOWN
```

2. Delete the `/dev/sdc` device resource of the shared disk unit executing the `cldelrsc(8)` command.

```
# /etc/opt/FJSVcluster/bin/cldelrsc -r 18
# /etc/opt/FJSVcluster/bin/cldelrsc -r 19
```





# Glossary

## [A]

### **active disk**

The operating disk between disks that are connected to a switch group.

### **active slice**

A switch slice.

### **Advanced Copy function**

The copy function of Fujitsu Storage Systems ETERNUS. A generic term for EC (Equivalent Copy) and OPC (One Point Copy). For details, see the Fujitsu Storage Systems ETERNUS manual.

### **array disk**

See "disk array."

### **attach slice**

To add the detached slice to mirror volume making it accessible again as a component of the mirror volume.

### **automatic resource registration**

The function that registers hardware units with the resource database in the PRIMECLUSTER system. This is performed with the clautoconfig command provided by PRIMECLUSTER CF. For details see the PRIMECLUSTER Cluster Foundation Configuration and Administration Guide and "Appendix H Register shared disk units"

### **available size of a disk**

The capacity on an SDX disk that is available for creating volumes. It equals the physical disk size rounded to a cylinder boundary, minus the private slice size.

### **available size of a group**

The capacity that is available for a volume in a group.

## [B]

### **background copy**

The actual copying that takes place between master and proxy, after virtual completion of creating a snapshot or restoring the master when performing instant snapshot or instant restoring. When performing normal copy, the volume to which the data is being copied cannot be activated. However, when performing background copy, the volume to which the data is being copied can be activated for users to access the valid data.

### **block**

A unit for disk capacity or data transmission. Usually, one block is 512 bytes. Sector is used as a synonym.

**break**

To cancel the master-proxy relationship, and make independent again.

**[C]****class**

1. A disk class.
2. The generic term for disk classes and shadow classes.

**close down class**

To make all objects within a class unavailable for reference.

**close down volume**

To make the volume unavailable (invalid).

**cluster application**

A resource categorized as a userApplication used to group resources into a logical collection. For details, see "PRIMECLUSTER Installation and Administration Guide."

**cluster control**

A component contained in PRIMECLUSTER responsible for detecting various abnormalities and state transition. For details, see "PRIMECLUSTER Installation and Administration Guide."

**cluster system**

A highly available and reliable system with multiple nodes connected via a special network. For details, see "PRIMECLUSTER Installation and Administration Guide."

**concatenate**

To connect more than one physical disk. It is helpful when using multiple disks as one virtual mass disk.

**concatenation**

To concatenate.

**concatenation group**

A group whose type attribute is "concat." It is also the set of disks that will be concatenated.

**configuration database**

Database storing the configuration data of the objects managed by GDS. The configuration database is stored on a private slice.

**connect disk**

To add an SDX disk to a disk group. User can also connect a single disk or an undefined disk. Spare disk will be connected automatically by GDS when required.

**connect group**

To add a group to another group.

The former group is called a "lower level group," and the latter, a "higher level group."

For example, when more than one stripe group is connected to a mirror group that is the highest level group, the mirror volume that is created within the highest level group will be a logical device unit that will be both mirrored and striped.

**detach slice**

To temporarily detach a mirror slice from a mirror volume, making it inaccessible as a component of the mirror volume.

Other than by user command, GDS will automatically detach slices to maintain synchronization when an I/O error of a mirrored slice is detected.

**disconnect disk**

To remove a disk from a disk group.

**disconnect group**

To remove the lower level group from a higher level group.

**disk**

1. An SDX disk.
2. The generic term for SDX disks and shadow disks.

**disk array**

A disk unit which combines multiple physical disks to be handled as one virtual disk. Spreading the access across multiple physical disks greatly improves performance and reliability. There are several access methods which are referred to as RAID (Redundant Array of Inexpensive Disks) levels. Main RAID level includes RAID 0 (striping), RAID 1 (mirroring), and RAID 5 (striping with parity).

**disk class**

A collection of SDX objects. Disk classes may be called "classes." Physical disks can be registered with a class and accordingly are managed by GDS.

