



CATE Storage and Backup Engineering
Citi Architecture & Technology Engineering

V-MAX TimeFinder/Clone

GDSE-09038-SG

Solutions Guide

Version 1.4

VERSION CONTROL

Version and Date of Issue	Author	Comment
V1.0. 13/10/09	Chris Clayton	First draft for V-MAX Peer reviewed by Pat Fogarty. Operational review by Jesse Reyes
V1.1. 20/04/10	Chris Clayton	Minor update to mention support for thick to thin cloning. Added some use case information for filesystems and databases.
V1.2 17/1/11	Chris Clayton Pat Fogarty	Add best practice for using gkselect to avoid AIX locking disk by the SE daemons.
V1.3 31/10/12	Chris Clayton	Added symauth role requirements for arrays using symauth for roles management
V1.4 30/10/13	Carl P Williams Chris Clayton	Added guidance on symqos settings for R2 Clones and Application Clones. Added instruction for generation of non-hardware based unique ID for symacl. General Edits removing historical information no longer relevant

CONTENTS

Legal Notice	5
Contact	5
Latest Versions	5
1 DOCUMENT SCOPE	6
1.1 Target Audience	6
1.2 Out of scope	6
1.3 Document update history	7
1.3.1 V1.4	7
1.3.2 V1.3	7
1.3.3 V1.2	7
1.3.4 V1.1	7
1.3.5 V1.0	7
2 INTRODUCTION TO CLONES	8
2.1 Overview of TimeFinder Clones	8
2.2 Applicable environments	8
2.2.1 Using clones with databases and file systems	9
2.3 Clone terminology	10
3 SUMMARY OF RECOMMENDATIONS	11
3.1 Use of symmir command	11
4 PRE-REQUISITES FOR USING CLONES	12
4.1 Licensing	12
4.2 Solutions Enabler & Symmetrix microcode requirements	12
4.3 Gatekeepers	12
4.4 SYMACL	12
4.4.1 Permissions	12
4.4.2 Non-hardware based Unique ID's	12
4.5 Symauth roles	13
5 BEST PRACTICE FOR USING CLONES	14
5.1 Using device files for clone control	14
5.1.1 Background	14

5.1.2	Using device files	14
5.1.3	Device groups, device files and GNS	15
5.2	Clone creation	16
5.2.1	Note on changing modes	16
5.2.2	Pre-copy clone cycling	16
5.2.3	Other notes relating to clone creation	17
5.3	Clone activation	18
5.3.1	Consistent activation	18
5.4	Clone recreation	19
5.5	Cloning from SRDF/A R2 devices	20
5.5.1	Detailed note on completing a pre-copy cycle	20
5.5.2	Use of Clone QoS to ensure high-copy rate for application clones	20
5.6	Monitoring the state of clone sessions	22
5.6.1	Symclone query	22
5.6.2	Symclone list	23
5.7	Clone state verification	25
5.8	Restoring from a clone	26
5.9	Terminating clone sessions	26
5.10	Multi-session cloning	27
5.10.1	Device files	27
5.10.2	Monitoring multi-sessions clones	28
5.10.3	Other multi-session clone issues	28
6	GATEKEEPER BEST PRACTICE FOR AIX	29
6.1	Background	29
6.2	gkselect file workaround	29
6.3	Example	30
6.3.1	Test environment	30
6.3.2	Test	30
APPENDIX A – EQUIVALENT SYMCLONE AND SYMMIR OPERATIONS		32
APPENDIX B – MULTI-SESSION CLONE EXAMPLE		33

Legal Notice

Information contained herein is for Internal Use and may be used only for business purposes authorized by Citigroup.

CITI, CITIBANK, CITICORP and CITIGROUP are trademarks and service marks of Citicorp and are used and registered throughout the world. Citibank, Citicorp, Citigroup and their subsidiaries also claim rights in certain other trademarks and service marks contained in these pages.

Contact

Please address any comments or questions regarding the information contained in this document via email to the “*CATE Storage and Backup Engineering” distribution list or alternatively to dl.CATE.Storage.and.Backup.Engineering.

Latest Versions

It is recommended that the latest version of the document be used from the CSBE website

<https://catecollaboration.citigroup.net/domains/platstor/gdse/products/techstdshmpgs/symmetrix.aspx>

Newer versions may include expanded scope, or comments on issues discovered after deployment.

1 Document Scope

This Solutions Guide recommends Best Practice for using TimeFinder/Clone technology on EMC V-MAX within Citigroup. It is not a replacement for the Vendor documentation referenced below. The following EMC documentation should also be consulted when implementing clones.

Furthermore, this Solutions Guide must be read in conjunction with the GDSE document V-MAX TimeFinder/Clone Release Notes, which covers in more detail product certification, known issues and test results (both functional and performance).

Document	Location
EMC documentation	
EMC Solutions Enabler Symmetrix TimeFinder Family CLI Product Guide Version 7.X	EMC powerlink site
EMC Engineering White Paper: Using SYMCLI to Perform TimeFinder/Clone Operations	EMC powerlink site
EMC White paper: Using SYMCLI to Perform Consistent Splits with TimeFinder Products	EMC powerlink site
EMC Solutions Enabler Symmetrix Access Control CLI Product Guide	EMC powerlink site
GDSE documentation	
V-MAX TimeFinder Clone Release Notes	https://catecollaboration.citigroup.net/domains/platsto/r/gdse/products/platforms/Symmetrix/09038-RN.pdf

1.1 Target Audience

This detailed document is aimed at SAN Operations teams and Citi architecture technology engineering team familiar with EMC technology and tasked with providing clone functionality to meet specific needs. It is not intended as an overview for end users, developers or Project Managers.

1.2 Out of scope

- Detailed advice on using the `symmir` TimeFinder/Mirror user interface. With V-MAX, `symmir` actually uses clone technology under the covers. Use of `symmir` is still supported for existing scripts, but for any new requirements for full point-in-time copies, the `symclone` command described in this document should be used.

1.3 Document update history

1.3.1 V1.4

Added Section 5.5.2: Information and instruction on using symqos for point in time R2 clones and R2 Clones utilised by applications.

Added section 4.4.2: Information and instruction on using non hardware based unique ID's for SymACL configuration.

Removed some historical references

1.3.2 V1.3

Added symauth role requirements for arrays using symauth for roles management.

1.3.3 V1.2

New Section 6. Added best practice for using gkselect to avoid AIX locking disk by the SE daemons.

1.3.4 V1.1

Updated to mention support for thick to thin cloning (and vice versa) from SE7.1.1 / 5874.207 onwards.

Added more detailed use case information to Section 2.2 for databases and file systems.

1.3.5 V1.0

Initial version for V-MAX

2 Introduction to clones

2.1 Overview of TimeFinder Clones

TimeFinder/Clone software for Symmetrix allows one to clone copies of data simultaneously on one or more target devices from a single source device within the same Symmetrix. Data can be copied to as many as 16 target devices (depending on configuration).

TimeFinder/Clone functionality is controlled via copy sessions, which pair the source and target devices. Sessions are maintained on the Symmetrix array and can be queried to verify the current state of the device pairs. A copy session must first be created to define and set up the clone devices. The session is then activated, enabling the target devices to be accessed by its host. When the clone is no longer needed, the session can be terminated. TimeFinder/Clone operations are controlled from the host using the `symclone` commands to `create`, `activate` and `terminate` the copy sessions.

