

Perforce Software, Inc.

510.864.7400

[consulting@perforce.com](mailto:consulting@perforce.com)

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

SDP Failover Guide   
(2018.1)

(first draft)

Table of Contents

[1 Commit-Edge UNIX Failover Overview 3](#_Toc528841426)

[2 Commit-Edge UNIX Failover Procedure 3](#_Toc528841427)

[2.1 Check replica status 3](#_Toc528841428)

[2.2 Limit access to master server 3](#_Toc528841429)

[2.3 Rotate the journal 3](#_Toc528841430)

[2.4 Check replica status 4](#_Toc528841431)

[2.5 Shutdown the replica and master instance(s) 4](#_Toc528841432)

[2.6 Copy the master journal 4](#_Toc528841433)

[2.7 Change the replica to become the master 4](#_Toc528841434)

[2.8 Update the crontab 4](#_Toc528841435)

[2.9 Change the original master to become the replica 5](#_Toc528841436)

[3. DR Failover UNIX “Master-Replica” 6](#_Toc528841437)

[3.1 Check replica status 6](#_Toc528841438)

[3.2 Limit access to master server 6](#_Toc528841439)

[3.3 Check replica status 6](#_Toc528841440)

[3.4 Stop the Servers 6](#_Toc528841441)

[3.5 Change the replica to become the master 7](#_Toc528841442)

[3.6 Update the crontab 7](#_Toc528841443)

[3.7 Check the edge and replica server status 7](#_Toc528841444)

[3.8 Change the original master to become the replica 8](#_Toc528841445)

[4 DR Failover Windows 10](#_Toc528841446)

[Procedure: Introduction 11](#_Toc528841447)

[Check replica status 11](#_Toc528841448)

[Change the replica to become the master 11](#_Toc528841449)

[Update the scheduled jobs 11](#_Toc528841450)

[Change the original master to become the replica 12](#_Toc528841451)

# Commit-Edge UNIX Failover Overview

This document provides the steps to fail over from the main server to the disaster recovery (DR) server **in a UNIX commit-edge environment**. This is a generic document that needs to be customized for the specific set up in your environment.

The DR server is being replicated using the Perforce built in server-to-server replication via the “**p4 pull**” command and a rsync script to replicate checkpoints.

Also, this process depends on you using **sync\_shared\_replica.sh** on the commit server cluster that is using shared depotdata storage.

If this is an unplanned failover, shut down replica instances and start at **Step 2.7**. You will have to reset the Edge servers as well by stopping them, removing the state and journal files and starting them back up after you bring the new master on-line. You also won't be able to do **Step 2.9** until the original master is working again.

# Commit-Edge UNIX Failover Procedure

## Check replica status

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

On the replica machine,

1. Run p4 pull -lj to check the status of the replication.
2. Verify that the command returns two matching journal sequence numbers.

## Limit access to master server

Change the protect table to block access from everyone except the admin user and the service user by editing the last three lines of the **protect** table to look like this:

list user \* \* -//…

super user service \* \* //…

super user perforce \* \* //…

## Rotate the journal

Rotate the journal on the master server instance(s) by using:

/p4/common/bin/p4master\_run <instance> p4 admin journal

## Check replica status

On the replica machine,

1. Run p4 pull -lj to check the status of the replication.
2. Wait for the replica to fully catch up before shutting down the master and replica instances.

## Shutdown the replica and master instance(s)

Once you have verified the replication is up to date:

1. Stop the replica by running /p4/1/bin/p4d\_1\_init stop on the replica server
2. Stop the master by running /p4/1/bin/p4d\_1\_init stop on the master server

## Copy the master journal

1. On the master server, for each instance, run:

cp /p4/<instance>/logs/journal /depotdata/p4/<instance>/journal

1. On the replica server, for each instance run:

mv /depotdata/p4/<instance>/journal /p4/<instance>/logs/journal

(The reason for copying the journals over from the original master is to keep them byte identical, and avoid having to reset all of the Edge servers.)

## 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.

1. 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.
2. Restart the server as the master by running /p4/1/bin/p4d\_1\_init start

*The former replica is now the master.*

1. Run:

rm /p4/1/root/state

rm /p4/1/root/rdb.lbr

1. Run p4 protect and remove the lines you added to block access.

## Update the crontab

It is a good idea to use crontab -l > /depotdata/p4/p4.crontab on the master and crontab -l > /depotdata/p4/p4.crontab.replica on the replica before you make changes on each machine 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.

Run the following commands as the perforce user.

1. On the new master server, originally the replica, run crontab /depotdata/p4/p4.crontab to reset the crontab to run the master server crontab settings.
2. On the original master server, now the replica, run crontab /depotdata/p4/p4.crontab.replica to change it to run the replica scripts.

