HITACHI
DATA SYSTEMS

Hitachi Freedom Storage™ Lightning 9900™ V Series

Hitachi DB Validator Reference Guide

Trademarks

Notice of Export Controls

Document Revision Level

Revision	Date	Description
MK-92RD140-0	September 2002	Initial Release
MK-92RD140-1	October 2002	Revision 1, supersedes and replaces MK-92RD140-0

Source Documents for this Revision

- R140_0 (Hitachi RSD review of this document).

Changes in this Revision

- Correction to section 2.2, Oracle® on LVM (VxVM), fifth dashed item: changed "online log" to "redo log".
- Correction to Figure 3.1: removed Oracle9iR2 HARD Test Kit.
- Added information that the default 9900V setting is that SIM reporting for DB Validator is not enabled (new note in section 3.7.1).

Referenced Documents

- *Hitachi Lightning 9900™ V Series User and Reference Guide*, MK-92RD100
- *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide*, MK-90RD011
- *Hitachi Lightning 9900™ V Series Remote Console – Storage Navigator User's Guide*, MK-92RD101
- *Hitachi Lightning 9900™ V Series LUN Manager User's Guide*, MK-92RD105
- *Hitachi Lightning 9900™ V Series Sun™ Solaris™ Configuration Guide*, MK-92RD123

Preface

The *Hitachi Lightning 9900™ V Series Hitachi DB Validator Reference Guide* provides an overview of the DB Validator feature of the Lightning 9900™ V Series (9900V) subsystem.

This document assumes that:

- the user has a background in data processing and understands direct-access storage device subsystems and their basic functions,
- the user is familiar with Oracle® database operations, and
- the user is familiar with the operating system(s) hosting the Oracle® database(s).

Note: The term "9900V" refers to the entire Lightning 9900™ V Series subsystem family (9980V, 9970V), unless otherwise noted. For further information on the 9900V subsystem, refer to *Hitachi Lightning 9900™ V Series User and Reference Guide* (MK-92RD100), or contact your Hitachi Data Systems account team.

Note: The use of Hitachi DB Validator and all other Hitachi Data Systems products is governed by the terms of your license agreement(s) with Hitachi Data Systems.

Microcode Level

This document revision applies to 9900V microcode versions 21-02-4*x* and higher.

COMMENTS

Please send us your comments on this document: doc.comments@hds.com.

Make sure to include the document title, number, and revision.
Please refer to specific page(s) and paragraph(s) whenever possible.

(All comments become the property of Hitachi Data Systems Corporation.)

Thank you!

Contents

List of Figures

List of Tables

Chapter 1 Overview of Hitachi DB Validator

1.1 Hitachi DB Validator

The Hitachi DB Validator function of the Lightning 9900™ V Series (9900V) subsystem is designed for the Oracle® database platform to prevent data corruption between the database and the storage subsystem and to protect existing Oracle® data on the subsystem. DB Validator prevents corrupted data blocks generated in the database-to-storage subsystem infrastructure from being written onto the disk storage. The conventional combination of networked storage and database management software has a risk of data corruption while writing data on the storage. This data corruption rarely occurs; however, once corrupted data is written into storage, it can be difficult and time-consuming to detect the underlying cause, restore the system, and recover the database. Hitachi DB Validator helps prevent potentially disastrous corrupted data environments and minimizes risk and potential costs in backup, restore, and recovery operations.

Hitachi DB Validator is provided as a program product option (license key) for the 9900V subsystem. Figure 1.1 shows a functional overview of DB Validator operations.

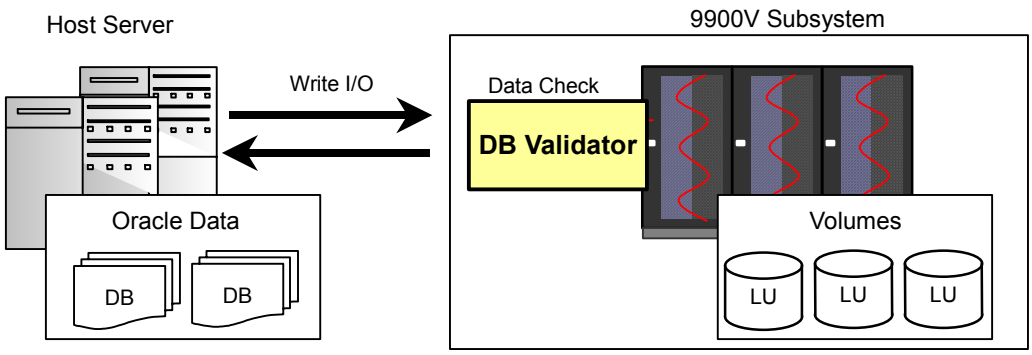

Figure 1.1 DB Validator Overview

Hitachi DB Validator provides the following important benefits for the user:

- **Prevents invalid Oracle® data from being written to the 9900V subsystem:**
  The data being stored on a disk subsystem is transferred through many channels and may become invalid during the process. DB Validator makes sure the Oracle® data is valid before it is written onto disk. If the data is invalid, DB Validator prevents the data from being written, thus achieving high-reliability of data in the 9900V subsystem.

  DB Validator also saves time and costs required for recovery when invalid data is written onto a volume, thereby increasing efficiency of data management and reducing costs.

- **Protects existing Oracle® data in the 9900V subsystem:**
  Once Oracle data is stored on a 9900V volume, DB Validator protects the existing data from being deleted or updated by mistake by checking the data to be stored on the same volume. For example, if another application attempts to write data onto a volume which contains Oracle® data, the 9900V will reject the write I/O and return an error.

Note: The use of Hitachi DB Validator and all other Hitachi Data Systems products is governed by the terms of your license agreement(s) with Hitachi Data Systems.

1.2 DB Validator Components

Figure 1.2 shows the DB Validator system components and their functions for Oracle® data validation checking. The Command Control Interface (CCI) software is used to specify the logical units (LUs) to be checked and the type of check to be performed (type depends on the Oracle® version). The files to be checked include data files (including tablespace files), redo log files, and control files. The files must be stored on raw volumes. If data is written (I/O) from the host server to the subsystem, the 9900V channel adapter (CHA) initiates the data check. If invalid data is detected, the write I/O is rejected, the CHA reports the error to CCI, and CCI outputs an error to the log (syslog).

