
Using SYMCLI to Perform TimeFinder/Clone Operations

Abstract

This white paper provides an introduction to TimeFinder/Clone software. Cloning allows you to make instant point-in-time copies of volume data on multiple target devices from a single source device.

Copyright © 2005 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS”. EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, Symmetrix, Celerra, CLARiiON, CLARAlert, Documentum, HighRoad, Legato, Navisphere, PowerPath, ResourcePak, SnapView/IP, SRDF, TimeFinder, VisualSAN, and where information lives are registered trademarks and EMC Automated Networked Storage, EMC ControlCenter, EMC Developers Program, EMC OnCourse, EMC Proven, EMC Snap, Access Logix, AutoAdvice, Automated Resource Manager, AutoSwap, AVALONidm, C-Clip, Celerra Replicator, Centera, CentraStar, CLARevent, Connectrix, CopyCross, CopyPoint, DatabaseXtender, Direct Matrix, Direct Matrix Architecture, EDM, E-Lab, Enginuity, FarPoint, FLARE, GeoSpan, InfoMover, MirrorView, NetWin, OnAlert, OpenScale, Powerlink, PowerVolume, RepliCare, SafeLine, SAN Architect, SAN Copy, SAN Manager, SDMS, SnapSure, SnapView, StorageScope, SupportMate, SymmAPI, SymmEnabler, Symmetrix DMX, Universal Data Tone, and VisualSRM are trademarks of EMC Corporation. All other trademarks used herein are the property of their respective owners.

Part Number 300-001-382 REV D

Table of Contents

Introduction	5
Purpose and Scope	5
Related Documentation	5
Practical Uses	5
Copying to a Clone	6
Deferred Copying to a Clone	7
Immediate Full Copying to a Clone.....	8
Copying Only Changed Data to a Clone.....	9
Restoring Data from a Clone	9
Setting a Pre-Copy Mode of Clone Operation	10
Cloning a Copy from a BCV Source Device	11
Cloning Multiple Copies from a Single Source	13
Cloning a Copy on a Remote Symmetrix.....	15
Cloning Copies of the Same Data Locally and Remotely.....	16
Cloning Multiple Copies Locally and Remotely	17
Using the –consistent Option When Activating a Copy Session.....	19
Beginning and Ending a Copy Session	19
Using a Composite Group to Control a Set of Clone Pairs That Spans Multiple Symmetrix Arrays.....	20
Example 1: Cloning a Copy of a Standard Device to Other Standard Devices	22
Example 2: Cloning a Copy of a Standard Device to BCV Devices	27
Example 3: Cloning Copies from a BCV Source Device.....	33
Example 4: Cloning Multiple Copies of a BCV	41

Example 5: Cloning 16 Copies from a Single Source Device.....47

Example 6: Using a Composite Group to Control Clone Pairs51

Introduction

TimeFinder®/Clone software (referred to as “TimeFinder” in this document) allows you to clone copies of data simultaneously on multiple target devices from a single source device. The data is available to a target’s host instantly. You can copy data from a single source device to as many as sixteen target devices. A source device can be either a Symmetrix® standard device or a BCV device. A target device is a Symmetrix *clone* that contains the full volume of data.

The BCV is a Symmetrix device with special attributes created when the Symmetrix is configured. It can function either as an additional mirror to a Symmetrix logical volume or as an independent, host-addressable volume. Establishing BCV devices as mirror images of active production volumes allows you to run multiple simultaneous business continuance tasks in parallel. The principal device, known as the *standard device*, remains online for regular Symmetrix operation from the original production server. Each BCV contains a unique host address, making it accessible to a separate backup/recovery server. When you use `symmir` commands to establish a BCV as a mirror of a standard device, that relationship is known as a *BCV pair*. The BCV is temporarily inaccessible to its host until you split the BCV pair.

The *clone* is a host-addressable Symmetrix standard device or BCV that is the target of a clone copy. Unlike a BCV mirror copy, which must be completely synchronized with its source and then split to access the data, the clone copy activation makes data on the clone immediately accessible to its host, even while copying is occurring in the background. You use `symclone` commands to create and activate a source/target *clone pair*.

Descriptions in this paper of operations on device groups are also relevant for composite groups.

Purpose and Scope

This document describes TimeFinder clone features included in versions of EMC® Solutions Enabler up through version 6.0 running on Symmetrix arrays using Enginuity™ versions 5568, 5569, 5670, and 5x71.

Related Documentation

The following white papers provide information related to the concepts discussed in this paper:

- *Using SYMCLI to Perform TimeFinder/Mirror Control Operations* (P/N 300-000-074)
- *Using SYMCLI to Perform TimeFinder/Snap Operations* (P/N 300-001-381)

Practical Uses

Using TimeFinder clones can satisfy your need for multiple uses of copies of the production data. You can designate one copy for testing, one copy for data warehousing, one copy for backups, one copy for report generation, and so forth. You can also use clones for improving the time for data access by assigning users to copies of the data instead of the one-and-only production copy, thereby reducing disk contention. You can also use a clone copy to access a single mailbox in an Exchange database.

You can use a clone to perform file recovery from an auxiliary host:

1. Use a good BCV copy of the standard device as the source device, and clone it to a target device.
2. Mount the target to the auxiliary host. A good copy of the relevant file via the BCV is now available on the clone target and can be accessed by the auxiliary host.
3. Copy the good file from the clone target to the production host.
4. Terminate the copy session.

Beginning with Solutions Enabler version 5.4, you can use a composite group to control a set of clone pairs that spans multiple Symmetrix arrays, cloning source data that spans more than one Symmetrix array.

Beginning with Solutions Enabler version 6.0, you can control specific clone pairs within a composite group instead of having to operate on the entire group. This allows composite groups the same granularity as device groups, although with the added scope of being able to span multiple Symmetrix arrays.

Beginning with Solutions Enabler version 6.0, you can perform differential cloning (copying only the changed data) and clone operations on remote Symmetrix arrays. You can also set various modes for clone copying: copy, no-copy, or pre-copy.

Copying to a Clone

Copying to a clone allows you to copy data from a single source device to as many as sixteen target devices, with automatic background copying of up to four target devices simultaneously. The target of a copy operation is a Symmetrix clone. Copying to a clone can be either immediate or deferred:

- A deferred copy of those tracks on a full device that contain the data of interest. Copying occurs when there are writes to the source device or reads/writes to the target device. The clone pair state remains CopyOnAccess until you terminate the copy session or all tracks have been accessed.
- An immediate copy of the *full* source device to the target device requires the `-copy` option.¹ The clone pair state is CopyInProgress until the clone pair reaches the Copied state. Beginning with Solutions Enabler version 6.0 and Enginuity version 5x71, you can also add the `-differential` option, which means that any subsequent clone operation will copy only the changed data.

When you copy to clones, the source or the target device can be either a standard device or BCV device but must be the same emulation type. The target device must be the same size as the source device. Figure 1 illustrates cloning a copy of a source device on a Symmetrix array to a target device on the same array.

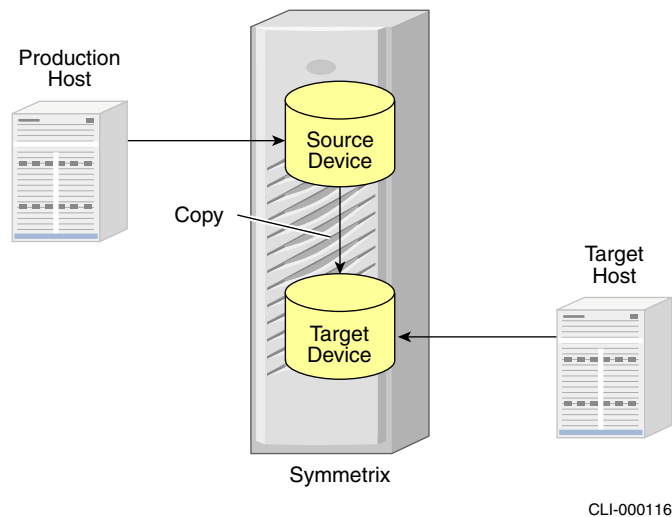


Figure 1. Copying a Source Device to a Target Clone

The production host performs I/O to a source device. At any point in time, you can generate a copy to a target device that can be accessed by the target's host.

¹ Beginning with Solutions Enabler version 6.0 and Enginuity version 5x71, you can use the `-precopy` option instead of the `-copy` option. For more information, refer to the section "Setting a Copy Mode."

When you issue the `symclone create` command, the target device is made not ready (NR) to its host. This operation then sets up the track protection bitmap to provide user access to cloned data as soon as the clone pair is activated. When you issue the `symclone activate` command, the target is made ready again. A user on the target's host can then access data on the target immediately:

- If the target's host attempts to read from or write to target device tracks that have not been copied yet, the system finds those tracks on the source device and copies them immediately to the target, then services the I/O request.
- If the source's host attempts to write to source device tracks that are not copied yet, the system finds those tracks on the source device and copies them immediately to the target, before those writes to the source can occur on the source.

A *source* standard device can have BCV pairing relationships and, at the same time, independent copy sessions with other devices. A source standard device can be established with a BCV as part of a BCV pair and clone a copy to a second device without splitting the BCV pair. However, the standard device can only have up to sixteen *total* relationships at any one time (BCV pairs and clone pairs).² You cannot use an established BCV as the source of a clone operation because an *established* BCV cannot act as an independent device. Only a *split* BCV can. The BCV must be split to be a source or a target.

A *target* standard device, if also part of a synchronizing BCV pair, can copy data to a BCV and *receive* data from its clone source at the same time. However, the clone copy drives the BCV copy. While the BCV pair is synchronizing, TimeFinder copies data to the BCV until it encounters indirect tracks on the standard device clone and then stops copying to the BCV until the clone pair either reaches the Copied state or is terminated (termination cancels all indirect tracks). At this point, the BCV pair can continue synchronizing.

Deferred Copying to a Clone

Deferred copying minimizes the background read-and-write activity in the Symmetrix array. With the copy-on-first-access type of copy to a clone, copying is deferred until there are:

- Writes to the source device, or
- Reads or writes to the target device

When you establish a clone pair relationship between a source device and target device as in Figure 1, the *target host* can access what is on the source device until you terminate the pair's copy session. In this way, a target host can list the source directories and read a file, thus initiating an on-demand clone copy of the file from the source device to the target device.

For example, after establishing the clone pair in the CopyOnAccess state, you could perform the following steps:

1. Mount the file system on the target host (for example, `mount /filesys2`)
2. List files in the file system directory or one of its subdirectories (for example, `ls /filesys2`)
3. Display (read) the file that you want to copy to the target device (for example, `cat bigfile`)

Reading the file (or attempting to write to it) initiates the actual copying of the file. When the clone copy completes, the clone pair state is still CopyOnAccess. You can either access other files, initiating clone copies of data from device tracks occupied by those files, or you can end the clone pair relationship.

² If differential cloning is being used, or if the source device has any snap pair relationships, the total possible relationships is fifteen.

A two-command sequence allows you to clone data from a source to a target. For example, from a source standard device with the logical device name SRC to a target standard device whose logical device name is TGT.³

1. `symclone -g ProdBgrp create SRC sym ld TGT`

Creates the clone pair session (assigns a session ID), sets up a track protection bitmap for each of the paired devices, places the clone pair in the Created state, and places the target device in the Not Ready state to prevent access to it prior to activating the clone pair. By default, the mode is NOCOPY, which means the clone pair will be in the CopyOnAccess state after activation (for information about using COPY mode, refer to the section “Immediate Full Copying to a Clone.”)⁴

2. `symclone -g ProdBgrp activate SRC sym ld TGT`

Places the clone pair in the CopyOnAccess state and places the target device in the Read/Write (RW) state (use the `-not_ready` option if you want to activate the point-in-time clone but keep the target device Not Ready). The actual copying of data is deferred until you either modify tracks on SRC or read/modify tracks on TGT.

Beginning with Solutions Enabler version 6.0, you can combine both commands into one command to save a step. The `symclone establish` command performs the same functions as `create` and `activate`. For example, to perform a *deferred* clone operation:

```
symclone -g ProdBgrp establish -full SRC sym ld TGT
```

Immediate Full Copying to a Clone

If you need to clone a *full* copy of the entire source device SRC to target device TGT, you can issue the `symclone create` command with the `-copy` option. Copying begins immediately upon issuing the `symclone activate` command.

```
symclone -g ProdBgrp create -copy SRC sym ld TGT
symclone -g ProdBgrp activate SRC sym ld TGT
```

When you issue the `symclone create` command, the target device becomes not ready (NR) to its host. This operation then sets up the track protection bitmap to provide user access to cloned data as soon as the clone pair is activated.

