
Using SYMCLI to Perform Consistent Splits with the TimeFinder Product Family

Abstract

This white paper introduces TimeFinder functionality that allows you to split off a consistent, DBMS-restartable BCV copy of your database without having to quiesce or shut down the database first.

Published 1/25/2005

Copyright © 2005 EMC Corporation. All rights reserved.

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

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

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

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

Part Number 300-000-283 REV I

Table of Contents

Introduction	4
Purpose and Scope	4
Related Documentation	4
Practical Uses	5
Consistent Split.....	5
Creating a Local DBMS-Restartable BCV Copy	8
Creating a DBMS-Restartable BCV Copy on a Remote Symmetrix	9
Creating DBMS-Restartable Copies on Both Sides	10
Using a Composite Group to Split a Set of Device Pairs That Spans Multiple Symmetrix Units.....	11
Performing a Split with Database Assist	13
Freezing an Oracle Database	14
Freezing an Informix Database.....	14
Freezing an SQL 2000 Database	14
Freezing an IBM UDB Database	15
Freezing a Sybase Database.....	15
Example 1: Creating an RDBMS-Restartable Copy on a Local Symmetrix..	16
Example 2: Creating an RDBMS-Restartable BCV Copy on a Remote Symmetrix Using PowerPath Support	23
Example 3: ECA Consistent Split of a Composite Group.....	29
Example 4: Performing Splits with Database Assist	33
Splitting BCV Pairs Using SQL 2000 Database Assist.....	33
Splitting BCV Pairs Using Oracle Database Assist.....	34

Introduction

The TimeFinder® product family (referred to as “TimeFinder” in this document) provides consistent split solutions that allow you to avoid inconsistencies and restart problems that can occur if you split a database-related BCV without first quiescing the database. Both PowerPath® software and TimeFinder/CG software support consistent splits. TimeFinder/CG is an add-on to the TimeFinder product family of base solutions, and it uses an EMC® technology called Enginuity Consistency Assist (ECA).

The TimeFinder BCV is a Symmetrix® device with special attributes created when the Symmetrix unit is configured. It can function either as an additional mirror to a Symmetrix logical volume or as an independent, host-addressable volume.

Establishing BCV devices as mirror images of active production volumes allows you to run multiple simultaneous business continuance tasks in parallel. The principal device, known as the *standard device*, remains online for regular Symmetrix operation from the original production server. Each BCV contains a unique host address, making it accessible to a separate backup/recovery server. When you establish a BCV as a mirror of a standard device, that relationship is known as a *BCV pair*. The BCV is temporarily inaccessible to its host until you *split* the BCV pair, an operation that discontinues the BCV mirroring function.

Configuring and splitting *remote* BCV pairs requires SRDF® product family software (referred to as “SRDF” in this document), which provides a mirrored data storage solution for duplicating production site data on one or more remote target Symmetrix systems. When your main systems are down for a planned or unplanned event, SRDF enables fast switchover from the source data to the target copy.

Purpose and Scope

This document describes TimeFinder functionality in versions of EMC Solutions Enabler up to version 6.0 running on Symmetrix units using Enginuity™ versions up to 5x71. The paper describes the TimeFinder consistent split operation for obtaining a local or remote DBMS-restartable BCV copy of the database.

Descriptions of split operations on device groups apply to composite groups also.

Related Documentation

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

- *Using SYMCLI to Perform TimeFinder/Mirror Control Operations* (P/N 300-000-074)
- *Using SYMCLI to Perform TimeFinder/Snap Operations* (P/N 300-001-381)
- *Using SYMCLI to Perform TimeFinder/Clone Operations* (P/N 300-001-382)
- *Using SYMCLI to Set Up TimeFinder/Mirror BCV Pairs* (P/N 300-000-072)
- *Using SYMCLI to Set Up TimeFinder/Mirror Multiple BCVs and Concurrent BCVs* (P/N 300-000-073)
- *Using SYMCLI and GNS to Propagate Group Definitions to Multiple Hosts* (P/N 300-001-384)
- *Using SYMCLI to Perform Control Operations with the SRDF Product Family* (P/N 300-000-076)
- *Using SYMCLI to Implement RDF Consistency Protection with the SRDF Product Family* (P/N 300-000-284)
- *Using SYMCLI to Perform SRDF/AR* (P/N 300-000-078)

Practical Uses

When databases become very large (larger than one terabyte), the time and resources required for host-based software to protect or run decision-support queries on these databases becomes a critical factor. The time required to quiesce or shut down the database for offline operations is no longer acceptable for most database applications. When you need to perform such database operations, the use of TimeFinder consistent split allows you to split off a consistent, DBMS-restartable copy of the database in seconds with no interruption to online service.

TimeFinder software with PowerPath® support provides a consistent-split implementation of instant split that allows you to split off a consistent, DBMS-restartable BCV copy of your database without having to quiesce or shut down the database first. The impact on your production environment is negligible. Also, beginning with EMC Solutions Enabler version 5.1, TimeFinder can use Enginuity Consistency Assist (ECA) to perform consistent split across multiple, heterogeneous hosts *without* PowerPath support.

Consistent split can also play a role in SRDF/AR environments that allow you to set up remote mirroring to be automatic according to your own predefined copy cycle (every hour, for example). You can use a single-hop disaster restart configuration that permits controlled data loss (at most, one copy cycle). If greater protection is required, a multi-hop restart configuration can provide synchronous disaster restart with zero data loss (that is, all completed transactions on the source are completed on the target). Using TimeFinder consistent split within a `symreplicate` session to copy data to remote sites provides consistency and automation benefits for successful restart solutions.

For more information about SRDF/AR, refer to the white paper *Using SYMCLI to Perform SRDF/AR* (P/N 300-000-078).

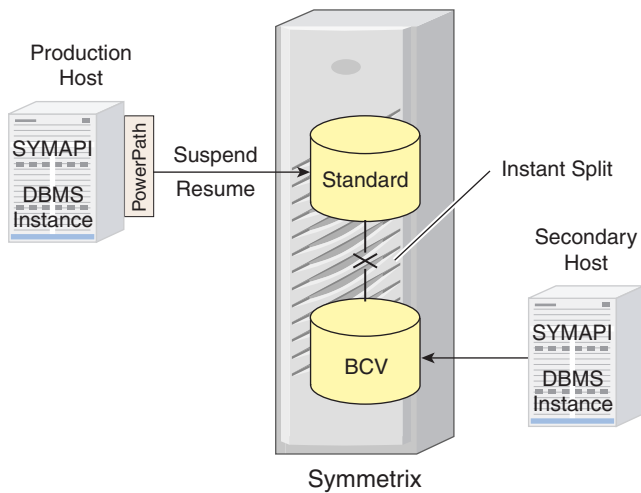
Consistent Split

Instant split reduces the wait period during a split to a short time by decomposing the splitting process into two phases: a *foreground split* and a *background split*. During an instant split, the system executes the foreground split almost instantaneously and returns a successful status to the host. This action allows I/O operations to the standard and BCV to continue immediately. At the same time, the background split continues to split the BCV pair until the split is complete.

Note: Beginning with the TimeFinder component of EMC Solutions Enabler version 5.0 running on Symmetrix units using Enginuity version 5568, all splits are instant splits. If you include the `-instant` option with the command `symmir split`, SYMCLI returns immediately after the foreground split. Omitting the `-instant` option causes SYMCLI to wait until the background split completes before returning.

Consistent split is an implementation of instant split that avoids inconsistencies and restart problems that can occur if you split a database-related BCV without first quiescing the database. The difference between a normal instant split and a consistent instant split is the addition of PowerPath or Enginuity Consistency Assist (ECA) to suspend the database device writes at the host level for a very short time while the foreground split occurs.

Figure 1 illustrates the use of local consistent split with a database management system (DBMS).