**disk group**

A collection of SDX disks and lower level groups. Disk groups may be called "groups." SDX disks and lower groups that belong to the same group become the unit of mirroring, striping, concatenating, or switching according to the type attribute of the group (mirror, stripe, concatenation, switch).

**disk switch**

To switch the roles of the active and inactive disks.

**domain**

One cluster system built by use of PRIMECLUSTER.

**[E]****EC**

Copying process using the EC (Equivalent Copy) function.

**EC function**

An abbreviation for Equivalent Copy function. It is one of the Advanced Copy functions provided by Fujitsu Storage Systems ETERNUS. It keeps duplicating disk data within a disk unit. By performing detaching control with software, a replicate of the disk will be created.

For details, see the Fujitsu Storage Systems ETERNUS manual.

**EC session**

EC request from the host (server) to Fujitsu Storage Systems ETERNUS.

For details, see the Fujitsu Storage Systems ETERNUS manual.

**ETERNUS3000 series**

Disk array unit manufactured by Fujitsu.  
For details see ETERNUS3000 series manuals.

**ETERNUS6000 series**

Disk array unit manufactured by Fujitsu.  
For details see ETERNUS6000 series manuals.

**[F]****FDU/FU**

A disk file unit with multiple physical disks mounted on a single enclosure.

**[G]****GDS Management View**

Software which allows various operations including GDS configuration setting and operation management using the Web screen.

**GFS**

Abbreviation of Global File Services. It is software that manages a GFS shared file system.  
For details see "PRIMECLUSTER Global File Services Configuration and Administration Guide."

**GFS Shared File System**

Shared file system that features simultaneous accessibility that maintains consistency between multiple Solaris systems to which a shared disk unit is connected and enables the continuation of processing on another node even if a certain node fails.  
For details, see "PRIMECLUSTER Global File Services Configuration and Administration Guide."

**group**

1. A disk group.
2. The generic term for disk groups and shadow groups.

**GR700 series**

Disk array unit manufactured by Fujitsu.  
For details see GR700 series manuals.

**[H]****hard copy**

To copy disk data created within the device using the disk array's hardware function so that no load is imposed on the server or the SAN.

**higher level group**

A group to which another group belongs.

**higher level shadow group**

A shadow group to which another shadow group belongs.

**hot spare**

A feature realizing automatic mirroring recovery while maintaining data accessibility at times of mirror disk failure.

**hot swap**

A feature allowing users to exchange the faulty physical disk components without stopping the application in use. Also referred to as active exchange.

**[I]****inactive disk**

A standby disk between disks that are connected to a switch group.

**inactive slice**

A standby slice that will be the operating slice when the active disk of the switch group is changed through a disk switch. The standby slice is a gathering of physically continuous blocks in the standby disk that belongs to the switch group."

**instant restore**

To instantly restore the master's data, by overwriting the master using the parted proxy data. Instant restore can be created by using the -e instant option with the sdxproxy RejoinRestore command or the -e instant option with the sdxproxy Restore command.

**instant snapshot**

To create a snapshot of a master instantly by cooperation with the OPC (One Point Copy) function provided by Fujitsu Storage Systems ETERNUS. Also it is a created snapshot. Snapshots can be created by using the -e instant option with the sdxproxy Part command.

**instant snapshot by OPC**

See "instant snapshot."

**[J]****JBOD**

An abbreviation for "Just a Bunch Of Disks." Multiple disk units are used as a single disk. See also "disk array."

**join**

To relate a pair of volumes or a pair of groups as a master and a proxy and synchronize the pair.

**joined**

The state when master and proxy are joined. Proxy volume in joined status cannot be accessed.

**JRM**

An abbreviation for Just Resynchronization Mechanism.

**just resynchronization copying**

A high-speed resynchronization copying process. Available when just resynchronization mode is valid.

**Just Resynchronization Mechanism**

A mechanism realizing just resynchronization copying. Abbreviated as JRM.

**logical slice**

The component of a logical volume. A logical slice consists of contiguous blocks of one physical disk or multiple blocks of multiple physical disks. A mirror volume consists of one or more logical slices that are mirrored. A single volume, a stripe volume, a volume in a concatenation group, and a switch volume consist of a single logical slice. Logical slices may be called "slices."

