HITACHI
Inspire the Next

Hitachi Universal Storage Platform V
Hitachi Universal Storage Platform VM
Hitachi Database Validator User's Guide

FASTFIND LINKS

Hitachi Data Systems

MK-96RD611-02

Contents

Preface

This document describes the Hitachi Database Validator feature on the Hitachi Universal Storage Platform V (USP V) and Hitachi Universal Storage Platform VM (USP VM) storage systems.

Please read this document carefully to understand this product, and maintain a copy for reference purposes. For instructions on performing Database Validator operations, see the *Command Control Interface User and Reference Guide*.

This preface includes the following information:

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

Intended Audience

This document is intended for system administrators, Hitachi Data Systems representatives, and Authorized Service Providers who are involved in installing, configuring, and operating the Hitachi Universal Storage Platform V and VM storage systems.

This document assumes the following:

- The user has a background in data processing and understands RAID storage systems and their basic functions.
- The user is familiar with the Universal Storage Platform V and/or VM storage system and has read the *Universal Storage Platform V/VM User and Reference Guide*.
- The user is familiar with the Command Control Interface (CCI) software for the Universal Storage Platform V/VM.
- The user is familiar with the Oracle® database software and the platform hosting the Oracle software.

Product Version

This document revision applies to USP V/VM microcode 60-05-0*x* and higher.

Document Revision Level

Revision	Date	Description
MK-96RD611-00	May 2008	Initial Release
MK-96RD611-01	August 2008	Revision 1, supersedes and replaces MK-96RD611-00
MK-96RD611-02	June 2009	Revision 2, supersedes and replaces MK-96RD611-01

Source Document(s) for this Revision

- mp-96rd611-01

Changes in this Revision

- Added information about troubleshooting when using Command Control Interface (see Troubleshooting When Using Command Control Interface).
- Updated Restriction and Cautions (see Requirements for the USP V/VM Storage System)
- Updated the support OS (see Requirements for the Host Server)

Document Organization

The following table provides an overview of the contents and organization of this document. Click the chapter title in the left column to go to that chapter. The first page of each chapter provides links to the sections in that chapter.

Chapter	Description
Chapter 1 - Overview of Database Validator	Provides an overview of Hitachi Database Validator.
Chapter 2 - About Database Validator Operations	Describes Hitachi Database Validator operations.
Chapter 3 - Preparing for Database Validator Operations	Describes the requirements for using Database Validator and provides instructions for installing and configuring the required software and preparing the Hitachi storage systems for Database Validator operations.
Chapter 4 - Performing Database Validator Operations	Provides information and instructions for starting Database Validator operations, describes the setup and operational information for a sample Database Validator system configuration, and provides guidelines for validation error detection and recovery.
Chapter 5 - Troubleshooting	Provides troubleshooting information for Database Validator and instructions for calling technical support.
Acronyms and Abbreviations	Defines the acronyms and abbreviations used in this document.
Index	Lists the topics in this document in alphabetical order.

Referenced Documents

Hitachi Universal Storage Platform V/VM:

- *User and Reference Guide*, MK-96RD635
- *Command Control Interface User and Reference Guide*, MK-90RD011
- *Hitachi LUN Manager User's Guide*, MK-96RD615
- *Hitachi Storage Navigator User's Guide*, MK-96RD621
- *Configuration Guide for Sun™ Solaris™ Host Attachment*, MK-96RD632
- *Configuration Guide for AIX Host Attachment*, MK-96RD636
- *Configuration Guide for HP-UX Host Attachment*, MK-96RD638
- *Configuration Guide for Windows Host Attachment*, MK-96RD639

Document Conventions

The terms "Universal Storage Platform V" and "Universal Storage Platform VM" refer to all models of the Hitachi Universal Storage Platform V and VM storage systems, unless otherwise noted.

This document uses the following typographic conventions:

Convention	Description
Bold	Indicates text on a window, other than the window title, including menus, menu options, buttons, fields, and labels. Example: Click **OK**.
Italic	Indicates a variable, which is a placeholder for actual text provided by the user or system. Example: copy *source-file target-file* ***Note:*** Angled brackets (< >) are also used to indicate variables.
`screen/code`	Indicates text that is displayed on screen or entered by the user. Example: `# pairdisplay -g oradb`
< > angled brackets	Indicates a variable, which is a placeholder for actual text provided by the user or system. Example: `# pairdisplay -g <group>` ***Note:*** Italic font is also used to indicate variables.
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a | b } indicates that you must choose either a or b.
| vertical bar	Indicates that you have a choice between two or more options or arguments. Examples: [a | b] indicates that you can choose a, b, or nothing. { a | b } indicates that you must choose either a or b.
underline	Indicates the default value. Example: [a | b]

This document uses the following icons to draw attention to information:

Icon	Meaning	Description
	Note	Calls attention to important and/or additional information.
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Caution	Warns the user of adverse conditions and/or consequences (e.g., disruptive operations).
	WARNING	Warns the user of severe conditions and/or consequences (e.g., destructive operations).