When you issue the `symclone activate` command, the target becomes ready again. You can then access data on the target immediately, even while copying is in progress.

Beginning with Solutions Enabler version 6.0, the `symclone establish` command performs the same functions as `create` and `activate`. For example, to perform an *immediate full* clone operation:

```
symclone -g ProdBgrp establish -full SRC sym ld TGT
```

Differential setup is implied with a full clone establish operation. Any subsequent clone operation will automatically be an incremental copy.

³ If you know that a target is a BCV device, you can use `bcv ld` on the command line as an alternative to `sym ld` when issuing a `symclone create`, `activate`, `establish`, `verify`, or `terminate` command. Optionally, you can also use `bcv dev` or `bcv pd` followed by the BCV's Symmetrix device name or physical device name, respectively.

⁴ An environment variable, `SYMCLI_CLONE_COPY_MODE`, supports the `symclone create` operation. You can set this variable to `NOCOPY`, `COPY`, `PRECOPY`, `COPY_DIFF`, or `PRECOPY_DIFF`. The clone session will then be created in the specified mode.

Copying Only Changed Data to a Clone

Beginning with Solutions Enabler version 6.0 and Enginuity version 5x71, subsequent cloning to the same target can be performed as differential copying. You can use the `-differential` option the first time you clone a full copy from the source to the target. All subsequent clones during that clone session automatically become incremental copies, capturing any new writes to the source device.

A differential or incremental clone operation copies only those device tracks that have changed since the initial full clone was performed. To do this, however, the clone session that existed for the full clone must still exist. You must also set up the differential clone during a full clone operation. The `-differential` option, therefore, must be used here with the `-copy` (or `-precopy`) option. For example:

```
symclone -g ProdBgrp create -copy -differential SRC sym ld TGT
```

The `-differential` option creates an SDDF session for the source and target.

The `symclone activate` command initiates a full copy of the source device to the target device.

```
symclone -g ProdBgrp activate SRC sym ld TGT
```

To capture subsequent writes to the source during this clone pair session, use `symclone recreate` to clone just those tracks that have changed since the full copy completed. For example:

```
symclone -g ProdBgrp recreate SRC sym ld TGT
symclone -g ProdBgrp activate SRC sym ld TGT
```

To save some steps, you can use the `symclone establish` command in the same way. For example, after the first full clone operation, you can capture any new writes with a subsequent clone operation that is automatically an incremental copy:

```
symclone -g ProdBgrp establish -full SRC sym ld TGT
:
:
symclone -g ProdBgrp establish SRC sym ld TGT
```

Restoring Data from a Clone

Prior to version 6.0, you restored data from a clone target to its source device or another device by performing a reverse clone operation. If restoring to the source device, you terminated all clone sessions involving the source and then used the existing target as the new source device in a clone session that copied data back to the original source. If restoring to a device other than the source device, you had to terminate the clone session involving the target from which the restore would occur.

Beginning with Solutions Enabler version 6.0 and Enginuity version 5x71, you can perform a restore operation from a clone target to its source device (or to a separate device) without terminating the original clone session. However, you must have created the original clone session using the `-differential` option described above. The following command restores data incrementally from the target device (TGT) back to the original source device (SRC). Only data that changed on the target is copied back to the source.

```
symclone -g ProdBgrp restore SRC sym ld TGT
```

The following command restores data fully from the original target device (TGT) to a device (RESTGT) that was not involved in the original clone session. All data on TGT is copied to RESTGT.

```
symclone -g ProdBgrp restore -full RESTGT sym ld TGT
```

Setting a Pre-Copy Mode of Clone Operation

Beginning with Solutions Enabler version 6.0 and Enginuity version 5x71, you can set a pre-copy mode of operation that initiates data copying in the background prior to actually activating the clone session. This allows the early movement of data before the point-in-time clone copy is established.⁵

Pre-copying can occur in the created or recreated state. While in this state, the pre-copy process never completes. Instead, the process checks for new writes and pre-copies them to the target device until the clone session is activated. Upon activation, the normal background copy mechanism begins, and the pre-copy operation ends.

You can set the pre-copy mode in one of two ways. One way is to include the `-precopy` option with the `symclone create` command. To set up incremental copying on subsequent copy operations, you need to *add* the `-differential` option. For example:

```
symclone -g ProdBgrp create -precopy SRC sym ld TGT -differential
```

Another way is to use the `symclone set mode` command while the clone pair is in the created or recreated state (not yet activated). For example:

```
symclone -g ProdBgrp set mode precopy SRC sym ld TGT
```

If the initial `symclone create` command specifies the `-copy` option, or `nocopy` by default, you can use `set mode` to change the mode to `precopy` prior to activation. However, the reverse is not true. Once `precopy` mode is set, you cannot change to `copy` or `nocopy`.

Recall that `copy` refers to a full copy, and `nocopy` refers to a deferred copy of those tracks on a full device that contain the data of interest. Deferred copying occurs only when there are writes to the source device or reads/writes to the target device.

Prior to activation, you can switch from `copy` mode to `nocopy` mode, and vice versa. For example, to switch from `copy` to `nocopy`:

```
symclone -g ProdBgrp set mode nocopy SRC sym ld TGT
```

After activation, you can use the `symclone verify` command to ascertain when one pre-copy cycle has completed. For example to check every five seconds for the cycle completion:

```
symclone -g ProdBgrp verify -precopy -cycled -i 5
```

The `-precopy` option alone verifies just the pre-copy state. By adding the `-cycled` option, you verify when one pre-copy cycle is complete.

⁵ This pre-copy technique is used in TimeFinder copying to a RAID-5 BCV. When SYMAPI encounters a RAID-5 BCV, it automatically employs clone emulation to initiate concurrent pre-copying to all members of the RAID-5 BCV. For information about using TimeFinder with clone emulation, refer to the white paper *Using SYMCLI to Perform TimeFinder/Mirror Control Operations* (P/N 300-000-074).

Cloning a Copy from a BCV Source Device

Figure 2 illustrates cloning a copy of a BCV on a Symmetrix array to a target device on the same array.

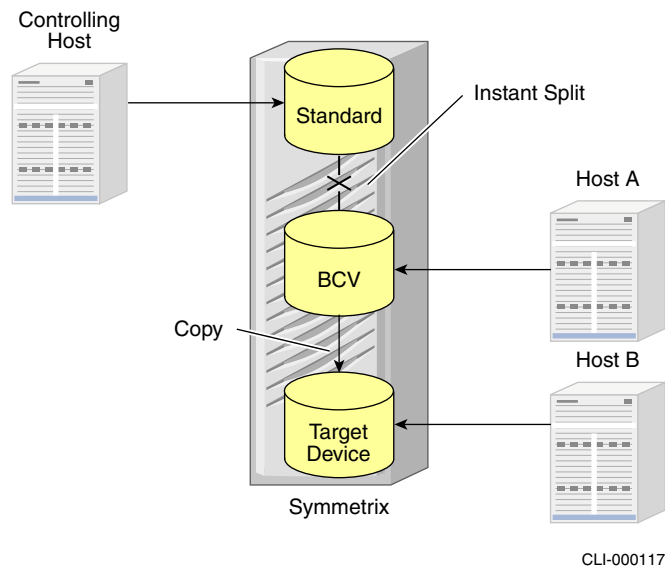


Figure 2. Cloning a Copy of a BCV Device

The controlling host performs I/O to a standard device that is established with a BCV as part of a BCV pair. At any point in time while the BCV is synchronized with the standard device, you can decide to split the BCV pair. When the BCV pair has finished splitting in the background, you can then clone the BCV copy to a target device that can be accessed by Host B. After you terminate a copy session (ending the clone pair relationship), you can re-establish the BCV pair. The following steps outline an example of setting up devices to clone a copy from a BCV source device. All commands are issued from the controlling host:

1. Create a Regular type device group.

```
symdmg create ProdBgrp -type regular
```
2. Add to the device group a standard device to hold the production data and a target standard device to hold the clone copy. Note that the target can also be a BCV, in which case you would add it in step 3.

```
symld -g ProdBgrp -sid 3264 add dev 286  
symld -g ProdBgrp -sid 3264 add dev 287
```
3. Associate with the device group a BCV device to be the source of the clone operation.

```
symbcv -g ProdBgrp -sid 3264 associate dev 39A
```
4. Establish Symmetrix device 286 (DEV001) with the only BCV in the device group.

```
symmir -g ProdBgrp establish -full DEV001 -noprompt
```
5. Verify that the BCV pair is in the Synchronized state (for example, checking every 30 seconds).

```
symmir -g ProdBgrp verify DEV001 -i 30
```

6. Perform an instant split (or a consistent split⁶) on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to starting the clone operation.

```
symmir -g ProdBgrp split DEV001 -instant -not_ready -noprompt
```

7. Verify that the background split is complete. The following command checks every five seconds.

```
symmir -g ProdBgrp verify DEV001 -split -bg -i 5
```

8. Create a copy session so that data can be cloned from the BCV source to the target standard device. Always use the logical device name for the source and any of the three device name formats for the target (for example, the target could also be `sym ld DEV002`).

```
symclone -g ProdBgrp create BCV001 sym dev 287 -noprompt
```

9. Activate the clone operation.

```
symclone -g ProdBgrp activate BCV001 sym dev 287 -noprompt
```

10. Optionally, make the BCV device ready again to its host.

```
symld -g ProdBgrp ready -bcv BCV001 -noprompt
```

11. To query the state of the clone operation or verify when the clone pair reaches the CopyOnAccess state, you can issue the following commands.

```
symclone -g ProdBgrp query
symclone -g ProdBgrp verify BCV001 -copyonaccess
```

With `symclone verify`, you can include one of three options: `-copied`, `-copyinprog`, or `-copyonaccess`. The default is to verify the Copied state, which occurs only if a source device is fully copied to a target device.

12. When you have cloned what you need from the source device, you can either maintain that clone pair relationship (by doing nothing) or you can terminate the clone pair relationship as follows.

```
symclone -g ProdBgrp terminate BCV001 sym dev 287 -noprompt
```

13. Once a clone pair copy session is terminated, you can incrementally re-establish the BCV pair (for example, DEV001 with BCV001).

```
symmir -g ProdBgrp establish DEV001 -noprompt
```

⁶ For information on using the ECA `-consistent` option to perform consistent splits, refer to the white paper *Using SYMCLI to Perform Consistent Splits with TimeFinder Family Products* (P/N 300-000-283).

Cloning Multiple Copies from a Single Source

While a source device can be concurrently copying the same data to four target devices at the same time, the source is allowed to have up to sixteen target devices (constituting sixteen clone pairs). If a source device has sixteen targets and you attempt to clone a seventeenth target device, SYMCLI displays an error message indicating that your source already has the maximum number of targets. In this case, to clone the new target device, you need to first issue a `symclone terminate` command on at least one of the current targets associated with this source device.

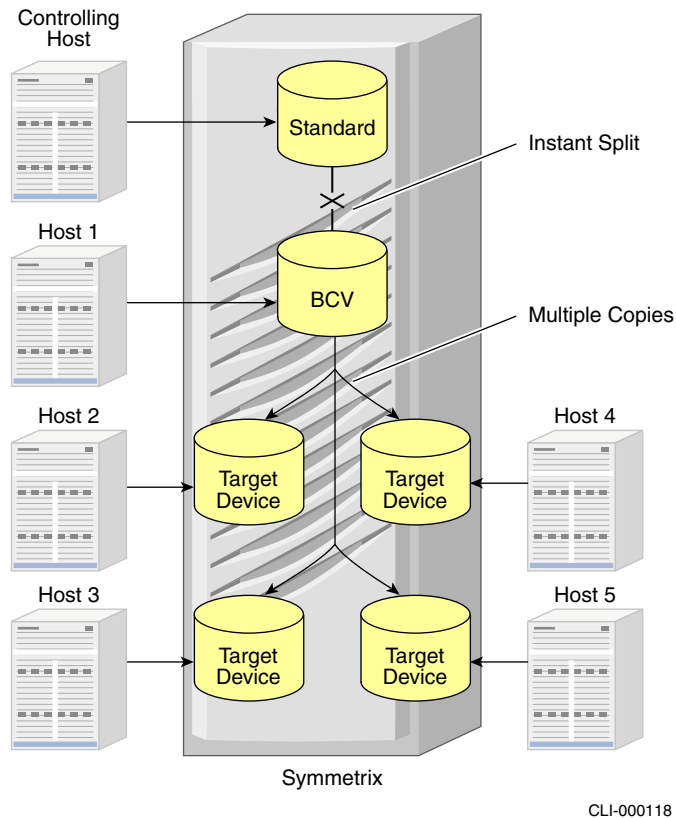


Figure 3. Cloning Multiple Copies of a BCV Device

The configuration in Figure 3 copies from a BCV source to *four* target devices on the same Symmetrix array, illustrating how a clone operation can copy a source device to four target devices simultaneously.

