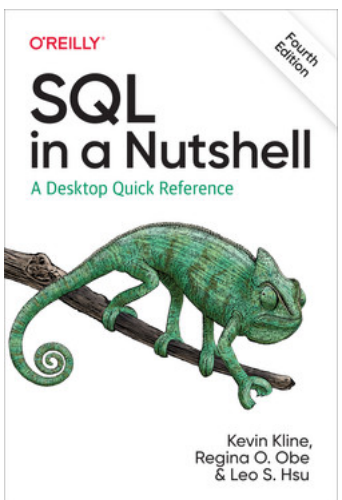
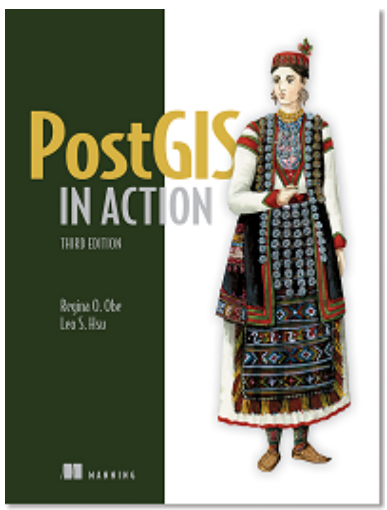
EXTENSION UPGRADE PAIN POINTS

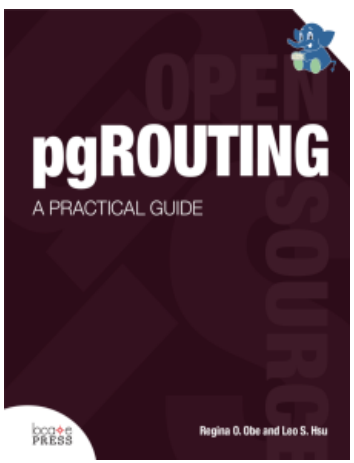
REGINA OBE

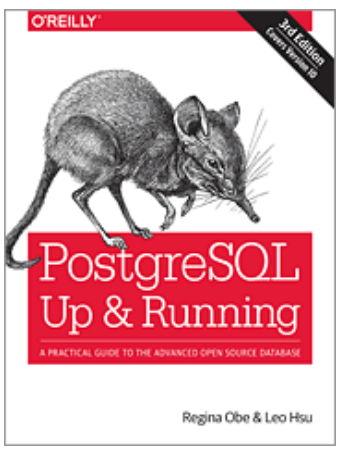
Buy our books! https://postgis.us/page_buy_book

LATEST BOOKS









Mastodon: https://mapstodon.space/@robe

Books Coming

pgRouting (a practical guide) 2nd Ed (Locate Press) in Progress.

Subscribe at locatepress.com/subscribe

Extension Upgrades

- Supporting Multiple versions of PostgreSQL on an extension version
- Managing upgrade scripts
- Versioning of your lib file
- What extension support is missing

USERS NIGHTMARE: PART 1

I had PostgreSQL 16 installed, and then upgraded to PostgreSQL 17. Why does the extension I compiled/installed for 16 not work on 17. It's the same server!

- `apt install postgresql-17-postgis-3 postgresql-17-pgrouting`
- `yum install pgrouting_17 postgis35_17`

USERS NIGHTMARE: PART 2

pg_upgrade refuses to upgrade me from PostgreSQL 16 to PostgreSQL 17, says my libpgrouting-3.6 is missing.

Note to extension authors, pg_upgrade doesn't care what new version the user has installed on their new PostgreSQL, it wants the version that was present in the old server. By that I mean the library must be named the same AND all the functions exposed via the SQL API, better be present in that library. **BUT** it better be compiled against the new version of PostgreSQL!

YUM: USERS HAVE MANY CHOICES!

yum.postgresql.org

```
yum search postgis
```

It is possible for multiple versions of an extension to support multiple versions of PostgreSQL. It's up to the packager to allow this. But, installing one might break another so proceed with caution.

```
postgis30_13.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis31_13.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis31_14.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis32_13.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis32_14.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis32_15.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis33_13.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis33_14.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis33_15.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis33_16.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis33_17.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis34_13.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis34_14.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis34_15.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis34_16.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis34_17.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis35_13.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis35_14.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis35_15.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis35_16.x86_64 : Geographic Information Systems Extensions to PostgreSQL
postgis35_17.x86_64 : Geographic Information Systems Extensions to PostgreSQL
```