Convention for Storage Capacity Values

Physical storage capacity values (e.g., disk drive capacity) are calculated based on the following values:

1 KB = 1,000 bytes
1 MB = $1{,}000^2$ bytes
1 GB = $1{,}000^3$ bytes
1 TB = $1{,}000^4$ bytes
1 PB = $1{,}000^5$ bytes

Logical storage capacity values (e.g., logical device capacity) are calculated based on the following values:

1 KB = 1,024 (2^{10}) bytes
1 MB = 1,024 KB or $1{,}024^2$ bytes
1 GB = 1,024 MB or $1{,}024^3$ bytes
1 TB = 1,024 GB or $1{,}024^4$ bytes
1 PB = 1,024 TB or $1{,}024^5$ bytes
1 block = 512 bytes

Getting Help

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 content of any error message(s) displayed on the host system(s).
- The data in the CCI error log file and trace data (all files in HORCM_LOG directory).
- The service information messages (SIMs), including reference codes and severity levels, displayed by Storage Navigator.

The Hitachi Data Systems customer support staff is available 24 hours a day, seven days a week. If you need technical support, please call:

- United States: (800) 446-0744
- Outside the United States: (858) 547-4526

Comments

Please send us your comments on this document. Make sure to include the document title, number, and revision. Please refer to specific section(s) and paragraph(s) whenever possible.

- **E-mail:** doc.comments@hds.com
- **Fax:** 858-695-1186
- **Mail:**
  Technical Writing, M/S 35-10
  Hitachi Data Systems
  10277 Scripps Ranch Blvd.
  San Diego, CA 92131

Thank you! (All comments become the property of Hitachi Data Systems.)

1

Overview of Database Validator

Hitachi Database Validator

The Hitachi Database Validator function of the Hitachi Universal Storage Platform V (USP V) and Hitachi Universal Storage Platform VM (USP VM) is designed for the Oracle® database platform. Database Validator prevents data corruption between the database and the storage system and protects existing Oracle data on the storage system. Figure 1-1 shows a functional overview of Database Validator operations.

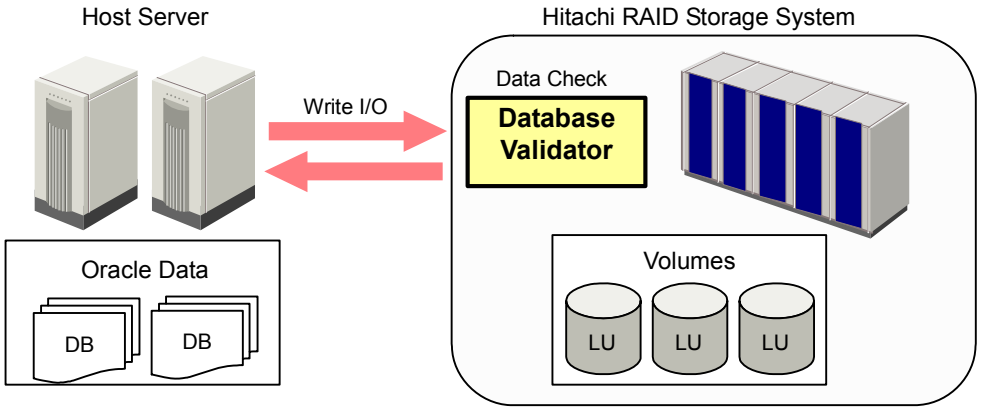


Figure 1-1 Database Validator Overview

The combination of networked storage and database management software has a risk of data corruption while writing data onto 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. Database Validator prevents corrupted data blocks generated in the database-to-storage infrastructure from being written onto the disk storage. This helps to prevent potentially disastrous corrupted data environments while at the same time minimizing risk and potential costs in backup, restore, and recovery operations.

Hitachi Database Validator performs the following functions for the user:

- **Prevents invalid Oracle data from being written to the USP V/VM:** The data stored on a storage system is transferred through many channels and may become invalid during the process. Database Validator makes sure the Oracle data is valid before it is written onto disk. If the data is invalid, Database Validator prevents the data from being written, thus achieving high-reliability of data in the USP V/VM. Database Validator also saves time and costs required for recovery when invalid data is written onto a volume.
- **Protects existing Oracle data in the USP V/VM:** Once Oracle data is stored on a volume, Database Validator protects the existing data from being deleted or updated by mistake by checking the data to be stored on that volume. For example, if another application attempts to write data onto a volume containing Oracle data, the USP V/VM will reject the write I/O and return an error.

2

About Database Validator Operations

This chapter describes Hitachi Database Validator operations.

- ☐ Hitachi Database Validator Components
- ☐ Overview of Oracle H.A.R.D. and Database Validator Operations

Hitachi Database Validator Components

Figure 2-1 shows the Hitachi Database 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 storage system, the USP V/VM front-end director (FED) initiates the data check. If invalid data is detected, the write I/O is rejected, the front-end director reports the error to CCI, and CCI outputs an error to the log (syslog).

