



E.4. Release 16

[E.4.1. Overview](#)[E.4.2. Migration to Version 16](#)[E.4.3. Changes](#)[E.4.4. Acknowledgments](#)

Release date: 2023-09-14

E.4.1. Overview

PostgreSQL 16 contains many new features and enhancements, including:

- Allow parallelization of FULL and internal right OUTER hash joins
- Allow logical replication from standby servers
- Allow logical replication subscribers to apply large transactions in parallel
- Allow monitoring of I/O statistics using the new `pg_stat_io` view
- Add SQL/JSON constructors and identity functions
- Improve performance of vacuum freezing
- Add support for regular expression matching of user and database names in `pg_hba.conf`, and user names in `pg_ident.conf`

The above items and other new features of PostgreSQL 16 are explained in more detail in the sections below.

E.4.2. Migration to Version 16

A dump/restore using `pg_dumpall` or use of `pg_upgrade` or logical replication is required for those wishing to migrate data from any previous release. See [Section 19.6](#) for general information on migrating to new major releases.

Version 16 contains a number of changes that may affect compatibility with previous releases. Observe the following incompatibilities:

- Change assignment rules for [PL/pgSQL](#) bound cursor variables (Tom Lane)

Previously, the string value of such variables was set to match the variable name during cursor assignment; now it will be assigned during `OPEN`, and will not match the variable name. To restore the previous behavior, assign the desired portal name to the cursor variable before `OPEN`.

- Disallow `NULLS NOT DISTINCT` indexes for primary keys (Daniel Gustafsson)
- Change `REINDEX DATABASE` and `reindexdb` to not process indexes on system catalogs (Simon Riggs)

Processing such indexes is still possible using `REINDEX SYSTEM` and `reindexdb --system`.

- Tighten `GENERATED` expression restrictions on inherited and partitioned tables (Amit Langote, Tom Lane)

Columns of parent/partitioned and child/partition tables must all have the same generation status, though now the actual generation expressions can be different.

- Remove `pg_walinspect` functions `pg_get_wal_records_info_till_end_of_wal()` and `pg_get_wal_stats_till_end_of_wal()` (Bharath Rupireddy)
- Rename server variable `force_parallel_mode` to `debug_parallel_query` (David Rowley)
- Remove the ability to `create views` manually with `ON SELECT` rules (Tom Lane)
- Remove the server variable `vacuum_defer_cleanup_age` (Andres Freund)

This has been unnecessary since `hot_standby_feedback` and `replication slots` were added.

- Remove server variable `promote_trigger_file` (Simon Riggs)

This was used to promote a standby to primary, but is now more easily accomplished with `pg_ctl promote` or `pg_promote()`.

- Remove read-only server variables `lc_collate` and `lc_ctype` (Peter Eisentraut)

Collations and locales can vary between databases so having them as read-only server variables was unhelpful.

- Role inheritance now controls the default inheritance status of member roles added during `GRANT` (Robert Haas)

The role's default inheritance behavior can be overridden with the new `GRANT ... WITH INHERIT` clause. This allows inheritance of some roles and not others because the members' inheritance status is set at `GRANT` time. Previously the inheritance status of member roles was controlled only by the role's inheritance status, and changes to a role's inheritance status affected all previous and future member roles.

- Restrict the privileges of `CREATEROLE` and its ability to modify other roles (Robert Haas)

Previously roles with `CREATEROLE` privileges could change many aspects of any non-superuser role. Such changes, including adding members, now require the role requesting the change to have `ADMIN OPTION` permission. For example, they can now change the `CREATEDB`, `REPLICATION`, and `BYPASSRLS` properties only if they also have those permissions.

- Remove symbolic links for the postmaster binary (Peter Eisentraut)

E.4.3. Changes

Below you will find a detailed account of the changes between PostgreSQL 16 and the previous major release.

E.4.3.1. Server

E.4.3.1.1. Optimizer

- Allow incremental sorts in more cases, including `DISTINCT` (David Rowley)
- Add the ability for aggregates having `ORDER BY` or `DISTINCT` to use pre-sorted data (David Rowley)

The new server variable `enable_presorted_aggregate` can be used to disable this.

- Allow memoize atop a `UNION ALL` (Richard Guo)
- Allow anti-joins to be performed with the non-nullable input as the inner relation (Richard Guo)
- Allow parallelization of `FULL` and internal right `OUTER` hash joins (Melanie Plageman, Thomas Munro)
- Improve the accuracy of `GIN` index access optimizer costs (Ronan Dunklau)

E.4.3.1.2. General Performance

- Allow more efficient addition of heap and index pages (Andres Freund)
 - During non-freeze operations, perform page `freezing` where appropriate (Peter Geoghegan)
- This makes full-table freeze vacuums less necessary.
- Allow window functions to use the faster `ROWS` mode internally when `RANGE` mode is active but unnecessary (David Rowley)
 - Allow optimization of always-increasing window functions `ntile()`, `cume_dist()` and `percent_rank()` (David Rowley)
 - Allow aggregate functions `string_agg()` and `array_agg()` to be parallelized (David Rowley)
 - Improve performance by caching `RANGE` and `LIST` partition lookups (Amit Langote, Hou Zhijie, David Rowley)
 - Allow control of the shared buffer usage by vacuum and analyze (Melanie Plageman)

The `VACUUM/ANALYZE` option is `BUFFER_USAGE_LIMIT`, and the `vacuumdb` option is `--buffer-usage-limit`. The default value is set by server variable `vacuum_buffer_usage_limit`, which also controls autovacuum.

