



Linux Ubuntu 26.04 Manual Pages on command 'pg_combinebackup.1'

\$ man pg_combinebackup.1

PG_COMBINEBACKUP(1) PostgreSQL 18.4 Documentation PG_COMBINEBACKUP(1)

NAME

pg_combinebackup - reconstruct a full backup from an incremental backup and dependent backups

SYNOPSIS

pg_combinebackup [option...] [backup_directory...]

DESCRIPTION

pg_combinebackup is used to reconstruct a synthetic full backup from an incremental backup and the earlier backups upon which it depends.

Specify all of the required backups on the command line from oldest to newest. That is, the first backup directory should be the path to the full backup, and the last should be the path to the final incremental backup that you wish to restore. The reconstructed backup will be written to the output directory specified by the -o option.

pg_combinebackup will attempt to verify that the backups you specify form a legal backup chain from which a correct full backup can be reconstructed. However, it is not designed to help you keep track of which backups depend on which other backups. If you remove one or more of the previous backups upon which your incremental backup relies, you will not be able to restore it. Moreover, pg_combinebackup only attempts to verify that the backups have the correct relationship to each other, not that each individual backup is intact; for that, use pg_verifybackup(1).

Since the output of pg_combinebackup is a synthetic full backup, it can be used as an input to a future invocation of pg_combinebackup. The synthetic full backup would be specified on the command line in lieu of the chain of backups from which it was reconstructed.

OPTIONS

-d

--debug

Print lots of debug logging output on stderr.

-k

--link

Use hard links instead of copying files to the synthetic backup. Reconstruction of the synthetic backup might be faster (no file copying) and use less disk space, but care must be taken when using the output directory, because any modifications to that directory (for example, starting the server) can also affect the input directories.

Likewise, changes to the input directories (for example, starting the server on the full backup) could affect the output directory. Thus, this option is best used when the input directories are only copies that will be removed after `pg_combinebackup` has completed.

Requires that the input backups and the output directory are in the same file system.

If a backup manifest is not available or does not contain checksum of the right type, hard links will still be created, but the file will be also read block-by-block for the checksum calculation.

-n

--dry-run

The `-n/--dry-run` option instructs `pg_combinebackup` to figure out what would be done without actually creating the target directory or any output files. It is particularly useful in combination with `--debug`.

-N

--no-sync

By default, `pg_combinebackup` will wait for all files to be written safely to disk. This option causes `pg_combinebackup` to return without waiting, which is faster, but means that a subsequent operating system crash can leave the output backup corrupt. Generally, this option is useful for testing but should not be used when creating a production installation.

-o outputdir

--output=outputdir

Specifies the output directory to which the synthetic full backup should be written.

Currently, this argument is required.

`-T olddir=newdir`

`--tablespace-mapping=olddir=newdir`

Relocates the tablespace in directory `olddir` to `newdir` during the backup. `olddir` is the absolute path of the tablespace as it exists in the final backup specified on the command line, and `newdir` is the absolute path to use for the tablespace in the reconstructed backup. If either path needs to contain an equal sign (=), precede that with a backslash. This option can be specified multiple times for multiple tablespaces.

`--clone`

Use efficient file cloning (also known as ?reflinks? on some systems) instead of copying files to the new data directory, which can result in near-instantaneous copying of the data files.

If a backup manifest is not available or does not contain checksum of the right type, file cloning will be used to copy the file, but the file will be also read block-by-block for the checksum calculation.

File cloning is only supported on some operating systems and file systems. If it is selected but not supported, the `pg_combinebackup` run will error. At present, it is supported on Linux (kernel 4.5 or later) with Btrfs and XFS (on file systems created with reflink support), and on macOS with APFS.

`--copy`

Perform regular file copy. This is the default. (See also `--copy-file-range`, `--clone`, and `-k/--link`.)

`--copy-file-range`

Use the `copy_file_range` system call for efficient copying. On some file systems this gives results similar to `--clone`, sharing physical disk blocks, while on others it may still copy blocks, but do so via an optimized path. At present, it is supported on Linux and FreeBSD.

If a backup manifest is not available or does not contain checksum of the right type, `copy_file_range` will be used to copy the file, but the file will be also read block-by-block for the checksum calculation.

`--manifest-checksums=algorithm`

Like `pg_basebackup(1)`, `pg_combinebackup` writes a backup manifest in the output directory. This option specifies the checksum algorithm that should be applied to each file included in the backup manifest. Currently, the available algorithms are NONE,

CRC32C, SHA224, SHA256, SHA384, and SHA512. The default is CRC32C.

`--no-manifest`

Disables generation of a backup manifest. If this option is not specified, a backup manifest for the reconstructed backup will be written to the output directory.

`--sync-method=method`

When set to `fsync`, which is the default, `pg_combinebackup` will recursively open and synchronize all files in the backup directory. When the plain format is used, the search for files will follow symbolic links for the WAL directory and each configured tablespace.

On Linux, `syncfs` may be used instead to ask the operating system to synchronize the whole file system that contains the backup directory. When the plain format is used, `pg_combinebackup` will also synchronize the file systems that contain the WAL files and each tablespace. See `recovery_init_sync_method` for information about the caveats to be aware of when using `syncfs`.

This option has no effect when `--no-sync` is used.

`-V`

`--version`

Prints the `pg_combinebackup` version and exits.

`-?`

`--help`

Shows help about `pg_combinebackup` command line arguments, and exits.

LIMITATIONS

`pg_combinebackup` does not recompute page checksums when writing the output directory.

Therefore, if any of the backups used for reconstruction were taken with checksums disabled, but the final backup was taken with checksums enabled, the resulting directory may contain pages with invalid checksums.

To avoid this problem, taking a new full backup after changing the checksum state of the cluster using `pg_checksums(1)` is recommended. Otherwise, you can disable and then optionally reenable checksums on the directory produced by `pg_combinebackup` in order to correct the problem.

ENVIRONMENT

This utility, like most other PostgreSQL utilities, uses the environment variables supported by `libpq` (see Section 32.15).

The environment variable `PG_COLOR` specifies whether to use color in diagnostic messages.

Possible values are `always`, `auto` and `never`.

SEE ALSO

`pg_basebackup(1)`

PostgreSQL 18.4

2026

`PG_COMBINEBACKUP(1)`