

Implementing an Extension Search Path

Peter Eisentraut

2025-03-26

peter@eisentraut.org
<https://peter.eisentraut.org/>
[@petereisentraut@mastodon.social](https://mstdn.social/@petereisentraut)

<https://www.enterprisedb.com/>

What do I know about extensions?

- my extensions: plsh, pguint, pgpcr, pguri, plxslt, pgemailaddr, pgtrashcan
- contributed to: orafce, pglogical, bdr, pg_failover_slots
- also pex, autopex

Problems I was asked to solve

1. extensions in Kubernetes (CloudNativePG) environment
2. extensions with Postgres.app
3. run tests during (Debian) package builds

Problems I wanted to solve

4. PGXS make check
5. must install locally built extensions into OS-controlled directories
6. poor support in macOS Homebrew

These are all the same problem!

The extension files location is hardcoded to be inside the PostgreSQL installation tree.

`/usr/share/postgresql/extension/`

(or similar)

Some history

- In 2001, I implemented `dynamic_library_path`.
- In 2011, extensions were implemented, and they broke this.
- I am now fixing this.

Recent history

- Apr. 2024: David Wheeler starts hackers thread, based on Christoph Berg's Debian patch.
- Sep. 2024: Gabriele Bartolini and I start discussing.
- Oct. 2024: PGConf.EU, requirements gathering
- Nov. 2024: I post first patch.
- Mar. 2025: committed :-)

Credits

commit 4f7f7b03758

Author: Peter Eisentraut <peter@eisentraut.org>

Date: Wed Mar 19 06:57:20 2025

extension_control_path

Author: Peter Eisentraut <peter@eisentraut.org>

Co-authored-by: Matheus Alcantara <matheusssilv97@gmail.com>

Reviewed-by: David E. Wheeler <david@justatheory.com>

Reviewed-by: Gabriele Bartolini <gabriele.bartolini@enterprisedb.com>

Reviewed-by: Marco Nenciarini <marco.nenciarini@enterprisedb.com>

Reviewed-by: Niccolò Fei <niccolo.fei@enterprisedb.com>

Stats

commit 4f7f7b03758

68	0	doc/src/sgml/config.sgml
14	5	doc/src/sgml/extend.sgml
4	2	doc/src/sgml/ref/create_extension.sgml
12	7	src/Makefile.global.in
261	142	src/backend/commands/extension.c
48	29	src/backend/utils/fmgr/dfmgr.c
13	0	src/backend/utils/misc/guc_tables.c
1	0	src/backend/utils/misc/postgresql.conf.sample
2	0	src/include/commands/extension.h
3	0	src/include/fmgr.h
1	0	src/test/modules/test_extensions/Makefile
5	0	src/test/modules/test_extensions/meson.build
80	0	src/test/modules/test_extensions/t/001_extension_control_path.pl

How to use

Let's say your PostgreSQL is in

`/usr/bin/, /usr/lib/postgresql/, /usr/share/postgresql/, etc.`

You can install your extension into

`/usr/local/mystuff/lib/, /usr/local/mystuff/share/`

`(make install prefix=/usr/local/mystuff)`

And set `(postgresql.conf)`

```
dynamic_library_path = $libdir:/usr/local/mystuff/lib/  
extension_control_path = $system:/usr/local/mystuff/share/extension/
```

This solves problem #5.

How to use

Use in Debian packaging:

```
PKGARGS="--pgoption extension_control_path=$PWD/debian/$PACKAGE/usr/share/postgresql/$v/extension:\$system  
--pgoption dynamic_library_path=$PWD/debian/$PACKAGE/usr/lib/postgresql/$v/lib:/usr/lib/postgresql/$v/lib"
```

This solves problem #3.

Why two path settings?

In general, both of these could be in different trees.

```
extension_control_path  
dynamic_library_path
```

is like

```
-I /usr/local/include  
-L /usr/local/lib
```

Easiest to implement and complete in 6 months.

Solution integration progress

1. CloudNativePG: see May 7 meeting
2. Postgres.app: tbd.
3. Debian: done / see April 9 meeting
4. make check: patch pending
5. local install: done
6. Homebrew: later (probably closer to PG18 release)

Other thoughts about extensions

- test building all extensions against PostgreSQL built with Meson
- build extensions with Meson
- postgresql-extension.pc (pkg-config)

Conclusion

- `extension_control_path` coming in PG18
 - (Fixed a 14-year old misfeature.)
- Help: test and validate
- Help: integrate into downstream users