The controlling host performs a clone operation on the Symmetrix array that results in all target devices having the same data. In this configuration, five hosts have access to copies of the same data (one copy on the BCV, and four copies on target devices).

The following steps outline an example of setting up devices to clone multiple copies from a BCV source device. All commands are issued from the controlling host:

1. Create a Regular type device group.

```
symdmg create ProdGrp -type regular
```
2. Add to the device group a standard device to hold the production data and four target standard devices to hold the clone copies. (If the targets were BCVs, you would add them in step 3.)

```
symld -g ProdGrp -sid 3264 add dev 286  
symld -g ProdGrp -sid 3264 addall dev -range 287:290
```

3. Associate with the device group a BCV device to be the source of the clone operation.
`symbcv -g ProdBgrp -sid 3264 associate dev 39A`
4. Establish Symmetrix device 286 (DEV001) with the only BCV in the device group.
`symmir -g ProdBgrp establish -full DEV001 -noprompt`
5. Verify that the BCV pair is in the Copied state (for example, checking every 300 seconds).
`symmir -g ProdBgrp verify DEV001 -i 300`
6. Perform an instant split (or a consistent split) on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation.
`symmir -g ProdBgrp split DEV001 -instant -not_ready -noprompt`
7. Verify that the background split is complete. The following command checks every five seconds.
`symmir -g ProdBgrp verify DEV001 -split -bg -i 5`
8. Create copy sessions that will permit you to clone data from the BCV source device (BCV001) to the four target devices (DEV002 – DEV005).
`symclone -g ProdBgrp create -noprompt BCV001 sym ld DEV002`
`symclone -g ProdBgrp create -noprompt BCV001 sym ld DEV003`
`symclone -g ProdBgrp create -noprompt BCV001 sym ld DEV004`
`symclone -g ProdBgrp create -noprompt BCV001 sym ld DEV005`
9. You can activate the clone operation with one command that activates all sessions simultaneously.
`symclone -g ProdBgrp activate -noprompt BCV001 sym ld DEV002 \`
`BCV001 sym ld DEV003 \`
`BCV001 sym ld DEV004 \`
`BCV001 sym ld DEV005`
10. Optionally, make the BCV device ready again to its host.
`symld -g ProdBgrp ready -bcv BCV001 -noprompt`
11. To display the state of *all* devices involved in the clone operation, you can perform a clone query with the `-multi` option.
`symclone -g ProdBgrp query -multi`
12. When you have cloned the data you need from each source device, you can stop the copy sessions between the source and its multiple targets.
`symclone -g ProdBgrp terminate -noprompt BCV001 sym ld DEV002`
`symclone -g ProdBgrp terminate -noprompt BCV001 sym ld DEV003`
`symclone -g ProdBgrp terminate -noprompt BCV001 sym ld DEV004`
`symclone -g ProdBgrp terminate -noprompt BCV001 sym ld DEV005`
13. You can now incrementally re-establish the BCV pair (DEV001 with BCV001).
`symmir -g ProdBgrp establish DEV001 -noprompt`

Cloning a Copy on a Remote Symmetrix

Beginning with Solutions Enabler version 6.0 and Enginuity version 5x71, you can use SRDF® technology and the TimeFinder RDF flag (the `-rdf` option) to clone devices on a remote Symmetrix array as shown in Figure 4. Performing SYMCLI commands from the controlling host allows the remote target device to receive a copy of the data from the R2 device. The cloned data can be accessed by remote host A.

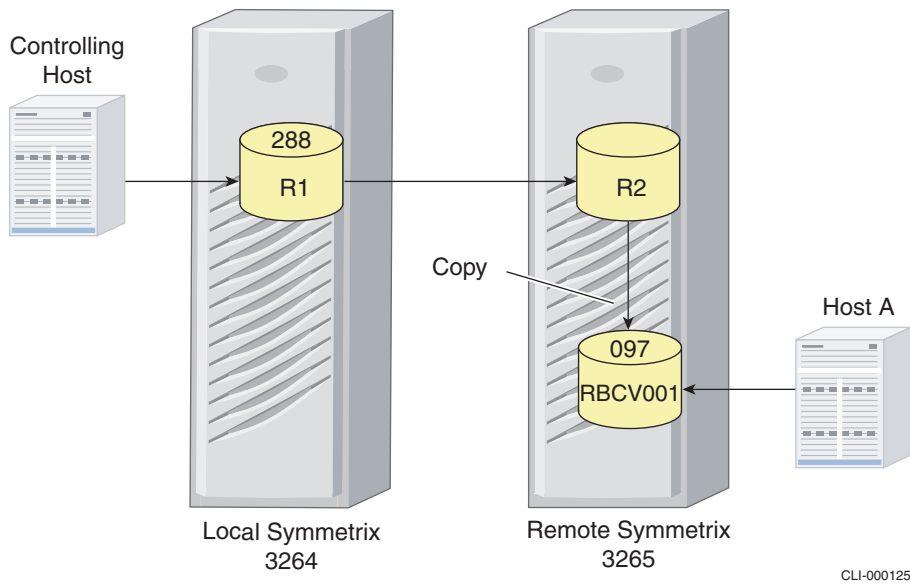


Figure 4. Cloning a Copy on a Remote Symmetrix Array

The following steps outline an example of cloning devices on a remote Symmetrix array:

1. Create an RDF1 type device group.

```
symdmg create Rdf1Grp -type rdf1
```
2. Add to the device group an R1 standard device (288) on the local Symmetrix (sid 3264) to be the source device. Add a target BCV device (097) on the remote Symmetrix to hold the clone copy. A *remote* target device must be a BCV.

```
symld -g Rdf1Grp -sid 3264 add dev 288  
symbcv -g Rdf1Grp associate dev 097 -rdf
```
3. Clone an immediate full copy from the source device (DEV001) to the remote BCV target device (RBCV001). DEV001 is the logical device name for device 288, and RBCV001 is the logical device name for BCV 097.

```
symclone -g Rdf1Grp establish -full DEV001 bcv ld RBCV001 -rdf
```
4. To query the progress of the clone operation or verify when the copy is completed, you can issue the following commands that examine the clone pair (source and target).

```
symclone -g Rdf1Grp query -rdf  
symclone -g Rdf1Grp verify -copied -rdf
```

Cloning Copies of the Same Data Locally and Remotely

Beginning with Solutions Enabler version 6.0, you can clone copies of the same data to devices on local and remote Symmetrix arrays at the same time. As Figure 5 illustrates, the controlling host can perform a local clone and a remote clone, resulting in target devices on both sides having the same data.

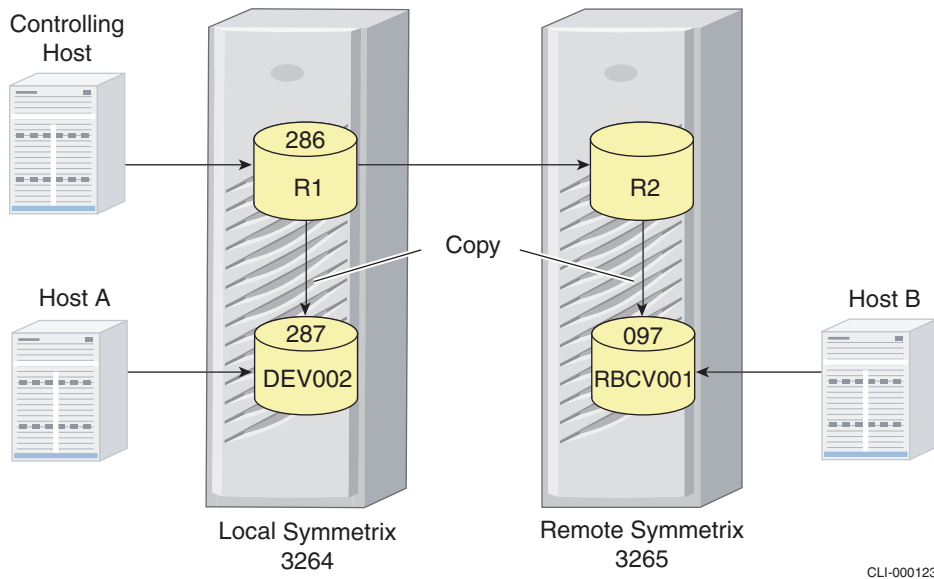


Figure 5. Cloning Copies on Local and Remote Symmetrix Arrays

The following steps clone the same data to targets on a local Symmetrix and a remote Symmetrix:

1. Create an RDF1 type device group.
2. Add to the device group an R1 standard device (286) on the local Symmetrix array (sid 3264) to be the source device. Add a standard device target (287) on the local array. Add a BCV target (097) on the remote array. A *remote* target device must be a BCV.

```
symdg create Rdf1Grp -type rdf1
symld -g Rdf1Grp -sid 3264 add dev 286
symld -g Rdf1Grp -sid 3264 add dev 287
symbcv -g Rdf1Grp associate dev 097 -rdf
```

3. Clone an immediate full copy from the source device (DEV001) to the local and remote target devices (DEV002 and RBCV001, respectively). If there is no I/O to the source R1 device between these two commands, the same data will exist on both the local and remote target devices.

```
symclone -g Rdf1Grp establish -full DEV001 sym ld DEV002
symclone -g Rdf1Grp establish -full DEV001 bcv ld RBCV001 -rdf
```

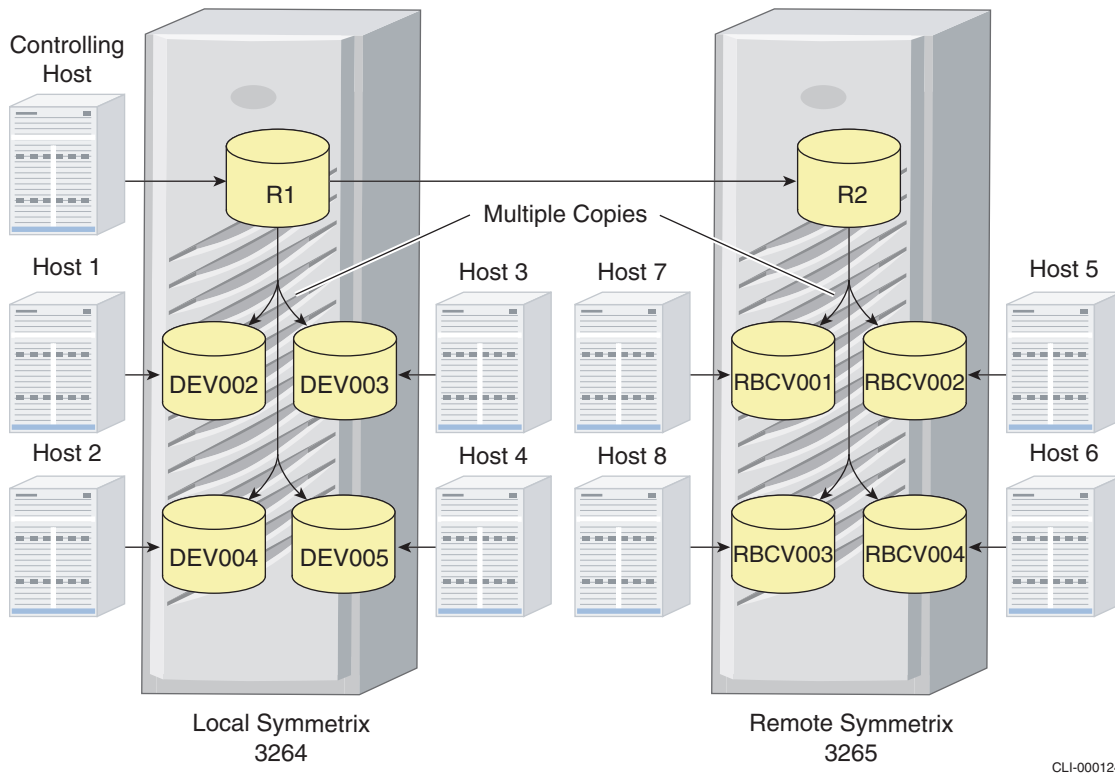
4. To query the progress of the local clone operation or verify when the local copy is completed, you can issue the following commands that examine the local cloned pair (source and target).

```
symclone -g Rdf1Grp query
symclone -g Rdf1Grp verify -copied
```

To query the progress of the remote clone operation or verify when the remote copy is completed, add the `-rdf` option to these commands.

