EMC²
where information lives®

White Paper

# EMC TimeFinder with Sybase ASE 15

*Best Practices Planning*


***Abstract***

The purpose of this white paper is to provide a comprehensive set of best practices and procedures when implementing Sybase ASE 15 with EMC storage-based replication technologies for business continuance.



April 2008

Part Number H4398

# Table of Contents

# Executive summary

The purpose of this white paper is to provide a comprehensive set of best practices and procedures when deploying Sybase ASE 15 with EMC® TimeFinder® for storage-based local replication.

# Introduction

This white paper demonstrates the integration of TimeFinder with Sybase Adaptive Server Enterprise (ASE) technology to optimally address emerging business continuance needs for a database or instance.

The need to increase nightly batch processing and backup procedure windows is growing out of proportion to the number of hours available to complete these time-consuming tasks. EMC TimeFinder combined with Sybase ASE can significantly reduce recovery times and shorten backup windows.

The TimeFinder (TF) product line consists of TimeFinder/Mirror, TimeFinder/Clone, TimeFinder/CG (consistency groups), and TimeFinder/Snap and provides business continuance software technology, by allowing users to create point-in-time copies of source data on multiple target devices. TimeFinder provides the functionality to create consistent, restartable database images, which can then be used for decision support, backup, restore, or application testing.

TimeFinder/Clone was specifically used throughout the project in order to demonstrate the versatility and flexibility of this software technology. Symmetrix® TimeFinder/Clone operations allow the creation of clone copies of a source device onto multiple target devices. The source and target devices can be either standard or BCV devices as long as they are all of the same size and emulation type (FBA or CKD). Clone copies of striped or concatenated metadevices can also be created, but the source and target metadevices must be completely identical in stripe count, stripe size, and capacity. In addition, clone target devices can be RAID 5 protected. Once activated, the copy can be instantly accessed by a target's host, even before the data is fully copied to the target device.

TimeFinder/Clone copies are appropriate in situations where multiple copies of production data are needed. Clone copies can also be used to reduce disk contention and improve data access speed by assigning users to copies of data rather than accessing the one production copy. Depending on whether a device has associated BCVs or snaps, a single source device may have as many as 16 clone copies.

This white paper will document the procedures and best practices for the following use cases with TimeFinder/Clone:

- For creating consistent database copies and refreshing the databases with periodic updates
- For log shipping as a backup mechanism
- For restore (combined with Sybase features) to provide roll-forward log capability

## *Audience*

This white paper is intended for those needing a better understanding of EMC Symmetrix TimeFinder best practices for backup, restart, and recovery of a Sybase ASE 15 database or instance.

# Technology overview

This section provides an overview of the EMC and Sybase products and software technology that are used by joint customers and are discussed throughout this white paper.

## EMC Symmetrix DMX

All Symmetrix systems provide advanced data replication capabilities, full mainframe and open systems support, and flexible connectivity options, including Fibre Channel, FICON, ESCON, Gigabit Ethernet, and iSCSI. Interoperability between Symmetrix storage systems enables current customers to migrate storage solutions from one generation to the next, protecting their investment even as their storage demands expand.

EMC delivers two software products with Enginuity™ that optimize performance with multi-tiered Symmetrix systems. Dynamic Cache Partitioning provides dedicated memory resource allocation. It enables the segregation of cache on a Symmetrix into as many as eight partitions for different applications. Partitions can expand and contract according to policies in order to maximize performance while providing isolation between applications. Symmetrix Policy Controls help manage multiple application workloads by setting priority levels for device groups, allowing higher-priority applications to have faster response times than lower-priority applications. In addition, Symmetrix features provide:

- Continuous cache and on-disk data integrity checking and error detection/correction
- Fault isolation
- Nondisruptive hardware and software upgrades
- Automatic diagnostics and phone-home capabilities

These systems provide scalable performance and capacity from as few as 32 disks to over 2,400. For a complete description of the Symmetrix DMX™ product line go to EMC.com: http://www.emc.com/products/family/symmetrix-family.htm

At the software level, advanced integrity features ensure information is always protected and available. By choosing RAID 1 (mirroring), RAID 5 (3+1 and 7+1), or RAID 6 (6 + 2 and 14 + 2) protection, users have the flexibility to choose the protection level most appropriate to the value and performance requirements of their information. Symmetrix DMX-4 is EMC's latest generation of high-end storage solutions.