- Support `wal_sync_method=fdatasync` on Windows (Thomas Munro)
- Allow `HOT` updates if only BRIN-indexed columns are updated (Matthias van de Meent, Josef Simanek, Tomas Vondra)
- Improve the speed of updating the `process title` (David Rowley)
- Allow xid/subxid searches and ASCII string detection to use vector operations (Nathan Bossart, John Naylor)

ASCII detection is particularly useful for **COPY FROM**. Vector operations are also used for some C array searches.

- Reduce overhead of memory allocations (Andres Freund, David Rowley)

E.4.3.1.3. Monitoring

- Add system view **pg_stat_io** view to track I/O statistics (Melanie Plageman)
- Record statistics on the last sequential and index scans on tables (Dave Page)

This information appears in **pg_stat_*_tables** and **pg_stat_*_indexes**.

- Record statistics on the occurrence of updated rows moving to new pages (Corey Huinker)

The **pg_stat_*_tables** column is **n_tup_newpage_upd**.

- Add speculative lock information to the **pg_locks** system view (Masahiko Sawada, Noriyoshi Shinoda)

The transaction id is displayed in the **transactionid** column and the speculative insertion token is displayed in the **obj id** column.

- Add the display of prepared statement result types to the **pg_prepared_statements** view (Dagfinn Ilmari Mannsåker)
- Create subscription statistics entries at subscription creation time so **stats_reset** is accurate (Andres Freund)

Previously entries were created only when the first statistics were reported.

- Correct the I/O accounting for temp relation writes shown in **pg_stat_database** (Melanie Plageman)
- Add function **pg_stat_get_backend_subxact()** to report on a session's subtransaction cache (Dilip Kumar)
- Have **pg_stat_get_backend_idset()**, **pg_stat_get_backend_activity()**, and related functions use the unchanging backend id (Nathan Bossart)

Previously the index values might change during the lifetime of the session.

- Report stand-alone backends with a special backend type (Melanie Plageman)
- Add wait event **SpinDelay** to report spinlock sleep delays (Andres Freund)
- Create new wait event **DSMAllocate** to indicate waiting for dynamic shared memory allocation (Thomas Munro)

Previously this type of wait was reported as **DSMFillZeroWrite**, which was also used by **mmap()** allocations.

- Add the database name to the **process title** of logical WAL senders (Tatsuhiko Nakamori)

Physical WAL senders do not display a database name.

- Add checkpoint and REDO LSN information to **log_checkpoints** messages (Bharath Rupireddy, Kyotaro Horiguchi)
- Provide additional details during client certificate failures (Jacob Champion)

E.4.3.1.4. Privileges

- Add predefined role **pg_create_subscription** with permission to create subscriptions (Robert Haas)
- Allow subscriptions to not require passwords (Robert Haas)

This is accomplished with the option **password_required=false**.

- Simplify permissions for **LOCK TABLE** (Jeff Davis)

Previously a user's ability to perform **LOCK TABLE** at various lock levels was limited to the lock levels required by the commands they had permission to execute on the table. For example, someone with **UPDATE** permission could perform all lock levels except **ACCESS SHARE**, even though it was a lesser lock level. Now users can issue lesser lock levels if they already have permission for greater lock levels.

- Allow **ALTER GROUP group_name ADD USER user_name** to be performed with **ADMIN OPTION** (Robert Haas)

Previously **CREATEROLE** permission was required.

- Allow **GRANT** to use **WITH ADMIN TRUE/FALSE** syntax (Robert Haas)

Previously only the **WITH ADMIN OPTION** syntax was supported.

- Allow roles that create other roles to automatically inherit the new role's rights or the ability to **SET ROLE** to the new role (Robert Haas, Shi Yu)

This is controlled by server variable `creatorole_self_grant`.

- Prevent users from changing the default privileges of non-inherited roles (Robert Haas)

This is now only allowed for inherited roles.

- When granting role membership, require the granted-by role to be a role that has appropriate permissions (Robert Haas)

This is a requirement even when a non-bootstrap superuser is granting role membership.

- Allow non-superusers to grant permissions using a granted-by user that is not the current user (Robert Haas)

The current user still must have sufficient permissions given by the specified granted-by user.

- Add `GRANT` to control permission to use `SET ROLE` (Robert Haas)

This is controlled by a new `GRANT . . . SET` option.

- Add dependency tracking to roles which have granted privileges (Robert Haas)

For example, removing `ADMIN OPTION` will fail if there are privileges using that option; `CASCADE` must be used to revoke dependent permissions.

- Add dependency tracking of grantors for `GRANT` records (Robert Haas)

This guarantees that `pg_auth_members.grantor` values are always valid.

- Allow multiple role membership records (Robert Haas)

Previously a new membership grant would remove a previous matching membership grant, even if other aspects of the grant did not match.