**logical volume**

A logical disk device that provides the user with a block device interface and a character (raw) device interface as well as a physical slice (disk partition). The user can access data on a physical disk through accessing the logical volume. Logical volumes are classified into single volumes, mirror volumes, stripe volumes, the volumes in concatenation groups ,and switch volumes. Logical volumes may be called "volumes."

**lower level group**

A group that belongs to another group. A volume can not be created within the lower level group.

**lower level shadow group**

A shadow group that belongs to another shadow group. A shadow volume cannot be created in the lower level shadow group.

**[M]****master**

See "master object."

**master group**

The original group from which the proxy group is copied.  
The term "master" can refer to both master group and master volume.

**master object**

A generic term for master volume and master group.  
Also simply referred to as "master."

**master volume**

The original volume from which the proxy volume is copied.  
Volumes within master group are also called master volumes.  
The term "master" can refer to both master volume and master group.

**mirror group**

A group whose type attribute is "mirror." It is the set of disks and lower level groups, which will be mirrored each other.

**mirror volume**

A volume that is created within a mirror group. Its data will be made redundant by mirroring.

**mirror slice**

A logical slice that composes a mirror volume. A mirror volume is created by mirroring more than one logical slice. For each mirror volume, there will be one logical slice in each disk and each lower level group belonging to the mirror group.

**mirroring**

To maintain redundancy by allocating the same data to multiple slices. A feature which allows you to continue access to a volume if one or more valid slices exist in the volume, even if some failures occur on the other slices.

**MPHD disk**

A virtual device that provides multiple access paths with the mphd driver. The user will be able to continue operation, even if an access path failure occurs.

**MPLB disk**

A virtual device that provides multiple access paths with the mplb driver. Being able to use multiple paths enables spreading out of the load when accessing the disk, and increases the availability at the time of path failure.

**[N]****node**

One computer (with one operating system). For details, see "PRIMECLUSTER Installation and Administration Guide."

**node identifier**

A general name for each node within a PRIMECLUSTER system.

**node name**

The name returned with the uname -n command

**NVURM**

An abbreviation for Non-Volatile Update Region Map. This is information of the updated volume region stored on a private slice for just resynchronization.

**[O]****object**

1. An SDX object. Objects are the generic term for classes, groups, disks, volumes, and slices.
2. The generic term for SDX objects and shadow objects.

**offline backup**

To create backups from a volume after stopping applications that use the volume.

**online backup**

To create volume backup while an application is accessing the volume.

**online disk migration**

To change from one disk to another without stopping applications accessing the disk data.

**online snapshot**

To create a volume snapshot while an application is accessing the volume.

**OPC**

Copying process using the OPC (One Point Copy) function.

**OPC function**

An abbreviation for One Point Copy function. It is one of the Advanced Copy function provided by Fujitsu Storage Systems ETERNUS. Disk data within the disk unit at a certain point will be instantly copied to separate area in the disk unit. The logical copying process is completed immediately after the request to start copying is sent from the server. The actual copying process will be performed later in the background.

For details, see the Fujitsu Storage Systems ETERNUS manual.

**OPC session**

OPC request from the host (server) to the Fujitsu Storage Systems ETERNUS.

For details, see the Fujitsu Storage Systems ETERNUS manual.

**[P]****part**

To temporarily separate proxy from master so that it may be accessed independently. The parted proxy becomes a snapshot with master data of the moment it was parted. Although parted, the master-proxy relationship is maintained.

**parted**

The status of a proxy isolated from the master. When a proxy volume is parted, access to the proxy that is independent of access to the master is available.

**physical disk**

A storage media possible of handling random input and output of data. Mainly used to refer to a set of physical slices.

**physical scope**

A group of nodes to which physical disks are connected.

**physical slice**

A part of physical disk and a series of physically continued blocks. The starting sector number and the number of slices occupied by the physical slice is registered with the partition table.

**physical special file**

A device special file of a physical disk, such as `/dev/sd[a-z]* [1-4]*`.