## TimeFinder

The EMC Symmetrix Command Line Interface (SYMCLI) TimeFinder component extends the basic SYMCLI command set to include TimeFinder or business continuity commands that allow control operations on device pairs within a local replication environment. TimeFinder provides four types of devices as follows:

- TimeFinder/Mirror — General monitor and control operations for business continuance volumes (BCV)
- TimeFinder/CG — Consistency groups
- TimeFinder/Clone — Clone copy sessions
- TimeFinder/Snap — Snap copy sessions

The best practices in this paper apply to TF/Mirror, TF/Clone, and TF/Snap, however, the examples used are specific for TF/Clone.

### TimeFinder/Clone

TimeFinder/Clone functionality is controlled via copy sessions that 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 TimeFinder/Clone devices. The session is then activated, enabling the target device to be accessed by its host. When the information is no longer needed, the session can be terminated. TimeFinder/Clone operations are controlled from the host by using the symclone command to create, activate, and terminate the copy sessions.

## *Sybase ASE 15*

ASE 15 meets the demands of large databases and high transaction volumes, while providing a database management system. Its key features include on-disk encryption, smart partitions and new, query processing technology that has demonstrated a significant increase in performance, as well as enhanced support for unstructured data management.

**Table 1. Sybase ASE 15.0.2 feature listing**

| Operating System | Sol Sparc 32 | Sol Sparc 64 | HP PA-RISC 64 | AIX64 | Linux x86 | Windows x86 | Linux Power | Linux x64 | Solaris x64 | HP-UX Itanium II | Windows x64 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Options | | | | | | | | | | | |
| High Availability | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | | | ✔ | |
| Security & Directory Services | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Secure Socket Layer | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ | ✔ | ✔ |
| Cybersafe Kerboros | ✔ | ✔ | | | | ✔ | | | | | |
| MIT Kerberos | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ | ✔ | ✔ | |
| Platform Native Kerberos | ✔ | ✔ | | | | | | | | | |
| Fine Grained Access Control | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Pluggable Authentication Module | ✔ | ✔ | | ✔ | ✔ | | ✔ | ✔ | ✔ | ✔ | |
| LDAP Server Directory | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| LDAP User Authentication | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ | ✔ | |
| Encrypted Columns | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | ✔ | ✔ | |
| Enhanced Full Text Search | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | ✔ | | | |
| Partitions | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Features included with ASE | | | | | | | | | | | |
| Cross Platform Dump and Load | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Distributed Transaction | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Job Scheduler | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| ASE Replicator | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | | | | | |
| IPv6 | ✔ | ✔ | ✔ | | ✔ | ✔ | | | | | |
| Java Option | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Native XML | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Web Services | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | |
| Content Management | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| Archived Database Access | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

For the complete set of Sybase ASE supported platforms and feature listings, refer to:
http://www.sybase.com/files/Data_Sheets/ASE_15.0.2_062507.pdf

# Best practices using TimeFinder

## *Creating and updating consistent copies*

This section describes how to create consistent database copies, and then refresh the database on a periodic basis with changed data. TimeFinder/Clone copies are appropriate in situations where multiple copies of production data are needed for testing, backups, or report generation. Customers generally decide, before implementing a solution, how often data should be refreshed based on business user needs.

The following TimeFinder/Clone tests were performed to create a complete set of best practice uses for creating consistent database copies. In addition, the steps for refreshing the database with periodic updates are described.

1. Create the Sybase primary and target instances if they do not already exist. In our test lab, the Sybase instances were created on Solaris x86 hosts using ASE 15.0.1.

