

---

# **HORCM utilities documentation**

## **Documentation**

*Release 0.1*

**Gratien D'haese**

**Apr 13, 2017**



---

# Contents

---

<b>1 License</b>	<b>3</b>
<b>2 Contributing</b>	<b>5</b>
<b>3 Contents:</b>	<b>7</b>
3.1 About HORCM utilities documentation . . . . .	7
3.2 User Documentation . . . . .	7
<b>4 Indices and tables</b>	<b>27</b>



The HORCM Utilities documentation contains comprehensive documentation on the HORCM Utilities. This page describes documentation's licensing, editions, and versions, and describes how to contribute to the project.



# CHAPTER 1

---

## License

---

This documentation is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (i.e. .CC-BY-NC-SA.) license.



## CHAPTER 2

---

### Contributing

---

Please, we encourage you to help us to improve this documentation.

To contribute to documentation the Github interface enables users to report errata or missing sections, discuss improvements and new sections through the issue-tracker at:

[HORCM Utilitis Docs GitHub Issue Tracker](#).

---

**Note:** This documentation is under constant development. Please be patient...

---



### About HORCM utilities documentation

The HORCM utilities documentation contains the guideline around BC-exec.sh script. However, some minor other scripts will be explained as well, such as CheckHorcmConsistency.sh.

### License

To contribute to documentation the Github interface enables users to report errata or missing sections, discuss improvements and new sections through the issue-tracker at: <https://github.com/gdha/horcm-utils-docs/issues>

All documentation is available under the terms of a [Creative Commons \(CC BY-NC-SA 4.0\) License](#).

### Contributing

Please, we encourage you to help us to improve this documentation.

– Gratien D’haese

### User Documentation

#### Objective

The purpose of this guideline is to describe the different script functionalities of HORCM utility BC-exec.sh for using Business Copy and mounting the Volume Group(s) on HP-UX or RedHat Linux. The principal usage of the BC-exec.sh script is to assist in making a backup of the Oracle databases on the Business Copy server (or backup server) side, while the Oracle databases can stay up and running on the production servers.

The other utilities will be discussed as well.

### Scope

The scope of this Guideline applies to Coordinating and Executing Engineers who will be installing and configuring HP XP Business Copy Software.

This Guideline is valid for HP-UX and Linux platforms.

### Out of Scope

The installation and configuration of the HP XP Raid Manager Software is beyond the scope of this Guideline.

### HP XP Business Copy

HP XP Business Copy Software is a local replication solution that provides data copies within a single HP XP Disk Array or in storage systems attached to an XP Disk Array as external storage. HP XP Business Copy Software uses array-based replication technologies that do not interfere with host-level operations, allowing customers to spend time conducting business rather than worrying about how backup and testing operations will impact your systems. Providing both snapshots and mirrors, HP XP Business Copy Software reduces the cost of backups, accelerates recovery from failures, provides deployment flexibility, and simplifies application testing. HP XP Business Copy Software can create full copies in physically separate locations within the array so that accesses to the copy will have no impact on the original production volume. In addition, you can use the space-efficient snapshot capability to create frequent point-in-time copies of important volumes so that you can quickly restore data to an earlier version if necessary.

### HP XP Business Copy Script (BC-exec.sh)

The HP XP Business Copy Software used is a combination of HP XP Raid Manager Software running on the host computer and a shell script (called BC-exec.sh) which assists the different phases foreseen within our organization to create off host backups.

### Pre-requisites for HP XP Business Copy

The HP XP Business Copy Software can only be used with the HP XP Storage Arrays such as the P9500. On the host computer we need also the HP XP Raid Manager Software fully installed and configured. On the host computer we also need a script, called BC-exec.sh, to assist in the creation of Business Copy snapshots or mirrors. The BC-exec.sh script is part of the HP-UX software depot BC\_UTILS.

### BC-exec.sh Script

The purpose of the script BC-exec.sh is to automate the Business Copy processes such as splitting fully mirrored disks, re-sync the disks, mount the volume groups of those disks and make the file systems available on the (backup) server for other purposes such as backup. The script should be present and able to run on all host systems involved in the Business Copy processes. This is on the P-VOL system itself, and on any system defined to use the S-VOL disks, be it MU#0 or MU#1. The script is useable on different host computer Operating Systems, such as HP-UX, Linux and Solaris. It is also backwards compatible within a certain Operating System type, e.g. HP-UX 11.11, HP-UX 11.23 and HP-UX 11.31. The script is able to handle LVM1 and LVM2 based Volume Groups. Raw disks (without a volume group) are out of scope within the BC-exec.sh script.

## BC-exec.sh Software Pre-requisites

The script is written in Korn Shell, therefore, it is important that the *ksh* program is available on the host computers. If that is not the case it must be installed before running the BC-exec.sh script.

The BC-exec.sh script also relies on the XPinfo program, therefore, it must be present on all host computers as well. The script is a kind of wrapper around the HP XP Raid Manager Software; therefore, Raid Manager should be installed and configured on all involved host computers.

## BC-exec.sh Command Arguments

The BC-exec.sh script has the following usage:

```
#-> /usr/local/sbin/BC-exec.sh -h
Usage: BC-exec.sh [-c /path/configurationfile] [-m mail_destination] [-D log_
↪directory] [-Fdvh] [Operation]

    -c /path/configurationfile

    -F : Force a path prefix for MU#0 BCV (MU#1 always uses a prefix)

    -m : mail destination (default: )

    -D /path_of_log_directory (default: /var/adm/log)

    -d : debug mode (default is OFF)

    -v : show version and exit

    -h : show help (usage) and exit

    Operation: supported operations are:
                validate (default)
                resync
                split
                extract
                mount
                umount
                purgelogs <number of days>

    Note that we need at minimum a "-c" option
    ----
2015-11-24 12:56:06 LOG: Exit code 1
```

## BC-exec.sh Software Requirements

The BC-exec.sh is designed to work in different workflows and the script relies on certain input or configuration files which are generated on the source system (where the P-VOLs are residing) and these input files should be made available via NFS export to the system where the S-VOLs are defined on. Therefore, the file system /opr\_<package-name> is NFS exported to all target host systems (S-VOLs). On environments where the /opr\_<package-name> is not available, it is also possible to create this directory on the S-VOL system (which is typically not clustered), nfs-exported to the P-VOL systems and then mounted on the P-VOL side. The configuration file location is provided as a parameter for the script: -c </path/configuration\_file>. Keep in mind, that the input files must be available on the /path directory, if not, the script will return an error.

The SAP teams required the possibility to exclude certain SAP mount points, such as `/oracle/<SID>`. This should be a variable in the configuration file for `BC-exec.sh` script.

Furthermore, the SAP teams like the S-VOL MU#0 disks be mounted on their original mount points (from the P-VOL) for backup reasons (with the original mount points). This makes the recovery on the source host (where P-VOL resides) much easier. However, `BC-exec.sh` is able to mount it on another path if required (by setting an argument option, such as `-F`). The S-VOL MU#1 (second set of Business Copy disks) will always be mounted with a path prefix, such as `/mnt/vgBC<SVOL_INST>_<Device-Group-name>`.

