Perforce Software, Inc.

510.864.7400

consulting@perforce.com

[www.perforce.com](http://www.perforce.com/)

Disaster Recovery Failover Plan

Unix

Standby Replica Server

Table of Contents

[1 Overview 1](#_Toc480964391)

[2 Failover Procedure 1](#_Toc480964392)

[2.1 Check replica status 1](#_Toc480964393)

[2.2 Stop access to master server – done on master server 1](#_Toc480964394)

[2.3 Check replica status 1](#_Toc480964395)

[2.4 Stop the Replica 1](#_Toc480964396)

[2.5 Change the replica to become the master 2](#_Toc480964397)

[2.6 Update the crontab (Run all commands as perforce user) 2](#_Toc480964398)

[2.7 Check the edge and replica server status 3](#_Toc480964399)

[2.8 Change the original master to become the replica 3](#_Toc480964400)

# Overview

The DR server is being replicated using the Perforce built in server to server replication via the “p4 journalcopy” command and a rsync script to replicate checkpoints. This document provides the steps to failover from the main server to the DR server.

NOTE: This is a generic document that needs to be customized for the specific set up in your environment.

For an unplanned failover, start at section 2.4 below.

# Failover Procedure

## Check replica status

This is just a verification that replication is up to date before we start the whole process.

On the replica machine, run p4 pull -lj to check the status of the replication. The command should return two matching journal sequence numbers.

## Stop access to master server – done on master server

Change the protect table to block access from everyone except the admin user and the service user. To do that, make the last three lines of the protect take look like this:

list user \* \* -//…

super user service \* \* //…

super user perforce \* \* //…

Next, run:

/p4/1/bin/p4d\_1 -r /p4/1/root -J /p4/1/logs/journal -jj /p4/1/checkpoints/p4\_1

The command above is run on the master, and the purpose of it is to make the cause all of the replicas to rotate their journal. That way, if you have to reset a replica after the failover, it has a very small journal to catch up from on the new master.

## Check replica status

On the replica machine, run p4 pull -lj to check the status of the replication. You want to wait for the replica to fully catch up before shutting down the master and replica instances. Once you have verified the replication is up to date:

stop the master by running /p4/1/bin/p4d\_1\_init stop on the master server

## Stop the Replica

stop the replica by running

/p4/1/bin/p4d\_1\_init stop on the replica server

Now look in /p4/1/journals.rep for the highest numbered p4\_1.jnl.XXX file in the folder and move that file to /p4/1/logs/journal replacing the empty one that is there.

Check the /p4/1/root/state and /p4/1/root/statejcopy files. If the sequence number in the files do not match, then replay the journal into the root folder:

/p4/1/bin/p4d\_1 -r /p4/1/root -jr -f /p4/1/logs/journal

If this was an unplanned failover where the master died, run:

/p4/1/bin/p4d\_1 -r /p4/1/root -J /p4/1/logs/journal -jj /p4/1/checkpoints/p4\_1

## Change the replica to become the master

Change the DNS entry or move the VIP from the master to the new master if you are using a VIP address.

On the replica server, edit /p4/<INSTANCE>/root/server.id and change the name to the name of the MASTER server id and save the file. Now you can restart the server as the master by running /p4/1/bin/p4d\_1\_init start

Edit the protect table and remove the list user \* \* -//… line that was added in the earlier step if this is a planned failover.

*After this is done, the former replica is now the master.*

Next run:

rm /p4/1/root/state

rm /p4/1/root/statejcopy

rm /p4/1/root/rdb.lbr

## Update the crontab (Run all commands as perforce user)

On the new master server, originally the replica, run crontab /p4/p4.crontab to reset the crontab to run the master server crontab settings.

On the original master server, now the replica, run crontab /p4/p4.crontab.replica to change it to run the replica scripts.