## Change the original master to become the replica

1. 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.
2. rm /p4/1/logs/journal
3. Delete the following files if they exist: /p4/1/root/state or /p4/1/root/rdb.lbr .
4. Log in to the master server as the service user on the replica so that replication can run.

/p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port -u service trust (Only if your site is using SSL)

/p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port login < /p4/common/bin/adminpass

/p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port login service

1. Start the server as a replica by running /p4/1/bin/p4d\_1\_init start
2. Check replication by running:

p4 login < /p4/common/bin/adminpass

p4 pull -lj

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

1. On the new replica server, run the sync\_shared\_replica.sh command that cron normally runs:

/p4/common/bin/p4master\_run 1

/p4/common/bin/sync\_shared\_replica.sh

# DR Failover UNIX “Master-Replica”

This sections provides the steps for a failover from the main server to the disaster recovery (DR) server in a **master-replica UNIX environment**. This is a generic document that needs to be customized for the specific set up in your environment.

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.

For an unplanned failover, start at **Section 3.4** below.

## Check replica status

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

On the replica machine,

1. Run p4 pull -lj to check the status of the replication.
2. Verify that the command returns two matching journal sequence numbers.

## Limit access to master server

On the master server, change the protect table to block access from everyone except the admin user and the service user by editing the last three lines of the **protect** table to look like this:

list user \* \* -//…

super user service \* \* //…

super user perforce \* \* //…

## Check replica status

On the replica machine,

1. Run p4 pull -lj to check the status of the replication.
2. Wait for the replica to fully catch up before shutting down the master and replica instances.

## Stop the Servers

Once you have verified the replication is up-to-date:

1. Stop the replica by running /p4/1/bin/p4d\_1\_init stop on the replica server
2. Stop the master by running /p4/1/bin/p4d\_1\_init stop on the master server

## Change the replica to become the master

Change the DNS entry for the machine you are failing over to point to the new master server.

On the replica server, edit /p4/<INSTANCE>/root/server.id and change the name to master and save the file.

1. Restart the server as the master by running /p4/1/bin/p4d\_1\_init start

*The former replica is now the master.*

1. Run:

rm /p4/1/root/state

rm /p4/1/root/rdb.lbr

## Update the crontab

It is a good idea to use crontab -l > /depotdata/p4/p4.crontab on the master and crontab -l > /depotdata/p4/p4.crontab.replica on the replica before you make changes on each machine 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.

Run the following commands as the perforce user.

1. On the new master server, originally the replica, run crontab /depotdata/p4/p4.crontab to reset the crontab to run the master server crontab settings.
2. On the original master server, now the replica, run crontab /depotdata/p4/p4.crontab.replica to change it to run the replica scripts.

## Check the edge and replica server status

On each edge and replica,

1. Make sure that the servers are replicating properly by running p4 pull -lj
2. If they are not, then do the following on the edge/replica:
3. /p4/1/bin/p4d\_1\_init stop
4. /p4/common/bin/p4master\_run 1 p4 -p <master\_server:port> login < /p4/common/config/.p4passwd.p4\_1.admin
5. /p4/common/bin/p4master\_run 1 p4 -p <master\_server:port> login service
6. rm -f /p4/1/root/state /p4/1/root/rdb.lbr
7. rm -f /p4/1/root/statejcopy /p4/1/journals.rep/\* (These only exist if the replica is using journalcopy rather than pull)
8. /p4/1/bin/p4d\_1\_init start

## Change the original master to become the replica

On the original master,

1. Edit /p4/1/root/server.id and change the name to the name of the REPLICA server id and save the file.
2. Move /p4/1/logs/journal, e.g. to /p4/1/logs/journal.orig.master
3. Delete the following files if they exist:

/p4/1/root/state

/p4/1/root/rdb.lbr

1. Log in into the master server as the service user on the replica so that replication can run.  
     
   /p4/common/bin/p4master\_run 1 p4 -p ssl:master\_dns\_name:1666 login < /p4/common/config/.p4passwd.p4\_1.admin

/p4/common/bin/p4master\_run <instance> p4 -p ssl:master:port login service

1. On the master server, check replication by running:

p4 login < /p4/common/bin/adminpass

p4 pull -lj

1. Verify that you see two matching journal sequence numbers. The p4 pull –lj will only succeed when run on a replica pulling from a master. This proves that the roles have been switched.
2. On the new replica server, run the sync\_shared\_replica.sh command that cron normally runs:

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

1. *If the original master server has been down for more than 7 days*, you must 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 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  
/p4/1/checkpoints/p4\_1.ckp.10.gz

Optionally, to make sure the depot files are up to date, 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

NOTE: 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 3.4 above

# DR Failover Windows

The DR server is being replicated using the Perforce built in server-to-server replication via the “**p4 pull**” command and the replication of checkpoints. This document provides the steps to failover from the main server to the DR server.