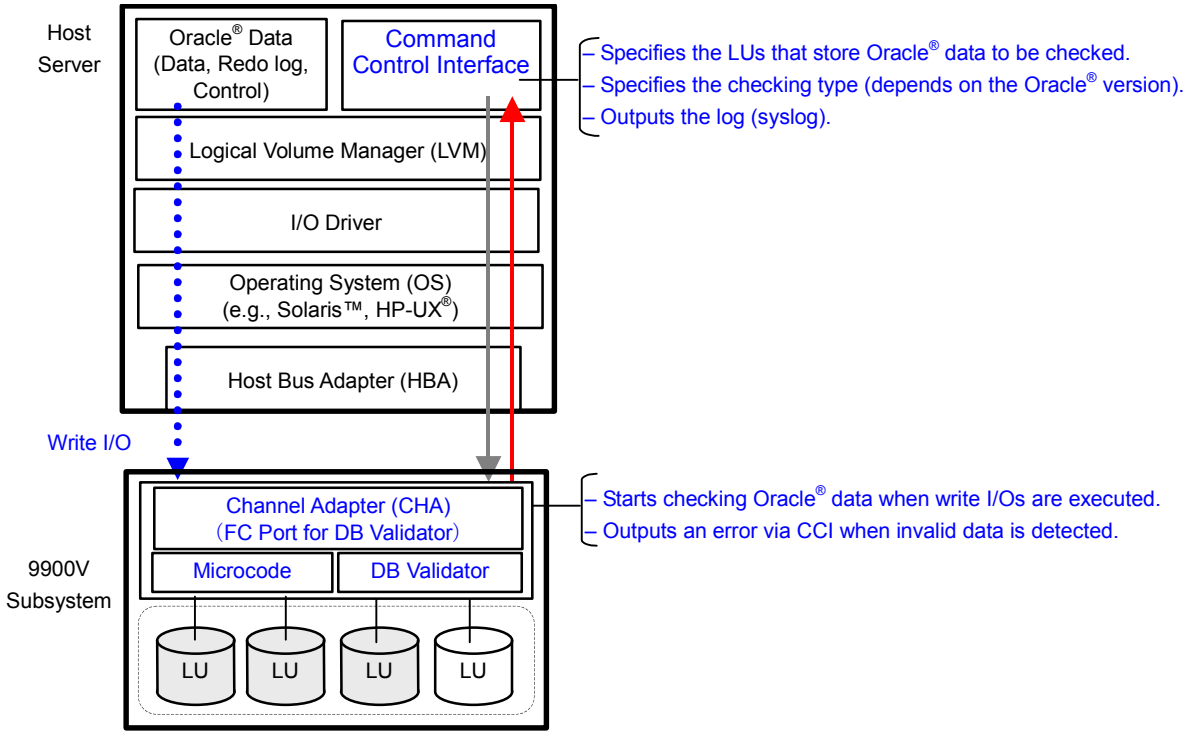

Figure 1.2 DB Validator Components

Figure 1.3 shows a typical database system configurations. As shown in this figure, there are many components between the Oracle® database program and the storage subsystem:

- Operating system
- I/O driver
- Logical Volume Manager (LVM)
- Fibre-channel (FC) host bus adapter (HBA)
- FC switch (and other network components)

These components may fail intermittently or persistently. When these failures are detected as errors, the system administrator can take appropriate recovery action. However, there still exists a very small probability that all layers might pass the data without detecting the failure. The Oracle® software and the 9900V have very robust data protection capabilities that can prevent data corruption within both DBMS and the subsystem, but until now they could not detect data corruption which occurred between DBMS and the subsystem.

The other data corruption risk is a non-Oracle® application overwriting the Oracle® data files on the storage subsystem. The 9900V DB Validator function also protects existing Oracle® data files from being overwritten by data from other applications.

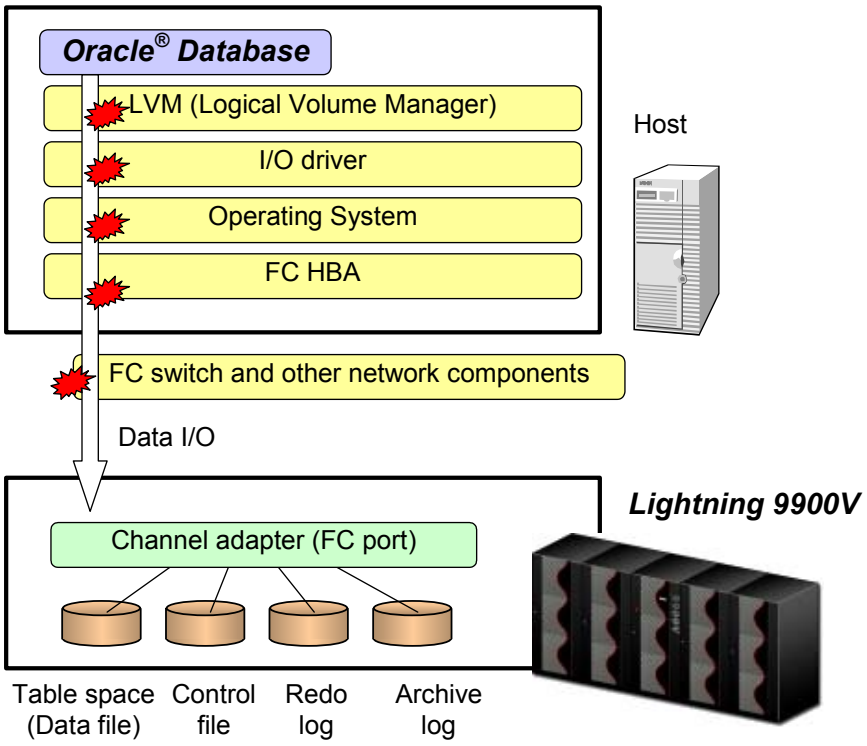

Figure 1.3 Typical Database System Configuration

1.3 Oracle® H.A.R.D. and DB Validator Operations

Figure 1.4 shows a typical system configuration with DB Validator. The Oracle® Hardware Assisted Resilient Data (H.A.R.D.) feature protects against data corruption. The Oracle® application software computes checksums which are included with every data block written to disk. The Lightning 9900™ V Series DB Validator feature recomputes the checksums upon receiving data blocks to be written to Oracle® database volumes. In the event that the Hitachi subsystem receives a corrupt block, the I/O operation is rejected with a checksum failure error. The ability of the subsystem to recognize that it has received corrupt data and reject the I/O prevents corrupt data from being written to the database, thus eliminating "silent" data corruption. Without DB Validator (but with Oracle® H.A.R.D.), corrupt data is discovered, but only when the data is read back at a later date.