CLI-000112

Figure 1. Consistent Split on a Local Symmetrix

When you issue a `symmir split` command with an ECA- or PowerPath-related option from the production host, I/O is temporarily suspended during the following sequence:

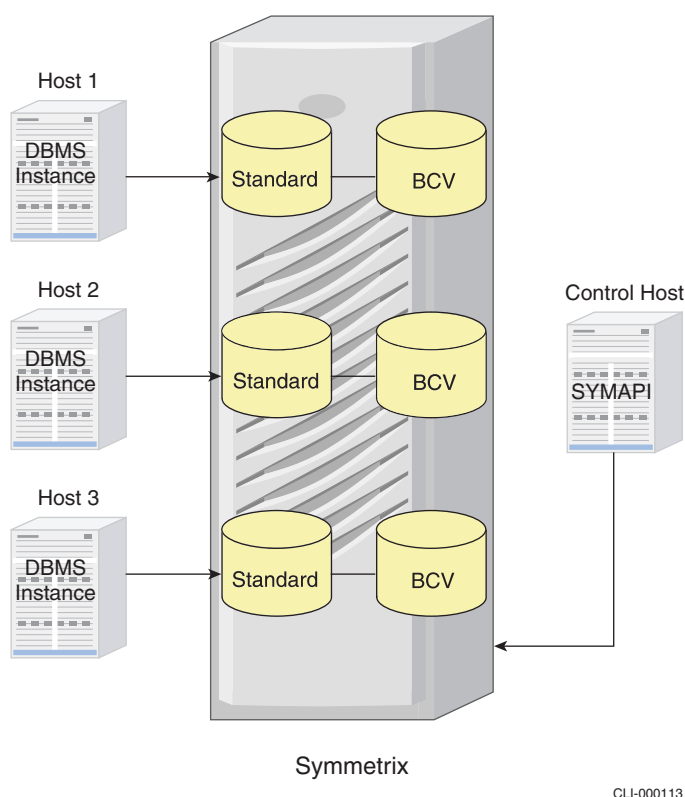
1. The Symmetrix API (SYMAPI) identifies the standard devices that hold the database.
2. SYMAPI validates that all identified BCV pairs can be split.
3. SYMAPI sends a “suspend I/O” log message to PowerPath or ECA.
4. PowerPath or ECA suspends I/O to the standard devices that hold the database.
5. The DBMS cannot write to the devices and subsequently waits for these devices to become available before resuming any further I/O. The device I/O is usually suspended for less than one second.¹
6. SYMAPI sends an instant split request to all BCV pairs in the specified device group and waits for the Symmetrix to acknowledge that the foreground split has occurred.
7. SYMAPI sends a “resume I/O” message to PowerPath or ECA.
8. The DBMS resumes writing to the standard devices that hold the database.

The BCV devices now contain a DBMS-restartable copy of the database that is consistent with the production database up until the time of the instant split. The production DBMS instance (on the production host in Figure 1) has no knowledge that the split or suspend/resume operation occurred. When a DBMS instance on the secondary host is started against the split BCVs, there is no record of a successful shutdown or backup. Therefore, the secondary DBMS instance views the BCV database copy as if it had crashed and proceeds to perform the normal crash recovery sequence in starting the database copy.

Before performing a consistent split, it is a good practice to issue a checkpoint from the production database to flush dirty pages out to disk and minimize recovery on the target database.

You can use ECA to perform consistent splits on BCV pairs across multiple, heterogeneous hosts *without* PowerPath support. To do so, however, requires that you have either a control host (with no database) or a database host with a dedicated channel from which to perform the control. Figure 2 shows a control host that can perform ECA consistent splits for three database hosts that access devices on the Symmetrix unit.

¹ At the time I/O is suspended, the 30-second Engenuity protection timer begins. If some host or I/O channel problem prevents the split operation from occurring during the 30-second window, a message notifies you that the consistency window was closed on some devices before the operation completed. In this case, the split is not consistent. You should re-establish the BCV pairs and attempt to repeat the consistent split later.



CLI-000113

Figure 2. Performing an ECA Consistent Split across Multiple, Heterogeneous Hosts

The nature of the ECA consistent split depends on the device groups that you define on the control host. For example, if you define a device group that includes all the devices being accessed by the three database hosts, then you can split all BCV pairs related to those hosts at the same time. If you define a device group that includes just the devices being accessed by Host 1, then you can split those BCV pairs related to Host 1 without affecting the other hosts. Similarly, you can define a device group on the control host for the Host 2 devices and another device group for the Host 3 devices.

Table 1 lists differences between the ECA `-consistent` option and the PowerPath `-ppath` option.

Table 1. Performing Consistent Splits Using ECA `-consistent` or PowerPath `-ppath`

	<code>-consistent</code>	<code>-ppath</code>
Minimum Enginuity support version	5x67	5x66
Minimum Solutions Enabler support version	5.1	4.3.1 ²
Multi-host support	Yes	No
Available on operating systems where PowerPath does not run ³	Yes	No
Can be run from a host not running the DBMS	Yes	No
Requires a control host (with no database) or a host with a dedicated channel	Yes	No

² With PowerPath version 2.0 software (or higher).

³ For currently supported operating systems, refer to the *EMC Solutions Enabler Support Matrix*.

Creating a Local DBMS-Restartable BCV Copy

The following general setup⁴ for creating consistent BCV copies of a database locally is similar to setting up for other TimeFinder operations:

1. Create a Regular type device group.

```
symmdg create ProdBgrp -type regular
```
2. Add to the device group those standard devices that hold the database (PowerPath-connected devices if you are using PowerPath support).

```
symld -g ProdBgrp addall dev -range 286:28B
```
3. Associate with the device group those BCV devices that will hold the restartable copy of the database.

```
symbcv -g ProdBgrp associateall dev -range 39A:39F
```
4. Fully establish all BCV pairs in the device group.

```
symmir -g ProdBgrp establish -full -noprompt
```

When the BCV pairs of a device group (ProdBGrp, for example) are synchronized, perform the consistent split using one of several options to the `symmir split` command:

- By providing a database type (`oracle`, `sqlserver`, `ibmudb`, or `informix`). This operation requires setting the appropriate environment variables first to allow SYMCLI to log in to the database:

```
symmir -g ProdBgrp split -instant -rdb -dbtype oracle
```

SYMAPI queries the Oracle database manager for the location of those devices that hold the database. PowerPath or ECA suspends I/O to the devices identified by the Oracle database manager. This option may be expensive in terms of the time required to discover the devices.

- By providing the physical device names of PowerPath-connected standard devices. In this case, you are responsible for specifying the correct list of standard devices that hold the database. For example, to specify PowerPath devices whose physical device names are `clt0d0`, `clt0d1`, and `clt0d2`:

```
symmir -g ProdBgrp split -instant -ppath /dev/rdisk/clt0d0 \
/dev/rdisk/clt0d1 /dev/rdisk/clt0d2
```

When the devices that hold the database are the only standard devices in the device group, it is simpler to substitute the `stddevs` keyword for the list of device names. For example:

```
symmir -g ProdBgrp split -instant -ppath stddevs
```

- Beginning with SYMCLI version 5.1, by providing the `-consistent` option to perform a consistent split on all BCV pairs in the device group. For example:

```
symmir -g ProdBgrp split -consistent
```

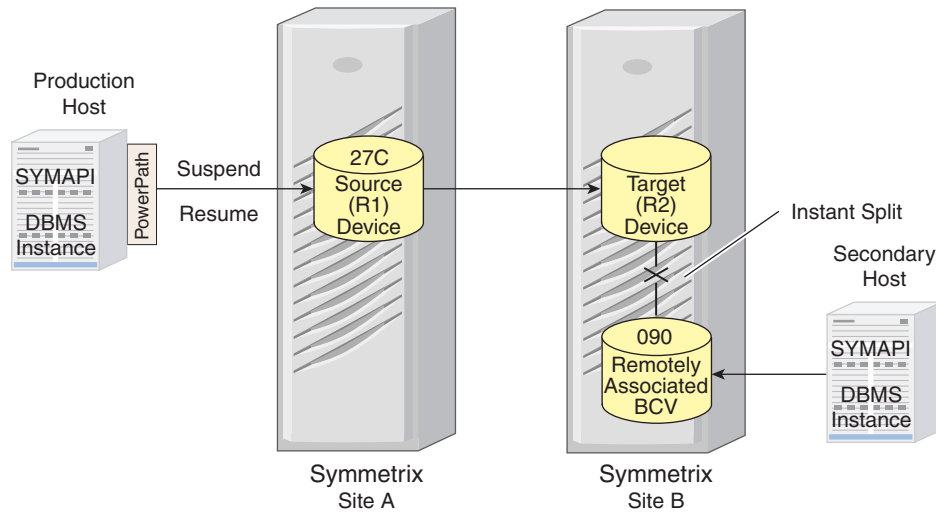
- By providing a list of VxFS file system names (such as `ora1`, `ora2`, `ora3`). This operation suspends I/O to these three file systems during the instant split operation but does not involve PowerPath:

```
symmir -g ProdBgrp split -instant -vxfs /ora1 /ora2 /ora3
```

⁴ An alternative method of creating this device group would be to use `symrdb db2dg` to automatically discover the standard devices that hold the database. However, you would still need to add the BCVs to the device group as described in step 3.