The Volume Group created on the BCV server will always use the following syntax:

```
/dev/vgBC<SVOL_INST>_<Device-Group-name>.
```

For example, when we run the following command then we mount to the original mount points:

```
#-> /usr/local/sbin/BC-exec.sh -c /opr_dbciRCS/BC/dbciRCS_BC2.cfg mount
#-> bdf
/dev/vgBC3_vgdbRCS/lvmntRCS 20480000 5520984 14845808 27% /export/sapmnt/RCS
```

Whereas, when we add the `-F` flag to the `BC-exec.sh` script as an option will mount it with a prefix `/mnt/vgBC3_vgdbRCS`:

```
#-> /usr/local/sbin/BC-exec.sh -F -c /opr_dbciRCS/BC/dbciRCS_BC2.cfg mount
#-> bdf
/dev/vgBC3_vgdbRCS/lvmntRCS 20480000 5520984 14845808 27% /mnt/vgBC3_vgdbRCS/
↪export/sapmnt/RCS
```

## BC-exec.sh Operations

The Business Copy Operations are defined within the `BC-exec.sh` script itself and these operations will be integrated as workflows by scheduling (e.g. Tidal) scripts in a later phase. The following operations are known:

- **Validate:** check if the host system is eligible to run the script. Operating System and version will be checked. All the pre-requisite software will be checked to see if these are available on the system itself. And, of course is this host system able to use Business Copy at all. Is the configuration file for the script accessible (via automount or direct access)?
- **Resync:** pair the Business Copy disks (MU#0 or MU#1).
- **Extract:** save the source input files for the P-VOL disks on the `/opr_<SID>` path.
- **Split:** split the paired disks (must be done before the mount workflow).
- **Mount:** the S-VOL disks (be it MU#0 or MU#1) must be imported and mounted on their mount point. For MU#0 the original mount points will be used, if it presented on another system then the P-VOL system, otherwise, a prefix will be used. For the MU#1 we will mount the disks with a prefix added to the original mount points.
- **Umount:** the S-VOL disks will be un-mounted on the system and the volume group will be exported.
- **Reversesync:** the S-VOL disks will be reversed sync.ed onto the P-VOL disks (only use in case of emergency as all data will be lost which was present on the P-VOL disks as these disks will be overwritten with older data from the S-VOL side). Therefore, reversesync is a hidden operation (not shown with the help option).
- **Purgelogs:** to remove old log files like `*BC-exec*` from the `/var/adm/log` directory older than `<number>` of days (default is 30 days; typically when no day argument was given).

The default operation is “validate” when no operation value is specified. Workflows itself are a combination of operations. The workflows will be discussed later with plenty of examples.

## BC-exec.sh Configuration File

The name convention of the configuration file is typically `<package-name>_BC<i>.cfg` where `<package-name>` is the name of Serviceguard package and `<i>` is “1” for the primary Business Copy Volume and “2” is the secondary Business Copy Volume. The preferred location for the configuration file is `/opr_<package-name>/BC/<package-name>_BC<i>.cfg`

However, the configuration file can be stored anywhere, but then it is up to the user to keep it in sync between the different cluster nodes and the BCV server.

**The configuration file can come in two layout formats:**

- **LAYOUT=1.0** . up to version 1.27 of BC-exec.sh script is using the LAYOUT=1.0 format.
- **LAYOUT=2.0** . BC-exec.sh version 1.28 and beyond are able to work with both LAYOUT 1.0 and 2.0 formats (one or the other of course). The main difference will be explained below.

### Layout 1.0

The configuration file described in **LAYOUT=1.0** format is only containing variable definitions and there can only be *one* volume group defined (with corresponding device group). The configuration file will be sourced by BC-exec.sh script. Therefore, treat the content of the configuration file as a shell script.

An example configuration file might look like:

```
# Configuration file used by BC-exec.sh
# Layout:
# Option value

# PVOL_INST HORCM instance number that defines the PVOLs
# SVOL_INST HORCM instance number that defines the SVOLs
PVOL_INST=0
SVOL_INST=1

# BC_TIMEOUT is the number of seconds that pairevtwait will wait
BC_TIMEOUT=300

# DEV_GRP Device group as specified in HORCM_LDEV section
DEV_GRP=vgdbRPS

# VOL_GRP Name of the volume group on the PVOL-side
VOL_GRP=/dev/vgdbRPS

# REMOVE_CLUSTER_MODE:
# set this flag to Y if you're importing the VG that's part of a
# cluster outside that cluster
REMOVE_CLUSTER_MODE=Y

# Exclude mount points from being mounted on the BCV side
# e.g. EXCLUDE_MOUNTPOINTS="/mnt1 /mnt2"
EXCLUDE_MOUNTPOINTS="/oracle/RPS /opr_dbciRPS"

# Suspend Sync Flag (is an empty file under the same directory as
# the #configuration file).
# Variable is by default empty, if not empty then it contains a
# file name
# Attention: absolute paths are ignored as it should be in the same # directory as
# ↳the configuration file.
# If file name is used it is preferred to use "configfile.suspend"
```

```
# (without the .cfg extention)
SUSPEND_SYNC=
```

The **PVOL\_INST** variable is an integer which refers to the `/etc/horcm<PVOL_INST>.conf` file on the P-VOL system. The **SVOL\_INST** variable is an integer which refers to the `/etc/horcm<SVOL_INST>.conf` file on the BCV side.

The **DEV\_GRP** variable is part of the **HORCM\_LDEV** section of horcm configuration files (must be defined on both sides). The **VOL\_GRP** variable defines the Volume Group belonging to the DEV\_GRP and the P-VOL side. On the BCV side the Volume Group is normally not required (but it is not prohibited as we import the Volume Group with a unique name as described in section *BC-exec.sh Software Requirements*).

If the Volume Group is part of a Serviceguard cluster then we should set the variable **REMOVE\_CLUSTER\_MODE** to “Y”.

The variable **EXCLUDE\_MOUNTPOINTS** allows us to define file system to be excluded during the mounting on the BCV side. Please note, if the `/opr_<package-name>` file system is part of the Serviceguard configuration file we should add this in the **EXCLUDE\_MOUNTPOINTS** variable as otherwise on the BCV side we will get an error that it cannot mount `/opr_<package-name>` file system because it gets auto-mounted.

There is one additional variable which could be added to the configuration file:

**FORCE\_MOUNT\_PREFIX=Y** to force a mount prefix name to the mount points on the BCV side. The mount points listed in the file systems file will get mounted under `/mnt`

Be aware, if for some reason the `/opr_<package-name>` directory (mount point) is not available, because the package is down, the `BC-exec.sh` script will search for a failback configuration file under `/var/tmp/BC/<package-name>/`

The **BC\_TIMEOUT** variable is the time in seconds used by the `pairevtwait` command will wait before it bails out with an error. Be aware, that the value mentioned in the configuration file will be used for the pair and split operation. However, for the split operation the **BC\_TIMEOUT** value will be multiplied with 12 automatically.

The **SUSPEND\_SYNC** variable allows you to prevent re-syncs to happen when set. To prevent re-syncs or splits to happen you could define the following in the configuration file:

```
SUSPEND_SYNC=<package-name>_BC<i>.suspend
```

Of course you need to touch this file as follow on the NFS (exported) directory:

```
# touch /opr_<package-name>/BC/<package-name>_BC<i>.suspend
```

Be careful, you must manual remove the suspend file to release the lock on re-syncs and splits.

### Layout 2.0

The main difference with LAYOUT 1.0 format is that LAYOUT 2.0 format allows defining more than one volume group with their corresponding device groups.

