# A raidcom cookbook for GSS personnel

Steve Burr

V3, June 2012

# Introduction

With the addition of the Virtual Storage Platform (VSP) to the Hitachi Data Systems storage product range we gain a powerful new tool for scripting storage management. This links with Hitachi’s goal of improving usability and cost of ownership of our storage. For GSS, I think this could be one of the most important developments of the VSP. I think **raidcom** will be used a great deal, especially by GSS and advanced customers. For GSS we will be able to perform projects faster and more accurately and I especially see it valuable generating reports and other data analyses and generally improving communication with the customer.

To enable advanced storage management through scripting the **raidcom** command has been added to Raid Manager (from version 01-24-xx). Unlike previous in-band CLI capabilities (CLIEX), **raidcom** has very wide functionality. As well as the usual LUN mapping you can perform tasks such as LDEV and pool creation, device formatting, MCU configuration and external storage management. This single command allows access to **almost all management features of the Virtual Storage Platform**, thisis a radical change. One of Hitachi’s design goals for the Virtual Storage Platform was that anything that can be managed through Storage Navigator 2 can also be managed through Raid Manager and vice versa. As a challenge, during my lab testing, I attempted to perform **all** storage management through the new CLI. With a few small exceptions I was successful.

The new storage management command line interface is an extension to Raid Manager. It is installed automatically as part of Raid Manager and uses a command device (or the new virtual command device q.v.) in the usual way. Most of Raid Manager is however identical to previous versions (and still backward compatible with the 7700E!)

Raidcom is very fast. You need to be aware of a key aspect of **raidcom**, which it shares with Storage Navigator 2 and CLIEX its predecessor: changes made by these methods, although fast, will not be recognized by Hitachi Device Manager (HDvM) and some sort of refresh will be required to use the HDvM database. Fortunately, there have been improvements in the time needed to refresh HDvM.

As well as implementing most management capabilities, raidcom includes the following, oft asked for features: check only mode, consistency check mode, batch consistency and out-of-band operation.

## Goal

This is not intended as a complete beginning to end manual. There are actually now three Raid Manager manuals, each documenting different aspects: installation and configuration, a user guide and a command reference. Instead, I am trying to document many examples of **raidcom** in real use and include some lessons learned in my testing. I will try to include a few more complex examples. All the enclosed have been tested in my “configure the box through scripting” project. Knowledge of Raid Manager is probably essential.

## Scope

This document covers information concerning the V01 release of VSP and raidcom. Raidcom also has limited functionality on USP V; I document a few of these in the text but this in not complete (please contribute, this is a community document). A few commands even work on USP.

There are other resources you may wish to look at in conjunction with this:

[Scripted Storage Management with the Hitachi Virtual Storage Platform 1.0.pptx](http%3A%2F%2Floop.hds.com%2Fdocs%2FDOC-1330), this presentation describes the new scripting capabilities that I think are going to radicalize the way we deliver services.

[Virtual Storage Platform for GSS BC and HDT 1.0.pptx](http%3A%2F%2Floop.hds.com%2Fdocs%2FDOC-1367), this presentation covers a little of the material in this paper for Business Continuity and some aspects of the new feature HDT.

[Hitachi Virtual Storage Platform for GSS Business Continuity.docx](http%3A%2F%2Floop.hds.com%2Fdocs%2FDOC-1417), this white paper covers business continuity aspects of the Virtual Storage Platform.

### Confidentiality

This document was written with an internal HDS audience in mind, specifically GSS Consultants. I expect it to be of value to Support and some Pre-Sales staff as well. It is not written for an external audience and has not been review for external distribution.

# Important background concepts

Feel free to skip forward to the examples by class but to understand raidcom fully there are several new concepts to introduce. These are:

* Background execution of “long” tasks
* New security in Raid Manager
* Out-of-band operation
* Command device creation and security
* Resource Naming

## Background execution of tasks

Nearly all management activities initiated through the GUI are now executed as a task run in the background, even simple rapid changes like password change and port mode changes. This means more work can be done but it is possible to fool yourself that a task has: a) finished or b) has succeeded. This “execution in parallel” may be a little irritating, but the upside is huge: you can have multiple Storage Navigator sessions logged in and changing the configuration at the same time, and never need to log out if you chose not to. To prevent problems you may lock the resources of the VSP from other users. But this is not enforced.

Some **raidcom** configuration changes are also executed as background tasks but not all like SN2; mainly longer tasks like formatting. This must be considered in correct scripting: E.G. a create Pool or create LDEV task would normally return a valid return code but the action is neither complete nor is it necessarily a valid request; it could fail later. As mentioned elsewhere, another user could grab the resource before you. Additional scripting features (**raidcom get|reset command\_status**) are provided for this.

You may wish to know [what commands are asynchronous?](#_Hlk278199964)

## Security

Obviously, greatly increasing the capabilities available through the command device introduces a security challenge. When introducing such capabilities it was important to improve security. On the Virtual Storage Platform we have new security capabilities: the command device has additional access rights and will not accept **raidcom** commands unless an authorized storage navigator user and password is used. If you wish, this may now be authenticated on corporate security servers using LDAP, Radius or Active Directory. This feature is not only used for raidcom but storage navigator too; we will need to improve GSS skills in using LDAP, Active Directory, Kerberos etc.

This security method is not available on the USP V, **raidcom** query commands are permitted but those which modify storage are blocked. There is a method for permitting storage modification but this is currently restricted in use.

## Out-of-band operation

One limitation of Raid Manager was that it required a command device and that meant installing and configuring associated fiber cabling, SAN, server and HBA. At the early stage of a project especially this would be restricting. The new Out-Of-Band command device (also known as virtual command device) facility permits you to use Raid Manager without a command device, SAN or server. An IP connection to the SVP can be used instead. This links the HORCM instance to the SVP to perform storage management. The out-of-band command device requires login but has full functionality.

This has great possibilities for remote management and improved “early installation” activities. However, it has security implications. The out-of-band command device cannot be used without login. Do not assume that customers will permit such usage; I know many sites where SVP access is in a highly restricted LAN, also it potentially widely increases the knowledge and availability of storage navigator access. Out-of-band command device is significantly slower (5-10 times) than in-band. See [Specifying OUT-OF-BAND in HORCM files](#_Hlk278201235) for usage.

## Command device creation and security

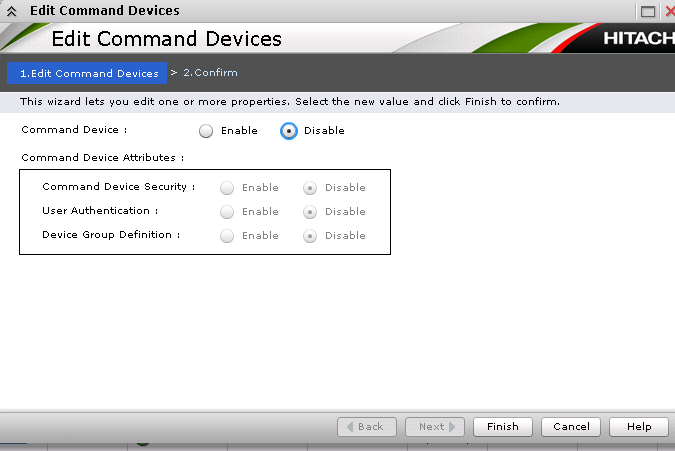
Creation of command devices is in the Logical Devices section of the VSP Storage Navigator 2. Creation of a command device with the rights to manage storage can be done through Storage Navigator 2, or through **raidcom**. With storage models before the VSP there were two levels of security on command devices:

* Command device without Command Device Security: an all-functions command device (although in reality this actually only permitted replication management functions)
* Command device with Command Device Security: the same as the above but with the restriction that you could only manage devices actually connected to the server you were running Raid manager on.

The VSP adds two new permissions to the command device.

* User authentication: this permission is a little confusing, when enabled the command device cannot be used unless the user logs in using a storage navigator user and password with the correct privilege using **raidcom login** or similar. However, this has a major side-effect; **raidcom** commands **cannot** be used on a VSP without user authentication enabled and authenticated. So the “user authentication” right is also the “enable raidcom” right.
* Device Group Definition: One of the new capabilities of VSP is that device and copy group information, previously stored in a HORCM file, can instead be stored on the DKC. This new privilege allows you to manage and use this capability.

The permissions are controlled by this screen:



A “full featured” device requires the above to be: **true, false, true, true**. A traditional USP V device would be: **true false, false, false**.

# Resource naming

One of the most powerful features added to Hitachi Device Manager 6.0 was LDEV naming. Before that, outside of the little used resource groups, it was impossible to link information about storage with management information outside the storage. You could not link physical configuration with your organizational structure, reporting or design and management documentation.

Most solutions to this were home-brewed often linking together spreadsheet of hexadecimal numbers. This capability has been greatly improved on the Virtual Storage Platform. VSP now supports object naming on the DKC. The information is stored in control memory and is protected along with everything else; it should not be possible for it to be lost. Virtual Storage Platform extends the initial naming concept and all the following can be named. Some of these are single objects, others group object together:

* LDEV and V-VOLs
* Host Groups
* WorldWide Names
* Pools
* Device Groups
* Copy Groups

These names can be used for identification, linking with external information and for management short-cuts. Groups are a more powerful concept than the simple names like LDEV and WWN, but require more setup. A device group is a named group of LDEVs, you can create them, map them and replicate them as a single object. Copy groups are pairs of device groups used for replication. Command Line shortcuts are available for working on these group objects at one time. Importantly, being stored on the DKC there is a single source for them. One of the challenges with the use of the HORCM file for grouping is that it is possible to get HORCM files out of date or incompatible, with in-DKC groups this is much harder.

## Naming schemes

This leaves an open question – what is the best naming scheme. This is a challenge for the future. Already we have seen the value of naming in our lab environment; VSP resources are all named including the creator. This solves our most common lab management problem: “who did this and can I safely delete it”, we can then implement the simple but powerful scheme “label it or lose it”. Obviously the scheme for a lab is different from a scheme for a corporation. Information you might include in a naming scheme might be: organizational group, project, application, data type, data class or tier, pool and capacity.

However, it is worth observing: the reason you might wish to include some data: raid-level, data type, tier, pool and capacity in the name was because this information used to be hard to obtain (I always found it incredible you could not get LUN size through Raid Manager). But this limitation no longer exists, see Reporting ldev information

## LDEV and V-VOL naming

Just as they can be named on Device Manager you may name LDEVS. This includes virtual LDEVs used for HDP, HDT and COW. This is probably the most useful part of naming. At the moment (V01 VSP+ HCS 7) Device Manager naming is not linked to DKC naming in any way.