- Prevent removal of superuser privileges for the bootstrap user (Robert Haas)

Restoring such users could lead to errors.

- Allow `makeaclitem()` to accept multiple privilege names (Robins Tharakan)

Previously only a single privilege name, like `SELECT`, was accepted.

E.4.3.1.5. Server Configuration

- Add support for Kerberos credential delegation (Stephen Frost)

This is enabled with server variable `gss_accept_delegation` and libpq connection parameter `gssdelegation`.

- Allow the SCRAM iteration count to be set with server variable `scram_iterations` (Daniel Gustafsson)

- Improve performance of server variable management (Tom Lane)

- Tighten restrictions on which server variables can be reset (Masahiko Sawada)

Previously, while certain variables, like `transaction_isolation`, were not affected by `RESET ALL`, they could be individually reset in inappropriate situations.

- Move various `postgresql.conf` items into new categories (Shinya Kato)

This also affects the categories displayed in the `pg_settings` view.

- Prevent configuration file recursion beyond 10 levels (Julien Rouhaud)

- Allow `autovacuum` to more frequently honor changes to delay settings (Melanie Plageman)

Rather than honor changes only at the start of each relation, honor them at the start of each block.

- Remove restrictions that archive files be durably renamed (Nathan Bossart)

The `archive_command` command is now more likely to be called with already-archived files after a crash.

- Prevent `archive_library` and `archive_command` from being set at the same time (Nathan Bossart)

Previously `archive_library` would override `archive_command`.

- Allow the postmaster to terminate children with an abort signal (Tom Lane)

This allows collection of a core dump for a stuck child process. This is controlled by `send_abort_for_crash` and `send_abort_for_kill`. The postmaster's `-T` switch is now the same as setting `send_abort_for_crash`.

- Remove the non-functional postmaster `-n` option (Tom Lane)
- Allow the server to reserve backend slots for roles with `pg_use_reserved_connections` membership (Nathan Bossart)

The number of reserved slots is set by server variable `reserved_connections`.

- Allow **huge pages** to work on newer versions of Windows 10 (Thomas Munro)

This adds the special handling required to enable huge pages on newer versions of Windows 10.

- Add `debug_io_direct` setting for developer usage (Thomas Munro, Andres Freund, Bharath Rupireddy)

While primarily for developers, `wal_sync_method=open_sync/open_datasync` has been modified to not use direct I/O with `wal_level=minimal`; this is now enabled with `debug_io_direct=wal`.

- Add function `pg_split_walfile_name()` to report the segment and timeline values of WAL file names (Bharath Rupireddy)

E.4.3.1.6. `pg_hba.conf`

- Add support for regular expression matching on database and role entries in `pg_hba.conf` (Bertrand Drouvot)

Regular expression patterns are prefixed with a slash. Database and role names that begin with slashes need to be double-quoted if referenced in `pg_hba.conf`.

- Improve user-column handling of `pg_ident.conf` to match `pg_hba.conf` (Jelte Fennema)

Specifically, add support for `all`, role membership with `+`, and regular expressions with a leading slash. Any user name that matches these patterns must be double-quoted.

- Allow include files in `pg_hba.conf` and `pg_ident.conf` (Julien Rouhaud)

These are controlled by `include`, `include_if_exists`, and `include_dir`. System views `pg_hba_file_rules` and `pg_ident_file_mappings` now display the file name.

- Allow `pg_hba.conf` tokens to be of unlimited length (Tom Lane)
- Add rule and map numbers to the system view `pg_hba_file_rules` (Julien Rouhaud)

E.4.3.1.7. Localization

- Determine the default encoding from the locale when using ICU (Jeff Davis)

Previously the default was always UTF-8.

- Have `CREATE DATABASE` and `CREATE COLLATION`'s `LOCALE` options, and `initdb` and `createdb` `--locale` options, control non-libc collation providers (Jeff Davis)

Previously they only controlled libc providers.

- Add predefined collations `unicode` and `ucs_basic` (Peter Eisentraut)

This only works if ICU support is enabled.

- Allow custom ICU collation rules to be created (Peter Eisentraut)

This is done using `CREATE COLLATION`'s new `RULES` clause, as well as new options for `CREATE DATABASE`, `createdb`, and `initdb`.

- Allow Windows to import system locales automatically (Juan José Santamaría Flecha)

Previously, only ICU locales could be imported on Windows.

E.4.3.2. Logical Replication

- Allow **logical decoding** on standbys (Bertrand Drouvot, Andres Freund, Amit Khandekar)

Snapshot WAL records are required for logical slot creation but cannot be created on standbys. To avoid delays, the new function `pg_log_standby_snapshot()` allows creation of such records.

- Add server variable to control how logical decoding publishers transfer changes and how subscribers apply them (Shi Yu)

The variable is `debug_logical_replication_streaming`.

- Allow logical replication initial table synchronization to copy rows in binary format (Melih Mutlu)

This is only possible for subscriptions marked as binary.