An example configuration file in LAYOUT 2.0 format might look like:

```
# Configuration file used by BC-exec.sh
# Layout:
[LAYOUT]
2.0

# Option value
# PVOL_INST HORCM instance number that defines the PVOLs
```

```

[PVOL_INST]
0

# SVOL_INST HORCM instance number that defines the SVOLs
[SVOL_INST]
1

# BC_TIMEOUT is the number of seconds that pairevtwait will wait
[BC_TIMEOUT]
300

# DEV_GRP and VOL_GRP have been merged into DEVGRP_VG
# DEV_GRP Device group as specified in HORCM_LDEV section
# Per line use: Device-Group Volume-Group
[DEVGRP_VG]
vgplulogs /dev/vgplulogs
vgpludata /dev/vgpludata

# REMOVE_CLUSTER_MODE:
# set this flag to Y if you're importing the VG that's part of a
# cluster outside that cluster
[REMOVE_CLUSTER_MODE]
Y

# Exclude mount points from being mounted on the BCV side
# e.g. [EXCLUDE_MOUNTPOINTS]
# /mnt1
# /mnt2
[EXCLUDE_MOUNTPOINTS]

[SUSPEND_SYNC]

```

The meanings of the variables (enclosed with bracket braces) are the same as with LAYOUT 1.0 format. If a variable has no definition then that means that the variable is empty (see above **EXCLUDE\_MOUNTPOINTS** and **SUSPEND\_SYNC** settings).

The layout 2.0 is the deferred format for the future.

## BC-exec.sh Logging

Logging actions and steps are key in business copy operations necessary to track what happened and in case of issues to find out what and when something went wrong. Therefore, we foresee two manners of logging:

- Logging to log files
- Logging to a central syslog file

### The log files created by BC-exec.sh

The BC-exec.sh sends all output to the **/var/adm/log** directory. After each run a new file is created:

```
<ConfigFile (without .cfg)>-<operation>-BC-exec-<YYYYMMDD>-<HHMMSS>-<PID>.log
```

To view the log files use the **cat** command:

```
$ cat dbciRPS-BC1-mount-BC-exec-20131030-141414-14773.log
```

We can also modify on the command line the target log directory, therefore, use the `-D` option.

### Information Logged via syslogd

The `BC-exec.sh` script sends short messages to the `syslogd` which will be added to the `/var/adm/syslog/syslog.log` (on HP-UX) or `/var/log/messages` (Linux). To view relevant messages use the `grep` command:

```
$ grep BC-exec /var/adm/syslog/syslog.log
Sep 17 13:48:45 gltbcp01 BC-exec.sh[1765]: <error> SUSPEND_SYNC_FLAG flag [suspend_
↪synC] was set
Sep 17 13:55:58 gltbcp01 BC-exec.sh[2004]: <info> pairedisplay -IBC6 -g vgdbRPS (VG_
↪vgBC6_vgdbRPS) executed with success
Sep 17 13:57:03 gltbcp01 BC-exec.sh[2095]: <info> pairsplit -BC6 -g vgdbRPS (VG vgBC6_
↪vgdbRPS) executed with success
```

### Scheduling Jobs to use BC-exec.sh

We have discussed the general use of the tool `BC-exec.sh`, but the real force lies in the use of scheduling jobs. What do we mean by this? In general way a database need to get backed up, and therefore, we should freeze the database during the backup which could take too much time. Using business copy is a practice that is in use for a long time to overcome these kind of delays.

`BC-exec.sh` was exactly designed with this in mind, that is using it with business copy devices and not doing anything with the main production data. In big lines we must do the following:

- **extract:** On the P-Vol side (where the real production data is residing on) extracting the information about the Volume Group that we want to back-up. This will *not* interfere with any daily operational procedure and is non-disruptive. You only need to run this once, and afterwards, when the Volume Group has changed (e.g. adding disks, logical volumes and so on) again.
- **resync:** we sync the S-Vol disks (on the BCV server) with the P-Vol disks to make sure the disks are exactly the same (paired). This operation waits until the pairs are 100% complete, or when the session times out (the `BC_TIMEOUT` variable in seconds multiplied with 12)
- **Put database in backup mode:** before splitting the paired disks we must put the database in backup mode (on the P-Vol side). During this time the redo logs will be used to keep up with the transactions (on the P-Vol side).
- **split:** once the disks are 100% in pair we should split these immediately before going on (on the BCV server)
- **Bring database out of backup mode:** when the disks are splitted, then we can bring the database out of backup mode and replay all the redo logs (on the P-Vol side).
- **mount:** mount the the S-Vol disks (Volume Group) on their original mount points (or with a prefix) on the BCV server.
- **start the backup process:** now we can backup all the data mounted with our backup software which we normally use (on the BCV server). Time is of no essence as we do not interfere with the production database running on the P-Vol side.
- **umount:** when we are done with the backup we can un-mount and destroy the Volume Group. Keeping the S-Vol disks intact until the next *resync* round. In case of emergency we could always execute a reverse sync (which resync the S-Vol disks to the P-Vol disks).

## BC-exec.sh Workflows in action

We are going to show in detail how to use *BC-exec.sh* in practice. We will start on the P-Vol side (the production side, or the node where the Serviceguard package runs on).

### BC-exec.sh help

We have always a built-in help available:

```
# ./BC-exec.sh help
2016-02-03 10:05:15 ERROR: Missing argument "-c configfile"
Usage: BC-exec.sh [-c /path/configurationfile] [-m mail_destination] [-D log_
↳directory] [-Fdvh] [Operation]

    -c /path/configurationfile

    -F : Force a path prefix for MU#0 BCV (MU#1 always uses a prefix)

    -m : mail destination (default: )

    -D /path_of_log_directory (default: /var/adm/log)

    -d : debug mode (default is OFF)

    -v : show version and exit

    -h : show help (usage) and exit

Operation: supported operations are:
    validate (default)
    resync
    split
    extract
    mount
    umount
    purgelogs <number of days>

Note that we need at minimum a "-c" option
----
2016-02-03 10:05:15 LOG: Exit code 1
```

### BC-exec.sh validate

The validate workflow is only meant to inspect the *BC-exec.sh* configuration file and verify we are dealing with a proper RAID manager setup. It can be run on the P-Vol and S-Vol side at any time, and it is always called with any other operation (except for the help). Furthermore, if we call *BC-exec.sh* without any *operations* parameter, *validate* is the default one (as you can see below):

```
# ./BC-exec.sh -c ./dbciRPS.cfg
2016-02-03 10:11:11 LOG: BC-exec.sh 1.31
2016-02-03 10:11:11 LOG: BC-exec.sh -c ./dbciRPS.cfg
2016-02-03 10:11:11 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↳contrib/bin:/HORCM
2016-02-03 10:11:11 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-03 10:11:11 LOG: OPERATION=validate
```

```
2016-02-03 10:11:11 LOG: MAILUSR=
2016-02-03 10:11:11 LOG: LOGFILE=/var/adm/log/dbciRPS-validate-BC-exec-20160203-
↳101110-9825.log
2016-02-03 10:11:11 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-03 10:11:11 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-03 10:11:11 ERROR: No horcmd daemons running! Please start it manually via
↳horcmstart.sh command
2016-02-03 10:11:11 ERROR: Exit code 1
2016-02-03 10:11:11 LOG: Exit code 1
```

The above output clearly mentions that the HORCM daemons are **not** running and therefore, we get a fatal error. We executed this command on the P-Vol side. On the S-Vol side we should do the same to verify if the HORCM daemons are running:

```
# /HORCM/usr/bin/horcmstart.sh 5
starting HORCM inst 5
HORCM inst 5 starts successfully.
```

If you wonder why we use 5 for the HORCM instance number? See the HORCM configuration file saved as */etc/horcm\*.conf* and also the BC-exec.sh configuration file mention this next as *PVOL\_INST=5* value.

