

# PostgreSQL Point-in-Time Recovery Procedure

Kiriakos Georgiou

February 27, 2015

This is a summary of the [PostgreSQL Recovering Using a Continuous Archive Backup](#) instructions.

## 1 Stop the server

Stop the server, if it's running.

## 2 Backup the data directory

Space permitting, save the contents of the data directory, and make a new data directory with the same ownership and premissions as the old one. For example:

```
cd /pg_data/test/current
mv data data_old
mkdir -m 700 data
```

If there isn't enough space, you should save the contents of the cluster's `pg_xlog` subdirectory (it might contain logs which were not archived before the system went down) before creating a new blank data directory, as shown above.

## 3 Restore cluster data directory

Be sure it is restored with the right ownership (the postgres user) and permissions. For example:

```
cd /pg_data/test/current/data
tar zpsxvf /pg_backup/test/current/data_files/20131024-153148/base.tar.gz
```

## 4 Confirm pg\_xlog is empty

The `pgbackup.pl` backup utility is smart and does not backup files in the `pg_xlog` directory, but if any files are there after restoring from the tar file, they should be deleted since these came from the backup and are therefore obsolete.

## 5 Copy un-archived wal files

Copy the `archive_status` directory and un-archived wal files from the old `data_old/pg_xlog` to the new `data/pg_xlog` directory. Run the `copy_unarchived.rb` utility to determine the copy command that should be executed.

## 6 Setup recovery

Create a recovery command file `recovery.conf` in the cluster data directory (the same directory that has `postgresql.conf`). See the [PostgreSQL Recovery Configuration](#) for details. The `recovery.conf` file should have something like:

```
# shell command that is executed to copy log files back from archival storage
#
restore_command = 'cp /pg_backup/test/current/archive_logs/%f "%p"'

# Recover up to some transaction, eg: 1815
# You would want this if for example you had a filesystem failure, and
# in the postgres server log, the first transaction that failed at the
# time of the filesystem failure was 1815
#
recovery_target_xid = '1815'

# up to, but not including the above (failed) transaction
#
recovery_target_inclusive = false
```

## 7 Start the server

The server will go into recovery mode and proceed to read through the archived WAL files it needs. Upon completion of the recovery process, the server will rename `recovery.conf` to `recovery.done` (to prevent accidentally re-entering recovery mode later) and then commence normal database operations.