Hitachi storage subsystems already perform extensive checksum-type validation of data integrity from the point of entry through the entire storage subsystem. The new DB Validator feature detects corruption occurring outside the subsystem by checking write data upon receipt. The objective is to prevent corrupt data destined for Oracle® volumes from being accepted by the subsystem.

Detection of corrupt data: If the DB Validator function detects corrupt data, the I/O operation from the host is rejected, just as if there had been a hardware failure. The error code indicates "H.A.R.D. failure". If the data corruption problem is transient, the condition may clear when the host retries the I/O. If there is no successful retry operation, the application I/O operation fails, and the Oracle® update transaction does not complete. The Hitachi solution resource manager logs the H.A.R.D. error to the host's syslog.

Hardware acceleration: The new 9900V channel adapters (CHAs) include hardware acceleration of DB Validator checksum computation. Hitachi's implementation of checksum validation using hardware acceleration minimizes performance impact (the impact depends on the system environment).

Multiple levels of checking: In addition to the verification of checksums for overall database blocks, the 9900V also performs checksum validation for lower-level data structures specific to the type of file. Tablespace and control files have one type of lower-level checking, while redo log files have another type. Hitachi's additional lower-level data structure checking further increases protection. When DB Validator is turned on using CCI, the type of data (data file vs. redo log) must be specified so that the correct type of lower-level data structure validation is performed.

Read operations: No additional Oracle® checksum validation is done for read operations. The Oracle® application verifies the checksum at the application end. (The 9900V subsystem always performs its own internal checksum verification on reads, and hence always resends the same data originally received.)

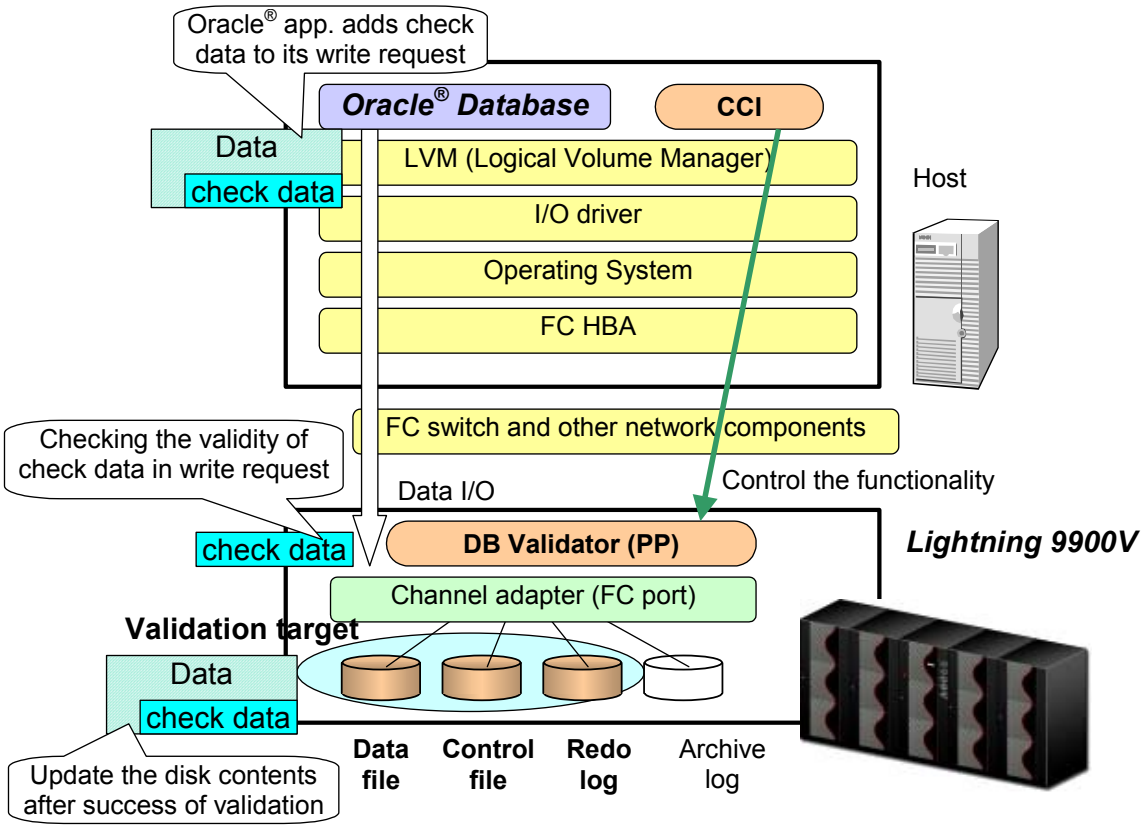

Figure 1.4 Typical Database System Configuration with DB Validator

DB Validator validates the check data at every write operation to the storage and checks out the abnormal data immediately after a failure occurs. DB Validator blocks the incorrect write request from the host, returns a SCSI Check Condition error, and protects the correct data in the storage. The Oracle® database would stop itself as an error or separate the damaged table, depending on the error situation, but the administrator can recover the database without the loss of data consistency because any invalid data is not written to the storage, nor any valid data is overwritten.

Chapter 2 Requirements and Restrictions

2.1 Operating Environments and Prerequisites

To use Hitachi DB Validator, you need the following:

- Lightning 9900™ V Series subsystem (see section 2.1.1),
- Host server that supports Oracle® database software (see section 2.1.2), and
- Oracle-data validation service (see section 2.1.3).

2.1.1 Requirements for the 9900V Subsystem

For Hitachi DB Validator operations, you need a subsystem that has all settings of the required hardware, microcode, and the DB Validator option.

- **Channel adapter:** The following 9900V CHA types support DB Validator operations:
  - DKC-F460I-8HSF
  - DKC-F460I-8HLF

  Caution: All FC ports (CHAs) in the 9900V subsystem should be upgraded to support DB Validator. If other CHAs are used, the data validity checking is not performed if the Oracle® data goes through a CHA which does not support DB Validator.
- **Microcode:** The minimum microcode level for DB Validator operations is 21-02-4*x*.
- **DB Validator option:** The DB Validator program product option (license key) must be installed on the target 9900V subsystem using the 9900V Storage Navigator (or SVP). For instructions on installing the DB Validator program product option, please refer to the *Lightning 9900™ V Series Remote Console - Storage Navigator User's Guide*.