Re-run the *./BC-exec.sh -c ./dbciRPS.cfg* command and now you will see at the end the following:

```
2016-02-03 10:25:16 LOG: validate completed successfully.
2016-02-03 10:25:16 LOG: Exit code 0
```

### BC-exec.sh extract (on P-Vol side)

The *extract* operation should only be run on the P-Vol side as its main purpose is to collect information about the volume groups and disks belonging to this Business Copy pairs:

```
# ./BC-exec.sh -c ./dbciRPS.cfg extract
2016-02-03 10:26:50 LOG: BC-exec.sh 1.31
2016-02-03 10:26:50 LOG: BC-exec.sh -c ./dbciRPS.cfg extract
2016-02-03 10:26:50 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↳contrib/bin:/HORCM
2016-02-03 10:26:50 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-03 10:26:50 LOG: OPERATION=extract
2016-02-03 10:26:50 LOG: MAILUSR=
2016-02-03 10:26:50 LOG: LOGFILE=/var/adm/log/dbciRPS-extract-BC-exec-20160203-102650-
↳16733.log
2016-02-03 10:26:50 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-03 10:26:50 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-03 10:26:50 LOG: Start extracting source data on gltdbp01
2016-02-03 10:26:50 LOG: FILESYSTEMS=./vgdbRPS.fs
2016-02-03 10:26:50 LOG: GRPFILE=./vgdbRPS.grp
2016-02-03 10:26:50 LOG: Getting Major Minor number: ls -l /dev/vgdbRPS/group
2016-02-03 10:26:50 LOG: MAPFILE=./vgdbRPS.map
2016-02-03 10:26:50 LOG: Create mapfile ./vgdbRPS.map for vgdbRPS
vgexport: Volume group "vgdbRPS" is still active.
vgexport: Preview of vgexport on volume group "vgdbRPS" succeeded.
2016-02-03 10:26:50 LOG: Gather filesystem information for VG /dev/vgdbRPS
2016-02-03 10:26:51 LOG: Making a copy of all files under . to /var/tmp/BC/dbciRPS
2016-02-03 10:26:51 LOG: extract completed successfully.
2016-02-03 10:26:51 LOG: Exit code 0
```

After this run we will get new or updated files (on HP-UX these are):

```
# ls
dbciRPS.cfg  vgdbRPS.fs  vgdbRPS.grp  vgdbRPS.map
```

If the current directory is not NFS shared (e.g. via automounting) then manually copy over these to the same location (very important) to the BCV server (or S-Vol side):

```
# scp * bcv-server:$PWD
```

Do not forget to re-run the *extract* operation every time you modify the Volume Groups belonging to these Business Copy Groups. And, make sure that the latest files are accessible on the S-Vol side as well.

### BC-exec.sh resync (on S-Vol side)

Resyncing the Business Copy pairs is an essential part in keeping the BC disks in sync. This step is always done before re-splitting the disks to prepare for backup mode:

```
# ./BC-exec.sh -c ./dbciRPS.cfg resync
2016-02-03 10:37:50 LOG: BC-exec.sh 1.31
2016-02-03 10:37:50 LOG: BC-exec.sh -c ./dbciRPS.cfg resync
2016-02-03 10:37:50 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↳contrib/bin:/HORCM
2016-02-03 10:37:50 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-03 10:37:50 LOG: OPERATION=resync
2016-02-03 10:37:50 LOG: MAILUSR=
2016-02-03 10:37:50 LOG: LOGFILE=/var/adm/log/dbciRPS-resync-BC-exec-20160203-103750-
↳25610.log
2016-02-03 10:37:50 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-03 10:37:50 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-03 10:37:50 LOG: Start Pair Resync S-VOL disks on gltbc01
2016-02-03 10:37:50 LOG: Check if VG vgBC6_vgdbRPS is inactive.
vgdisplay: Volume group "/dev/vgBC6_vgdbRPS" does not exist in the "/etc/lvmtab" file.
vgdisplay: Volume group "/dev/vgBC6_vgdbRPS" does not exist in the "/etc/lvmtab_p"
↳file.
vgdisplay: Cannot display volume group "vgBC6_vgdbRPS".
2016-02-03 10:37:50 LOG: Execute: pairdisplay -IBC6 -g vgdbRPS -fcx
Group   PairVol(L/R) (Port#,TID, LU-M) ,Seq#,LDEV#.P/S,Status, % ,P-LDEV# M
vgdbRPS 40:06_40:4b(L) (CL1-A-3, 3, 7-0 )85827 404b.S-VOL PAIR, 99 4006 -
vgdbRPS 40:06_40:4b(R) (CL1-A-1, 0, 4-0 )85827 4006.P-VOL PAIR, 99 404b -
vgdbRPS 40:07_40:4c(L) (CL1-A-3, 4, 0-0 )85827 404c.S-VOL PAIR, 99 4007 -
vgdbRPS 40:07_40:4c(R) (CL1-A-1, 0, 5-0 )85827 4007.P-VOL PAIR, 99 404c -
vgdbRPS 40:45_40:50(L) (CL1-A-3, 2, 7-0 )85827 4050.S-VOL PAIR, 100 4045 -
vgdbRPS 40:45_40:50(R) (CL1-A-1, 3, 7-0 )85827 4045.P-VOL PAIR, 100 4050 -
2016-02-03 10:37:50 LOG: Execute: pairresync -IBC6 -g vgdbRPS
2016-02-03 10:37:51 LOG: Execute: pairevtwait -IBC6 -g vgdbRPS -t 3600 -s pair -ss
↳pair
pairevtwait : Wait status done.
2016-02-03 10:37:54 LOG: Execute: pairdisplay -IBC6 -g vgdbRPS -fcx (should show PAIR)
Group   PairVol(L/R) (Port#,TID, LU-M) ,Seq#,LDEV#.P/S,Status, % ,P-LDEV# M
vgdbRPS 40:06_40:4b(L) (CL1-A-3, 3, 7-0 )85827 404b.S-VOL PAIR, 99 4006 -
vgdbRPS 40:06_40:4b(R) (CL1-A-1, 0, 4-0 )85827 4006.P-VOL PAIR, 99 404b -
vgdbRPS 40:07_40:4c(L) (CL1-A-3, 4, 0-0 )85827 404c.S-VOL PAIR, 99 4007 -
vgdbRPS 40:07_40:4c(R) (CL1-A-1, 0, 5-0 )85827 4007.P-VOL PAIR, 99 404c -
vgdbRPS 40:45_40:50(L) (CL1-A-3, 2, 7-0 )85827 4050.S-VOL PAIR, 100 4045 -
vgdbRPS 40:45_40:50(R) (CL1-A-1, 3, 7-0 )85827 4045.P-VOL PAIR, 100 4050 -
```

```
2016-02-03 10:37:54 LOG: resync completed successfully.
2016-02-03 10:37:54 LOG: Exit code 0
```

In above output we could see that the BC disks were already paired and therefore, no *resync* was necessary. Otherwise, it would have taken more time to finish the resync operation.