Creating a DBMS-Restartable BCV Copy on a Remote Symmetrix

Using SRDF technology and the TimeFinder RDF flag (the `-rdf` option), you can establish a BCV pair on the remote Symmetrix unit (Site B in Figure 3). By doing this, the remotely associated BCV is synchronized with the target (R2) device until such time that you may decide to perform a consistent split to create a DBMS-restartable database copy on the BCVs of a remote Symmetrix unit.



CLI-000114

Figure 3. Consistent Split on a Remote Symmetrix

To perform a consistent split successfully in an SRDF environment, you must run SRDF with Synchronous replication. The following steps perform a consistent split on a remote BCV pair:

1. Create an RDF1 type device group. For example:

```
symdmg create Rdf1Grp -type rdf1
```
2. Add to the device group those R1 standard devices that hold the database (PowerPath-connected devices if using PowerPath support).

```
symld -g Rdf1Grp add dev 27C
```
3. Determine which BCV devices on the remote Symmetrix unit (sid 505, for example) are not already part of a device group and are free to be added to your device group:

```
symdev list -sid 505 -bcv
```
4. Remotely associate with the device group BCV devices that will hold the restartable database copy.

```
symbcv -g Rdf1Grp associate dev 090 -rdf
```
5. Fully establish all BCV pairs in the device group.

```
symmir -g Rdf1Grp establish -full -rdf
```
6. Perform a consistent split of all PowerPath-connected standard devices with their remotely associated BCVs:

```
symmir -g Rdf1Grp split -rdf -instant -ppath stddevs
```

Creating DBMS-Restartable Copies on Both Sides

Figure 4 illustrates how you can copy the same data to BCV devices on two Symmetrix units. By performing a consistent split operation from the production host using the `-both_sides` option, you can split the BCV pairs on both sides at the same time. If the R1 source and R2 target devices are paired with concurrent BCVs, the consistent split is applied to four BCVs. The BCVs on both sides then contain data that is consistent with the R1 source data up to the time of the split and are accessible to their hosts.

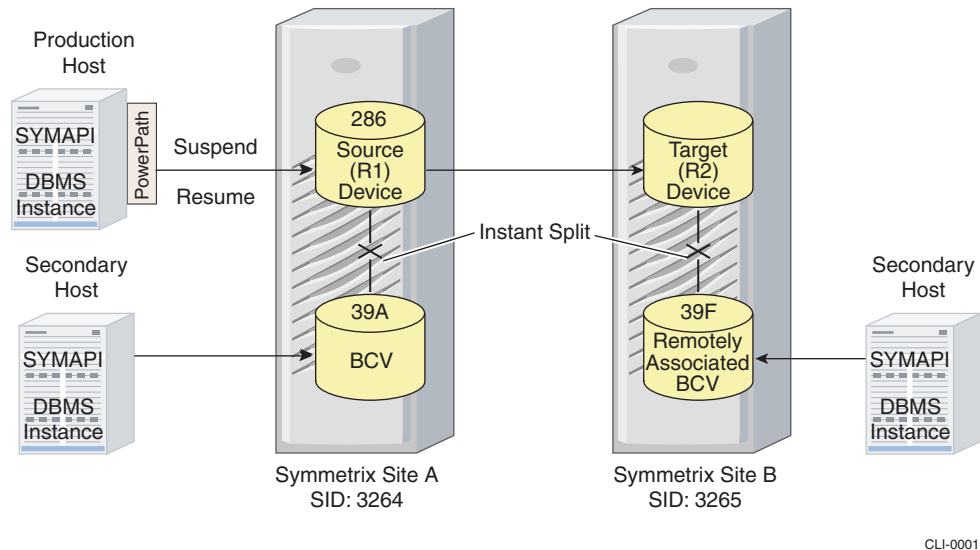


Figure 4. Copying the Same Data on Local and Remote Symmetrix Units

The following steps outline an example of copying the same data to BCVs on two Symmetrix units:

1. Create an RDF1 type device group.


```
symdbg create Rdf1Grp -type rdf1
```
2. Add to the device group the R1 standard device (286) on the local Symmetrix (sid 3264) that will hold the production data (a PowerPath-connected device if using PowerPath support).


```
symld -g Rdf1Grp -sid 3264 add dev 286
```
3. Associate with the device group the BCVs on both Symmetrix units.


```
symbcv -g Rdf1Grp -sid 3264 associate dev 39A
symbcv -g Rdf1Grp -sid 3265 associate dev 39F -rdf
```
4. Establish the BCV pairs on both Symmetrix units. DEV001 is the logical name of the PowerPath device (286). The first command establishes the local BCV pair at Site A. The second command using the `-rdf` option establishes the remotely associated BCV at Site B with the R2 standard device there.

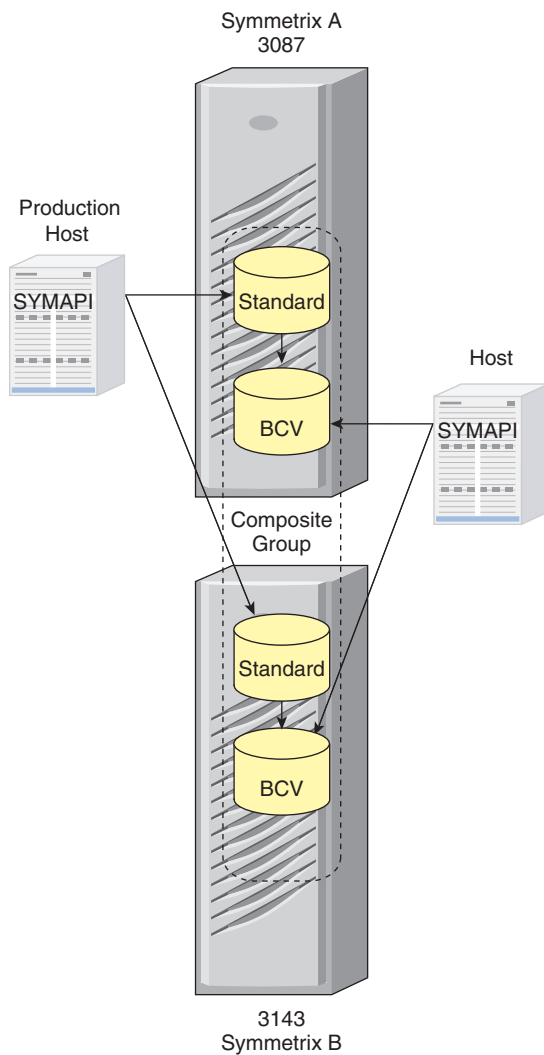

```
symmir -g Rdf1Grp establish -full DEV001 -noprompt
symmir -g Rdf1Grp establish -full DEV001 -rdf -noprompt
```
5. When the BCV pairs are fully synchronized, perform a consistent split on both BCV pairs. Specify the `-ppath` option with either the `stddevs` keyword (all PowerPath devices in the device group) or a PowerPath device name (for example, `/dev/rsk/emcpower1c`).


```
symmir -g Rdf1Grp split -instant -both_sides -ppath stddevs
```

Using a Composite Group to Split a Set of Device Pairs That Spans Multiple Symmetrix Units

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

Figure 5 illustrates a production host that is locally connected to two Symmetrix units (A and B). A composite group is defined on the production host and includes BCV pairs from each Symmetrix unit. Another locally connected host allows you to access the BCVs once the BCV pairs are split.