2. UNIX symbolic links were created in order to define the database devices for Sybase.

#Sybase data & log devices on primary host:

```
lrwxrwxrwx   1 root     20 Sep 18 20:04 datadev1 -> /dev/rdsk/emcpower0c
lrwxrwxrwx   1 root     20 Sep 18 20:04 datadev2 -> /dev/rdsk/emcpower1c
lrwxrwxrwx   1 root     20 Sep 18 20:04 datadev3 -> /dev/rdsk/emcpower2c
lrwxrwxrwx   1 root     20 Sep 18 20:04 datadev4 -> /dev/rdsk/emcpower3c
lrwxrwxrwx   1 root     20 Sep 18 20:04 logdev1 -> /dev/rdsk/emcpower4c
lrwxrwxrwx   1 root     20 Sep 18 20:04 logdev2 -> /dev/rdsk/emcpower5c
```

# Sybase data & log devices on target host:

```
lrwxrwxrwx   1 root     20 Sep 28 16:03 datadev1 -> /dev/rdsk/emcpower6c
lrwxrwxrwx   1 root     20 Sep 28 16:03 datadev2 -> /dev/rdsk/emcpower7c
lrwxrwxrwx   1 root     20 Sep 28 16:03 datadev3 -> /dev/rdsk/emcpower8c
lrwxrwxrwx   1 root     20 Sep 28 16:03 datadev4 -> /dev/rdsk/emcpower9c
lrwxrwxrwx   1 root     20 Sep 28 16:03 logdev1 -> /dev/rdsk/emcpower10c
lrwxrwxrwx   1 root     20 Sep 28 16:03 logdev2 -> /dev/rdsk/emcpower11c
```

3. Use Sybase "disk init" commands to create the data and log devices for the database. In this example, a device named datadev1 of size 1000 MB is created.

```
1> disk init name=datadev1, physname="/usr/sybaseASE/ASE-
15_0/devs/datadev1", vstart=1, size="1000M"
```

4. Create the database, tables, and indexes (if they do not already exist), and then populate the tables with data.

The following steps are necessary to create a consistent copy of the database, and refresh it as needed.

1. Create the device group:

```
symdg create ctest –type regular
```

2. Add the source devices:

```
symld –g ctest addall –sid 703 –RANGE 037:042
```

3. Ensure that the target Sybase instance is shut down.

4. Create the source and target Symmetrix device pairings:

```
symclone -g ctest –file ctest_devs create -precopy –differential
```

For ease of use and manageability, use the "-file" option to specify a device file. The file "ctest_devs" contains all device pairs that the clone copy session will be created for. Using the "-precopy" option with the create argument allows the process to start copying tracks in the background, before activating the copy session. This allows the early movement of data before the point-in-time clone copy is established. Using the "-differential" option allows users to perform incremental updates for subsequent copy operations.

After creating the clone pairs, we see the devices are in a CopyOnAccess state as shown in Figure 1.

```
                    Device Group (DG) Name: ctest
                         DG's Type           : REGULAR
                    DG's Symmetrix ID    : 000190300703
          Source Device                Target Device          State     Copy
-------------------------------- ---------------------------- ------------ ----
                                 Protected Modified                Modified
Logical   Sym  Tracks    Tracks  Logical  Sym  Tracks  CGDP SRC <=> TGT  (%)
-------------------------------- ---------------------------- ------------ ----
 DEV001   0037  589860        0 TGT001   003D       0 .X.. CopyOnAccess   0
 DEV002   0038  589860        0 TGT002   003E       0 .X.. CopyOnAccess   0
 DEV003   0039  589860        0 TGT003   003F       0 .X.. CopyOnAccess   0
 DEV004   003A  589860        0 TGT004   0040       0 .X.. CopyOnAccess   0
 DEV005   003B  589860        0 TGT005   0041       0 .X.. CopyOnAccess   0
 DEV006   003C  589860        0 TGT006   0042       0 .X.. CopyOnAccess   0
                 Total          -------- --------            --------
                 Track(s)       3539160        0                   0
                 MB(s)           110599        0                   0
```

**Figure 1. symclone -g ctest query displays each device state**

5. Activate the copy session. This makes the data instantly accessible to multiple target hosts. Activate is essentially a split operation.

```
symclone -g ctest –file ctest_devs activate
```

6. Last, bring the target instance online:

```
# startserver -f RUN_LICOC036
```

Using raw devices eliminates the need for mounting. File system or LVM devices must first be mounted to the operating system before restarting the Sybase instance.

