



Global Distributed Storage Engineering

Restricted Release

For use in SRDF/A environments only

TimeFinder Clone

GDSE-05008-SG

Solutions Guide

Version 1.3

VERSION CONTROL

Version and Date of Issue	Author	Comment
V1.0.7 03/05/05	Chris Clayton	First draft
V1.1 11/05/05	Chris Clayton	Revised version following Peer Review by: Peter Carnegie, Colin Bradley and Stephen Hopkins.
V1.2 04/08/05	Chris Clayton	Revised version following FRC feedback from Kery Hines on device file usage
V1.3 04/10/05	Chris Clayton	Updated as restricted release for SRDF/A only

CONTENTS

Legal Notice	5
Contact	5
Latest Versions	5
Confidentiality	5
1 DOCUMENT SCOPE	6
1.1 Target Audience	6
1.2 Out of scope	6
2 INTRODUCTION TO CLONES	7
2.1 Overview of TimeFinder Clones	7
2.1.1 Key differences between TimeFinder/Clone and TimeFinder/Mirror	8
2.2 Applicable environments	8
2.3 Clone terminology	9
3 SUMMARY OF RECOMMENDATIONS	10
4 PRE-REQUISITES FOR USING CLONES	11
4.1 Licensing	11
4.2 Solutions Enabler & Symmetrix microcode requirements	11
4.3 SYMACL permissions	11
5 BEST PRACTICE FOR USING CLONES	12
5.1 When to use BCVs vs Native Clones vs Clone Emulation	12
5.1.1 Native clones	13
5.1.2 TimeFinder/Mirror (BCVs)	13
5.1.3 TimeFinder/Mirror Clone Emulation	13
5.2 Using device files for clone control	15
5.2.1 Device groups, device files and GNS	16
5.3 Clone creation	17
5.3.1 Note on changing modes	17
5.3.2 Precopy clone cycling	18
5.3.3 Other notes relating to clone creation	18
5.4 Clone activation	19

5.4.1	Consistent activation	19
5.5	Clone recreation	20
5.6	Monitoring the state of clone sessions	21
5.6.1	Symclone query	21
5.6.2	Symclone list	22
5.7	Clone state verification	24
5.8	Restoring from a clone	25
5.9	Cloning to SRDF-enabled devices	26
5.9.1	Features of SRDF-enabled clones	26
5.9.2	Failover when using clones	27
5.10	Terminating clone sessions	28
5.11	Multi-session cloning	29
5.11.1	Device files	29
5.11.2	Restoring to source in a multi-session environment	30
5.11.3	Monitoring multi-sessions clones	30
5.11.4	Other multi-session clone issues	31
6	TIMEFINDER/MIRROR CLONE EMULATION	32
6.1	Recommended usage	32
6.2	Clone Emulation Issues	32
6.3	Using TimeFinder/Mirror in Clone Emulation Mode	32
7	CLONE PERFORMANCE	34
7.1	Key findings	34
7.2	Copy rates	34
7.2.1	Impact of SRDF replication on copy rates	34
7.2.2	Impact of source IO on copy rates	34
7.2.3	Parallel session copy rates.	34
7.2.4	Single vs multi-session copy rates	34
7.2.5	Example Copy Rates	34
7.3	IO impact	35
7.3.1	Impact of clone copying on source IO response	35
7.3.2	Impact of multiple sessions on host IO response	35
APPENDIX A – EQUIVALENT SYMCLONE AND SYMMIR OPERATIONS		37
APPENDIX B – MULTI-SESSION CLONE EXAMPLE		38

Legal Notice

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

CITI, CITIBANK, CITICORP, CITIGROUP and the Umbrella Device 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 "GT Global Distributed Storage Engineering" distribution list or alternatively to gdse@citigroup.com.

Latest Versions

It is recommended that the latest version of the document be used from the GDSE website <http://gdse.ny.smb.com>. Newer versions may include expanded scope, or comments on issues discovered after deployment.

Confidentiality

This document is confidential and no part of it should be shared by EMC with any of its other customers.