Hitachi Database Validator is provided as a program product (license key) for the Universal Storage Platform V/VM.

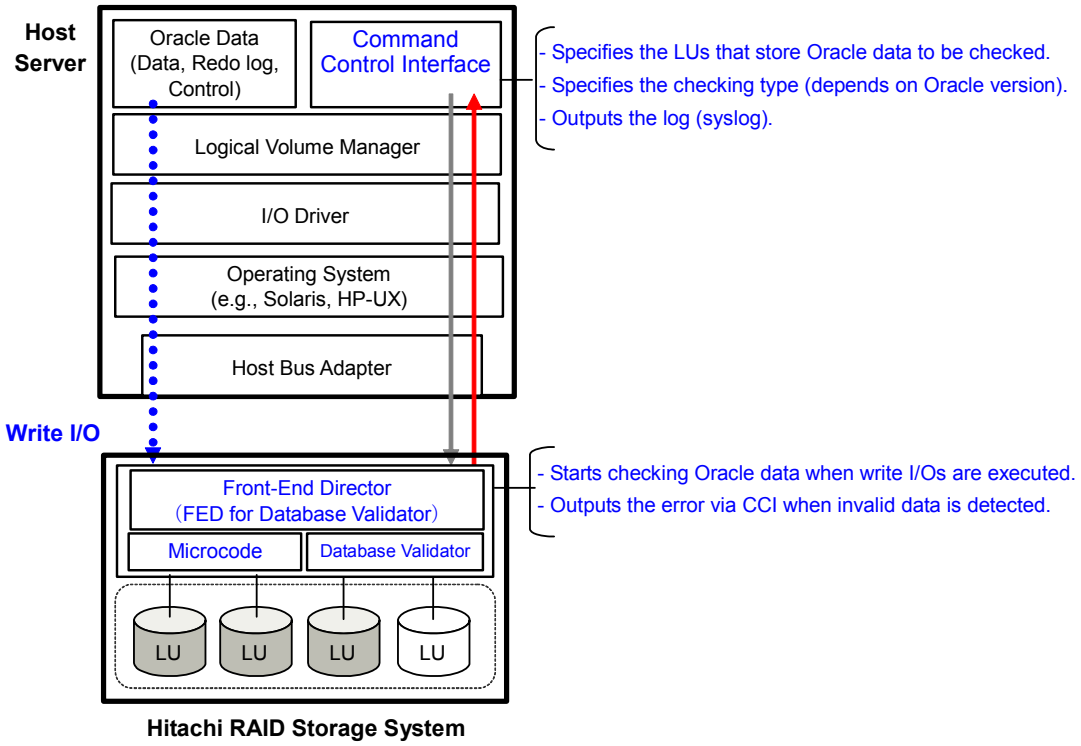


Figure 2-1 Database Validator Components

Figure 2-2 shows a typical database system configuration. As shown in this figure, there are many components between the Oracle database program and the storage system:

- 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, it is possible that all layers might pass the data without detecting the failure. The Oracle software and the USP V/VM have very robust data protection capabilities that can prevent data corruption within both the database management system (DBMS) and the storage system, but until now they could not detect data corruption that occurred between DBMS and the storage system.

The other data corruption risk is a non-Oracle application overwriting the Oracle data files on the storage system. The Database Validator function also protects existing Oracle data files from being overwritten by data from other applications.