- Allow parallel application of logical replication (Hou Zhijie, Wang Wei, Amit Kapila)

The `CREATE SUBSCRIPTION` `STREAMING` option now supports `parallel` to enable application of large transactions by parallel workers. The number of parallel workers is controlled by the new server variable `max_parallel_apply_workers_per_subscription`. Wait events `LogicalParallelApplyMain`, `LogicalParallelApplyStateChange`, and `LogicalApplySendData` were also added. Column `leader_pid` was added to system view `pg_stat_subscription` to track parallel activity.

- Improve performance for **logical replication apply** without a primary key (Onder Kalaci, Amit Kapila)

Specifically, `REPLICA IDENTITY FULL` can now use btree indexes rather than sequentially scanning the table to find matches.

- Allow logical replication subscribers to process only changes that have no origin (Vignesh C, Amit Kapila)

This can be used to avoid replication loops. This is controlled by the new `CREATE SUBSCRIPTION ... ORIGIN` option.

- Perform logical replication `SELECT` and DML actions as the table owner (Robert Haas)

This improves security and now requires subscription owners to be either superusers or to have `SET ROLE` permission on all roles owning tables in the replication set. The previous behavior of performing all operations as the subscription owner can be enabled with the subscription `run_as_owner` option.

- Have `wal_retrieve_retry_interval` operate on a per-subscription basis (Nathan Bossart)

Previously the retry time was applied globally. This also adds wait events `>LogicalRepLauncherDSA` and `LogicalRepLauncherHash`.

E.4.3.3. Utility Commands

- Add `EXPLAIN` option `GENERIC_PLAN` to display the generic plan for a parameterized query (Laurenz Albe)

- Allow a `COPY FROM` value to map to a column's `DEFAULT` (Israel Barth Rubio)

- Allow `COPY` into foreign tables to add rows in batches (Andrey Lepikhov, Etsuro Fujita)

This is controlled by the `postgres_fdw` option `batch_size`.

- Allow the `STORAGE` type to be specified by `CREATE TABLE` (Teodor Sigaev, Aleksander Alekseev)

Previously only `ALTER TABLE` could control this.

- Allow **truncate triggers** on foreign tables (Yugo Nagata)

- Allow `VACUUM` and `vacuumdb` to only process `TOAST` tables (Nathan Bossart)

This is accomplished by having `VACUUM` turn off `PROCESS_MAIN` or by `vacuumdb` using the `--no-process-main` option.

- Add `VACUUM` options to skip or update all **frozen** statistics (Tom Lane, Nathan Bossart)

The options are `SKIP_DATABASE_STATS` and `ONLY_DATABASE_STATS`.

- Change `REINDEX DATABASE` and `REINDEX SYSTEM` to no longer require an argument (Simon Riggs)

Previously the database name had to be specified.

- Allow `CREATE STATISTICS` to generate a statistics name if none is specified (Simon Riggs)

E.4.3.4. Data Types

- Allow non-decimal **integer literals** (Peter Eisentraut)

For example, `0x42F`, `0o273`, and `0b100101`.

- Allow `NUMERIC` to process hexadecimal, octal, and binary integers of any size (Dean Rasheed)

Previously only unquoted eight-byte integers were supported with these non-decimal bases.

- Allow underscores in integer and numeric **constants** (Peter Eisentraut, Dean Rasheed)

This can improve readability for long strings of digits.

- Accept the spelling `+infinity` in datetime input (Vik Fearing)

- Prevent the specification of epoch and infinity together with other fields in datetime strings (Joseph Koshakow)
- Remove undocumented support for date input in the form **`YyearMmonthDday`** (Joseph Koshakow)
- Add functions **`pg_input_is_valid()`** and **`pg_input_error_info()`** to check for type conversion errors (Tom Lane)

E.4.3.5. General Queries

- Allow subqueries in the FROM clause to omit aliases (Dean Rasheed)
- Add support for enhanced numeric literals in SQL/JSON paths (Peter Eisentraut)

For example, allow hexadecimal, octal, and binary integers and underscores between digits.

E.4.3.6. Functions

- Add SQL/JSON constructors (Nikita Glukhov, Teodor Sigaev, Oleg Bartunov, Alexander Korotkov, Amit Langote)

The new functions **`JSON_ARRAY()`**, **`JSON_ARRAYAGG()`**, **`JSON_OBJECT()`**, and **`JSON_OBJECTAGG()`** are part of the SQL standard.

- Add SQL/JSON object checks (Nikita Glukhov, Teodor Sigaev, Oleg Bartunov, Alexander Korotkov, Amit Langote, Andrew Dunstan)

The **`IS JSON`** checks include checks for values, arrays, objects, scalars, and unique keys.

- Allow JSON string parsing to use vector operations (John Naylor)
- Improve the handling of full text highlighting function **`ts_headline()`** for OR and NOT expressions (Tom Lane)
- Add functions to add, subtract, and generate **`timestampz`** values in a specified time zone (Przemyslaw Sztoch, Gurjeet Singh)

The functions are **`date_add()`**, **`date_subtract()`**, and **`generate_series()`**.

- Change **`date_trunc(unit, timestampz, time_zone)`** to be an immutable function (Przemyslaw Sztoch)

This allows the creation of expression indexes using this function.