1 Document Scope

This Solutions Guide recommends Best Practice for using TimeFinder/Clone technology on EMC DMX 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 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 6.0	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	
TimeFinder Clone Certification Project Scope	http://gdse.ny.ssmb.com/GDSE_docs/GDSE-05008-PS.pdf
TimeFinder Clone Release Notes	http://gdse.ny.ssmb.com/GDSE_docs/GDSE-05008-RN.pdf

1.1 Target Audience

This detailed document is aimed at SAN BAU teams and applied 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

- Use of RAID/5 BCVs (which use clone emulation) as RAID/5 is not currently supported on DMX by Citigroup. Support of RAID/5 BCV activities would need to be done after certification of RAID/5 BCV devices as part of any new bin file standards.
- Control of clone pairs that span multiple Symmetrix arrays

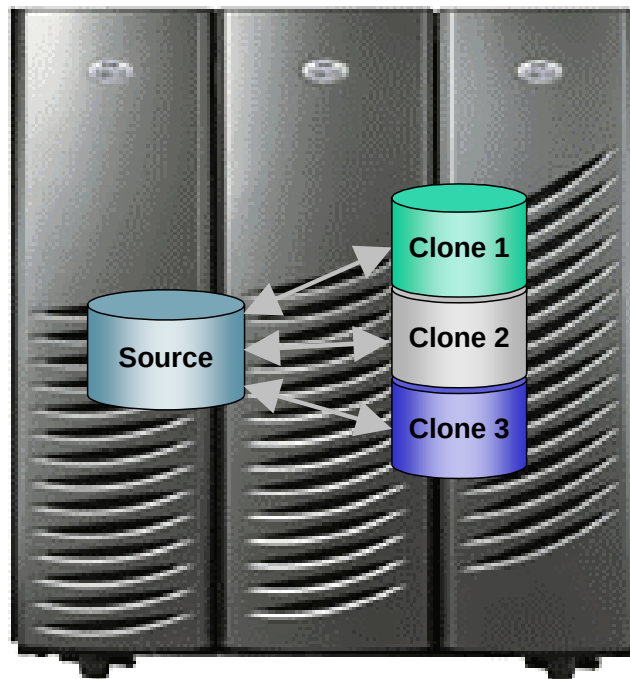
2 Introduction to clones

2.1 Overview of TimeFinder Clones

TimeFinder/Clone software for DMX allows one to clone copies of data simultaneously on one or more target devices from a single source device within the same DMX. 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 DMX 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.

The illustration below shows a single source device with three concurrent copy or clone 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.

It is also possible to use TimeFinder/Mirror commands (symmir) to manipulate BCV-based clones when used in clone emulation mode.

2.1.1 Key differences between TimeFinder/Clone and TimeFinder/Mirror

The following are the most important differences between Clones and BCV mirrors:

- Clones do not consume mirror positions. Hence, it is possible to:
 - o have many (up to 16) simultaneous clone targets from a single source
 - o have mirrored copy session targets in an SRDF environment
 - o have a copy session with a concurrent SRDF source

The above are not possible with TimeFinder/Mirror BCVs

- It is possible to use Standard devices as clone targets. These can also be R1 devices (for copy and precopy clones only). TimeFinder mirrors, on the other hand, cannot use standard devices as targets; BCV devices must be configured in the bin file, which is a major drawback.
- The clone target can be accessed before the copying has completed (with however a performance impact on the source and target). This is not possible with TimeFinder Mirrors.

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 Symmetrix devices
- 4) Are on locally mirrored (protected) storage
- 5) Are (optionally) on SRDF replicated storage (although the synchronous nature of the clone target device is not respected by the clone copy process).
- 6) That can be differentially resynchronised with the source data
- 7) Can be used to differentially restore to the source data (limited functionality)

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 at present).
- 4) To create rollback point-in-time copies for application development testing.

Clones also can be used to create point-in-time copies of data at the source site of a concurrent SRDF/A environment where there are insufficient mirror positions to use BCV mirrors.

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 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.

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.

CopyOnAccess (CoA) clones are not certified by GDSE for general use. The main issues are:

- The copy is never completed and hence the clone is never independent of the source. Hence, there will always be some non-deterministic, residual source and target IO response impact.
- It is not possible to perform differential activities with a CoA clone (e.g. incremental resync and incremental restore)
- It is not possible to do even a full restore with a CoA clone as the copy never completes
- It is not possible to use CoA cloning to an SRDF-enabled R1 target

TimeFinder/Mirror Clone Emulation should only be used when native clones or standard TimeFinder/Mirror cannot provide the required functionality, as there are issues related to using clone emulation. These are detailed in Section 6.

Further information on when it is appropriate to use TimeFinder/Mirror Clone Emulation mode can be found in Section 5 of this document.

4 Pre-requisites for using clones

Potential users of clones should first confirm with their SAN Bau team that clones are supported on the DMX storage array on which their application resides. Use of clones has certain licensing, microcode version and security configuration requirements which may not be met by all DMX installations.