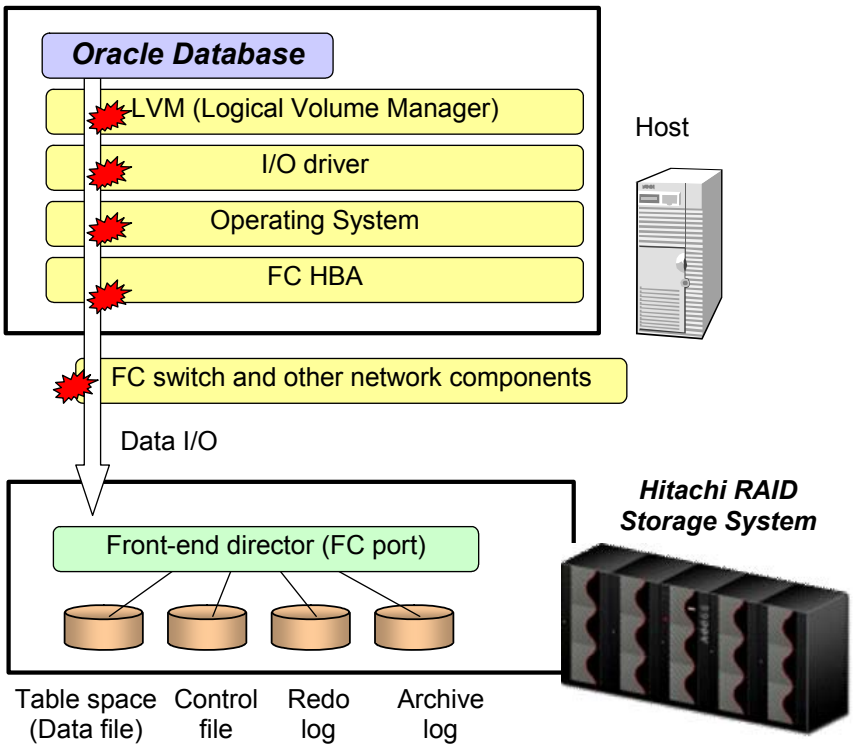


Figure 2-2 Typical Database System Configuration

Overview of Oracle H.A.R.D. and Database Validator Operations

Figure 2-3 shows a typical system configuration with Database Validator. The Oracle Hardware Assisted Resilient Data (H.A.R.D.) feature protects against data corruption. The Oracle application software computes checksums that are included with every data block written to disk. The Database Validator feature recomputes the checksums upon receiving data blocks to be written to Oracle database volumes. In the event that the Hitachi storage system receives a corrupt block, the I/O operation is rejected with a checksum failure error. The ability of the storage system 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 Database 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 systems already perform extensive checksum-type validation of data integrity from the point of entry through the entire storage system. The new Database Validator feature detects corruption occurring outside the storage system by checking write data upon receipt. The objective is to prevent corrupt data destined for Oracle volumes from being accepted by the storage system.

Detection of corrupt data: If the Database 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 USP V/VM front-end directors (FEDs) include hardware acceleration of Database 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 USP V/VM 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 Database 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 USP V/VM always performs its own internal checksum verification on reads, and hence always resends the same data originally received.)

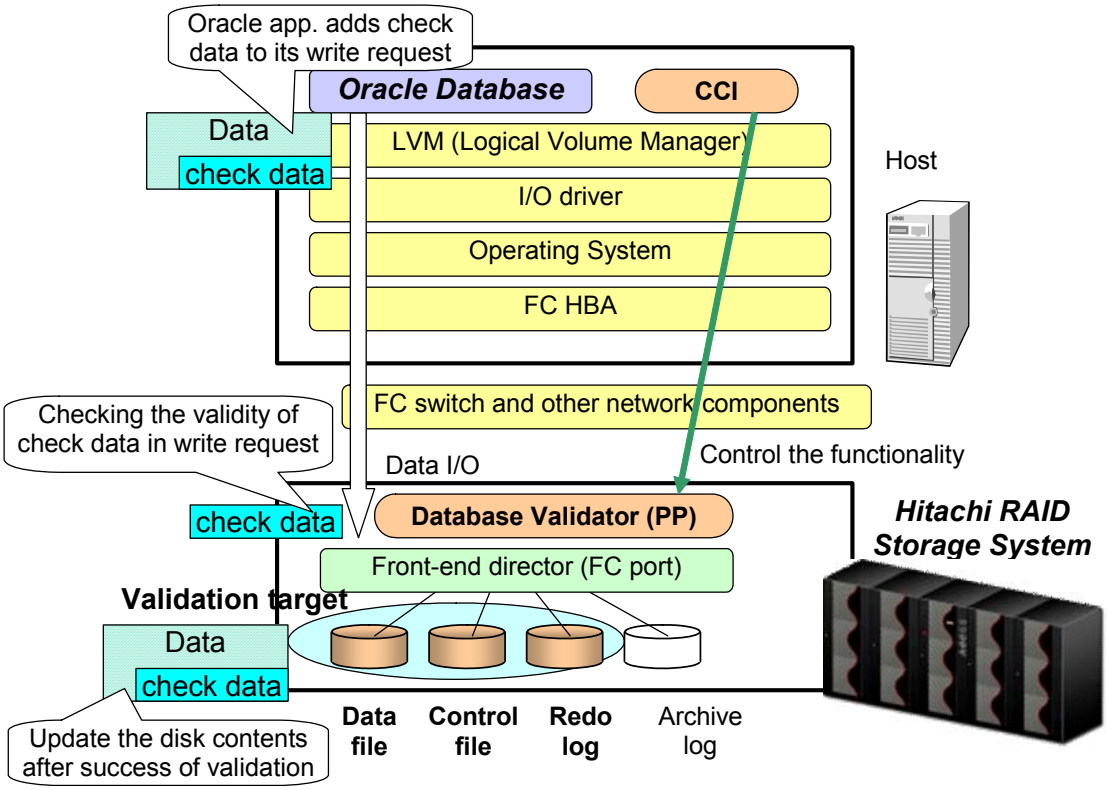


Figure 2-3 Typical Database System Configuration with Database Validator

Database Validator validates the check data at every write operation to the storage and checks out the abnormal data immediately after a failure occurs. Database 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.

3

Preparing for Database Validator Operations

This chapter describes the requirements for using Database Validator and provides instructions for installing and configuring the required software and preparing the Hitachi storage systems for Database Validator operations.