APT: USERS HAVE FEWER CHOICES

`apt search postgis`

apt.postgresql.org, generally only allows one version of an extension per version of PostgreSQL.

```
postgresql-16-postgis-3/now 3.4.2+dfsg-1 amd64 [installed,local]
  Geographic objects support for PostgreSQL 16

postgresql-16-postgis-3-scripts/now 3.4.2+dfsg-1 all [installed,local]
  Geographic objects support for PostgreSQL 16 -- SQL scripts

postgresql-17-postgis-3/testing 3.5.2+dfsg-1 amd64
  Geographic objects support for PostgreSQL 17

postgresql-17-postgis-3-scripts/testing 3.5.2+dfsg-1 all
  Geographic objects support for PostgreSQL 17 -- SQL scripts

postgresql-postgis/testing 3.5.2+dfsg-1 amd64
  Geographic objects support for PostgreSQL -- Metapackage

postgresql-postgis-scripts/testing 3.5.2+dfsg-1 all
  Geographic objects support for PostgreSQL -- SQL scripts metapackage
```

EXTENSION FOLDER HAS A LOT OF SCRIPT FILES

- PostGIS 3.5.2 for **postgis**, **postgis_raster**, **postgis_topology**, **postgis_sfcgal**, **postgis_tiger_geocoder**, **address_standardizer** has 114*6 = 684 files.
- Regardless if you chain your scripts from version to version, the way extension is managed in PostgreSQL you will need to have a version bump file for each version you have ever released for pg_extension table to register correctly. At the very least any point at which you need to make an sql api change.

HOW POSTGIS MANAGES UPGRADE SCRIPTS

Versions that need to be generated managed by upgradeable_versions.mk

```
UPGRADEABLE_VERSIONS = \
  2.0.0 \
  2.0.1 \
  2.0.2 \
  2.0.3 \
  ..
```

WHAT'S IN THESE SCRIPTS?

In the past these were symlinks (but symlinks don't work well on windows and possibly some other systems). So we decided to switch to essentially 0-byte files. If you look at the list most have nothing in them except for the ANY--3.5.2 and --3.5.2 where 3.5.2 is the version we are releasing.

```
:
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.2.4--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.2.5--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.2.6--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.2.7--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.3.0--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.3.1--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.3.2--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.3.3--ANY.sql
:
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.4.0--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.4.1--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.4.2--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.4.3--ANY.sql
:
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.5.0--ANY.sql
-rw-r--r-- 1 lr lr  112 Mar 22 20:20 postgis--3.5.1--ANY.sql

-rw-r--r-- 1 lr lr 7.2M Mar 22 20:19 postgis--3.5.2.sql
-rw-r--r-- 1 lr lr 7.5M Mar 22 20:19 postgis--ANY--3.5.2.sql
```

WE HAVE OUR OWN UPGRADE FUNCTION

Handles upgrading of all packaged postgis extensions. Also allows upgrade from same version to same version. ALTER EXTENSION assumes if source and target version are the same, nothing needs to be done.

```
SELECT postgis_extensions_upgrade();
SELECT postgis_full_version();
```

SHOULD YOU VERSION YOUR LIB FILES?

- Prior to PostGIS 3.0, our lib files were minor versioned postgis-2.4.{so,dll,whatever}, postgis-2.5.{so,dll,whatever}, now by default we major version
- pgrouting lib files are still minor versioned e.g. pgrouting-3.6, pgrouting-3.5 and so on. This may change in upcoming pgRouting 4

PROS OF NOT VERSIONING LIB FILES

PostgreSQL in contrib of source code are not versioned at all e.g. hstore.so, pg_trgm.so and so on.

- Plays nicely with pg_upgrade

CONS OF NOT VERSIONING LIB FILES

- Every single function that you have ever exposed via SQL API must be present, or pg_upgrade will scream.
- No way to run two versions of an extension in the same PostgreSQL install.
- If ever you need to make key data structural changes, not clear to user they can't use pg_upgrade

PROS OF VERSIONING LIB FILES

- Can run multiple versions in the same PostgreSQL cluster
- Less need to worry about ripping out lib functions used in SQL api.

CONS OF VERSIONING LIB FILES

- Does not play nicely with pg_upgrade, cause old install references old named library.
- Every single function that references the extension lib, needs to be replaced when upgrading.

POSTGIS: WHAT'S IN THOSE SCRIPTS?

- Basic set of sql scripts glued together to form CREATE EXTENSION script.
- Comments are code: comments in a script are rules to our perl post-processor.