## Host group naming

Host Groups have always been named on previous storage, and unlike LDEVs, the names visible in Device Manager are the same as those in the DKC. Although they could be viewed and changed in the past they had no extra value. Now, Host Group names can be used throughout CLI operation.

##### Use of Host Group names instead of numbers:

While port assignment for a server is relatively easy to remember, Host Group number is much harder, especially if they differ between odd and even ports. I repeat the mapping part of the earlier example using Host Group name instead:

**raidcom add lun -port CL3-A SWB\_BC-008 –ldev\_id 0xBD01**

Note: I am allowed to specify lun number (-lun\_id 42) but if not specified it allocates the next free slot.

## Worldwide names

These have also always been named. Raidcom allows you to set and view these. It is best practice for WWN names to be the same as hostgroup name with an extension.

## Pool naming

Pool naming is now supported. In some CLI commands you can refer to a pool by name or by number.

## Device Group naming

As well as individual names for LDEVs we can group several together as a Device Group. There are two uses of device groups. Use for provisioning and use for replication. For a device group to be suitable for replication it is more restricted: specifically it must follow HORCM file rules, each device name in a device group must be unique. Groups for provisioning may have repeated device names.

Warning: LDEV, device group and copy group naming in the DKC is very interesting but is not yet supported by HRpM or HDvM replication management: this is a technology to watch once they become integrated.

### Use of device groups:

The functions for device\_groups are **get, delete,** and **add**. Add and delete may also be used to modify a device group as they can: create a new group, add to and remove from a group, and rename devices. Add and delete functions support operation on multiple devices (with <from>-<to> or <from> **-cnt** <num>) but this leads to non-unique device names. These device groups will not be rejected but can only be used for provisioning, not replication. The group cannot be used in a copy group. Let’s do some examples on just four LUNs 0xbd12- 0xbd15. First some provisioning:

The **add lun** function maps one device at a time, if a lot of LUs are being mapped this can be tiresome. Although simple to script in a loop you can also leverage **device groups**. We wish to map four LDEVs to two Host groups.

Create a device group, note the use of **–cnt 4**, this selects four consecutive LDEVs to group:

**raidcom add device\_grp -device\_grp\_name SWBDG1 DEV1 -ldev\_id 0xbd12 -cnt 4**

**raidcom get device\_grp -device\_grp\_name SWBDG1**

LDEV\_GROUP LDEV\_NAME LDEV# Serial#

SWBDG1 DEV1 bd12 53086

SWBDG1 DEV1 bd13 53086

SWBDG1 DEV1 bd14 53086

SWBDG1 DEV1 bd15 53086

Map device group to two ports:

**raidcom add lun -port CL3-A SWB14 -grp\_opt ldev -device\_grp\_name SWBDG1 DEV1**

**raidcom add lun -port CL4-A SWB\_BC-14 -grp\_opt ldev -device\_grp\_name SWBDG1 DEV1**

**raidcom get lun -port CL3-A SWB\_BC-014 -fx**

**PORT GID HMD LUN NUM LDEV CM Serial# HMO\_BITs**

CL3-A 6 LINUX/IRIX 0 1 401 CM 53086 40

CL3-A 6 LINUX/IRIX 1 1 bd12 - 53086 40

CL3-A 6 LINUX/IRIX 2 1 bd13 - 53086 40

CL3-A 6 LINUX/IRIX 3 1 bd14 - 53086 40

CL3-A 6 LINUX/IRIX 4 1 bd15 - 53086 40

Such a group can be used for many provisioning functions including LU mapping and unmapping and management of pools and journals and some ldev functions.

##### Use of device groups for replication:

The above group will not work for replication, we need unique device names. To illustrate replication we will create two groups for our four LUNs, one group for primary and one for secondary.

First the P-VOLs:

**raidcom add device\_grp -device\_grp\_name P\_SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_1 -ldev\_id 0xbd12**

**raidcom add device\_grp -device\_grp\_name S\_SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_1 -ldev\_id 0xbd14**

**raidcom add device\_grp -device\_grp\_name P\_SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_2 -ldev\_id 0xbd13**

**raidcom add device\_grp -device\_grp\_name S\_SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_2 -ldev\_id 0xbd15**

**raidcom add device\_grp -device\_grp\_name SWB\_BC014\_P1 DEVSI\_1\_1 -ldev\_id 0xbd12**

**raidcom add device\_grp -device\_grp\_name SWB\_BC014\_P1 DEVSI\_1\_2 -ldev\_id 0xbd13**

Then the S\_VOLs:

**raidcom add device\_grp -device\_grp\_name SWB\_BC014\_S1 DEVSI\_1\_1 -ldev\_id 0xbd14**

**raidcom add device\_grp -device\_grp\_name SWB\_BC014\_S1 DEVSI\_1\_2 -ldev\_id 0xbd15**

**raidcom get device\_grp -device\_grp\_name SWB\_BC014\_P1**

LDEV\_GROUP LDEV\_NAME LDEV# Serial#

SWB\_BC014\_P1 DEVSI0\_1\_1 48402 53086

SWB\_BC014\_P1 DEVSI0\_1\_2 48403 53086

**raidcom get device\_grp -device\_grp\_name SWB\_BC014\_S1**

LDEV\_GROUP LDEV\_NAME LDEV# Serial#

SWB\_BC014\_S1 DEVSI0\_1\_1 48404 53086

SWB\_BC014\_S1 DEVSI0\_1\_2 48405 53086

Things to Note:

* We have to add each device individually; **-cnt** cannot create different device names. Obviously a loop would be useful.

### Device and Copy Group naming with HORCM: ShadowImage or COW

Replication copy group definitions can be stored in the Virtual Storage Platform instead of in a HORCM file. We will continue our example of setting up a ShadowImage pair:

**raidcom add copy\_grp -copy\_grp\_name SWB\_BC-014\_A\_S0 P\_SWB\_BC-014\_A\_S0 S\_SWB\_BC-014\_A\_S0 -mirror\_id 0**

**raidcom get copy\_grp | grep -E "COPY\_GROUP|SWB\_BC-014\_A\_S0"**

**COPY\_GROUP LDEV\_GROUP MU# JID# Serial#**

SWB\_BC-014\_A\_S0 P\_SWB\_BC-014\_A\_S0 h0 - 53086

SWB\_BC-014\_A\_S0 S\_SWB\_BC-014\_A\_S0 - - 53086

We now need to use these groups. We create two HORCM files, but with the new HORCM\_LDEVG format they can be reduced to fewer lines:

|  |  |
| --- | --- |
| **horcm6.conf** | **horcm7.conf** |
| HORCM\_CMD  \\.\CMD-53086:/dev/sd | HORCM\_CMD  \\.\CMD-53086:/dev/sd |
| HORCM\_LDEVG  SWB\_BC-014\_A\_S0 P\_SWB\_BC-014\_A\_S0 53086 | HORCM\_LDEVG  SWB\_BC-014\_A\_S0 S\_SWB\_BC-014\_A\_S0 53086 |
| HORCM\_INST  SWB\_BC-014\_A\_S0 localhost 31008 | HORCM\_INST  SWB\_BC-014\_A\_S0 localhost 31007 |

Of course this isn’t much of a reduction. If there were more devices the saving would be greater. But this is to neglect the other advantage of DKC copy groups – there can only be one copy of the copy definition, changes are immediately available for all to update from. Our copy group and HORCM are set up so I can start HORCM to see it:

**export HORCMINST=6**

**horcmstart.sh**

**raidqry -g -ISI**

GNo Group RAID\_type IV/H IV/M MUN/H MUN/M

1 SWB\_BC-014\_A\_S0 HTC\_RAID 18 13 4 64

It is managed in the usual way:

**pairdisplay -g SWB\_BC-014\_A\_S0 -CLI -ISI**

Group PairVol L/R Port# TID LU-M Seq# LDEV# P/S Status Seq# P-LDEV# M

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_1 L CL3-A-6 0 1 0 53086 48402 SMPL - - - -

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_1 R CL3-A1-6 0 3 0 53086 48404 SMPL - - - -

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_2 L CL3-A-6 0 2 0 53086 48403 SMPL - - - -

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_2 R CL3-A1-6 0 4 0 53086 48405 SMPL - - - -

Here we create pairs. I use a new extension to Raid Manager **–pvol <device group>** Instead of the usual **–vl** or **–vr** (which are still available but the new format is better).

**paircreate -g SWB\_BC-014\_A\_S0 –ISI -pvol P\_SWB\_BC-014\_A\_S0**

**pairdisplay -g SWB\_BC-014\_A\_S0 -CLI -ISI**

Group PairVol(L/R) (Port#,TID, LU-M) ,Seq#,LDEV#.P/S,Status, Seq#,P-LDEV# M

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_1 L CL3-A-6 0 1 0 53086 48402 P-VOL PAIR 53086 48404 -

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_1 R CL3-A1-6 0 3 0 53086 48404 S-VOL PAIR - 48402 -

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_2 L CL3-A-6 0 2 0 53086 48403 P-VOL PAIR 53086 48405 -

SWB\_BC-014\_A\_S0 SWB\_BC-014\_A\_S0\_2 R CL3-A1-6 0 4 0 53086 48405 S-VOL PAIR - 48403 -

The **–pvol** feature helps with a common customer question. We often meet customers who like the asymmetry present in products like SRDF. They like this because it will be harder to create pairs “the wrong way”. With device and copy group naming we can name data separately as primary and DR and such an error will at least be obvious. We gain this feature without losing the flexibility of a totally symmetric replication implementation[[1]](#footnote-1).

NOTE: if you change copy groups or device groups then you must restart HORCM for the changes to be seen.

### Device and Copy Group naming with HORCM: TrueCopy or UR

We create device groups in an identical way with slight naming changes. But here we create groups on two or more different subsystems. Note: in the examples, I have primary and secondary command devices connected to a single server and use a HORCM file to manage that can manage both DKC.

**raidcom add device\_grp -s 53086 -device\_grp\_name P\_SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_1 -ldev\_id 0xbd12**

**raidcom add device\_grp -s 53004 -device\_grp\_name S\_SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_1 -ldev\_id 0xbd12**

**raidcom add device\_grp -s 53086 -device\_grp\_name P\_SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_2 -ldev\_id 0xbd13**

**raidcom add device\_grp -s 53004 -device\_grp\_name S\_SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_2 -ldev\_id 0xbd13**

We add two copy groups, one for each “site”. For UR we can add MU and Journal information to the copy group.