Requirements and Restrictions

To use Hitachi Database Validator, you need the following:

- USP V/VM storage system
- Host server that supports Oracle database software
- Oracle-data validation service

Requirements for the USP V/VM Storage System

For Hitachi Database Validator operations, you need a storage system that has all settings of the required hardware, microcode, and Database Validator:

- **Microcode:** The minimum microcode level for Database Validator operations on the USP V/VM is 60-03-0x.
- **Database Validator:** Hitachi Database Validator must be installed on the target USP V/VM by using the license key via Storage Navigator (or SVP). For instructions on installing Database Validator, see the *Storage Navigator User's Guide*.
- **CCI command device:** The CCI command device is required for Database Validator operations. The command device is configured using the LUN Manager software on Storage Navigator.
- **Restrictions and Cautions:**
  - Do not downgrade the USP V/VM microcode (DKCMAIN) version.
  - Before uninstalling Database Validator, you need to delete the settings for the volumes being checked by Database Validator using CCI.
  - Database Validator settings can only be deleted using CCI.
  - Before changing the configuration of the logical devices (LDEVs) (e.g., decreasing the set LDEVs, creating custom-size volumes), you need to delete the Database Validator settings using CCI, and then perform the reconfiguration operation.
  - Table 3-1 shows the range that Database Validator checks when the capacity of virtual volumes is increased by using Dynamic Provisioning.

Table 3-1 Range to be Checked for Virtual Volumes whose Capacity Increased by Using Dynamic Provisioning

Setting of Database Validator*	Range that Database Validator Checks
Set to check the specified range with LBA	Specified range with LBA
Set to check the whole range of volumes (LUs)	Whole range of increased virtual volumes. The range is enlarged automatically according to the increase of the capacity of virtual volumes.
* For details on Database Validator settings, see the *Command Control Interface User and Reference Guide*.	

Requirements for the Host Server

For Database Validator operations, you need a host server that is connected to USP V/VM with fibre-channel interface.

Note: Information on supported versions may change in future releases. For the latest information, contact your Hitachi Data Systems account team.

The host server requirements for Hitachi Database Validator operations are:

- **Command Control Interface:** Version 01-22-03/xx or later is required for Database Validator operations on USP V/VM.
- **Operating System (OS):**
  - **Sun™ Solaris versions 8 and 10**

    For further information on USP V/VM operations with the Solaris platform, see the *Sun Solaris Configuration Guide*. For further information on the Solaris OS, refer to the Solaris user documentation, or contact Sun technical support.
  - **HP-UX versions 11.0, 11i v1, 11i v2, and 11i v3**

    For further information on USP V/VM operations with the HP-UX platform, see the *HP-UX Configuration Guide*. For further information on the HP-UX OS, refer to the HP-UX user documentation, or contact HP technical support.
  - **Windows® 2000 Advanced Server (Service Pack 3 or later), Windows® 2003**

    For further information on USP V/VM operations with the Windows platform, see the *Windows Configuration Guide*. For further information on the Windows OS, refer to the Windows user documentation, or contact Microsoft technical support.
  - **AIX® 5L versions 5.1 ML3 and 5.2**

    For further information on USP V/VM operations with the AIX platform, see the *AIX Configuration Guide*. For further information on the AIX OS, refer to the AIX user documentation, or contact IBM technical support.
  - **Red Hat Enterprise Linux 4.0**

- **Oracle Database:**
  - Oracle8i™
  - Oracle9i™
  - Oracle9i™ Release 2
  - Oracle10g™ Release 2

Notes:

- The Oracle database must be constructed on raw devices (i.e., no file system).
- The **DB_BLOCK_CHECKSUM** parameter must be set to **TRUE**.
- Oracle database functionality of check data is slightly enhanced with Oracle9i Release2, so Database Validator has two distinct modes: one mode for Oracle8i and Oracle9i databases, and another mode for Oracle9i Release2 (and later) databases.

- **Logical Volume Manager:**
  - VxVM (for the Solaris OS)
  - LVM (for the HP-UX OS)
  - LVM (for the AIX OS)
- **Restrictions and Cautions:**
  - Do not write data other than Oracle data to a volume (LU) that is being checked by Database Validator.
  - If an LU being checked by Database 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 Data Validation Service

In order to use Database Validator, you need to set up the hardware and software for Database Validator in both the USP V/VM 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.

Restrictions and Cautions:

- **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). Database 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.

    Because other files (archive log files) are not the target of the validation, no special consideration for these files is required.
  - **Block size:** Do not store Oracle data of different block sizes in the same LU. Make sure that you write only the data of the same block size to the same 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 LDEVs after LVM mapping must obey the two separation rules: data/control files separate from redo log files, and same block size.
  - If LVM is used, configure the LUs before enabling Database 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 Database Validator checking can be enabled.
  - If you need to change the LVM configuration of an LU that is set as a target of Oracle data checking, first delete the settings of the target LUs and disable the Oracle check function (Database Validator), and then change the configuration. After changing the configuration, you can re-enable Database Validator and reset the LUs as targets of Oracle data checking.
  - 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 Database Validator by using the **-vs <bsize> SLBA ELBA** option.

