

**NAME**

`pg_archivecleanup` - clean up PostgreSQL WAL archive files

**SYNOPSIS**

**`pg_archivecleanup`** [*option...*] *archivelocation oldestkeptwalfile*

**DESCRIPTION**

`pg_archivecleanup` is designed to be used as an `archive_cleanup_command` to clean up WAL file archives when running as a standby server (see Section 27.2). `pg_archivecleanup` can also be used as a standalone program to clean WAL file archives.

To configure a standby server to use `pg_archivecleanup`, put this into its `postgresql.conf` configuration file:

```
archive_cleanup_command = 'pg_archivecleanup archivelocation %r'
```

where *archivelocation* is the directory from which WAL segment files should be removed.

When used within `archive_cleanup_command`, all WAL files logically preceding the value of the `%r` argument will be removed from *archivelocation*. This minimizes the number of files that need to be retained, while preserving crash-restart capability. Use of this parameter is appropriate if the *archivelocation* is a transient staging area for this particular standby server, but *not* when the *archivelocation* is intended as a long-term WAL archive area, or when multiple standby servers are recovering from the same archive location.

When used as a standalone program all WAL files logically preceding the *oldestkeptwalfile* will be removed from *archivelocation*. In this mode, if you specify a `.partial` or `.backup` file name, then only the file prefix will be used as the *oldestkeptwalfile*. This treatment of `.backup` file name allows you to remove all WAL files archived prior to a specific base backup without error. For example, the following example will remove all files older than WAL file name `000000010000003700000010`:

```
pg_archivecleanup -d archive 000000010000003700000010.00000020.backup
```

```
pg_archivecleanup: keep WAL file "archive/000000010000003700000010" and later
```

```
pg_archivecleanup: removing file "archive/00000001000000370000000F"
```

```
pg_archivecleanup: removing file "archive/00000001000000370000000E"
```

`pg_archivecleanup` assumes that *archivelocation* is a directory readable and writable by the server-owning user.

## OPTIONS

`pg_archivecleanup` accepts the following command-line arguments:

**-d**

Print lots of debug logging output on stderr.

**-n**

Print the names of the files that would have been removed on stdout (performs a dry run).

**-V****--version**

Print the `pg_archivecleanup` version and exit.

**-x** *extension*

Provide an extension that will be stripped from all file names before deciding if they should be deleted. This is typically useful for cleaning up archives that have been compressed during storage, and therefore have had an extension added by the compression program. For example: `-x .gz`.

**-?****--help**

Show help about `pg_archivecleanup` command line arguments, and exit.

## ENVIRONMENT

The environment variable **PG\_COLOR** specifies whether to use color in diagnostic messages. Possible values are always, auto and never.

## NOTES

`pg_archivecleanup` is designed to work with PostgreSQL 8.0 and later when used as a standalone utility, or with PostgreSQL 9.0 and later when used as an archive cleanup command.

`pg_archivecleanup` is written in C and has an easy-to-modify source code, with specifically designated sections to modify for your own needs

## EXAMPLES

On Linux or Unix systems, you might use:

```
archive_cleanup_command = 'pg_archivecleanup -d /mnt/standby/archive %r 2>>cleanup.log'
```

where the archive directory is physically located on the standby server, so that the *archive\_command* is

accessing it across NFS, but the files are local to the standby. This will:

⊕

debugging output in cleanup.log

⊕

no-longer-needed files from the archive directory