CLI-000081

Figure 5. Using a Composite Group When a Set of Devices Spans Two Symmetrix Units

Although TimeFinder control operations on BCV pairs might normally be performed from the production host (as shown in Figure 5) because the composite group is defined there in its SYMAPI database, there are methods that would allow you to initiate copy sessions from another locally connected host. One way is to copy the composite group definition to another host. A more efficient method is to enable Group Name Services (GNS), which automatically propagates the composite group definition to the Symmetrix units and

other locally attached hosts that are running the GNS daemon. For more information, refer to the white paper *Using SYMCLI and GNS to Propagate Group Definitions to Multiple Hosts* (P/N 300-301-384).

Prior to Solutions Enabler version 6.0, composite group support did not allow you to copy from a single source standard device to more than one BCV (that is, not to concurrent BCVs). Consequently, you had to add an equal number of standard devices and BCV devices to the composite group. Solutions Enabler version 6.0 lifts this restriction by allowing you to create each BCV pair explicitly, including concurrent BCVs.

If you do not create each BCV pair explicitly, certain options such as `-opt`, `opt_rag`, and `-exact` allow you to control how multiple devices in a composite group are paired. Otherwise, a device-pairing algorithm checks if there were any previous pair assignments among the devices and, if not, pairs standards and BCVs of equal sizes.

Used only with a full establish operation, the optimize option that you choose depends on whether you are establishing local BCV pairs or remote BCV pairs. The `-opt` option is for local. It optimizes pairings across the local Symmetrix array without regard for whether the devices belong to different RDF (RA) groups. The `-opt_rag` option is for remote and requires the `-rdf` option. It uses optimization rules to create remote BCV pairs from devices within the same RDF (RA) group on a Symmetrix array.

The following steps outline the setup required for performing an ECA consistent split on a set of BCV pairs that spans two Symmetrix units as shown in Figure 5:

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

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

```
symcgs -cg MyGrp -sid 3087 add dev 0076
symcgs -cg MyGrp -sid 3143 add dev 0091
```
3. Add a BCV device from each Symmetrix unit to the composite group.

```
symbcv -cg MyGrp -sid 3087 associate dev 0051
symbcv -cg MyGrp -sid 3143 associate dev 004F
```
4. Create the BCV pairs and initiate full copying from the standards to the BCVs.

```
symmir -cg MyGrp establish -full
```
5. When the BCV pairs are fully synchronized, you can perform an ECA consistent split on all BCV pairs in the composite group.

```
symmmir -cg MyGrp split -consistent
```

For more details about splitting a composite group of BCV pairs, refer to “Example 3: ECA Consistent Split of a Composite Group.”

For more information about splitting a composite group of snap pairs or clone pairs, refer to the white papers *Using SYMCLI to Perform TimeFinder/Snap Operations* (P/N 300-001-381) and *Using SYMCLI to Perform TimeFinder/Clone Operations* (P/N 300-001-382).

Beginning with Solutions Enabler version 6.0, you can control specific BCV pairs within the composite group instead of having to control the group as a whole. For example, to establish only the DEV001/BCV001 pair from all devices in the composite group MyGrp:

```
symmir -cg MyGrp establish DEV001 bcv 1d BCV001
```

Performing a Split with Database Assist

In database environments where more than one BCV pair is being split at the same moment, typical TimeFinder instant split operations are performed in an unpredictable order because of the unpredictable nature of I/O load on each device. If you split a database without quiescing it first, it is possible to end up with BCV devices holding user data that is more recent than BCV devices holding log data.

Performing splits without PowerPath or ECA is similar for most supported databases. If you do not have PowerPath or EMC Solutions Enabler version 5.1 (or higher) installed, you can use the `symioctl` command to place your database in backup mode (for Oracle) or momentarily freeze the database (such as Informix, SQL Server, etc.), suspending any database updates being written to the disk for the time it takes for the foreground split to occur. When the foreground split completes, use the `symioctl` command again to resume access to the database. Keep in mind, however, that the consistent split described in previous sections circumvents this operation and is a cleaner and safer split on the BCV database environment.

For SYMCLI to access a database, set the `SYMCLI_RDB_CONNECT` environment variable and those database application variables that are relevant for your database system. The following example sets the `SYMCLI_RDB_CONNECT` variable to a username of “system” and a password of “manager.” (The Bourne and Korn shells use the `export` command to set environment variables; the C shell uses the `setenv` command.)

```
export SYMCLI_RDB_CONNECT=system/manager
```

Before beginning any SYMCLI database command session, set the appropriate database application variables on the host where the database instance (or server) is running (refer to Table 2). For example, setting `ORACLE_HOME` specifies the location of the Oracle binaries. Setting `ORACLE_SID` specifies the database instance name:

```
export ORACLE_HOME=/disks/symapidvt/oraclehome/ap179
export ORACLE_SID=ap179
```

Table 2. Defining Environment Variables for Database Applications

For Database	Set Variables
Oracle	<code>ORACLE_HOME</code> and <code>ORACLE_SID</code> .
Informix Dynamic Server	<code>INFORMIXDIR</code> , <code>ONCONFIG</code> , and <code>INFORMIXSERVER</code> .
Sybase	<code>SYBASE</code> and <code>DSQUERY</code> . If the <code>OCS-12_0</code> directory is not located in the <code>\$SYBASE</code> directory, set <code>LD_LIBRARY_PATH</code> , <code>LIBPATH</code> , or <code>SHLIB_PATH</code> , depending on whether you are running Solaris, AIX, or HP-UX.
IBM DB2/UDB	<code>DB2INSTANCE</code> . Set <code>LD_LIBRARY_PATH</code> , <code>LIBPATH</code> , or <code>SHLIB_PATH</code> , depending on whether you are running Solaris, AIX, or HP-UX.
SQL Server; Exchange	None required.

SQL Server and Exchange do not have variables that specify a server instance. Beginning with EMC Solutions Enabler version 5.1, SYMCLI supports two or more named SQL Server instances running on the same Windows 2000 box. To connect to an SQL server instance, add a service name (an ODBC system DSN) to the `RDB_CONNECT` variable:

```
set SYMCLI_RDB_CONNECT=username/password@service
```

To create this service name (ODBC system DSN), click on Control Panel, Administrative Tools, Data Sources (ODBC), and the System DSN tab. Create an ODBC system DSN. When prompted to choose a data source from a list of sources, choose “SQL Server.” When prompted for the named instance of SQL Server, enter the instance name.

Freezing an Oracle Database

The following commands place the Oracle database in hot backup mode, freeze the database, perform an instant split on all BCV pairs in a device group named ProdBgrp, thaw the database, and end backup mode.

```
symioctl begin backup -type oracle
symioctl freeze -type oracle
symmir -g ProdBgrp split -instant -noprompt
symioctl thaw -type oracle
symioctl end backup -type oracle
```

This operation on an Oracle database does not directly affect metadata I/O. The operation does not affect any Oracle files that are not related to this database instance or any non-Oracle files in the same file system. Consequently, freeze-and-thaw alone does not guarantee Oracle database integrity as consistent split does. However, this operation does enhance the performance of splits if you are unable to use Enterprise TimeFinder Consistent Split.

Freezing an Informix Database

The following commands freeze an Informix database (`-type is informix`), perform an instant split on all BCV pairs in a device group named ProdBgrp, and thaw the database.

```
symioctl freeze -type informix
symmir -g ProdBgrp split -instant -noprompt
symioctl thaw -type informix
```

Once the foreground split completes (usually a fraction of a second), the thaw action resumes full I/O access to the database.

Freezing an SQL 2000 Database

Snapshot commands for SQL Server 2000 and higher support BACKUP and RESTORE database WITH SNAPSHOT operations using the Virtual Device Interface (VDI). The `symioctl begin snapshot` command sends a “BACKUP DATABASE TO VIRTUAL_DEVICE WITH SNAPSHOT” SQL command to SQL Server, beginning the snapshot backup and suspending writes for the specified database.