- Add server variable **`SYSTEM_USER`** (Bertrand Drouvot)

This reports the authentication method and its authenticated user.

- Add functions **`array_sample()`** and **`array_shuffle()`** (Martin Kalcher)
- Add aggregate function **`ANY_VALUE()`** which returns any value from a set (Vik Fearing)
- Add function **`random_normal()`** to supply normally-distributed random numbers (Paul Ramsey)
- Add error function **`erf()`** and its complement **`erfc()`** (Dean Rasheed)
- Improve the accuracy of numeric **`power()`** for integer exponents (Dean Rasheed)
- Add **`XMLSERIALIZE()`** option **`INDENT`** to pretty-print its output (Jim Jones)
- Change **`pg_collation_actual_version()`** to return a reasonable value for the default collation (Jeff Davis)

Previously it returned NULL.

- Allow **`pg_read_file()`** and **`pg_read_binary_file()`** to ignore missing files (Kyotaro Horiguchi)
- Add byte specification (B) to **`pg_size_bytes()`** (Peter Eisentraut)
- Allow **`to_reg*`** functions to accept numeric OIDs as input (Tom Lane)

E.4.3.7. PL/PgSQL

- Add the ability to get the current function's OID in PL/pgSQL (Pavel Stehule)

This is accomplished with **`GET DIAGNOSTICS variable = PG_ROUTINE_OID.`**

E.4.3.8. Libpq

- Add libpq connection option **`require_auth`** to specify a list of acceptable authentication methods (Jacob Champion)

This can also be used to disallow certain authentication methods.

- Allow multiple libpq-specified hosts to be randomly selected (Jelte Fennema)

This is enabled with `load_balance_hosts=random` and can be used for load balancing.

- Add libpq option `sslcertmode` to control transmission of the client certificate (Jacob Champion)

The option values are `disable`, `allow`, and `require`.

- Allow libpq to use the system certificate pool for certificate verification (Jacob Champion, Thomas Habets)

This is enabled with `sslrootcert=system`, which also enables `sslmode=verify-full`.

E.4.3.9. Client Applications

- Allow `ECPG` variable declarations to use typedef names that match unreserved SQL keywords (Tom Lane)

This change does prevent keywords which match C typedef names from being processed as keywords in later EXEC SQL blocks.

E.4.3.9.1. `psql`

- Allow `psql` to control the maximum width of header lines in expanded format (Platon Pronko)

This is controlled by `xheader_width`.

- Add `psql` command `\drg` to show role membership details (Pavel Luzanov)

The `Member` of output column has been removed from `\du` and `\dg` because this new command displays this information in more detail.

- Allow `psql`'s access privilege commands to show system objects (Nathan Bossart)

The options are `\dpS` and `\zS`.

- Add FOREIGN designation to `psql \d+` for foreign table children and partitions (Ian Lawrence Barwick)

- Prevent `\df+` from showing function source code (Isaac Morland)

Function bodies are more easily viewed with `\sf`.

- Allow `psql` to submit queries using the extended query protocol (Peter Eisentraut)

Passing arguments to such queries is done using the new `psql \bind` command.

- Allow `psql \watch` to limit the number of executions (Andrey Borodin)

The `\watch` options can now be named when specified.

- Detect invalid values for `psql \watch`, and allow zero to specify no delay (Andrey Borodin)

- Allow `psql` scripts to obtain the exit status of shell commands and queries (Corey Huinker, Tom Lane)

The new `psql` control variables are `SHELL_ERROR` and `SHELL_EXIT_CODE`.

- Various `psql` tab completion improvements (Vignesh C, Aleksander Alekseev, Dagfinn Ilmari Mannsåker, Shi Yu, Michael Paquier, Ken Kato, Peter Smith)

E.4.3.9.2. `pg_dump`

- Add `pg_dump` control of dumping child tables and partitions (Gilles Darold)

The new options are `--table-and-children`, `--exclude-table-and-children`, and `--exclude-table-data-and-children`.

- Add LZ4 and Zstandard compression to `pg_dump` (Georgios Kokolatos, Justin Pryzby)

- Allow `pg_dump` and `pg_basebackup` to use `long` mode for compression (Justin Pryzby)

- Improve `pg_dump` to accept a more consistent compression syntax (Georgios Kokolatos)

Options like `--compress=gzip:5`.

E.4.3.10. Server Applications

- Add `initdb` option to set server variables for the duration of `initdb` and all future server starts (Tom Lane)

The option is `-c name=value`.

- Add options to **createuser** to control more user options (Shinya Kato)

Specifically, the new options control the valid-until date, bypassing of row-level security, and role membership.

- Deprecate **createuser** option `--role` (Nathan Bossart)

This option could be easily confused with new createuser role membership options, so option `--member-of` has been added with the same functionality. The `--role` option can still be used.

- Allow control of **vacuumdb** schema processing (Gilles Darold)

These are controlled by options `--schema` and `--exclude-schema`.

- Use new **VACUUM** options to improve the performance of **vacuumdb** (Tom Lane, Nathan Bossart)

- Have **pg_upgrade** set the new cluster's locale and encoding (Jeff Davis)

This removes the requirement that the new cluster be created with the same locale and encoding settings.