4.1 Licensing

Use of clones (either via symclone commands or via TimeFinder/Mirror Clone Emulation) requires a TimeFinder/Clone license. See the GDSE document "TimeFinder/Clone Release Notes" for further details on clone licensing.

The license key 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 is 6.0. The symclone command is part of the Solutions Enabler package.

The minimum certified level of Symmetrix microcode required for clone operations is 5671.3236. Hence, clone operations are only possible on certain DMX storage arrays.

4.3 SYMACL permissions

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

- as a member of the symacl AdminGrp on the DMX to be used. This should only be used for SAN management servers as this provides complete control over the DMX.
- 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.

5 Best Practice for Using Clones

5.1 When to use BCVs vs Native Clones vs Clone Emulation

There are potentially three ways in which fully source-independent point-in-time copies of LUNs can be made within a DMX storage array:

- Using Standard mirrored storage and TimeFinder/clone (symclone commands), a.k.a native clones, or just clones.
- Using BCVs and TimeFinder/Mirror (symmir commands). Typically just referred to as BCVs.
- Using BCVs and TimeFinder/Mirror in Clone Emulation mode.

All offer high-performance, differential restore and resynchronisation. The technology you should use will depend on the constraints of the application requirements, and more importantly, what is supported in your local DMX environment.

Summary

Clones should be used for application copies wherever possible. BCVs should be used where the local Symmetrix environment cannot support clones or where full multi-session restore capability is required. Clone emulation should only be used where BCVs are needed but cannot be used due to mirror position limitations.

The differences between the three options are summarized in the following table.

Characteristic	“BCV”s TimeFinder/Mirror (symmir) with BCV devices	“Clones” TimeFinder/Clone (symclone) with Standard devices	“Clone Emulation” TimeFinder/Mirror (symmir) with BCV devices
Available on Symmetrix below 5671 microcode?	Yes	No	No
Can copy to Standard (non-BCV) mirrored storage?	No	Yes	No
Can copy to SRDF replicated storage?	No	Yes, but synchronous status of R1 is not respected	No
Number of concurrent copies	Maximum of 2. 0 or 1 in some configurations.	8 (with differential resync/restore capabilities)	8
Permits differential resynchronisation with source	Yes	Yes	Yes
Works with device groups	Yes	Limited	Yes
Permits restores in a multi-session environment?	Yes	Yes, but limited today	Yes, but limited today
Permits on-going copy session activities after a restore?	Yes	No today	Yes

5.1.1 Native clones

You should use native clones under the following circumstances:

- If the SAN BAU team confirm that cloning is supported at your site and on the DMX to be used:
 - o If the DMX to be used is running 5671.3236 microcode or later
 - o The DMX is licensed for TimeFinder/Clone
- If the target of your copy operation must be protected (locally-mirrored) storage.
- If the target of your copy operation must be to storage that is SRDF replicated. Note, however that:
 - o when the clone session is created, the RDF Pair State for the clone target device changes from Synchronized to SyncInProg i.e. the copying to the clone is not done in a manner that respects the Synchronous replication status of the R1. Instead, the clone target R2 lags behind in an inconsistent state.

The issue applies to both precopy mode and activated copy mode clones.

Hence, cloning to an SRDF/S R1 device does NOT result in a clone copy that is identical at both the Production and COB site at all times as one might expect.

- o copy times will be much greater when cloning to R1 devices as this generates a lot of RDF traffic and hence reads from the source.

Hence, cloning to an Synchronous SRDF R1 device has limited appeal.

Note that it is also possible to clone from both an R1 and its corresponding R2 to achieve a similar result to replicating the clone itself, without the SRDF performance overhead and without having to determine if the clone target SRDF synchronization has completed.

- If you require copies that do not consume mirror positions (e.g. if your source volumes are concurrent SRDF R1 volumes or you need multiple copies).

5.1.2 TimeFinder/Mirror (BCVs)

You should only use TimeFinder/Mirror with BCVs under the following circumstances:

- If you require the ability to do a restore to your source volumes from one of your copies, whilst still maintaining the ability to retain the copies for further possible restores

5.1.3 TimeFinder/Mirror Clone Emulation

You should only use TimeFinder/Mirror Clone Emulation with BCVs under the following circumstances:

If (and only if) you require:

- the ability do restores from one of your copies to your source volumes whilst still maintaining the ability to retain the copies for further possible differential restores

and you also require

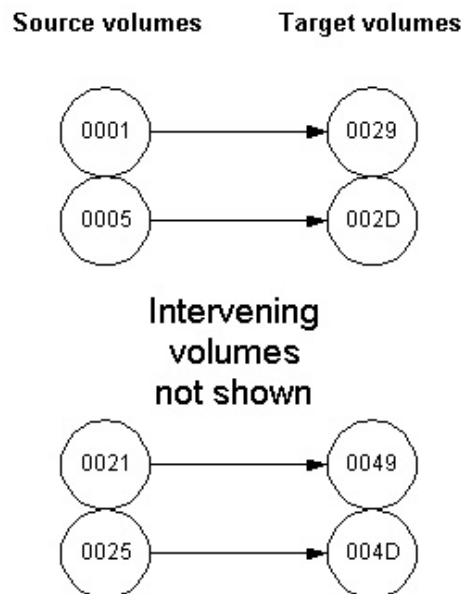
- copies that do not consume mirror positions

5.2 Using device files for clone control

Control of clones using standard devices as targets cannot be easily achieved using device groups. The fundamental design of Solutions Enabler relies on three different kinds of device groups (RDF1, RDF2 and REGULAR) and local replication targets i.e. BCVs. To allow standard devices to replicate to other standard devices, or standards to replicate to RDF1 devices, etc will required a redesign of the basic device group architecture by EMC. Further information on the issue can be found in the TimeFinder/Clone Release Notes document GDSE-05008-RN.

As a consequence, it is 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 is recommended that device files be used to control clone operations to overcome the restrictions imposed by device groups.

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.

5.2.1 Device groups, device files and GNS

There is one significant 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. Furthermore, if device groups are used, the command lines used to manage clones become very long with errors possible.

For example, to activate/create/terminate/etc a clone of a DB covering 10 LUNs would need the following syntax for files and group respectively:

DeviceFile

```
# symclone -f <device-file> activate - consistent -noprompt
```

Device group

```
# symclone -g <DGName> activate - consistent \
DEV001 sym ld DEV011 DEV002 sym ld DEV012 \
DEV003 sym ld DEV013 DEV004 sym ld DEV014 \
DEV005 sym ld DEV015 DEV006 sym ld DEV016 \
DEV007 sym ld DEV017 DEV008 sym ld DEV018 \
DEV009 sym ld DEV019 DEV010 sym ld DEV020 \
-noprompt
```

Maintenance of this script would be greater than maintaining a device file. Hence, the use of device groups for clone control is not forbidden but is discouraged for all the above reasons.

If the application spans multiple frames, and a "consistent" clone across multiple frames is needed, then this will require the use of Composite Groups and not device files.

5.3 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 to 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
```

Additionally `–force` will be required if the target devices are active R1s as takes the RDF Pair State out of synchronized state, and hence should be done knowingly. `–force` is not needed if the SRDF link is suspended.

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 in 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).

5.3.1 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.

5.3.2 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.

5.3.3 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.

5.4 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 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).

5.4.1 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. Use of the –consistent option when activating a clone session uses 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.

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

Additionally -force will be required if the target devices are active R1s. –force is not needed if the SRDF link is suspended.

5.5 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. Additionally `-force` will be required if the target devices are active R1s. `–force` is not needed if the SRDF link is suspended.

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
```

5.6 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 follow is example output from a clone command.

```
# symclone -f clone_pairfile -sid 0822 query -multi
```

```
Device File Name      : ./clone_pairfile
Device's Symmetrix ID : 000187700822
```

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 DMX.

```
# 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 0822
```

```
Symmetrix ID: 000187700822
```

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

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.

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.

With Solutions Enabler 6.0 and microcode 5671.3236, there are two very important limitations with clone restores:

- A restore operation is not allowed if there is more than one active clone session i.e. restores are not possible in a multi-session environment. It is necessary to first terminate all other copy sessions with the source device before a restore can take place from the last remaining session. This is described in more detail in a later section.

Note that after termination, fully copied clones that had been activated are fully independent copies of the data. All that termination does is remove the copy session relationship between the source and target. If the target data is then to be restored to the original source, then it will be necessary to create a new clone session with the original source as the target and vice versa. The "restore" will be a full copy from the original target to the original source.

- After a restore operation is completed, the only permitted clone operation is termination of the remaining clone session. This is serious limitation as to continue using the clone (after a restore) it will be necessary to terminate and create the session again, with a full copy of the data, even though the volumes are identical.

Both of the above issues are being addressed by the vendor and will be fixed in a future release.

The syntax for a restore is

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

Additionally `-force` will be required if the source devices that you are restoring to are active R1s. `-force` is not needed if the SRDF link is suspended.