Clones can be created in one of three modes:

PreCopy – Copying of data to the target clone begins as soon as the clone session is created.

Copy – Copying of data to the target clone begins only when the clone is activated

CopyOnAccess – Copy of data to the target clone only occurs after activation and when either data is written to the source or reads or writes take place to the target device. This mode is not supported by GDSE.

TimeFinder/Mirror commands (`symmir`) can also be used to manipulate BCV-based clones. On V-MAX, `symmir` always uses clone technology under the covers. As a consequence, clones controlled by either `symmir` or `symclone` do not consume any mirror positions.

This document focuses on the “native” control of clones using the `symclone` command.

2.2 Applicable environments

Clones offer the ability to make multiple copies of production data within a Symmetrix storage array that:

- 1) Are fully independent of the source volumes once copied
- 2) Can be created within relatively short periods of time
- 3) Uses Standard (thick) or thin Symmetrix devices. The following combinations are supported:
 - a. Thick to thick cloning (standard → standard)
 - b. Thin to thin cloning
 - c. Thick to thin cloning (for thick to thin migrations rather than normal operations)
 - d. Thin to thick cloning (usually to reverse a thick to thin change)
- 4) Are on locally mirrored (protected) storage

- 5) Are (optionally) on SRDF replicated storage (although GDSE does not support cloning to SRDF replicated storage).
- 6) That can be differentially resynchronised with the source data
- 7) Can be used to differentially restore to the source data

These can be used for a number of purposes:

- 1) To create a copy of production data for parallel batch runs. The copy can be quickly re-synchronized with the master copy at the end of the run.
- 2) To create copies of production data for application test purposes.
- 3) To create multiple point-in-time copies of data at different points during a batch run or during on-line day for potential restore back to the source (limited functionality).
- 4) To create rollback point-in-time copies for application development testing.

The `symclone` command can be used to create point-in-time copies of data at the target site of an SRDF/A environment prior to resyncing the SRDF/A link to ensure a consistent copy of data at the DR site at all times. However, at the time of writing, it is recommended that the `symmir` command still be used on V-MAX for this specific purpose, to provide consistency with DMX processes and backwards compatibility with existing scripts used to manage SRDF/A point-in-time copies.

2.2.1 Using clones with databases and file systems

If you take a point-in-time copy of an active file system or database, you have a copy of the data that is equivalent to what you get when a server crashes with the file system or database active. Hence, on mounting the file system and/or database you need to go through fsck and/or DBMS recovery processes.

If you wish to avoid the fsck and/or DBMS recovery processes, you have several options:

- a) Cease I/O (i.e. unmount filesystem and/or suspend database) prior to activating the clone. This ensures that all data has been flushed down to disk and data set is consistent.
- b) Do not cease I/O or do not put database into backup mode, but activate the clone with the `-consistent` flag (you will need a zero cost license for this). This ensures that the database is consistent, but you still get a “server crash consistent” type image i.e. you still need to go through fsck or DMBS recovery processes. Use of the `-consistent` flag is covered in Section 5.3.1.
- c) Put databases into backup mode and use `-consistent` with the activate command. This will provide you with both a consistent copy of the data and avoids the DMBS recovery step for databases. However, this will not help with the filesystem fsck issues. Using a sync command is a possibility here (not tested by GDSE), but one also needs to close open files.

2.3 Clone terminology

The following terms relating to clone operations are used throughout the document. A table showing the equivalence between TimeFinder/Clone (symclone) and the older, but perhaps more familiar TimeFinder/Mirror (symmir) commands can be found in Appendix A.

Copy or Clone Session – A copy session pairs source and a target devices i.e. it defines the relationship between the devices. Sessions are maintained on the Symmetrix Array.

Create – Creates the relationship between a source device and a clone target device. Also starts copying of data for precopy mode clones. After creation, the target device is not accessible by its host. Similar to an establish operation with symmir for precopy mode clones.

Activate – Marks the point-in-time that the clone copy represents, and hence is similar to a split operation with symmir. Also starts the copying of data for copy mode clones. After activation, the target device is accessible to a host, and is no longer synchronized with its source.

Establish – Creates and activates a clone session with a single command. Only available with copy mode clones. Similar to an establish operation with symmir.

Recreate – Requests that only changed tracks since the last activate action are copied across to the target device. Equivalent to an incremental establish with symmir.

Restore – Copies data from the target to the source. Can either be a full or incremental restore. A restore operation can only be followed by a termination operation or a split operation.

Split – Used to *split* a clone device pair that is in the Restored state. This command will change the direction of the clone relationship (that is, the original source device will become the source device for a future copy), which will enable you to use either the establish or recreate command.

Terminate – Destroys the clone session relationship between source and target. Equivalent to a cancel operation with symmir.

3 Summary of recommendations

GDSE is certifying the use of the following TimeFinder two clone types:

- Precopy clones (recommended)
- Copy clones

The benefits and usage of each type of clone are covered later in this document.

Cloning from SRDF R1 and R2 devices is supported, but cloning from an SRDF/A R2 has certain qualifications (described later).

The following are not supported (see Release Notes for further details of issues):

- Cloning to SRDF R1 devices
- CopyOnAccess clones

3.1 Use of symmir command

The `symmir` command should only be used in SRDF/A environments for the management of R2 point-in-time copies prior to SRDF/A resyncs. The `symclone` command should be used for any other new point-in-time requirements, in particular for creating copies for application usage.

4 Pre-requisites for using clones

4.1 Licensing

Potential users of clones should first confirm with their local SAN management team that clones are supported on the Symmetrix storage array on which their application resides.

Use of clones requires a TimeFinder/Clone license for the Symmetrix in question. See the GDSE document "TimeFinder/Clone Release Notes" for further details on current clone licensing arrangements.

The license key (TF-Clone) must be registered into the license file `symapi_licenses.dat` using the `symlmf` command before clone operations can be executed.

4.2 Solutions Enabler & Symmetrix microcode requirements

The minimum certified level of Solutions Enabler required for clone operations with V-MAX is 7.2 (it was previously supported on version 7.1, but this is no longer supported by EMC). The `symclone` command is part of the Solutions Enabler package.

The minimum certified level of Symmetrix microcode required for V-MAX is 5874.

4.3 Gatekeepers

A host using TimeFinder/Clone should have a minimum of 6 gatekeepers allocated to it from the array in which the clone volumes to be controlled are located.

4.4 SYMACL

4.4.1 Permissions

For a host to issue `symclone` commands to specific volumes on a Symmetrix Access Control (`symacl`) enabled Symmetrix, it can be set up in one of two ways:

- as a member of the `symacl AdminGrp` on the Symmetrix to be used. This should only be used for SAN management servers as this provides complete control over the Symmetrix.
- as a member of a `symacl Access Group` which is allowed to act on a `symacl Device Pool` containing the clone source and target devices. The group must also have the BCV ACE granted to it. This should be used for application servers, as it provides fine-grain control over which servers can control which devices and with which commands.

4.4.2 Non-hardware based Unique ID's

Host `symacl` set up requires solutions enabler to be deployed on the host in question that will be issuing clone commands.