  Note: A temporary license key (trial key) for DB Validator is not provided. You need a permanent license key. For details, contact your Hitachi Data Systems account team.
- **Command Control Interface:** The minimum CCI version for DB Validator operations is 01-10-03/01.

  Note: The CCI command device is configured using the 9900V LUN Manager component of the 9900V Storage Navigator remote console software. If Storage Navigator and/or LUN Manager is not installed, please contact your Hitachi Data Systems account team about LUN configuration services.

2.1.2 Requirements for the Host Server

The host server requirements for Hitachi DB Validator operations are:

- **Operating System:**
  - Sun™ Solaris™ version 8 or higher. For further information on 9900V operations with the Solaris™ platform, please refer to *Hitachi Lightning 9900™ V Series Sun™ Solaris™ Configuration Guide* (MK-92RD123). For further information on the Solaris™ OS, please refer to the Solaris™ user documentation, or contact Sun™ technical support.
  - HP-UX® version 11.0 or higher. For further information on 9900V operations with the HP-UX® platform, please refer to *Hitachi Lightning 9900™ V Series HP-UX® Configuration Guide* (MK-92RD122). For further information on the HP-UX® OS, please refer to the HP-UX® user documentation, or contact HP® technical support.
- **Oracle® Database:**
  - Oracle8i™ database product
  - Oracle9i™ database product
  - Oracle9i™ database product, Release2

  Note: Oracle® database functionality of check data is slightly enhanced with Oracle9i™ Release2, so DB Validator has two distinct modes: one mode for Oracle8i™ and Oracle9i™ databases, and another mode for Oracle9i™ Release2 (and later) databases.

  Note: The Oracle® database needs to be constructed on raw devices (i.e., no file system). Also, you need to set DB_BLOCK_CHECKSUM parameter to TRUE.
- **Logical Volume Manager:**
  - VERITAS® VxVM (for the Solaris™ OS)
  - LVM (for the HP-UX® OS)

Note: Versions other than those listed above may be supported in future enhancements. For the latest information on supported OS and software versions, please contact your Hitachi Data Systems account team.

2.1.3 Oracle-Data Validation Service

In order to use DB Validator, you need to set up the hardware and software for DB Validator in both the 9900V subsystem and the host server depending on the user's environment in advance. Operations for these settings are provided as an Oracle-Data Validation Service, and this service is required. For details, contact your Hitachi Data Systems account team.

2.2 Restrictions

When DB Validator is installed or in use, the following restrictions and conventions apply.

- **9900V subsystem:**
  - Do not downgrade the 9900V microcode (DKCMAIN program) version.
  - Do not deinstall an existing CHA channel feature which supports DB Validator and replace it with a CHA channel feature which does not support DB Validator. Note that the microcode will guard replacements for single defective CHAs, not allowing the CE to replace a DB Validator capable PCB with one that is not capable.
  - If a volume (LU) being checked by DB Validator has multiple paths, you must make sure that all paths use a 9900V CHA that supports DB Validator. Hitachi Data Systems recommends that you update all CHAs to support DB Validator.
  - If you uninstall the DB Validator program product option, you need to delete the settings for the volumes being checked by DB Validator using CCI in advance.
- **Host server:**
  - Do not write data other than Oracle® data to a volume (LU) that is being checked by DB Validator.
  - If an LU being checked by DB Validator has multiple paths, make sure that each path is set as a target of checking.
  - If host-based striping (e.g., LVM with striping functionality) is used on an LU, then the stripe size must be an exact multiple of the Oracle® data block size for that LU.
- **Oracle® application:**
  - Make sure that the Oracle® DB_BLOCK_CHECKSUM parameter is set to TRUE.
  - Make sure that the Oracle® application is set to run in Archive Log mode.
- **Oracle® files:**
  - **Raw devices:** All Oracle® tablespace files must be placed on raw volumes (including LVM raw volumes). DB Validator does not support file system-based Oracle® files.
  - **Data files separate from redo log files:** The Oracle® redo log files (including archive logs) must be on different LUs than the data files (including control files). Make sure that the Oracle® redo log files and data files are not located on the same LU.

    Note: Since other files (archive log files) are not the target of the validation, no special consideration for these files is required.
  - **Block size:** Use different 9900V LUs for Oracle® files which have different data block size. Do not store Oracle® data of different block sizes on the same volume (LU).
  - **Restoring Oracle® files:** If the database was running without checksum enabled in the past, old blocks without checksum information may exist on disk. If this is the case and you need to restore Oracle® data files from a backup, you need to disable validation checking for those data files that were backed up prior to Oracle® checksum being enabled.

- **Oracle® on LVM (VxVM)**
  - If LVM is used, the 9900V LDEVs after LVM mapping must obey the above two separation rules (data/control files separate from redo log files, and block size).
  - If LVM is used, configure the LUs before enabling DB Validator checking using CCI. The LVM performs some writes to disk as part of the configuration process. Once the LUs are configured, the LVM does not issue any more writes, so DB Validator checking can be enabled.
  - LVM block size must be a multiple of the Oracle® block size, so that whole (integral) Oracle® blocks with checksums are written to disk.
  - The Oracle® block size must be less than or equal to the minimum of the LVM stripe size and the largest block size that LVM will not fracture (known as "Logical Track Group" in LVM), which is 256 kB in LVM.
  - When adding new physical volumes (PVs) to a logical volume (LV) to be used as an Oracle® data file, control file, or redo log, in order to have H.A.R.D. checking take effect on those new PVs, re-enable the data validation for the LV.

    Similarly, in order to have H.A.R.D. checking no longer performed on PVs that have been removed from an LV that had previously been used by Oracle®, H.A.R.D. checking should be explicitly disabled on the device corresponding to the PV.
  - If host-based mirroring (e.g., LVM mirroring) is used, all component PV mirrors must be H.A.R.D.-enabled, otherwise the entire logical volume (LV) is exposed to data corruption. That is, if a user takes an unmirrored H.A.R.D.-enabled LV and makes it mirrored "on the fly" without H.A.R.D.-enabling all sides of the mirror, that entire LV is exposed.
  - LVM bad block relocation will not be allowed on PVs that are H.A.R.D.-enabled.

- **Oracle® and LVM (VxVM) on HA Cluster Server**
  - Some HA cluster software may write data to volumes at regular intervals. Make sure to adjust the validation target range for such software. Its LVM area must be out of checking for DB Validator by using the **-vs <bsize> SLBA ELBA** option.