### BC-exec.sh split (on S-Vol side)

We split the BC disks normally after we have put the database in backup mode so that we are sure that the data inside the database is consistent:

```
# ./BC-exec.sh -c ./dbciRPS.cfg split
2016-02-03 10:51:46 LOG: BC-exec.sh 1.31
2016-02-03 10:51:46 LOG: BC-exec.sh -c ./dbciRPS.cfg split
2016-02-03 10:51:46 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↳contrib/bin:/HORCM
2016-02-03 10:51:46 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-03 10:51:46 LOG: OPERATION=split
2016-02-03 10:51:46 LOG: MAILUSR=
2016-02-03 10:51:46 LOG: LOGFILE=/var/adm/log/dbciRPS-split-BC-exec-20160203-105146-
↳26068.log
2016-02-03 10:51:46 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-03 10:51:46 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-03 10:51:46 LOG: Start Splitting S-VOL disks on gltbcp01
2016-02-03 10:51:46 LOG: Check if VG vgBC6_vgdbRPS is inactive.
2016-02-03 10:51:47 LOG: VG vgBC6_vgdbRPS is "not" active.
2016-02-03 10:51:47 LOG: Execute: pairdisplay -IBC6 -g vgdbRPS -fcx
Group   PairVol(L/R) (Port#,TID, LU-M) ,Seq#,LDEV#.P/S,Status, % ,P-LDEV# M
vgdbRPS 40:06_40:4b(L) (CL1-A-3, 3, 7-0 )85827 404b.S-VOL PAIR, 99 4006 -
vgdbRPS 40:06_40:4b(R) (CL1-A-1, 0, 4-0 )85827 4006.P-VOL PAIR, 99 404b -
vgdbRPS 40:07_40:4c(L) (CL1-A-3, 4, 0-0 )85827 404c.S-VOL PAIR, 99 4007 -
vgdbRPS 40:07_40:4c(R) (CL1-A-1, 0, 5-0 )85827 4007.P-VOL PAIR, 99 404c -
vgdbRPS 40:45_40:50(L) (CL1-A-3, 2, 7-0 )85827 4050.S-VOL PAIR, 100 4045 -
vgdbRPS 40:45_40:50(R) (CL1-A-1, 3, 7-0 )85827 4045.P-VOL PAIR, 100 4050 -
2016-02-03 10:51:47 LOG: Execute: pairsplit -IBC6 -g vgdbRPS
2016-02-03 10:51:47 LOG: Execute: pairevtwait -IBC6 -g vgdbRPS -t 300 -s psus -ss ssus
pairevtwait : Wait status done.
2016-02-03 10:51:53 LOG: Execute: pairdisplay -IBC6 -g vgdbRPS -fcx
Group   PairVol(L/R) (Port#,TID, LU-M) ,Seq#,LDEV#.P/S,Status, % ,P-LDEV# M
vgdbRPS 40:06_40:4b(L) (CL1-A-3, 3, 7-0 )85827 404b.S-VOL SSUS, 100 4006 -
vgdbRPS 40:06_40:4b(R) (CL1-A-1, 0, 4-0 )85827 4006.P-VOL PSUS, 100 404b W
vgdbRPS 40:07_40:4c(L) (CL1-A-3, 4, 0-0 )85827 404c.S-VOL SSUS, 100 4007 -
vgdbRPS 40:07_40:4c(R) (CL1-A-1, 0, 5-0 )85827 4007.P-VOL PSUS, 100 404c W
vgdbRPS 40:45_40:50(L) (CL1-A-3, 2, 7-0 )85827 4050.S-VOL SSUS, 100 4045 -
vgdbRPS 40:45_40:50(R) (CL1-A-1, 3, 7-0 )85827 4045.P-VOL PSUS, 100 4050 W
2016-02-03 10:51:54 LOG: split completed successfully.
2016-02-03 10:51:54 LOG: Exit code 0
```

Once the split was successfully executed we can bring the database back out of backup mode to avoid too many redo log files are created and therefore, filling up the redo log directory.

### BC-exec.sh mount (on S-Vol side)

The purpose on the BCV server is to create a backup residing on the S-Vol disks without interrupting the production data (on the P-Vol disks). The backup can run as long as necessary to fulfill its job. However, before starting the