For the ACL set up on the array, the hosts unique ID is provided to the SAN administration team (collected by running `symacl -unique` on the host).

Typically the Unique ID is based on server hardware (normally MAC address of the first NIC).

In some virtualised host environments if a unique ID has been generated and applied based on the underlying physical hardware, and then the virtual host is logically moved/migrated to different underlying hardware, post move, the host will not be able to issue commands as the unique ID would be based on the original underlying hardware.

This issue has been encountered in production on an AIX LPAR's post Live Partition Mobility.

The fix at the time was to update the ACL with the new unique ID, but obviously only this was a temporary fix, as it would break again once the LPAR is migrated/rolled back to original frame.

A permanent fix is provided in solutions enabler whereby a randomly generated unique ID is used.

This was successfully tested with frame engineering, and can be applied as followed:

- Add the following entry to /var/symapi/config/options file.
SYMAPI_ALTERNATE_ACCESS_ID = ENABLE
- stop & start the storsrvd daemon
Example:
 - # stordaeomon shutdown storsrvd -immediate
 - # stordaeomon start storsrvd
- Generate Unique ID and provide to storage team
Example:
 - # symacl -unique
9FA72347-5384869B-3A646E78
- Storage team to update and commit Symacl based on new unique id

NOTE: The alternate access ID is stored on disk in the following files:

- /var/symapi/config/lockboxp
- /var/symapi/config/lockboxb

It is the responsibility of the host system administrator to backup these 2 files, because if these file are lost, the host will not be able to access the Symmetrix.

4.5 Symauth roles

If that array they you are using has symauth enabled (your SAN management team will know) then you will also need to ask the SAN management team to ensure that the username that you use to execute the symclone commands has the StorageAdmin role granted to it so that it can control clone copies.

5 Best Practice for Using Clones

5.1 Using device files for clone control

5.1.1 Background

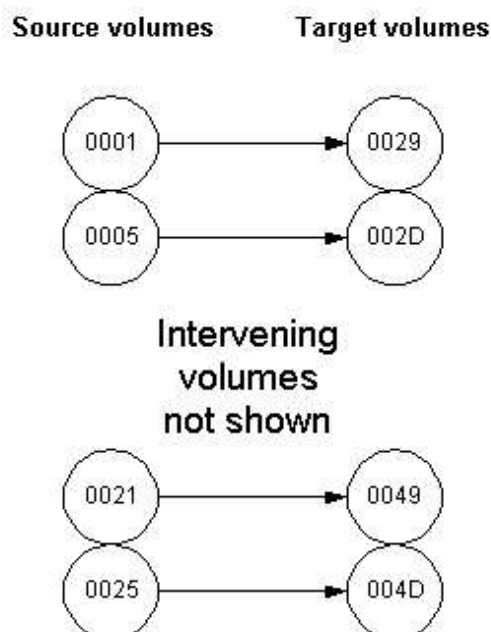
When Clones were first certified by Citi, control of clones using standard devices as targets could not be easily achieved using device groups. For example, it was not possible to issue symclone control commands using device groups without also defining each source-target relationship on each and every command line. For clone set ups with multiple source volumes, this would result in unmanageable command lines. Hence, it was recommended then that device files be used to control clone operations to overcome the restrictions imposed by device groups.

Today it is possible to overcome this issue by using feature known as a clone target list. The **EMC Solutions Enabler Symmetrix Array Management CLI Product Guide** contains information on creating and managing clone target lists.

Use of device files is still recommended (although certainly not mandatory) as device files simplify clone operation in SRDF environments. Also, the important symclone query and verify commands do not work with clone target lists (i.e. the `-tgt` qualifier). Hence, the examples in this document are all still based on using device files, rather than device groups and clone target lists.

5.1.2 Using device files

Single Session Clones



For example, to single-session clone ten devices 0001 through 0025 to devices 0029 through 004D, as shown above, one would need a device file of the form shown below. The left hand

column shows the source volumes and the entries on the right are the corresponding target devices.

```
# cat clone_pairfile
0001 0029
0005 002D
0009 0031
000D 0035
0011 0039
0015 003D
0019 0041
001D 0045
0021 0049
0025 004D
```

This file can be used to manage all clone operations relating to the set of relationships in the above diagram with relatively simple command lines, and is used for many of the examples in this document.

5.1.3 Device groups, device files and GNS

There is one issue with using device files rather than device groups.

When device groups are used, GNS can be used to ensure that different servers have the same view of the device group definition, as GNS places this definition on the Symmetrix rather than store it on local hosts. Hence, when GNS is running, any changes made to the device group definitions are automatically seen by all hosts using GNS with the relevant Symmetrix.

When device files are used, any changes made to the device files need to be copied to all hosts that will need to use those device files. This will usually be a manual process.

For SAN management servers where:

1. Changes are regularly being made to device group definitions as there are many device groups defined on these servers as they are used to manage devices on behalf of other systems.
2. There are usually multiple SAN management servers (for redundancy)

then copying device file definitions could be a significant overhead.

However, clones are being certified principally for application server usage. In this case:

1. There will only be device group definitions for that server's devices and changes will only be made infrequently
2. Application servers are not usually configured with multiple units

Hence, it is considered that for application servers, the overhead of managing the device files will be very similar to that of managing a device group.

5.2 Clone creation

Clones should usually be created using the `–precopy` and `–differential` options.

`–precopy` ensures that the copying of data begins PRIOR to the activation of the clone and hence offers the potential to take point-in-time copies of data without impacting performance of the source or target at the time of activation.

`–differential` maintains changed track information and permits differential resynchronisation and restores. Note that use of `–differential` limits the maximum number of copy sessions for a source device to eight. However, this is a limit that is not likely to impact many requirements. Note that differential status cannot be set after creation. `–differential` can only be used with `–copy` or `–precopy`.

If you are planning never to use the clone for restore purposes, nor to resync it with the source, but are simply using cloning to make a copy for test purposes, then the differential flag is not required. The only benefit of not specifying `–differential` is that it extends the number of copy sessions that a source volume can have from eight to sixteen.

A differential precopy clone session for multiple volumes can be created with the command

```
# symclone -f ./clone_pairfile -sid <SID> create –precopy –differential
```

When you issue the `symclone create` command, the target device is made not ready (NR) to the host. When you issue the `symclone activate` command, the target is made ready again and accessible to its host. Note that when the target is NR, it cannot be mounted by a host.

If for some reason it is desirable to create the clone but not start copying immediately, the `–copy` option can be used. Again `–differential` should also be specified. Copying can then be initiated either by changing the mode to precopy or by activating the clone (which also marks the point-in-time of the clone).

5.2.1 Note on changing modes

The `set mode` argument allows the changing of the clone status from one mode to another. For example, a clone session created with the default `nocopy` option can be changed to either the `copy` or `nocopy` state. The `set mode` argument can only be used on a clone pair in state `Created` or `Copied`.

Note, however, that it is not possible to create a `nocopy` clone with the option `–differential`. Also it is not possible to add the `–differential` after creation time. Hence, if differential clones are required, they must be created with `–copy` or `–precopy`, rather than having the mode changed after creation by using `set mode`.