The `symioctl begin snapshot` command suspends writes to a database named mydb and creates a savefile named mydb.sav. Then an instant split is performed on BCV pairs in the device group.

```
symioctl begin snapshot -type sqlserver mydb savefile c:\temp\mydb.sav
symmir -g ProdBgrp split -instant -noprompt
symioctl end snapshot -type sqlserver mydb
```

The `symioctl end snapshot` command causes the “BACKUP DATABASE” SQL command for the mydb database to complete. Writes to the database can resume, and the snapshot metadata is saved to the savefile. The metadata from the savefile is used by SQL Server to logically restore the database if needed (by using a subsequent `symioctl restore snapshot` command). For example:

```
symioctl restore snapshot -type sqlserver mydb savefile c:\temp\mydb.sav
```

Freezing an IBM UDB Database

The following commands freeze an IBM UDB database (`-type is ibmudb`), perform an instant split on all BCV pairs in a device group named ProdBgrp, and thaw the database.

```
symioctl suspend -type ibmudb
symmir -g ProdBgrp split -instant -noprompt
symioctl resume -type ibmudb
```

Freezing a Sybase Database

The following sequence freezes a Sybase database (`-type is sybase`), performs an instant split on all BCV pairs in a device group named ProdBgrp, and thaws the database.

```
symioctl suspend -type sybase
symmir -g ProdBgrp split -instant -noprompt
symioctl resume -type sybase
```

Example 1: Creating an RDBMS-Restartable Copy on a Local Symmetrix

The hardware setup for this example consists of an HP-UX production host (api183) and an HP-UX secondary host (api213). Both hosts are connected to a local Symmetrix (sid 814). PowerPath 2.1 is installed on api183. Oracle 8.1.7.0.0 is installed on api183 and api213. The example uses PowerPath devices 286–28B and BCV devices 39A–39F. An Oracle database was created on the production host. All Oracle objects (data files, control files and redo logs) must be on the PowerPath devices.

The example demonstrates consistent split operations both with and without PowerPath support.

The `sympd list` command issued from the production host displays a list of PowerPath devices on this Symmetrix that are visible to the host. The display below shows a portion of this list (PowerPath standard devices 286 through 28B). The ellipsis (...) represents truncated output.

Note that each PowerPath standard device has two SA (SCSI adapter) connections and physical device names (parent and child) that provide alternate paths for load-balancing and failover capabilities.

```
api183 # sympd list -powerpath
```

```
Symmetrix ID: 000185500814
```

P O W E R P A T H D E V I C E S							
Device Name	Directors			Device			
Physical	Sym	SA :P	DA :IT	Config	Attribute	Sts	Cap (MB)

	0286		02B:C5	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c0t1d2	- 13A:0		- -		-	-	-
/dev/rdisk/c1t1d2	- 04A:0		- -		-	-	-
	0287		01B:C5	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c0t1d3	- 13A:0		- -		-	-	-
/dev/rdisk/c1t1d3	- 04A:0		- -		-	-	-
	0288		16A:C5	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c0t1d4	- 13A:0		- -		-	-	-
/dev/rdisk/c1t1d4	- 04A:0		- -		-	-	-
	0289		15A:C5	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c0t1d5	- 13A:0		- -		-	-	-
/dev/rdisk/c1t1d5	- 04A:0		- -		-	-	-
	028A		02A:C5	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c0t1d6	- 13A:0		- -		-	-	-
/dev/rdisk/c1t1d6	- 04A:0		- -		-	-	-
	028B		01A:C5	2-Way Mir	N/Grp'd	RW	4315
/dev/rdisk/c0t1d7	- 13A:0		- -		-	-	-
/dev/rdisk/c1t1d7	- 04A:0		- -		-	-	-
.....							

The `sympd list` command issued from the secondary host (api213) displays all devices on the Symmetrix that are visible to the host. This partial list shows some devices that are configured as BCV devices. The example will use these BCVs to hold the DBMS-restartable copy of the database. To display just a list of BCVs on a Symmetrix, regardless of whether they are host-visible, use the `symdev list` command with the `-bcv` option.

```
api213 # sympd list
```

Device Name	Directors			Device			
-----	-----	-----	-----	-----	-----	-----	-----
Physical	Sym	SA	:P DA :IT	Config	Attribute	Sts	Cap (MB)
-----	-----	-----	-----	-----	-----	-----	-----
/dev/rdisk/c0t0d0	039A	13B:0	16B:D5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d1	039B	13B:0	01A:D5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d2	039C	13B:0	16B:C5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d3	039D	13B:0	15B:C5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d4	039E	13B:0	02B:C5	BCV	N/Asst'd	RW	4315
/dev/rdisk/c0t0d5	039F	13B:0	01B:C5	BCV	N/Asst'd	RW	4315

Creating a device group and adding devices to it are prerequisites for performing the following operations. The `symdg create` command issued from the production host creates a device group named `oracle` on this host. The `symlld addall` command adds the PowerPath standard devices to the group, using the command's `-range` option to limit the selections to those devices between 286 and 28B. The `symbcv` command associates the BCV devices with the device group; the `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
api183 # symdg create oracle
```

```
api183 # symlld -g oracle addall dev -range 286:28B
```

```
api183 # symbcv -g oracle associateall dev -range 39A:39F
```

The `symmir establish` command initiates a full establish operation on all BCV pairs in the device group.

```
api183 # symmir -g oracle establish -full -noprompt
```

```
'Full Establish' operation execution is in progress for device group 'oracle'.
Please wait...
```

```
'Full Establish' operation successfully initiated for device group 'oracle'.
```

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends. The BCV pairs must be fully synchronized before you attempt a consistent split.

```
api183 # symmir -g oracle verify -i 30
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

When you issue the `symmir split` command with the `-rdb -dbtype` option and the database type is Oracle, SYMCLI utilizes the SYMAPI mapping features to automatically determine which PowerPath devices need to be frozen. SYMCLI logs into Oracle and queries Oracle's catalog. For SYMCLI to access Oracle, you need to set the environment variable `SYMCLI_RDB_CONNECT` to the username and password of the Oracle system administrator's account. The following command sets the variable to a username of "system" and a password of "manager".

```
api183 # export SYMCLI_RDB_CONNECT=system/manager
```

The `export ORACLE_HOME` command specifies the location of the Oracle binaries. The `export ORACLE_SID` command specifies the database instance name.

```
api183 # export ORACLE_HOME=/disks/symapidvt/oraclehome/api183  
api183 # export ORACLE_SID=api183
```

The following command performs a consistent split using the `-rdb` option and specifies an Oracle database type. However, as the output from this command illustrates, the mapping facilities of SYMAPI determined that Oracle objects exist on only three of the six PowerPath devices in the device group. All six BCV pairs are split, but PowerPath freezes and thaws only those devices that contain the Oracle database (that is, the Oracle data files, control files and redo logs).

```
api183 # symmir -g oracle split -instant -rdb -dbtype oracle -noprompt
```

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

Freezing 3 device(s).....Done.

Thawing 3 device(s).....Done.

'Split' operation successfully executed for device group 'oracle'.

A `symmir verify` command checks status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

The `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are currently in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following command performs a consistent split using the `-ppath` option and specifies the `stddevs` keyword to indicate that this operation applies to *all* PowerPath standard devices in the device group.

```
api183 # symmir -g oracle split -instant -ppath stddevs -noprompt
```

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

Freezing 6 device(s).....Done.

Thawing 6 device(s).....Done.

'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

Another `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following command performs a consistent split using the `-ppath` option and specifies explicitly the parent physical device name for each PowerPath device in the device group. However, EMC recommends using the `stddevs` keyword (as in the previous example) to specify all PowerPath standard devices in a device group.

```
api183 # symmir -g oracle split -instant -noprompt -ppath /dev/rdsd/c0t1d2 \  
         /dev/rdsd/c0t1d3 /dev/rdsd/c0t1d4 /dev/rdsd/c0t1d5 /dev/rdsd/c0t1d6 \  
         /dev/rdsd/c0t1d7
```

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

Freezing 6 device(s).....Done.

Thawing 6 device(s).....Done.

