

# DataProtector High Availability (DPHA)

Greg Baker (gregb@ifost.org.au)

November 19, 2013

## Contents

<b>1</b>	<b>device-replicator.pl</b>	<b>5</b>
<b>2</b>	<b>mcfSEND.pl</b>	<b>6</b>
<b>3</b>	<b>mcfRECEIVE.pl</b>	<b>8</b>
<b>4</b>	<b>pool-replicator.pl</b>	<b>10</b>
<b>5</b>	<b>dp-move-clients.pl</b>	<b>11</b>

## Introduction

HP DataProtector has a central cell manager which requires some work to avoid it being a single point of failure. This is because no backups run if the cell manager is down.

There are a few other minor single points of failure which can easily be worked around (e.g. make sure that you have at least two servers able to perform the role of robotic control for each library, and likewise for the role of media agent for each tape drive), but the cell manager is trickier. There are five ways of making the HP DataProtector cell manager highly available.

- Traditional clustering using ServiceGuard, Veritas or Windows Clusters. HP supports this and the procedures for doing so are in the Installation and Licensing Guide. Generally these solutions can only exist in a single IP subnet. If you want to have cell-manager capabilities across different sites, this will require some sophisticated networking setup.

- Virtualisation solutions (such as VMWare Site Recovery) let you abstract the hardware dependency. If the hardware your cell manager is running on fails, you can restart it on another ESX server. With version 8 this has become more practical – the database in version 7 and before imposed a heavy I/O load which would be exacerbated by virtualisation. Again, clever networking set up is required to let this work across different IP subnets.
- Automated internal database restore to another machine was relatively simple in version 7, but much harder to get working in version 8. The general idea is to have an internal database backup job run `omnir` to recover the database to another directory, replicate that to another server and have a script at the receiving server which can activate that replicated data.
- Use write-ahead-logs from the version 8 (postgresql) database. I've no idea how to do this yet, and don't know if it would even work.
- Run two cell managers and keep them synchronised. There is no constraint on where these servers are – one can be in a disaster recovery site and the other can be in a production site.

This document describes the scripts that I have put together to automate the last of these. They can be used for other purposes as well. In particular `pool-replicator.pl` and `device-replicator.pl` are very useful when transitioning between servers.

## Overview and Installation

The `dpha` scripts only require DataProtector version 6.11 or newer.

To begin, install one HP DataProtector cell manager and configure the cell as you normally would: install clients, configure media agents, configure any tape libraries and media pools that you need to.

Then, install a second cell manager.

There are five parts to synchronise between the two cells:

1. The device definitions and media pools are part of the binary database. The two scripts `pool-replicator.pl` and `device-replicator.pl` do this. They can be run by hand after changes are made or scheduled to run automatically.
2. Media used and sessions are also part of the binary database. The pair `mcfSEND.pl` and `mcfRECEIVE.pl` can keep these in sync between servers. Create a Notification to run `mcfSEND.pl` at the end

of each backup, and create a scheduled task to run `mcfreceive.pl`. Do both of these on both servers. On Windows also create a shared folder on both servers and alter the destination directories listed in `mcfreceive.pl` to reflect where this shared folder is.

3. The configuration files. See the notes in the next paragraph.
4. Integration oddities and fragments. The only one I have found so far is the files in the folder `Config/Server/Integ/Config/E2010` which seem to be updated after each backup.
5. Session messages aren't particularly important. I haven't even bothered to try synchronising these. There are also checkpoint files in `Config/Server/Sessions/checkpoint` which

At the moment I don't have a config-file synchroniser: it is something I want to fix, but I'm not sure of the best way to proceed. Anyway, because the are plain text files, it's only a matter of copying them. Between the two servers. I've used `robocopy` (or `rsync` between Linux boxes) but I suspect `unison` would be better even though I've never tried it. There are only a few files to handle carefully:

**Config/Server/Cell/cell.info** You can't just copy from the first cell manager to the second cell manager initially, because that would remove the second cell manager from its own cell! Append the contents of the second cell manager's `cell.info` file on to the first cell manager's and *then* copy the resulting appended file. This will work, but be aware of client needs (in the next section).

**Config/Server/Cell/installation.servers** Same technique as for `cell.info`

**Config/Server/Cell/lic.dat** This could probably be merged somehow; I haven't investigated this yet.

**Config/Server/Users/userlist** Similarly to `cell.info`, append the two cell manager's files together and then remove any duplicates.