- Add **pg_upgrade** option to specify the default transfer mode (Peter Eisentraut)

The option is `--copy`.

- Improve **pg_basebackup** to accept numeric compression options (Georgios Kokolatos, Michael Paquier)

Options like `--compress=server-5` are now supported.

- Fix **pg_basebackup** to handle tablespaces stored in the PGDATA directory (Robert Haas)

- Add **pg_waldump** option `--save-fullpage` to dump full page images (David Christensen)

- Allow **pg_waldump** options `-t/--timeline` to accept hexadecimal values (Peter Eisentraut)

- Add support for progress reporting to **pg_verifybackup** (Masahiko Sawada)

- Allow **pg_rewind** to properly track timeline changes (Heikki Linnakangas)

Previously if `pg_rewind` was run after a timeline switch but before a checkpoint was issued, it might incorrectly determine that a rewind was unnecessary.

- Have **pg_receivewal** and **pg_recvlogical** cleanly exit on SIGTERM (Christoph Berg)

This signal is often used by systemd.

E.4.3.11. Source Code

- Build ICU support by default (Jeff Davis)

This removes **build flag** `--with-icu` and adds flag `--without-icu`.

- Add support for SSE2 (Streaming SIMD Extensions 2) vector operations on x86-64 architectures (John Naylor)

- Add support for Advanced SIMD (Single Instruction Multiple Data) (NEON) instructions on ARM architectures (Nathan Bossart)

- Have Windows binaries built with MSVC use `RandomizedBaseAddress` (ASLR) (Michael Paquier)

This was already enabled on MinGW builds.

- Prevent extension libraries from exporting their symbols by default (Andres Freund, Tom Lane)

Functions that need to be called from the core backend or other extensions must now be explicitly marked `PGDLL_EXPORT`.

- Require Windows 10 or newer versions (Michael Paquier, Juan José Santamaría Flecha)

Previously Windows Vista and Windows XP were supported.

- Require Perl version 5.14 or later (John Naylor)

- Require Bison version 2.3 or later (John Naylor)

- Require Flex version 2.5.35 or later (John Naylor)

- Require MIT Kerberos for GSSAPI support (Stephen Frost)

- Remove support for Visual Studio 2013 (Michael Paquier)

- Remove support for HP-UX (Thomas Munro)
- Remove support for HP/Intel Itanium (Thomas Munro)
- Remove support for M68K, M88K, M32R, and SuperH CPU architectures (Thomas Munro)
- Remove **libpq** support for SCM credential authentication (Michael Paquier)

Backend support for this authentication method was removed in PostgreSQL 9.1.

- Add **meson** build system (Andres Freund, Nazir Bilal Yavuz, Peter Eisentraut)

This eventually will replace the Autoconf and Windows-based MSVC build systems.

- Allow control of the location of the openssl binary used by the build system (Peter Eisentraut)

Make finding openssl program a configure or meson option

- Add build option to allow testing of small table segment sizes (Andres Freund)

The build options are **--with-segsize-blocks** and **-Dsegsize_blocks**.

- Add **pgindent** options (Andrew Dunstan)

The new options are **--show-diff**, **--silent-diff**, **--commit**, and **--help**, and allow multiple **--exclude** options. Also require the typedef file to be explicitly specified. Options **--code-base** and **--build** were also removed.

- Add **pg_bsd_indent** source code to the main tree (Tom Lane)
- Improve **make_ctags** and **make_etags** (Yugo Nagata)
- Adjust **pg_attribute** columns for efficiency (Peter Eisentraut)

E.4.3.12. Additional Modules

- Improve use of extension-based indexes on boolean columns (Zongliang Quan, Tom Lane)
- Add support for Daitch-Mokotoff Soundex to **fuzzystrmatch** (Dag Lem)
- Allow **auto_explain** to log values passed to parameterized statements (Dagfinn Ilmari Mannsåker)

This affects queries using server-side **PREPARE/EXECUTE** and client-side parse/bind. Logging is controlled by **auto_explain.log_parameter_max_length**; by default query parameters will be logged with no length restriction.

- Have **auto_explain**'s **log_verbose** mode honor the value of **compute_query_id** (Atsushi Torikoshi)

Previously even if **compute_query_id** was enabled, **log_verbose** was not showing the query identifier.

- Change the maximum length of **ltree** labels from 256 to 1000 and allow hyphens (Garen Torikian)
- Have **pg_stat_statements** normalize constants used in utility commands (Michael Paquier)

Previously constants appeared instead of placeholders, e.g., \$1.

- Add **pg_walinspect** function **pg_get_wal_block_info()** to report WAL block information (Michael Paquier, Melanie Plageman, Bharath Rupireddy)

- Change how **pg_walinspect** functions **pg_get_wal_records_info()** and **pg_get_wal_stats()** interpret ending LSNs (Bharath Rupireddy)

Previously ending LSNs which represent nonexistent WAL locations would generate an error, while they will now be interpreted as the end of the WAL.

- Add detailed descriptions of WAL records in **pg_walinspect** and **pg_waldump** (Melanie Plageman, Peter Geoghegan)

- Add **pageinspect** function **bt_multi_page_stats()** to report statistics on multiple pages (Hamid Akhtar)

This is similar to **bt_page_stats()** except it can report on a range of pages.