Preparing for Database Validator Operations

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

1. **Setup and configuration:** Particular consideration is required for USP V/VM volume setup and Oracle volume mapping:
   - All requirements and restrictions in Requirements and Restrictions 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 Database Validator checking disabled.
   - Identify all LUs that will be the targets of Database Validator checking, and write down the path(s) for each LU (port, TID, LUN). Make sure that all paths use FEDs that support the Database Validator function.

   Important: Mapping information between the Oracle files and the LDEVs is very helpful for 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.

Installing and Configuring Command Control Interface

After you have prepared for Database Validator operations, you are ready to install and configure the CCI, if not already installed and configured:

- **Installing CCI:** The CCI software is installed on the host server. For instructions on installing CCI, see the *Command Control Interface User and Reference Guide*.
- **Configuring the command device:** The command device can be configured on the USP V/VM using the LUN Manager software. For instructions on configuring the command device, see the *LUN Manager User's Guide*.
- **Configuring CCI:** CCI controls Database 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. For detailed information on the CCI configuration definition file, see the *Command Control Interface User and Reference Guide*.

See Sample Implementation of Database Validator in the next chapter for an example of CCI configuration for Database Validator operations.

4

Performing Database Validator Operations

This chapter provides information and instructions for starting Database Validator operations, describes the setup and operational information for a sample Database Validator system configuration, and provides guidelines for validation error detection and recovery.

Starting Database Validator Operations

Database Validator operations are controlled through CCI commands. No other program product 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, see the *Command Control Interface User and Reference Guide*.

To start Database Validator operations, use the **raidvchkset** CCI command to enable Database 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 Sample Implementation of Database Validator for an example of configuring, starting, and performing Database 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 that 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. You must be careful when multiple paths are defined for one volume.

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

Sample Implementation of Database Validator

Figure 4-1 shows a sample Database Validator system configuration. This section presents the setup and operational information for this sample configuration:

- Volume setup
- CCI setup
- Oracle setup
- Enabling Database Validator checking
- Displaying Database Validator volumes
- Disabling and restarting Database Validator checking

This section describes the setup operations in a Solaris and VxVM environment. These operations can be easily translated to other platforms.

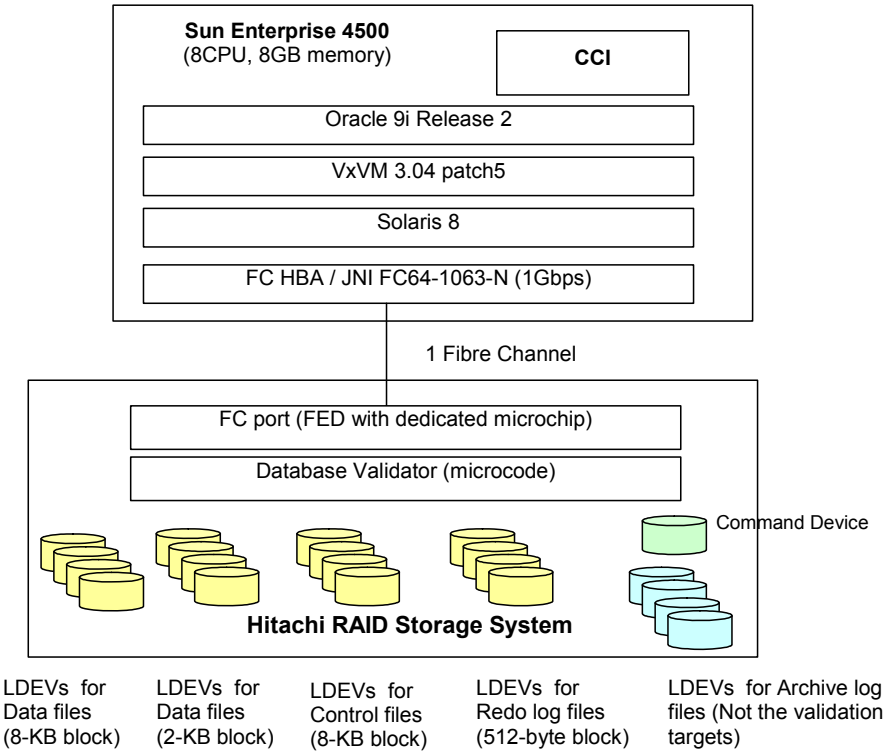


Figure 4-1 Sample Database Validator System Configuration

Volume Setup

Table 4-1 shows the volume setup for the sample configuration shown in Figure 4-1. All mapping operations must be done with Database Validator disabled.