Cloning Multiple Copies Locally and Remotely

The configuration in Figure 6 is basically the same as Figure 5 except that this configuration clones from a source to *four* target devices on each Symmetrix array instead of to one target on each Symmetrix. This illustrates how a single `symclone` command can copy from a source device to four target devices simultaneously. In this configuration, eight hosts have access to copies of the same target data.



CLI-000124

Figure 6. Cloning Multiple Copies on Local and Remote Symmetrix Arrays

While a source device can clone to only four target devices at the same time, the source is allowed to have up to sixteen target devices (constituting sixteen clone pairs). If a source device has sixteen targets and you attempt to clone a seventeenth target device, SYMCLI displays an error message indicating that your source already has the maximum number of targets. In this case, to clone the new target device, you need to first issue a `symclone terminate` command on at least one of the current clone targets associated with this source device.

The process for building an RDF1 type device group is similar to the previous section. For example, to add a source standard R1 device, a range of four standard device targets from the local Symmetrix array, and a range of four BCV targets from the remote Symmetrix array:

```
symld -g Rdf1Grp -sid 3264 add dev 100
symld -g Rdf1Grp -sid 3264 addall dev range 101:104
symbcv -g Rdf1Grp associateall dev range 214:218 -rdf
```

What is different about this configuration is the cloning of copies from a single source to multiple target devices. To clone the four *local* target devices (DEV002 – DEV005) from the local source device (DEV001), you need to issue four `symclone establish` commands, specifying the same source device with each of the four targets:

```
symclone -g Rdf1Grp establish -full DEV001 sym ld DEV002
symclone -g Rdf1Grp establish -full DEV001 sym ld DEV003
symclone -g Rdf1Grp establish -full DEV001 sym ld DEV004
symclone -g Rdf1Grp establish -full DEV001 sym ld DEV005
```

To display the progress of *all* devices involved in the *local* clone operation, you need to perform a clone query with the `-multi` option:

```
symclone -g Rdf1Grp query -multi
```

You can verify the clone completion of one or more clone pairs in the device group specifically, or all clone devices. For example:

```
symclone -g Rdf1Grp verify -copied DEV001 sym ld DEV002
symclone -g Rdf1Grp verify -copied
```

You need to terminate the four local clone sessions before cloning new targets (the four remote targets, for example):

```
symclone -g Rdf1Grp terminate DEV001 sym ld DEV002
symclone -g Rdf1Grp terminate DEV001 sym ld DEV003
symclone -g Rdf1Grp terminate DEV001 sym ld DEV004
symclone -g Rdf1Grp terminate DEV001 sym ld DEV005
```

To clone the four remote target devices (RBCV001 – RBCV004) from the source device (DEV001), you need to issue four `symclone establish` commands with the `-rdf` option:

```
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RBCV001
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RBCV002
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RBCV003
symclone -g Rdf1Grp establish -full -rdf DEV001 bcv ld RBCV004
```

To display the progress of all devices involved in the *remote* clone operation, you need to perform a clone query with the `-multi` option and the `-rdf` option:

```
symclone -g Rdf1Grp query -multi -rdf
```

You can verify the clone completion of one or more remote clone pairs in the device group specifically, or all remote clone devices, by adding the `-rdf` option. For example:

```
symclone -g Rdf1Grp verify -copied -rdf DEV001 sym ld RBCV004
symclone -g Rdf1Grp verify -copied -rdf
```

Using the `-consistent` Option When Activating a Copy Session

Using the `-consistent` option with the `symclone activate` or the `symclone establish` commands uses Enginuity Consistency Assist (ECA) to make copies that are restartable for dependent-write applications. The Symmetrix API (SYMAPI) suspends I/O before activation, which prevents any pending records from being included in the copy. After activation occurs, SYMAPI allows I/O to resume.

The target devices now contain a copy of the source device that is consistent with the production database up until the time of the activation. The production DBMS instance has no knowledge that the suspend/resume operation occurred.

Beginning and Ending a Copy Session

A copy session for a clone pair originates when you issue the `create` command and ends when you issue the `terminate` command.⁷ A device cannot participate in two copy sessions simultaneously unless it is a source device that has multiple target devices. To pair an existing target device with a different source device, you need to first terminate the target's original copy session.

To ensure that you do not copy over previously copied data, each clone operation results in putting a hold on the target device. Terminating a copy session removes any hold⁸ on the target device and deletes pair information about the terminated pair from the Symmetrix array. The target host can no longer reference data on the source device through *indirection* (indirect target tracks that provide visibility to source tracks).

The following command ends the clone pair relationship between SRC and TGT:

```
symclone -g ProdBgrp terminate SRC sym ld TGT
```

When terminating a pair relationship, no conflicts exist if the pair is in the Created state or the Copied state. However, stopping a copy session for a pair whose state is CopyOnAccess or CopyOnWrite may end the session prematurely if an application has not finished accessing data or if writes to the source device are ongoing. Terminating a pair is not allowed when the pair state is CopyInProgress (unless you do so by specifying the `-symforce` option).

Beginning with Solutions Enabler version 6.0, you can set the `SYMCLI_CLONE_PAIR_POLICY` environment variable to either allow or deny a new clone session when the maximum number of sessions is already running. In this case, setting the variable to `TERM_OLDEST` terminates the oldest session when the new clone session is created. Setting the variable to `DONT_TERM` blocks a new session in this case until you explicitly terminate one of the sixteen current sessions.⁹

⁷ If you fail to issue the `symclone activate` command prior to terminating the copy session, the target device (including the label) is still modified. All tracks on the target are set as "aborted."

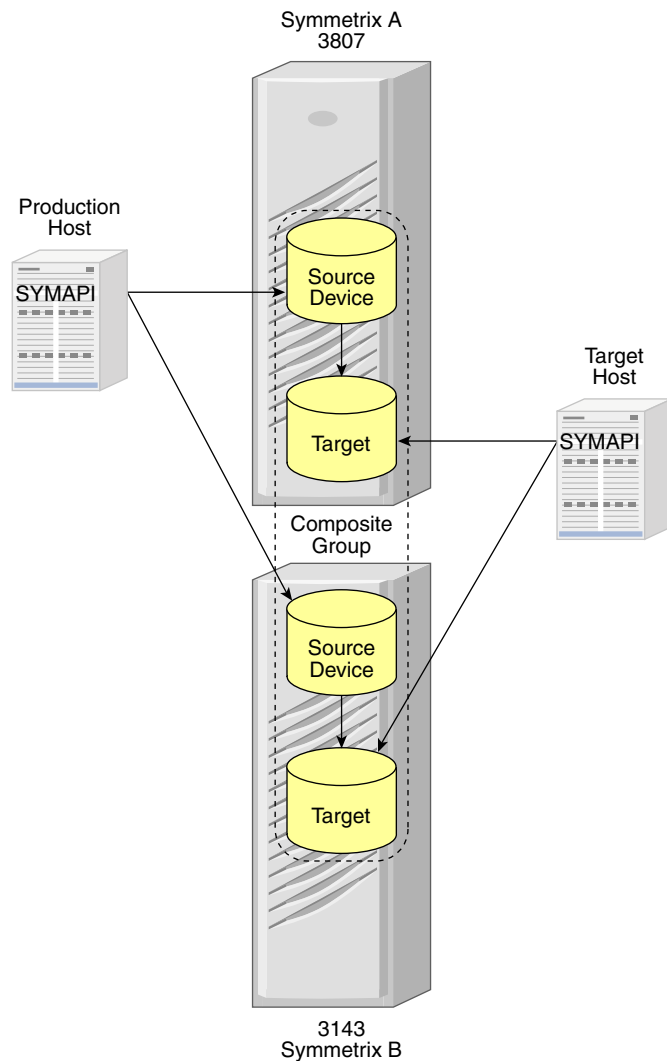
⁸ If for any reason the `symclone terminate` command does not release the hold, try releasing it using the format `symld -g ProdBgrp unhold DEV002`. If the target is a BCV, include the `-bcv` option on the command line.

⁹ You can copy data from a single source device to as many as 16 target devices unless you are using differential copying, in which case 8 clone sessions is the maximum.

Using a Composite Group to Control a Set of Clone Pairs That Spans Multiple Symmetrix Arrays

A *composite group* is a user-defined group of devices that can span multiple Symmetrix arrays. This feature provides greater flexibility than a device group, which can define devices only on a single Symmetrix. Beginning with Solutions Enabler version 6.0, you can control specific clone pairs within the composite group instead of having to operate on the entire group as in previous versions.

Figure 7 illustrates a production host that is locally connected to two Symmetrix arrays (A and B). A composite group is defined on the production host and includes source devices and target devices from each Symmetrix array. The target devices can be standard devices or BCV devices. Another locally connected host allows you to access the clone targets. A locally connected host allows you to access the clone targets.



CLI-000119

Figure 7. Using a Composite Group When a Set of Devices Spans Two Symmetrix Arrays

Although clone copy operations might normally be performed from the production host (shown in Figure 7) because the composite group is defined there in its SYMAPI database, there are methods that would allow you to control clone operations from another locally connected host like the target host. One way is to copy

the composite group definition to another host. A more efficient method is to enable Group Naming Services (GNS), which automatically propagates the composite group definition to the Symmetrix arrays and other locally attached hosts that are running the GNS daemon. For more information, refer to the white paper *Using SYMCLI and GNS to Propagate Group Definitions to Multiple Hosts* (P/N 300-301-384).

Prior to Solutions Enabler version 6.0, composite group support did not allow you to clone from a single source to multiple targets. You had to add an equal number of source devices and target devices to the composite group so that the group was populated with one-to-one source/target relationships and managed as a group. Solutions Enabler version 6.0 lifts this restriction by allowing you to create clone pairs individually within a composite group and thus create clone pairs using multiple targets and a single source.

The following steps outline the setup required for controlling a set of clone pairs that spans two Symmetrix arrays as shown in Figure 7:

1. From the production host, create a Regular type composite group (for example, MyGrp).

```
symcg create MyGrp -type regular
```
2. Add to the composite group those standard devices on Symmetrix A (3087) and Symmetrix B (3143) that are the source devices.

```
symcg -cg MyGrp -sid 3087 add dev 0072
symcg -cg MyGrp -sid 3143 add dev 0095
```
3. Associate a BCV target device from each Symmetrix array with the composite group.

```
symbcv -cg MyGrp -sid 3087 associate dev 0053
symbcv -cg MyGrp -sid 3143 associate dev 004B
```
4. Create clone pair sessions from those devices in the composite group.

```
symclone -cg MyGrp create
```
5. Activate these clone pair sessions.

```
symclone -cg MyGrp activate
```

Beginning with Solutions Enabler version 6.0, you can control specific clone pairs within the composite group. For example, to create and activate only the DEV001/DEV002 clone pair from all devices in the group:

```
symclone -cg MyGrp create DEV001 sym ld DEV002
symclone -cg MyGrp activate DEV001 sym ld DEV002
```

Or using the one-step method:

```
symclone -cg MyGrp establish DEV001 sym ld DEV002
```

Example 1: Cloning a Copy of a Standard Device to Other Standard Devices

The hardware setup consists of a single Symmetrix array (sid 041) connected to a Solaris controlling host. The source of the clone is Symmetrix device 0101. The targets are Symmetrix devices 0102 and 0103.

The `symdmg` command creates a Regular type device group named `clonegrp`.

```
# symdmg create clonegrp -type regular
```

The `symld` command adds a device range (0101 through 0103) to the device group. By default, their logical device names will be DEV001, DEV002, and DEV003, respectively.

```
# symld -g clonegrp addall dev -range 101:103
```

The `symclone create` command, by default, creates a CopyOnAccess copy session so that selected data can be cloned from source device DEV001 to target DEV002 when the clone operation is activated.

```
# symclone -g clonegrp create DEV001 sym ld DEV002 -noprompt
```

'Create' operation execution is in progress for device 'DEV001' paired with target device 'DEV002' in device group 'clonegrp'. Please wait...

'Create' operation successfully executed for device 'DEV001' in group 'clonegrp' paired with target device 'DEV002'.

The `symclone query` command indicates that the clone pair is in the Created state. The CG columns relate to the Legend: an X in the column means that a Legend item applies to the clone pair; a dot (.) in the column means that item does not apply. Only (G) in the Legend applies to this clone pair.

```
# symclone -g clonegrp query DEV001
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device		State		Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG SRC <=> TGT		(%)
DEV001	0101	138090	DEV002	0102	.X Created		0
Total		-----					
MB(s)		4315.3					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The `symclone activate` command activates the clone operation.

```
# symclone -g clonegrp activate DEV001 sym ld DEV002 -noprompt
```

'Activate' operation execution is in progress for device 'DEV001' paired with target device 'DEV002' in device group 'clonegrp'. Please wait...

'Activate' operation successfully executed for device 'DEV001' in group 'clonegrp' paired with target device 'DEV002'.

The `symclone query` command indicates that the clone pair is now in the CopyOnAccess state.