This is the contents of the Sybase “run server” file.

```
/usr/sybaseASE/ASE-15_0/bin/dataserver \
-d/usr/sybaseASE/ASE-15_0/devs/master \
-e/usr/sybaseASE/ASE-15_0/install/LICOC036.log \
-c/usr/sybaseASE/ASE-15_0/LICOC036.cfg \
-M/usr/sybaseASE/ASE-15_0 \
-sLICOC036 \
```

7. Check the data for accuracy/consistency between primary and target.

    a. Check the database log file to ensure that all databases recovered successfully.

    b. Log in to the instance and issue sp_helpdb. Ensure that the database state is normal and “online”.

8. Modify the primary database by updating or inserting data.

9. Shut down the target instance.

10. Incrementally establish the clone devices:

```
symclone -g ctest –file ctest_devs recreate
```

11. Activate the copy session again, to make the devices host accessible:

```
symclone -g ctest –file ctest_devs activate
```

12. Restart Sybase on the target host, and check for new data/consistency.

13. Repeat this part of the process each time data is modified or inserted at the primary (steps 8-13).

## *TimeFinder/Clone for log shipping*

Using TimeFinder/Clone for log shipping is an effective backup mechanism with Sybase databases. This log shipping method using TimeFinder is a restart solution that allows works in conjunction with the Sybase backup server, and the dump/load process. The steps outlined in the previous section, "Creating and updating consistent copies," should be used to materialize (or instantiate) a copy of the target database. Once that process is complete, the following procedures describe how to use TimeFinder/Clone for log shipping.

Note: The Sybase backup dump/load process may be used to materialize the target database as well.

1. On the primary host, create a UNIX symbolic link for a device that will be used for shipping the log file:

```
ln -s /dev/rdsk/emcpower7c logshipdev
```

2. On the target host, create a link for the log shipping device as well:

```
ln –s /dev/rdsk/emcpower13c logshipdev
```

3. Both on the primary and target Sybase instances, using isql, initialize a logical device using the physical device that has been mapped to the UNIX host. The database size is 5000 MB, therefore this device must be large enough to hold the entire database.

```
1> sp_addumpdevice "disk", logshipdev, "/usr/sybaseASE/ASE-
15_0/devs/logshipdev"
2> go
```

   Note: Be sure to create "logshipdev" on both the primary and target instance.

4. Create a device group containing the log shipping device:

```
symdg create lggrp -type regular
```

5. Add the source and target devices to this group:

```
symld -g lggrp add dev 043 -sid 703
symld -g lggrp add dev 044 -tgt -sid 703
```

6. Create the source and target Symmetrix device pairings using the symcli:

```
symclone -g lggrp create DEV001 sym ld TGT001 -precopy -differential
```

7. Quiesce the primary database using the "for external dump" clause:

```
1> quiesce database <tagname> hold <database name> for external dump
2> go
(For example; quiesce database tag1 hold TEST for external dump)
```

Figure 2 shows the result of querying the database in quiesce mode.

```
1> sp_helpdb
2> go
 name         db_size      owner dbid   created
      status
 ------------- ------------ ----- ------ -----------------
      ------------------------------------------------------------------
 TEST             5000.0 MB sa         4 Nov 05, 2007
      quiesce database
 master            100.0 MB sa         1 Nov 05, 2007
      mixed log and data
 model              3.0 MB sa         3 Nov 05, 2007
      mixed log and data
 sybsystemdb        4.0 MB sa     31513 Nov 05, 2007
      mixed log and data
 sybsystemprocs   132.0 MB sa     31514 Nov 05, 2007
      trunc log on chkpt, mixed log and data
 tempdb             4.0 MB sa         2 Nov 07, 2007
      select into/bulkcopy/pllsort, trunc log on chkpt, mixed log and data
```