'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

Another `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are currently in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following command performs a consistent split using ECA (the `-consistent` option). This operation applies to all BCV pairs in the device group. When using ECA, you do not need PowerPath support. With ECA, the devices in the device group do not need to be PowerPath devices.

```
api183 # symmir -g oracle split -consistent -noprompt
```

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

'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

Another `symmir establish` command re-establishes incrementally the six BCV pairs in the device group that are in the Split state.

```
api183 # symmir -g oracle establish -noprompt
```

'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the BCV pairs in the device group every 30 seconds until the BCV pairs are synchronized. Then the verify loop ends.

```
api183 # symmir -g oracle verify -i 30
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following command performs a consistent split using the `-vxfs` option and specifies the names of four VERITAS file systems. When using VERITAS file systems, PowerPath is not required. The freeze and thaw tasks are carried out by the VERITAS file system.

```
api183 # symmir -g oracle split -instant -noprompt -vxfs /ora1 /ora2 /ora3 /ora4
```

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

Freezing 4 filesystem(s).....Done.

Thawing 4 filesystem(s).....Done.

'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split every 10 seconds until it completes.

```
api183 # symmir -g oracle verify -split -bg -i 10
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

Example 2: Creating an RDBMS-Restartable BCV Copy on a Remote Symmetrix Using PowerPath Support

The hardware setup for this example consists of a Solaris production host (api179) and a Solaris secondary host (api180). The production host is connected to a local Symmetrix (sid 814); the secondary host is connected to a remote Symmetrix (sid 505). PowerPath 2.1 is installed on api179. Oracle 8.1.7.0.0 is installed on api179 and api180. The example uses local PowerPath standard devices 3C2–3C6 and remotely associated BCV devices 38A–38E.

An Oracle database was created on the production host. All Oracle objects (data files, control files and redo logs) must be on the PowerPath devices.

To perform a consistent split in an SRDF environment, you must run SRDF with Synchronous replication.

All commands in this example are issued from the production host (api179).

The `symcfg list` command displays configuration information on the two Symmetrix units.

```
api179# symcfg list
```

S Y M M E T R I X						
SymmID	Attachment	Model	Mcode Version	Cache Size (MB)	Num Phys Devices	Num Symm Devices
000185500814	Local	8530	5568	4088	450	1052
000185500505	Remote	8530	5568	8184	0	950

The `sympd list` command with the `-powerpath` option displays a list of PowerPath standard devices on the local Symmetrix that are visible to the host. The display below shows a portion of this list (Symmetrix devices 3C2 through 3C6). The ellipsis (...) represents truncated output.

```
api179# sympd list -powerpath
```

```
Symmetrix ID: 000185500814
```

P O W E R P A T H D E V I C E S						
Device Name	Directors			Device		
Physical	Sym	SA :P	DA :IT	Config	Attribute	Cap (MB)
	03C2		16B:D5	RDF1	N/Grp'd	RW 4315
/dev/rdsk/c4t0d102s2	- 05B:0		- -		-	-
/dev/rdsk/c5t0d102s2	- 12B:0		- -		-	-
	03C3		15B:D5	RDF1	N/Grp'd	RW 4315
/dev/rdsk/c4t0d103s2	- 05B:0		- -		-	-
/dev/rdsk/c5t0d103s2	- 12B:0		- -		-	-

	03C4	02B:D5	RDF1	N/Grp'd	RW	4315
/dev/rdisk/c4t0d104s2	- 05B:0	- -	-	-	-	-
/dev/rdisk/c5t0d104s2	- 12B:0	- -	-	-	-	-
	03C5	01B:D5	RDF1	N/Grp'd	RW	4315
/dev/rdisk/c4t0d105s2	- 05B:0	- -	-	-	-	-
/dev/rdisk/c5t0d105s2	- 12B:0	- -	-	-	-	-
	03C6	16A:D5	RDF1	N/Grp'd	RW	4315
/dev/rdisk/c4t0d106s2	- 05B:0	- -	-	-	-	-
/dev/rdisk/c5t0d106s2	- 12B:0	- -	-	-	-	-

The `symdev list` command with the `-bcv` option displays those BCV devices on the remote Symmetrix (sid 505) that are not already part of a device group (N/Asst'd) and are free to be added to your device group. The display below has been edited to show only those devices that will be used in the example. Although these BCVs are not visible (accessible) to the production host, whose point of view is displayed, they are visible to the secondary host connected to the remote Symmetrix unit. The secondary host can access these BCVs and start the DBMS-restartable copy of the database that these BCVs will hold.

```
api179# symdev list -bcv -sid 505
```

Symmetrix ID: 000185500505

Device Name		Directors			Device		
Sym	Physical	SA	:P	DA :IT	Config	Attribute	Sts
							Cap (MB)
038A	Not Visible	???:?		02B:C5	BCV	N/Asst'd	RW 4315
038B	Not Visible	???:?		01B:C5	BCV	N/Asst'd	RW 4315
038C	Not Visible	???:?		16A:C5	BCV	N/Asst'd	RW 4315
038D	Not Visible	???:?		15A:C5	BCV	N/Asst'd	RW 4315
038E	Not Visible	???:?		02A:D4	BCV	N/Asst'd	RW 4315

Creating a device group and adding devices to it are prerequisites for performing the following operations. The `symdmg create` command issued from the production host creates an RDF1 type device group named `oracle` on this host. The `symld addall` command adds the PowerPath standard devices to the group, using the `-range` option to limit the selections to those devices between 286 and 28B. The `symbcv` command with the `-rdf` option associates the remote BCV devices with the device group; the command's `-range` option is used with the `associateall` action to limit the selection to those BCVs that are within the specified range.

```
api179 # symdmg create oracle -type rdf1
api179 # symld -g oracle addall dev -range 3C2:3C6
api179 # symbcv -g oracle associateall dev -range 38A:38E -rdf
```

The `symrdf establish` command initiates a full establish operation on all SRDF pairs in the device group. The local PowerPath standard (R1) devices will become synchronized with their respective target (R2) devices located on the remote Symmetrix.

```
api179# symrdf -g oracle establish -full -noprompt
```

An RDF 'Full Establish' operation execution is in progress for device group 'oracle'. Please wait...

```
Suspend RDF link(s).....Done.
Mark target (R2) devices for full copy from source (R1)...Started.
Device: 03C2 ..... Marked.
Device: 03C3 ..... Marked.
Device: 03C4 ..... Marked.
Device: 03C5 ..... Marked.
Device: 03C6 ..... Marked.
Mark target (R2) devices for full copy from source (R1)...Done.
Merge device track tables between source and target.....Started.
Device: 03C2 ..... Merged.
Device: 03C3 ..... Merged.
Device: 03C4 ..... Merged.
Device: 03C5 ..... Merged.
Device: 03C6 ..... Merged.
Merge device track tables between source and target.....Done.
Read/Write Enable device(s) on SA at source (R1).....Done.
Resume RDF link(s).....Done.
```

The RDF 'Full Establish' operation successfully initiated for device group 'oracle'.

The `symrdf verify` command checks the state of the SRDF pairs in the device group every 30 seconds until the pairs are synchronized. Then the verify loop ends. The SRDF pairs must be fully synchronized before you attempt a consistent split.

```
api179# symrdf -g oracle verify -i 30
```

NONE of the mirrored pairs are in the 'Synchronized' state.

NOT all of the mirrored pairs are in the 'Synchronized' state.

All devices in the RDF group 'oracle' are in the 'Synchronized' state.

The `symmir establish` command with the `-rdf` option initiates a full establish operation on the BCV pairs of the remote Symmetrix unit (sid 505). The BCV devices on that Symmetrix will become synchronized with the appropriate R2 devices on that Symmetrix.

```
api179# symmir -g oracle establish -full -rdf -noprompt
```

Remote 'Full Establish' operation execution is in progress for device group 'oracle'. Please wait...

Remote 'Full Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command with the `-rdf` option checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized. The BCV pairs must be fully synchronized before you attempt a consistent split.

```
api179# symmir -g oracle verify -i 5 -rdf
```

None of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following `symmir` command performs a consistent split on the remote BCV pairs, using the command's `-ppath` option and specifying the `stddevs` keyword to indicate that this operation applies to *all* PowerPath standard devices in the device group.

```
api179# symmir -g oracle split -instant -rdf -ppath stddevs -noprompt
```

Remote 'Split' operation execution is in progress for device group 'oracle'.
Please wait...