```
# symclone -g clonegrp query DEV001
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG SRC <=> TGT	(%)	
DEV001	0101	138090	DEV002	0102	.X CopyOnAccess	0	
Total		-----					
MB(s)		4315.3					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

To ensure that you do not copy over previously copied data, the clone operation results in putting a hold on the target device. The `symld list` command with the `-held` option displays any devices in the device group that are currently being held. This hold is removed when you terminate the clone pair session.

```
# symld -g clonegrp -held list
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Standard Device Name			Directors			Device	
Logical	Physical	Sym	SA :P	DA :IT	Config	Att	Cap
Sts	(MB)						
DEV002	rdmp/c4t0d161s2	0102	01C:0	16B:C2	2-Way Mir	RW	4315

The following `symclone create` command creates a copy session between the same source device (DEV001) and a different target device (DEV003). The `-copy` option will cause the *immediate* initiation of a *full* copy between the devices *when the copy session is activated*.

```
# symclone -g clonegrp create DEV001 sym ld DEV003 -copy -noprompt
```

'Create' operation execution is in progress for device 'DEV001' paired with target device 'DEV003' in device group 'clonegrp'. Please wait...

'Create' operation successfully executed for device 'DEV001' in group 'clonegrp' paired with target device 'DEV003'.

A `symclone query` command with the `-multi` option displays the multiple copy sessions for source device DEV001 and the current state of each clone pair (DEV001/DEV003 and DEV001/DEV002).

```
# symclone -g clonegrp query DEV001 -multi
```

```
Device Group (DG) Name: clonegrp
DG's Type                : REGULAR
DG's Symmetrix ID        : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
DEV001	0101	138090	DEV003	0103	XX	Created	0
		138090	DEV002	0102	.X	CopyOnAccess	0
Total		-----					
MB (s)		4315.3					

Legend:

(C): The background copy setting is active for this pair.
(G): The Target device is associated with this group.

The `symclone activate` command initiates the full copy from DEV001 to DEV003.

```
# symclone -g clonegrp activate DEV001 sym ld DEV003 -noprompt
```

'Activate' operation execution is in progress for device 'DEV001' paired with target device 'DEV003' in device group 'clonegrp'. Please wait...

'Activate' operation successfully executed for device 'DEV001' in group 'clonegrp' paired with target device 'DEV003'.

Another `symclone` query displays again the multiple copy sessions for source device DEV001 and an updated state for each clone pair. The state of the DEV001/DEV003 pair has changed from Created to CopyInProgress (which is 1 % complete). The state of the DEV001/DEV002 pair is still CopyOnAccess.

```
# symclone -g clonegrp query DEV001 -multi
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
DEV001	0101	136639	DEV003	0103	XX	CopyInProgress	1
		138090	DEV002	0102	.X	CopyOnAccess	0
Total		-----					
MB(s)		4270.0					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The following `symclone terminate` command ends the DEV001/DEV002 clone pair session normally. Normal termination is possible whenever a clone pair is in the Created, Copied, or CopyOnAccess state.

```
# symclone -g clonegrp terminate DEV001 sym ld DEV002 -noprompt
```

```
'Terminate' operation execution is in progress for device 'DEV001' paired with
target device 'DEV002' in device group 'clonegrp'. Please wait...
```

```
'Terminate' operation successfully executed for device 'DEV001' in group
'clonegrp' paired with target device 'DEV002'.
```

The following `symclone terminate` command attempts to end the DEV001/DEV003 clone pair session normally, but cannot do so because the DEV001/DEV003 pair state is CopyInProgress.

```
# symclone -g clonegrp terminate DEV001 sym ld DEV003 -noprompt
```

```
'Terminate' operation execution is in progress for device 'DEV001' paired with
target device 'DEV003' in device group 'clonegrp'. Please wait...
```

The session cannot be terminated because a copy is in progress unless the Symmetrix force flag is used

The following `symclone terminate` command uses the `-symforce` option to force an end to the DEV001/DEV003 clone pair session. If you need to use the `-symforce` option, you should unmount file systems and shut down applications and databases prior to issuing this command.

```
# symclone -g clonegrp terminate DEV001 sym ld DEV003 -noprompt -symforce
```

```
'Terminate' operation execution is in progress for device 'DEV001' paired with  
target device 'DEV003' in device group 'clonegrp'. Please wait...
```

```
'Terminate' operation successfully executed for device 'DEV001' in group  
'clonegrp' paired with target device 'DEV003'.
```

Example 2: Cloning a Copy of a Standard Device to BCV Devices

In this example, the three standard devices (0101, 0102, and 0103) currently in the device group from Example 1 are all source devices for this clone operation. Three BCV devices (0215, 0216, and 0217) are added as targets.

The following command associates three BCV devices (0215, 0216, and 0217) with the device group.

```
# symbcv -g clonegrp associateall dev -range 215:217
```

The `symmir query` command checks the device group for any current copy sessions. Because all copy sessions for devices in the device group were stopped in Example 1, the output below indicates no clone devices in the group.

```
# symclone -g clonegrp query
```

Device group 'clonegrp' does not have any devices that are Clone source devices

The following command, by default, creates CopyOnAccess copy sessions for the standards and BCVs, pairing them in the order that they were added to the device group (for example, the first standard device, 0101, with the first BCV device, 0215).

```
# symclone -g clonegrp create -v -noprompt
```

'Create' operation execution is in progress for device group 'clonegrp'.
Please wait...

SELECTING Source devices in the group:

```
Device: 0101 [SELECTED]
Device: 0102 [SELECTED]
Device: 0103 [SELECTED]
```

SELECTING Target devices in the group:

```
Device: 0215 [SELECTED]
Device: 0216 [SELECTED]
Device: 0217 [SELECTED]
```

PAIRING of Source and Target devices:

```
Devices: 0101(S) - 0215(T) [PAIRED]
Devices: 0102(S) - 0216(T) [PAIRED]
Devices: 0103(S) - 0217(T) [PAIRED]
```

STARTING a Clone 'CREATE' operation.

The Clone 'CREATE' operation SUCCEEDED.

'Create' operation successfully executed for device group 'clonegrp'.

The following query displays the clone pairs and the state of each (Created).

```
# symclone -g clonegrp query -multi
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
-----			-----			-----	-----
Logical	Sym	Protected Tracks	Logical	Sym	CG SRC <=> TGT		(%)
-----			-----			-----	-----
DEV001	0101	138090	BCV001	0215	.X Created		0
DEV002	0102	138090	BCV002	0216	.X Created		0
DEV003	0103	138090	BCV003	0217	.X Created		0
Total		-----					
Track(s)		414270					
MB(s)		12945.9					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The symclone activate command activates the clone operation for all clone pairs in the device group.

```
# symclone -g clonegrp activate -noprompt
```

'Activate' operation execution is in progress for device group 'clonegrp'.
 Please wait...

'Activate' operation successfully executed for device group 'clonegrp'.

Another query displays the clone pairs and changed state of each (from Created to CopyOnAccess).

```
# symclone -g clonegrp query -multi
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
-----			-----			-----	-----
Logical	Sym	Protected Tracks	Logical	Sym	CG SRC <=> TGT		(%)
-----			-----			-----	-----
DEV001	0101	138090	BCV001	0215	.X CopyOnAccess		0
DEV002	0102	138090	BCV002	0216	.X CopyOnAccess		0
DEV003	0103	138090	BCV003	0217	.X CopyOnAccess		0

Total	-----
Track(s)	414270
MB(s)	12945.9

Legend:

(C): The background copy setting is active for this pair.

(G): The Target device is associated with this group.

The following `symclone terminate` command ends the clone pair sessions of all three clone pairs in the device group.

```
# symclone -g clonegrp terminate -noprompt
```

'Terminate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Terminate' operation successfully executed for device group 'clonegrp'.

The `symclone query` command with the `-multi` option checks the device group for any current copy sessions. Because all copy sessions for devices in the device group were stopped, the output below confirms that there are no longer any clone devices in the group.

```
# symclone -g clonegrp query -multi
```

Device group 'clonegrp' does not have any devices that are Clone source devices

The `symclone create` command creates three new clone pair sessions with the same standards and BCVs. However, this time, the `-copy` option is used to set up the clone operation for *immediate* copying of an entire source device instead of the deferred clone operation of CopyOnAccess.

```
# symclone -g clonegrp create -copy -noprompt
```

'Create' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Create' operation successfully executed for device group 'clonegrp'.

This `symclone activate` command also includes `-preaction` and `-postaction` options to run scripts named `pre.sh` and `post.sh`, respectively. An example of a preaction script would be one that unmounts a file system on the target devices; a postaction script might be one that mounts the file system on the target devices.

```
# symclone -g clonegrp activate -noprompt -preaction ./pre.sh \
                                     -postaction ./post.sh
```

'Activate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Activate' operation successfully executed for device group 'clonegrp'.

The following query shows the state of the clone pairs (CopyInProgress).

```
# symclone -g clonegrp query -multi
```

```
Device Group (DG) Name: clonegrp
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
DEV001	0101	134341	BCV001	0215	XX	CopyInProgress	2
DEV002	0102	135729	BCV002	0216	XX	CopyInProgress	1
DEV003	0103	134205	BCV003	0217	XX	CopyInProgress	2
Total		-----					
Track(s)		404275					
MB(s)		12633.6					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The `symclone verify` command with the `-copied` option checks all clone pairs every 120 seconds until all are in the Copied state.

```
# symclone -g clonegrp verify -copied -i 120
```

None of the devices in group 'clonegrp' are in the 'Copied' state.

None of the devices in group 'clonegrp' are in the 'Copied' state.

All of the devices in group 'clonegrp' are in the 'Copied' state.

One `symclone terminate` command can stop multiple copy sessions. Stopping a copy session deletes clone pair information from the SYMAPI database and removes any hold on a target device.

```
# symclone -g clonegrp terminate -noprompt DEV001 sym 1d BCV001 \
                                         DEV002 sym 1d BCV002 \
                                         DEV003 sym 1d BCV003
```

'Terminate' operation execution is in progress for device group 'clonegrp'.
 Please wait...

'Terminate' operation successfully executed for device group 'clonegrp'.

The `symclone create` command with the `-optimize` option uses optimization to create clone pairs from the devices in the device group. Optimization attempts to match devices in such a way as to achieve a high-speed copy between them, first trying to pair devices on separate disk adapters (DA) and, failing that, devices on the same disk adapter but on different SCSI interfaces of the disk adapter and, failing that, devices that may be on the same DA and SCSI interface but on a separate spindle.

```
# symclone -g clonegrp create -optimize -v -noprompt
```

'Create' operation execution is in progress for device group 'clonegrp'.
Please wait...

SELECTING Source devices in the group:

```
Device: 0101 [SELECTED]
Device: 0102 [SELECTED]
Device: 0103 [SELECTED]
```

SELECTING Target devices in the group:

```
Device: 0215 [SELECTED]
Device: 0216 [SELECTED]
Device: 0217 [SELECTED]
```

PAIRING of Source and Target devices:

```
Devices: 0101(S) - 0216(T) [PAIRED]
Devices: 0102(S) - 0215(T) [PAIRED]
Devices: 0103(S) - 0217(T) [PAIRED]
```

STARTING a Clone 'CREATE' operation.

The Clone 'CREATE' operation SUCCEEDED.

'Create' operation successfully executed for device group 'clonegrp'.

The following query displays the new clone pairs. Optimization matched two of the three pairs differently than the previous clone operation: DEV001 with BCV002, and DEV002 with BCV001.

```
# symclone -g clonegrp query -multi
```

```
Device Group (DG) Name: clonegrp
DG's Type                : REGULAR
DG's Symmetrix ID        : 000187900041
```

Source Device			Target Device			State	Copy
-----			-----			-----	-----
Logical	Sym	Protected Tracks	Logical	Sym	CG SRC <=> TGT		(%)
-----			-----			-----	-----
DEV001	0101	138090	BCV002	0216	.X Created		0
DEV002	0102	138090	BCV001	0215	.X Created		0
DEV003	0103	138090	BCV003	0217	.X Created		0

Total	-----
Track(s)	414270
MB(s)	12945.9

Legend:

(C): The background copy setting is active for this pair.

(G): The Target device is associated with this group.

The `symclone activate` command activates the clone operation for all clone pairs in the device group.

```
# symclone -g clonegrp activate -noprompt
```

'Activate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Activate' operation successfully executed for device group 'clonegrp'.