Chapter 3 Hitachi DB Validator Operations

3.1 Preparing for DB Validator Operations

To implement the Hitachi DB Validator function effectively, you need to consider the following points:

1. **Setup and configuration:** Particular consideration is required for Lightning 9900V volume setup and Oracle® volume mapping:
   - All requirements and restrictions in sections 2.1 and 2.2 must be observed (e.g., raw files, data/control files separate from redo log files, block size, stripe size, LVM bad block relocation, HA cluster software, host-based mirroring, etc.).
   - All mapping operations should be done with DB Validator checking disabled.
   - Identify all LUs that will be the targets of DB Validator checking, and write down the path(s) for each LU (port, TID, LUN). Make sure that all paths use CHAs that support the DB Validator function.

   Important: Mapping information between the Oracle® files and the Lightning LDEVs is very helpful to investigation of configuration and database recovery. We strongly recommend that you record this information at setup time.
2. **Error recovery:** The potential error causes and corresponding recovery procedures should be confirmed before the database goes into production phase.
3. **Redundancy:** Redundant system design (e.g., cluster, disaster recovery, etc.) is strongly recommended to be combined with this functionality to take over immediately after the failure detection and to minimize downtime.

3.2 Installing and Configuring CCI

After you have prepared for DB Validator operations as described in section 3.1, you are ready to install and configure the CCI software, if not already installed and configured.

- **Installing CCI:** The CCI software product is installed on the host server. For instructions on installing CCI, please refer to the *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide* (MK-90RD011).
- **Configuring the command device:** The command device is configured on the 9900V subsystem using the LUN Manager component of the Storage Navigator remote console software. For instructions on configuring the command device, please refer to the *Hitachi Lightning 9900™ V Series Remote Console – Storage Navigator User's Guide* (MK-92RD101).
- **Configuring CCI:** CCI controls DB Validator operations based on the configuration definition file (**horcm.conf**) that stores information about target volumes, command devices, etc. The **horcm.conf** file is created and stored on the host running the CCI software. For detailed information on the CCI configuration definition file, please refer to the *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide* (MK-90RD011).

Note: See section 3.5 for an example of CCI configuration for DB Validator operations.

3.3 Installing the DB Validator Option

Before you can begin DB Validator operations, you must install the DB Validator option on the 9900V subsystem using the 9900V Storage Navigator software. For instructions on installing the DB Validator option, please refer to the *Hitachi Lightning 9900™ V Series Remote Console - Storage Navigator User's Guide* (MK-92RD101).

Note: If you need to uninstall the DB Validator option, you need to use CCI first to delete the settings for all volumes being checked by DB Validator.

3.4 Starting DB Validator Operations

DB Validator operations are controlled through CCI commands. No other software or application can be used. CCI provides the **raidvchkset**, **raidvchkdsp**, and **raidvchkscan** commands to set and verify the parameters for Oracle® data validation checking at the LU level. For detailed information on these commands, please refer to the *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide* (MK-90RD011).

To start DB Validator operations, use the **raidvchkset** CCI command to enable DB Validator checking at the volume level for the desired LU(s). For LUs with multiple paths, make sure to enable checking for all paths. Specify the type of check according to the Oracle® version.

See section 3.5 for an example of configuring, starting, and performing DB Validator operations.

Make sure to follow these restrictions:

- Validation must temporarily be turned off before changing LVM configuration. Initial configuration of LVM should be completed before enabling validation.
- Validation must temporarily be turned off before restoring data from a backup which was taken prior to enabling this function.
- The CREATE TABLESPACE command must be used with the REUSE option. A validation error may occur without the REUSE option, depending on the host environment.
- All programs other than the Oracle® Database application should not write to validation target volumes. Please be careful when multiple paths are defined for one volume.

Note: Once the DB Validator checking function is enabled for an LU, all write operations to that LU must have a valid Oracle® checksum.

3.5 Sample Implementation of DB Validator

Figure 3.1 shows a sample DB Validator system configuration. This section presents the setup and operational information for this sample configuration:

- Volume setup (see section 3.5.1),
- CCI setup (see section 3.5.2),
- Oracle® setup (see section 3.5.3),
- Enabling DB Validator checking (see section 3.5.4),
- Displaying DB Validator volumes (see section 3.5.5),
- Creating the database (see section 3.6), and
- Disabling and restarting DB Validator checking (see section 3.5.6).

Note: This section describes the setup operations for the Solaris® and VERITAS® VxVM environment. These operations can be easily translated to other platforms.

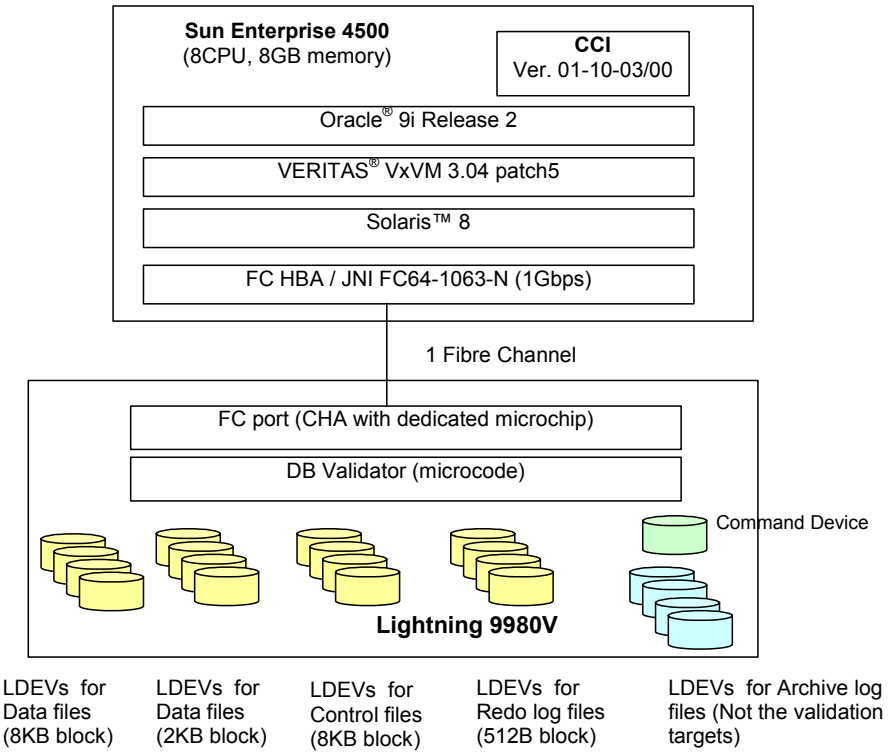

Figure 3.1 Sample DB Validator System Configuration

3.5.1 Volume Setup

Table 3.1 shows the volume setup for the sample configuration shown in Figure 3.1. All mapping operations must be done with DB Validator disabled.