Freezing 5 device(s).....Done.

Thawing 5 device(s).....Done.

Remote 'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command with the `-rdf` option checks the status of the background split (`-bg`) of the remote BCV pairs every 10 seconds until it completes.

```
api179 # symmir -g oracle verify -split -bg -i 10 -rdf
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

The `symmir establish` command re-establishes incrementally the five remote BCV pairs in the device group that are currently in the Split state.

```
api179# symmir -g oracle establish -rdf -noprompt
```

Remote 'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

Remote 'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized.

```
api179# symmir -g oracle verify -i 5 -rdf
```

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

Some of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

The following `symmir` command performs a consistent split on the remote BCV pairs, using the command's `-ppath` option and specifying explicitly the parent physical device name for each PowerPath device in the device group. However, EMC recommends using the `stddevs` keyword (as in the previous example) to specify all PowerPath standard devices in a device group.

```
api179# symmir -g oracle split -instant -rdf -noprompt -ppath \  
         /dev/rdisk/c4t0d102s2 /dev/rdisk/c4t0d103s2 /dev/rdisk/c4t0d104s2 \  
         /dev/rdisk/c4t0d105s2 /dev/rdisk/c4t0d106s2
```

Remote 'Split' operation execution is in progress for device group 'oracle'.
Please wait...

```
Freezing 5 device(s).....Done.
```

```
Thawing 5 device(s).....Done.
```

Remote 'Split' operation successfully executed for device group 'oracle'.

The `symmir verify` command checks the status of the background split (`-bg`) of the remote BCV pairs every 10 seconds until it completes.

```
api179 # symmir -g oracle verify -split -bg -i 10 -rdf
```

None of the devices in group 'oracle' have finished splitting in the background.

Not all devices in group 'oracle' have finished splitting in the background.

All devices in group 'oracle' have finished splitting in the background.

The `symmir establish` command re-establishes incrementally the five remote BCV pairs in the device group that are in the Split state.

```
api179# symmir -g oracle establish -rdf -noprompt
```

Remote 'Incremental Establish' operation execution is in progress for device group 'oracle'. Please wait...

Remote 'Incremental Establish' operation successfully initiated for device group 'oracle'.

The `symmir verify` command checks the state of the device group's remote BCV pairs every five seconds until the pairs are synchronized.

```
api179# symmir -g oracle verify -i 5 -rdf
```

Not all devices in group 'oracle' are in the 'Synchronized or Restored' state.

Some of the devices in group 'oracle' are in the 'Synchronized or Restored' state.

All devices in group 'oracle' are in the 'Synchronized or Restored' state.

Example 3: ECA Consistent Split of a Composite Group

This example was performed using Solutions Enabler version 5.4. The hardware setup includes a host connected to two source Symmetrix units (sid 35 and 60). A composite group defined on the host contains a set of BCV pairs that spans the two Symmetrix units. The devices include standard devices and BCV devices. However, this same functionality can be applied to TimeFinder snap pairs and clone pairs.

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

```
# symcg create TimeFinder -type regular
```

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

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

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

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

The `symmir establish` command creates optimized standard/BCV pairings among devices within each Symmetrix unit and performs a full establish operation on them.

```
# symmir -cg TimeFinder establish -full -optimize -noprompt
```

```
'Full Establish' operation execution is in progress for composite group
'TimeFinder'. Please wait...
```

```
'Full Establish' operation successfully initiated for composite group
'TimeFinder'.
```

The `symmir query` command shows synchronization is in progress (SyncInProg) for the BCV pairs. Devices 0056 and 0059 are meta devices (as are their paired BCV devices), and only the meta head device is displayed.

```
# symmir -cg TimeFinder query
```

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

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

Standard Device		BCV Device		State
Sym	Inv. Tracks	Sym	Att.	Inv. Tracks
0056	0 0182	*	344150	SyncInProg
0059	0 0185	*	228145	SyncInProg
005B	0 0187	*	120030	SyncInProg

005C	0 0188	*	120181	SyncInProgress
005D	0 0189	*	120878	SyncInProgress
005E	0 018A	*	110078	SyncInProgress
005F	0 018B	*	41523	SyncInProgress
0060	0 018C	*	33900	SyncInProgress
0061	0 018D	*	123432	SyncInProgress
0062	0 018E	*	17008	SyncInProgress
0063	0 018F	*	53389	SyncInProgress
0064	0 0190	*	42266	SyncInProgress
0065	0 0191	*	7995	SyncInProgress
0066	0 0192	*	37293	SyncInProgress
0067	0 0193	*	45200	SyncInProgress
0068	0 0194	*	44386	SyncInProgress
0069	0 0195	*	45943	SyncInProgress
006A	0 0196	*	55380	SyncInProgress
006B	0 0197	*	55525	SyncInProgress
006C	0 0198	*	43544	SyncInProgress
006D	0 0199	*	48643	SyncInProgress
006E	0 019A	*	13424	SyncInProgress

Symmetrix ID : 000184500160 (Microcode Version: 5568)

Standard Device		BCV Device		State	
Sym	Inv. Tracks	Sym	Att.	Inv. Tracks	STD <=> BCV
0014	0 03B6	*		131367	SyncInProgress
0015	0 03B7	*		120967	SyncInProgress
0016	0 03B8	*		114601	SyncInProgress
0017	0 03B9	*		101797	SyncInProgress
0018	0 03BA	*		122833	SyncInProgress
0019	0 03BB	*		117237	SyncInProgress
001A	0 03BC	*		115669	SyncInProgress
001B	0 03BD	*		121046	SyncInProgress
001C	0 03BF	*		132493	SyncInProgress
001D	0 03BE	*		125720	SyncInProgress
001E	0 03C0	*		113095	SyncInProgress
001F	0 03C1	*		103544	SyncInProgress
0020	0 03C2	*		124720	SyncInProgress
0021	0 03C3	*		109486	SyncInProgress
0022	0 03C4	*		109383	SyncInProgress
0023	0 03C5	*		122115	SyncInProgress
0024	0 03C6	*		133037	SyncInProgress
0025	0 03C7	*		110196	SyncInProgress
0026	0 03C8	*		118622	SyncInProgress
0027	0 03C9	*		118089	SyncInProgress
Total -----					
Track(s)	0			4118330	
MB(s)	0			128698	

Legend:

(*): The paired BCV device is associated with this composite group.
 (p): The paired BCV device was restored using the protect option.
 (a): All mirrors of this BCV were moved to the STD.

The `symmir split` command performs an ECA consistent split on all BCV pairs in the composite group.

```
# symmir -cg TimeFinder split -consistent -noprompt
```

'Split' operation execution is in progress for composite group 'TimeFinder'.
Please wait...

'Split' operation successfully executed for composite group 'TimeFinder'.

The `symmir query` command with the `-bg` and `-percent` options shows the percentage completion of the split that is currently occurring in the background.