The `symclone verify` command with the `-copyonaccess` option checks all clone pairs every 120 seconds until all are in the CopyOnAccess state.

```
# symclone -g clonegrp verify -copyonaccess -i 120
```

All devices in group 'clonegrp' have a paired TGT device that is in the 'CopyOnAccess' state.

The following `symclone terminate` command initiates termination of the three copy sessions.

```
# symclone -g clonegrp terminate -noprompt
```

'Terminate' operation execution is in progress for device group 'clonegrp'.
Please wait...

'Terminate' operation successfully executed for device group 'clonegrp'.

The `symclone list` command indicates that the copy sessions were successfully terminated and no longer exist.

```
# symclone list
```

Symmetrix ID: 000187900041

No Copy sessions found

Example 3: Cloning Copies from a BCV Source Device

This example creates a new device group and pairs Symmetrix standard device 0101 with BCV 0215 as a BCV pair. The source of the clone is BCV 0215. The target is standard device 0102.

The `symdg` command creates a device group named `clonebcv`. The `syml` commands add two standard devices to the group. By default, the logical device name of device 0101 is `DEV001`. The example assigns `CLONETGT` as the logical device name for device 0102. The `syml` command associates one BCV with the group. The example assigns `CLONESRC` as the BCV's logical device name.

```
# symdg create clonebcv
# syml -g clonebcv add dev 101
# syml -g clonebcv add dev 102 CLONETGT
# syml -g clonebcv associate dev 215 CLONESRC
```

The `symdg show` command displays the details of the device group. The logical device names `CLONETGT` and `CLONESRC` make it easier to identify the clone target and clone source.

```
# symdg show clonebcv
```

Group Name: clonebcv

```
Group Type           : REGULAR
Valid                : Yes
Symmetrix ID         : 000187900041
Group Creation Time  : Tue Mar 18 14:15:31 2003
Vendor ID            : EMC Corp
Application ID       : SYMCLI
```

```
Number of STD Devices in Group : 2
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 1
Number of Locally-associated VDEV's : 0
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 0
```

Standard (STD) Devices (2):

```
{
-----
LdevName          PdevName          Sym      Cap
                  Dev    Att.  Sts    (MB)
-----
DEV001             /dev/vx/rdmp/c4t0d160s2 0101      RW      4315
CLONETGT           /dev/vx/rdmp/c4t0d161s2 0102      RW      4315
}
```

BCV Devices Locally-associated (1):

```
{
-----
LdevName          PdevName          Sym      Cap
                  Dev    Att.  Sts    (MB)
-----
CLONESRC          N/A              0215      RW      4315
}
```

The `symmir establish` command initiates a full establish operation on the BCV pair identified by their logical device names (standard device DEV001 with the BCV named CLONESRC).

```
# symmir -g clonebcv establish -full DEV001 bcv ld CLONESRC -noprompt
```

'Full Establish' operation execution is in progress for device 'DEV001' paired with BCV device 'CLONESRC' in device group 'clonebcv'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in group 'clonebcv' paired with BCV device 'CLONESRC'.

The `symmir verify` command checks the establish operation every five seconds and verifies when the operation is complete. That is, DEV001 has been fully copied to CLONESRC.

```
# symmir -g clonebcv verify DEV001 -i 5
```

Device 'DEV001' in group 'clonebcv' is NOT in the 'Synchronized or Restored' state.

Device 'DEV001' in group 'clonebcv' is NOT in the 'Synchronized or Restored' state.

Device 'DEV001' in group 'clonebcv' is in the 'Synchronized or Restored' state.

The `symmir split` command performs an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation.

```
# symmir -g clonebcv split DEV001 -instant -not_ready -noprompt
```

'Split' operation execution is in progress for device 'DEV001' in device group 'clonebcv'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group 'clonebcv'.

The `symmir verify` command checks the status of the background split every five seconds until it completes.

```
# symmir -g clonebcv verify DEV001 -split -bg
```

Device 'DEV001' in group 'clonebcv' has finished splitting in the background.

The following command, by default, creates a CopyOnAccess copy session for source device CLONESRC and target device CLONETGT.

```
# symclone -g clonebcv create CLONESRC sym ld CLONETGT -noprompt
```

'Create' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group 'clonebcv' paired with target device 'CLONETGT'.

The following query displays the new clone pair and its state (Created).

symclone -g clonebcv query

Device Group (DG) Name: clonebcv
 DG's Type : REGULAR
 DG's Symmetrix ID : 000187900041

Source Device			Target Device			State	Copy
-----			-----			-----	-----
Logical	Sym	Protected Tracks	Logical	Sym	CG SRC <=> TGT		(%)
-----			-----			-----	-----
CLONESRC	0215	138090	CLONETGT	0102	.X Created		0
Total		-----					
MB(s)		4315.3					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The **symdev show** command displays the details of device 0215 (the CLONESRC device).

symdev show 215

Symmetrix ID: 000187900041

```

Device Physical Name      : Not Visible

Device Symmetrix Name     : 0215
Device Serial ID         : N/A
Symmetrix ID             : 000187900041

Device Group Name        : clonebcv
Device Logical Name      : CLONESRC

Vendor ID                : EMC
Product ID               : SYMMETRIX
Product Revision         : 5669

Device Emulation Type    : FBA
Device Defined Label Type: N/A
Device Defined Label     : N/A
Device Sub System Id     : 0x0142

Device Block Size        : 512
  
```

```

Device Capacity
{
  Cylinders      :      9206
  Tracks         :      138090
  512-byte Blocks :      8837760
  MegaBytes      :      4315
  KiloBytes      :      4418880
}

Device Configuration      : 2-Way BCV Mir      (Non-Exclusive Access)

Device is WORM Enabled    : No
Device is WORM Protected : No

SCSI-3 Persistent Reserve: Disabled

Dynamic Spare Invoked     : No

Dynamic RDF Capability     : None

Device Service State      : Normal

Device Status             : Not Ready          (NR)
Device SA Status          : N/A                (N/A)

Mirror Set Type           : [Data,Data,N/A,N/A]

Mirror Set DA Status      : [RW,RW,N/A,N/A]

Mirror Set Inv. Tracks    : [0,112458,0,0]

Back End Disk Director Information
{
  Hyper Type              : Data
  Hyper Status             : Ready              (RW)
  Disk [Director, Interface, TID] : [01A, C, 4]
  Disk Director Volume Number : 61
  Hyper Number             : 9
  Disk Capacity            : 69879m

  Hyper Type              : Data
  Hyper Status             : Ready              (RW)
  Disk [Director, Interface, TID] : [15A, C, 4]
  Disk Director Volume Number : 61
  Hyper Number             : 9
  Disk Capacity            : 69879m
}

```

```

BCV Pair Information
{
  Standard (STD) Device Symmetrix Name      : 0101
  Standard (STD) Device Serial ID           : 41101330
  Standard (STD) Device Group Name          : clonebcv

  BCV Device Symmetrix Name                  : 0215
  BCV Device Serial ID                       : Not Visible
  BCV Device Associated Group Name           : clonebcv

  BCV Device Status                          : Not Ready          (NR BCV)

  State of Pair ( STD <=\=> BCV )            : Split
  Time of Last BCV Action                    : Tue Mar 18 14:28:02 2003

  State of BCV Mirrors                       : SyncInProgress

  BCV State Flags                           : (AllReady)
  Percent Split                             : 100%

  Number of Inv. Tracks for STD Device       :          0
  Number of Inv. Tracks for BCV Device       :          30
}

```

```

Clone Device Information
{
  Source (SRC) Device Symmetrix Name         : 0215
  Source (SRC) Device Group Name             : clonebcv

  Target (TGT) Device Symmetrix Name         : 0102
  Target (TGT) Device Group Name             : clonebcv

  State of Session ( SRC ==> TGT )           : Created
  Percent Copied                             : 0%

  Time of Last Clone Action                   : Tue Mar 18 14:29:14 2003

  Clone State Flags                          : None

  Number of Prot. Tracks for SRC Device       : 138090
  Number of Indir Tracks for TGT Device      : 138090
}

```

The `symclone activate` command activates the clone operation for the clone pair.

```
# symclone -g clonebcv activate CLONESRC sym ld CLONETGT -noprompt
```

'Activate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Activate' operation successfully executed for device 'CLONESRC' in group 'clonebcv' paired with target device 'CLONETGT'.

The `symclone query` command displays the state of the clone operation (CopyOnAccess).

```
# symclone -g clonebcv query
```

```
Device Group (DG) Name: clonebcv
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
CLONESRC	0215	138090	CLONETGT	0102	.X	CopyOnAccess	0
Total		-----					
MB(s)		4315.3					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The following `symclone terminate` command stops the copy session. The verbose (`-v`) option provides a more detailed output.

```
# symclone -g clonebcv terminate CLONESRC sym ld CLONETGT -noprompt -v
```

'Terminate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT' in device group 'clonebcv'. Please wait...

PAIRING of Source and Target devices:

Devices: 0215(S) - 0102(T) [PAIRED]

STARTING a Clone 'TERMINATE' operation.

The Clone 'TERMINATE' operation SUCCEEDED.

'Terminate' operation successfully executed for device 'CLONESRC' in group 'clonebcv' paired with target device 'CLONETGT'.

The `symmir query` command displays that the copy session was stopped by indicating that devices in the device group are no longer listed in the SYMAPI database as clone devices.

```
# symclone -g clonebcv query
```

Device group 'clonebcv' does not have any devices that are Clone source devices

The `symld list` command with the `-held` option displays that the hold on the target device was released by indicating that no devices in the device group are being held.

```
# symld -g clonebcv -held list
```

```
Device Group (DG) Name: clonebcv
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

```
No devices are held
```

The `symmir establish` command initiates an incremental establish on the BCV pair, which refreshes the BCV with any new writes that occurred on DEV001 while the BCV pair was split.

```
# symmir -g clonebcv establish DEV001 bcv ld CLONESRC -noprompt
```

```
'Incremental Establish' operation execution is in progress for device 'DEV001'
paired with BCV device 'CLONESRC' in device group 'clonebcv'. Please wait...
```

```
'Incremental Establish' operation successfully initiated for device 'DEV001'
in group 'clonebcv' paired with BCV device 'CLONESRC'.
```

The `symmir verify` command verifies that the establish operation is complete.

```
# symmir -g clonebcv verify DEV001
```

```
Device 'DEV001' in group 'clonebcv' is in the 'Synchronized or Restored' state.
```

The `symmir split` command performs an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the next clone operation.

```
# symmir -g clonebcv split DEV001 -instant -not_ready -noprompt
```

```
'Split' operation execution is in progress for device 'DEV001' in device group
'clonebcv'. Please wait...
```

```
'Split' operation successfully executed for device 'DEV001' in group 'clonebcv'.
```

The `symmir verify` command checks every five seconds and verifies when the background split completes.

```
# symmir -g clonebcv verify DEV001 -split -bg -i 5
```

```
Device 'DEV001' in group 'clonebcv' has finished splitting in the background.
```

The following command creates a new CopyOnAccess copy session for source device CLONESRC and target device CLONETGT, in preparation for cloning the refreshed data on the BCV source device.

```
# symclone -g clonebcv create CLONESRC sym ld CLONETGT -noprompt
```

```
'Create' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT' in device group 'clonebcv'. Please wait...
```

```
'Create' operation successfully executed for device 'CLONESRC' in group
'clonebcv' paired with target device 'CLONETGT'.
```

The `symclone activate` command activates the clone operation for the clone pair.

```
# symclone -g clonebcv activate CLONESRC sym ld CLONETGT -noprompt
```

'Activate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT' in device group 'clonebcv'. Please wait...

'Activate' operation successfully executed for device 'CLONESRC' in group 'clonebcv' paired with target device 'CLONETGT'.

The following command makes the BCV device ready again to its host. Reads or writes to the BCV source will clone data to the target.

```
# symld -g clonebcv ready -bcv CLONESRC -noprompt
```

'Ready' Device operation successfully completed for the device.

After cloning operations are complete, terminate the clone pair session.

```
# symclone -g clonebcv terminate CLONESRC sym ld CLONETGT -noprompt -v
```

'Terminate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT' in device group 'clonebcv'. Please wait...

PAIRING of Source and Target devices:

Devices: 0215(S) - 0102(T) [PAIRED]

STARTING a Clone 'TERMINATE' operation.

The Clone 'TERMINATE' operation SUCCEEDED.

'Terminate' operation successfully executed for device 'CLONESRC' in group 'clonebcv' paired with target device 'CLONETGT'.