**PRIMECLUSTER**

Software to configure the Cluster System. For details, see "PRIMECLUSTER Installation and Administration Guide."

**PRIMECLUSTER system**

A cluster system configured using PRIMECLUSTER. For details, see "PRIMECLUSTER Installation and Administration Guide."

**private slice**

A special slice used by GDS for control. Part of the SDX disk. Cannot be used as a logical slice.

**proxy**

See "proxy object."



**proxy group**

Replication group of master group.

It has the same volume configuration as the master group, and each volume is the proxy volume of the master volume that is within the master group.

The term "proxy" can refer to both proxy group and proxy volume.

**proxy object**

A generic term for proxy volume and proxy group.

Also simply referred to as "proxy."

**proxy volume**

A replication volume of a master volume. This may be called an alternative volume. Proxy volumes and proxy groups may be called "proxies" collectively.

**[R]****REC**

Copying process by REC (Remote Equivalent Copy) function.

**REC function**

An abbreviation Remote Equivalent Copy function. It is one of the Advanced Copy function provided by Fujitsu Storage Systems ETERNUS. REC is EC function available between Fujitsu Storage Systems ETERNUS boxes. For details, see the Fujitsu Storage Systems ETERNUS manual.

**register disk**

To add a physical disk to a certain class. The registered disk is managed by GDS subsequently.

**rejoin**

To re-join the parted proxy with master.

**relate**

To relate a pair of volumes or a pair of groups as a master and a proxy and leave them parted.

**remove disk**

To remove an SDX disk from disk class.

**resource**

The concept describing the elements managed by PRIMECLUSTER among hardware and software configuring the PRIMECLUSTER system. A PRIMECLUSTER system manages GDS disk classes as resources.

For details, see "PRIMECLUSTER Installation and Administration Guide."

**restore**

To restore the master's data, by overwriting the master using the parted proxy data.

**resynchronization copying**

To copy data between mirror slices to recover mirroring state when rebooting after a system failure, after a cluster failover, or after re-attaching a detached slice to a volume.

Or, to copy data between master and proxy to match their data when rejoining a parted proxy to a master.

**ROPC**

Copying process by ROPC (Remote One Point Copy) function.

**ROPC function**

An abbreviation Remote One Point Copy function. It is one of the Advanced Copy functions provided by Fujitsu Storage Systems ETERNUS. ROPC is OPC function available between Fujitsu Storage Systems ETERNUS boxes. For details, see the Fujitsu Storage Systems ETERNUS manual.

**[S]****SAN**

An abbreviation for "Storage Area Network." It is a network connecting multiple servers and storage devices. Usually, Fibre Channel is deployed for connection.

**scope**

One of the class attributes. The scope indicates a group of nodes on which objects of the class are available.

**SDX disk**

A disk registered with a disk class. SDX disks may be called single disks, spare disks, and undefined disks according to their usage. SDX disks may be called "disks."

**SDX object**

A virtual resource managed by GDS. SDX objects are the generic term for classes, groups, disks, volumes, and slices. SDX objects may be called "objects."

**sector**

See "block."

**shadow class**

A collection of shadow objects. Disk classes and shadow classes may be called "classes" collectively. Physical disks managed as SDX disks in another domain and those as the hard copy destinations of SDX disk data copied with disk unit copy functions, can be registered with a shadow class and accordingly can be managed and accessed as shadow objects. Objects that belong to a shadow class can be used only on the local node.

**shadow disk**

A disk registered with a shadow class. SDX disks and shadow disks may be called "disks" collectively.

**shadow group**

A group created in a shadow class. The shadow group is a collection of shadow disks and lower shadow groups. Disk groups and shadow groups may be called "groups" collectively.

**shadow object**

A virtual resource managed by GDS. Shadow objects are the generic term for shadow classes, shadow groups, shadow disks, shadow volumes, and shadow slices. These shadow objects are available on a node where GDS Snapshot has been implemented. SDX objects and shadow objects may be called "objects" collectively.

**shadow slice**

The slice that composes a shadow volume. Logical slices and shadow slices may be called "slices" collectively.