**Config/Server/Notifications** Because `mcfsend.pl` is triggered from here, the two cell managers will have different `Notifications` files.

**Config/Server/IDB** Don't replicate this folder on version 8 (it's not present on earlier versions). This has the usernames and passwords to access the database.

**Schedule files** Obviously you only want one server to initiate backups normally. So there's a requirement to manually de-schedule all backups on one server. The relevant folders in which schedule

files are found: barschedules, schedules, amoschedules, copy-lists/scheduled, consolidationlist/scheduled, verificationlists/scheduled and rptschedules.

One final note: in patches to version 6.11, HP introduced a new parameter: `SmFirstSessionOfDay`. While DataProtector can cope with two sessions having the same session ID, it is very confusing for the human operators, so it is best if this is avoided. By setting the two cell managers to completely different start points, separated by more than a day's session count (e.g. 100 versus 500 when you have less than 400 backups running per day) you can guarantee that two cell managers are not going to generate sessions with the same session number. This only takes effect on the next day, so if you are in a hurry, you can use `omnidbutil -set-session-counter`.

## Client Needs During Fail-over

The cell manager lists all the clients in the cell in `Config/Server/Cell/cell.info`. The clients keep track of which cell manager they report to.

On Windows clients, it is stored in a registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Hewlett-Packard\OpenView\OmniBackII\Site\CellServer`. On other platforms, it is stored in `/etc/opt/omni/client/cell.server`.

Most DataProtector clients can cope just fine if the cell manager listed in their local config does not match the cell manager which is sending them orders. The disk agent and media agent check against the `cell_secure` file (which is not present by default), but do not care whether an incoming connection comes from a cell manager or not.

However, most of the integration agents do have problems. MS-SQL, MS-VSS, Exchange and VEAagents (and possibly others) keep some of their configuration on the cell manager. When an integration backup is started, the agent connects to the *cell manager listed in their local configuration* to get permission information and to record messages. This means that an integration session initiated from the "other" cell manager will fail, or weider, generate a session in the original cell manager.

Ultimately, it's just a matter of fixing up the registry key / `cell.server` config file on the clients. This happens automatically if the client is exported and then imported. `dp-move-clients.pl` generates shell and batch scripts to run `omnicc -export_host` and `omnicc -import_host`.

## 1 device-replicator.pl

A script to make a two cell managers have the same pools

### SYNOPSIS

device-replicator.pl *from cellmgr*

device-replicator.pl *to cellmgr*

### DESCRIPTION

When called with *from* as the first argument, it will connect to *cellmgr* and fetch all device and library information from it. It will then create any libraries and devices which the local cell manager does not have, and modify any that already exist to match the remote server.

When called with *to* it works the opposite way around: the devices and libraries from the local cell manager are recreated on the cell manager given as an argument.

### REQUIREMENTS

You will need the cell console software installed on the computer running *device-replicator.pl* and the user that runs it needs to have media configuration rights on both cells.

You almost definitely will want to run *pool-replicator.pl* before you run *device-replicator.pl*

### SEE ALSO

*pool-replicator.pl*

### BUGS

Probably many. It hasn't been tested on every variation yet.

### COPYRIGHT

Copyright 2013 [[The Institute for Open Systems Technologies Pty Ltd.]]

## **2 mcfSEND.pl**

A script to export MCF files after a backup

### **SYNOPSIS**

`mcfSEND.pl destination-path`

### **DESCRIPTION**

If you want to keep two HP DataProtector cells in sync, you need to transfer data about each backup and copy session. Set this script to run as a post-exec after each backup job, or better: set this script as an external notification for EndOfSession.

*OK, I should make an install option for this script so that it automatically sets up a notification.*

Don't forget to set `SmFirstSessionOfDay` differently on the two cell managers, and run `omnidbutil -set_session_counter` if you don't plan on waiting until tomorrow to start working.

Pass a path name as an argument. It will pick up `SESSIONID` from the environment to find the session number and to look up what media were used.

This is a pair with `mcfreceive.pl` which would run as a scheduled job.