It should be stressed that changing the mode to `precopy` causes the background data copy to start before activation, but that the point in time copies are created as the moment of activation. When mode is changed to `copy`, the background copy from source to target is started only after activation.

5.2.2 Precopy clone cycling

Precopying can occur in the created or recreated state. While in this state, the precopy never completes. Instead, the process cycles round checking 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 precopy operation ends.

5.2.3 Other notes relating to clone creation

The source and the target device must be of the same size and emulation type (usually Fixed Block Addressing (FBA) on Open Systems Symmetrix). Source and target meta devices must be identical in stripe count, stripe size and capacity.

5.3 Clone activation

Clone activation marks the point in time that the clone copy represents. It is stressed again here that cloned data is made available as a Point-in-time copy at the time of activation and not at the time that the session was created. Activation also places the target device into RW state. For copy mode clones, clone activation also initiates the background copy from source to target.

An activated precopy clone becomes a copy clone and enters CopyInProgress state until copying is complete (Copied state).

A source volume may only have a maximum of 4 sessions that are in CopyInProgress state at any one time. Attempts to activate more copies will result in the error

```
"The maximum number of Concurrent Copies has been exceeded"
```

The above error is issued only when an attempt is made to put more than 4 copy sessions for a source device into CopyInProg state at the same time. Clone sessions created with the default nocopy option (CopyOnAccess clones) are not subject to this limitation, because activating them does not start a background copy. However, CopyOnAccess clones are not recommended for the reasons given elsewhere in this document.

It is preferable to allow the copying to complete before accessing target devices. Access would impact IO response on both the source and target, as reads or writes to the target device causes the data to be copied immediately from the source (if not already copied).

5.3.1 Consistent activation

If clones are being made of multiple volumes that are part of the same data set, the activation should be done in a consistent manner.

One option is to use the `-consistent` option when activating a clone session to use Enginuity Consistency Assist (ECA) to make copies that are restartable for dependent-write applications. The Symmetrix API pauses I/O during activation to ensure consistency across all the volumes. Note however that this option requires a consistency license on your Symmetrix which may not be available as this is a significant additional cost.

```
# symclone -f clone_pairfile -sid <SID> activate -consistent
```

Alternatively, one can quiesce the application to ensure that a consistent copy is taken without using the `-consistent` flag, and hence without the need for a consistency license on your Symmetrix.

5.4 Clone recreation

To capture subsequent writes made to the source during this clone pair session, symclone recreate can be used to copy from the source to the target just those tracks that have changed since the last copy completed. Note that to permit this incremental update, the clone session must have been created with either the `-copy` or `-precopy` option, and with the `-differential` option. Clone recreation is similar to a TimeFinder/Mirror incremental establish (see Appendix A).

Note that during a recreate, the host must not have the target devices in use i.e. not mounted or imported if in a volume manager disk group.

Recreate defaults to `-copy`. Recreate can also take the `-precopy` option if you wish to start copying tracks in the background, before the clone session is activated. You must specify `-precopy` even if the original create defined `precopy`. While in the Recreated state, the `precopy` never actually completes. Instead, the process keeps checking for new writes to be the `pre-copied` to the target device until the session is activated. Once activated, the normal background copy mechanism takes over and the `pre-copy` operation ceases.

```
# symclone -f clone_pairfile -sid <SID> recreate -precopy
```

This command differentially resynchronises the target with the source.

The new point in time that the clone copy represents can then be marked with another `activate` command:

```
# symclone -f clone_pairfile -sid <SID> activate [-consistent]
```

The `activate` command also makes the target accessible to the host again i.e. the device can be imported and mounted.

The above two steps can be combined into a single command that in effect just captures the clone new writes since the last activation of the clone. This is referred to as an “establish” and should not be confused with a BCV establish. Clone establish (with `symclone`) results in refreshing of an independent volume whereas a BCV establish (with `symmir`) updates the content of the BCV but leaves the BCV mirroring any further changes to the Standard device to which it is established. A table showing the equivalence between TimeFinder/Clone (`symclone`) and TimeFinder/Mirror (`symmir`) commands can be found in Appendix A.

By default, an `establish` command implies the `-copy` and `-diff` options. `-precopy` is not a valid option as an `establish` also activates the clone, in effect changing it to a copy mode clone.

```
# symclone -f clone_pairfile -sid <SID> establish [-consistent]
```

5.5 Cloning from SRDF/A R2 devices

The following is EMC's strongly recommended Best Practice when cloning from an SRDF/A R2 device:

- 1) TimeFinder/Clone from an active SRDF/A R2 device is supported, provided you include the -precopy option in the command.
- 2) With an SRDF/A R2 source device the precopy should be allowed to complete one full pass around the device (i.e. complete on pre-copy cycle) prior to issuing the activate command. Failure to do so can have a performance impact on SRDF/A replication. See subsection below for further details of the issue and how to determine in a precopy pass has completed.
- 3) Use CLONE Quality of Service (QoS) to prioritize the copying to application clones over SRDF/A BCVs/clones. See subsection below for further details.

5.5.1 Detailed note on completing a pre-copy cycle

When the precopy completes the first pass an indication bit is set. This is displayed by an 'X' in the 'P' column of the symclone query output. The reason that this is necessary is that writes to the clone source device are not destaged to the target as they arrive because the clone is not an actual mirror of the Standard like an established BCV is. The precopy scan finds newly written tracks and copies them to the clone target as they are found. When the precopy is complete, the device will be "synchronized" so to speak and SRDF/A will not be affected when the clone is activated.

Confirmation that precopy has completed one cycle can be obtained from the following command:

```
# symclone -f ./clone_pairsfile -sid <SID> verify -precopy -cycled
```

Once one precopy cycle has completed, it is safe to activate the R2 clone.

```
# symclone -f ./clone_pairsfile -sid <SID> activate -nop [-consistent]
```

5.5.2 Use of Clone QoS to ensure high-copy rate for application clones

All SRDF/A R2 devices have a "BCV" associated with it to enable a point-in-time copy to be made during SRDF/A resynchronization. This ensures that a consistent copy of the data is available at the DR site at all times. On VMAX (unlike on DMX) BCVs are actually implemented as clones. Hence, there can be competition between the many established SRDF/A point-in-time BCVs (clones) and application clones, resulting in a much slower default copy rate for application clones from SRDF/A R2 source devices than from SRDF/A R1 devices.

This can be resolved by using CLONE QoS to prioritise the copy rate from application clones. The following tasks will need to be executed by the SAN management team on the R2 array.

- a) **For SRDF/A resynchronization point-in-time copy BCVs/Clones**, the CLONE QoS priority should be set to "4" for all the SRDF/A BCV/clone target devices in the array. This should be all devices of type BCV-TDEV For example:

```
# symqos -devs SymDevStart:SymDevEnd set clone priority 4
```

where the devices specific by `SymDevStart:SymDevEnd` are the SRDF/A R2 BCV (clone target) devices. As this needs to be applied to all SRDF/A R2 BCV (clone target) devices in the array, the above command may need to be run multiple times depending on whether your SRDF/A R2 BCV (clone target) devices symdev numbers are in a contiguous range or not.