backup we should mount the file systems:

```
# ./BC-exec.sh -c ./dbciRPS.cfg mount
2016-02-03 10:58:44 LOG: BC-exec.sh 1.31
2016-02-03 10:58:44 LOG: BC-exec.sh -c ./dbciRPS.cfg mount
2016-02-03 10:58:44 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↳contrib/bin:/HORCM
2016-02-03 10:58:44 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-03 10:58:44 LOG: OPERATION=mount
2016-02-03 10:58:44 LOG: MAILUSR=
2016-02-03 10:58:44 LOG: LOGFILE=/var/adm/log/dbciRPS-mount-BC-exec-20160203-105844-
↳26277.log
2016-02-03 10:58:44 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-03 10:58:44 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-03 10:58:44 LOG: Start mounting S-VOL disks on gltbcp01
2016-02-03 10:58:44 LOG: MAPFILE=./vgdbRPS.map
2016-02-03 10:58:44 LOG: Check if we have a map file for VG vgBC6_vgdbRPS.
2016-02-03 10:58:44 LOG: GRPFILE=./vgdbRPS.grp
2016-02-03 10:58:44 LOG: Check if we have a group file for VG vgBC6_vgdbRPS.
2016-02-03 10:58:44 LOG: Check if VG vgBC6_vgdbRPS is inactive.
2016-02-03 10:58:44 LOG: VG vgBC6_vgdbRPS is "not" active.
2016-02-03 10:58:44 LOG: mkdir -p -m 755 /dev/vgBC6_vgdbRPS
2016-02-03 10:58:44 LOG: Check if our PID (26277) is locked
2016-02-03 10:58:44 LOG: lock succeeded: 26277 - /tmp/BC-exec-LOCKDIR/BC-exec-PIDFILE
2016-02-03 10:58:45 LOG: Create the /dev/vgBC6_vgdbRPS/group file
2016-02-03 10:58:45 LOG: Successfully removed the lock directory (/tmp/BC-exec-
↳LOCKDIR)
2016-02-03 10:58:45 LOG: Major, minor VG nrs are 128 0x006000 /dev/vgBC6_vgdbRPS/group
2016-02-03 10:58:45 LOG: Change the VG id on /dev/vgBC6_vgdbRPS
2016-02-03 10:58:46 LOG: Import vgBC6_vgdbRPS via mapfile ./vgdbRPS.map
vgimport: Beginning the import process on Volume Group "vgBC6_vgdbRPS".
Logical volume "/dev/vgBC6_vgdbRPS/lvmntRPS" has been successfully created
with minor number 1.
Logical volume "/dev/vgBC6_vgdbRPS/lvtransRPS" has been successfully created
with minor number 2.
Logical volume "/dev/vgBC6_vgdbRPS/lvascsRPS" has been successfully created
with minor number 4.
Logical volume "/dev/vgBC6_vgdbRPS/lvoracleRPS" has been successfully created
with minor number 5.
Logical volume "/dev/vgBC6_vgdbRPS/lvoriglogARPS" has been successfully created
with minor number 6.
Logical volume "/dev/vgBC6_vgdbRPS/lvoriglogBRPS" has been successfully created
with minor number 7.
Logical volume "/dev/vgBC6_vgdbRPS/lvmirrlogARPS" has been successfully created
with minor number 8.
Logical volume "/dev/vgBC6_vgdbRPS/lvmirrlogBRPS" has been successfully created
with minor number 9.
Logical volume "/dev/vgBC6_vgdbRPS/lvoraarcRPS" has been successfully created
with minor number 10.
Logical volume "/dev/vgBC6_vgdbRPS/lvsapreorgRPS" has been successfully created
with minor number 11.
Logical volume "/dev/vgBC6_vgdbRPS/lvsapdata1RPS" has been successfully created
with minor number 12.
Logical volume "/dev/vgBC6_vgdbRPS/lvoprRPS" has been successfully created
with minor number 13.
Logical volume "/dev/vgBC6_vgdbRPS/lvtidal" has been successfully created
with minor number 3.
Volume group "/dev/vgBC6_vgdbRPS" has been successfully created.
Warning: A backup of this volume group may not exist on this machine.
```

```

Please remember to take a backup using the vgcfgbackup command after activating the
↪volume group.
2016-02-03 10:58:46 LOG: vgchange -c n if REMOVE_CLUSTERMODE(Y) = Y
Configuration change completed.
Volume group "vgBC6_vgdbRPS" has been successfully changed.
2016-02-03 10:58:46 LOG: Activating VG vgBC6_vgdbRPS.
Activated volume group.
Volume group "vgBC6_vgdbRPS" has been successfully changed.
2016-02-03 10:58:46 LOG: Using existing mount point /export/sapmnt/RPS.
2016-02-03 10:58:46 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvmntRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:47 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,delaylog,
↪nodatainlog /dev/vgBC6_vgdbRPS/lvmntRPS /export/sapmnt/RPS
2016-02-03 10:58:48 LOG: Using existing mount point /export/usr/sap/transRPSPRD.
2016-02-03 10:58:48 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvtransRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:48 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,delaylog,
↪nodatainlog /dev/vgBC6_vgdbRPS/lvtransRPS /export/usr/sap/transRPSPRD
2016-02-03 10:58:48 LOG: Using existing mount point /usr/sap/RPS/ASCS05.
2016-02-03 10:58:48 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvascsRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:48 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,delaylog,
↪nodatainlog /dev/vgBC6_vgdbRPS/lvascsRPS /usr/sap/RPS/ASCS05
2016-02-03 10:58:48 LOG: Using existing mount point /opr_dbciRPS.
2016-02-03 10:58:48 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvoprRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:49 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,delaylog,
↪nodatainlog /dev/vgBC6_vgdbRPS/lvoprRPS /opr_dbciRPS
2016-02-03 10:58:49 LOG: Using existing mount point /opt/TIDAL/dbciRPS.
2016-02-03 10:58:49 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvtidal
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:49 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,delaylog,
↪nodatainlog /dev/vgBC6_vgdbRPS/lvtidal /opt/TIDAL/dbciRPS
2016-02-03 10:58:49 LOG: Using existing mount point /oracle/RPS/origlogA.
2016-02-03 10:58:49 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvoriglogARPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:49 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↪mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↪lvoriglogARPS /oracle/RPS/origlogA
2016-02-03 10:58:49 LOG: Using existing mount point /oracle/RPS/origlogB.
2016-02-03 10:58:49 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvoriglogBRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:50 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↪mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↪lvoriglogBRPS /oracle/RPS/origlogB
2016-02-03 10:58:50 LOG: Using existing mount point /oracle/RPS/mirrlogA.
2016-02-03 10:58:50 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvmirrlogARPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:50 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↪mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↪lvmirrlogARPS /oracle/RPS/mirrlogA

```

```

2016-02-03 10:58:50 LOG: Using existing mount point /oracle/RPS/mirrlogB.
2016-02-03 10:58:50 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvmirrlogBRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:50 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↳lvmirrlogBRPS /oracle/RPS/mirrlogB
2016-02-03 10:58:50 LOG: Using existing mount point /oracle/RPS/oraarch.
2016-02-03 10:58:50 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvoraarcRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:50 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↳lvoraarcRPS /oracle/RPS/oraarch
2016-02-03 10:58:50 LOG: Using existing mount point /oracle/RPS/sapdata1.
2016-02-03 10:58:50 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvsapdata1RPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:51 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↳lvsapdata1RPS /oracle/RPS/sapdata1
2016-02-03 10:58:51 LOG: Using existing mount point /oracle/RPS/sapreorg.
2016-02-03 10:58:51 LOG: Running fsck on /dev/vgBC6_vgdbRPS/lvsapreorgRPS
log replay in progress
replay complete - marking super-block as CLEAN
2016-02-03 10:58:51 LOG: Mounting -F vxfs -o ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct /dev/vgBC6_vgdbRPS/
↳lvsapreorgRPS /oracle/RPS/sapreorg
2016-02-03 10:58:51 LOG: Making a copy of all files under . to /var/tmp/BC/dbciRPS
2016-02-03 10:58:51 LOG: mount completed successfully.
2016-02-03 10:58:51 LOG: Exit code 0

```

You should be able to see the mounted file systems:

```

#-> mount | grep BC6
/export/sapmnt/RPS on /dev/vgBC6_vgdbRPS/lvmntRPS ioerror=mwdisable,largefiles,
↳delaylog,nodatainlog,dev=80006001 on Wed Feb 3 10:58:48 2016
/export/usr/sap/transRSPRD on /dev/vgBC6_vgdbRPS/lvtransRPS ioerror=mwdisable,
↳largefiles,delaylog,nodatainlog,dev=80006002 on Wed Feb 3 10:58:48 2016
/usr/sap/RPS/ASCS05 on /dev/vgBC6_vgdbRPS/lvascsRPS ioerror=mwdisable,largefiles,
↳delaylog,nodatainlog,dev=80006004 on Wed Feb 3 10:58:48 2016
/opt/TIDAL/dbciRPS on /dev/vgBC6_vgdbRPS/lvtidal ioerror=mwdisable,largefiles,
↳delaylog,nodatainlog,dev=80006003 on Wed Feb 3 10:58:49 2016
/oracle/RPS/origlogA on /dev/vgBC6_vgdbRPS/lvoriglogARPS ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct,dev=80006006 on Wed Feb 3
↳10:58:49 2016
/oracle/RPS/origlogB on /dev/vgBC6_vgdbRPS/lvoriglogBRPS ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct,dev=80006007 on Wed Feb 3
↳10:58:50 2016
/oracle/RPS/mirrlogA on /dev/vgBC6_vgdbRPS/lvmirrlogARPS ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct,dev=80006008 on Wed Feb 3
↳10:58:50 2016
/oracle/RPS/mirrlogB on /dev/vgBC6_vgdbRPS/lvmirrlogBRPS ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct,dev=80006009 on Wed Feb 3
↳10:58:50 2016
/oracle/RPS/oraarch on /dev/vgBC6_vgdbRPS/lvoraarcRPS ioerror=mwdisable,largefiles,
↳mincache=direct,delaylog,nodatainlog,convosync=direct,dev=8000600a on Wed Feb 3
↳10:58:50 2016