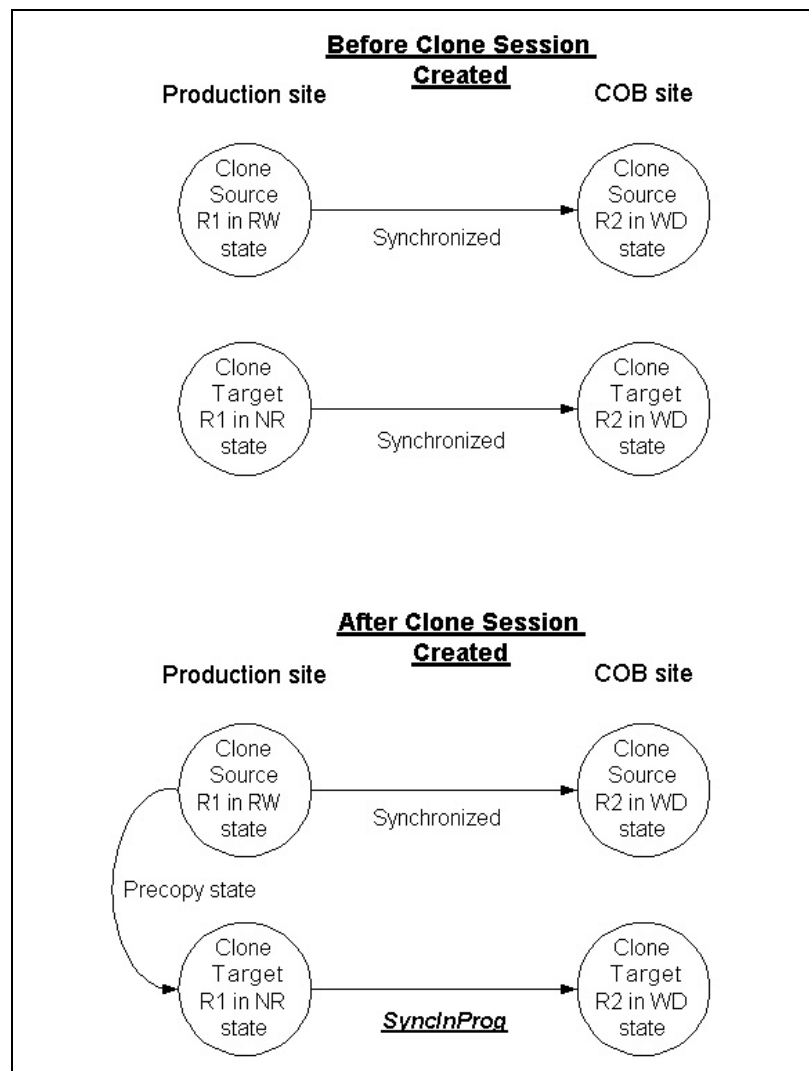
5.9 Cloning to SRDF-enabled devices

It is possible to clone to SRDF-enabled devices. However, there are some unexpected limitations and complications when mixing SRDF and cloning which need to be understood. In practice, it is unlikely that cloning to SRDF devices will meet your expectations.

5.9.1 Features of SRDF-enabled clones

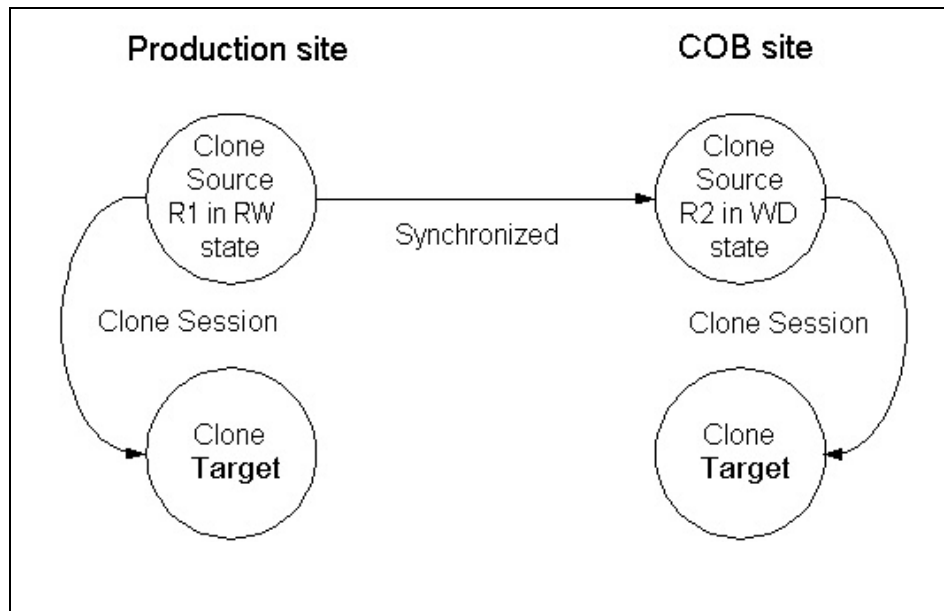
When a clone session is created precopy state, (or if a clone session in copy mode is activated) in any data that is copied to the target R1 device is **NOT** replicated to the clone target's R2 device in a synchronous manner.