**shadow volume**

A volume created in a shadow group. Logical volumes and shadow volumes may be called "volumes" collectively.

**shared class**

A "shared" type class. Objects that belong to a shared class are available on multiple nodes. With the shared class, disks connected to multiple nodes, including the local node, can be registered. Shared classes are also the units of resources used by cluster applications.

**shared object**

An object that belongs to a "shared" class.

**single disk**

An SDX disk which does not belong to any group and is capable of creating single volumes.

**single node**

A node that is not Cluster System configuration.

**single slice**

A logical slice that consists a single volume. Single volume consists of one single slice. It is also a series of physically continued block in a single disk.

**single volume**

A volume that is created within a single disk that does not belong to a group. Its data is not made redundant.

**size of a group**

The total of available size of a group and the size of one private slice.

**slice**

1. A logical slice.
2. The generic term for logical slices and shadow slices.

**snapshot**

To create a replica of a volume at a certain moment in another. It is also a created replica. GDS provides snapshot functions by slice detachment and by proxy volumes. Through use of proxy volumes, snapshot by synchronization and instant snapshot by OPC are available.

**snapshot by slice detachment**

To create a snapshot of a volume at a moment to a slice by detaching a slice from a mirror volume temporarily. Also it is a created snapshot.

**snapshot by synchronization**

To create a snapshot of a master at a moment to a proxy by parting the master and proxy which were previously joined and synchronized. Also it is a created snapshot.

**soft copy**

To copy disk data, using the GDS 's driver.

**spare disk**

A reserved disk to recover mirroring when a mirrored disk fails. When an I/O error is detected in a disk belonging to a mirror group or its lower level group, this spare disk will automatically be connected to recover normal mirroring.

**start volume**

To make volume accessible.

**stop volume**

To make volume inaccessible.

**Storage Area Network**

See "SAN."

**stripe column**

Disks and lower level groups by which striping is conducted. In other words, disks and lower level groups belonging to a stripe group.

**stripe group**

A disk group whose type attribute is "stripe." This is a group of disks and lower level groups by which striping is conducted.

**stripe volume**

A volume that is created within a stripe group. Stripe volume can distribute the I/O load among multiple disks with its striping function. Its data will not be redundant.

**stripe width**

The size of divided data when striping.

**striping**

To divide data into a certain size and then allocate the divided data alternately to multiple slices. Striping distributes the I/O load across multiple physical disks.

**swap**

To swap the slices comprising master and proxy that are joined and synchronized.

**switch**

Disk switch.

**switch group**

A group of the switch type. This is a gathering of an active disk and an inactive disk that are switched.

**switch slice**

An operating logical slice comprising a switch volume. A switch volume consists of one switch slice. The switch slice is a gathering of physically continuous blocks in the active disk that belongs to the switch group.

**switch volume**

A volume created in a switch group. With the disk switch function, the roles of the active and inactive disks can be switched. Data will not be redundant.

**synchronization**

The state when mirroring is operating normally.  
Or, a state when the master and proxy data are the same.

**synchronization copying**

To copy data from a volume to an attached slice in order to make a mirrored state or to increase the multiplicity of the mirroring configuration, or to copy data between a master and a proxy in order to make their contents equivalent.

**synchronized**

The state when synchronization is maintained.

**system disk**

The disk that contains the Linux operating system running on the system. It refers to the entire disk containing slice operating as one of the following file systems (or swap area).

/, /usr, /var (or swap area)

**the highest level group**

A group that does not belong to any other group. Volumes can be created within the highest level group.

**the highest level shadow group**

A shadow group that does not belong to another shadow group. A shadow volume can be created in the highest level shadow group.

**[U]****update**

To copy (overwrite) data from a master to a parted proxy. The updated proxy is a snapshot that retains an image of the master at a certain moment.

**undefined disk**

An SDX disk that is neither a single disk or a spare disk. Its usage

**[M]****volume**

1. A logical volume.
2. The generic term for logical volumes and shadow volumes.

**Web-Based Admin View**

Software for system operation and management using the Web screen.  
For details, see "Web-Based Admin View Operation Guide."