Note:

- This document does not describe USP V/VM device configuration. See the USP V/VM Configuration Guide for the host platform (e.g., *Sun Solaris Configuration Guide*).
- This document does not describe path allocation using LUN Manager. Please refer to the *LUN Manager User's Guide.*
- This document does not describe VxVM volume configuration. Please refer to the VxVM user documentation.

Table 4-1 Volume Setup for Sample Configuration

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

***** Type refers to the size, device emulation, sharing conditions, etc. If the 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 path definition settings of USP V/VM, OS internal rule, and HBA device driver.

CCI Setup

Figure 4-2 shows the **horcm.conf** file used for the sample configuration shown in Figure 4-1. All CCI operations in the following sections use this **horcm.conf** file:

- HORCM_MON and HORCM_INST are parameters used by CCI to monitor 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 USP V/VM 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 USP V/VM.

```
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 4-2 CCI Configuration Definition File (horcm.conf) for Sample Configuration

Oracle Setup

The following steps must be performed before Database 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**.

Enabling Database Validator Checking

Figure 4-3 shows the CCI **raidvchkset** commands used to enable Database Validator for the sample configuration shown in Figure 4-1:

- 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 4-2.
- 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 Database 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 4-3 Enabling Database Validator Checking for Sample Configuration

Table 4-2 CCI Parameters for File Types and Oracle Versions

Oracle Version	File Type	Parameter for -vt Option
Oracle9i Release 2	Data file Control file	data9
	Redo log file	redo9
Oracle9i prior to Release 2 Oracle8i	Data file Control file	data8
	Redo log file	redo8

Enabling Database 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. See 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 4-4 is an example of using the 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 4-4 Example of Enabling Database Validator Checking with LVM Consideration

Displaying Database Validator Volumes

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

- **inqraid**
- **raidvchkdsp**
- **raidvchkscan**

Displaying Database Validator Volumes using inqraid

To verify the validation volumes, you can use the **inqraid –fp** command with the **–CLI** option (see Figure 4-5). Database Validator checking is enabled for the volumes with an asterisk (*). For further information on the **inqraid** command, see 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-V
c4t0d1*        CL1-E    15005    01   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d2*        CL1-E    15005    02   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d3*        CL1-E    15005    03   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d4*        CL1-E    15005    04   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d5*        CL1-E    15005    05   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d6*        CL1-E    15005    06   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d7*        CL1-E    15005    07   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d8*        CL1-E    15005    08   -  s/s/ss  0004 5:01-03 OPEN-V
c4t0d9         CL1-E    15005    09   -  s/s/ss  0004 5:01-03 OPEN-V
```

Figure 4-5 Displaying Database Validator Volumes for Sample Configuration

Displaying Database Validator Volumes using raidvchkdsp

The **raidvchkdsp** command (see Figure 4-6) 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 *Command Control Interface User and Reference Guide* 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 4-6 Raidvchkdsp Command Examples showing -fd and -v cflag Options

Displaying Database Validator Volume using raidvchkscan

The **raidvchkscan** command (see Figure 4-7) displays the fibre port of the USP V/VM, target ID (TID), LDEV mapped for LUN# and the parameters for validation checking, regardless of the configuration definition file.

- See the *Command Control Interface User and Reference Guide* 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 4-7 Raidvchkscan Command Example with -v cflag Option

Disabling and Restarting Database Validator Checking

You can disable Database Validator checking at the volume level using the **raidvchkset –vt** command with no parameter (see Figure 4-8). 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 4-8 Disabling Database 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 4-9 Restarting Database Validator Checking for Sample Configuration

Creating the Database

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

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.

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 that cause data corruption are also detected by Database Validator immediately, thus minimizing the damage from error and the cost for recovery.

Error Detection

When the USP V/VM detects a validation error, the USP V/VM 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 USP V/VM outputs the error information in three ways: service information message (SIM), syslog (on the host), USP V/VM-internal sense byte (SSB):

- **SIM:** Users can view the SIMs using the Storage Navigator software.

Note: The default USP V/VM setting is that SIM reporting for Database Validator is not enabled.

- **Syslog:** CCI writes a syslog message that includes the error volume number and error counters for each validation item. CCI polls the 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 4-10 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 4-10 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 (USP V/VM 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.

Investigating Error Causes

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 that 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, check the following. Validation parameters can be investigated using the CCI **raidvchkdsp** command:

   - Ensure that the block size (raidvchkset -vs <size>) is correct.
   - Ensure that the validation file type (raidvchkset -vt <type>) is correct.
   - Ensure that the data validations are disabled for LVM configuration changes.
   - Ensure that the redo log files and data files are separated among the volumes.
   - Ensure Oracle files with different block sizes are not mapped to the same LDEV.
   - Ensure 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, 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. **Database Validator protects your database from even these rare error conditions, demonstrating the value of this functionality.**

5

Troubleshooting

This chapter provides troubleshooting information for Database Validator and instructions for calling technical support.

Troubleshooting

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

CCI: For troubleshooting information on the CCI, see the *Command Control Interface User and Reference Guide*.