**raidcom add copy\_grp -s 53086 -copy\_grp\_name SWB\_BC-014\_A\_U1 P\_SWB\_BC-014\_A\_U1 \**

**–mirror\_unit 1 –journal\_id 1**

**raidcom add copy\_grp -s 53004 -copy\_grp\_name SWB\_BC-014\_A\_U1 S\_SWB\_BC-014\_A\_U1 \**

**–mirror\_unit 1 –journal\_id 1**

We add the HUR group to the HORCM files and restart. [Leveraging that both command devices are visible to keep to two files]

|  |  |
| --- | --- |
| **horcm6.conf** | **horcm7.conf** |
| HORCM\_CMD  \\.\CMD-53086:/dev/sd | HORCM\_CMD  \\.\CMD-53004:/dev/sd  \\.\CMD-53086 |
| HORCM\_LDEVG  SWB\_BC-014\_A\_U1 P\_SWB\_BC-014\_A\_U1 53086  SWB\_BC-014\_A\_S0 P\_SWB\_BC-014\_A\_S0 53086 | HORCM\_LDEVG  SWB\_BC-014\_A\_U1 S\_SWB\_BC-014\_A\_U1 53004  SWB\_BC-014\_A\_S0 S\_SWB\_BC-014\_A\_S0 53086 |
| HORCM\_INST  SWB\_BC-014\_A\_U1 localhost 31008  SWB\_BC-014\_A\_S0 localhost 31008 | HORCM\_INST  SWB\_BC-014\_A\_U1 localhost 31007  SWB\_BC-014\_A\_S0 localhost 31007 |

**paircreate -g SWB\_BC-014\_A\_U1 -pvol P\_SWB\_BC-014\_A\_U1 -ITC -f async**

**pairdisplay -g SWB\_BC-014\_A\_U1**

Group PairVol(L/R) (Port#,TID, LU),Seq#,LDEV#.P/S,Status,Fence,Seq#,P-LDEV# M

SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_1(L) (CL3-A-6, 0, 1)53086 48402.P-VOL COPY ASYNC ,53004 48402 -

SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_1(R) (CL3-A-6, 0, 1)53004 48402.S-VOL COPY ASYNC ,----- 48402 -

SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_2(L) (CL3-A-6, 0, 2)53086 48403.P-VOL COPY ASYNC ,53004 48403 -

SWB\_BC-014\_A\_U1 SWB\_BC-014\_A\_U1\_2(R) (CL3-A-6, 0, 2)53004 48403.S-VOL COPY ASYNC ,----- 48403 -

Benefits of this new approach include:

1. Changes to the group composition are more easily accommodated
2. MU numbers are assigned permanently to the copy group
3. Journal numbers are assigned permanently to the copy group
4. We name primary and secondary differently so are less likely to copy the wrong way [For additional protection research DRU S-VOL disable].

### Copy Group naming: ShadowImage and COW

Using this feature it is possible to have a single HORCM file with just two lines to manage all in-system replication pairs in a DKC. If your HORCM file contains no HORCM\_LDEVG section then HORCM will attempt to read the copy groups from the DKC to populate HORCM. In the ShadowImage example above, if we use this HORCM file:

|  |
| --- |
| **horcm6.conf** |
| HORCM\_CMD  \\.\CMD-53086:/dev/sd |

Then we can start HORCM and it will read the copy groups automatically. Here we use raidqry –g to list the copy groups usable:

**raidqry -g -ISI2**

GNo Group RAID\_type IV/H IV/M MUN/H MUN/M

1 SWB\_BC014\_SI1 HTC\_RAID 18 13 4 64

It is then managed in the usual way:

**pairdisplay -g SWB\_BC014\_SI1 –ISI6**

Group PairVol(L/R) (Port#,TID, LU-M) ,Seq#,LDEV#.P/S,Status, Seq#,P-LDEV# M

SWB\_BC014\_SI1 DEVSI\_1\_1(L) (CL3-A-6, 0, 1-0 )53086 48402.SMPL ----,----- ----- -

SWB\_BC014\_SI1 DEVSI\_1\_1(R) (CL3-A-6, 0, 3-0 )53086 48404.SMPL ----,----- ----- -

SWB\_BC014\_SI1 DEVSI\_1\_2(L) (CL3-A-6, 0, 2-0 )53086 48403.SMPL ----,----- ----- -

SWB\_BC014\_SI1 DEVSI\_1\_2(R) (CL3-A-6, 0, 4-0 )53086 48405.SMPL ----,----- ----- -

You may have noticed something unusual at this point: only one HORCM instance has been used, and only one is needed! This is only supported if just one DKC is involved and correct device and copy groups have been configured. This means ShadowImage and COW only as there is no way to know the information about the remote DKC.

Problem: In my testing I have been able to get “one line HORCM file” to work only with ShadowImage and COW. Not only does it not work with TrueCopy or HUR, **if you have any TrueCopy or HUR copy groups defined in the DKC it stops this from working. No groups are seen. You must use HORCM\_LDEVG**.

## A scheme for Device and Copy Group naming

The naming I have used is quite carefully chosen. It is not the only scheme and I don’t put it forward as a best practice but might be a starting point for your own scheme. Here is the breakdown:

* Application: SWB\_BC-014\_A
  + SWB my initials (in our lab it is invaluable to know who did what, in a production system it might be: owner, LOB, business group, team, project or business application)
  + BC014 the server name, alternatively application name or cluster name[[2]](#footnote-2)
  + A the application name (this is a single unit of data we may wish to manage so LOGS and DATA are often separated).
  + Other examples: FINANCE\_SAP\_LOGS, WEB\_SERV1\_DATA
* Copy Group: SWB\_BC-014\_A\_U1
  + Application
  + U U for UR, S for ShadowImage, T for TrueCopy and C for COW. It is important to distinguish products if you use more than one of them as MU numbers are ambiguous.
  + 1 MU number
* Device Group name: P\_SWB\_BC-014\_A\_U1
  + P\_ and S\_ P is for primary S for secondary. I have used P and S. Alternatives: Prim and DR (for TC/UR), Prim and Backup (For SI/COW) or London and Frankfurt.
  + Copy Group
* Note that device group name (LDEV\_GROUP) is unique on each DKC. For multiple DKC operations you can re-use names but at the moment I think making it unique will be best.
* Device Name: SWB\_BC-014\_A\_U1\_1
  + Copy Group
  + \_1 unique number for the device within the device group (you could use LDEV number but that works best for TC and where P-VOL and S-VOL numbers are always the same at both sites).
* Note that device names (LDEV\_NAME) are unique in the group but must match to the group you will be pairing with (just like a HORCM\_LDEV section of a HORCM file must)

# Command examples

## Basic management

### raidcom access and login

Once you have a command device with raidcom (authentication) enabled you will be required to log in to use Raid manager, this is required even for replication. All Raid Manager commands require login except: -x commands and inqraid. For storage provisioning you will need a Storage Manager 2 login with storage modify rights.

Login requested by first Raid Manager command:

**pairdisplay –g grp**

User for Serial#[53004] : sburr

Password:

To prevent scripts failing you should login before running commands:

**raidcom -login sburr myob**

**...**

**raidcom –logout**

Login rights are stored for user and serial in files in: /HORCM/usr/bin. One login works for all RM instances and all serial numbers if the login is the same. It uses the SVP Storage Navigator authentication mechanism. If this is configured to use external authentication can use group authentication through: LDAP, RADIUS, Active Directory.

Note: some manuals refer to individual access rights “External path setting authority”: that is not implemented. Group rights are fixed.

### What about VSS and HRpM agents?

Do they break if the command device has authentication enabled and is not logged in? I do not yet know but it would be worth being cautious.

## Resource[[3]](#footnote-3) locking

Resource locking is optional but recommended.

**raidcom get resource**

**RS\_GROUP RGID stat Lock\_owner Lock\_host Serial#**

**meta\_resource 0 Unlocked - - 53086**

Resource is free, lock it. **-time** is the time to wait for resource to become free:

**raidcom lock resource -resource\_name meta\_resource -time 60**

Check who has resource locked:

**raidcom get resource -s 53004**

**RS\_GROUP RGID stat Lock\_owner Lock\_host Serial#**

**meta\_resource 0 Locked sburr BC-014 53086**

Unlock resource:

**raidcom unlock resource -resource\_name meta\_resource**

Lock is for the specific user and server, not user alone. Do not leave it locked! You can see which server, but does that always help? Correct coding will use exception handling carefully.

## Simple queries

**set HORCMINST=99**

#one of the small annoyances of raidcom; HORCMINST must be set and does not support –I.

**raidcom -login USER01 PASS01**

##### Reporting ports ordered by port type (USP V[[4]](#footnote-4))

**raidcom get port | sort -k3,3 -k1**

PORT TYPE ATTR SPD LPID FAB CONN SSW SL Serial# WWN

CL1-B FIBRE ELUN AUT E1 Y PtoP N 0 53086 50060e8006cf5e01

CL2-B FIBRE ELUN AUT D3 Y PtoP N 0 53086 50060e8006cf5e11

CL3-B FIBRE ELUN AUT E0 Y PtoP N 0 53086 50060e8006cf5e21

CL4-B FIBRE ELUN AUT D2 Y PtoP N 0 53086 50060e8006cf5e31

CL5-B FIBRE MCU AUT DC Y PtoP N 0 53086 50060e8006cf5e41

CL6-B FIBRE MCU AUT D1 Y PtoP N 0 53086 50060e8006cf5e51

CL7-B FIBRE RCU AUT DA Y PtoP N 0 53086 50060e8006cf5e61

CL8-B FIBRE RCU AUT CE Y PtoP N 0 53086 50060e8006cf5e71

CL1-A FIBRE TAR AUT EF Y PtoP Y 0 53086 50060e8006cf5e00

CL2-A FIBRE TAR AUT D9 Y PtoP Y 0 53086 50060e8006cf5e10

CL3-A FIBRE TAR AUT E8 Y PtoP Y 0 53086 50060e8006cf5e20

CL4-A FIBRE TAR AUT D6 Y PtoP Y 0 53086 50060e8006cf5e30

CL5-A FIBRE TAR AUT E4 Y PtoP Y 0 53086 50060e8006cf5e40

CL6-A FIBRE TAR AUT D5 Y PtoP Y 0 53086 50060e8006cf5e50

CL7-A FIBRE TAR AUT E2 Y PtoP Y 0 53086 50060e8006cf5e60

CL8-A FIBRE TAR AUT D4 Y PtoP Y 0 53086 50060e8006cf5e70

### Nearly useful

The following additional format gives information on which WWN (HBA) are actually logged in to a port :

raidcom get port -port CL1-A

PORT LOGIN\_WWN Serial# -

CL1-A 10000000c9407116 53086 -

CL1-A 210000e08b1c3c50 53086 -

CL1-A 10000000c943463f 53086 -

CL1-A 10000000c997b13d 53086 -

This can tell you whether an HBA is active or not but unlike the equivalent Storage Navigator 2 function does not list WWN that are logged in but have no matching WWN entry in a Host Group. [Customers often ask for this.]