The RDF Pair state is marked as *SyncInProg*. This is unexpected and indicates that when you clone to a Synchronous SRDF R1 device, the copy to the clone is **NOT** happening in a synchronous manner i.e. the R1 and R2 are not identical and invalid tracks build up on the R1 side.



The same issue applies to copy mode clones that have been activated and which are copying.

Hence, if a clone of production data is required at both the Production and COB site, it is preferable to create separate clone sessions from both the source R1 and R2, as illustrated in the diagram below:



5.9.2 Failover when using clones

Failover of the source (production) volumes should be done independently from failover of the target (clone) devices. This is necessary, as SRDF-enabled devices cannot be failed over (without use of `-symforce`) if:

- They are in precopy state. R1 clone device is in NR state and the
- They are activated but the SRDF link is not yet in Synchronized state.

Hence, if the source and target devices are both in the same SRDF failover device group, then the state of the clone sessions may prevent the group and hence the production volumes from being failed over using standard failover commands.

It is recommended that device groups set up by the SAN BAU team for production SRDF failover do not include clone target devices. These should be in a separate device group.

5.10 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
```

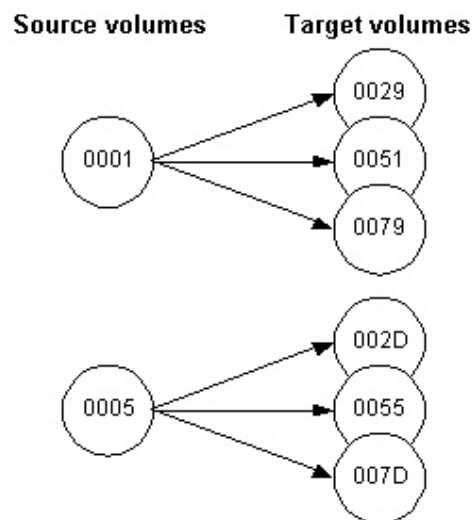
Note that testing has found that clone termination has a large negative impact on source IO response, in particular if data is still being copied from the source to the target. Hence, termination should be scheduled at a time when IO response is not critical.

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.

5.11 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.

Multi-Session Clones



5.11.1 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
```

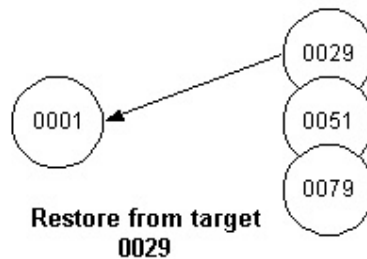
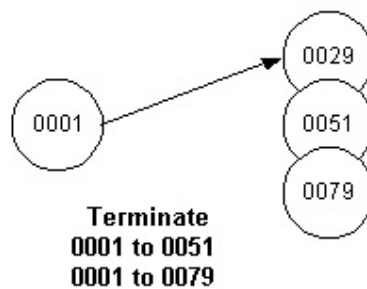
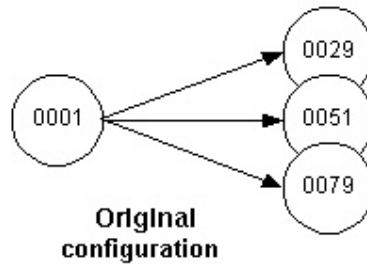
```
# cat clone_pairfile03
0001 0079
0005 007D
```

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

5.11.2 Restoring to source in a multi-session environment

At the time of writing, it is not possible to restore from a clone if there is more than one active session. It is necessary to first terminate the other relationships. In the example below, one must terminate the relationships between 0001 & 0051, and 0001 & 0079 before one can restore from 0029 to 0001. For example:

Source volumes Target volumes



5.11.3 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 0822

Symmetrix ID: 000187700822

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

5.11.4 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 (see TimeFinder/Clone Release Notes document Appendix B for more details). Hence, simultaneous full copies should be avoided.