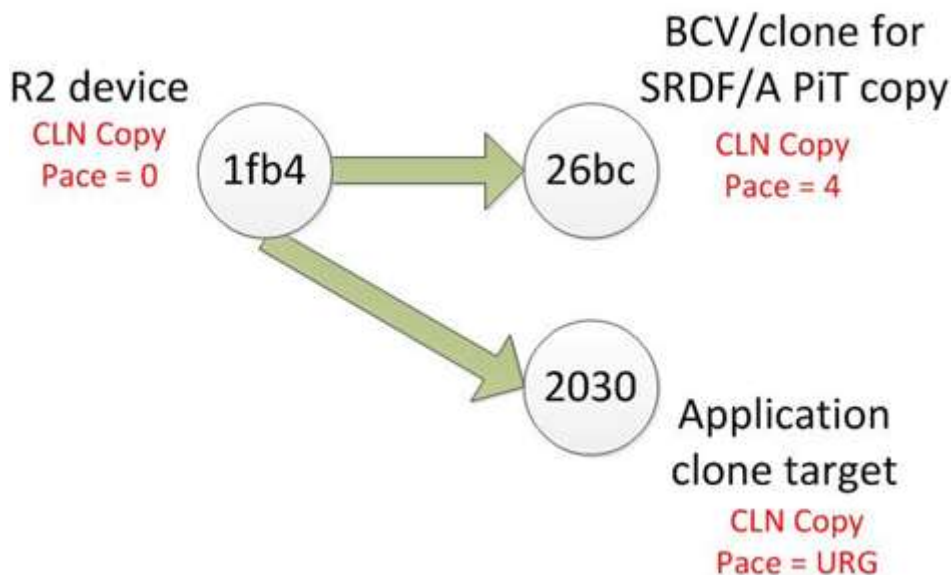
- b) **For application clones from SRDF/A R2 devices**, the CLONE QoS priority should be set to "URGENT" for the clone target devices

```
# symqos -devs SymDevStart:SymDevEnd set clone priority URGENT
```

where the devices specific by `SymDevStart:SymDevEnd` are the application clone target devices. These will be devices of type TDEV, but note that not TDEV devices in the array will necessarily be clone targets, so do not set the clone priority for ALL devices of type TDEV, just the clone targets for applications.

- c) **For the SRDF/R2 devices themselves** (the source devices in all the clone operations) the CLONE QoS should be left at the default value of 0. These will be devices of type RDF2+TDEV.

The diagram below gives an example of a single R2 device and the two clone relationships associated with it.



The commands `symqos list` or `symqos query` (different output formats) can be used to confirm that the priority is set correctly (which appears as "copy pace" in the output of the above commands for historical reasons). The simple `symqos list` examples below (just for one device of each type to limit the size of the output) shows that:

- The CLONE priority (CLN Copy Pace) of the SRDF/A R2 BCV/clone 26bc has been changed to 4
- The CLONE priority (CLN Copy Pace) of the (clone target) device 2030 has been changed to URGENT (URG).
- The CLONE priority (CLN Copy Pace) of the SRDF/A R2 (clone pair source) device 1fb4 has been left at the default value of 0.

These commands can be used to confirm that you have correctly set QoS values.

```
$ symqos -sid 2231 list -dev 26bc
Symmetrix ID      : 000292602231
```

Device Name			Copy Pace					
Sym	Physical	Config	BCV	RDF	MIR	CLN	VLN	
26BC Not Visible		BCV+TDEV	0	0	0	4	0	

```
$ symqos -sid 2231 list -dev 2030
Symmetrix ID      : 000292602231
```

Device Name			Copy Pace					
Sym	Physical	Config	BCV	RDF	MIR	CLN	VLN	
2030 Not Visible		TDEV	0	0	0	URG	0	

```
$ symqos -sid 2231 list -dev 1fb4
Symmetrix ID      : 000292602231
```

Device Name			Copy Pace					
Sym	Physical	Config	BCV	RDF	MIR	CLN	VLN	
1FB4 Not Visible		RDF2+TDEV	0	0	0	0	0	

5.6 Monitoring the state of clone sessions

5.6.1 Symclone query

Symclone query may be used to determine:

- The number of clone sessions
- % of copy completed
- State of each clone pair
- Clone mode

The status of clone sessions can be checked with the following command:

```
# symclone -f clone_pairfile -sid <SID> query -multi
```

The `-multi` flag forces the command to show all targets that are paired with the source. It is not strictly needed in a single-session environment. However, it is good practice to always use it as failure to do so in a multi-session environment will result in inaccurate display of the clone status.

The following is example output from a clone command.

```
# symclone -f clone_pairfile -sid 1380 query -multi
```

Device File Name : ./clone_pairfile
 Device's Symmetrix ID : 000192601380

Source Device				Target Device				State		Copy
Logical	Sym	Protected Tracks	Modified Tracks	Logical	Sym	Modified tracks	CGDP	SRC <=>	TGT	(%)
N/A	0001	662708	0	N/A	0029	0	X.X.	PreCopy		39
N/A	0005	657284	0	N/A	002D	0	X.X.	PreCopy		40
N/A	0009	659658	0	N/A	0031	0	X.X.	PreCopy		39
N/A	000D	661988	0	N/A	0035	0	X.X.	PreCopy		39
N/A	0011	703365	0	N/A	0039	0	X.X.	PreCopy		35
N/A	0015	662535	0	N/A	003D	0	X.X.	PreCopy		39
N/A	0019	658363	0	N/A	0041	0	X.X.	PreCopy		40
N/A	001D	727331	0	N/A	0045	0	X.X.	PreCopy		33
N/A	0021	662518	0	N/A	0049	0	X.X.	PreCopy		39
N/A	0025	628355	0	N/A	004D	0	X.X.	PreCopy		42
Total		-----								
Track(s)		6684105								
MB(s)		208878								

Legend:

- (C): X = The background copy setting is active for this pair.
 . = The background copy setting is not active for this pair.
 (G): X = The Target device is associated with this group.
 . = The Target device is not associated with this group.
 (D): X = The Clone session is a differential copy session.
 . = The Clone session is not a differential copy session.
 (P): X = The precopy operation has completed one cycle
 . = The precopy operation has not completed one cycle

The output shows that there are 10 single-session (i.e. single source to single-target) clone sessions, all in precopy state (i.e. precopy clones that are not yet activated). The X entries in the CGDP columns indicate that these are differential Copy/Precopy clones. The Copy column shows that approximately 40% of the source data has been copied to the targets.

Note that one cannot determine the point-in-time that an activated clone represents from any query command. If this information is needed, it is necessary to record it separately when activation takes place.

Note that symclone query does not show clone sessions set up under Clone Emulation. These must be managed using the symmir query command as with native BCVs.

5.6.2 Symclone list

The symclone list command is very useful for getting an overview of all clone sessions on an entire Symmetrix unit and does not require a device file. However, it is limited in that it does not show multiple relationships for a source volume clearly (there is no –multi flag).

In practice it is used for checking that all clone sessions have been removed from a Symmetrix.

```
# symclone list –sid <SID>
```

Note that symclone list does not show emulated clones that have been created using TimeFinder/Mirror under Clone Emulation.