Note: This document does not describe 9900V device configuration. Please refer to the 9900V Configuration Guide for the host platform (e.g., *Hitachi Lightning 9900™ V Series Sun™ Solaris™ Configuration Guide*, MK-92RD123).

Note: This document does not describe path allocation using the LUN Manager software. Please refer to the *Hitachi Lightning 9900™ V Series LUN Manager User's Guide* (MK-92RD105).

Note: This document does not describe VxVM volume configuration. Please refer to the VERITAS® VxVM user documentation.

Table 3.1 Volume Setup for Sample Configuration

Lightning 9900™ V Series				VERITAS™ VxVM		
CU:LDEV	**LUN**	**Type ***	**Raw Device Name****	**VxVM Settings**	**Volume Name**	**Purpose**
00:00	0	OPEN-3	/dev/dsk/c4t0d0	Concatenated	/mnt/oralog1	Oracle redo log file (block size=512B)
00:01	1	OPEN-3	/dev/dsk/c4t0d1			
00:02	2	OPEN-3	/dev/dsk/c4t0d2			
00:03	3	OPEN-3	/dev/dsk/c4t0d3	Concatenated	/mnt/oractl1	Oracle control file (block size=8KB)
00:04	4	OPEN-3	/dev/dsk/c4t0d4			
00:05	5	OPEN-3	/dev/dsk/c4t0d5	Concatenated, and divided	/mnt/oradata1, /mnt/oradata2	Oracle data files (block size=8KB)
00:06	6	OPEN-3	/dev/dsk/c4t0d6			
00:07	7	OPEN-3	/dev/dsk/c4t0d7			
00:08	8	OPEN-3	/dev/dsk/c4t0d8	Divided	/mnt/orastok1, /mnt/orastok2	Oracle data files (block size=2KB)
00:09	9	OPEN-3	/dev/dsk/c4t0d9	None	---	Command device

* Type refers to the size, device emulation, sharing conditions, etc. If the 9900V LUN Expansion (LUSE) and Virtual LVI/LUN features are installed, you can define any size of volume appropriate for your environment.

** Device names recognized on the host are determined by the settings of path definition of Lightning 9900V, OS internal rule, and HBA device driver.

3.5.2 CCI Setup

Figure 3.2 shows the **horcm.conf** file used for the sample configuration shown in Figure 3.1. All CCI operations in the following sections use this **horcm.conf** file.

- HORCM_MON and HORCM_INST are parameters used by CCI to monitor Hitachi TrueCopy™ and ShadowImage pair status. If you do not plan to use these functions, set localhost and default parameters. Set the proper parameters to use these functions.
- HORCM_CMD describes the special file names of command device(s). CCI finds the command device(s) by referring to HORCM_CMD.
- HORCM_DEV defines group names and device names to physical volumes in 9900V and uses them in succeeding operations. In the above example, "oralog" and "log1-1" are defined as the group name and the device name, respectively, referring to the device file "/dev/dsk/c4t0d0" connecting to port "CL1-E" on the Lightning 9900V subsystem.

```
HORCM_MON
#ip_address service poll(10ms) timeout(10ms)
localhost    horcm     1000     3000

HORCM_CMD
#dev_name           dev_name         dev_name
/dev/rdsk/c4t0d9s2

HORCM_DEV
#dev_group dev_name  port# TargetID LU# MU#
oralog      log1-1   CL1-E     0     0
oralog      log1-2   CL1-E     0     1
oralog      log1-3   CL1-E     0     2
oractl      ctrl1-1  CL1-E     0     3
oractl      ctrl1-2  CL1-E     0     4
oradata     data1    CL1-E     0     5
oradata     data2    CL1-E     0     6
oradata     data3    CL1-E     0     7
orastok     stokdata CL1-E     0     8

HORCM_INST
#dev_group     ip_address      service
oralog         localhost        horcm
oractl         localhost        horcm
oradata        localhost        horcm
orastok        localhost        horcm
```

Figure 3.2 CCI Configuration Definition File (horcm.conf) for Sample Configuration

3.5.3 Oracle® Setup

The following steps must be performed before DB Validator checking is enabled:

- The Oracle® database must run in **Archive Log** mode.
- The **DB_BLOCK_CHECKSUM** parameter of init.ora must be set to **TRUE**.
- In case of Tru64™ UNIX® or Windows® platform, the user must set the parameter in the **init.ora** file to **HARD_PROTECTION = TRUE**.

3.5.4 Enabling DB Validator Checking

Figure 3.3 shows the CCI **raidvchkset** commands used to enable DB Validator for the sample configuration shown in Figure 3.1:

- Parameters with **-g** specify the group of LDEVs with group name written in **horcm.conf**.
- Parameters with **-vt** specify the validation file type described in Table 3.2.
- Parameters with **-vs** specify the Oracle® data block size in units of 512 bytes. For example, 16 means 8KB.

Important: See the next section for important information on enabling DB Validator checking with LVM considerations.

```
# raidvchkset –g oralog –vt redo9 –vs 1
# raidvchkset –g oractl –vt data9 –vs 16
# raidvchkset –g oradata –vt data9 –vs 16
# raidvchkset –g orastok –vt data9 –vs 4
```

Figure 3.3 Enabling DB Validator Checking for Sample Configuration

Table 3.2 CCI Parameters for File Types and Oracle Versions

Oracle® Version	File Type	Parameter for -vt Option
Oracle® 9i Release 2	Data file Control file	data9
	Redo log file	redo9
Oracle® 9i prior to Release 2 Oracle® 8i	Data file Control file	data8
	Redo log file	redo8

3.5.4.1 Enabling DB Validator Checking with LVM Considerations

LVM may write meta data to the volume depending on the volume configuration. In this case, you must exclude the meta data area from the validation target range. You can specify the target range for each volume with the extended parameters of **raidvchkset –vs** option. Please refer to the *CCI User and Reference Guide* for the exact syntax. The **vxdisk list** and **vxdg list** commands show the range of meta data regions for the VxVM LVM, which are called private regions. Please refer to the VxVM documentation for information on private regions.