WORKING AROUND USERS DATA

ine in postgis.sql.in

```
-- Availability: 2.0.0
-- Changed: 3.1.0 to add gridSize default argument
-- Replaces ST_UnaryUnion(geometry) deprecated in
CREATE OR REPLACE FUNCTION ST_UnaryUnion(geometry,
 gridSize float8 DEFAULT -1.0)
 RETURNS geometry
 AS 'MODULE_PATHNAME','ST_UnaryUnion'
 LANGUAGE 'c' IMMUTABLE STRICT PARALLEL SAFE
  _COST_HIGH;
```

Generates lines in ANY--postgis-3.5.2

```
-- Rename st_unaryunion ( geometry )
-- deprecated in PostGIS 301, if needed
DO LANGUAGE 'plpgsql'
$postgis_proc_upgrade$
DECLARE
    detail TEXT;
    argnames TEXT[];
BEGIN

-- Check if the deprecated function exists
-- Rename the replaced function, to avoid ambigui
-- The renamed function will eventually be droppe

END;
$postgis_proc_upgrade$;
-- definition of new function follows
```

KEEP ALIVE THE DEAD SO PG_UPGRADE IS HAPPY

postgis_legacy.c

```
/** throws an error if these are called, but pg_upgrade can load it**/
POSTGIS_DEPRECATE("2.5.0", pgis_abs_in)
POSTGIS_DEPRECATE("3.5.0", check_authorization)
```

POSTGIS: HANDLING CASES NOT ADDRESSED BY PERL PROCESSOR

postgis_before_upgrade.sql, postgis_after_upgrade.sql

ACCOUNTING FOR DIFFERENT VERSIONS OF POSTGRESQL AT COMPILE TIME

- Just have one version of extension for each version of PostgreSQL, e.g PostgreSQL contrib extensions do that
- Read PostgreSQL version from pg_config, and have IF defs that utilize it.

```
#if POSTGIS_PGSQL_VERSION >= 120
/** do stuff for newer versions **/
#end if
```

ACCOUNTING FOR DIFFERENT VERSIONS OF POSTGRESQL PG_UPGRADE

If your extension behavior changes when version of PostgreSQL changes, make sure you've got a sql function that tells you what version was running when it was installed.

```
CREATE OR REPLACE FUNCTION _postgis_scripts_pgsql_version() RETURNS text
  AS _POSTGIS_SQL_SELECT_POSTGIS_PGSQL_VERSION
  LANGUAGE 'sql' IMMUTABLE;
```

PGROUTING APPROACH TO UPGRADE

- **build-extension-file.pl**: Chaining SQL files to form a CREATE EXTENSION file
- **build-extension-upgrade-files.pl**: Perl post-processor: explicit drops are in here, not in comments

WHEN YOUR EXTENSION NEEDS TO REFERENCE OTHER EXTENSIONS

How do you schema qualify your references to other extensions if you don't know what schema they are installed in?

New feature in PostgreSQL 16, introduced by me

- .control file: add `no_relocate = 'postgis,pgrouting'` to prevent relocation of a dependent extension. These should be a subset of your requires.
- .control file: use existing `requires = 'postgis,pgrouting'` and will check for schema qualification of required extensions
- Schema qualify in your scripts using syntax `@extschema:reqextname@`. e.g. `@extschema:postgis@.ST_Intersects(..)`

WHAT CAN IMPROVE IN EXTENSION SUPPORT

POSSIBILITY OF HAVING ONE UPGRADE SCRIPT FOR EACH EXTENSION

Discussion is here and draft patch - Support % wildcard in extension upgrade filenames

Many extensions that don't follow a linear model have same issue as PostGIS.

BETTER SUPPORT FOR DDL CHANGES

- VIEW dependency option to autofix. E.g. you can't drop a function if a VIEW depends on it without resorting to dropping the view. Can't change the type of a column if a view depends on it.
- ALTER DOMAIN - has no support for changing underlying datatype, e.g. int4 to int8, have to resort to updating system catalogs.
- IF NOT EXISTS is missing in a lot of object types, e.g. none for CREATE TYPE, CREATE DOMAIN, CREATE OPERATOR

LISTING THE TRUE EXTENSION VERSION

Many extensions have their own extension version function, that lists the true version of the library or sql api installed. Would be nice if perhaps as part of the control file, extensions can specify what this function is and PostgreSQL can have a function like pg_extension_true_version('postgis') that calls this.

e.g PostGIS has `postgis_full_version()` and `postgis_version()`, pgRouting has a `pgr_version()`, `pgr_full_version()`

It's hard to remember all these function names when you need to inspect what you are really running.

CROSS SHARING OF FUNCTIONS AT THE LIB LEVEL

Easier way for other extensions to call lib functions in another extension lib without resorting to go thru the SQL API, which can be very slow. E.g. MobilityDb extension embeds a bit of PostGIS to get around this issue.

PG_UPGRADE OPTION TO USE CREATE EXTENSION

Instead of loading the functions from the current DB, would be nice, if pg_upgrade had a switch to allow CREATE EXTENSION, that way errors like pgrouting-3.6 can't be found will be a thing of the past. Also wouldn't need to run ALTER EXTENSION ... UPDATE for every single extension you have after upgrade.

FIN

BUY OUR BOOKS HTTPS://POSTGIS.US