### Reporting parity groups

**raidcom get parity\_grp**

T GROUP Num\_LDEV U(%) AV\_CAP(GB) R\_LVL R\_TYPE SL CL DRIVE\_TYPE

R 1-1 1 99 0 RAID5 7D+1P 0 0 DKS5B-J300SS

R 1-2 1 99 0 RAID5 7D+1P 0 0 DKS5B-J300SS

R 1-5 1 99 0 RAID6 6D+2P 0 0 DKS5B-J300SS

R 1-6 1 99 0 RAID6 6D+2P 0 0 DKS5B-J300SS

R 5-8 1 100 0 RAID5 3D+1P 0 0 DKS5B-K146SS

R 5-9 1 99 0 RAID5 3D+1P 0 0 SDT5A-S200SS

R 6-1 0 0 536 RAID1 2D+2D 0 0 DKS5B-J300SS

...

### Reporting parity group detail

This format lists all LDEVs in the parity group, its location in the parity group and size (in hex).

**raidcom get parity\_grp -parity\_grp\_id[[5]](#footnote-5) 5-9**

T GROUP P\_NO LDEV# STS LOC\_LBA SIZE\_LBA Serial#

R 5-9 0 128 NML 0x000000000000 0x000044c64000 53086

R 5-9 1 - NML 0x000044c64000 0x000000000600 53086

### Reporting ldev information

These are probably the most valuable reports. As well as the basic reports this is the command most amenable to scripting using pipes. There are three simplified one per line formats and one full format. The one-line formats are: front-end information (mapped LDEVs), back-end information (internal LDEVS) and external device information[[6]](#footnote-6).

**raidcom get ldev -ldev\_id 0xbd00 -cnt 4 -key frnt**

Serial# LDEV# SL CL VOL\_TYPE VOL\_Cap(BLK) PID ATTRIBUTE Ports PORT\_No:LU#:GRPNAME ...

53086 48384 0 0 OPEN-V-CVS 1920000 0 CVS|HORC|AOU 1 CL3-A-3:1:SWB\_BC-008

53086 48385 0 0 OPEN-V-CVS 1920000 0 CVS|AOU 1 CL3-A-3:2:SWB\_BC-008

53086 48386 0 0 OPEN-V-CVS 1920000 0 CVS|AOU 1 CL3-A-3:3:SWB\_BC-008

Note: V-VOL size, pool number and used size, replication, and server mappings.

**raidcom get ldev -ldev\_id 0 -key rgrp**

Serial# LDEV# SL CL PID ATTRIBUTE R\_LVL RAID\_TYPE DRV\_TYPE DRV\_Cap GRPs RAID\_GRP ...

53086 0 0 0 4 POOL RAID5 7D+1P DKS5B-J300SS 571679368 1 01-01

Now we have LDEV role (POOL-vol), device type, raid-level, concatenation (none).

**raidcom get ldev -ldev\_id 1280 -key elun**

Serial# LDEV# SL CL PID ATTRIBUTE E\_VendorID E\_ProductID E\_VOLID

"E\_VOLID\_C" E\_PORTs PORT\_NO:LU#:WWN ...

53086 1280 0 0 - ELUN HITACHI OPEN-V 484954414348492052353030323733313031

363400000000000000000000000000000000 "HITACHI R50027310164................" 2 CL1-B-0:0:50060e8005273120 CL2-B-0:0:50060

e8005273130

Serial# LDEV# SL CL PID ATTRIBUTE E\_VendorID E\_ProductID

53086 1280 0 0 - ELUN HITACHI OPEN-V

E\_VOLID "E\_VOLID\_C" E\_PORTs PORT\_NO:LU#:WWN ...

0000000000000000000000000000000 "HITACHI R50027310164................" 2 CL1-B-0:0:50060e8005273120 CL2-B-0:0:50060

e8005273130

There are similar reports for External ldevs.

A report giving all information in one report is shown below but is much harder to parse. Only this full format reports all devices:

**raidcom get ldev -ldev\_id 0xbd06**

Serial# : 53086

LDEV : 48390

SL : 0

CL : 0

VOL\_TYPE : OPEN-V-CVS

VOL\_Capacity(BLK) : 192000000

NUM\_PORT : 1

PORTs : CL3-A-3 7 SWB\_BC-008

F\_POOLID : NONE

VOL\_ATTR : CVS : HDP

B\_POOLID : 0

LDEV\_NAMING : SWB\_H0A

STS : NML

OPE\_TYPE : NONE

OPE\_RATE : 100

MP# : 0

Used\_Block(BLK) : 190611456

TIER\_Relocation : Enable

TIER#1(MB) : 10878

TIER#2(MB) : 50610

TIER#3(MB) : 31584

Note: in addition to both the above sets of information we also have: size of data in each tier (for HDT devices), MP assignment, LDEV name, and if undergoing ZPR or Quick format, progress in **OPE\_\***.

## Using raidcom get ldev for useful stuff:

Tip: when searching for devices you will often search all devices. Here is the magic number for the highest LDEV 65279. So reports on all LDEVs is:

**raidcom get ldev -ldev\_id 0-65279 [-key frnt|rgrp|elun]**

We could pipe this data each time but each full scan takes 30-60 seconds so we’ll save the output and reuse the data:

**raidcom get ldev -ldev\_id 0-65279 -key frnt | grep -v "NOT DEF" > frnt.txt**

**raidcom get ldev -ldev\_id 0-65279 -key rgrp > rgrp.txt**

### Some I find useful:

**All Pool-vols in pool 0:**

**gawk "** **/Serial/ || $6~/POOL/ && $5==0" < rgrp.txt**

Serial# LDEV# SL CL PID ATTRIBUTE R\_LVL RAID\_TYPE DRV\_TYPE DRV\_Cap GRPs RAID\_GRP ...

53086 8 0 0 0 POOL RAID5 3D+1P DKS5B-J300SS 571679368 1 01-09

53086 9 0 0 0 POOL RAID5 3D+1P DKS5B-J300SS 571679368 1 01-10

53086 10 0 0 0 POOL RAID5 3D+1P DKS5B-J300SS 571679368 1 01-11

53086 11 0 0 0 POOL RAID5 3D+1P DKS5B-J300SS 571679368 1 01-12

53086 26 0 0 0 POOL RAID5 3D+1P DKS5B-K146SS 285177528 1 05-07

53086 27 0 0 0 POOL RAID5 3D+1P DKS5B-K146SS 285177528 1 05-08

53086 128 0 0 0 POOL RAID5 3D+1P SDT5A-S200SS 390625002 1 05-09

**pool-vols for all pools (in order):**

**gawk "/Serial/ || $6~/POOL/" < rgrp.txt | sort –k1,1r -k5**

Serial# LDEV# SL CL PID ATTRIBUTE R\_LVL RAID\_TYPE DRV\_TYPE DRV\_Cap GRPs RAID\_GRP ...

53086 8 0 0 0 POOL RAID5 3D+1P DKS5B-J300SS 571679368 1 01-09

...

**All V-VOLs in Pool 0 with paths (are used) and are not command devices:**

**gawk "/Serial/ || $8~/AOU/ && $8!~/CMD/ && $9>0 && $7==0" < frnt.txt**

**Serial# LDEV# SL CL VOL\_TYPE VOL\_Cap(BLK) PID ATTRIBUTE Ports PORT\_No:LU#:GRPNAME ...**

53086 1538 0 0 OPEN-V-CVS 209715840 0 CVS|AOU 1 CL1-A-1:1:JoeCarlisle\_bc00

53086 1539 0 0 OPEN-V-CVS 209715840 0 CVS|AOU 1 CL1-A-1:2:JoeCarlisle\_bc00

53086 1540 0 0 OPEN-V-CVS 209715840 0 CVS|AOU 1 CL1-A-1:3:JoeCarlisle\_bc00

53086 1541 0 0 OPEN-V-CVS 209715840 0 CVS|AOU 1 CL1-A-1:4:JoeCarlisle\_bc00

53086 8228 0 0 OPEN-V-CVS 10487040 0 CVS|AOU 2 CL3-A-3:19:SWB\_BC-008 CL4-A-3:1:SWB\_BC-008

53086 8229 0 0 OPEN-V-CVS 10487040 0 CVS|AOU 2 CL3-A-3:20:SWB\_BC-008 CL4-A-3:2:SWB\_BC-008

53086 48384 0 0 OPEN-V-CVS 1920000 0 CVS|HORC|AOU 1 CL3-A-3:1:SWB\_BC-008

53086 48385 0 0 OPEN-V-CVS 192000000 0 CVS|AOU 1 CL3-A-3:2:SWB\_BC-008

53086 48386 0 0 OPEN-V-CVS 192000000 0 CVS|AOU 1 CL3-A-3:3:SWB\_BC-008

53086 48387 0 0 OPEN-V-CVS 192000000 0 CVS|AOU 1 CL3-A-3:4:SWB\_BC-008

53086 48388 0 0 OPEN-V-CVS 192000000 0 CVS|AOU 1 CL3-A-3:5:SWB\_BC-008

53086 48389 0 0 OPEN-V-CVS 192000000 0 CVS|AOU 1 CL3-A-3:6:SWB\_BC-008

**Free LUNs in Pool 0, not command devices and an exact size:**

**gawk "/Serial/ || $6==1920000 && $8~/AOU/ && $8!~/CMD/ && $9==0 && $7==0" < frnt.txt**

Serial# LDEV# SL CL VOL\_TYPE VOL\_Cap(BLK) PID ATTRIBUTE Ports PORT\_No:LU#:GRPNAME ...

53086 1099 0 0 OPEN-V-CVS 1920000 0 CVS|AOU 0

53086 48427 0 0 OPEN-V-CVS 1920000 0 CVS|AOU 0

53086 48428 0 0 OPEN-V-CVS 1920000 0 CVS|AOU 0

## Advanced:

I found it hard to work out exactly which report(s) were needed to include all devices. The manual has a table which tells you but its use is not obvious. This may be a little better:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Get ldev | Frnt | Rgrp (internal) | Elun (external) |
| “NOT DEFINED” | Y | Y | N | N |
| INT POOL | Y | N | Y | N |
| INT JNL | Y | N | Y | N |
| INT OTHERS | Y | Y | Y | N |
| EXT POOL | Y | N | N | Y |
| EXT JNL | Y | N | N | Y |
| EXT OTHERS | Y | Y | N | Y |

So, to get all devices you must either use full format or frnt, rgrp **and** elun. Hint: frnt does not mean “mapped” as you might think; “not system” would be more accurate.