It is a good idea to use crontab -l > p4.crontab on the master and crontab -l > p4.crontab.replica on the replica before you make these changes on each machine just to be sure that you have the current crontab settings saved. You can copy these files to the other machine to make sure you load the current settings on the other machine as well.

## Check the edge and replica server status

On each edge and replica, make sure that the servers are replicating properly by running:

p4 pull -lj

If they are not, then do the following on the edge/replica:

1. /p4/1/bin/p4d\_1\_init stop
2. /p4/common/bin/p4master\_run 1 p4 -p <master\_server:port> login < /p4/common/config/.p4passwd.p4\_1.admin
3. /p4/common/bin/p4master\_run 1 p4 -p <master\_server:port> login service
4. rm -f /p4/1/root/state /p4/1/root/rdb.lbr
5. rm -f /p4/1/root/statejcopy /p4/1/journals.rep/\* (These only exist if the replica is using journalcopy rather than pull)
6. /p4/1/bin/p4d\_1\_init start

## Change the original master to become the replica

On the original master, edit /p4/1/root/server.id and change the name to the name of the REPLICA server id and save the file.

Move /p4/1/logs/journal, e.g. to /p4/1/logs/journal.orig.master

Delete the following files if they exist:

/p4/1/root/state

/p4/1/root/statejcopy

/p4/1/root/rdb.lbr

You also need to login into the master server as the service user on the replica so that replication can run.

Run:

1. /p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port -u service trust (Only if your site is using SSL)
2. /p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port login < /p4/common/config/.p4passwd.p4\_1.admin
3. /p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port login service

Now, start the server as a replica by running:

/p4/1/bin/p4d\_1\_init start

Check replication by running:

p4 login < /p4/common/config/.p4passwd.p4\_1.admin

p4 pull -lj

You should see two matching journal sequence numbers. The p4 pull –lj will only succeed when run on a replica pulling from a master. So this proves that the roles have been switched.

Finally, run these final two steps:

On the new replica server, run the sync\_replica.sh command that cron normally runs in order to make sure it is working properly.

/p4/common/bin/p4master\_run 1 /p4/common/bin/sync\_replica.sh

NOTE 1: *If the original master server has been down for more than 7 days*, you will have to reset the replica as follows before starting it. The reason for the 7 days value is that the number of old checkpoints and journals to keep is set to 7 in the p4\_vars file. The replication uses the old journals in order to catch up from where it last stopped, so if the old ones have rotated off, you have to reset the replica with a new checkpoint and updated versioned files.

To reset the replica, from the replica run:

If your servers are set up to rsync without a password, you can reset the replica by running:

/p4/common/bin/p4master\_run 1 /p4/common/bin/recreate\_db\_sync\_replica.sh

recreate\_db\_sync\_replica.sh performs the following steps:

rsync -avz --delete perforce@master\_server:/p4/1/checkpoints/ /p4/1/checkpoints

rm -r -f /p4/1/root/db.\*

rm -r -f /p4/1/offline\_db/db.\*

rm -r -f /p4/1/logs/journal

ls -lah /p4/1/checkpoints

Make note of the highest numbered p4\_1.ckp.#.gz file in that directory. Assume it is 10 for this example:

/p4/1/bin/p4d\_1 -r /p4/1/root -jr -z /p4/1/checkpoints/p4\_1.ckp.10.gz

/p4/1/bin/p4\_1 -p master\_server:port -u service login

/p4/1/bin/p4d\_1\_init start

/p4/1/bin/p4d\_1 -r /p4/1/offline\_db -jr –z /p4/1/checkpoints/p4\_1.ckp.10.gz

Optionally, you may want to run:

rsync -avz --delete perforce@master\_server:/p4/1/depots/ /p4/1/depots

or

/p4/common/bin/p4master\_run 1 /p4/common/bin/p4verify.sh

to make sure the depot files are up to date.

NOTE 2: If the master server is inoperable, follow these steps:

1. On the replica server, run /p4/1/bin/p4d\_1\_init stop
2. Do the steps in Section 2.4 above