- Add empty range output column to **pageinspect** function **brin_page_items()** (Tomas Vondra)

- Redesign archive modules to be more flexible (Nathan Bossart)

Initialization changes will require modules written for older versions of Postgres to be updated.

- Correct inaccurate **pg_stat_statements** row tracking extended query protocol statements (Sami Imseih)

- Add **pg_buffercache** function `pg_buffercache_usage_counts()` to report usage totals (Nathan Bossart)
- Add **pg_buffercache** function `pg_buffercache_summary()` to report summarized buffer statistics (Melih Mutlu)
- Allow the schemas of required extensions to be referenced in extension scripts using the new syntax `@extschema:referenced_extension_name@` (Regina Obe)
- Allow required extensions to be marked as non-relocatable using **no_relocate** (Regina Obe)

This allows `@extschema:referenced_extension_name@` to be treated as a constant for the lifetime of the extension.

E.4.3.12.1. **postgres_fdw**

- Allow `postgres_fdw` to do aborts in parallel (Etsuro Fujita)

This is enabled with `postgres_fdw` option **parallel_abort**.

- Make **ANALYZE** on foreign `postgres_fdw` tables more efficient (Tomas Vondra)

The `postgres_fdw` option **analyze_sampling** controls the sampling method.

- Restrict shipment of **reg*** type constants in `postgres_fdw` to those referencing built-in objects or extensions marked as shippable (Tom Lane)
- Have `postgres_fdw` and **dblink** handle interrupts during connection establishment (Andres Freund)

E.4.4. Acknowledgments

The following individuals (in alphabetical order) have contributed to this release as patch authors, committers, reviewers, testers, or reporters of issues.

Abhijit Menon-Sen
Adam Mackler
Adrian Klaver
Ahsan Hadi
Ajin Cherian
Ajit Awekar
Alan Hodgson
Aleksander Alekseev
Alex Denman
Alex Kozhemyakin
Alexander Korolev
Alexander Korotkov
Alexander Lakhin
Alexander Pyhalov
Alexey Borzov
Alexey Ermakov
Alexey Makhmutov
Álvaro Herrera
Amit Kapila
Amit Khandekar
Amit Langote
Amul Sul
Anastasia Lubennikova
Anban Company
Andreas Dijkman
Andreas Karlsson
Andreas Scherbaum
Andrei Zubkov
Andres Freund
Andrew Alsup
Andrew Bille
Andrew Dunstan
Andrew Gierth
Andrew Kesper
Andrey Borodin
Andrey Lepikhov

Andrey Sokolov
Ankit Kumar Pandey
Ante Kresic
Anton Melnikov
Anton Sidyakin
Anton Voloshin
Antonin Houska
Arne Roland
Artem Anisimov
Arthur Zakirov
Ashutosh Bapat
Ashutosh Sharma
Asim Praveen
Atsushi Torikoshi
Ayaki Tachikake
Balazs Szilfai
Benoit Lobréau
Bernd Helmle
Bertrand Drouvot
Bharath Rupireddy
Bilva Sanaba
Bob Krier
Boris Zentner
Brad Nicholson
Brar Piening
Bruce Momjian
Bruno da Silva
Carl Sopchak
Cary Huang
Changhong Fei
Chris Travers
Christoph Berg
Christophe Pettus
Corey Huinker
Craig Ringer
Curt Kolovson
Dag Lem
Dagfinn Ilmari Mannsåker
Daniel Gustafsson
Daniel Vérité
Daniel Watzinger
Daniel Westermann
Daniele Varrazzo
Daniil Anisimov
Danny Shemesh
Dave Page
David Christensen
David G. Johnston
David Geier
David Gilman
David Kimura
David Rowley
David Steele
David Turon
David Zhang
Davinder Singh
Dean Rasheed
Denis Laxalde
Dilip Kumar
Dimos Stamatakis
Dmitriy Kuzmin
Dmitry Astapov
Dmitry Dolgov

Dmitry Koval
Dong Wook Lee
Dongming Liu
Drew DeVault
Duncan Sands
Ed Maste
Egor Chindyaskin
Ekaterina Kiryanova
Elena Indrupsкая
Emmanuel Quincerot
Eric Mutta
Erik Rijkers
Erki Eessaar
Erwin Brandstetter
Etsuro Fujita
Eugeny Zhuzhnev
Euler Taveira
Evan Jones
Evgeny Morozov
Fabrício de Royes Mello
Farias de Oliveira
Florin Irion
Franz-Josef Färber
Garen Torikian
Georgios Kokolatos
Gilles Darold
Greg Stark
Guillaume Lelarge
Gunnar Bluth
Gunnar Morling
Gurjeet Singh
Haiyang Wang
Haiying Tang
Hamid Akhtar
Hans Buschmann
Hao Wu
Hayato Kuroda
Heath Lord
Heikki Linnakangas
Himanshu Upadhyaya
Hisahiro Kauchi
Hongyu Song
Hubert Lubaczewski
Hung Nguyen
Ian Barwick
Ibrar Ahmed
Ilya Gladyshev
Ilya Nenashev
Isaac Morland
Israel Barth Rubio
Jacob Champion
Jacob Speidel
Jaime Casanova
Jakub Wartak
James Coleman
James Inform
James Vanns
Jan Wieck
Japin Li
Jeevan Ladhe
Jeff Davis
Jeff Janes
Jehan-Guillaume de Rorthais