We want to go from the above situation to:



## Procedure: Introduction

This section provides the steps to fail over from the main server to the disaster recovery (DR) server. This is a generic document that needs to be customized for the specific set up in your environment.

The following values are defined in sdp\_config.ini within the <metadata> drive.

* <metadata> - default value is E:
* <depotdata> - default value is F:
* <logs> - default value is G:

## Check replica status

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

Do these steps if the master is still functioning. (If the master is not running and not able to run, assume that the replica is up-to-date.)

1. On the replica machine, run p4 pull -l to check the status of the replication.

2. Wait for the replica to fully catch up before shutting down the master and replica instances.

3. You can also verify that the replica is up to date by checking the values on the servers. For instance, run p4 counters on both the master and the replica to make sure they match.

Once you have verified the replication is up to date:

1. Stop the replica by running net stop p4\_1 on the replica server
2. Stop the master by running net stop p4\_1 on the master server

## Change the replica to become the master

Change the DNS entry on your network or move the VIP from the master to the new master (previously the replica) if you are using a VIP address.

On the replica server:

* + - 1. Edit <metadata>:\p4\1\root\server.id to change the name to Master and save the file. Note that the file is read-only and needs to be saved as read-only.
      2. Restart the server on the replica machine as the master by running net start p4\_1
      3. Run:

del <metadata>:\p4\1\root\state

del <metadata>:\p4\1\root\rdb.lbr

## Update the scheduled jobs

On the new master server (originally the replica), run:

at /delete

at 03:00 /every:M,T,W,Th,F,Su <depotdata>:\p4\1\bin\daily\_backup.bat

at 03:00 /every:S <depotdata>:\p4\1\bin\weekly\_backup.bat

at 06:00 /every:S <depotdata>:\p4\1\bin\p4verify.bat

On the original master server, now the replica, use task scheduler, and schedule:

Every M,T,W,Th,F,Su <depotdata>:\p4\1\bin\sync\_replica.bat to run at 06:00am

Every S <depotdata>:\p4\1\bin\weekly\_sync\_replica.bat to run at 06:00am

Both tasks must be scheduled to run as a local account on the machine or network rather than the SYSTEM account because they need to be able to map a drive to the (new) master server.

## Change the original master to become the replica

On the original master (the new replica),

1. Edit <metadata>\p4\1\root\server.id and change it to replica1 or whatever the value in your original installation was.
2. Delete the following files:

<logs>:\p4\1\logs\journal

<metadata>:\p4\1\root\state (if it exists)

<metadata>:\p4\1\root\rdb.lbr” (if it exists)

On the new replica (old master) machine,

1. Log in into the (new) master server as the service user on the (new) replica so that replication can run:

p4 -p master\_server\_address:1667 -u service login

1. Enter the service user’s password. (Same as p4admin password.)
2. Start the server as a replica by running

net start p4\_1

1. Check replication by running:

p4 pull –l

1. Run:

<depotdata>:\p4\1\bin\sync\_replica.bat

The command that task scheduler normally runs to make sure it is working properly.

NOTE: If the 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 sdp\_config.ini 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:

net use z: [\\master\_server\<depotdata\_drive>$](file:///\\master_server\%3cdepotdata_drive%3e$)

attrib –r <depotdata>:\p4\1\checkpoints\\*.\*

del <depotdata>:\p4\1\checkpoints\\*.\*

xcopy /S z:\p4\1\checkpoints <depotdata>:\p4\1\checkpoints

xcopy /S z:\p4\1\depots <depotdata>:\p4\1\depots

del <metadata>:\p4\1\root\db.\*

del <metadata>:\p4\1\offline\_db\db.\*

del <logs>:\p4\1\logs\journal

dir <depotdata>:\p4\1\checkpoints

We need to use the highest numbered p4\_1.ckp.\*.gz file in that directory. Assume it is 10 for this example:

<depotdata>:\p4\1\bin\p4d -r <metadata>:\p4\1\root -jr -z <depotdata>:\p4\1\checkpoints\p4\_1.ckp.10.gz

<depotdata>:\p4\1\bin\p4 -p master\_server:1666 -u service login

net start p4\_1

<depotdata>:\p4\1\bin\p4d -r <metadata>:\p4\1\offline\_db -jr -z <metadata>:\p4\1\checkpoints\p4\_1.ckp.10.gz