### **EXAMPLE**

```
mcfSEND.pl \\otherserver\replication
```

### **SEE ALSO**

*mcfreceive.pl*

### **COPYRIGHT**

Copyright 2013 [The Institute for Open Systems Technologies Pty Ltd.]

## **BUGS**

This script has only been tested on Windows. It won't be hard to change to run on Unix.

This script always dumps the MCF files in the same hard-coded directory (`$base_dir`). This should be configurable and/or it should always use the default and/or it should be smart enough to figure out a reasonable default.

If the remote server is down, this script never gets around to trying to copy the MCF files over again.

### 3 mcfreceive.pl

Process incoming MCF files

#### SYNOPSIS

mcfreceive.pl

#### DESCRIPTION

This script is a pair with *mcfsend.pl*.

If you want to keep two HP DataProtector cells in sync, you need to transfer data about each backup and copy session. *mcfsend.pl* does this and is normally installed as an external notification for EndOfSession.

*mcfreceive.pl* is normally run as a scheduled task.

*OK, I should set up an installer option to set this up automatically.*

*mcfreceive.pl* looks at an incoming directory (hard-coded in the script) looking for folders *\*.ready*. It then looks at all the media being mentioned by the *\*.mcf* files in that directory and then exports those media from the database. (To do this it has to change the backup protection on all sessions that use that media).

Finally, once it has forgotten everything to do with that media, it imports the MCF files, which should re-create all relevant information about the session.

#### BUGS

Protected data could get lost under the following circumstances:

- A backup object fills up one tape and then runs over into the next because the pool is set up as appendable.
- That session finishes and is replicated with *mcfsend.pl* to the other server.
- Another backup is run on to the second tape.
- That second session is replicated over to the other server
- *mcfreceive.pl* has to expire the data on the first backup object in order to export the second tape.



- The first session will not be completely re-created by the MCF files which are imported.

This appears to be a limitation of `omnim -import_from_mcf`

At the moment this script only works on Windows, but it won't be hard to fix to get it to run on HP-UX or Linux. It should work with DataProtector 6.11 or later (including version 8), but it is probably only safe to have this kind of catalog merging in version 8 and beyond.

## **COPYRIGHT**

Copyright 2013 [[The Institute for Open Systems Technologies Pty Ltd.]]

## 4 pool-replicator.pl

A script to make a two cell managers have the same pools

### SYNOPSIS

pool-replicator.pl *from* *cellmgr*

pool-replicator.pl *to* *cellmgr*

### DESCRIPTION

When called with *from* as the first argument, it will connect to *cellmgr* and fetch all pool information from it, including free pool information. It will then create any pools which the local cell manager does not have, and modify any that already exist to match the remote server.

When called with *to* it works the opposite way around: the pools from the local cell manager are recreated on the cell manager given as an argument.

### REQUIREMENTS

You will need the cell console software installed on the computer running *pool-replicator.pl* and the user that runs it needs to have media configuration rights on both cells.

### SEE ALSO

*device-replicator.pl*

### BUGS

Probably many. It hasn't been tested on every variation yet.

### COPYRIGHT

Copyright 2013 [[The Institute for Open Systems Technologies Pty Ltd.]]

## 5 dp-move-clients.pl

Generate a script to export/import every client in a cell

### SYNOPSIS

```
dp-move-clients.pl [-export_only] [client-skip-list...]
```

### DESCRIPTION

This program generates a script which can export/import every client in a cell. This is a very convenient thing to do when you are migrating from one cell manager to another. On the old cell manager you run:

```
dp-move-clients.pl  
dp-move-clients.pl -export_only
```

Redirect each to a `.bat` file on Windows or to a `.sh` file on Unix. Run the second script on the old cell manager first, and then copy the first script to the new cell manager, and run it there.

### OUTPUT

`dp-move-clients.pl` reads through the output of `omnicellinfo -cell` and writes `omnicc` commands for each one of them.

If `-export_only` is specified, it will just output `omnicc -export_host` commands, otherwise there will also be a following `omnicc -import_host` command.

If a *client-skip-list* is specified, it is a list of names of servers to ignore – no `omnicc` command will be generated for these.

### COPYRIGHT

Copyright 2013 [The Institute for Open Systems Technologies Pty Ltd.]

### BUGS

If a client is unreachable then the import might fail.

It's not much good having a script to `-export_only` if the original cell manager is down.

It doesn't handle clusters or virtualisation properly.