An example of symclone list output is shown below:

```
# symclone list -sid 1380
```

Symmetrix ID: 000192601380

Source Device		Target Device		Status
Sym	Protected Tracks	Sym	CGDP	SRC <=> TGT
0001	0	0029	XXX.	Copied
0005	0	002D	XXX.	Copied
0001	1080121	0051	XXX.	CopyInProg
0005	1084299	0055	XXX.	CopyInProg
0001	1097749	0079	XXX.	PreCopy
0005	1098869	007D	XXX.	PreCopy
Total				
Tracks	4361038			
MB(s)	136282			

Legend:

(C): X = The background copy setting is active for this pair.
 . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with a group.
 . = The Target device is not associated with a group.
(D): X = The Clone session is a differential copy session.
 . = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
 . = The precopy operation has not completed one cycle

5.7 Clone state verification

The status of a clone session can be determined using the verify option. This is essential for automating clone operations and the command can be set not to exit until the condition being tested against is met.

For example

```
# symclone -f clone_pairfile -sid <SID> verify -copied -i 10
```

will check the status of all the clone sessions defined in file clone_pairing every 10 seconds and only exit when ALL pairs are in copied state.

Verify can be used with the following self-explanatory options:

- created
- copied
- copyinprog
- copyonaccess
- precopy
- cycled
- recreated
- restored
- restinprog
- split

The -cycled option verifies that a copy session has completed one precopy cycle and should be used in conjunction with the -precopy flag. Using the -precopy flag alone simply tells you if it is in precopy state and not if it has been through the first precopy cycle. For example:

```
# symclone -f clone_pairfile -sid <SID> verify -precopy -cycled -i 10
```

This would be used to check that a precopy clone is close to fully copied before it is activated. ***This is essential before taking activating a clone from an SRDF/A R2 clone source device.***

5.8 Restoring from a clone

The symclone restore command can be used to copy target data back to the original source device (incremental restore). It is also possible to copy data from a target to another device.

To support incremental restore, the session must have been created with the `--differential` option and the device must be in fully copied state.

The syntax for a restore is

```
# symclone -f clone_pairfile --sid <SID> restore
```

After a restore, a split command should be issued. This command will change the direction of the clone relationship (that is, the original source device will become the source device for a future copy), which will enable you to use either the establish or recreate command.

5.9 Terminating clone sessions

Clone sessions should be terminated when no longer needed. Termination permanently destroys the relationship between the source and the target devices.

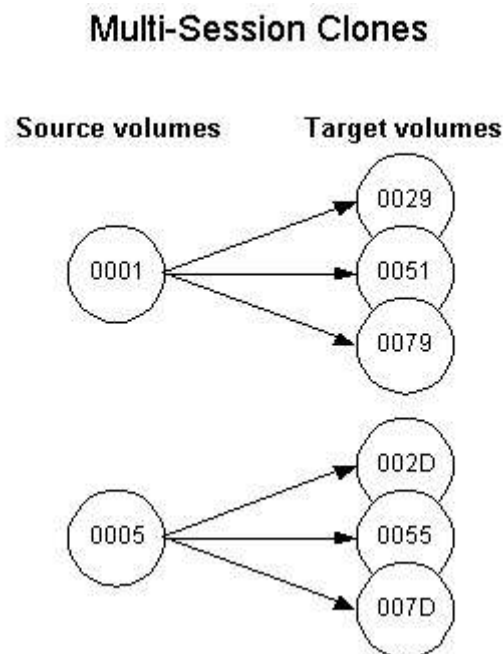
```
# symclone -f clone_pairfile --sid <SID> terminate
```

Note that previous DMX testing has found that clone termination has a large negative impact on source IO response, in particular if data is still being copied from the source to the target. Hence, termination should be scheduled at a time when IO response is not critical.

Pre-copy sessions that have not yet been activated can be terminated before they have completed copying. Sessions in state "CopyInProgress" (e.g. pre-copy sessions that have been activated) can only be terminated with the use of `--symforce`, as this will leave the target in an incomplete state.

5.10 Multi-session cloning

Multi-session cloning occurs when there is more than one active clone session from a source device. In the example below, source volumes 0001 and 0005 each has three active clone sessions. Each of the three relationships is managed separately.



5.10.1 Device files

The clone relationships in the diagram above can be controlled with the following device files. It is assumed in this example that the source volumes 0001 and 0005 contain parts of the same dataset and hence cloning operations need to be executed on both as a single entity.

Each device file defines one clone relationship for each source device. Hence, an operation involving one of these device files will involve one clone copy of each source device.

```
# cat clone_pairfile01
0001 0029
0005 002D
```

```
# cat clone_pairfile02
0001 0051
0005 0055
```

```
# cat clone_pairfile03
0001 0079
0005 007D
```

Examples of manipulation of multi-session clones in the above configuration using these device files can be found in Appendix B.

5.10.2 Monitoring multi-sessions clones

The `-multi` option must be used with a symclone query operation. Failure to do so will result in the query only showing the most recently created clone session.

Symclone list shows multiple clone sessions for a source, but does not group them together by source volume. There is no `-multi` option for symclone list.

An example of output from symclone list in a multi-session environment is shown on the next page.

```
# symclone list -sid 1380
```

```
Symmetrix ID: 000192601380
```

Source Device		Target Device		Status
Sym	Protected Tracks	Sym	CGDP	SRC <=> TGT
0001	0	0029	XXX.	Copied
0005	0	002D	XXX.	Copied
0001	1080121	0051	XXX.	CopyInProg
0005	1084299	0055	XXX.	CopyInProg
0001	1097749	0079	XXX.	PreCopy
0005	1098869	007D	XXX.	PreCopy
Total				
Tracks	4361038			
MB(s)	136282			

Legend:

(C): X = The background copy setting is active for this pair.
 . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with a group.
 . = The Target device is not associated with a group.
(D): X = The Clone session is a differential copy session.
 . = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
 . = The precopy operation has not completed one cycle

5.10.3 Other multi-session clone issues

The following points are specific to multi-session clones:

- The maximum number of active concurrent differential clone sessions per source volume is eight.
- The maximum number of concurrent copy sessions i.e. data actually being copied simultaneously from a source device is four. However, testing has shown that simultaneous copies very significantly slow down copy rates and also impact IO response time. Hence, simultaneous full copies should be avoided.

6 Gatekeeper Best Practice for AIX

6.1 Background

It has been noticed that when symcli is run on an AIX host with powerpath that devices chosen as gatekeepers are locked when the stordaemon storapid, storrdfd and storwatchd run. This could cause a serious problem for a host with no gatekeepers because a data disk will then be picked up by the daemon to act as a gatekeeper and because of this it will be locked. This only happens when powerpath is operational and the power device is chosen as the gatekeeper.

A permanent fix will be available in AIX PowerPath 5.5.

A suggested workaround for earlier versions of PowerPath to avoid the lock is to generate a gkselect file which contains the device names of the gatekeepers that should be used by the system but not addressing these devices using their power device names. Therefore there should not be any power devices listed in the gkselect file. Then any devices that are listed in the gkselect file will be usable as gatekeepers and because their hdisk names are used they will not be locked by the stordaemons.

This workaround has been tested and it has been confirmed that generating a gkselect file in /var/symapi /config does provide a fix to the problem. It also means that none of the gatekeepers are locked except while they are being used for appropriate commands by the symcli software.