6 TimeFinder/Mirror Clone emulation

6.1 Recommended usage

You should only use TimeFinder/Mirror Clone Emulation with BCVs under the following circumstances:

If (and only if) you require:

- the ability do restores from one of your copies to your source volumes whilst still maintaining the ability to retain the copies for further possible differential restores

and you also require

- copies that do not consume mirror positions

TimeFinder Mirror Clone Emulation allows one to execute legacy symmir BCV control commands without concern for mirror position limitations. This is achieved under clone emulation mode by executing the BCV operations using clone technology under the covers. For example, under clone emulation a `symmir establish -full` command maps to `symclone create -establish`, and `symmir split` maps to `symclone activate`.

6.2 Clone Emulation Issues

- EMC have provided the clone emulation mode for TimeFinder/Mirror to enable customers with legacy TimeFinder scripts to continue using their scripts whilst providing the enhanced functionality that comes with emulation e.g. larger number of concurrent BCVs, RAID-5 BCVs, etc. They do not recommend using clone emulation for any other purpose.
- Testing found issues when restoring from BCVs to standards under clone emulation. Illegal activities are not currently trapped by the API and can result in the DMX requiring dial-in remediation by EMC. Testing also found issues with reporting of invalid track information in Emulation mode.
- TimeFinder/Mirror Clone Emulation requires BCVs, whereas native clones can utilize standard storage, hence avoiding the need for bespoke bin files.

6.3 Using TimeFinder/Mirror in Clone Emulation Mode

Clone emulation is used if the clone emulation environment variable is set to ENABLED.

```
# SYMCLI_CLONE_EMULATION=ENABLED
# export SYMCLI_CLONE_EMULATION
```

One can confirm if this (and indeed other) SYMCLI environment variables have been set with the following command:

```
# symcli -def
```

```
Symmetrix Command Line Interface (SYMCLI) Version V6.0.0.0 (Edit Level: 623)  
built with SYMAPI Version V6.0.0.0 (Edit Level: 623)
```

Current settings of the SYMCLI environmental variables:

```
SYMCLI_CLONE_EMULATION: ENABLED
```

Under Clone Emulation, clones are controlled exclusively via TimerFinder/Mirror commands (symmir). When using Clone Emulation, the clones created via symmir cannot be controlled or even monitored using symclone commands. A primer on using TimeFinder/Mirror (symmir) is beyond the scope of the present document; emulation mode should really only be used in existing TimeFinder/Mirror implementations with mirror position issues.

7 Clone Performance

7.1 Key findings

The full results of the clone performance testing undertaken during evaluation can be found in Appendix B of the GDSE document TimeFinder/Clone Release Notes. However, some of the key findings relating to Best Practice are briefly summarized below:

7.2 Copy rates

7.2.1 Impact of SRDF replication on copy rates

Full copy times to synchronously replicated targets will be significantly impacted by latency and bandwidth of the RDF link. Incremental updates may similarly be impacted depending on the amount of data to be synchronized.

7.2.2 Impact of source IO on copy rates

High IO to source volumes does impact copy rates, but only by a very modest amount (a few percent for high IO load in test conducted – see Release Notes for more detail).

7.2.3 Parallel session copy rates.

Aggregate copy throughput increases, as more volumes are copied in parallel, up to a maximum of 200MB/sec or 12GB/minute (for the test configuration). The limit will vary depending on DMX configuration.

7.2.4 Single vs multi-session copy rates

Copy durations increase with concurrent session count in a non-linear manner. Concurrent full copies should be avoided if possible. Sequential copies can be created by sequential creation of precopy clones.

7.2.5 Example Copy Rates

Copy rates (including those for restore and resynchronisation) will vary depending on:

- 1) Bin file layout
- 2) Choice of LUNs to be used and what other applications they are sharing with
- 3) RDF bandwidth and latency (if target is an active SRDF device)

Hence, it will be necessary to check copy rates for your particular implementation of clones once disks are allocated. The following examples are based on a standard GDSE bin file (with 34GB LUNs comprised of four 8GB hypers) on a DMX2000P. Your mileage will vary.

In the table on the following page, 3 → 3 indicates a copy (in parallel) of 3 * 34GB source LUNs to 3 * 34Gb target LUNs.