## Basic LUN management

### Some commands support multiple objects.

These syntaxes can be used to specify more than one LDEV number:

* -ldev\_id 300 –cnt 6
* -ldev\_id 300 301 302 303 304 305
* -ldev\_id 300-305
* -ldev\_id 0x12C-0x131
* -ldev\_id 01:2C-01:31

They are useful for commands like: raidcom get ldev, raidcom add journal, raidcom add snap\_pool, raidcom add dp\_pool, raidcom add device\_grp, and raidcom add or delete device\_grp.

### but do not always do what you expect or would hope for

**raidcom get ldev –ldev\_id 0 1**

does not work but the “-“ and –cnt formats do.

**raidcom add lun –port CL3-A-4 –ldev\_id 42 43**

**raidcom add lun –port CL3-A-4 –ldev\_id 42 –cnt 2**

Neither of these map two LUNs – instead they create a LUSE. Nobody uses LUSE any more.

This is OK for some pool and journal commands. Device groups seem to work and are probably the best way to manage multiple objects together.

## Bugs and warnings

Be careful when virtualizing UVM external devices that have data on them. The wrong process (especially a size error) will treat the External LUN as “space” and subdivide it into CVS volumes. This will corrupt the existing data.

Raidcom is not actually a command but a command wrapper. In early versions it meant not all syntax errors were handled correctly. This is largely corrected but sometimes the error messages are obscure.

The lengths of names differ between Storage navigator 2 and raidcom. The character set might be different too. This may bite you. Be especially aware of truncated names on USP V and USP. After truncation sometimes you end with two HSD with apparently the same name. Poor scripting will crash and/or correct.

### Raidcom illustration: create a new internal LUN, format it and add two paths

We start by clearing any command errors:

**raidcom reset command\_status**

Creating the LDEV is straightforward:

**raidcom add ldev -parity\_grp\_id 2-10 -capacity 960000K -ldev\_id 0xBD01**

Always name it!

**raidcom modify ldev -ldev\_id 0xBD01 -device\_naming SWB\_JNL1\_2**

Use raidcom get ldev to check the command is finished

**raidcom get ldev -ldev\_id 0xBC01 | grep STS**

STS : BLK

Also check the status.

**raidcom get command\_status**

We can format the device:

**raidcom initialize ldev -ldev\_id 0xBD01 -operation qfmt**

**raidcom get command\_status**

We can use the device but if we need to check if format is complete:

**raidcom get ldev -ldev\_id 0xBD01 | grep OPE\_**

**OPE\_TYPE : QFMT**

**OPE\_RATE : 100**

Then when finished:

**OPE\_TYPE : NONE**

**OPE\_RATE : 100**

Sometimes we do not need to format and can just clear the BLK status (but when?):

**raidcom modify ldev -ldev\_id 0xBD01 –status nml**

Now map it to two ports:

**raidcom add lun -port CL3-A-5 –ldev\_id 0xBD01**

**raidcom add lun -port CL4-A-5 –ldev\_id 0xBD01**

### Raidcom illustration: attachment of a new server

The steps for a new server are commonly repeated and cry out for a script. Here is the core:

**raidcom add host\_grp -port CL3-A-1 -host\_grp\_name SWB\_D\_2850\_4 -s 53004**

**raidcom add host\_grp -port CL4-A-1 -host\_grp\_name SWB\_D\_2850\_4 -s 53004**

**raidcom modify host\_grp -port CL3-A-1 -s 53004 -host\_mode 0 -host\_mode\_opt 40**

**raidcom modify host\_grp -port CL4-A-1 -s 53004 -host\_mode 0 -host\_mode\_opt 40**

**raidcom add hba\_wwn -port CL3-A-1 -s 53004 -hba\_wwn 10000000c9537bb2**

**raidcom add hba\_wwn -port CL4-A-1 -s 53004 -hba\_wwn 10000000c9537d06**

### Raidcom illustration: creation of command devices

I do this in a pool so no space is used. Note: at least one command device per OS type. Labelled.

**raidcom add ldev -s 53004 -pool\_id 0 -ldev\_id 1024 -capacity 96000**

**raidcom modify ldev -s 53004 -ldev\_id 1024 -device\_name SWB\_WIN\_CMD**

**raidcom add ldev -s 53004 -pool\_id 0 -ldev\_id 1025 -capacity 96000**

**raidcom modify ldev -s 53004 -ldev\_id 1025 -device\_name SWB\_LINUX\_CMD**

**raidcom add lun -port CL3-A-1 -s 53086 -ldev\_id 1025 -lun\_id 0 -command\_device**

**raidcom add lun -port CL4-A-1 -s 53086 -ldev\_id 1025 -lun\_id 0 -command\_device**

However, only a simple command device may be created, not one with security or provisioning.

## HDP and HDT

### Create V-VOL basic version:

**raidcom add ldev -pool\_id 0 -ldev\_id 0xBD02 -capacity 960000K**

### Create V-VOL production version:

I add here some examples of error checking. **raidcom** **add ldev** is a background task so we need to use **get** **command\_status**.

**raidcom reset command\_status**

**raidcom add ldev -pool\_id 0 -ldev\_id 0xBD02 -capacity 960000K**

**if errorlevel 1 goto FAILED**

**raidcom get command\_status | gawk "$4~/[1-9]/" | grep "[1-9]"**

**if errorlevel 1 goto FAILED**

**raidcom modify ldev -ldev\_id 0xBD02 -device\_naming SWB\_H0A**

**if errorlevel 1 goto FAILED**

This example is for Windows but with use of public domain: gawk, grep, sort etc. It will also work on unix/linux with appropriate changes of the **if errorlevel** lines to **$?**. Any unix scripter should be able to do this. (If you can’t close this manual now). A die hard BAT scripter could do this with **for /f**. Good luck to him.

The **add ldev** command could fail immediately (the first if) or fail later (the second if) we test for both.

What you do on **FAILED** is up to you.

The gawk here looks for non zero in field 4 and outputs the error information. See sidebar below.

Gawk as scripted returns no error code. The grep part turns it into an error return. There is probably an awk way of doing this, I know very little awk. In fact this example is 99% of what I know but is really useful for field extraction and test.

I include device naming. If in charge of storage management on a VSP I would design a naming scheme and enforce naming of **every** object.

Obviously these lines are screaming out to be a single short wrapper script.

## raidcom command\_status

As mentioned before, some commands will use **command\_status** to indicate errors. With long running commands you may even need to wait for completion before checking final status.

##### Good status

**raidcom get command\_status**

HANDLE SSB1 SSB2 ERR\_CNT Serial#

00c4 - - 0 53086

##### Bad status

**raidcom get command\_status**

HANDLE SSB1 SSB2 ERR\_CNT Serial#

00c4 2e10 6014 1 53086

The SSB1 and SSB2 information can be used to diagnose issues further. This may be in several manuals including: Program Product manuals, Raid Manager manuals (look in all) and the provisioning manual. You may have to ask GSC or SANTeam. Sometimes nobody knows.

### Create HDP Pool

An extended example; I start by looking for a free pool number:

**raidcom get pool -s 53004**

PID POLS U(%) SSCNT Available(MB) Capacity(MB) Seq# Num LDEV# H(%)

000 POLN 5 77 3112662 3293808 53004 4 3 80

001 POLN 1 14 1640436 1641024 53004 4 64 80

002 POLN 0 8 1644804 1644804 53004 2 7 80

003 POLN 0 0 520044 520044 53004 2 768 80

010 POLN 0 0 102400 102400 53004 1 13 80

We have a free pool slot 005.

I find suitable LDEVS, this will be HDP so Raid1 is OK:

**raidcom get ldev -ldev\_id 0 -cnt 20000 -key frnt -s 10044 | awk '$9=="0" && $5!="NOT" && $8!~/AOU/ && $8!~/ELUN/ && $5!~/33/'**

10044 19 0 0 OPEN-V-CVS 20971520 - CVS 0

10044 20 0 0 OPEN-V-CVS 20971520 - CVS 0

10044 21 0 0 OPEN-V-CVS 20971520 - CVS 0

10044 22 0 0 OPEN-V-CVS 20971520 - CVS 0

**raidcom get ldev -ldev\_id 19 -cnt 8 -key rgrp -s 10044**

Serial# LDEV# SL CL PID ATTRIBUTE R\_LVL RAID\_TYPE DRV\_TYPE DRV\_Cap GRPs RAID\_GRP ...

10044 19 0 0 - CVS RAID1 2D+2D DKR2F-J300FC 571679368 1 01-14

10044 20 0 0 - CVS RAID1 2D+2D DKR2F-J300FC 571679368 1 01-15

10044 21 0 0 - CVS RAID1 2D+2D DKR2F-J300FC 571679368 1 01-16

10044 22 0 0 - CVS RAID1 2D+2D DKR2F-J300FC 571679368 1 01-17

We have a free LDEVs 19-22.

Add pool is asynchronous, so we add the four pool vols:

**raidcom reset command\_status -s 53004**

**raidcom add dp\_pool[[7]](#footnote-7) -pool\_id 5 -ldev\_id 19 –cnt 4 -s 53004**

**raidcom get command\_status -s 53004**

HANDLE SSB1 SSB2 ERR\_CNT Serial#

00c6 - - 0 53004

### Grow pool

Note this is the same as create pool:

**raidcom add dp\_pool -pool\_id 5 -ldev\_id 23 –s 53004**

### Shrink pool

I have not found a way of doing this; one of the few GUI Only exceptions. “The deletion of pool VOL is not supported on Version01”

### Create HDP Pool

As before the pool is created by adding pool vols:

**raidcom add dp\_pool -pool\_id 5 -ldev\_id 40**

This has created an HDP pool. If we wanted HDT above we would continue.