The following commands perform cleanup tasks so that these same devices are available for use in Example 4, which follows. The `symmir cancel` command cancels the BCV pair relationship between DEV001 and CLONESRC so that the Symmetrix array no longer recognizes this relationship.

```
# symmir -g clonebcv cancel DEV001 bcv ld CLONESRC
```

Execute 'Cancel' operation for device 'DEV001' in device group 'clonebcv' (y/[n]) ? **y**

'Cancel' operation execution is in progress for device 'DEV001' paired with BCV device 'CLONESRC' in device group 'clonebcv'. Please wait...

'Cancel' operation successfully executed for device 'DEV001' in group 'clonebcv' paired with BCV device 'CLONESRC'.

The `symdg delete` command deletes the device group clonebcv. The `-force` option is required when there are still devices in the device group.

```
# symdg -force delete clonebcv
```

Example 4: Cloning Multiple Copies of a BCV

This example creates a new device group and pairs Symmetrix standard device 00FF with BCV 0215 as a BCV pair. The clone source is BCV 0215. The targets are standard devices 0100, 0101, 0102, and 0103.

The `symdmg` command creates a device group named `multiclone`. The first `symld` command adds standard device 00FF to the group. By default, the logical device name of device 00FF is DEV001. The `symbcv` command associates the source BCV with the group. The example assigns CLONESRC as the BCV's logical device name to make it easier to identify the clone source device. The remaining `symld` commands add four target standard devices (0100, 0101, 0102, and 0103) and assign CLONETGT logical device names that make it easier to identify the clone target devices.

```
# symdmg create multiclone
# symld -g multiclone add dev FF
# symbcv -g multiclone associate dev 215 CLONESRC
# symld -g multiclone add dev 100 CLONETGT1
# symld -g multiclone add dev 101 CLONETGT2
# symld -g multiclone add dev 102 CLONETGT3
# symld -g multiclone add dev 103 CLONETGT4
```

The `symdmg show` command displays the details of the device group named `multiclone`.

```
# symdmg show multiclone
```

Group Name: multiclone

```
Group Type           : REGULAR
Valid                : Yes
Symmetrix ID         : 000187900041
Group Creation Time  : Tue Mar 18 14:51:12 2003
Vendor ID            : EMC Corp
Application ID       : SYMCLI
```

```
Number of STD Devices in Group : 5
Number of Associated GK's      : 0
Number of Locally-associated BCV's : 1
Number of Locally-associated VDEV's : 0
Number of Remotely-associated BCV's (STD RDF) : 0
Number of Remotely-associated BCV's (BCV RDF) : 0
```

Standard (STD) Devices (5):

```
{
-----
LdevName          PdevName          Sym   Dev   Att. Sts   Cap
                    Dev   Att. Sts   (MB)
-----
DEV001             /dev/vx/rdmp/c4t0d158s2 00FF   RW   4315
CLONETGT1          /dev/vx/rdmp/c4t0d159s2 0100   RW   4315
CLONETGT2          /dev/vx/rdmp/c4t0d160s2 0101   RW   4315
CLONETGT3          /dev/vx/rdmp/c4t0d161s2 0102   RW   4315
CLONETGT4          /dev/vx/rdmp/c4t0d162s2 0103   RW   4315
}
```

BCV Devices Locally-associated (1):

LdevName	PdevName	Sym Dev	Att. Sts	Cap (MB)

CLONESRC	N/A	0215	RW	4315
}				

The `symmir establish` command initiates a full establish operation on the BCV pair identified by their logical device names (standard device DEV001 with the BCV named CLONESRC).

```
# symmir -g multiclone establish -full DEV001 bcv ld CLONESRC -noprompt
```

'Full Establish' operation execution is in progress for device 'DEV001' paired with BCV device 'CLONESRC' in device group 'multiclone'. Please wait...

'Full Establish' operation successfully initiated for device 'DEV001' in group 'multiclone' paired with BCV device 'CLONESRC'.

The `symmir verify` command checks the establish operation every five seconds and verifies when the operation is complete. That is, DEV001 has been fully copied to CLONESRC.

```
# symmir -g multiclone verify DEV001 -i 5
```

Device 'DEV001' in group 'multiclone' is NOT in the 'Synchronized or Restored' state.

Device 'DEV001' in group 'multiclone' is NOT in the 'Synchronized or Restored' state.

Device 'DEV001' in group 'multiclone' is in the 'Synchronized or Restored' state.

The `symmir split` command performs an instant split on the BCV pair. Use the `-not_ready` option to prevent the BCV's host from writing to it prior to the clone operation.

```
# symmir -g multiclone split -instant -not_ready DEV001 bcv ld CLONESRC -noprompt
```

'Split' operation execution is in progress for device 'DEV001' paired with BCV device 'CLONESRC' in device group 'multiclone'. Please wait...

'Split' operation successfully executed for device 'DEV001' in group 'multiclone' paired with BCV device 'CLONESRC'.

The `symmir verify` command checks the status of the background split every five seconds until it completes.

```
# symmir -g multiclone verify -bg -split DEV001 -i 5
```

Device 'DEV001' in group 'multiclone' has finished splitting in the background.

The following command creates a copy session for source device CLONESRC and target device CLONETGT1, specifying the `-copy` option for a full copy of the source device.

```
# symclone -g multiclone create -copy CLONESRC sym ld CLONETGT1 -noprompt
```

'Create' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT1' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT1'.

The following command creates a copy session for source device CLONESRC and target device CLONETGT2, specifying the `-copy` option for a full copy of the source device.

```
# symclone -g multiclone create -copy CLONESRC sym ld CLONETGT2 -noprompt
```

'Create' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT2' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT2'.

The following command creates a copy session for source device CLONESRC and target device CLONETGT3, specifying the `-copy` option for a full copy of the source device.

```
# symclone -g multiclone create -copy CLONESRC sym ld CLONETGT3 -noprompt
```

'Create' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT3' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT3'.

The following command creates a copy session for source device CLONESRC and target device CLONETGT4, specifying the `-copy` option for a full copy of the source device.

```
# symclone -g multiclone create -copy CLONESRC sym ld CLONETGT4 -noprompt
```

'Create' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT4' in device group 'multiclone'. Please wait...

'Create' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT4'.

The `symclone query` command with the `-multi` option displays that the source device and its four clone targets have been successfully created.

```
# symclone -g multiclone query -multi
```

```
Device Group (DG) Name: multiclone
```

```
DG's Type : REGULAR
```

```
DG's Symmetrix ID : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
CLONESRC	0215	138090	CLONETGT4	0103	XX	Created	0
		138090	CLONETGT3	0102	XX	Created	0
		138090	CLONETGT2	0101	XX	Created	0
		138090	CLONETGT1	0100	XX	Created	0
Total		-----					
Track(s)		552360					
MB(s)		17261.2					

Legend:

(C): The background copy setting is active for this pair.

(G): The Target device is associated with this group.

The following `symclone activate` command activates the four copy sessions at the same time.

```
# symclone -g multiclone activate CLONESRC sym ld CLONETGT1 \
                                CLONESRC sym ld CLONETGT2 \
                                CLONESRC sym ld CLONETGT3 \
                                CLONESRC sym ld CLONETGT4
```

```
Execute 'Activate' operation for the 4 specified devices in device group
'multiclone' (y/[n]) ? y
```

```
'Activate' operation execution is in progress for the device list in device group
'multiclone'. Please wait...
```

```
'Activate' operation successfully executed for the device list in device group
'multiclone'.
```

The following command makes the BCV device ready again to its host.

```
# symld -g multiclone ready -bcv CLONESRC -noprompt
```

```
'Ready' Device operation successfully completed for the device.
```

The following `symclone` query command with the `-multi` option displays the state of the clone operation, showing that all four copy operations are still in progress (CopyInProg) and the completion percent of each.

```
# symclone -g multiclone query -multi
```

```
Device Group (DG) Name: multiclone
DG's Type           : REGULAR
DG's Symmetrix ID   : 000187900041
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
CLONESRC	0215	129275	CLONETGT1	0100	XX	CopyInProg	6
		129277	CLONETGT2	0101	XX	CopyInProg	6
		129276	CLONETGT3	0102	XX	CopyInProg	6
		129276	CLONETGT4	0103	XX	CopyInProg	6
Total		-----					
Track(s)		517104					
MB(s)		16159.5					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The `symclone verify` command checks all clone pairs every 60 seconds until all are in the Copied state.

```
# symclone -g multiclone verify CLONESRC -copied -i 60
```

```
Device 'CLONESRC' in group 'multiclone' is NOT in the 'Copied' state.
```

```
Device 'CLONESRC' in group 'multiclone' is NOT in the 'Copied' state.
```

```
Device 'CLONESRC' in group 'multiclone' is in the 'Copied' state.
```

The following commands terminate the copy sessions. Stopping a copy session deletes clone pair information from the SYMAPI database and removes any hold on a target device.

```
# symclone -g multiclone terminate CLONESRC sym ld CLONETGT1 -noprompt
```

```
'Terminate' operation execution is in progress for device 'CLONESRC' paired with
target device 'CLONETGT1' in device group 'multiclone'. Please wait...
```

```
'Terminate' operation successfully executed for device 'CLONESRC' in group
'multiclone' paired with target device 'CLONETGT1'.
```

```
# symclone -g multiclone terminate CLONESRC sym ld CLONETGT2 -noprompt
```

'Terminate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT2' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT2'.

```
# symclone -g multiclone terminate CLONESRC sym ld CLONETGT3 -noprompt
```

'Terminate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT3' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT3'.

```
# symclone -g multiclone terminate CLONESRC sym ld CLONETGT4 -noprompt
```

'Terminate' operation execution is in progress for device 'CLONESRC' paired with target device 'CLONETGT4' in device group 'multiclone'. Please wait...

'Terminate' operation successfully executed for device 'CLONESRC' in group 'multiclone' paired with target device 'CLONETGT4'.

The `symmir establish` command initiates an incremental establish on the BCV pair, which refreshes the BCV with any new writes that occurred on DEV001 while the BCV pair was split.

```
# symmir -g multiclone establish DEV001 bcv ld CLONESRC -noprompt
```

'Incremental Establish' operation execution is in progress for device 'DEV001' paired with BCV device 'CLONESRC' in device group 'multiclone'. Please wait...

'Incremental Establish' operation successfully initiated for device 'DEV001' in group 'multiclone' paired with BCV device 'CLONESRC'.

The `symmir verify` command checks the status of the establish operation every five seconds and verifies when the operation is complete.

```
# symmir -g multiclone verify DEV001 -i 5
```

Device 'DEV001' in group 'multiclone' is in the 'Synchronized or Restored' state.

Example 5: Cloning 16 Copies from a Single Source Device

This example creates a new device group containing a single standard device (023E) as a clone source and 16 BCV devices (0257 – 0266) as clone targets.

The `symdmg` command creates a device group named `clone16`. The `symld` command adds standard device 023E to the group. By default, the logical device name of device 023E is DEV001. The `symbcv` command associates a range of 16 BCV devices (0257 through 0266) with the group and assigns them the logical device names BCV001 through BCV016.

```
# symdmg create clone16
# symld -g clone16 add dev 23e
# symbcv -g clone16 associateall dev -range 257:266
```

When cloning to multiple targets from the same source device, you must create each of the 16 source/target pairings separately. The first `symclone create` command creates the DEV001/BCV001 clone pair and displays the resulting output from the command. The rest of the `symclone create` commands create the remaining 15 clone pairs (these outputs are omitted for brevity). By default, each clone pair session created will be a CopyOnAccess session.

```
# symclone -g clone16 create DEV001 sym ld BCV001 -noprompt
```

```
'Create' operation execution is in progress for device 'DEV001' paired with
target device 'BCV001' in device group 'clone16'. Please wait...
```

```
'Create' operation successfully executed for device 'DEV001' in group
'clone16' paired with target device 'BCV001'.
```

```
# symclone -g clone16 create DEV001 sym ld BCV002 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV003 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV004 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV005 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV006 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV007 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV008 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV009 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV010 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV011 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV012 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV013 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV014 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV015 -noprompt
# symclone -g clone16 create DEV001 sym ld BCV016 -noprompt
```

The `symclone query` command with the `-multi` option displays that the source device and its 16 clone targets have been successfully created. Note that the last clone pair created is located at the top of the display list.

```
# symclone -g clone16 query -multi
```

```
Device Group (DG) Name: clone16
DG's Type           : REGULAR
DG's Symmetrix ID   : 000000005231
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
DEV001	023E	138090	BCV016	0266	XX	Created	0
		138090	BCV015	0265	XX	Created	0
		138090	BCV014	0264	XX	Created	0
		138090	BCV013	0263	XX	Created	0
		138090	BCV012	0262	XX	Created	0
		138090	BCV011	0261	XX	Created	0
		138090	BCV010	0260	XX	Created	0
		138090	BCV009	025F	XX	Created	0
		138090	BCV008	025E	XX	Created	0
		138090	BCV007	025D	XX	Created	0
		138090	BCV006	025C	XX	Created	0
		138090	BCV005	025B	XX	Created	0
		138090	BCV004	025A	XX	Created	0
		138090	BCV003	0259	XX	Created	0
		138090	BCV002	0258	XX	Created	0
		138090	BCV001	0257	XX	Created	0
Total		-----					
Track(s)		2209440					
MB(s)		69045.0					

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with this group.

The following `symclone activate` command activates the 16 clone pair sessions at the same time. When you include the `-consistent` option, SYMAPI suspends any further I/O before activation. Suspending I/O during the brief period when activation occurs prevents any pending records from being included in the clone copy. When activation completes, SYMAPI allows I/O to resume.