Test configuration (all 34GB LUNs)	Operation	Copy Time	Aggregate Copy Rate
Single-session copies			
1 → 1 (no SRDF)	Full copy	815 sec	42 MB/sec
3 → 3 (no SRDF)	Full copy	815 sec	127MB/sec
5 → 5 (no SRDF)	Full copy	830 sec	208 MB/sec
10 → 10 (no SRDF)	Full copy	1680 sec	206 MB/sec (12 GB/min)
Single session copies to RDF-enabled target devices			
1 → 1 (SRDF-enabled)	Full copy	1130 sec	30 MB/sec
10 → 10 (SRDF-enabled)	Full copy	5810 sec	59 MB/sec
Restore operations			
10 ← 10 (no SRDF; 15% data change)	Incremental Restore/Resync	290 sec	170MB/sec
10 ← 10 (no SRDF; 50% data change)	Incremental Restore/Resync	930 sec	182MB/sec
Multi-session copies			
10 → 10 (no SRDF)	Full copy	1680 sec	206 MB/sec
10 → 20 (no SRDF)	Full copy	6250 sec	55 MB/sec
10 → 30 (no SRDF)	Full copy	9320 sec	37MB/sec

7.3 IO impact

7.3.1 Impact of clone copying on source IO response

There is a large impact on source IO response time, especially for writes, during the copy and precopy clone copy phase, which reduces to “no clone” values as the copy progresses. This is because for an activated clone you are no longer simply writing to cache (for blocks yet to be copied) but first reading the block to be written to, copying that block to the clone and then finally writing the new IO.

Hence, it is advisable to schedule copying at a time when source IO response is not critical. This can most easily be achieved by using precopy clones where the copying (initiated by a create or recreate action) can be divorced from the splitting off of the point-in-time copy (activation).

Once the copy is complete, and the clone session has been activated, there is no IO response impact on the source volumes.

CopyOnAccess clones spread the IO impact over a longer period but at the loss of significant functional benefits. Use of CoA clones is not certified production use.

7.3.2 Impact of multiple sessions on host IO response

The existence of multiple active copy targets increases write, and to a lesser extent, source read IO response. The impact increases non-linearly with the number of active copy sessions.

Hence, multiple copy sessions where data is actually being copied simultaneously should be avoided, if possible.

Termination of clone sessions has a large negative impact on IO response if the termination occurs when copying is in progress. This impact continues beyond the completion of the termination command.

Appendix A – Equivalent symclone and symmir operations

The following table compares the equivalent operations for TimeFinder/Clone (Native Clone) and TimeFinder/Mirror (BCV).

“Native Clones” TimeFinder/Clone command	“BCVs” TimeFinder/Mirror equivalent operation
symclone create (default is to create a Copy On Access Clone)	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

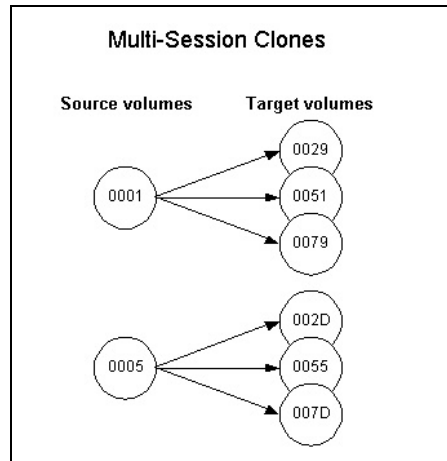
Appendix B – Multi-session clone example

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 0822
```

```
'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 0822 -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 0822 query -multi
```

```
Device File Name      : ./clone_pairfile01
Device's Symmetrix ID : 000187700822
```

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 0822
```

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 0822
```

'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 0822 query -multi
```

```

Device File Name      : ./clone_pairfile01
Device's Symmetrix ID : 000187700822

```

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 0822
```

'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 0822 query -multi
```

```
Device File Name      : ./clone_pairfile01
Device's Symmetrix ID : 000187700822
```

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 0822
```

'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 0822 query -multi
```

Device File Name : ./clone_pairfile01
Device's Symmetrix ID : 000187700822

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 0822
```

Symmetrix ID: 000187700822

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

Before a restore operation can take place from the clone associated with file clone_pairfile01, all the other sessions need to be terminated. –symforce is used as termination of CopyInProg sessions requires this.

```
# symclone -file ./clone_pairfile02 terminate -nop -sid 0822 -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 0822
```

```
'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'.
```

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

```
# symclone -file ./clone_pairfile01 restore -nop -sid 0822
```

```
'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, the only possible action on the clone session is a terminate.

```
# symclone -file ./clone_pairfile01 terminate -nop -sid 0822
```

```
'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'.
```

Confirm no copy sessions are left.

```
# symclone list -sid 0822
```

```
Symmetrix ID: 000187700822
```

```
No Copy sessions found
```