For further information on gkselect files see the EMC documentation:

6.2 gkselect file workaround

This issue will be addressed in PP AIX 5.5 which is provisionally expected to go on general release 04/04/11.

The work around consists of putting hdisk names into a gkselect file:

"Create a gkslect file in the var/symapi/config directory. This works exactly opposite as the gkavoid file. Insert only the path of the devices you want Solutions Enabler to select as GK's. Put in only the rhdisk device of the GK. DO NOT include any rhdiskpower devices. If you do not have any GK's then you may want to select a device that is not in use by the OS or any applications.

Here is an example of the gkslect file,

```
# This device path will be selected by SE when selecting a GK
/dev/rhdisk1
```

After adding a gkavoid, or a gkselect, it will be necessary to stop / start the storapid daemon.

```
stordaemon shutdown storapid -immediate.
```

Once started please check to make sure the file is in effect by issuing the command

```
stordaemon action storapid -cmd list_gks xxxx | grep 0
```

make sure SE has not selected a power device.

An alternative would be to avoid using hdiskpower devices by doing the following:

Create a gkavoid file in the var/symapi/config directory, and insert the path for all the rhdiskpower devices. Anyone of these may be used as a GK so you want to make sure SE avoids using it. This file would contain the path to the Rhdiskpower device that would be avoided by Solutions Enabler. Remember, if you have no GK's Solutions Enabler will select a data device. Here is an example of the gkavoid file:

```
# This device path will be ignored by SE when selecting a GK.  
/dev/rhdiskpower1
```

6.3 Example

The following example is for the test that was run to confirm the workaround.

6.3.1 Test environment

An AIX Lpar was used running	AIX 6.1.0.0
Powerpath	Version 5.3 SP 1 (build 84)
Solutions Enabler	Version V7.1.1.18 (Edit Level: 1026)

6.3.2 Test

A test was run to see if the gkselect file did stop the gatekeeper devices from being locked. Firstly a gkselect file containing a hdiskpower device was set up in /var/symapi/config. This was tested to show the error that occurs when this is done .

This also illustrates the original problem with power devices being locked by the storage daemon when it “grabs” a power device. On a system where there are no well defined gatekeepers it is possible for the daemons to “grab” data disks.

Therefore a gkselect file was written containing a power device:

```
/dev/rhdiskpower2
```

With the rhdiskpower in the gkselect file it is locked when a storage daemon is started issuing the command

```
# stordaeomon start storapid
```

And then

```
# dd if=/dev/rhdiskpower2 of=/dev/null count=1
```

Gives output

```
dd: /dev/rhdiskpower2: Permission denied
```

Changing the content of the gkselect file so that there is no hdiskpower improves matters since the power device is not locked.

Contents of the gkselect file – (no hdiskpower appears in the file):

```
/dev/rhdisk37  
/dev/rhdisk17
```

On starting the storapid a gatekeeper is chosen from the list in the gkselect file and one is opened:

```
# stordaeomon action storapid -cmd list_gks 314  
.
```

.
.

Symmetrix ID: 000294900314
GK Select : Enabled

Priority	State	Flags	PdevName
-----	-----	-----	-----
0	CL		/dev/rhdiskpower14
1	CL		/dev/rhdiskpower15
2	CL		/dev/rhdiskpower16
3	CL		/dev/rhdiskpower17
4	CL		/dev/rhdiskpower18
5	OP	S	/dev/rhdisk17
6	CL		/dev/rhdisk18
7	CL		/dev/rhdisk19
8	CL		/dev/rhdisk20
9	CL		/dev/rhdisk21
10	CL	S	/dev/rhdisk37

.
.
.

Doing a dd we can actually access the open gatekeeper unlike when the hdiskpower is used and open.

```
# dd if=/dev/rhdisk17 of=/dev/null count=1  
1+0 records in  
1+0 records out
```

Appendix A – Equivalent symclone and symmir operations

The following table compares the equivalent operations for symclone and symmir.

“Clones” symclone command	“BCVs” symmir equivalent operation
symclone create (default is to create a Copy On Access Clone) <i>Not recommended; use -precopy</i>	No direct equivalent
symclone create –precopy	symmir establish –full
symclone create –copy	No direct equivalent
symclone activate	symmir split
symclone establish –full	Symmir establish -full followed by symmir split
symclone establish	symmir establish followed by symmir split
symclone recreate (default is to recreate a copy mode clone)	No direct equivalent
symclone recreate -precopy	symmir establish (Incremental if differential clone)
symclone restore	symmir restore (Incremental if differential clone)
symclone restore -full	symmir restore -full

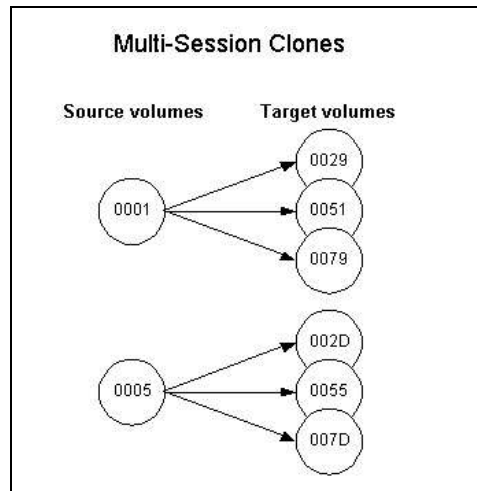
Appendix B – Multi-session clone example

The following is an example of the creation and usage of multi-session clones. The pairfiles used and the resulting source-target configuration is shown below.

```
# cat clone_pairfile01
0001 0029
0005 002D
```

```
# cat clone_pairfile02
0001 0051
0005 0055
```

```
# cat clone_pairfile03
0001 0079
0005 007D
```



Create the first clone for each source volume.

```
# symclone -file ./clone_pairfile01 create -precopy -differential -nop -sid 1380
```

```
'Create' operation execution is in progress for the device list in
device file './clone_pairfile01'. Please wait...
```

```
'Create' operation successfully executed for the device list
in device file './clone_pairfile01'.
```

Activate the clones consistently across both source volumes.

```
# symclone -file ./clone_pairfile01 activate -nop -sid 1380 -consistent -force
```

```
'Activate' operation execution is in progress for the device list in
device file './clone_pairfile01'. Please wait...
```

```
'Activate' operation successfully executed for the device list
in device file './clone_pairfile01'.
```

Confirm the status of the clone sessions. Note each source has a single clone. The state is CopyInProg showing that these are activated clones. A non-activated precopy clone would be in precopy state. A non-activated copy clone would be in created state. The D flag also confirms that these are differential clones.

```
# symclone -file ./clone_pairfile01 -sid 1380 query -multi
```

```
Device File Name      : ./clone_pairfile01
Device's Symmetrix ID : 000192601380
```

Source Device				Target Device				State	Copy
Protected Modified				Modified					
Logical	Sym	Tracks	Tracks	Logical	Sym	tracks	CGDP	SRC <=> TGT	(%)
N/A	0001	1096915	0	N/A	0029	0	X.X.	CopyInProg	0
N/A	0005	1097001	0	N/A	002D	0	X.X.	CopyInProg	0

```
Total
Track(s)      2193916
MB(s)         68559.9
```

Legend:

```
(C): X = The background copy setting is active for this pair.
      . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
      . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
      . = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
      . = The precopy operation has not completed one cycle
```

Wait until the clone is fully copied using the verify command.

```
# symclone verify -file ./clone_pairfile01 -copied -i 30 -sid 1380
```

None of the devices in the list are in the 'Copied' state.