Jelte Fennema
Jian He
Jim Jones
Jinbao Chen
Joe Conway
Joel Jacobson
John Naylor
Jonathan Katz
Josef Simanek
Joseph Koshakow
Juan José Santamaría Flecha
Julien Rouhaud
Julien Roze
Junwang Zhao
Justin Pryzby
Justin Zhang
Karina Litskevich
Karl O. Pinc
Keisuke Kuroda
Ken Kato
Kevin McKibbin
Kieran McCusker
Kirk Wolak
Konstantin Knizhnik
Koshi Shibagaki
Kotaro Kawamoto
Kui Liu
Kyotaro Horiguchi
Lakshmi Narayanan Sreethar
Laurence Parry
Laurenz Albe
Luca Ferrari
Lukas Fittl
Maciek Sakrejda
Magnus Hagander
Maja Založnik
Marcel Hofstetter
Marina Polyakova
Mark Dilger
Marko Tiikkaja
Markus Winand
Martijn van Oosterhout
Martin Jurca
Martin Kalcher
Mary Xu
Masahiko Sawada
Masahiro Ikeda
Masao Fujii
Mason Sharp
Matheus Alcantara
Mats Kindahl
Matthias van de Meent
Matthijs van der Vleuten
Maxim Orlov
Maxim Yablokov
Mehmet Emin Karakas
Melanie Plageman
Melih Mutlu
Micah Gates
Michael Banck
Michael Paquier
Michail Nikolaev
Michel Pelletier

Mike Oh
Mikhail Gribkov
Mingli Zhang
Miroslav Bendik
Mitsuru Hinata
Myo Wai Thant
Naeem Akhter
Naoki Okano
Nathan Bossart
Nazir Bilal Yavuz
Neha Sharma
Nick Babadzhanian
Nicola Contu
Nikhil Shetty
Nikita Glukhov
Nikolay Samokhvalov
Nikolay Shaplov
Nishant Sharma
Nitin Jadhav
Noah Misch
Noboru Saito
Noriyoshi Shinoda
Nuko Yokohama
Oleg Bartunov
Oleg Tselebrovskiy
Olly Betts
Onder Kalaci
Onur Tirtir
Pablo Federico
Palle Girgensohn
Paul Guo
Paul Jungwirth
Paul Ramsey
Pavel Borisov
Pavel Kulakov
Pavel Luzanov
Pavel Stehule
Peifeng Qiu
Peter Eisentraut
Peter Geoghegan
Peter Smith
Phil Florent
Philippe Godfrin
Platon Pronko
Przemyslaw Sztoch
Rachel Heaton
Ranier Vilela
Regina Obe
Reid Thompson
Reiner Peterke
Richard Guo
Riivo Kolka
Rishu Bagga
Robert Haas
Robert Sjöblom
Robert Treat
Roberto Mello
Robins Tharakan
Roman Zharkov
Ronan Dunklau
Rushabh Lathia
Ryo Matsumura
Samay Sharma

Sami Imseih
Sandeep Thakkar
Sandro Santilli
Sebastien Flaesch
Sébastien Lardière
Sehrope Sarkuni
Sergey Belyashov
Sergey Pankov
Sergey Shinderuk
Shi Yu
Shinya Kato
Sho Kato
Shruthi Gowda
Shveta Mallik
Simon Riggs
Sindy Seniorita
Sirisha Chamarthi
Sravan Kumar
Stéphane Tachaires
Stephen Frost
Steve Chavez
Stone Tickle
Sven Klemm
Takamichi Osumi
Takeshi Ideriha
Tatsuhiko Nakamori
Tatsuo Ishii
Teja Mupparti
Tender Wang
Teodor Sigaev
Thiago Nunes
Thom Brown
Thomas Habets
Thomas Mc Kay
Thomas Munro
Tim Carey-Smith
Tim Field
Timo Stolz
Tom Lane
Tomas Vondra
Tor Erik Linnerud
Torsten Förtsch
Tristan Partin
Troy Frericks
Tushar Ahuja
Valerie Woolard
Vibhor Kumar
Victor Spirin
Victoria Shepard
Vignesh C
Vik Fearing
Vitaly Burovoy
Vitaly Davydov
Wang Wei
Wenjing Zeng
Whale Song
Will Mortensen
Wolfgang Walther
Xin Wen
Xing Guo
Xingwang Xu
Xuejing Zhao
Yanliang Lei

Youmiu Mo
Yugo Nagata
Yura Sokolov
Yuta Katsuragi
Zhen Mingyang
Zheng Li
Zhihong Yu
Zhijie Hou
Zongliang Quan
Zuming Jiang

Submit correction

If you see anything in the documentation that is not correct, does not match your experience with the particular feature or requires further clarification, please use **this form** to report a documentation issue.