```

```
/oracle/RPS/sapdata1 on /dev/vgBC6_vgdbRPS/lvsapdata1RPS ioerror=mwdisable,largefiles,
↪mincache=direct,delaylog,nodatainlog,convosync=direct,dev=8000600c on Wed Feb 3
↪10:58:51 2016
/oracle/RPS/sapreorg on /dev/vgBC6_vgdbRPS/lvsapreorgRPS ioerror=mwdisable,largefiles,
↪mincache=direct,delaylog,nodatainlog,convosync=direct,dev=8000600b on Wed Feb 3
↪10:58:51 2016
```

### BC-exec.sh umount (on S-Vol side)

After the backup has been finished there is no need to keep the file systems mounted. And, before we run a *resync* operation we must be sure that all file systems are un-mounted and the Volume Groups are exported. We can do this in one go with this **umount** operation as you see below:

```
# ./BC-exec.sh -c ./dbciRPS.cfg umount
2016-02-03 12:04:26 LOG: BC-exec.sh 1.31
2016-02-03 12:04:26 LOG: BC-exec.sh -c ./dbciRPS.cfg umount
2016-02-03 12:04:26 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↪contrib/bin:/HORCM
2016-02-03 12:04:26 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-03 12:04:26 LOG: OPERATION=umount
2016-02-03 12:04:26 LOG: MAILUSR=
2016-02-03 12:04:26 LOG: LOGFILE=/var/adm/log/dbciRPS-umount-BC-exec-20160203-120426-
↪28254.log
2016-02-03 12:04:26 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-03 12:04:26 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-03 12:04:26 LOG: Check if VG vgBC6_vgdbRPS is active.
2016-02-03 12:04:26 LOG: VG vgBC6_vgdbRPS is active.
2016-02-03 12:04:26 LOG: Umount file system /export/sapmnt/RPS
2016-02-03 12:04:26 LOG: Umount file system /export/usr/sap/transRPSPRD
2016-02-03 12:04:26 LOG: Umount file system /usr/sap/RPS/ASCS05
2016-02-03 12:04:26 LOG: Umount file system /opr_dbciRPS
2016-02-03 12:04:26 LOG: Umount file system /opt/TIDAL/dbciRPS
2016-02-03 12:04:26 LOG: Umount file system /oracle/RPS/origlogA
2016-02-03 12:04:27 LOG: Umount file system /oracle/RPS/origlogB
2016-02-03 12:04:27 LOG: Umount file system /oracle/RPS/mirrlogA
2016-02-03 12:04:27 LOG: Umount file system /oracle/RPS/mirrlogB
2016-02-03 12:04:27 LOG: Umount file system /oracle/RPS/oraarch
2016-02-03 12:04:27 LOG: Umount file system /oracle/RPS/sapdata1
2016-02-03 12:04:27 LOG: Umount file system /oracle/RPS/sapreorg
2016-02-03 12:04:27 LOG: De-activating VG vgBC6_vgdbRPS
Volume group "vgBC6_vgdbRPS" has been successfully changed.
2016-02-03 12:04:27 LOG: Export the VG vgBC6_vgdbRPS
Beginning the export process on Volume Group "vgBC6_vgdbRPS".
/dev/disk/disk38
/dev/disk/disk40
/dev/disk/disk29
vgexport: Volume group "vgBC6_vgdbRPS" has been successfully removed.
2016-02-03 12:04:27 LOG: umount completed successfully.
2016-02-03 12:04:27 LOG: Exit code 0
```

After this the cyclus can restart with a new **resync** operation and so on.

## Suspending BC-exec.sh

There are several reasons you can think of to overrule the BC-exec.sh workflows which are normally triggered via a scheduling system. For example, for doing upgrade tests of your database on the BCV server. Afterwards, you could resync the original data and try it again as many times as you wish.

In the configuration part we already mentioned that there is a variable called `SUSPEND_SYNC` which can be set in the configuration file that belongs to your workflow. The name we use for the *suspend flag* is normally `<configuration-file-with-extention>.suspend`, e.g.:

```
SUSPEND_SYNC=vgdbRPS.suspend
```

Of course, if this file is not found under the same directory as the configuration file it will have *no* effect on the workflows of BC-exec.sh. Therefore, it is safe, to configure this always in the configuration file.

### Set the suspend flag

You need two fullfill two items:

- make sure the `SUSPEND_SYNC` variable has been defined in the configuration file
- *touch* the suspend file under the *same* directory as the configuration file on the BCV server (S-Vol side)

### Suspending the Resync

When the suspend flag has been defined on the BCV server (S-Vol side) then it will not be possible anymore to resync the disks until we clear the suspend flag again:

```
# ./BC-exec.sh -c ./dbciRPS.cfg resync
2016-02-16 09:36:05 LOG: BC-exec.sh 1.31
2016-02-16 09:36:05 LOG: BC-exec.sh -c ./dbciRPS.cfg resync
2016-02-16 09:36:05 LOG: PATH=/bin:/usr/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/
↳contrib/bin:/HORCM
2016-02-16 09:36:05 LOG: CONFIGFILE=./dbciRPS.cfg
2016-02-16 09:36:05 LOG: OPERATION=resync
2016-02-16 09:36:05 LOG: MAILUSR=
2016-02-16 09:36:05 LOG: LOGFILE=/var/adm/log/dbciRPS-resync-BC-exec-20160216-093605-
↳6135.log
2016-02-16 09:36:05 LOG: Layout of config file ./dbciRPS.cfg is layout version 1.0
2016-02-16 09:36:05 LOG: WARNING: we prefer a directory name like CONFIGDIR=./BC
2016-02-16 09:36:05 LOG: Start Pair Resync S-VOL disks on gltbcp01
2016-02-16 09:36:05 ERROR: SUSPEND_SYNC_FLAG flag [dbciRPS.suspend] was set
2016-02-16 09:36:05 ERROR: Exit code 1
2016-02-16 09:36:05 LOG: Exit code 1
```

### CheckHorcmConsistency.sh Script

In the package we deliver also another script called `CheckHorcmConsistency.sh`, which has a purpose to investigate the HORCM configuration file. We noticed that adding new disk configuration is not always done in a consistent manner. To bring these bad configurations to our attention we wrote this script.

It can be run on the P-Vol and/or S-Vol side.

### Example output of CheckHorcmConsistency.sh (P-Vol side)

The script CheckHorcmConsistency.sh output without arguments is quite short:

```
#-> /usr/local/sbin/CheckHorcmConsistency.sh
2016-02-29 07:48:41 LOG: === Horcm daemon active with instance number 0 ===
2016-02-29 07:49:24 LOG: Pairedisplay of device group vg_dataxp2 contains the same_
↳amount of disks as VG vg_dataxp2 [OK]
2016-02-29 07:49:25 LOG: Pairedisplay of device group vg_mlogxp2 contains the same_
↳amount of disks as VG vg_mlogxp2 [OK]
2016-02-29 07:49:26 LOG: Pairedisplay of device group vg_ologxp2 contains the same_
↳amount of disks as VG vg_ologxp2 [OK]
2016-02-29 07:49:26 LOG: Error count: 0
See logfiles: /var/adm/log/CheckHorcmConsistency.log and /var/tmp/
↳CheckHorcmConsistency-20160229-074839-44471.log
```