**raidcom modify pool -pool 5 -pool\_attribute dt\_manual[[8]](#footnote-8)**

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

000 POLN AUT RLM VAL 1 00000bf6 000002ae 134946 563388 0 58 0

000 POLN AUT RLM VAL 2 00000021 0000000c 384258 822612 8 60 0

000 POLN AUT RLM VAL 3 00000000 00000000 1224174 3293808 8 8 0

005 POLN DEF STP INV 1 - - 0 8064 8 0 100

Note that this is an HDT with manual monitor and relocation. If you want automatic relocation and a schedule then you have to use GUI. For details on manual control see the next section:

### How do I do manually control?

You must first disable automatic monitoring and relocation, then you may use the manual control commands. The following are available:

Sets HDT manual mode (converting HDP to HDT if necessary):

**raidcom modify pool -pool 6 -pool\_attribute dt\_manual**

Convert pool to HDP:

**raidcom modify pool -pool 6 -pool\_attribute dp[[9]](#footnote-9)**

The commands for controlling monitoring are:

**raidcom monitor pool -pool {pool# | pool name} –operation start**

**raidcom monitor pool -pool {pool# | pool name} –operation stop**

The commands for controlling relocation are:

**raidcom reallocate pool -pool {pool# | pool name} –operation start**

**raidcom reallocate pool -pool {pool# | pool name} –operation stop**

Usage: You could use the above if you wanted monitoring from 9-5 but only Monday to Friday.

### Manual HDT monitoring and relocation example

I will concentrate on one pool in the below pool 11. It is a manual pool **DEF** and has had no data collected, the monitor is not running **STP** and there is no valid data yet **INV**, the tier properties graph will show an error:

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

011 POLN **DEF STP INV** 1 - - 19681326 25006422 8 0 100

I start monitoring and **MON** is indicated:

**raidcom monitor pool -pool 11 -operation start**

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

011 POLN DEF MON INV 1 - - 19681326 25006422 8 0 100

I stop it; initially it goes to **PND** and then **VAL, note tier range has been calculated**:

**raidcom monitor pool -pool 11 -operation stop**

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

011 POLN DEF STP PND 1 - - 19681326 25006422 8 0 100

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

011 POLN DEF STP VAL 1 00000000 00000000 19681326 25006422 8 0 100

We can now restart the monitor and start relocation if we want (this pool has one tier so relocation is not appropriate and will fail):

**raidcom monitor pool -pool 11 -operation start**

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

011 POLN DEF MON VAL 1 00000000 00000000 19681326 25006422 8 0 100

### zero page reclaim

Zero Page Reclaim can be initiated from the CLI:

**raidcom modify ldev –ldev\_id 42 -status discard\_zero\_page**

### V-VOL size extension

You are now able to extend V-VOL size online through Storage Navigator 2 or raidcom without creating a device group or even having the device mapped to a port (as required by the old tool **raidvchkset**):

**raidcom extend ldev –ldev\_id 42 –capacity 100G**

NOTE: the **raidcom** **extend ldev** command above, like the earlier **raidvchkset** is “extend **by”** 100G not “extend **to”** 100G. Storage Navigator 2 works the opposite. Beware, extension is irreversible: there is no LDEV shrink function (yet). This command will not function on internal or external LDEVs, only HDP or HDT V-VOLs. You still may not extend a V-VOL that is engaged in any replication (SI, TC, UR or COW). It will not work on COW V-VOLs.

### Delete pool

Very straightforward but you need to delete all V-VOLs first so a full sequence would be:

**raidcom delete lun -ldev\_id 0x11 -port CL3-A-6**

**raidcom get ldev -ldev\_id 0x11**

**raidcom delete pool -pool\_id 5**

See above for ways of getting all the LUN paths and V-VOLs for a pool.

### Can I control where HDT data is placed?

This question most often comes when you are in a “feature race” with a competitor. The feature we support is “lock the specified LDEV in its current tier allocation”. Here is a script example:

**raidcom modify ldev –ldev\_id 42 -status disable\_reallocation**

This command (and the opposite **enable\_reallocation**) is valuable if you convert pool types (you often end with a pool where V-VOLs are the wrong state). Note: it is not: **disable\_relocation** even though the get ldev report is **TIER\_Relocation**.

### Where does new data go?

Each tier has a configurable amount of space reserved for new pages. This configuration may be viewed with **raidcom get dp\_pool –key opt** (T%) and tuned with **raidcom modify pool** (this should be considered advanced operation and is best left alone). By default the freespace allocation is: SSD 0% of capacity, other types 8% of capacity.

**raidcom get dp\_pool -key opt**

PID POLS MODE STS DAT TNO TL\_RANGE TD\_RANGE TU\_CAP(MB) TT\_CAP(MB) T(%) P(%) R(%)

000 POLN AUT RLM VAL 1 00000018 00000007 552132 563388 0 48 0

000 POLN AUT RLM VAL 2 0000000b 00000006 522060 822612 8 2 0

000 POLN AUT RLM VAL 3 00000000 00000000 656166 3293808 8 1 0

## What is going on inside the Virtual Storage Platform?

One common complaint with USP V was that you could not see the progress of reclaim, rebalance and other operations. This is improved with the VSP. Monitoring of progress is available through raidcom and includes many of the functions you may wish to monitor. Here is an example of **raidcom get ldev** (I have removed some parts for clarity, full example is earlier):

**raidcom get ldev -ldev\_id 0xbd06**

Serial# : 53086

LDEV : 48390

VOL\_Capacity(BLK) : 192000000

VOL\_ATTR : CVS : HDP

B\_POOLID : 0

LDEV\_NAMING : SWB\_H0A

STS : NML

OPE\_TYPE : NONE

OPE\_RATE : 100

MP# : 0

Used\_Block(BLK) : 190611456

TIER\_Relocation : Enable

TIER#1(MB) : 10878

TIER#2(MB) : 50610

TIER#3(MB) : 31584

We can see:

* LDEV number, type, pool, name, attributes and allocated and used capacities
* VSP (MP) allocation
* For HDT: how much of this LDEV is on each tier.
* **STS, OPE\_TYPE** and **OPE\_RATE** give valuable insight into other things that may be going on in the background:
  + STS: **NML**: Normal **BLK**: Blocked **BSY**: Status is changing
  + OPE\_TYPE: **NONE**: Not in operation **FMT**: Formatting   
    **QFMT**: Quick Formatting **CCOPY**: correction copy   
    **CACCS**: Accessing correction data **SHRD**: Shredding  
    **ZPD**: Page Discarding
  + OPE\_RATE gives information on progress but only valid is some scenarios (I believe format and shred only)

Progress information for relocation in Hitachi Dynamic Tiering is available in the **raidcom get pool –key opt** command. This will be visited later as it contains a huge amount of valuable information in other areas.

**raidcom get ldev -ldev\_id 0xbd02**

...

VOL\_Capacity(BLK) : 192000000

Used\_Block(BLK) : 190267392

TIER\_Relocation : Enable

TIER#1(MB) : 12600

TIER#2(MB) : 42294

TIER#3(MB) : 38010

**raidcom get command\_status -s 53004**

HANDLE SSB1 SSB2 ERR\_CNT Serial#

00c6 - - 0 53004

**raidcom get pool -s 53004**

PID POLS U(%) SSCNT Available(MB) Capacity(MB) Seq# Num LDEV# H(%)

000 POLN 5 77 3112662 3293808 53004 4 3 80

001 POLN 1 14 1640436 1641024 53004 4 64 80

002 POLN 0 8 1644804 1644804 53004 2 7 80

003 POLN 0 0 520044 520044 53004 2 768 80

010 POLN 0 0 102400 102400 53004 1 13 80

Quick illustration: When I run the **raidcom get ldev –ldev\_id 0xBD01** command to get a report on an LDEV (Internal, External or V-VOL), one line shows the MPPK/VSD:

MP# : 2

LDEV Ownership can be changed, by Storage Navigator 2, CLI (Example: **raidcom modify ldev 0xbd02 -mp\_blade\_id 2**), or will change automatically on failure.

# Replication

## configuring MCU and RCU

This is quite straightforward provided you want CU FREE (most will). In the example I have two storage subsystems: 53086 and 53004 and are using port 5B and 6B for INITIATOR and 7B and 8B for RCU on both storage. Bi-directional connection is required (as it always should be):

**raidcom reset command\_status**

**raidcom add rcu -s 53086 -cu\_free 53004 R700 1 -mcu\_port CL5-B -rcu\_port CL7-B**

**raidcom get command\_status**

**raidcom add rcu -s 53004 -cu\_free 53086 R700 1 -mcu\_port CL5-B -rcu\_port CL7-B**

**raidcom get command\_status**

This configures one (odd) path but we need a second (even) path. This is a different command.

**raidcom add rcu\_path -s 53086 -cu\_free 53004 R700 1 -mcu\_port CL6-B -rcu\_port CL8-B**

**raidcom get command\_status**

**raidcom add rcu\_path -s 53004 -cu\_free 53086 R700 1 -mcu\_port CL6-B -rcu\_port CL8-B**

**raidcom get command\_status**

raidcom add rcu is not supported on USP V or earlier. You can however configure the VSP end with raidcom and use storage navigator on the USP V.

**raidcom add rcu -s 53086 -cu\_free 10033 R600 1 -mcu\_port CL5-B -rcu\_port CL3-F**

**raidcom get command\_status**

**raidcom add rcu\_path -s 53086 -cu\_free 10033 R600 1 -mcu\_port CL6-B -rcu\_port CL4-F**

**raidcom get command\_status**

You can report, but only on the VSP:

**raidcom get rcu**

Serial# ID PID MCU RCU M/R T STS MP NP IM FZ RTO(s) RTT(ms)

53004 R7 - 10 10 RCU F NML 1 2 MR D 15 1

10033 R6 - 10 20 RCU F NML 1 2 MR D 15 1

53004 R7 0 - - RCU F NML 1 2 - - 15 0

10033 R6 1 - - RCU F NML 1 2 - - 15 1

53004 R7 0 - - MCU F UNK - - - - - -

**raidcom get rcu -cu\_free 53004 R7 0**

Serial# ID PID MCU RCU M/R T PNO MPORT RPORT STS\_CD SSIDs ...

53004 R7 0 - - RCU F 0 CL5-B CL8-B NML\_01 -

53004 R7 0 - - RCU F 1 CL6-B CL7-B NML\_01 -

A “get everything” script”:

**raidcom get rcu -s 53086 | awk '$6=="RCU" {if($3=="-"){printf "raidcom get rcu -cu\_free %d %s 0\n",$1,$2}else{printf "raidcom get rcu -cu\_free %d %s %s\n",$1,$2,$3}}' |sort -u | sh**

Serial# ID PID MCU RCU M/R T PNO MPORT RPORT STS\_CD SSIDs ...

10033 R6 1 - - RCU F 0 CL5-B CL3-F NML\_01 -

10033 R6 1 - - RCU F 1 CL6-B CL4-F NML\_01 -

Serial# ID PID MCU RCU M/R T PNO MPORT RPORT STS\_CD SSIDs ...