**Figure 2. Result of querying the database state when in quiesce mode**

8. On the primary, perform a transaction log dump:

```
1> dump transaction TEST to logshipdev with standby_access
2> go Dump the transaction log
```

9. Activate the clone pair:

```
symclone -g lggrp activate DEV001 sym ld TGT001
```

10. We're now ready to restore the database from the TimeFinder clone. If the target database has been materialized using Sybase dump/load, then the target server can be started using normal procedures. If the target has been materialized using a TimeFinder clone, then the instance must be started using the –q flag. This is a signal (to the Sybase server) that the materialization has been performed using external storage. Restart the target Sybase instance using a "run server" file that looks like the following:

```
/usr/sybaseASE/ASE-15_0/bin/dataserver \
-d/usr/sybaseASE/ASE-15_0/devs/master \
-e/usr/sybaseASE/ASE-15_0/install/LICOC036.log \
-c/usr/sybaseASE/ASE-15_0/LICOC036.cfg \
-M/usr/sybaseASE/ASE-15_0 \
```

```
-sLICOC036 -q \
```

11. Apply the transaction log:

```
1> load transaction TEST from logshipdev
2> go
```

NOTE: At least one transaction log must be loaded to the target database after activating the clones so that a subsequent roll forward of transaction logs can be applied.

12. Put the database in an "online" state. Using the "for standby_access" clause allows the user to continue updating the target database by applying transaction logs; however the database is not fully read/write.

```
1> online database TEST for standby_access
2> go
```

13. To make the database fully functional (for read, write, and/or update processing) issue this command. Once the database state is "online", the process of rematerializing and re-creating transaction logs must be started again from the beginning:

```
1> online database TEST
2> go
```

## Restoring with quiesce for external dump

The `quiesce database hold for external dump` command allows the user to quiesce databases to a consistent state, split point-in-time replicas of production databases, and continuously apply transactions to the secondary server to achieve warm standby databases.

When the secondary server is started with `-q`, any database that has been split off in a quiescent state using the external dump command will be recovered similarly to the load database command. Once recovery is complete, the database will remain offline, leaving the secondary offline, and subsequent transaction logs are permitted to roll forward. In this way, databases on the secondary server can act as warm standby databases, and periodically be refreshed from the production host. The secondary replicas do not require server (instance) rebooting and the copies can be kept up to date at all times.

# Conclusion

Symmetrix TimeFinder operations using SYMCLI have been available since Enginuity version 5568. TimeFinder/Clone can create up to 16 copies from a source device onto multiple target devices. The use of TimeFinder with Sybase databases is efficient business continuance software for mission-critical applications. The TimeFinder/Mirror, TimeFinder/Snap, TimeFinder/Clone and TimeFinder/CG products provide flexibility and manageability for even the most complex Sybase environments.

In addition, EMC TimeFinder can be used in conjunction with SRDF to provide business continuity and protect the business at all times. Remote TimeFinder operations allow users to create additional copies of the SRDF target, which can then be used for test, development, and reporting. For more information, refer to the white paper *EMC SRDF/S and SRDF/A with Sybase ASE – Best Practices Planning.*

# References

The following can be found on EMC.com and Powerlink®:

- EMC TimeFinder Family page on EMC.com
- EMC Solutions Enabler Symmetrix TimeFinder Family CLI version 6.2 documentation on Powerlink
- *EMC SRDF/S and SRDF/A with Sybase ASE – Best Practices Planning* white paper

The following can be found on the Sybase website:

- Sybase ASE supported platforms and features: http://www.sybase.com/files/Data_Sheets/ASE_15.0.2_062507.pdf