```
# symclone -g clone16 activate -consistent \
    DEV001 sym ld BCV001 \
    DEV001 sym ld BCV002 \
    DEV001 sym ld BCV003 \
    DEV001 sym ld BCV004 \
    DEV001 sym ld BCV005 \
    DEV001 sym ld BCV006 \
    DEV001 sym ld BCV007 \
    DEV001 sym ld BCV008 \
    DEV001 sym ld BCV009 \
    DEV001 sym ld BCV010 \
    DEV001 sym ld BCV011 \
    DEV001 sym ld BCV012 \
    DEV001 sym ld BCV013 \
    DEV001 sym ld BCV014 \
    DEV001 sym ld BCV015 \
    DEV001 sym ld BCV016 -noprompt
```

'Activate' operation execution is in progress for the device list in device group 'clone16'. Please wait...

'Activate' operation successfully executed for the device list in device group 'clone16'.

The `symclone query` command with the `-multi` option displays that the source device and its 16 clone targets are in the CopyOnAccess state. You can now clone data to any four targets simultaneously. You can continue to clone data to these targets as long as these copy sessions exist.

```
# symclone -g clone16 query -multi
```

```
Device Group (DG) Name: clone16
DG's Type                : REGULAR
DG's Symmetrix ID        : 000000005231
```

Source Device			Target Device			State	Copy
Logical	Sym	Protected Tracks	Logical	Sym	CG	SRC <=> TGT	(%)
DEV001	023E	138090	BCV016	0266	XX	CopyOnAccess	0
		138090	BCV015	0265	XX	CopyOnAccess	0
		138090	BCV014	0264	XX	CopyOnAccess	0
		138090	BCV013	0263	XX	CopyOnAccess	0
		138090	BCV012	0262	XX	CopyOnAccess	0
		138090	BCV011	0261	XX	CopyOnAccess	0
		138090	BCV010	0260	XX	CopyOnAccess	0
		138090	BCV009	025F	XX	CopyOnAccess	0
		138090	BCV008	025E	XX	CopyOnAccess	0
		138090	BCV007	025D	XX	CopyOnAccess	0
		138090	BCV006	025C	XX	CopyOnAccess	0

138090	BCV005	025B	XX	CopyOnAccess	0
138090	BCV004	025A	XX	CopyOnAccess	0
138090	BCV003	0259	XX	CopyOnAccess	0
138090	BCV002	0258	XX	CopyOnAccess	0
138090	BCV001	0257	XX	CopyOnAccess	0

Total	-----
Track(s)	2209440
MB(s)	69045.0

Legend:

(C): The background copy setting is active for this pair.

(G): The Target device is associated with this group.

When you are finished cloning data to these targets, you need to terminate each of the 16 source/target pairings separately. The first `symclone terminate` command terminates the DEV001/BCV001 clone pair session and displays the resulting output from the command. The rest of the `symclone terminate` commands terminate the remaining 15 clone pairs (these outputs are omitted for brevity).

```
# symclone -g clone16 terminate DEV001 sym ld BCV001 -noprompt
```

'Terminate' operation execution is in progress for device 'DEV001' paired with target device 'BCV001' in device group 'clone16'. Please wait...

'Terminate' operation successfully executed for device 'DEV001' in group 'clone16' paired with target device 'BCV001'.

```
# symclone -g clone16 terminate DEV001 sym ld BCV002 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV003 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV004 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV005 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV006 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV007 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV008 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV009 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV010 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV011 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV012 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV013 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV014 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV015 -noprompt
# symclone -g clone16 terminate DEV001 sym ld BCV016 -noprompt
```

Example 6: Using a Composite Group to Control Clone Pairs

This example was performed using Solutions Enabler version 5.4. The hardware setup includes a host connected to two source Symmetrix arrays (SIDs 35 and 60). A composite group defined on the host contains a set of clone pairs that spans the two Symmetrix arrays.

(Beginning with Solutions Enabler version 6.0, you can create, activate, and control specific clone pairs within the composite group as well as performing control operations on the entire group as shown here. For example, you can terminate one clone session in the group without terminating the group's other sessions.)

The `symcg create` command creates a Regular type composite group named TFClone.

```
# symcg create TFClone -type regular
```

The following `symcg` commands add to the composite group a range of standard devices from each of the two source Symmetrix arrays. These are the source devices.

```
# symcg -cg TFClone addall dev -range 56:6E -sid 35
# symcg -cg TFClone addall dev -range 14:27 -sid 60
```

The following `symbcv` commands add to the composite group a range of BCV devices from each of the two source Symmetrix arrays. These are the target devices.

```
# symbcv -cg TFClone associateall dev -range 182:19A -sid 35
# symbcv -cg TFClone associateall dev -range 3B6:3C9 -sid 60
```

The `symclone create` command creates optimized standard/BCV pairings among devices within each Symmetrix array and places each clone pair in the Created state. The `-copy` option specifies that full copying be performed upon activation.

```
# symclone -cg TFClone create -copy -optimize -noprompt
```

```
'Create' operation execution is in progress for composite group 'TFClone'.
Please wait...
```

```
'Create' operation successfully executed for composite group 'TFClone'.
```

The `symclone query` command displays the clone pairs and that they are in the Created state. Note that devices 0056 and 0059 are meta devices (as are their target devices), and only the meta head device is displayed.

`symclone -cg TFClone query`

Composite Group Name : TFClone
 Composite Group Type : REGULAR
 Number of Symmetrix Units : 2
 Number of RDF (RA) Groups : 2

Symmetrix ID : 000187900035 (Microcode Version: 5670)

Source Device		Target Device		State	Copy
-----		-----		-----	-----
Sym	Protected Tracks	Sym	CG	SRC <=> TGT	(%)
-----		-----		-----	-----
0056	414270	0182	XX	Created	0
0059	276180	0185	XX	Created	0
005B	138090	0187	XX	Created	0
005C	138090	0188	XX	Created	0
005D	138090	0189	XX	Created	0
005E	138090	018A	XX	Created	0
005F	138090	018B	XX	Created	0
0060	138090	018C	XX	Created	0
0061	138090	018D	XX	Created	0
0062	138090	018E	XX	Created	0
0063	138090	018F	XX	Created	0
0064	138090	0190	XX	Created	0
0065	138090	0191	XX	Created	0
0066	138090	0192	XX	Created	0
0067	138090	0193	XX	Created	0
0068	138090	0194	XX	Created	0
0069	138090	0195	XX	Created	0
006A	138090	0196	XX	Created	0
006B	138090	0197	XX	Created	0
006C	138090	0198	XX	Created	0
006D	138090	0199	XX	Created	0
006E	138090	019A	XX	Created	0

Symmetrix ID : 000184500160 (Microcode Version: 5568)

Source Device		Target Device		State	Copy
-----		-----		-----	-----
Sym	Protected Tracks	Sym	CG	SRC <=> TGT	(%)
-----		-----		-----	-----
0014	138090	03B6	XX	Created	0
0015	138090	03B7	XX	Created	0
0016	138090	03B8	XX	Created	0
0017	138090	03B9	XX	Created	0
0018	138090	03BA	XX	Created	0
0019	138090	03BB	XX	Created	0
001A	138090	03BC	XX	Created	0

001B	138090	03BD	XX Created	0
001C	138090	03BF	XX Created	0
001D	138090	03BE	XX Created	0
001E	138090	03C0	XX Created	0
001F	138090	03C1	XX Created	0
0020	138090	03C2	XX Created	0
0021	138090	03C3	XX Created	0
0022	138090	03C4	XX Created	0
0023	138090	03C5	XX Created	0
0024	138090	03C6	XX Created	0
0025	138090	03C7	XX Created	0
0026	138090	03C8	XX Created	0
0027	138090	03C9	XX Created	0

Total	-----
Track(s)	6214050
MB(s)	194189

Legend:

(C): The background copy setting is active for this pair.

(G): The Target device is associated with a composite group.

The `symclone activate` command places each clone pair in the CopyInProgress state. Full copying of the data begins immediately. The `-consistent` option causes the clone pairs to be consistently activated.

symclone -cg TFClone activate -consistent -noprompt

'Activate' operation execution is in progress for composite group 'TFClone'.
Please wait...

'Activate' operation successfully executed for composite group 'TFClone'.

The `symclone query` command displays the clone pairs and their CopyInProgress state.

symclone -cg TFClone query

Composite Group Name	:	TFClone
Composite Group Type	:	REGULAR
Number of Symmetrix Units	:	2
Number of RDF (RA) Groups	:	2

Symmetrix ID	:	000187900035	(Microcode Version: 5670)
--------------	---	--------------	---------------------------

Source Device		Target Device	State	Copy
Sym	Protected Tracks	Sym	CG SRC <=> TGT	(%)
0056	412378	0182	XX CopyInProgress	0
0059	276180	0185	XX CopyInProgress	0
005B	138090	0187	XX CopyInProgress	0
005C	138090	0188	XX CopyInProgress	0
005D	135506	0189	XX CopyInProgress	1
005E	138090	018A	XX CopyInProgress	0
005F	136329	018B	XX CopyInProgress	1
0060	138090	018C	XX CopyInProgress	0

0061	138090	018D	XX	CopyInProg	0
0062	138090	018E	XX	CopyInProg	0
0063	137681	018F	XX	CopyInProg	0
0064	138090	0190	XX	CopyInProg	0
0065	138090	0191	XX	CopyInProg	0
0066	136981	0192	XX	CopyInProg	0
0067	138090	0193	XX	CopyInProg	0
0068	138090	0194	XX	CopyInProg	0
0069	137020	0195	XX	CopyInProg	0
006A	138090	0196	XX	CopyInProg	0
006B	138090	0197	XX	CopyInProg	0
006C	138090	0198	XX	CopyInProg	0
006D	138090	0199	XX	CopyInProg	0
006E	138090	019A	XX	CopyInProg	0

Symmetrix ID : 000184500160 (Microcode Version: 5568)

Source Device		Target Device	State	Copy
Sym	Protected Tracks	Sym	CG SRC <=> TGT	(%)
0014	136681	03B6	XX CopyInProg	1
0015	138090	03B7	XX CopyInProg	0
0016	138090	03B8	XX CopyInProg	0
0017	138090	03B9	XX CopyInProg	0
0018	138090	03BA	XX CopyInProg	0
0019	138090	03BB	XX CopyInProg	0
001A	138090	03BC	XX CopyInProg	0
001B	138090	03BD	XX CopyInProg	0
001C	138090	03BF	XX CopyInProg	0
001D	138090	03BE	XX CopyInProg	0
001E	138090	03C0	XX CopyInProg	0
001F	138090	03C1	XX CopyInProg	0
0020	138090	03C2	XX CopyInProg	0
0021	137328	03C3	XX CopyInProg	0
0022	138090	03C4	XX CopyInProg	0
0023	138090	03C5	XX CopyInProg	0
0024	138090	03C6	XX CopyInProg	0
0025	138090	03C7	XX CopyInProg	0
0026	138090	03C8	XX CopyInProg	0
0027	138090	03C9	XX CopyInProg	0

Total	-----
Track(s)	6203054
MB(s)	193845

Legend:

(C): The background copy setting is active for this pair.
 (G): The Target device is associated with a composite group.

The `symclone terminate` command attempts to end all clone sessions in the composite group but cannot do so because the clone pairs are in the CopyInProg state.

```
# symclone -cg TFClone terminate -noprompt
```

```
'Terminate' operation execution is in progress for composite group 'TFClone'.  
Please wait...
```

The session cannot be terminated because the device is 'Copy In Progress' or 'Restore In Progress' unless the Symmetrix force flag is used

The `symclone terminate` command with the `-symforce` option forces all clone sessions in the composite group to terminate.

```
# symclone -cg TFClone terminate -symforce -noprompt
```

```
'Terminate' operation execution is in progress for composite group 'TFClone'.  
Please wait...
```

```
'Terminate' operation successfully executed for composite group 'TFClone'.
```