53004 R7 0 - - RCU F 0 CL5-B CL8-B NML\_01 -

53004 R7 0 - - RCU F 1 CL6-B CL7-B NML\_01 -

Error reporting is asynchronous, here is the use of the wrong Raid Type (undocumented I think):

**raidcom get command\_status**

HANDLE SSB1 SSB2 ERR\_CNT Serial#

00c4 2ed6 ffff 1 53086

## Configuring Journals

**raidcom add journal -journal\_id 0 -ldev\_id 0xBC00 0xBC08**

**raidcom add journal -journal\_id 1 -ldev\_id 0xBC01 0xBC09**

**raidcom add journal -journal\_id 2 -ldev\_id 0xBC02 0xBC0A**

**raidcom add journal -journal\_id 3 -ldev\_id 0xBC03 0xBC0B**

**raidcom get journal**

JID MU CTG JNLS AP U(%) Q-Marker Q-CNT D-SZ(BLK) Seq# Num LDEV#

000 - - SMPL - - - - 3708415 53086 2 48128

001 - - SMPL - - - - 3708415 53086 2 48129

002 - - SMPL - - - - 3708415 53086 2 48130

003 - - SMPL - - - - 3708415 53086 2 48131

**raidcom get journal -key opt**

JID MU CTG JNLS TYPE TTYPE MODE IF DOW(S) PBW(M) CR CS(bps) DM MP# Seq#

000 0 0 SMPL OPEN - CACHE E 60 5 L 256 Y 0 53086

000 1 0 SMPL OPEN - CACHE E 60 5 L 256 Y 0 53086

000 2 0 SMPL OPEN - CACHE E 60 5 L 256 Y 0 53086

000 3 0 SMPL OPEN - CACHE E 60 5 L 256 Y 0 53086

Error reporting is asynchronous, I have left out **get command\_status** commands.

The above are default settings. Many of the settings can be changed:

**raidcom modify journal -journal\_id <journal ID#> [-data\_overflow\_watch <time>][- timer\_type {system|local}] [-mirror\_id <mun#>]**

**raidcom modify journal -journal\_id <journal ID#> [-path\_blocked\_watch <time>] [-timer\_type {system|local}] [-mirror\_id <mun#>]**

**modify journal -journal\_id <journal ID#> [-cache\_mode {y|n}]**

**raidcom modify journal -journal\_id <journal ID#> [-mp\_blade\_id <mp#>]**

## Best Practice Configuration of Raid Manager: HORCM FILE

### HORCM\_MON section

There is a new convention for HORCM file network port numbers:

31000 – horcm.conf

31001 – horcm0.conf

31001+N – horcmN.conf

Try to use these in all new deployments. If you do – for Provisioning, Shadowimage and COW you can avoid the need to have a HORCM\_MON section at all, it will decide port number from the file name and bind to localhost. A minimal HORCM file can start with HORCM\_CMD and in some cases (like use for provisioning only) this is the only section required. The original format is still supported.

Best Practice: where HORCM\_MON is used I recommend **NONE** instead ofserver name as this works better on clusters. Also, if you have more than one HORCM file you are recommended to specify poll and timeout in only one HORCM file.

## HORCM\_CMD section

For HORCM\_CMD I recommend you only use the **\\.\CMD-serial-ldev#-port** syntax and use the exact formats described below. All other formats can be broken by hardware failures, reboots or configuration changes; these are resilient:

**Windows:**  \\.\CMD-53086-1024-CL3 \\.\CMD-53086-1024-CL4

**Linux:** \\.\CMD-53086-1025-CL3:/dev/sd/\* \\.\CMD-53086-1024-CL4

**Solaris:** \\.\CMD-53086-1025-CL3:/dev/rdsk/\* \\.\CMD-53086-1024-CL4

Host storage domain is not needed (and leaving it out means you can use the same name on cluster nodes). For other operating systems substitute a valid search string for raw devices. This will also be needed where HDLM or Symantec DMP path management is used and the raw devices are named differently.

**Do not** be tempted to shorten these, yes it will work in testing. It will fail when you need it to be resilient. None of these are resilient:

\\.\CMD-53086-1024-CL3

\\.\CMD-53086-1024

\\.\CMD-53086

These are wrong; one line only if the serial numbers are the same (the manual is wrong/misleading):

HORCM\_CMD

\\.\CMD-53086-1024-CL3

\\.\CMD-53086-1028-CL4

The following is OK only because 1024!=1028:

\\.\CMD-53086-1024 \\.\CMD-53086-1028

It is recommended to only specifying **:/dev/rdsk** etc. once. Not:

\\.\CMD-53086-1025-CL3:/dev/sd \\.\CMD-53086-1025-CL4:/dev/sd

## Minimal files

Now we can miss out the HORCM\_MON section a two line file with just HORCM\_CMD can be used for initial management. With only the HORCM\_CMD section you can use raidscan (to provide information needed to write HORCM files) or raidcom (to provision storage and report on configuration).

## Specifying OUT-OF-BAND in HORCM files

The format is simple \\IPCMD\<SVP ip address>-31001. E.G:

\\.\IPCMD-172.17.45.35-31001

Note the use of standard port numbers for HORCM0. This format can be mixed with normal command devices in the same file but is probably not a good idea. Out-of-band devices are 5-10 times slower than normal. You **must** login is to use an out-of-band command device; it has all privileges.

## HORCM\_DEV, HORCM\_LDEV and HORCM\_LDEVG

HORCM\_DEV and HORCM\_LDEV sections remain as before. The new format supported is HORCM\_LDEVG. This format refers to a copy group and device groups stored in the DKC. One line can specify a whole group. The best way of understanding this is that raidcom copy\_groups and device\_groups together must have exactly the same information as the HORCM files would have in the HORCM\_LDEV and HORCM\_INST sections.

## HORCM\_INSTP

This parameter will be rarely used. It can be used as an alternate to HORCM\_INST to specify pathID for TrueCopy; pathID allows you to control which ports are used for replication:

HORCM\_INSTP

dev\_group ip\_address service pathID

VG01 HSTA 31001 1

VG02 HSTB 31001 2

PathID can only be specified for TrueCopy. It cannot be specified for UR. PathID is used for paircreate and pairresync -swapp[s] commands, so must be specified at both PVOL and SVOL sites. If PathID is not specified, CU free is used. There are no new command options just the HORCM file option. Warning: such a new feature may not be supported on all platforms or on all microcodes.

## Provisioning and replication use on one server

Customers with existing replication solutions will not be coded for raidcom login. Enabling authentication on the device will cause scripts to fail. My recommendation where replication and provisioning management is required on the same server is to have two command devices (each with two paths): one for replication has user authentication disabled and can be used without login, the second is for raidcom and has authentication enabled. To prevent confusion during build I recommend you always have the ldev number of the replication device lower than the raidcom device. This will make constructing the files easier. Additionally you can use ldev labeling with something like this:

**inqraid -fgn -CLI $Phys –sort | grep CMD**

DEVICE\_FILE PORT SERIAL LDEV CTG H/M/12 SSID R:Group LDEV\_NIC\_NAME

Harddisk21 CL3-A-3 53004 1024 - - 3504 A:00000 CMDWINREPLICATION

Harddisk38 CL4-A-3 53004 1024 - - 3504 A:00000 CMDWINREPLICATION

Harddisk22 CL3-A-3 53004 1025 - - 3504 A:00000 CMDWINPROVISION

Harddisk39 CL4-A-3 53004 1025 - - 3504 A:00000 CMDWINPROVISION

**Inqraid** can be run before any other commands or installation. Note: because different operating systems require their own command device, my name includes CMDWIN, to add as an additional check. Command devices may be used by multiple servers. I would probably not use more that 16 servers per device. Also do not share devices used for performance measurement with HtNM Agents, TMEA or similar.

# Advanced: raidcom batch files

Customers often ask for additional script batch capabilities. Specifically:

* Execute a command in check mode, do not action
* Execute a list of commands checking they are all consistent.

These capabilities have been partially implemented in raidcom. Let’s work an example, starting without batch checking and at least one error:

**raidcom add lun -port CL3-A SWB\_BC-008 -ldev\_id 9999 -lun\_id 2**

raidcom: [EX\_CMDRJE] An order to the control/command device was rejected

It was rejected due to SKEY=0x05, ASC=0x26, SSB=0xB958,0x095D on Serial#(53086)

This failed. The reason is clear if you investigate:

**raidcom get ldev -ldev\_id 9999 |grep VOL\_TYPE**

VOL\_TYPE : NOT DEFINED

The simple solution to this would be to check the error return with if errorlevel or $? in the usual way. But if this was a sequence and we wanted the first command to fail if any later were in error it could not be done.

## Raidcom batch facility

You can create a batch file which executes several raidcom commands, then execute (or check) them. A very common example, we map one LUN to both odd and even ports.

**echo raidcom add lun –port CL3-A SWB\_BC-008 –lun\_id 2 –ldev\_id 9999 > bad.bat**

**echo raidcom add lun –port CL4-A SWB\_BC-008 –lun\_id 2 –ldev\_id 9999 >> bad.bat**

**echo raidcom modify ldev -ldev\_id 0xbd00 -ldev\_name "SWB\_BC008“ >> bad.bat**

**raidcom -zt bad.bat**

raidcom: [EX\_CMDRJE] An order to the control/command device was rejected...

raidcom: [EX\_CMDRJE] An order to the control/command device was rejected...

A simple batch sequence works but is not as good as a batch script with error checking. As currently used the third line will succeed in renaming the LDEV despite the first two errors.

NOTE: the syntax is picky. On Windows the file name **must** end .bat.On Unix execute permission is required and the file path must be fully qualified, and PATH is ignored (I.E. use ./bad.bat not bad.bat).

## Raidcom precheck facility

The precheck facility uses a data file containing the “DKC configuration” that it can check the batch file against. The first step is to obtain the “configuration”. Here is an example of getting the LDEV configuration data:

**raidcom get ldev -ldev\_id 0 -cnt 65280 -store ldevconf > ldevconf.txt**

This gets all ldevs so takes a minute, quicker would be –ldev\_id 9999 but that requires more careful scripting. We can now run our batch file in precheck mode:

**raidcom -zt bad.bat -load ldevconf -checkmode precheck**

*raidcom add lun -port CL3-A SWB\_BC-008 -lun\_id 2 -ldev\_id 9999*

*raidcom: LDEV(9999) does not exist as status is [2] on UnitID# 0.*

*raidcom\_#6 : [EX\_CTXCHK] Context Check error*

*raidcom add lun -port CL4-A SWB\_BC-008 -lun\_id 2 -ldev\_id 9999*

*raidcom: LDEV(9999) does not exist as status is [2] on UnitID# 0.*

*raidcom\_#7 : [EX\_CTXCHK] Context Check error*

*raidcom modify ldev -ldev\_id 0xbd00 -ldev\_name "works“*

**echo %errorlevel%**

*199*

Note the whole command sequence will be rejected and returns a suitable errorlevel.