None of the devices in the list are in the 'Copied' state.

[Intermediate output deleted]

None of the devices in the list are in the 'Copied' state.

None of the devices in the list are in the 'Copied' state.

Not all devices in the list are in the 'Copied' state.

All devices in the list are in the 'Copied' state.

Create a second differential precopy clone for each source device.

```
# symclone -file ./clone_pairfile02 create -precopy -differential -nop -sid 1380
```

'Create' operation execution is in progress for the device list in device file './clone_pairfile02'. Please wait...

'Create' operation successfully executed for the device list in device file './clone_pairfile02'.

The query command now shows that each standard device now has two clones; the first set that was created which is now fully (100%) copied, and a second set which is in precopy mode and which has only just begun to copy across.

```
# symclone -file ./clone_pairfile01 -sid 1380 query -multi
```

```
Device File Name      : ./clone_pairfile01
Device's Symmetrix ID : 000192601380
```

Source Device				Target Device				State		Copy
Logical	Sym	Protected Tracks	Modified Tracks	Logical	Sym	Modified tracks	CGDP	SRC <=> TGT	(%)	
N/A	0001	1086058	0	N/A	0051	0	X.X.	PreCopy	1	
		0	0	N/A	0029	0	X.X.	Copied	100	
N/A	0005	1090473	0	N/A	0055	0	X.X.	PreCopy	0	
		0	0	N/A	002D	0	X.X.	Copied	100	
Total		-----								

Track(s) 2176531
MB(s) 68016.6

Legend:

(C): X = The background copy setting is active for this pair.
. = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
. = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
. = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
. = The precopy operation has not completed one cycle

Create the third clone pair.

```
# symclone -file ./clone_pairfile03 create -precopy -differential -nop -sid 1380
```

'Create' operation execution is in progress for the device list in device file './clone_pairfile03'. Please wait...

'Create' operation successfully executed for the device list in device file './clone_pairfile03'.

Query command now shows that each device has three clones; one fully copied and two in precopy mode i.e. both copying to the target but not yet activated.

```
# symclone -file ./clone_pairfile01 -sid 1380 query -multi
```

Device File Name : ./clone_pairfile01
Device's Symmetrix ID : 000192601380

Source Device				Target Device				State		Copy
Logical	Sym	Protected Tracks	Modified Tracks	Logical	Sym	Modified tracks	CGDP	SRC <=>	TGT	(%)
N/A	0001	1100903	0	N/A	0079	0	X.X.	PreCopy		0
		1080121	0	N/A	0051	0	X.X.	PreCopy		2
		0	0	N/A	0029	0	X.X.	Copied		100
N/A	0005	1101932	0	N/A	007D	0	X.X.	PreCopy		0
		1084366	0	N/A	0055	0	X.X.	PreCopy		1
		0	0	N/A	002D	0	X.X.	Copied		100
Total										
Track(s)		4367322								
MB(s)		136479								

Legend:

(C): X = The background copy setting is active for this pair.
. = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
. = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
. = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
. = The precopy operation has not completed one cycle

Activate the second clone set, again in a consistent manner.

```
# symclone -file ./clone_pairfile02 activate -consistent -nop -sid 1380
```

'Activate' operation execution is in progress for the device list in device file './clone_pairfile02'. Please wait...

'Activate' operation successfully executed for the device list in device file './clone_pairfile02'.

Query now shows the clone copies associated with device file clone_pairfile02 are now in CopyInProg state.

```
# symclone -file ./clone_pairfile01 -sid 1380 query -multi
```

Device File Name : ./clone_pairfile01
Device's Symmetrix ID : 000192601380

Source Device				Target Device				State		Copy
-----				-----				-----		-----
Logical	Sym	Protected Tracks	Modified Tracks	Logical	Sym	Modified tracks	CGDP	SRC <=>	TGT	(%)
-----				-----				-----		-----
N/A	0001	1080121	0	N/A	0051	0	X.X.	CopyInProg		2
		1097916	0	N/A	0079	0	X.X.	PreCopy		0
		0	0	N/A	0029	0	X.X.	Copied		100
N/A	0005	1084299	0	N/A	0055	0	X.X.	CopyInProg		1
		1099053	0	N/A	007D	0	X.X.	PreCopy		0
		0	0	N/A	002D	0	X.X.	Copied		100
Total				-----						
Track(s)		4361389								
MB(s)		136293								

Legend:

(C): X = The background copy setting is active for this pair.
. = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
. = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
. = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
. = The precopy operation has not completed one cycle

Note that a symclone list does show all the clones sessions, but in order of creation rather than ordered by source device.

```
# symclone list -sid 1380
```

Symmetrix ID: 000192601380

Source Device			Target Device		Status	
-----			-----		-----	
Sym	Protected Tracks	Sym	CGDP	SRC <=>	TGT	
-----			-----	-----	-----	-----
0001	0	0029	XXX.	Copied		
0005	0	002D	XXX.	Copied		
0001	1080121	0051	XXX.	CopyInProg		
0005	1084299	0055	XXX.	CopyInProg		
0001	1097749	0079	XXX.	PreCopy		
0005	1098869	007D	XXX.	PreCopy		
Total			-----			

Tracks	4361038
MB(s)	136282

Legend:

(C): X = The background copy setting is active for this pair.
 . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with a group.
 . = The Target device is not associated with a group.
(D): X = The Clone session is a differential copy session.
 . = The Clone session is not a differential copy session.
(P): X = The precopy operation has completed one cycle
 . = The precopy operation has not completed one cycle

Incrementally restore from the first set of clones to the source devices.

```
# symclone -file ./clone_pairfile01 restore -nop -sid 1380
```

```
'Incremental Restore' operation execution is in progress for the device list in  
device file './clone_pairfile01'. Please wait...
```

```
'Incremental Restore' operation successfully initiated for the device list  
in device file './clone_pairfile01'.
```

After a restore operation has completed, we execute a terminate to clean up.

```
# symclone -file ./clone_pairfile01 terminate -nop -sid 1380
```

```
'Terminate' operation execution is in progress for the device list in  
device file './clone_pairfile01'. Please wait...
```

```
'Terminate' operation successfully executed for the device list  
in device file './clone_pairfile01'.
```

```
# symclone -file ./clone_pairfile02 terminate -nop -sid 1380 -symforce
```

```
'Terminate' operation execution is in progress for the device list in  
device file './clone_pairfile02'. Please wait...
```

```
'Terminate' operation successfully executed for the device list  
in device file './clone_pairfile02'.
```

```
# symclone -file ./clone_pairfile03 terminate -nop -sid 1380
```

```
'Terminate' operation execution is in progress for the device list in  
device file './clone_pairfile03'. Please wait...
```

```
'Terminate' operation successfully executed for the device list  
in device file './clone_pairfile03'.
```

Confirm no copy sessions are left.

```
# symclone list -sid 1380
```

```
Symmetrix ID: 000192601380
```

```
No Copy sessions found
```