USP V/VM: For troubleshooting information on the USP V/VM storage system, see the *User and Reference Guide*.

Troubleshooting When Using Command Control Interface

When an error has occurred in Database Validator operation using CCI, you might identify the cause of the error by referring to the log displayed on the CCI window or the CCI operation log file.

The CCI operation log file is stored in the following directory by default:

/HORCM/log*/curlog/horcmlog_HOST/horcm.log

where

- * is the instance number.
- HOST is the host name.

To identify the error code using the log file:

1. Open the CCI log file, and find the error code.

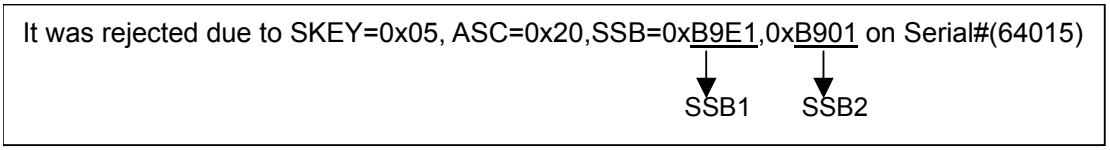
   Example: 11:06:03-37897-10413- SSB = 0xB901,4A96

   Error codes appear on the right of the equal symbol (=). The alphanumeric characters of last four digits on the left of the comma (,) indicates SSB1 (e.g., B901), and the right of the comma (,) indicates SSB2 (e.g., 4A96).
2. See Table 5-1 and find the meaning of the error code.

   For details about the error codes that are not described in Table 5-1, contact the Hitachi Data Systems Support Center (see Calling the Hitachi Data Systems Support Center).

To identify the error code using the log displayed on the CCI window:

1. Find the error code from the logs displayed on the CCI window. For example:



   Error codes appear on the right of "SSB=". The alphanumeric characters of last four digits on the left of the comma(,) indicates SSB1 (e.g., B9E1), and the alphanumeric characters of last four digits on the right of the comma (,) indicates SSB2 (e.g., B901).
2. Refer to Table 5-1 and find the meaning of the error code.

Table 5-1 Error Code and Error Contents When Operating CCI

Error Code (SSB1)	Error Code (SSB2)	Description
B9B0	B9B3	The command was rejected because the setting information of DataBase Validator was incorrect.
	B9B4	The command was rejected because the LU being used by Database Validator is not defined.
	B9B5	The command was rejected because Database Validator program product is not installed.
	B9B6	The command was rejected because the microcode which supports Database Validator and the microcode which does not support Database Validator exist together.
	B9B7	The command was rejected because the LU being used by Database Validator is a command device.
B9B1	B9B4	The command was rejected because the LU being checked by Database Validator is not defined.

For details about error codes that are not described in Table 5-1, contact the Hitachi Data Systems Support Center (see Calling the Hitachi Data Systems Support Center).

For more information about CCI, see the Hitachi Command Control Interface (CCI) User and Reference Guide.

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 service information messages (R-SIMs), including reference codes and severity levels, displayed by Storage Navigator.

The Hitachi Data Systems customer support staff is available 24 hours a day, seven days a week. If you need technical support, please call:

- United States: (800) 446-0744
- Outside the United States: (858) 547-4526

Acronyms and Abbreviations

ALPA	arbitrated loop-physical address
CCI	Command Control Interface
CDB	control data block
CHA	channel adapter (another name for front-end director)
CTG	consistency group
CU	control unit
DB	database
DBMS	database management system
FED	front-end director
GB	gigabyte (see Convention for Storage Capacity Values)
HA	high availability
H.A.R.D.	Hardware Assisted Resilient Data
HBA	host bus adapter
KB	kilobyte (see Convention for Storage Capacity Values)
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
MB	megabyte (see Convention for Storage Capacity Values)
MU	mirror unit (used only for ShadowImage)
OS	operating system
PB	petabyte (see Convention for Storage Capacity Values)
PV	physical volume
RAID	redundant array of independent disks
SIM	service information message
SSB	sense byte

SSID	storage system identification
SVP	service processor
TB	terabyte (see Convention for Storage Capacity Values)
TID	target ID
USP V	Hitachi Universal Storage Platform V
USP VM	Hitachi Universal Storage Platform VM
VM	Volume Manager
WWN	worldwide name

Index

Hitachi Data Systems

Corporate Headquarters
750 Central Expressway
Santa Clara, California 95050-2627
U.S.A.
Phone: 1 408 970 1000
www.hds.com
info@hds.com

Asia Pacific and Americas
750 Central Expressway
Santa Clara, California 95050-2627
U.S.A.
Phone: 1 408 970 1000
info@hds.com

Europe Headquarters
Sefton Park
Stoke Poges
Buckinghamshire SL2 4HD
United Kingdom
Phone: + 44 (0)1753 618000
info.eu@hds.com

Hitachi Data Systems

MK-96RD611-02