You can also ask for a more verbose output with the -v option:

```
#-> /usr/local/sbin/CheckHorcmConsistency.sh -v
2016-03-22 08:09:35 VERBOSE: CheckHorcmConsistency.sh revision 1.2
2016-03-22 08:09:35 VERBOSE: Started as: /usr/local/sbin/CheckHorcmConsistency.sh -v
2016-03-22 08:09:35 VERBOSE: LOGFILE=/var/adm/log/CheckHorcmConsistency.log
2016-03-22 08:09:35 VERBOSE: tmpLOGFILE=/var/tmp/CheckHorcmConsistency-20160322-
↳080935-44706.log
2016-03-22 08:09:35 VERBOSE: Raid Manager version is 01.29.05
2016-03-22 08:09:35 VERBOSE: Creating temporary directory /tmp/CheckHorcmConsistency_
↳11185
2016-03-22 08:09:35 VERBOSE: Capturing the Volume groups with their devices
2016-03-22 08:09:37 LOG: === Horcm daemon active with instance number 0 ===
2016-03-22 08:09:37 VERBOSE: Found /etc/horcm0.conf - analyzing...
2016-03-22 08:09:37 VERBOSE: Capturing the disks with corresponding cu_ldev number_
↳for instance number 0
2016-03-22 08:09:37 VERBOSE: Busy Processing using raidscan...
2016-03-22 08:09:45 VERBOSE: *** Inspect device group vg_dataxp2 defined with HORCM_
↳instance 0 ***
2016-03-22 08:09:46 VERBOSE: Device group vg_dataxp2 with INST (0) is defined as BC_
↳(status: SPLIT)
2016-03-22 08:09:46 VERBOSE: Save the cu:ldev numbers of the disks into culdev.vg_
↳dataxp2
2016-03-22 08:09:46 VERBOSE: Find the according culdev of /dev/mapper/devices in_
↳lvmtab.out
2016-03-22 08:10:22 VERBOSE: Created file lvmtab.culdev which maps culdev to volume_
↳groups
2016-03-22 08:10:22 VERBOSE: Find the according culdev to device group vg_dataxp2
2016-03-22 08:10:22 VERBOSE: Created file lvmtab.BC0.vg_dataxp2 which maps culdev to_
↳volume group of device group vg_dataxp2
2016-03-22 08:10:22 VERBOSE: Compare the devices in Device Group vg_dataxp2 with the_
↳corresponding Volume Group vg_dataxp2
2016-03-22 08:10:22 LOG: Pairedisplay of device group vg_dataxp2 contains the same_
↳amount of disks as VG vg_dataxp2 [OK]
2016-03-22 08:10:22 VERBOSE: *** Inspect device group vg_mlogxp2 defined with HORCM_
↳instance 0 ***
2016-03-22 08:10:23 VERBOSE: Device group vg_mlogxp2 with INST (0) is defined as BC_
↳(status: SPLIT)
2016-03-22 08:10:23 VERBOSE: Save the cu:ldev numbers of the disks into culdev.vg_
↳mlogxp2
2016-03-22 08:10:23 VERBOSE: Find the according culdev to device group vg_mlogxp2
2016-03-22 08:10:23 VERBOSE: Created file lvmtab.BC0.vg_mlogxp2 which maps culdev to_
↳volume group of device group vg_mlogxp2
```

```

2016-03-22 08:10:23 VERBOSE: Compare the devices in Device Group vg_mlogxp2 with the_
↳corresponding Volume Group vg_mlogxp2
2016-03-22 08:10:23 LOG: Pairedisplay of device group vg_mlogxp2 contains the same_
↳amount of disks as VG vg_mlogxp2 [OK]
2016-03-22 08:10:23 VERBOSE: *** Inspect device group vg_ologxp2 defined with HORCM_
↳instance 0 ***
2016-03-22 08:10:23 VERBOSE: Device group vg_ologxp2 with INST (0) is defined as BC_
↳(status: SPLIT)
2016-03-22 08:10:23 VERBOSE: Save the cu:ldev numbers of the disks into culdev.vg_
↳ologxp2
2016-03-22 08:10:24 VERBOSE: Find the according culdev to device group vg_ologxp2
2016-03-22 08:10:24 VERBOSE: Created file lvmtab.BC0.vg_ologxp2 which maps culdev to_
↳volume group of device group vg_ologxp2
2016-03-22 08:10:24 VERBOSE: Compare the devices in Device Group vg_ologxp2 with the_
↳corresponding Volume Group vg_ologxp2
2016-03-22 08:10:24 LOG: Pairedisplay of device group vg_ologxp2 contains the same_
↳amount of disks as VG vg_ologxp2 [OK]
2016-03-22 08:10:24 VERBOSE: Remove all temporary files [executed: rm -rf /tmp/
↳CheckHorcmConsistency_11185]
2016-03-22 08:10:24 LOG: Error count: 0
See logfiles: /var/adm/log/CheckHorcmConsistency.log and /var/tmp/
↳CheckHorcmConsistency-20160322-080935-44706.log

```

### Example output of CheckHorcmConsistency.sh (S-Vol side)

We can run this script also on the BCV server (S-Vol side):

```

#-> /usr/local/sbin/CheckHorcmConsistency.sh
2016-02-29 07:47:22 LOG: === Horcm daemon active with instance number 3 ===
2016-02-29 07:48:06 LOG: Pairedisplay of device group vg_datagpz contains the same_
↳amount of disks as VG vgBC3_vg_datagpz [OK]
2016-02-29 07:48:06 LOG: Pairedisplay of device group vg_mloggpz contains the same_
↳amount of disks as VG vgBC3_vg_mloggpz [OK]
2016-02-29 07:48:07 LOG: Pairedisplay of device group vg_ologgpz contains the same_
↳amount of disks as VG vgBC3_vg_ologgpz [OK]
2016-02-29 07:48:07 LOG: === Horcm daemon active with instance number 1 ===
2016-02-29 07:48:13 LOG: Pairedisplay of device group vg_dataxp2 contains the same_
↳amount of disks as VG vgBC1_vg_dataxp2 [OK]
2016-02-29 07:48:14 LOG: Pairedisplay of device group vg_mlogxp2 contains the same_
↳amount of disks as VG vgBC1_vg_mlogxp2 [OK]
2016-02-29 07:48:15 LOG: Pairedisplay of device group vg_ologxp2 contains the same_
↳amount of disks as VG vgBC1_vg_ologxp2 [OK]
2016-02-29 07:48:15 LOG: Error count: 0
See logfiles: /var/adm/log/CheckHorcmConsistency.log and /var/tmp/
↳CheckHorcmConsistency-20160229-074719-55395.log

```

Whenever, an issue is noticed you will see a warning so you can investigate deeper to resolve the HORCM configuration on both sides (P-Vol and S-Vol side).

### Script horcmd-initscript-rhel-script.sh

The HORCM installation comes without an `init1` start-up script, which means after each reboot of the system you need to restart their horcmd daemons manually again. This is not very practical and even bad system administration

<sup>1</sup> <http://fedoraproject.org/wiki/EPEL:SysVInitScript>

practice. Therefore, we wrote this small script to automate this task for you at least on RedHat Linux versions (in my customer base it was mostly RHEL 6). For RHEL 7 we probably need a systemd kind of script. Also, for other Linux distributions this script may not be 100% compatible, but at least it will give an indication how an initscript should look like.

The script is self explaining and does not need any manually intervention. If horcmd daemons are running it will automatically created the proper settings in */etc/sysconfig/horcmd*

The source of the script is available at [GitHub](#)

If you create a similar init script for another kind of Linux distribution let me know via opening an [issue on GitHub](#) and we will be glad to add it to the HORCM Utilities.

### Citations

## CHAPTER 4

---

### Indices and tables

---

- `genindex`