## Running a sequence with checking

The same sequence can be run with checks enabled:

**raidcom -zt bad.bat -load ldevconf**

*raidcom add lun -port CL3-A SWB\_BC-008 -lun\_id 2 -ldev\_id 9999*

*raidcom: LDEV(9999) does not exist as status is [2] on UnitID# 0.*

*raidcom\_#6 : [EX\_CTXCHK] Context Check error*

*raidcom add lun -port CL4-A SWB\_BC-008 -lun\_id 2 -ldev\_id 9999*

*raidcom: LDEV(9999) does not exist as status is [2] on UnitID# 0.*

*raidcom\_#7 : [EX\_CTXCHK] Context Check error*

*raidcom modify ldev -ldev\_id 0xbd00 -ldev\_name "works“*

This does what we want, the first two commands fail and the third is not executed.

**raidcom get ldev -ldev\_id 0xbd00 | grep NAM**

LDEV\_NAMING : previousname

## Why it’s a bit more complex than that!

You might note that I created an ldevconf file to check against. Raidcom supports checking against files with ldev, port **or** hostgroup information: but not all at the same time.

**raidcom get host\_grp -port CL3-A -store hsd3Aconf**

**raidcom -zt bad.bat -load hsd3Aconf -checkmode precheck**

This passes check but would fail for the above ldev error or if there was a problem with group 4A. Also note we need a file (and check) for each host group used.

## The full monty

Here is a script with every check available.

**echo raidcom add lun –port CL3-A SWB\_BC-008 –lun\_id 2 –ldev\_id 9999 > bad.bat**

**echo raidcom add lun –port CL4-A SWB\_BC-008 –lun\_id 2 –ldev\_id 9999 >> bad.bat**

**echo raidcom modify ldev -ldev\_id 0xbd00 -ldev\_name "works“ >> bad.bat**

**raidcom lock resource -resource\_name meta\_resource**

**raidcom get ldev -ldev\_id 9999 -store ldevconf9999 > ldevconf9999.txt**

**raidcom get host\_grp -port CL3-A -store hsd3Aconf**

**raidcom get host\_grp -port CL4-A -store hsd4Aconf**

**raidcom -zt bad.bat -load ldevconf9999 -checkmode precheck**

**if [$? < 0 ];exit 42**

**raidcom -zt bad.bat -load hsd3Aconf -checkmode precheck**

**if [$? < 0 ];exit 42**

**raidcom -zt bad.bat -load hsd4Aconf -checkmode precheck**

**if [$? < 0 ];exit 42**

**raidcom -zt bad.bat**

**raidcom unlock resource -resource\_name meta\_resource**

***So is this perfect? Sorry no.***

If we correct our LDEV error (9999 should have been 0xbd02) and run the full-monty sequence we can get the precheck and context checks all to pass. But the sequence still fails.

*raidcom: [EX\_CMDRJE] An order to the control/command device was rejected*

*It was rejected due to SKEY=0x05, ASC=0x26, SSB=0xB958,0x015A on Serial#(53086)*

And the last part of the sequence will be executed.

You may wish to know [***what check file for which commands?***](#_Hlk278199908)

# Other resources on raidcom

There is a lot more to be said on using **raidcom**; you might want to check out my presentation on Jive: [Scripted Storage Management with the Hitachi Virtual Storage Platform 1.0.pptx](http%3A%2F%2Floop.hds.com%2Fdocs%2FDOC-1330). This covers several aspects not covered here and in more detail:

* Batch operation
* Checked batch operation
* Consistency checks
* Copy Group details
* Resource locking
* Asynchronous action

# Appendices

### What commands are asynchronous?

See the reference manual. This list summarizes:

**blocking|add|delete|format|restore LDEV**

**add|delete DP\_VOL**

**add|delete|unblock POOL**

**add|delete RCU|RCU Path**

**create|delete Journal**

**restore|set|delete|disconnect Epath**

**map|unmap ELUN**

## What check file for which commands?

The reference manual has a table but I found it “back to front”. Here is the table in command order:

add lun ldev attribute

add lun hgrp attribute

delete lun ldev attribute

delete lun hgrp attribute

add device\_grp ldev attribute

add external\_grp port attribute

add hba\_wwn hgrp attribute

add journal ldev attribute

add ldev ldev add

add path port attribute

add rcu port attribute

add snap\_pool ldev attribute

add dp\_pool ldev attribute

check\_ext\_storage ldev attribute

check\_ext\_storage\_path port attribute

delete device\_grp ldev attribute

delete hba\_wwn hgrp attribute

delete host\_grp hgrp delete

delete journal ldev attribute

delete ldev ldev delete

delete path port attribute

disconnect path port attribute

extend ldev ldev attribute

initialize ldev ldev attribute

modify host\_grp hgrp attribute

modify ldev ldev attribute

modify port port attribute

reset hba\_wwn hgrp attribute

set hba\_wwn hgrp attribute

Contents

[A raidcom cookbook for GSS personnel 1](#_Toc279060758)

[Introduction 1](#_Toc279060759)

[Goal 1](#_Toc279060760)

[Scope 2](#_Toc279060761)

[Confidentiality 2](#_Toc279060762)

[Important background concepts 2](#_Toc279060763)

[Background execution of tasks 2](#_Toc279060764)

[Security 3](#_Toc279060765)

[Out-of-band operation 3](#_Toc279060766)

[Command device creation and security 4](#_Toc279060767)

[Resource naming 4](#_Toc279060768)

[Naming schemes 5](#_Toc279060769)

[LDEV and V-VOL naming 5](#_Toc279060770)

[Host group naming 6](#_Toc279060771)

[Worldwide names 6](#_Toc279060772)

[Pool naming 6](#_Toc279060773)

[Device Group naming 6](#_Toc279060774)

[Use of device groups: 6](#_Toc279060775)

[Device and Copy Group naming with HORCM: ShadowImage or COW 7](#_Toc279060776)

[Device and Copy Group naming with HORCM: TrueCopy or UR 8](#_Toc279060777)

[Copy Group naming: ShadowImage and COW 9](#_Toc279060778)

[A scheme for Device and Copy Group naming 10](#_Toc279060779)

[Command examples 10](#_Toc279060780)

[Basic management 10](#_Toc279060781)

[raidcom access and login 10](#_Toc279060782)

[What about VSS and HRpM agents? 11](#_Toc279060783)

[Resource locking 11](#_Toc279060784)

[Simple queries 11](#_Toc279060785)

[Reporting parity groups 12](#_Toc279060786)

[Reporting parity group detail 12](#_Toc279060787)

[Reporting ldev information 12](#_Toc279060788)

[Nearly useful 12](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060789)

[Using raidcom get ldev for useful stuff: 14](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060790)

[Some I find useful: 14](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060791)

[Basic LUN management 15](#_Toc279060792)

[Advanced: 15](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060793)

[Some commands support multiple objects. 15](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060794)

[but do not always do what you expect or would hope for 15](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060795)

[Raidcom illustration: create a new internal LUN, format it and add two paths 16](#_Toc279060796)

[Bugs and warnings 16](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060797)

[HDP and HDT 17](#_Toc279060798)

[Create V-VOL basic version: 17](#_Toc279060799)

[Create V-VOL production version: 17](#_Toc279060800)

[Create HDP Pool 17](#_Toc279060801)

[raidcom command\_status 17](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060802)

[Grow pool 18](#_Toc279060803)

[Shrink pool 18](#_Toc279060804)

[Create HDP Pool 18](#_Toc279060805)

[How do I do manually control? 18](#_Toc279060806)

[Manual HDT monitoring and relocation example 19](#_Toc279060807)

[zero page reclaim 19](#_Toc279060808)

[V-VOL size extension 19](#_Toc279060809)

[Delete pool 19](#_Toc279060810)

[Can I control where HDT data is placed? 20](#_Toc279060811)

[Where does new data go? 20](#_Toc279060812)

[What is going on inside the Virtual Storage Platform? 20](#_Toc279060813)

[Enhancements to HORCM files 21](#_Toc279060814)

[HORCM\_MON 21](#_Toc279060815)

[HORCM\_CMD 22](#_Toc279060816)

[Minimal files 22](#_Toc279060817)

[Specifying OUT-OF-BAND in HORCM files 22](#_Toc279060818)

[HORCM\_DEV, HORCM\_LDEV and HORCM\_LDEVG 22](#_Toc279060819)

[HORCM\_INSTP 22](#_Toc279060820)

[Advanced: raidcom batch files 24](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060821)

[Raidcom batch facility 24](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060822)

[Raidcom precheck facility 24](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060823)

[Running a sequence with checking 25](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060824)

[Why it’s a bit more complex than that! 25](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060825)

[The full monty 25](file%3A%2F%2F%2FC%3A%2FDocuments%2520and%2520Settings%2Fsburr%2FMy%2520Documents%2FProjects%2FVictoria%2FDocument%2520review%2FDelivery%2520Kit%2FA%2520raidcom%2520cookbook.docx#_Toc279060826)

[Other resources on raidcom 26](#_Toc279060827)

[Appendices 26](#_Toc279060828)

[What commands are asynchronous? 26](#_Toc279060829)

[What check file for which commands? 26](#_Toc279060830)

1. With the launch of High Availability Manager we actually lose replication symmetry. HAM is asymmetric; this makes the HAM solution simpler whilst retaining the highest integrity. [↑](#footnote-ref-1)
2. In this case the primary and secondary volumes are both on the same server. [↑](#footnote-ref-2)
3. Although this command works in V01 this will have more use when we have a future capability of the VSP. It is intended to include a complete hierarchical resource management capability in the VSP. This would ultimately replace the SLPR capability. In the V01 microcode there is only one resource **meta\_resource**, this corresponds to all resources in the VSP (SLPR 0). [↑](#footnote-ref-3)
4. Where I know a command also works on USP V I will note it like this. [↑](#footnote-ref-4)
5. You will find the command syntax a bit repetitive and unclear. One tip I have found to help remember the options: if a number is required use xxxx\_id, if a name is wanted use: xxxx\_name. [↑](#footnote-ref-5)
6. Unfortunately this misses one important class: V-VOL devices which are not yet mapped. Annoying. I must write an extension. [↑](#footnote-ref-6)
7. In older versions of Raid Manager thp\_pool was used for HDP this is now dp\_pool for all commands. [↑](#footnote-ref-7)
8. Was **smart\_manual.** [↑](#footnote-ref-8)
9. Was **thp.** [↑](#footnote-ref-9)