Figure 3.4 is an example of using the VERITAS VxVM **vxdisk** list command to determine the target range and the correct syntax for excluding the meta data from the validation target range. Specifically this example is for the Redo log.

```
# raidvchkset -g oradata -d data -vt data9 -vs 4 2880 71184960
# raidvchkset -g oralog -d log1 -vt redo9 -vs 1 2880 4799520    Note: 2879 + 1 = 2880

The SLBA and the ELBA is the offset from "vxdisk list" command.

Example for the redo log
#  fire-2 >vxdisk list c4t0d9s2
Device:    c4t0d9s2
devicetag: c4t0d9
type:      sliced
hostid:    fire-2
disk:      name=dg901_1 id=1031839952.2594.fire-2
group:     name=dg901 id=1031840145.2607.fire-2
info:      privoffset=1
flags:     online ready private autoconfig autoimport imported
pubpaths:  block=/dev/vx/dmp/c4t0d9s2 char=/dev/vx/rdmp/c4t0d9s2
privpaths: block=/dev/vx/dmp/c4t0d9s1 char=/dev/vx/rdmp/c4t0d9s1
version:   2.2
iosize:    min=512 (bytes) max=256 (blocks)
public:    slice=4 offset=0 len=4799520
private:   slice=3 offset=1 len=2879
update:    time=1031840146 seqno=0.5
headers:   0 248
configs:   count=1 len=2104
logs:      count=1 len=318
Defined regions:
 config   priv 000017-000247[000231]: copy=01 offset=000000 enabled
 config   priv 000249-002121[001873]: copy=01 offset=000231 enabled
 log      priv 002122-002439[000318]: copy=01 offset=000000 enabled
Multipathing information:
numpaths:   1
c4t0d9s2    state=enabled
```

Figure 3.4 Example of Enabling DB Validator Checking with LVM Consideration

3.5.5 Displaying DB Validator Volumes

The DB Validator volumes can be displayed using the following CCI commands:

- inqraid (see section 3.5.5.1),
- raidvchkdsp (see section 3.5.5.2), and
- raidvchkscan (see section 3.5.5.3).

3.5.5.1 Displaying DB Validator Volumes Using inqraid

To verify the validation volumes, you can use the **inqraid -fp** command with the **-CLI** option (see Figure 3.5). DB Validator checking is enabled for the volumes with an asterisk (*). For further information on the **inqraid** command, please refer to the *CCI User and Reference Guide*.

```
# ls /dev/rdsk/c4t0d* | ./inqraid –CLI –fp
DEVICE_FILE     PORT    SERIAL  LDEV CTG H/M/12  SSID R:Group PRODUCT_ID
c4t0d0*        CL1-E    15005    00   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d1*        CL1-E    15005    01   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d2*        CL1-E    15005    02   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d3*        CL1-E    15005    03   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d4*        CL1-E    15005    04   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d5*        CL1-E    15005    05   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d6*        CL1-E    15005    06   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d7*        CL1-E    15005    07   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d8*        CL1-E    15005    08   -  s/s/ss  0004 5:01-03 OPEN-3
c4t0d9         CL1-E    15005    09   -  s/s/ss  0004 5:01-03 OPEN-3
```

Figure 3.5 Displaying DB Validator Volumes for Sample Configuration

3.5.5.2 Displaying DB Validator Volumes Using raidvchkdsp

The **raidvchkdsp** command (see Figure 3.6) displays the relation between the Device_File and the paired volumes, based on the group (as defined in the local instance configuration definition file). If Device_File column shows "Unknown" to HOST (instance), then the volume is not recognized on own HOST, and **raidvchkdsp** command will be rejected in protection mode. Non-permitted volume is shown without LDEV# information (LDEV# is " - ").

Please refer to the *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide* (MK-90RD011) for more examples of the raidvchkdsp command, and parameter definition and displayed field definitions.

```
raidvchkdsp -g vg01 -fd -v cflag          ← Example of -fd option showing Unknown vol.
Group    PairVol Device_File    Seq# LDEV#  BR-W-E-E  MR-W-B  BR-W-B  SR-W-B-S
vg01     oradb1  Unknown        2332    -    - - - -   - - -   - - -   - - - -
vg01     oradb2  c4t0d3         2332    3    D E B R   D D D   D E E   D E D D

raidvchkdsp -g vg01 -fd -v cflag                          ← Example of -v cflag option.
Group    PairVol Device_File    Seq# LDEV#  BR-W-E-E  MR-W-B  BR-W-B  SR-W-B-S
vg01     oradb1  c4t0d2         2332    2    D E B R   D D D   D E E   D E D D
vg01     oradb2  c4t0d3         2332    3    D E B R   D D D   D E E   D E D D
```

Figure 3.6 Raidvchkdsp Command Examples showing -fd and -v cflag Options

3.5.5.3 Displaying DB Validator Volume Using raidvchkscan

The **raidvchkscan** command (see Figure 3.7) displays the fibre port of the 9900V, target ID, LDEV mapped for LUN# and the parameters for validation checking, regardless of the configuration definition file.

Please refer to the *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide* (MK-90RD011) for more examples of the raidvchkscan command, and parameter definition and displayed field definitions.

```
raidvchkscan -p CL1-A -v cflag
PORT# /ALPA/C TID# LU#   Seq#  Num LDEV#   BR-W-E-E  MR-W-B  BR-W-B  SR-W-B-S
CL1-A / ef/ 0    0   0   2332    1     0    D E B R   D D D   D E E   D E D D
CL1-A / ef/ 0    0   1   2332    1     1    D E B R   D D D   D E E   D E D D
CL1-A / ef/ 0    0   2   2332    1     2    D E B R   D D D   D E E   D E D D
CL1-A / ef/ 0    0   3   2332    1     3    D E B R   D D D   D E E   D E D D
CL1-A / ef/ 0    0   4   2332    1     4    D E B R   D D D   D E E   D E D D
```

Figure 3.7 Raidvchkscan Command Example with -v cflag Option

3.5.6 Disabling and Restarting DB Validator Checking

You can disable DB Validator checking at the volume level using the **raidvchkset -vt** command with no parameter (see Figure 3.8). To restart validation checking, you can use the **raidvchkset -vt** command with the validation type only. With this syntax, the previous Oracle data block size setting (-vs option) is preserved.

```
# raidvchkset –g oralog –vt
# raidvchkset –g oractl –vt
# raidvchkset –g oradata –vt
# raidvchkset –g orastok –vt
```

Figure 3.8 Disabling DB Validator Checking for Sample Configuration

```
# raidvchkset –g oralog –vt redo9
# raidvchkset –g oractl –vt data9
# raidvchkset –g oradata –vt data9
# raidvchkset –g orastok –vt data9
```

Figure 3.9 Restarting DB Validator Checking for Sample Configuration

3.6 Creating the Database

You can use the standard procedure described in the Oracle® documentation to create a database.