```
# symmir -cg TimeFinder query -bg -percent
```

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

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

Standard Device		BCV Device		State		Split
Sym	Inv. Tracks	Sym	Att.	Inv. Tracks	STD <=> BCV	(%)
0056	0	0182	*	0	Split (bg)	14
0059	0	0185	*	0	Split (bg)	14
005B	0	0187	*	0	Split (bg)	15
005C	0	0188	*	0	Split (bg)	15
005D	0	0189	*	0	Split (bg)	15
005E	0	018A	*	0	Split (bg)	13
005F	0	018B	*	0	Split (bg)	15
0060	0	018C	*	0	Split (bg)	15
0061	0	018D	*	0	Split (bg)	15
0062	0	018E	*	0	Split (bg)	13
0063	0	018F	*	0	Split (bg)	15
0064	0	0190	*	0	Split (bg)	15
0065	0	0191	*	0	Split (bg)	15
0066	0	0192	*	0	Split (bg)	13
0067	0	0193	*	0	Split (bg)	15
0068	0	0194	*	0	Split (bg)	15
0069	0	0195	*	0	Split (bg)	15
006A	0	0196	*	0	Split (bg)	13
006B	0	0197	*	0	Split (bg)	15
006C	0	0198	*	0	Split (bg)	15
006D	0	0199	*	0	Split (bg)	15
006E	0	019A	*	0	Split (bg)	13

Symmetrix ID : 000184500160 (Microcode Version: 5568)

Standard Device		BCV Device		State		Split
Sym	Inv. Tracks	Sym	Att.	Inv. Tracks	STD <=> BCV	(%)
0014	0	03B6	*	0	Split (bg)	5
0015	0	03B7	*	0	Split (bg)	5
0016	0	03B8	*	0	Split (bg)	5
0017	0	03B9	*	0	Split (bg)	2
0018	0	03BA	*	0	Split (bg)	2
0019	0	03BB	*	0	Split (bg)	2
001A	0	03BC	*	0	Split (bg)	2
001B	0	03BD	*	0	Split (bg)	2
001C	0	03BF	*	0	Split (bg)	5
001D	0	03BE	*	0	Split (bg)	2
001E	0	03C0	*	0	Split (bg)	9
001F	0	03C1	*	0	Split (bg)	8
0020	0	03C2	*	0	Split (bg)	6
0021	0	03C3	*	0	Split (bg)	6
0022	0	03C4	*	0	Split (bg)	18
0023	0	03C5	*	0	Split (bg)	17
0024	0	03C6	*	0	Split (bg)	9
0025	0	03C7	*	0	Split (bg)	8
0026	0	03C8	*	0	Split (bg)	6
0027	0	03C9	*	0	Split (bg)	7
Total						
Track(s)	0			0		
MB(s)	0.0			0.0		

Legend:

(*): The paired BCV device is associated with this composite group.
 (p): The paired BCV device was restored using the protect option.
 (a): All mirrors of this BCV were moved to the STD.
 (bg): The paired BCV device is splitting in the background.

The `symmir verify` command checks the background split at 60-second intervals until the split operation on the composite group has completed.

```
# symmir -cg TimeFinder verify -bg -split -i 60
```

Not all devices in Composite Group 'TimeFinder' have finished splitting in the background.

Not all devices in Composite Group 'TimeFinder' have finished splitting in the background.

All devices in Composite Group 'TimeFinder' have finished splitting in the background.

Example 4: Performing Splits with Database Assist

Performing splits without PowerPath or ECA is similar for most supported databases in suspending writes to a database momentarily while an instant split occurs (see the section “Performing a Split with Database Assist”). The following examples show how to perform splits this way for an SQL 2000 database and an Oracle database.

Splitting BCV Pairs Using SQL 2000 Database Assist

This setup consists of a Windows 2000 host connected to a Symmetrix unit (sid 505). SQL Server 2000, Enterprise Edition, software is installed on the host. An SQL 2000 database called SQL2000a was created on Symmetrix devices. A device group called sqlserv was created beforehand. The devices that hold the database were added to this device group. The BCV pairs in the device group were fully established.

The `symmir query` command displays the configuration and status of the BCV pairs that hold the database. All BCV pairs are currently in the Synchronized state.

```
C:\> symmir -g sqlserv query
```

```
Device Group (DG) Name: sqlserv
DG's Type           : REGULAR
DG's Symmetrix ID    : 000185500505
```

Standard Device			BCV Device			State
Logical	Sym	Inv. Tracks	Logical	Sym	Inv. Tracks	STD <=> BCV
DEV001	0000	0	BCV001	00C8 *	0	Synchronized
DEV002	0001	0	BCV002	00C9 *	0	Synchronized
DEV003	0002	0	BCV003	00CA *	0	Synchronized
DEV004	0003	0	BCV004	00CB *	0	Synchronized
DEV005	0004	0	BCV005	00CC *	0	Synchronized
DEV006	0005	0	BCV006	00CD *	0	Synchronized
DEV007	0006	0	BCV007	00CE *	0	Synchronized
DEV008	0007	0	BCV008	00CF *	0	Synchronized
DEV009	0008	0	BCV009	00D0 *	0	Synchronized
DEV010	0009	0	BCV010	00D1 *	0	Synchronized
DEV011	000A	0	BCV011	00D2 *	0	Synchronized
DEV012	000B	0	BCV012	00D3 *	0	Synchronized
DEV013	000C	0	BCV013	00D4 *	0	Synchronized
DEV014	000D	0	BCV014	00D5 *	0	Synchronized
DEV015	000E	0	BCV015	00D6 *	0	Synchronized
DEV016	000F	0	BCV016	00D7 *	0	Synchronized
DEV017	0010	0	BCV017	00D8 *	0	Synchronized
DEV018	0011	0	BCV018	00D9 *	0	Synchronized
DEV019	0012	0	BCV019	00DA *	0	Synchronized
DEV020	0013	0	BCV020	00DB *	0	Synchronized
DEV021	0014	0	BCV021	00DC *	0	Synchronized
Total	-----			-----		
Track(s)		0			0	
MB(s)		0.0			0.0	

Legend:

(*): The paired BCV device is associated with this group.

For SYMCLI to access a specified database, set the `SYMCLI_RDB_CONNECT` environment variable to the username and password of the system administrator's account. The `export` command sets this variable to a username of "system" and a password of "manager," allowing a local connection.

```
C:\> set SYMCLI_RDB_CONNECT=system/manager
```

The `symioctl begin snapshot` command sends the SQL command "BACKUP DATABASE TO VIRTUAL_DEVICE WITH SNAPSHOT" to SQL Server, which begins the snapshot backup and suspends writes to the database named SQL2000a. It also creates a savefile named `sql.save`.

```
C:\> symioctl begin snapshot SQL2000a savefile sql.save -type sqlserver -nop
```

The `symmir split` command performs an instant split on all BCV pairs in the device group `sqlserv`.

```
C:\> symmir -g sqlserv split -instant -noprompt
```

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

'Split' operation successfully executed for device group 'sqlserv'.

The `symioctl end snapshot` command causes the SQL command "BACKUP DATABASE" to complete. Writes to the database can resume, and the snapshot metadata is saved to the save file.

```
C:\> symioctl end snapshot -type sqlserver SQL2000a -noprompt
```

Splitting BCV Pairs Using Oracle Database Assist

The hardware setup for this example consists of a Solaris host (api179) connected to a local Symmetrix unit. Oracle 8.1.7.0.0 software is installed on api179. A device group called `oracle` was created beforehand. The devices that hold the database were added to this device group. The BCV pairs in the device group were fully established.

For SYMCLI to access a specified database, set the `SYMCLI_RDB_CONNECT` environment variable to the username and password of the system administrator's account. The first `export` command sets this variable to a username of "system" and a password of "manager," allowing a local connection to Oracle. The `export ORACLE_HOME` command specifies the location of the Oracle binaries. The `export ORACLE_SID` command specifies the database instance name. (Other RDBMS database systems use environment variables specific to their system. For example, Sybase uses the variable `SYBASE` for the location of the Sybase binaries and uses `DSQUERY` to specify the server name.)

```
# export SYMCLI_RDB_CONNECT=system/manager
# export ORACLE_HOME=/disks/symapidvt/oraclehome/api179
# export ORACLE_SID=api179
```

The `symrdb list` command allows you to test basic database connectivity.

```
# symrdb list -type oracle
```

DATABASE NAMES (ORACLE 8.1.7.0.0):

```
Database Name
-----
```

```
api179
```

The `symioctl begin backup` command places all tables in the Oracle database (api179) in hot backup mode.

```
# symioctl -type oracle begin backup -noprompt
```

The `symioctl freeze` command suspends writes to the Oracle database (api179).

```
# symioctl -type oracle freeze -noprompt
```

The `symmir split` command performs an instant split on all BCV pairs in the device group called oracle.

```
# symmir -g oracle split -instant -noprompt
```

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

```
'Split' operation successfully executed for device group 'oracle'.
```

The `symioctl thaw` command allows writes to the database to resume.

```
# symioctl -type oracle thaw -noprompt
```

The `symioctl end backup` command terminates hot backup mode.

```
# symioctl -type oracle end backup -noprompt
```