Note: When you issue the CREATE TABLESPACE command, it must be used with the REUSE option. If the REUSE option is not specified, a validation error may occur depending on the host environment.

3.7 Error Detection and Recovery

This section provides general guidelines for error detection and recovery. Most errors are caused by hardware failure or misoperation/misconfiguration. Recovery and restoration can be easy for these cases. In addition, rare but more serious intermittent errors which cause data corruption are also detected by DB Validator immediately, thus minimizing the damage from error and the cost for recovery.

3.7.1 Error Detection

When the Lightning 9900V detects a validation error, the 9900V reports a 'Check Condition' error to the host. If the faulty write request came from the Oracle® application, the following actions may be taken, depending on where the error occurred:

- The Oracle® application turns the damaged data file offline, and/or
- The Oracle® instance stops.

The Lightning 9900V outputs the error information in three ways: service information message (SIM), syslog (on the host), 9900V-internal sense byte (SSB).

- **SIM:** Users can view the SIMs using the 9900V Storage Navigator remote console software. SIMs are automatically reported to the maintenance center (if contracted). ***Note:*** The default 9900V setting is that SIM reporting for DB Validator is not enabled.
- **Syslog:** CCI writes a syslog message which includes the error volume number and error counters for each validation item. CCI polls the 9900V volume status and outputs an error message if the error counter is incremented. Multiple errors that occur in one volume during the polling period are put together in one error message. Figure 3.10 shows an example of the syslog message output by CCI.

```
[HORCM_103] Detected a validation check error on this volume(log1-1 unit#0,ldev#2)
: CfEC=1, MNEC=0, SCEC=0, BNEC=0
```

Figure 3.10 Example of Syslog Message Output by CCI

- **SSB:** The SSB log on the SVP can only be accessed by Hitachi Data Systems service personnel. The SSB log includes the following information for a validation error:
  - Error volume (9900V LDEV).
  - Start address of write request control data block (CDB).
  - Transfer length of write request CDB.
  - WWN (worldwide name) of host port.
  - Validation item(s) judged as invalid.

3.7.2 Investigating the Cause of the Error

To investigate the cause of a validation error:

1. Check all hardware components of the system. If there is a failure, repair that failure, and recover the system.
2. If there is no hardware failure, then check the software.
   The most common error cause is misconfiguration and misoperation. In the system setup phase this error may often occur. Confirm every operation step by step. For production phase databases, error sources can be easier to find, because no configuration changes are made to a production phase database. The misoperation or misconfiguration must have been executed outside Oracle® Database. For validation errors which did not come from the Oracle application, Oracle® is not affected and continues to operate normally. This can be confirmed by checking the Oracle® log files (e.g., Alert.log).

   If the error cause seems to be misconfiguration, please check the following. Validation parameters can be investigated using the CCI raidvchkdsp command.

   - Make sure that the block size (raidvchkset -vs <size>) is correct.
   - Make sure that the validation file type (raidvchkset -vt <type>) is correct.
   - Make sure that the data validations are disabled for LVM configuration changes.
   - Make sure that the redo log files and data files are separated among the volumes.
   - Make sure Oracle® files with different block sizes are not mapped to the same LDEV.
   - Make sure that the syntax of the horcm.conf file is correct: there should be no duplicated TIDs or LUNs in the HORCM_DEV section.
3. If the Oracle® application has stopped abnormally and you did not find a hardware error or misconfiguration, please contact Oracle® technical support for assistance.
4. Finally, if the error cannot be resolved with the above steps, the error may have been caused by the OS, LVM, driver layer, or an intermittent hardware error. In this case, you should contact each vendor to resolve the problem. **DB Validator protects your database from even these rare error conditions, demonstrating the value of this functionality.**

Chapter 4 Troubleshooting

4.1 General Troubleshooting

For troubleshooting information on Oracle® database operations, please refer to the Oracle® user documentation, or contact Oracle® customer support.

For troubleshooting information on the CCI software, please refer to the *Hitachi Lightning 9900™ V Series and Lightning 9900™ Command Control Interface (CCI) User and Reference Guide* (MK-90RD011).

For troubleshooting information on the 9900V subsystem, please refer to the *Hitachi Lightning 9900™ V Series User and Reference Guide* (MK-92RD100).

If you need to call the Hitachi Data Systems Support Center, please see section 4.2 for instructions.

4.2 Calling the Hitachi Data Systems Support Center

If you need to call the Hitachi Data Systems Support Center, make sure to provide as much information about the problem as possible, including:

- The circumstances surrounding the error or failure,
- The exact content of any error messages displayed on the host system(s),
- The data in the CCI error log file and trace data (all files in HORCM_LOG directory),
- The remote service information messages (R-SIMs) logged on the Remote Console PC and the reference codes and severity levels of the recent R-SIMs.

The worldwide Hitachi Data Systems Support Centers are:

- Hitachi Data Systems North America/Latin America
  San Diego, California, USA
  1-800-348-4357
- Hitachi Data Systems Europe
  Contact Hitachi Data Systems Local Support
- Hitachi Data Systems Asia Pacific
  North Ryde, Australia
  011-61-2-9325-3300

Acronyms and Abbreviations

CCI	Command Control Interface
CDB	control data block
CHA	channel adapter
CU	control unit
DB	database
HA	high availability
H.A.R.D.	Hardware Assisted Resilient Data
HBA	host bus adapter
kB	kilobytes
LDEV	logical device
LU	logical unit (also called volume)
LUN	logical unit number
LUSE	LUN Expansion
LV	logical volume
LVI	logical volume image (used for mainframe volumes)
LVM	logical volume manager
MU	mirror unit (used only for ShadowImage)
OS	operating system
PV	physical volume
SIM	service information message
SVP	service processor
TID	target ID
VxVM	VERITAS® Volume Manager
WWN	worldwide name